(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
18 April 2013 (18.04.2013)

WIPO | PCT

(10) International Publication Number
WO 2013/056104 A1

(51) International Patent Classification:
H04W 12/00 (2009.01) *G06Q 40/02* (2012.01)
G06F 9/44 (2006.01) *G06Q 20/40* (2012.01)

(21) International Application Number:
PCT/US2012/060038

(22) International Filing Date:
12 October 2012 (12.10.2012)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
61/546,084 12 October 2011 (12.10.2011) US
61/619,751 3 April 2012 (03.04.2012) US

(71) Applicant: **C-SAM, INC.** [US/US]; One Tower Lane, Suite 2535, Oakbrook Terrace, Illinois 60181 (US).

(72) Inventors; and
(71) Applicants : **DESAI, Mehul** [US/US]; 573 Hamilton Avenue, Westmont, Illinois 60559 (US). **PITRODA, Satyan** [IN/US]; 301 Trinity Lane, Oak Brook, Illinois 60523 (US). **MANIAR, Nehal** [US/US]; 2 Concord Dr., Oak Brook, Illinois 60523 (US).

(74) Agent: **AMBROZIAK, Jeffrey**; GTC Law Group LLP & Affiliates, c/o CPA Global, P.O. Box 52050, Minneapolis, Minnesota 55402 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

— *with international search report (Art. 21(3))*

— *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments (Rule 48.2(h))*

(54) **Title**: A MULTI-TIERED SECURE MOBILE TRANSACTIONS ENABLING PLATFORM

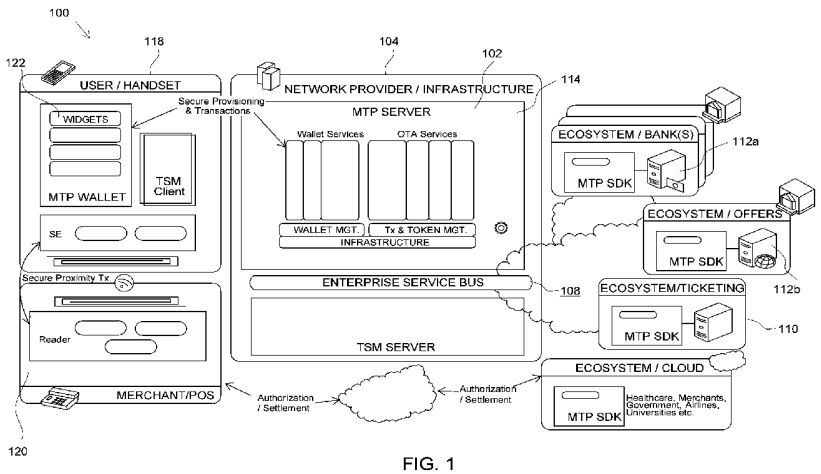


FIG. 1

(57) **Abstract**: A platform for performing secure personalized transactions in a multi-domain ecosystem includes a personalization tier that enables service provider personalization for one or more ecosystem elements stored on a mobile device. Further, the platform includes an enabling tier for facilitating interoperation between the personalization tier and a client device. The platform further includes a service tier that may be operating independently of the enabling tier and may enable service delivery for a plurality of services.

WO 2013/056104 A1

A MULTI-TIERED SECURE MOBILE TRANSACTIONS ENABLING PLATFORM

CROSS-REFERENCE TO RELATED APPLICATIONS

[001] This application is a continuation-in-part of U.S. Pat. Appl. 11539024, filed October 5, 2006, entitled "Transactional Services", presently pending. That application claims the benefit of U.S. Provisional Patent Application No. 60/724,066, filed October 5, 2005, entitled "Methods and Systems for Transactional Services".

[002] This application claims the benefit of U.S. provisional patent application 61546084 filed October 12, 2011 and U.S. provisional patent application 61619751 filed April 3, 2012.

[003] This application is related to the following U.S. patent applications: U.S. Pat. App. No. 11/931,872, filed October 31, 2007, entitled "System and methods for servicing electronic transactions"; and U.S. Pat. App. No. 10/284,676, filed October 31, 2002, entitled "System and methods for servicing electronic transactions".

[004] Each of the foregoing applications is incorporated herein by reference in its entirety.

[005] This application is related to the following U.S. patents, each of which is incorporated herein by reference in its entirety: U.S. Pat. No. 6,769,607, filed June 6, 2000, entitled "Point of sale and display adapter for electronic transaction device"; U.S. Pat. 6,705,520, filed November 15, 1999, entitled "Point of sale adapter for electronic transaction device"; U.S. Pat. No. 5,590,038, filed June 20, 1994, entitled "Universal electronic transaction card including receipt storage and system and methods of conducting electronic transactions"; U.S. Pat. No. 5,884,271, filed September 6, 1996, entitled "Device, system and methods of conducting paperless transactions device", U.S. Pat. No. 6,925,439, filed March 10, 1999, entitled "Device, system and methods of conducting paperless transactions"; U.S. Pat. No. 7,308,426, filed August 11, 1999, entitled "System and methods for servicing electronic transactions"; and U.S. Pat. No. 7,366,990, filed January 19, 2001, entitled "Method and system for managing user activities and information using a customized computer interface".

BACKGROUND

[006] Field:

[007] The methods and systems of a multi-tier mobile transaction platform for secure personalized transactions in a multi-domain ecosystem generally relates to mobile transaction processing security.

[008] Description of the Related Art:

[009] Mobile transactions are becoming the most prevalent form of point of sale payments. Existing systems for mobile transaction processing do not have the scalability and flexibility of configurability needed to address the ever growing demand for secure personalized transaction services and client contact to meet future business needs.

SUMMARY

[0010] A mobile transaction platform (MTP) may comprise a carrier-grade, infrastructure platform that may facilitate the development and delivery of secure transactional services, both mCommerce (e.g. over-the-air) and mPayment (e.g. Near Field Communication (NFC) Proximity), to a broad range of mobile phone environments. A MTP may include an enhanced software development methodology to further simplify development and hide phone environment-specific complexities from the developer by providing a consistent, common development environment. Native phone features may also be exposed during development to allow developers to define native phone experiences and the like while still benefitting from a common and robust development methodology and environment. One benefit of using the MTP is rapid mobile service development for a broad range of mobile handsets to quickly build a profitable mobile transaction ecosystem.

[0011] In addition, disclosed herein is an end-to-end secure transaction environment in which a trust model is applied that ensures not only secure electronic transactions but provides secure and safe electronic business operations among users, payment facilities, and merchants through the use of dynamic authentication of each user, each device operated by a user, and each transaction and/or service domain. The disclosed trust model ensures that, for example, users and merchants can conduct transactions without the use of "middle men" to ensure safe and reliable commerce. By authenticating a user

through one or more main service facilities and by authenticating merchants and other providers through such main service facilities it is possible for each merchant to have a unique yet secure relationship with each user through any device on which the user has been authenticated to the main service facility. However, through use of a main service facility for authentication and enabling a robust trust model, multiple merchants (e.g. providers of products, financial services, and many other services described herein) may securely and directly transact with a single user. In this way a plurality of authenticated merchants may be available to the user in an aggregated portfolio that may facilitate greatly improved secure electronic commerce.

[0012] Aspects of an e-business environment may include a provider domain (e.g. banks, merchants, etc) that may be accessed through a wallet platform and provides the ability to build out wallet and widget capability to enable looking at an electronic transaction environment from a risk management perspective. In an example, cross selling across the ecosystems may take advantage of the methods and systems described herein thereby allowing mapping of risk management into the e-commerce process.

[0013] Aspects of a secure multi-domain ecosystem environment may extend the environment to the consumer. Once a consumer is authenticated into such an environment one may have the ability to establish trust models that may enable delivery of a particular instance of services by a single trust model. Other products/services may be delivered by the use of other trust models.

[0014] Also, the methods and systems herein may facilitate extending a security tunnel to merchants that may be embodied as a tunnel within a tunnel, which may enable a core security architecture. In this way businesses can avoid getting bogged down with security rather than focusing on applications that will provide a sustainable business model. In an example, the methods and systems herein may enable a business person to look at the frequency of use of applications (e.g., 3x/day vs. 3x/month) to help envision a significantly different trust model for each of a range of different user interaction models.

[0015] These and other systems, methods, objects, features, and advantages of the present invention will be apparent to those skilled in the art from the following detailed description of the preferred embodiment and the drawings. All documents mentioned herein are hereby incorporated in their entirety by reference.

BRIEF DESCRIPTION OF THE FIGURES

[0016] The invention and the following detailed description of certain embodiments thereof may be understood by reference to the following figures:

[0017] FIG. 1 illustrates a dynamic ecosystem 100 for secure mCommerce and mPayment transactions.

[0018] FIG. 2 illustrates a mobile transaction platform (MTP) high level structure.

[0019] FIG. 3 illustrates mobile transaction platform (MTP) architecture.

[0020] FIG. 4 illustrates a high level architectural view of a wallet application with a MTP mClient runtime.

[0021] FIG. 5 illustrates an overview of a widget development process with reference to a development cycle.

[0022] FIG. 6 illustrates an overview of an ecosystem for a secure personalized transaction.

[0023] FIG. 7 illustrates an exemplary client architecture for use in an ecosystem for secure personalized transactions.

[0024] FIG. 8 illustrates an ecosystem for secure personalized transactions including an MTP structural configuration with a protocol abstraction support capability.

[0025] FIG. 9 illustrates an alternate configuration of an ecosystem for secure personalized transactions as illustrated in the FIG. 8.

[0026] FIG. 10 illustrates a notion of a tiered platform that may span a range of ecosystem elements.

[0027] FIG. 11 illustrates an embodiment of selecting ecosystem features for inclusion in each of an enabling tier, a service tier, and a personalization tier of a multi-domain ecosystem platform for secure personalized transactions.

[0028] FIG. 12 illustrates an expert engine for facilitating modifying each of a personalization and a service tier for each of a plurality of service providers in a multi-domain ecosystem for secure personalized transactions.

[0029] FIG. 13 illustrates an embodiment of selecting business needs and device functionality for inclusion in each of an enabling tier, a service tier, and a personalization tier of a multi-domain ecosystem platform for secure personalized transactions.

[0030] FIG. 14 illustrates a method of developing a widget.

[0031] FIG. 15 illustrates a method of widget usage.

[0032] FIG. 16 illustrates a transaction flow diagram for installing a widget.

[0033] FIG. 17 illustrates a state diagram depicting a lifecycle of a widget in a MTP system.

[0034] FIG. 18 illustrates a widget lifecycle from development to execution.

[0035] FIG. 19 illustrates a high level view of an access control mechanism.

[0036] FIG. 20 illustrates a method for loading and execution of a widget with respect to the access controller.

[0037] FIG.21 illustrates a widget runtime footprint based on a widget access control policy.

[0038] FIG. 22 illustrates a widget approval process.

[0039] FIG. 23 illustrates an exemplary and non-limiting embodiment of widget architecture.

[0040] FIG. 24 illustrates a high level architecture view of a client-device container.

[0041] Fig. 24A illustrates a detailed view of mobile transaction platform related elements of a mobile device.

[0042] Fig. 24B illustrates wallet companion domain features, and control and data flow.

[0043] FIG. 25 illustrates an offer or coupon-related transaction flow diagram.

[0044] FIG. 26 illustrates a state machine diagram of the coupon applet during the transaction flow according to an example.

[0045] FIG. 27 illustrates an interaction between an applet and a Point of sale terminal.

[0046] FIG. 28 illustrates a state machine diagram of the coupon applet during the transaction flow according to an example.

[0047] FIG. 29 illustrates a high level architecture of a system for a deployment of the present invention in an exemplary embodiment.

[0048] FIG. 30 illustrates a flow chart describing a method of installing the mobile wallet on the mobile phone of the user using SMS with a download URL.

[0049] FIGs. 31A and 31B illustrates a method for wallet activation by the user.

[0050] FIG. 32 illustrates a method as an example of the search and download use case.

[0051] FIG. 33 illustrates an exemplary method for the OTA-transaction.

[0052] FIG. 34 illustrates a method as an example for the card issuance use case.

[0053] FIG. 35 illustrates a method as an example to terminate a wallet.

[0054] FIG. 36 illustrates a method as an example for publishing a widget.

[0055] FIG. 37 is a method for adding a widget for the first time along with a TSM interaction..

[0056] FIG.38 is a behavioral model of a system for widget access verification with reference to the widget loading time signature, verification, and execution time access verification.

[0057] FIG. 39 illustrates a dynamic ecosystem for secure commerce and payment transactions.

[0058] FIG. 40 illustrates a method of protocol abstraction in a dynamic ecosystem for secure commerce and payment transactions.

[0059] FIG. 41 illustrates a method of tokenization of transaction content in a dynamic ecosystem for secure commerce and payment transactions.

[0060] FIG. 42 illustrates a point-of-sale device to receive a dynamic token from a mobile wallet running on a mobile device.

[0061] FIG. 43 illustrates a method of generating a token for an electronic transaction wrapper for secure content traversing a network in a dynamic ecosystem for secure commerce and payment transactions.

[0062] FIG. 44 illustrates a method of tokenization of a transaction wrapper in a dynamic ecosystem for secure commerce and payment transactions.

[0063] FIG. 45 illustrates a method of performing electronic transactions across an open network by transmitting a dynamically generated tokenized wrapper that includes a token representing transaction content in a dynamic ecosystem for secure commerce and payment transactions.

[0064] FIG. 46 illustrates a method of protocol abstraction in transaction flow that includes multiple acquisition methods with a single issuing bank.

[0065] FIG. 47 illustrates a method of protocol abstraction in over-the-air transaction flow that includes single acquisition method and multiple issuing banks.

[0066] FIG. 14 illustrates a method of tokenized transactions, tokenized protocol abstraction, and tokenized wrapper.

[0067] FIG. 49 illustrates an single trust domain model..

[0068] FIG. 50 illustrates the single trust cluster model as illustrated in FIG. 50, which may assume that a trusted third part is accessible in the ecosystem.

[0069] FIG. 51 illustrates a multiple trust cluster model.

[0070] FIG. 52 illustrates a direct trust relationship where no two entities are compelled to trust each other.

[0071] FIG. 53 illustrates message bus communication architecture.

[0072] FIG. 54 illustrates a high level block diagram of various subsystems of a payment facilitator system.

[0073] FIG. 55 shows various components of the wallet.

[0074] FIG. 56 illustrates a high level block diagram of the deployment model 5600 maybe based in a payment facilitator container.

[0075] While described herein with reference to various embodiments, it is understood that, in all cases, unless otherwise specified, references to an "embodiment" or "embodiments" refer to one or more exemplary and non-limiting embodiments. Also, it is understood that, in all descriptions herein, unless otherwise specified, even when not explicitly being referenced to an “embodiment” or “embodiments” refer to one or more exemplary and non-limiting embodiments.

[0076] A multi-domain ecosystem may be configured for secure personalized transactions via a multi-tiered secure mobile transactions enabling platform that may comprise an enabling tier, a service tier, a personalization tier, an expert tier, and an experience framework tier. A multi-domain ecosystem may include a plurality of service providers, instrument issuers, mobile transaction processors, transaction acquirers, third-party authenticators, and a plurality of users accessing the ecosystem via a diversity of electronic devices. A multi-tiered secure mobile transactions ecosystem-enabling platform as described herein may facilitate any ecosystem participant to configure aspects of the ecosystem to satisfy a set of unique business model needs. The multi-tiered secure mobile transactions ecosystem enabling platform may offer software development and element configuration tools to prepare and personalize electronic

wallets, active device widgets, back-end infrastructure runtime code, and the like that target specific business model and/or trust model needs. By facilitating secure, trusted, ecosystem operations (e.g. issuance, maintenance, and use of electronic wallets) the platform may support securely transacting via a single mobile device with a large number of distinct domains. By providing tools to personalize functional elements (e.g. widgets) for deployment in mobile devices, the platform may facilitate, secure, customized transaction-related communication between the plurality of distinct domains (e.g. distinct service providers) and a user, such as a customer, that may increase opportunities for interacting with the customer by a factor of ten or more.

[0077] A multi-tiered secure mobile transactions enabling platform, which may alternatively be coined a mobile transaction platform or MTP, may deliver secure transaction functionality for users, enabling an ecosystem of participants which include service providers, issuers / acquirers of instruments and services, related support and value added service providers, and distribution networks, among others. A multi-tiered secure mobile transaction enabling platform may be delivered through well structured software and/or hardware modules focused on delivering capabilities such as direct service provider to user communication, messaging and notification, security and authentication, over-the-air (OTA) provisioning, OTA trusted service manager (TSM) proxy operation, client-based secure ecosystem container operation, wallet and widget management, transaction engine operation, digital instrument management, location and/or geo-fencing support, augmented reality, and the like.

[0078] The methods and systems of a multi-tier mobile transaction platform for secure personalized transactions in a multi-domain ecosystem may facilitate establishing and operating a market place or exchange of transaction-related credentials. These credentials may be tokenized or may not be tokenized, subject to the requirement of their respective issuers and acquirers, related to payment and non-payment services, and the like. These credentials may be transacted over-the-air and in proximity environments over host of channels, such as point of sale terminals, and the like. By employing these methods and systems that may enable such a transaction-related credential exchange, a key business problem related to scaling such an ecosystem may be solved. This exchange of transaction-related credentials may completely remove the need for a single

service provider to embed themselves as the effective exchange of such transaction credentials, and rather allow for existing issuers, acquirers and service providers (and/or consumers of such credentials) to continue to leverage their existing systems, business, and risk-management practices. Such an exchange will add value by leveraging the aspects of the mobile transaction platform described herein including various deployment models, configurable ecosystem rules, underlying aggregation security architecture and configurable security trust models, necessary tools, utilities, web-services, mobile device and server applications (e.g. wallets, containers, widgets, wallet companion domains), and the like. The result may be high quality service delivery in a highly scalable enterprise environment for ensuring the different ecosystem participants can be integrated and new participants can be added without burdensome overhead.

[0079] FIG. 1 illustrates a dynamic ecosystem 100 for secure mCommerce and mPayment transactions. The ecosystem 100 may be described herein. The MTP 114 may be configured to power a dynamic ecosystem for secure mCommerce and mPayment transactions as described below. The MTP may be set up in a network provider's domain 102 and integrated into a system of the network provider 104. In an example, an enterprise service bus (ESB) 108 of the network provider 104 may be used for integration. The MTP 114 may be connected to an ecosystem of service and security providers 110, which may include one or more trusted service managers (TSMs), for example, 112a, 112b. The network provider 104 may use the MTP 114 to create a branded mobile application targeting a specific business vertical. In an example, the mobile application may be a mWallet for the retail/financial domain or a mHealth application for the healthcare vertical. In an example, the mobile application may provide a core set of services to a user, specific to a domain for which it may have been created. For the purpose of this invention, the mobile application may be a wallet assuming a retail domain application. In an example, the wallet may also facilitate, application lifecycle and security, standardized user experience and widget management responsibilities. The widget may provide specific business services and complies with the wallet guidelines. The mobile application may be downloaded onto a user handset 118. The ecosystem of services 110 may be configured to authenticate any transaction done over the handset 118. In an example, the user handset 118 may be used at a merchant

terminal 120. The merchant terminal 120 may be configured to receive communication from the ecosystem cloud 110 regarding the widget authentication of the user handset 118 widget 122, when used in vicinity of the merchant terminal 120 for effecting a transaction.

[0080] In an example, individual, independent service providers may use the MTP SDK to build widgets that may be loaded into the wallet. These widgets may provide incremental, functionality that may enhance the overall appeal of the mobile application. The wallet and the widgets may provide the users with OTA value added services (VAS) and proximity NFC services. The container (or MTP mClient runtime) may provide the wallet and the widgets a secure runtime environment and the services to communicate OTA with the server (OTA services) and also communicate with the secure element to manage payment and non-payment applets for proximity NFC transactions.

[0081] In an example, the widgets may have an associated applet running in the secure element. In an example, the widgets may not have an associated applet running in the secure element. In an example, the service providers may only choose to provide only OTA services. In an example, the MTP and TSMs may collaborate together to provide the entire NFC experience. In an example, the wallet and the widget (including soft card) provisioning, setup and management may be handled by the MTP, while applet issuance and personalization may be done by the TSM. In an example, the network provider may ensure that the NFC readers at merchant locations may be equipped to handle the proximity protocols that may be used by the various applets that may have been loaded into the phone's SE. These readers may then process the transaction through an acquiring network that may switch the transaction to the appropriate ecosystem service provider in the cloud. The wallet and/or the widget may communicate with the ecosystem cloud to determine service status and various other VAS requirements like balance, transaction history, stored value top-up, and the like.

[0082] In exemplary and non-limiting embodiments, the MTP may be configured for providing the wallet, the widget, and manage the transaction along with all the relevant business services that may be associated with delivering wallets to a universe of diverse mobile environments. In exemplary and non-limiting embodiments, the MTP may be configured for providing an improved SDK that may to help build a wallet and ecosystem

of service providers that may create widgets with elaborate feature sets without having to worry about handset fragmentation. In exemplary and non-limiting embodiments, the MTP may be configured for establishing connectors with various ecosystems to deliver a dynamic user experience and diverse set of services to the end user that may be accessed irrespective of time and geographical location.

[0083] The MTP may be deployed for customers from various countries and may be tailored to specific country requirements. For example, the MTP may be used in The United States of America (USA), Japan, China, India, Singapore, Mexico, and Bolivia, for a variety of mCommerce and mPayments services. In exemplary and non-limiting embodiments, the MTP may be configured for providing banking services, ticketing services, and payment services.

[0084] In exemplary and non-limiting embodiments, the MTP may be deployed in a cloud and may be configured to aggregate various disparate service providers, to effectively provide end users with a suite of personalized secure transactional services. In an example, the network providers may deploy the enabling layer of the MTP (for example, the wallet and the transaction management along with infrastructure components), which may be agnostic to the services and channels. The enabling layer of the MTP may be configured to support an ecosystem of service providers for example, banks, merchants, healthcare providers, and the like. The enabling layer of the MTP may be configured to support all third party developers that may use the SDK to build their individual personalized applications for example, secure transaction applications.

[0085] FIG. 2 illustrates a MTP high level structure 200. A MTP 212 may be a client server platform. The client may operate on various mobile handsets and the server may operate in the service provider's "cloud". The client and the server may follow the high level architecture as shown in FIG. 1. The high level architecture may have three distinct components that may be required to build out an ecosystem. The distinct components may include a mobile enabling tier, a service tier and a software development kit (SDK). Each of the distinct components may interact with the client and the server. In exemplary and non-limiting embodiments, the mobile enabling tier may include infrastructure or business independent components that may be necessary to develop mobile applications. In exemplary and non-limiting embodiments, the service tier may include a collection of

reference service implementations for wallet ecosystems and value added services. In exemplary and non-limiting embodiments, the SDK may include a set of tools and APIs that may be necessary to develop the wallet and the widget services (described in this document) and build a comprehensive ecosystem.

[0086] In exemplary and non-limiting embodiments, the mobile enabling tier may include different components for the client and for the server. For example, the mobile enabling tier for the client may include infrastructure runtime container software (SW) built using the mobile phone's specific development environment. In an example, the mobile enabling tier may provide the client a common abstraction to underlying phone capabilities with common Developer APIs, UI Interpretation and rendering engine, business execution environment, widget and wallet lifecycle management, Patterns for OTA/proximity transactions, etc. In an example, the mobile enabling tier may provide the server infrastructure software deployed in network provider's domain that may be configured for establishing and managing a secure mobile transactional ecosystem. In an example, the mobile enabling tier may provide the server with software equipped for wallet management, stateful transaction engine, object lifecycle management, channel specific communication, multi-domain security policy and risk management, OTA provisioning, and the like.

[0087] In exemplary and non-limiting embodiments, the service tier may include different components for the client and for the server. In an example, the service tier for the client may include a library of common platform services that may be used to implement a plurality of business use cases. The common platform services may run within the business execution environment provided by the enabling tier. In an example, the service tier may include three basic units. A first unit may be a use case, for example, a client side workflows and templates built for the various transaction services. A second unit may be an instrument, for example, a corresponding single wallet instrument that may facilitate mapping to real world instruments. In an example, the instrument may provide data for a visual or a display. In an example, the instrument may provide data for an over-the-air (OTA) transaction. In an example, the instrument may provide data for a proximity transaction. In an example, the instrument may provide data for an administration. A third unit may include at least one trust model. The trust model may

facilitate implementation for authentication schemes, which may include a tow factor authentication (2FA). The 2FA may be required by different widget providers and may be relevant for the ecosystem.

[0088] In an example, the service tier for the server may include a library of business specific components and associated service templates that may be partially or entirely reused to develop a customer centric offering. In an example, the services may include wallet services 202 and value added services 204. The wallet services 202 may include managing the lifecycle of the wallet and payment tokens. The value added services may include providing business services. The service tier may include three units. A first unit may be a use case, for example, the server side template may be built using the mobile enabling tier's transactional framework for implementing a specific use case. In an example, there may be one or more varying templates for a business service (example, bill payment, movie ticketing, and the like). A second unit may include at least one instrument or token. The instruments may include business specific data objects (for example, credit card, bank account, bill, coupons, etc.) that may be reused across various services and may provide a commonly understood definition for the object and implementation for the rules governing the object's lifecycle and transaction. A third component may include at least one trust model. The trust model may facilitate implementation for different authentication schemes for the services, including support to operate within individual trust clusters with their independent TSMs.

[0089] In exemplary and non-limiting embodiments, the SDK may include different components for the client and for the server. In an example, the SDK for the client may be divided into 2 parts. A first part may be a wallet SDK 208 and a second part may be a widget SDK 210. In an example, the wallet SDK 208 may provide access to the wallet services 202, privileged APIs, and the necessary components to build the wallet container application or user experience. In an example, the widget SDK 210 may facilitate the ecosystem partners to build their widgets that may operate within the wallet container. In exemplary and non-limiting embodiments, the SDK may essentially facilitate client side configuration and user experience. In an example, the SDK may facilitate configuration by facilitating customizability for parameters exposed by the client container. This may include device specific parameters, variables (like PIN size, exit confirmation, selecting

inclusion of modules, etc.). In an example, the SDK may provide user experience by facilitating the development of differentiating user experiences across handsets.

[0090] In an example, the SDK for the server may include at least one library that may allow any ecosystem server to communicate with the wallet, format content for the widgets, and expose services to the wallet application. The library may include a configuration module and an interface module. The configuration module may allow customizability for parameters exposed by the server's mobile enabling and service tier. The parameters may be configured in property files and resource bundles. The interface module may provide connectors developed to facilitate communication with various backend systems that may collaborate to fulfill the complete end to end service. The interface module may also include web services exposed by the platform to integrate peripheral systems like customer care, user portals, and the like.

[0091] SDK vision- In an example, the SDK may refer to a collective set of tools and the underlying platform's APIs, utilities, patterns, interfaces, XML definitions, abstractions, and the like that may be used and/or followed to facilitate creation of a mobile client application or widget that runs within the MTP 212. The SDK's may facilitate development of the application without requiring in depth knowledge of the MTP or underlying phone environments. In an example, the complexities of secure, reliable mobile transactions may be designed so as to be "hidden under the hood" of the mobile enabling tier.

[0092] The SDK may support an open tool driven development for third party partners and sound illusion (SI) teams. The SDK may empower developers to independently build and distribute their own personalized secure transaction widgets based on the guidelines issued by the wallet management / enabling layer provider (typically a network provider). The SDK may open up various layers (essentially top to bottom). The SDK may provide access to the platform's lateral capabilities with respect to supporting multiple handset OS, robust messaging and communications, security and authentication, and NFC based secure proximity transactions for payment as well as non-payment services. The SDK may simplify development of the wallet and the widget. In an example, the MTP may also have a provision for building native wallets directly to leverage underlying functionality that may not yet be exposed through the SDK. This

may give a developer the best of both the SDK and the MTP with enough flexibility to build an improved user experience.

[0093] FIG.3 illustrates a detailed architecture of an MTP server 300. The MTP server may be an enabling infrastructure for building a wallet ecosystem that may support mCommerce (for example by OTA) and mPayment (for example by NFC Proximity) transactional services. The server may provide a front end gateway thereby extending the services offered by the ecosystem partners or service providers 324 (for example, the merchants, the banks, and the carriers) to a plurality of mobile handsets 328 through the mobile wallet application. In an exemplary and non-limiting embodiment, the server may include a set of components that may collaborate to provide wallet management, transaction management, and ecosystem integration. The server may be developed in Java and may support various application servers and databases. In an example, the server may be a container for the services being offered by the MTP. In an example, the server may be hosted in a secure DMZ (demilitarized zone) in a clustered setup for mission critical applications with connectivity to the ecosystem cloud.

[0094] As illustrated in FIG. 3, the MTP architecture may include server infrastructure components, service templates and reference implementations, NFC infrastructure components, and host integration framework.

[0095] In an example, the MTP server may include a plurality of channels 302, a communication module 304, an OTA component 308, a wallet management module 310, a transaction engine 312, a notification and service bus 314, an instruction management module 318, at least one value added services (VAS) module 320, NFC infrastructure 322, and a host integration framework. In an example, the channels 302 may provide a channel specific adaptation for requests to be received through multiple channels such as Mobile Data, SMS, USSD, and the like. In an example, the communication module 304 may provide the framework for supporting secure mobile client server interactions. The communication module 304 may facilitate the communication, authentication, authorization, and the like from the individual business components. The flexible architecture may facilitate a service developer to extend the security or introduce new requirements by adding additional layers. In an example, the OTA component 308 may provide a support that may be required for facilitating OTA delivery of applications,

widgets, and personalized tokens that may be used during transactions. In an example, the OTA provisioning may include devices discovery and ensuring the appropriate version may be delivered to the device. In an example, the wallet management 310 module may provide the implementation for the wallet state management, lifecycle services, version management, wallet history, etc. The payment services, for example, provisioning of soft cards and personalizing applets, may also be included as a part of wallet management. In an example, the transaction engine 312 may provide the framework for developing OTA transactional services. The transactional service implementation may be developed by following the templates provided by the transaction framework. The requirements of reliability, security, logging, etc. may be transparently managed by the transaction framework. The transaction framework may provide a centralized access to interfaces with internal virtual accounts and external payment gateways. In an example, the notification and message bus 314 may provide an object exchange bus with the client application. In an example, the notification and message bus 314 may include support for pushing alerts and message to the wallet. In an example, an instrument management module 318, there may be several components which may provide various wallet instruments (for example, credit cards, bills, tickets, etc.) and manage various lifecycle services and configurations associated with these instruments within the platform. In an example, at least one of the VAS modules 320 may include a library of service templates and implementations for common business services that may have been built and deployed using the MTP platform. In an example, the NFC infrastructure 322 components a set of components that may integrate with the NFC ecosystem hosts for orchestrating the issuance and personalization of soft card and applets. The integration may support be required for enabling global platform compliant NFC transactional services from the ecosystem. In an example, the host integration framework may a layer including various connectors that may be plugged into the server to communicate with the back-end systems for facilitating the transactional services. The host integration framework may include a series of interface definitions which may have to be implemented by the service developer in the form of connectors to communicate with external host systems.

[0096] In various exemplary and non-limiting embodiments of the present invention, the primary responsibilities of the MTP server may include providing wallet management for the community of wallet users. In an example, the wallet management may include wallet lifecycle services, payment, widget management, and the like. In various exemplary and non-limiting embodiments of the present invention, the primary responsibilities of the MTP server may include establishing a service ecosystem and orchestrate transactions from the wallet application with the ecosystem service providers in a secure and reliable manner. In various exemplary and non-limiting embodiments of the present invention, the primary responsibilities of the MTP server may include supporting a third party developer community in building and distributing transactional widgets and manage the widget discovery and delivery. In various exemplary and non-limiting embodiments of the present invention, the primary responsibilities of the MTP server may include providing defined patterns and templates for implementing mobile financial transactions that hide the complexity of security, reliability, etc. from the service developer.

[0097] FIG.4 illustrates a high level architectural view of the Wallet application with a MTP mClient runtime 400. For the purpose of this figure a wallet 404 and widget 408a and 408b may be similar as modules described above but differ in functionality and responsibilities. In various exemplary and non-limiting embodiments of the present invention the MTP mClient 402 may be an enabling infrastructure for rapid mobile application development. The mClient may provide a container based approach to supporting multi-tenant application architecture. In an example, the mClient container may be a phone specific runtime environment for the wallet 404 and the widgets 408a and 408b that may be developed using the MTP SDK. The mClient container may facilitate application execution, transactional security, reliability, device specific implementations, and the like thereby allowing the developer to focus more on a service's unique functions. The mClient container may be built on top of the device specific APIs and available for popular device environments for example, phone APIs 444. The mClient runtime may generate a native application screen from the widget descriptors. The screen may ensure that a button on the screen may appear like a native Android®, Blackberry®, or iPhone® button on the respective devices. The mClient

server may include at least one phone independent API 410. The phone independent API 410 may ensure that widgets 408a and 408b developed using the SDK run across all supported handsets. The phone independent APIs 410 may be specifically designed to provide all the support that a developer may need to create an improved user experience. The mClient server 402 may be configured so as to facilitate the native implementation within the underlying phone O/S as it may be abstracted away through the runtime and API. In an example, the wallet application may be like a widget as the wallet application may be interpreted and executed by the mClient runtime 402. In an example, the wallet 404 may have a privileged control over the mobile application. The wallet 404 may also define additional APIs for the widget developers that may be specific to the application's context (for example, related to financial sector, retail sector, healthcare sector, government).

[0098] In various exemplary and non-limiting embodiments of the present invention, the mClient runtime may have components that may wrap all the major phone platform developer APIs. In an example, the UI framework and the workflow engine may render the user interface and handle the event based execution of local business functions.

[0099] In various exemplary and non-limiting embodiments of the present invention, the mClient runtime may include following components. The mClient runtime may include an UI framework 412. In an example, the UI framework 412 may provide the necessary infrastructure to render the user interface. In an example, the UI framework 412 may include a UI engine 414, a UI screen & layouts 418, a set of styles 420, and a plurality of UI elements 422. The UI framework 412 may be configured to manage the diversity of handsets and screen capabilities. In an example, the UI engine 414 may manage the generation and display of screens 418 based on the screen configuration provided in the application XML. The mClient runtime 402 may include an action framework 424. The action framework 424 may provide the necessary support to execute business specific logic (actions). The action framework 424 may be in the form of custom JavaScripts or pre-existing platform services. The action framework may include a script engine 428. The script engine 428 may control the execution of the action and interaction with the user interface. The mClient runtime 402 may include a widget management component 430. The widget management component 430 may

facilitate managing the widget lifecycle services 432 such as installation, loading, unloading, removal, update, etc. The widget management component may include an access controller 434 to manage access control to only permit widgets to access resources for which they have permissions in their widget descriptor. The mClient runtime 402 may include a communication component 438. The communication component 438 may be configured for providing the basic communication services between the client and server applications. The communication component 438 may provide communications support for HTTP (& HTTPS) and SMS channels. In an example, the communication component 438 may include packet construction, encoding/decoding and basic HTTP error handling. In an example, the communication component 438 may support attachments to the communication packet for non-essential business objects like alerts and advertisements. The mClient runtime 402 may include a database component 440. The database 440 may provide services to store, update, remove and find data from a local database. In an example, the database 440 may provide APIs which, may not available from the native phone platform. The mClient runtime 402 may include a security component 442. The security components 442 may provide crypto APIs for encryption, decryption, hash based message authentication code (HMAC), secure random hashing, and the like. The mClient runtime may include a NFC component. The NFC 448 may implement the global platform compliant NFC services and provides APIs for interact with the applets on the phone's secure element. The NFC 448 may contain APIs for NFC tags and peer-to-peer (P2P) mode. The mClient runtime may include a phone component. The phone may provide platform services to interact with the phone components like calling, SMS, LED, vibrate command, calendar function, contact management, and the like. The mClient runtime may include the widgets 408a and 408b. The widget may be a service bundle that may encapsulate the screens, content, resources, JavaScript, and styles for a collection of business services. The widgets may be built using the MTP SDK which may include a set of tools that operate on a pre-defined set of XML files

[00100] In various exemplary and non-limiting embodiments of the present invention, the mClient may be a container within which widgets may be installed and run. In various exemplary and non-limiting embodiments of the present invention, mClient may

simplify development of end to end visual workflows for the entire mobile application. In various exemplary and non-limiting embodiments of the present invention, the mClient may provide a set of well tested, standardized APIs and platform services for application developers to quickly use to build their applications across mobile handset platforms without a risk of device fragmentation issues. In various exemplary and non-limiting embodiments of the present invention, the mClient may generate an improved user interface that may utilize the native smart phone capabilities, but may still be responsive on limited capability devices. In various exemplary and non-limiting embodiments of the present invention, the mClient may provide defined patterns for implementing mobile financial transactions that may be configured so that to have complexity of security, reliability, and the like as a backend operation.

[00101] FIG.5 illustrates an overview of a widget development process with reference to a development cycle.

[00102] In exemplary and non-limiting embodiments, the MTP SDK may be configured for facilitating the overall widget development 500 and provide a greater degree of flexibility and independence to the widget developers. A first stage in widget development may be a design stage 502. In an example, a developer may create the mobile application using a Microsoft office Visio® or a similar tool & any XML and JavaScript editor. Further to development, a widget or widget file 504 may be packaged. In an example, the developer may use an eclipse plug-in to package the widget workspace and create a widget package. The widget package may be sent to an automated build server for widget generation 508. The widget files 504 may not be in a form of a widget binary 514. In an example, the automated build server may generate an application binary consisting of the mClient runtime as well as the widget package. Further to packaging, a widget may be tested using a test container 510. In an example, the developer may utilize any off-the-shelf device simulator to verify widget flows and script execution. Further to testing, the widget may be published to a MTP server 512. In an example, the wallet users may be able to discover widgets and download them from the MTP server 512.

[00103] In exemplary and non-limiting embodiments, a plurality of tools may be used for widget development, for example, the MTP-SDK, the MTP Microsoft Visio add-on, and the MTP eclipse plug-in.

[00104] In an example, the MTP SDKs for the server side and the client side development. The client side SDK (also referred to as mClient SDK) may be essential for developing any mClient Widget. The server SDK may provide a set of useful protocols and contracts that may facilitate the developers to build on MTP's proven enabling frameworks. In an example, independent web services for the widgets may also be developed, without using the MTP Server.

[00105] In exemplary and non-limiting embodiments, the MTP mClient SDK may consists of a set of APIs to access various native phone platform services like database, contactless, communication, and the like. The APIs may be provided as JavaScript functions and may be utilized by the widget developers. In exemplary and non-limiting embodiments the MTP server SDK may consists of useful libraries and contracts which may enable rapid integration with mClient widget's utilizing MTP's underlying implementation for communication, security, reliability, failover, and the like.

[00106] In an example, Development of elements of an MTP, such as with a SDK, may be supported by a visualization development capability (e.g. Microsoft Visio®) that facilitates developers defining user interface workflows by performing drag-n-drop actions with MTP templates, stencils, and the like. The MTP add-on may facilitate export of the workflow into an MTP, phone independent, XML format (for example, MTP application descriptors).

[00107] In exemplary and non-limiting embodiments, the MTP eclipse plug-in, the MTP SDK may be supported by an Eclipse plug-in that may be used to generate the wallet application or widgets from the application descriptors. The MTP eclipse plug-in may provide a single click build option for all the supported device platforms.

[00108] A mobile transaction platform may include reference implementations for elements such as containers, wallets, widgets, and for applications such as finance, retail, health, government, and the like. The features and functions of a mobile transaction platform may be delivered through multiple deployment models, which may include the use of a native client application (e.g. a container or mClient) that may be supported by

MTP tools that providers can use to develop client functions such as wallets, widgets, companion applets, couponing applets, and the like. Similarly, the MTP may facilitate deployment of web-services (that may be provided and/or customized for each provider) that may facilitate exposing different features of the MTP, and the like. An MTP, such as a multi-tiered secure mobile transactions enabling platform described herein, may include tools that can be used to develop wallets, widgets, companion applets, etc. These tools and methods for configuring and delivering wallets, widgets, and the like may also be configured with a wizard-like utility to expedite ecosystem development. Additional MTP tools can be configured to monitor usage and transactional information, and further offer up intelligent options and solutions for optimization based on various capability and performance criteria, etc.

[00109] A multi-tiered secure mobile transactions enabling platform may satisfy enterprise service delivery environment expectations and demands with the necessary support for scalability, redundancy, inter-operability, service delivery, message definitions, and the like. The platform may be delivered either through a hosted model for service / issue / acquisition providers and/or through a managed service model. The MTP facilitates implementing rules (e.g. business, transaction, trust, and the like) across an ecosystem as described herein, which can govern various ecosystem operations including on-boarding of participants into the ecosystem, how the participants access and consume the various services through the ecosystem, how a participant may be removed from the ecosystem, how different participants within the ecosystem may cooperate with each other, how different participants may cooperate with entities outside of the ecosystem, how participants acquire customers, how participants will support customers, how participants may capture data related to users and transactions (static and dynamic), how participants will use such data, share such data, and monetize such data – within the ecosystem as well as outside, among other things.

[00110] A MTP may be associated with an expert engine for providing customer support to consumers as well as providers/participants in the ecosystem. The expert engine may be leveraged to deliver a simplified user experience to consumers as well as providers/participants through the methods and systems described herein. An MTP may be architected with several tiers or facets as described herein and labeled for the sake of

description as enabling, service, personalization and the like, with each tier potentially further divided into other tiers, such as logical and physical tiers or aspects. Such a multi-tiered architecture or platform may uniquely help deliver the services and functionality in a multi-domain (and optionally multi-tenant) ecosystem as described herein while ensuring independent ownership among service providers and supporting evolution of differentiated capabilities, especially in a multi-domain ecosystem environment.

[00111] An MTP may be delivered through containers that may facilitate ecosystem-wide capabilities, wallets that may facilitate differentiated instrument implementation related capabilities and features, (e.g. NFC, NFC with Secure Element (SE), tokenization, tokenization exposed through QR Code/BarCode and the like), and widgets that may facilitate differentiated capabilities and features that providers can directly expose out to the end-user without disintermediation within the ecosystem.

[00112] A MTP, such as the multi-tiered secure mobile transactions enabling platform described herein, may ensure aggregation of different front-end protocols across an ecosystem, such as a multi-domain ecosystem described herein, based on back-end requirements of the ecosystem participants, rules and the like. Aggregation of different front-end protocols may be optimized further by providing an abstraction or wrapper capability, and/or a tokenization of the wrapper capability.

[00113] A MTP may enable an ecosystem to progressively deliver protocol level functionality into the front-end, through the methods and systems described herein over an enterprise service delivery environment. Protocol level functionality delivery may further be enhanced by providing tokenization of the wrapper or aggregation capability described herein, or by use of ecosystem rules.

[00114] A MTP may be set up in a network provider's domain and portions thereof integrated into their systems. The MTP may additionally be connected to an ecosystem of service and security providers, including one or more Trusted Service Managers (TSMs). A network provider may use the MTP to create a branded mobile application targeting a specific business vertical. Such a branded mobile application may be a wallet for, for example, a retail/financial domain or, for example, a wallet for the healthcare vertical. Such a wallet application may provide a core set of services to a user specific to the

domain for which it is created. In addition to providing core domain services, wallets may also handle application lifecycle and security, standardized user experience and widget management responsibilities. Widgets may provide specific business services and comply with the wallet guidelines.

[00115] FIG. 6 depicts an overview of an ecosystem for secure personalized transactions as described herein. The ecosystem 600 may include a plurality of participants such as a user of a client device 602, issuers of instruments 610, acquirers of transaction instruments 620, service providers or merchants 608, and a variety of an infrastructure elements including transaction and processing resources 604 that may include a mobile transaction platform (MTP) 622, rules, interfaces to tools and participants that maybe embodied as one or more enterprise service busses (ESB), point of sale facilities 612, secure point of sale switching functionality 618, functionality for abstracting transaction protocols 614, and the like. The client device 602 may include a plurality of containers that may include a plurality of wallets, wallet companion applets, and a plurality of widgets for performing a transaction (e.g., payment transaction) for the one or more services/products that may be offered by a service provider, such as service provider 608.

[00116] The MTP 622 depicted in FIG. 6 may be set up in a network provider's domain and integrated into the network provider's systems (optionally through their ESB). As shown, the MTP may be configured to be connected within the ecosystem 600 of service providers. Individual, independent service providers or merchants 608 may use the MTP software development kit and environment to build widgets that can be loaded into a wallet or a wallet container accessible on the client device 602. These widgets may provide incremental functionality that may enhance the overall appeal of the wallet application. Further, the MTP infrastructure 604 may include rules such as to provide wallet, widget, and transaction management along with all the relevant business services that are associated with delivering wallets to a universe of diverse mobile environments. Further, the MTP may establish connections with various ecosystems to deliver a dynamic user experience and diverse set of services to an end user that can be accessed anytime, anywhere.

[00117] The ecosystem 600 may further include a plurality of issuers of transaction instruments 610. The issuer 610 may provide various transaction instruments such as a

virtual credit card, a gift card, a loyalty card, tickets and the like. The user of the client 602 may select these instruments using one or more wallets and widgets to perform a transaction. In an example, the plurality of issuers 610 may use tools provided in association with the MTP 622 to develop customized widgets to enhance the user experience, provide security, comply with issuer workflows, and the like. Further, the ecosystem 600 may include digital wallets (e.g. that may be installed on a mobile client 602) that may include functionality for storing these instruments, performing a transaction for these instruments, and the like.

[00118] The ecosystem 600 may further be characterized by capabilities and features such as three-dimensional security for personalized transactions (e.g. device, domain, and user), multiple functionality tiers (e.g., an architecture with personalization, service, enabling tier, expert, and experience framework tiers or layers), support for multiple independent containers on a client device 602, multiple distinct widgets within a container, multiple distinct and independent wallets within a container, single widget access to multiple wallets, single wallet access by multiple widgets, support for a plurality of trust models, and the like. The ecosystem 600 may further include a point of sale facility or functionality 612 that may be configured to perform real (e.g. proximity-based) and/or virtual (e.g. on-line) transactions. The ecosystem 600 may include a protocol abstraction component 614 that may include a wrapper such as to perform transactions across the multiple types of protocols between the client 602 and the point of sale facility 612, and the like. For example, the protocol abstraction component 614 may enable a client device 602 perform a transaction independent of the protocol by which the point of sale device 612 or the aggregator switch 618 operates.

[00119] The ecosystem 600 may further include a plurality of acquirers 620. Each of the acquirers 620 may be use tools provided in association with the MTP 622 for developing customized widgets, wallets, and the like. In an example, the ecosystem 600 may include a switch 618 for selecting the acquirer 608 from the plurality of acquirers for performing a transaction.

[00120] As depicted in FIG. 6, issuers 610, service providers 608, acquirers 620, and the like may have access to or may configure their portions of the ecosystem for secure personalized transaction with only portions of the ecosystem 600 functionality. In an

example, acquirer 620A includes wallet functionality, whereas acquirer 620n includes widget functionality. Likewise, issuer 610a includes wallet and widget functionality, whereas issuer 610n only includes widgets. Similarly, service provider 608a includes widgets, wallets, and containers, whereas service provider 608n includes only widgets and wallets.

[00121] FIG. 7 depicts an exemplary client architecture 700 for use in an ecosystem for secure personalized transactions. The client architecture 700 may include a container based approach such as to support a multi-tenant application architecture. The client architecture 700 may be configured into business specific services, wallet functionality, enabling infrastructure, and the like that may be supported by device independent APIs, device-specific operating system environment(s), and the like. Wallet functionality may be container-based to provide functionality including business rules adherence, wallet and widget application lifecycle management, user experience management, and may further include one or more containers 714 that may include one or more wallets 702. Business specific services may be provided by one or more widgets such as a widget 704a, a widget 704b, a widget 704c….., and a widget 704n (collectively referred to herein as a widget 704). Enabling infrastructure may be facilitated by device agnostic APIs and a service execution environment in a client runtime environment 708.

[00122] Further, the client architecture 700 may include a mClient runtime container 708 that may a phone specific run-time environment for the wallet 702 and the widget 704 that are developed using the MTP Software Development Kit (SDK). This container may handle complexity related to application execution, transactional security, reliability, etc. thereby, this may allow the developer to focus more on an service's unique functions and not worry about most of the complexity, including device specific implementations. The mClient container 708 may be built on top of the device specific APIs and available for popular device operating system environments 710. The runtime container 708 may generate a native application screen from descriptors associated with widgets 704. This ensures that a button looks like a native ANDROID, BLACKBERRYOS, or IPHONEOS button on the respective devices.

[00123] The client architecture 700 may further include a device independent API layer 710 that may ensure that widgets developed using the SDK may run across all supported

handsets. These APIs are specifically designed to provide all the support that a developer needs to create rich user experiences. However, the developer may not need to worry about the native implementation within the underlying phone O/S since this is abstracted away through the client runtime container 708 and the API layer 712. The wallet 702 may be executed like a widget in that it may be interpreted and executed by the client runtime 708. However, the wallet 702 may have privileged control over the application. The wallet 702 may also define additional APIs for the widget developers that may be specific to the application's context (financial, retail, healthcare, government, etc.).

[00124] FIG. 8 depicts an ecosystem for secure personalized transactions including an MTP structural configuration with a protocol abstraction support capability. The system 800 may include a mobile device 802, a MTP server 804 and a plurality of service providers such as a service provider 808a, a service provider 808b, a service provider 808c……, and a service provider 808n communicatively coupled to a network 810. The mobile device 802 may include a plurality of containers such as a container 812a and a container 812n. Each of the containers may include a plurality of wallets such as a wallet 814a and a wallet 814n (collectively referred herein to as a wallet 814). Each of the containers may include a plurality of widgets such as a widget 818a and a widget 818n (collectively referred herein to as a widget 818).

[00125] Different service providers may leverage different features of a mobile transaction platform and/or wallet service center (WSC) as described herein. For example, the service provider 808a may access a wallet service center toolkit capability of the MTP, the service provider 808b may access a wallet service center plug-in capability, the service provider 808c may access a wallet service center web portal service, and the service provider 808n may provide a package of services that do not require any dedicated WSC or MTP resident software. The various service providers or merchants may access the WSC development environment and WSC run time through the network 810. For example, the service provider 808n may not be required to run the toolkit and may directly access the WSC toolkit as available on the MTP server 804. Similarly, the MTP server 804 may provide access to a library to the various service providers, wherein the library may include codecs for encoding/decoding,

authentication/security schemes and the like. Not every service provider is required to run a WSC library.

[00126] In an example, the MTP server 804 may be configured to provide a development environment as well as a runtime environment to the plurality of the service providers. At any instant, a service provider may either have the plug-ins, libraries or may access these components directly from the MTP server 804. As a result, this system allows the service providers or merchants to customize the configuration as per the desired requirements. Alternatively, the MTP server 804 may connect directly with a client device, such as the mobile device 802 to perform transactions, such as on behalf of a service provider (e.g. PL808n).

[00127] The ecosystem may include acquisition functionality, such as through various point-of-sale (PoS) terminals and over the air transactions. Protocol abstraction associated with transaction acquisition may optionally be facilitated by an acquisition module 824 that may incorporate MTP and/or WSC acquisition functionality 822 that may add more functionality to the system such as to make the transaction more reliable, robust and secure. A PoS terminal 820 may communicate with the mobile device 802 through one or more proximity communication channels including but not limited to NFC, Bluetooth, barcode reader and the like via any of a variety of acquisition protocols. Protocol abstraction may be handled via wrapping such communication between the mobile device 802 and the PoS terminal 820. In an example, the wrapper may be implemented for abstracting a protocol that embodies a transaction between a wallet 824 and the PoS terminal 820. For over the air transactions (e.g. that do not directly use a PoS terminal 820), protocol abstraction can be similarly generated via wrappers, tokens, tokenized wrappers, and the like that may be used to exchange information over a public network 810 (e.g. the Internet) to effect a transaction.

[00128] FIG. 9 depicts an alternate configuration of an ecosystem for secure personalized transactions as depicted in FIG. 8 without the use of an acquisition facility (e.g. 824 in FIG. 8).

[00129] The methods and systems described herein of multi-domain ecosystem secure personalized transactions include a system such as a platform for secure personalized transactions in a multi-domain ecosystem. The platform may include a personalization

tier for enabling service provider personalization of one or more ecosystem elements stored on a mobile device. The one or more ecosystem elements may be selected from the group consisting of widgets, wallets, and applets. The platform may further include an enabling tier for facilitating interoperation between the personalization tier and a client device. The platform may further include a service tier operating independently of the enabling tier to enable service delivery for a plurality of services. In an example, the plurality of services may be selected from a group consisting of banking, ticketing and bill payment. The platform may be operable in a distributed network of ecosystem participants. The ecosystem participants may include such as service providers, instrument issuers, instrument acquirers, and mobile device users. The platform may further include a plurality of functional modules. The functional modules may include one or more module for such as communication, notification, security, authentication, over the air operation, and device runtime container to manage wallets and widgets. The platform may further include a software development environment that may include such as reference implementations of containers, wallets, and widgets. The software development environment may further include such as reference implementations of finance, retail, health, and government domains. In an example, the platform may be deployed through the use of a native client application. In an example, the platform may be deployed through the use of a native client application and a set of software development tools. The set of software development tools may be used to develop client functions. The set of client functions may be selected from the group of wallets, widgets, companion applets, and couponing applets. In an example, the platform may be deployed through the use of a native client application and a set of web services that may be exposed by the platform to integrate peripheral systems. The peripheral systems may include such as customer care and user portals. In an example, the platform may be deployed with tools that may be used to such as develop wallets, widgets, and companion applications. The tools may include a wizard-like utility. In an example, the tools may be used to monitor usage and transaction information. In an example, the tools may be used to provide intelligent optimization options and solutions. The optimizations solutions may be based on monitored capability and performance criteria. In an example, the platform may be deployed with the necessary support in an enterprise service delivery

environment. The necessary support may include such as support for scalability, redundancy, inter-operability, service delivery, and message definitions. In an example, the platform may be deployed through a hosted model for a plurality of providers. The plurality of providers may be selected from the group including such as service, issue, and acquisition providers. In an example, the platform may be delivered by implementing rules which may govern the use of services by the ecosystem participants. In an example, the rules may govern the access of the services by the participants. In an example, the rules may govern the consumption of the services by the participants. In an example, the rules may govern how a participant may be removed from the ecosystem. In an example, the rules may govern how participants may cooperate with each other within the ecosystem. In an example, the rules may govern how participants may cooperate with entities outside of the ecosystem. In an example, the rules may govern how participants may acquire customers. In an example, the rules may govern how participants may support customers. In an example, the rules may govern how participants may capture data related to users and transactions. In an example, the users and transactions may be static. In an example, the users and transactions may be dynamic. In an example, the rules may govern how participants may use the captured data within the ecosystem. In an example, the rules may govern how participants may use the captured data outside of the ecosystem. In an example, the rules may govern how participants may share the captured data within the ecosystem. In an example, the rules may govern how participants may share the captured data outside of the ecosystem. In an example, the rules may govern how participants may monetize the captured data within the ecosystem. In an example, the rules may govern how participants may monetize the captured data outside of the ecosystem. In an example, the platform may be delivered by leveraging an expert engine to provide customer support to participants such that the participants may be selected from a group of consumers and providers. In an example, the platform may be delivered by leveraging an expert engine to provide a simplified user experience to participants such that the participants may be selected from a group of consumers and providers. In an example, the platform may be architected using multiple tiers. The multiple tiers may be selected from the group including such as an enabling tier, a service tier, and a personalization tier. Each tier may further be divided into

additional tiers. The additional tiers may be selected from the group including such as logical tiers and physical tiers

[00130] A multi-domain ecosystem platform for secure personalized transactions may comprise a personalization tier for service provider personalization of ecosystem elements, an enabling tier for facilitating interoperation with client device software and hardware that operates independently of a service tier that facilitates service delivery for a variety of services such as banking, ticketing, bill payment, and the like. Such a multi-domain ecosystem platform may facilitate generic and specialized transactions, including secure personalized transactions, may be logically and/or physically segmented into multiple functionally independent blocks, segments, units, layers or tiers such that they operate separately, in conjunction through interfaces using exchange of data or information, or seamlessly with each other in a predefined or configurable order to fulfill a larger function or unit of transaction or a complete transaction. As an example, the multi-tiered ecosystem-enabling platform as described herein may be comprised of three tiers, namely, the personalization tier, service tier and the enabling tier.

[00131] Further, as an example, the personalization tier may be used to effect specific branding elements (such as trademarks, logos, themes, colors or the like), user interaction elements, workflows (e.g., stored account and transaction data to enable efficient completion of repeated transactions, shopping carts, single-click transactions and many others), user interface effects (e.g., transitions, content formats, dynamics or any other action), cause and / or effect that may be required by a service provider for a specific requirement. A personalization layer may facilitate the deployment of a service provider's corporate design or interaction philosophy as well as giving room to consumers to adjust to their personal needs and preferences, including for the visually impaired, and the like.

[00132] The enabling tier may facilitate interaction with the underlying layers, platform or substrate which may be in the form of software or hardware or a combination of both to communicate and interact therewith. The enabling tier may or may not serve as an abstraction effectively isolating the layers or tiers above it from those that are below the enabling layer.

[00133] The service tier may cater to the specific requirements such as work flows, business logic, rules, intelligence, and needs of a service vertical which includes service vertical such as, but not limited to, banking, ticketing, bill payment, couponing, loyalty and point card programs, parking, and other ecosystem services.

[00134] Services enabled by such a service tier may include mobile banking, money transfer, bill payments, ticketing, couponing, loyalty programs, marketing and advertising related services, education related services, city services, mobile health services, mobile insurance, transit services such as parking coupons, travel services, emergency or 911 services, retail payments, logistics support services, business intelligence solutions, branding, shopping, product authentication services, regulatory services, records management, interactive television services, text-to-voice services, location based services and the like

[00135] While the examples used herein illustrate the concept of multi-tiered ecosystem enabling platforms, the methods and systems described herein actually cover more generalized platform architectural and/or structural concepts including multiple functionally independent blocks, segments, units, layers or tiers. FIG. 10 depicts a notion of a tiered platform that may span a range of ecosystem elements including client devices (e.g. mobile devices), wallet service center functionality, trusted security managers, mobile transaction platform servers, issuers of wallets, merchants (aka service providers), and the like. Each such ecosystem element may comprise at least enabling, service, and personalization tiers. As noted herein, additional tiers, such as expert engine and experience framework tiers may also be incorporated in deployments of a multi-tiered secure mobile transactions ecosystem-enabling platform.

[00136] The methods and systems described herein of multi-domain ecosystem secure personalized transactions includes a method that includes analyzing one or more of a set consisting of ecosystem participants, service requirements, customer requirements, regulatory requirements, transaction security requirements, trust models, and electronic transaction communication channels. The method may further include determining, based in the analysis, one or more ecosystem features to configure in a platform comprising at least one of a service tier, an enabling tier, and a personalization tier. The method may further include configuring the one or more ecosystem features in

accordance with the determination. In an embodiment, the one or more ecosystem features may be configured to ensure satisfaction of each analyzed requirement within a context of a service provider business model. The platform may additionally include an expert engine tier and an experience framework tier.

[00137] Methods for selecting eco system features for inclusion in each of an enabling tier, a service tier, and a personalization tier may include analyzing ecosystem participants, service requirements, customer requirements, regulatory requirements, transaction security requirements, trust models, and electronic transaction communication channels to determine ecosystem features to configure in a service tier, an enabling tier, and a personalization tier to ensure satisfaction of each analyzed requirement within the context of a service provider business model. Features may be similarly determined for additional tiers, such as expert engine and experience framework tiers.

[00138] A method or process of breaking down parts of the overall transaction, service or process into the multi-tiered ecosystem platform may be based on a combination of analytics, heuristics and requirements, optionally aided by tools including expert engines based on artificial intelligence concepts. Such a method or process may take into account the perspectives and requirements of all participating – present and anticipated in future – entities in the ecosystems including, but not limited to, service providers; service requirements; customer requirements; end-user requirements; regulatory mandates & guidelines; security mandates, guidelines and requirements; trust models; economic models; business logic and workflow; commonality with other services; operational factors; design & technology factors; target devices & platforms; projected technology evolution; other related & unrelated services; and operating constraints.

[00139] In most cases, the aspects and parameters mentioned above can result in the service requirement to be logically segmented or partitioned into three tiers, namely the personalization, service and enabling tiers.

[00140] Referring to FIG. 11, an embodiment of selecting ecosystem features for inclusion in each of an enabling tier, a service tier, and a personalization tier of a multi-domain ecosystem platform for secure personalized transactions. Based on participant business goals and requirements, an ecosystem feature allocation engine 1102 may

process rules and controls such as to allocate features of the ecosystem to the enabling tier 1104, a service tier 1108, and a personalization tier 1110.

[00141] In an example, the enabling tier 1104 may include infrastructure or business independent components that may be necessary to develop mobile applications. The service tier 1108 may include a collection of reference service implementations for wallet ecosystems and value added services. The personalization tier 1110 may include a set of tools and APIs that may be necessary to develop the wallet and the widget services (described in this document) and build a comprehensive ecosystem.

[00142] In an example, the enabling tier 1104 may include features different for the client and the server. In the enabling tier 1104, the client may be configured to include infrastructure runtime container software (SW) that may be built using a specific development environment associated with the mobile device. In an example, the enabling tier 1104 may provide the client a common abstraction to underlying device capabilities with common Developer APIs, UI Interpretation and rendering engine, business execution environment, widget and wallet lifecycle management, patterns for OTA/proximity transactions, and the like. On the server side, the enabling tier 1104 may provide to the server an infrastructure software that may be deployed in network provider's domain that may be configured for establishing and managing a secure mobile transactional ecosystem. In an example, the enabling tier 1104 may configure the server features such as wallet management, stateful transaction engine, object lifecycle management, channel specific communication, multi-domain security policy and risk management, OTA provisioning, and the like.

[00143] In an example, the service tier 1108 may include multiple features for the client. For example, the client may be configured to include a library of common platform services that may be used to implement a plurality of business use cases. The common platform services may run within the business execution environment provided by the enabling tier 1104. In an example, the service tier 1108 may include three basic units. A first unit may be a use case, for example, a set of client side workflows and templates built for the various transaction services. A second unit may be an instrument, for example, a corresponding single wallet instrument that may facilitate mapping to real world instruments. In an example, the instrument may provide data for a visual or a

display. In an example, the instrument may provide data for an over-the-air (OTA) transaction. In an example, the instrument may provide data for a proximity transaction. In an example, the instrument may provide data for an administration. A third unit may include at least one trust model. The trust model may facilitate implementation for authentication schemes, which may include a two factor authentication (2FA). The 2FA may be required by different widget providers and may be relevant for the ecosystem.

[00144] In an example, the service tier 1108 may include multiple features for the server. The server may be configured to include a library of business specific components and associated service templates that may be partially or entirely reused to develop a customer centric offering. In an example, the services may include wallet services and value added services. The wallet services may include managing the lifecycle of the wallet and payment tokens. The value added services may include providing business services. The service tier 1108 may include three units. A first unit may be a use case, for example, the server side template may be built using the mobile enabling tier 1104's transactional framework for implementing a specific use case. In an example, there may be one or more varying templates for a business service (e.g., bill payment, movie ticketing, and the like). A second unit may include at least one instrument or token. The instruments may include business specific data objects (for example, credit card, bank account, bill, coupons, etc.) that may be reused across various services and may provide a commonly understood definition for the object and implementation for the rules governing the object's lifecycle and transaction. A third component may include at least one trust model. The trust model may facilitate implementation for different authentication schemes for the services, including support to operate within individual trust clusters with their independent TSMs.

[00145] In exemplary and non-limiting embodiments, the personalization tier 1110 (may also be referred to as a SDK tier) may include different features for the client and for the server. In an example, features associated with the client in the SDK tier may be divided into 2 parts. A first part may be a wallet SDK and a second part may be a widget SDK. In an example, the wallet SDK may provide access to the wallet services, privileged APIs, and the necessary components to build the wallet container application or user experience. In an example, the widget SDK may facilitate the ecosystem partners to

build their widgets that may operate within the wallet container. In exemplary and non-limiting embodiments, the SDK may essentially facilitate client side configuration and user experience. In an example, the SDK may facilitate configuration by facilitating customizability for parameters exposed by the client container. This may include device specific parameters, variables (like PIN size, exit confirmation, selecting inclusion of modules, etc.). In an example, the SDK may provide user experience by facilitating the development of differentiating user experiences across handsets.

[00146] In an example, the SDK for the server may include at least one library that may allow any ecosystem server to communicate with the wallet, format content for the widgets, and expose services to the wallet application. The library may include a configuration module and an interface module. The configuration module may allow customizability for parameters exposed by the server's mobile enabling tier 1104 and service tier 1108. The parameters may be configured in property files and resource bundles. The interface module may provide connectors developed to facilitate communication with various backend systems that may collaborate to fulfill the complete end to end service. The interface module may also include web services exposed by the platform to integrate peripheral systems like customer care, user portals, and the like.

[00147] The methods and systems described herein of multi-domain ecosystem secure personalized transactions includes a method in a platform including such as at least one of a service tier, an enabling tier, and a personalization tier. The method includes applying an expert engine to analyze one or more mobile device user and transaction environment requirements. In an example, the expert engine may include an expert engine tier. The expert engine tier may interact with an experience framework to adapt the one or more user requirements. In an example, applying the expert engine may include applying the expert engine to analyze a user's habits. The method may further include configuring one or more platform features in accordance with the analysis. In an embodiment, configuring the one or more ecosystem features may include configuring a user interface based on one or more transaction requirements. In an example, configuring the one or more ecosystem features may include configuring one or more display features of a mobile device. In an example, the one or more display features may be configured based on a regional preference associated with a location of the mobile

device. In an example, the one or more display features may be configured based on an identified client base.

[00148] A multi-tiered secure mobile transaction enabling platform may facilitate scaling a multi-domain ecosystem for secure personalized transactions to tens of millions of users, comprising applying an expert engine for analyzing user and transaction environment requirements and configuring ecosystem features to simplify transactions, such as configuring a user interface based on transaction requirements. The expert engine may interact with an experience framework to facilitate adapting to user requirements.

[00149] It is anticipated that users will have varying preferences, habits, perceptions, perspectives, requirements and styles when it comes to using devices and applications. In addition to the parameters mentioned above, users have individual preferences, tastes, and requirements for the presentation of results of their operations. Many or most applications currently available constrain users to follow a specific regimen to carry out transactions and obtain and view results from the system.

[00150] The expert engine tier described in this document may learn user habits, preferences and other parameters mentioned above, and may adjust the system and/or the application to present itself in the manner most suitable to individual users including regional preferences ("most French like it this way while the Americans like it that way"), mobile device capabilities (screen size, keyboard available or only on-screen, and the like), and differences in client base (low-end versus luxury customers).

[00151] An expert engine tier may also manage the vagaries in the ecosystem such as breaks in communication, delays in the server's response, abnormal operating conditions such as a rundown battery or signal drops, etc. to ensure an uninterrupted or interruption managed experience for the user. The expert engine may ensure that operation breakdowns are managed and the user is able to seamlessly resume operations as and when normalcy has been restored.

[00152] The issues mentioned above are further compounded when we add to them the dimension of having a large number of disparate users. The numbers are estimated to run into tens of millions with thousands transacting simultaneously.

[00153] An expert engine may essentially manage all users as outlined earlier, and also carry out analytics on the transactions, usage patterns, and other parameters that will help the 'learning' process of an inference engine.

[00154] Referring to FIG. 12, an expert engine as described herein is exemplary depicted for facilitating modifying each of a personalization and a service tier for each of a plurality of service providers in a multi-domain ecosystem for secure personalized transactions.

[00155] An expert engine tier/facility 1208 may take provider specific personalization tiers 1202 and adapt them to provide adapted personalization tiers 1210 by analyzing a variety of aspects 1214 of the ecosystem, including transaction environment aspects, user history aspects, user environment aspects, historical transaction related data aspects, device related aspects, ecosystem status (e.g. responsiveness and/or stability and the like), eco system size as may be determined from a number of enrolled participants and/or ecosystem transaction activity, provider specific aspects, and the like. Similarly, provider specific service tier data 1204 may be expertly adapted by the expert engine tier / facility 1208 to produce one or more expertly adapted service tiers 1212.

[00156] The methods and systems described herein of multi-domain ecosystem secure personalized transactions includes a platform for enabling multiple types of electronic wallets in an ecosystem. The platform may include a software development kit for developing a provider specific wallet. The software development kit may be executable at least in part by a networked server. The platform may further include a plurality of reference wallets accessible through the software development kit. The plurality of reference wallets may include such as a reference wallet for each of a plurality of providers including an instrument issuer, a service provider, and an instrument acquirer. The platform may further include a runtime container operable on a client device for receiving different types of provider specific wallets. The container may facilitate distinct operation of each type of provider specific wallet for performing secure transactions between a provider specific wallet and a provider that developed the provider specific wallet.

[00157] The methods and systems described herein of multi-domain ecosystem secure personalized transactions include a computer readable medium containing program

instructions. The execution of the program instructions by one or more processors of a computer system may cause the one or more processors to carry out the steps that include configuring one or more widget modules in a wallet container on a mobile device. The steps may further include accessing one or more wallet modules in the wallet container through the one or more widget modules. In an example, accessing the one or more wallet modules may include receiving a communication from a service provider module to a first widget module in the wallet container and enabling authenticated access by the first widget module to the wallet module. In an example, each of the one or more wallet modules may be configured and installed in the wallet container over a network by a service provider. In an example, each of the one or more wallet modules may be configured and installed in the wallet container over a network by a user. The steps carried out by computer readable medium may further include authenticating each of the one or more widget modules prior to accessing the one or more wallet modules.

[00158] A secure ecosystem infrastructure that may enable multiple types of electronic wallets in a multi-domain ecosystem of issuers, service providers, and acquirers of instruments may include client-side functionality including a wallet container operable on a mobile device that facilitates isolation of a plurality of distinct wallets and further facilitates accessing each distinct wallet through widget modules configured into the wallet container and securely maintained as separate modules, wherein each widget must be authenticated to access any wallet. Such an ecosystem infrastructure may also include personalized service provider modules operating remotely from the mobile device and distinctly from each other wherein a service provider module communicates to a widget in the wallet container and the wallet container facilitates authenticated access by the widget to a wallet. In addition, ecosystem infrastructure elements may include capabilities for configuring and installing each of the distinct wallets in the wallet container over a network by a service provider, issuer, proxy, and the like. These and other features of a secure ecosystem infrastructure for enabling multiple types of electronic wallets is further depicted and described in Figs. 8 and 9.

[00159] The methods and systems described herein of multi-domain ecosystem secure personalized transactions include a method that includes analyzing at least one requirement. In an example, the at least one requirement may be selected from the set

consisting of business strategies, go-to-market strategies, exit strategies, competitive strategies, product roadmaps, hardware release schedules and feature lists, software release schedules, feature lists, and application programming interfaces. In an example, the at least one requirement may be analyzed within the context of a service provider business model. The method may further include configuring a service tier, an enabling tier and a personalization tier based, at least in part, upon the analysis. In an example, the enabling tier may be configured to include infrastructure runtime container software built using a specific development environment associated with a mobile device. In an example, the service tier may be configured to include a library of common platform services that may be used to implement a plurality of business use cases.

[00160] Methods for selecting new business needs and device functionality for inclusion in each of an enabling tier, a service tier, and a personalization tier may include analyzing requirements related to business strategies, go-to-market strategies, exit strategies, competitive strategies, product roadmaps, hardware release schedules and feature lists, and software release schedules, feature lists, and application programming interfaces to configure in a service tier, an enabling tier, and a personalization tier to ensure satisfaction of each analyzed requirement within the context of a service provider business model. Features may be similarly determined for additional tiers, such as expert engine and experience framework tiers.

[00161] A method or process of breaking down parts of the overall transaction, service or process into the multi-tiered ecosystem platform may be based on a combination of analytics, heuristics and requirements, optionally aided by tools including expert engines based on artificial intelligence concepts. Such a method or process may take into account the perspectives and requirements, both present and anticipated in future, related to – business strategies, go-to-market strategies, exit strategies, competitive strategies, product roadmaps, hardware release schedules and feature lists, and software release schedules, feature lists, and application programming interfaces.

[00162] In most cases, the aspects and parameters mentioned above can result in the service requirement to be logically segmented or partitioned into three tiers, namely the personalization, service and enabling tiers.

[00163] Referring to FIG. 13, an embodiment of selecting business needs and device functionality for inclusion in each of an enabling tier, a service tier, and a personalization tier of a multi-domain ecosystem platform for secure personalized transactions. Based on participant business needs and device functionality, an ecosystem feature allocation engine 1302 may process rules and controls such as to allocate features of the ecosystem to the enabling tier 1304, a service tier 1308, and a personalization tier 1310.

[00164] In an example, the enabling tier 1304 may include infrastructure or business independent components that may be necessary to develop mobile applications. The service tier 1308 may include a collection of reference service implementations for wallet ecosystems and value added services. The personalization tier 1310 may include a set of tools and APIs that may be necessary to develop the wallet and the widget services (described in this document) and build a comprehensive ecosystem.

[00165] In an example, the enabling tier 1304 may include features different for the client and the server. In the enabling tier 1304, the client may be configured to include infrastructure runtime container software (SW) that may be built using a specific development environment associated with the mobile device. In an example, the enabling tier 1304 may provide the client a common abstraction to underlying device capabilities with common Developer APIs, UI Interpretation and rendering engine, business execution environment, widget and wallet lifecycle management, patterns for OTA/proximity transactions, and the like. On the server side, the enabling tier 1304 may provide to a server infrastructure software that may be deployed in a network provider's domain that may be configured for establishing and managing a secure mobile transactional ecosystem. In an example, the enabling tier 1304 may configure the server features such as wallet management, stateful transaction engine, object lifecycle management, channel specific communication, multi-domain security policy and risk management, OTA provisioning, and the like.

[00166] In an example, the service tier 1308 may include multiple features for the client. For example, the client may be configured to include a library of common platform services that may be used to implement a plurality of business use cases. The common platform services may run within the business execution environment provided by the enabling tier 1304. In an example, the service tier 1308 may include three basic units. A

first unit may be a use case, for example, a set of client side workflows and templates built for the various transaction services. A second unit may be an instrument, for example, a corresponding single wallet instrument that may facilitate mapping to real world instruments. In an example, the instrument may provide data for a visual or a display. In an example, the instrument may provide data for an over-the-air (OTA) transaction. In an example, the instrument may provide data for a proximity transaction. In an example, the instrument may provide data for an administration. A third unit may include at least one trust model. The trust model may facilitate implementation for authentication schemes, which may include two factor authentication (2FA). The 2FA may be required by different widget providers and may be relevant for the escosystem.

[00167] In an example, the service tier 1308 may include multiple features for the server. The server may be configured to include a library of business specific components and associated service templates that may be partially or entirely reused to develop a customer centric offering. In an example, the services may include wallet services and value added services. The wallet services may include managing the lifecycle of the wallet and payment tokens. The value added services may include providing business services. The service tier 1308 may include three units. A first unit may be a use case, for example, the server side template may be built using the mobile enabling tier 1304's transactional framework for implementing a specific use case. In an example, there may be one or more varying templates for a business service (e.g., bill payment, movie ticketing, and the like). A second unit may include at least one instrument or token. The instruments may include business specific data objects (for example, credit card, bank account, bill, coupons, etc.) that may be reused across various services and may provide a commonly understood definition for the object and implementation for the rules governing the object's lifecycle and transaction. A third component may include at least one trust model. The trust model may facilitate implementation for different authentication schemes for the services, including support to operate within individual trust clusters with their independent TSMs.

[00168] In exemplary and non-limiting embodiments, the personalization tier 1310 (may also be referred to as a SDK tier) may include different features for the client and for the server. In an example, features associated with the client in the SDK tier may be divided

into 2 parts. A first part may be a wallet SDK and a second part may be a widget SDK. In an example, the wallet SDK may provide access to the wallet services, privileged APIs, and the necessary components to build the wallet container application or user experience. In an example, the widget SDK may facilitate the ecosystem partners to build their widgets that may operate within the wallet container. In exemplary and non-limiting embodiments, the SDK may essentially facilitate client side configuration and user experience. In an example, the SDK may facilitate configuration by facilitating customizability for parameters exposed by the client container. This may include device specific parameters, variables (like PIN size, exit confirmation, selecting inclusion of modules, etc.). In an example, the SDK may provide user experience by facilitating the development of differentiating user experiences across handsets.

[00169] In an example, the SDK for the server may include at least one library that may allow any ecosystem server to communicate with the wallet, format content for the widgets, and expose services to the wallet application. The library may include a configuration module and an interface module. The configuration module may allow customizability for parameters exposed by the server's mobile enabling tier 1304 and service tier 1308. The parameters may be configured in property files and resource bundles. The interface module may provide connectors developed to facilitate communication with various backend systems that may collaborate to fulfill the complete end to end service. The interface module may also include web services exposed by the platform to integrate peripheral systems like customer care, user portals, and the like.

[00170] In an example, a device manufacturer may determine that NFC functionality will be offered as a new communication option on a new device, in order to support the business objectives of the company. Upon detection of the availability of the new NFC functionality, the business needs and device functionality allocation engine 1302 may review the requirements of the device manufacturer and allocate the device functionality to the enabling tier 1304. The enabling tier 1304 may then make the NFC device functionality available to the service providers who offer that device to its customers. The service provider may then enable NFC functionality on a device through the service tier 1308 and enable an end user to activate NFC functionality as a payment method by

adding an NFC icon to a mobile widget using the SDK provided by the personalization tier 1310.

[00171] In accordance with exemplary and non-limiting embodiments, the present disclosure provides a mobile transaction platform (MTP) that may build a secure and a robust ecosystem for mPayment and mCommerce. In an example, the MTP may incorporate a widget architecture that may allow service providers to build their own independent widgets that can be downloaded and installed within a mobile wallet application. In an example, the MTP may offer a comprehensive Software Development Kit (SDK) to the service providers. The SDK may allow the service providers to quickly build their widgets for all MTP handsets without having to write phone-specific code. While the mobile wallet application may be distributed and promoted by an ecosystem operator, the service providers may build and offer their own unique services through the widgets and leverage the distribution channels created by the ecosystem operator. The mobile wallet application may offer the widgets a secure and isolated runtime environment.

[00172] In an example, the MTP may manage the one or more widgets using a widget management component that may manage use cases, design, architecture, data structures, lifecycle, versions and the like of the widgets within the MTP.

[00173] The MTP may include an mClient such as to enable infrastructure for rapid mobile application development. The mClient may provide a container-based approach such as to support multi-tenant application architecture. In an example, an mClient runtime may be a container for running the mobile wallet applications. The wallet application may be an overarching and a primary application that may provide an overall user experience and server as an entry point to widgets. In an example, the widgets may include service groups developed by third parties and run within the mClient container, independent and isolated from the wallet application. The mClient may handle complexity related to application execution, security, reliability, and the like and thereby, may allow the developers to focus more on service's unique functions. The mClient may enable the developer not to worry about most of the complexity, including device specific implementations. In an example, the mClient container may be built on top of the device specific application programming interfaces (APIs) and available for popular device

environments. The device independent API may ensure that applications that may be developed using the SDK may run across all supported MTP handsets. These APIs may be designed to provide all the support that developers may need to create dynamic user experiences. In an example, wallet application and widgets may be built using the MTP SDK. In an example, the wallet application and the widgets may be identical in structure (screens, JavaScript, & CSS Styles), except that the wallet application may access a relatively wider set of privileged APIs. When the wallet application may be first invoked, the wallet application may take control until the user may select an entry point into a particular widget (a card image, menu option, etc.). The wallet developer, as a part of the overall UI design, may determine the one or more entry points for the widget. On selection of the entry point of the widget, the widget may be launched (within the same runtime application) and the wallet may be suspended. In an example, the wallet may stop executing. Once the widget execution is completed, the wallet may be resumed and the screen from where the user left the wallet may be returned to the display.

[00174] In accordance with the exemplary and non-limiting embodiments, the widget developer may develop the widgets and test depending on the requirement that may be provided by the service provider. The service provider may submit the developed widget to an MTP server. Further, the ecosystem provider may certify and publish the widget so that the widget may be accessible to the wallet users. In an example, the wallet user may discover and download the widget to access the mobile services provided by the service provider. In an example, the widget may already be installed in the mobile phone of the wallet user. The wallet user may discover the higher version of the widget and accordingly, may update the existing version of the widget with the updated version. In an example, the wallet user may not be interested in using the installed widget and accordingly, the wallet user may terminate and remove (i.e., uninstall) the widget from the mobile device.

[00175] In an example, the widget developer may be trained to use the SDK and understand the services to build the widgets. Additionally, the widget developer may have guidelines that may be specified by the ecosystem provider such as to ensure that the widget may comply with the guidelines.

[00176] In an example, the service provider may be an entity that may issue or own a service product. The service may include a Credit Card for mPayment, a Movie Ticketing service, an mBanking service, and the like. In an example, the service provider may commission the widget development and enter into a business agreement with the ecosystem provider.

[00177] In an example, the ecosystem operator may be an entity that may set up an mPayment and mCommerce ecosystem within the region/geography or business domain. The ecosystem may not necessarily be a payment or commerce ecosystem. In an example, the ecosystem can be a healthcare or other complex business domain.

[00178] In an example, the wallet user may be a consumer that may have downloaded and activated the mobile application (or wallet if related to a payment domain) on their phone.

[00179] In accordance with the exemplary and non-limiting embodiments, the present disclosure provides a plurality of use cases for the widget. In an example, one or more use cases for the widget may include a Develop Widget use case, wherein the developer may build the widgets as per the specifications provided by the service provider. In an example, one or more use cases for the widget may include a Test Widget use case, wherein the developer may test the widgets using both a dummy container and possible a sandbox wallet application. In an example, one or more use cases for the widget may include an Approve Widget use case, wherein the business sponsor within the service provider may review and approve the widget. Such use cases may not be on the MTP server or client.

[00180] In an example, one or more use cases for the widget may include a Submit Widget use case, wherein the service provider may send the widget to the ecosystem operator along with a set of test cases. The widget may be installed on the ecosystem provider's test/UAT environment. In an example, one or more use cases for the widget may include a Validate Widget use case, wherein the ecosystem operator may review the widget test cases, test the widget, validate all functionality of the widget, and check the widget against all defined guidelines. In an example, one or more use cases for the widget may include an Approve Widget use case, wherein the ecosystem operator may approve the widget and permit the widget to be included within the wallet application.

[00181] In an example, one or more use cases for the widget may include a Reject Widget use case, wherein the ecosystem operator may reject or deny the widget for some reason. The entire process so far may be repeated once the issues are fixed and a new widget may be submitted. In an example, one or more use cases for the widget may include a Publish Widget use case, wherein the ecosystem operator may publish the widget and load the widget to the MTP Server so that the wallet users may download the published widget. In an example, one or more use cases for the widget may include a Suspend Widget use case, wherein the ecosystem operator may suspend the widget that has been published due to some problem until this problem may be rectified. In an example, one or more use cases for the widget may include a Discover Widget use case, wherein the wallet users may be alerted to the availability of the widget through an alert or through a proactive search for available widgets. In an example, one or more use cases for the widget may include a Download Widget use case, wherein the widget may be provisioned to the user's MTP mobile client application and may be installed and ready for use. In an example, one or more use cases for the widget may include an Upgrade Widget use case, wherein the wallet users may upgrade their widget to a newer version from within the mobile wallet application. In an example, one or more use cases for the widget may include a Remove Widget use case, wherein the user may decide that the widget may be no longer desired. The wallet user may remove the widget and all associated data stores from the wallet application.

[00182] In an example, the widget may include a plurality of services that may provide the wallet user with meaningful, domain specific functionality that may augment the capabilities of the mobile application that may be used by the wallet user. The widget may be referred to as a widget bundle or a service capsule. In an example, the widget bundle may include a widget profile data, a widget user interface, widget business logic, and widget permissions.

[00183] In an example, the widget profile data may include information regarding the widget, service provider, version, and the like. In an example, the widget user interface may include screens, content, and styles that may be used by the wallet container to render the widget's user interface. In an example, the widget business logic may contain scripts that may encapsulate the widget's business logic that may get executed when the

widget may be loaded on the mobile device. In an example, the widget permissions may include an access control list that may include permissions that the ecosystem operator may have offered to the widget while running within the mobile application.

[00184] In accordance with the exemplary and non-limiting embodiments, the present disclosure provides management of a widget that may be grouped into two large phases. The first phase may include the widget developments and the second phase may include widget usage. The widget development phase may include all steps leading up to the widget being ready for use and the widget usage may include all steps relating to the utilization of the widget and its lifecycle.

[00185] In accordance with the exemplary and non-limiting embodiments, the present disclosure provides a method 1400 as illustrated in FIG. 14 of developing a widget in accordance with an example. At 1402, the method 1400 may start and at 1404, the method 1400 may allow to build the widget. In an example, the developer may use the SDK offered by the MTP to build the widget in accordance with the requirements specified by the service provider. At 1408, the method 1400 may test the widget. In an example, the system may perform static code verification on the widget's Java-scripts. At 1410, a decision may be made such as to complete the widget development. If the widget development is not completed, the method 1400 may move to 1404, where the developer may work further on the widget. If the widget development is completed, the method 1400 may move to 1412 where the service provider may review the widget. If it is determined at 1414 that the service provider is not satisfied with the widget, the method 1400 may move to 1404 where the developer may incorporate the feedback of the service provider to build the widget.

[00186] On satisfaction of the service provider, at 1418, the widget may be submitted to the ecosystem operator. In an example, an employee of the service provider may submit the widget through an online mode of submission to the ecosystem operator. At 1420, the ecosystem operator may verify the widget such as to identify any errors in the widget. At 1422, the ecosystem operator may determine passing or failing of the widget. If the widget has not passed the requirements of the ecosystem operator, at 1424, the ecosystem operator may reject the widget and the method 1400 may end at 1428.

[00187] If the widget has met the requirements of the ecosystem operator, at 1430, the widget may be certified and at 1432, the widget may be published. Further, at 1434, a smoke test may be performed on the widget. At 1438, it may be determined as whether the widget has passed the smoke test. If the widget may not pass the smoke test, at 1440, troubleshoot may be initiated and the method 1400 may move to 1420 to re-verify the widget. Otherwise, at 1442, the ecosystem operator may open the widget to the public so that the public may download and use the widget. At 1444, the method 1400 may end.

[00188] In accordance with the exemplary and non-limiting embodiments, the present disclosure provides a method 1500 of widget usage in accordance with an example as illustrated in FIG. 15. At 1502, the method 1500 may start and branch into three different steps. At 1504, the wallet user may browse the widget registry such as to identify one or more widgets. At 1508, the wallet user may select the required widget and at 1522, the method 1500 may allow the wallet user to download the widget.

[00189] In an example, at 1510, the wallet user may receive an update notification for the widget that may be installed in the device of the wallet user. The wallet user may desire to download the update for the widget and accordingly, at 1522, the method 1500 may allow the widget user to download the latest version of the widget that may be available at the MTP server. In an example, at 1512, the service provider may enroll with the MTP server for providing one or more services to the end users. At 1514, one or more service may be approved and at 1518, the ecosystem operator may notify the wallet user regarding the approved service of the service provider. AT 1520, the wallet user may read the notification and at 1522, the method 1500 may allow the wallet user to download the widget associated with the notified service of the service provider.

[00190] At 1524, the wallet user may install the widget and accordingly, at 1528, the ecosystem operator may be notified regarding installation or updation of the widget by the wallet user. At 1530, service provider may be notified regarding the installation of the widget.

[00191] In accordance with the exemplary and non-limiting embodiments, the present disclosure provides a method 1500 for notifying the service provider regarding removal of the installed widget. At 1532, the wallet user may use the widget that may be installed in the device of the wallet user. At 1534, a determination may be made as to whether the

wallet user is interested in the continuous use of the installed widget. If the user may want to use the widget in the future, the method 1500 may move to 1532 where the user may continue to use the widget. Otherwise, the method 1500 may move to 1538, where the widget may be removed from the device. At 1540, the server may be notified regarding the removal of the widget and at 1542, the service provider may be notified regarding the removal of the widget.

[00192] In accordance with the exemplary and non-limiting embodiments, the present disclosure describes a use case wherein the service provider may be created. The service provider may also be referred to as a widget provider. As discussed above, widget providers may have entered into business relationships with the ecosystem operators such as to create and promote their widgets. In an example, before a widget can be created in the system, the service provider entity may be required to be updated in the system. For example, an employee of the ecosystem operator may log into a system management console and may select an option of widget management. Further, the employees may select a link to add the service provider, fill the corresponding details associated with the service provider and create the service provider record. In an example, the system may e-mail the service provider details of their enrolment along with their unique identifier.

[00193] In accordance with the exemplary and non-limiting embodiments, the present disclosure describes a use case Submit Widget, wherein the widget may be stored on the MTP test bed/UAT system within the ecosystem operator's domain. In an example, the widget may have been developed and created using a correct version of the MTP SDK tools. Further, the widget may be internally tested and approved for submission by the service provider. In an example, the MTP system may allow submission of the widget in two modes: on-line and offline.

[00194] In an example of offline submission of the widget, an employee of the service provider may send the widget bundle associated with the widget to an authorized employee of the ecosystem that may select the widget management from the MTP system management console. Further, the ecosystem's employee may select a link to submit the widget and upload the widget bundle. The MTP system may validate the widget bundle and the system may create widget records and set the state of the widget as

SUBMITTED. Accordingly, the system may confirm the employee of the service provider that the widget has been successfully submitted.

[00195] In an example of online submission of the widget, the employee of the service provider may configure an Eclipse plug-in tool to point to the correct MTP server. Further, the service provider's employee may configure their unique widget provider ID and select the option of "submit widget". The Eclipse tool may create a widget submission request to upload the widget to the MTP server that may validate the widget bundle, create widget records and set state of the widget as SUBMITTED. The MTP system may return confirmation that the widget has been successfully submitted and may send a notification to the employee of the ecosystem that a new widget has been submitted.

[00196] In accordance with the exemplary and non-limiting embodiments, the present disclosure describes a use case Validate Widget, wherein the widget may be successfully tested. In an example, the widget has been submitted to the MTP and may be ready for testing. The employee of the ecosystem (EOE) may select the widget management from the MTP system console. Further, the EOE may select a link for a widget listing that may show all widgets in the SUBMITTED state. The EOE may select the widget to be tested and the MTP may prompt a confirmation with a checklist of all required verifications that may be performed. The EOE may select the Test option and the system may change the state of widget from SUBMITTED to TESTING. In an example, the system may perform static code verification on the widget's Java scripts and may confirm that static code verification has passed. The system may display the verification result and prompt the EOE to review and confirm it. The EOE may confirm the verification result approval and may perform all other tests as per the ecosystem guidelines.

[00197] In accordance with the exemplary and non-limiting embodiments, the present disclosure describes a use case Approve Widget, wherein the widget may be approved and published. The EOE may log into the MTP System Console and select a link for approving the widget in the widget management section. The system may show all widgets in the TESTING state and thereby, the EOE may select the widget to approve. Accordingly, the system may prompt confirmation with a checklist of all required

verifications to be performed and the EOE may confirm and check all tests that have been performed. Further, the EOE may enter any comments/remarks and select an Approve option. Thereafter, the system may change the state of widget to APPROVED and may confirm that the widget has been approved.

[00198] In accordance with the exemplary and non-limiting embodiments, the present disclosure describes a use case Publish widget, wherein the widget may be published and ready to be downloaded to the user's wallet application. The EOE may log into the MTP system console and select a link for publishing the widget in the widget management section. The system may show all widgets in the APPROVED state and the EOE may select the widget to be published. The EOE may enter any comments/remarks and select a publish option. The system may prompt for confirmation and the EOE may confirm that widget may be published. Accordingly, the system may change the state of widget to PUBLISHED and generate a signature for the widget. In an example, the system may create a HASH of the widget bundle to be securely stored on the wallet in the SE and sign the widget bundle with an HMAC and have a shared key with the wallet. In an example, the system may sign the widget bundle with an RSA key and include certificate within wallet. Thereafter, the system may confirm that the widget has been published.

[00199] In accordance with the exemplary and non-limiting embodiments, the present disclosure describes a use case Suspend Widget, wherein the widget state may be transitioned to SUSPENDED and it may no longer be actively downloaded. In an example, the EOE may log into the MTP system management console and select a link for suspending widgets in the widget management section. The system may show all widgets in the PUBLISHED state and the EOE may select a widget to suspend the widget. The EOE may enter any comments/remarks and select a SUSPEND publish. Further, the system may prompt for confirmation and the EOE may confirm the suspension of the widget. Accordingly, the system may change the state of widget to SUSPENDED and may confirm that the widget has been suspended. In an example, the system may generate an email notification to the service provider that the widget has been suspended.

[00200] FIG. 16 illustrates a transaction flow diagram 1600 for installing a widget, according to an example. As shown in FIG. 16, the client 1602 may include a wallet

1604 and a wallet runtime 1608. The one or more components of the server 1610 may include mserver 1612, personalization module 1614, and an enterprise service bus (ESB) 1618.

[00201] In an example, at 1620, the server 1608 may use the ESB 1618 to issue soft card, service provider ID, and the product ID to the personalization module 1614. At 1622, a request such as get widget profile, identities such as provider ID, product ID, platform, status, and SDK version may be sent to the mserver 1612. At 1624, an object is pushed towards the wallet runtime 1608. At 1628, the personalization module 1614 may communicate a request to the mserver 1612 to create widget instance that may include a wallet ID, provider ID, and an applet ID. At 1630, it may be determined that the object is a platform object. At 1632, the platform object may be handled and processed. At 1634, the wallet runtime 1608 may communicate a request to consume object JS. At 1638, the wallet 1604 may respond with an object for downloading object JS.

[00202] At 1640, the wallet runtime 1608 may communicate a request to download the widget to the mserver 1612. At 1642, the mserver 1612 may prepare the widget response and mark widget status as SENT_FOR_DOWNLOAD. At 1644, the mserver 1612 may send widget binary to the wallet runtime 1608. At 1648, the wallet runtime 1608 may install the widget and accordingly, at 1650, the wallet runtime 1608 may communicate to the wallet 1604 that widget installation is completed. At 1652, the wallet 1604 may send an acknowledgment to the wallet runtime 1608, which in turn at 1654, may acknowledge to the mserver 1612. At 1658, the mserver 1612 may mark the widget status as INSTALLED. At 1660, the mserver 1612 may further communicate the acknowledgement to the personalization module 1614 which in turn, at 1662, may execute the business flow. At 1664, the personalization module 1614 may notify the card issuer via ESB 1618. In addition, at 1668, the wallet runtime 1608 may acknowledge back to the acknowledgement of the wallet 1604. At 1670, the wallet 1604 may send a confirmation and at 1672, the wallet runtime 1608 may clear the reliability queue.

[00203] In accordance with the exemplary and non-limiting embodiments, the present disclosure describes association of an instrument and wallet. In an example, the platform (i.e., MTP) may get an instruction such as to push an instrument (e.g., soft card) in the platform. Various APIs may be used and/or published such as to integrate the issuer

system with the MTP. The MTP may get the service product from the master records based on the identification parameters. In an example, the identification parameter may include provider ID, type of the service product, a network provider type, and a correlation identity (e.g., a mobile number), and the like. The platform may fetch wallet account information using the correlation identity. The platform may create the instrument (i.e., service account reference) record that may be associated with the wallet account. The platform may mark the instrument record status as REGISTERED, which may be updated by the issuer system. Further, lifecycle of the can be controlled using the status such as WAITING_FOR_ACTIVATION, ACTIVE, CLOSED and the like. The platform may deliver the instrument by triggering the wallet using a PUSH notification or a piggyback channel.

[00204] FIG. 17 illustrates an example state diagram depicting a lifecycle of the widget in the MTP system. At 1702, the widget may be submitted with the MTP. In an example, the EOE may submit the widget with the MTP. In another example, the employee of the service provider may use online submission feature to submit the widget with the MTP. As shown in the FIG. 17, at 1704, the widget may be submitted any number of times while the state is SUBMITTED. This may allow the developer to update the widget before testing may begin. In an example, the widget may not be re-submitted when the widget is in a state other than the SUBMITTED state.

[00205] At 1708, the widget state may be updated to TESTING. During this state, the system may perform a plurality of tests (for example, static code verification) on the widget. At this state, the system may change the state of the widget either to APPROVED or REJECTED, depending on the results of the tests. At 1710, the REJECTED state of the widget may not allow the developer to update the rejected widget. In other words, once the system may reject the widget, a new widget bundle (with the same widget ID & version number) may be submitted and the process may be repeated. As a result, a new record in the widget table may be created. Further, the widget may not enter the SUBMITTED state if a widget with the same widget ID and version number already exists in any state except SUBMITTED or REJECTED. At 1712, the widget state may be updated to APPROVED, depicting that the widget may have cleared the testing stage. At 1714, the approved widget may be published and may

be available for downloading. Further, it may be possible that the published widget may need to be updated or the system may need to suspend the published widget. Accordingly, at 1718, the widget may be suspended depicting that the widget may not be available for downloading. In an example, a widget with a mathematically greater version number can be SUBMITTED only if all previous version numbers are in the APPROVED, PUBLISHED, REJECTED, or SUSPENDED state. In an example, the system may log the state transition of the widget along with timestamp, unique identification of the actor, and audit trail of comments that have been entered by the actor.

[00206] The system may be configured to include various APIs for service account management. For example, the system may be configured to include creation related APIs wherein the APIs may create the service product record, service provider record, service account reference record and the like. In an example, the system may be configured to include updation related APIs wherein the APIs may update the service account reference record, service account reference status (such as to control the service account reference lifecycle), and the like. In an example, the system may be configured to include listing related APIs wherein the APIs may list of all registered service products available in the system, all registered service providers available in the system, and the like. In an example, the system may be configured to include retrieval related APIs wherein the APIs may retrieve service account reference records based on wallet id filter criteria, service account reference records based on widget ID filter criteria, service account reference records based on product ID filter criteria, service account reference records based on status filter criteria, service account reference records based on service account reference number, service account reference records based on service product provider id filter criteria, service account reference records based on widget instance id filter criteria, service product based on service product id filter criteria, service product provider records by provider id filter criteria, and the like. In an example, the status values can be REGISTERED, WAITING_FOR_ACTIVATION and ACTIVE. The status values can be extended for customization or to introduce more granular level of Service Account Reference status. In an example, the service account reference number can be a unique identifier shared by issuer system which may act as the correlation id

within wallet platform and issuer system. In an example, the system may be configured to include updation related APIs wherein the APIs may update service product record, service product provider record, and the like.

[00207] In accordance with the exemplary and non-limiting embodiments, the present disclosure describes details of various data mappings. In an example, the system may include various data fields for data associated with the service provider. An ID may be a unique system identifier of the service product provider and may be auto generated. The name field may be used for displaying the name of the service provider and data associated with the service provider may be inputted while creating the entry of the service provider. Type filed may identify the service type of the service provider and category field may define the category of the service product that may be offered by the service provider. Email filed may include an email address of the service provider. Further, the system may include one or more attributes that may be stored in the form of a key value pair (e.g., contact number) and may be used to introduce any custom attributes to the service provider.

[00208] In an example, the system may include various data fields for data associated with the service product. An ID may be a unique system identifier of the service product and the ID may be auto-generated by the system. A provider ID may be the unique system identifier of the service product provider as discussed above. A product ID may be a unique name (e.g., Gold bank card, Premium credit card, and the like) of the product and the system may include an application ID of an application (e.g., banking) associated with the service product. Type filed may identify the type (e.g., credit card) of the service product and category filed may identify category (e.g., payment) of the service product. Name filed may identify display name of the service product and description may identify description of the service product. Further, the system may include attributes stored in the form of key value pair and attributes can be used to introduce any custom attributes to the service product entity.

[00209] In an example, the system may include various data fields for data associated with the instrument (service account reference). An ID may identify a unique system identifier of the instrument and the system may auto generate the ID. A wallet ID may identify a unique identifier of the wallet. An issuer ID may identify a unique identifier of

the issuer and a product ID may identify a unique identifier of the service product. A widget instance ID may identify a unique identifier of the widget instance and an account reference number may identify a unique identifier of the instrument at the issuer end. Status may identify status (e.g., created, activated, suspended, and the like) of the instrument. Similarly, the system may include wallet platform status. Provisioning time stamp may identify provisioning date (time stamp) of the instrument and download time stamp may identify download date (time stamp) of the instrument. Activation time stamp may identify an activation date (time stamp) of the instrument and discontinue time stamp may identify discontinue date (time stamp) of the instrument. Last activity time stamp may identify last activity date (time stamp) of the instrument and last state change time stamp may identify time stamp of the last status change. Further, the system may include nickname filed such as to identify user provided name of the instrument, other fields such as OTA TX data, PROXIMITY TX data and the like.

[00210] An expiry date may identify an expiry date of the instrument and due date may identify due date of the instrument. The system may include extra attributes which can be used to accommodate any additional information. These attributes may not be used by platform and may be used mainly for the widgets. Further, Logo Image may identify an image which can be used to display Soft Card on the Wallet application. This image can be cobranded as per the issuer's branding definition. Card Image may identify an image which can be used to display Soft Card on the Wallet application. This image can be cobranded as per the issuer's branding definition. A product type may identify a unique identifier of the service product configured at the wallet platform during setup. An AID (e.g. HEXSTRING) may identify an applet ID that may be a unique identifier of the applet installed on secure element during personalization. OTA Permission may identify Over the Air service permissions flag that can be managed by the issuer to control OTA service permissions and Proximity Permission filed may identify proximity service permissions flag that can managed by the issuer to control proximity transaction or services. Soft card brand ID may include branding information ID and a network operator ID may identify a mobile network operator.

[00211] In an example, the system may include various data fields for data associated with the widget provider. An ID may identify a unique system identifier of the widget

provider and the system may auto generate the ID. Name file may identify display name of the widget provider and description may identify description of the widget provider. An email may identify an email address of the widget provider and status may indicate status of the widget provider. A creation date time stamp may identify creation date (time stamp) of the widget provider and an approval date time stamp may identify approval date (time stamp) when the widget provider has been approved.

[00212] In an example, the system may include various data fields for data associated with the widget provider. An ID may identify a unique system identifier of the widget profile and the system may automatically generate the ID. An application ID may identify a widget application ID and an application name filed may identify an application name with which the widget may be associated. A provider ID may identify a unique identifier of the widget provider and a provider name may identify the name of the widget provider. A minimum runtime version may identify minimal required wallet platform version and a version may identify version of the widget. Status filed may identify the status (e.g., created, published, and the like) of the widget profile.

[00213] In an example, the system may include various data fields for data associated with the widget instance. An ID may identify a unique system identifier of the widget instance and the system may auto generate the ID. A widget bundle ID may identify a foreign key to wsc_widget_bundle. A widget profile ID may identify a unique identifier of the widget and the system may define foreign key: ID of widget profile. A wallet ID may identify a unique identifier of the wallet. A provision time stamp may identify a widget provisioning date (in time stamp) and an installation time stamp may identify widget installation date (in time stamp). An activation time stamp is an optional field and a removal time stamp may identify widget instance removal date. A status filed may identify the status of the widget instance.

[00214] In an example, the system may include various data fields for data associated with the widget permissions. An ID may identify a unique system identifier of the widget instance and the system may automatically generate the ID. A widget profile ID may identify a reference to the widget profile ID and a widget provider ID may identify a reference to the widget provider ID.

[00215] In an example, the system may include various data fields for data associated with the widget bundle. An ID may identify a unique system identifier of the widget instance and the system may automatically generate this ID. A widget bundle may identify binary data of the widget file (.wgt) and a device software class may identify device software class associated with widget bundle. A widget profile ID may identify a reference to the widget profile ID and a bund hash may identify hash of the widget bundle file.

[00216] In some exemplary and non limiting embodiments, The methods and systems described herein of multi-domain ecosystem secure personalized transactions includes a mobile wallet (also referred to as a wallet) container that may support installing and managing multiple widgets. Each of these widgets may be developed and owned by different issuers and service providers. The wallet may be configured to fulfill various security requirements. Some of the exemplary key security environments described here include widget integrity, widget access control, and widget isolation. These are described as follows:

[00217] In an example, the widgets may be provisioned to a wallet runtime securely using a protected provisioning server like a wallet server. The widgets may be installed on handset. Any tampering to the widgets may be automatically detected while on the handset, and necessary actions may be taken.

[00218] In an example, access to an issuer security domain applet may be exclusive to the issuer widget. The system may prevent any issuer widget to access another issuer's security domain. The system may prevent the widget from accessing System Actions which may not be granted explicitly to the given widget. The system may prevent any issuer widget from accessing URLs/domains other than what are explicitly granted to the given widget.

[00219] In an example, any issuer widget may have exclusive access to the issuer's workflow code. The system may be configured so that any issuer widget may not access any other issuer's workflow code. The system can be configured so that any issuer widget may have exclusive access to the issuer's user interface (UI) Code. The system may be configured so that any issuer widget may not access any other issuer's UI code. The system may be configured so that any issuer widget cannot access any other widget's

persistent data. The system may be configured so that any issuer widget may not access any other widget's static resources (Images, Files, etc).

[00220] In some exemplary and non limiting embodiments, The methods and systems described herein of multi-domain ecosystem secure personalized transactions includes an access control mechanism (ACM) may be configured to manage "resources" against a given "policy". In some exemplary and non limiting embodiments, The methods and systems described herein of multi-domain ecosystem secure personalized transactions includes, few exemplary resources being managed may include Widget Workflow Code, Widget Data, Widget UI Code (Screens), Secure Element, Server Communication. In an example, to determine the "policy", the present disclosure may have an Access Controller that may use the given widget's "Widget Descriptor" as the policy file. The Widget Descriptor may be a simple Key-Value pair structure which is created by a widget developer (to describe the necessary resources) and signed by a mobile transaction platforms (MTP) server based on the wallet operator's processes . The access control mechanism may involve a widget descriptor life cycle and a mClient access control in the overall process of managing resources against a given policy. These two aspects of the process may be described below.

[00221] FIG. 18 illustrates the widget lifecycle from development to execution. In some exemplary and non limiting embodiments, the widget life cycle (represented by 1800) may have phases with responsibility demarcation between issuer and JVL's sub systems. The phases may include widget development, widget verification and approval, widget delivery, widget installation and widget execution. The widget development phase of the widget life cycle may include a toolkit 1802. In an example, the toolkit 1802 may be based on Microsoft visio®. A mobile transaction platform tool (MTP) 1804 may support the toolkit 1802 to facilitate widget development. The widget development may be followed by a widget package generation stage. During, the widget package generation a widget bundle 1808 may be formed by the MTP tool 1804. The widget bundle 1808 may comprise a widget descriptor, a screen support file and a theme support file (these may be explained below in greater detail). The widget bundle 1808 may be sent for a static verification, policy approval or policy provisioning by the MTP tool 1804 to a mobile transaction platform (MTP) 1810. In an example, the policy provisioning and the widget

provisioning profile may be provided using a trusted service manager (TSM) 1814. In an example, the widget provisioning may be providing using an access control list (ACL), as explained by way of later sections. The MTP 1810 may be configured for publishing the widget bundle 1808 after verification. In an example, widget provisioning and publishing may be synchronized and take place simultaneously over the MTP 1810 and the TSM 1816. The MTP 1810 and the TSM 1816 may send the widget bundle 1808 after verification and publishing to an mClient 1818. The mClient 1818 may be configured to include the widget bundle 1808, a block for platform services, a block for data services, a block for communication services and a block for a secure element. The mClient 1818 may define a runtime footprint for the widget bundle 1808. The mClient may be explained later by way of FIG. 21.

[00222] In some exemplary and non limiting embodiments a first phase in widget life-cycle may be widget development. In an example, the mobile transaction platform (MTP) may support a Microsoft Visio based widget development toolkit to automate Widget development. The widget development may take place in an issuer domain. The widget may be made up of a plurality of widget artifacts. In an example, the widget may be made up of a screen.xml file, a content.xml file, a theme.xml file, a plurality of script files, a plurality of images, and a widget descriptor. In an example, the widget development toolkit may generate widget's artifacts like screen, content etc. along with the widget descriptor. The widget descriptor may capture widget access requirements (or requests) for different resources like Actions/URLs/Applets etc by the developer.

[00223] In some exemplary and non limiting embodiments, a next phase in widget life-cycle may be widget verification and approval. In an example, the widget approval step may involve validation of the Widget. In an example, the widget verification may take place at a wallet platform (for example, a JVL wallet platform – certification tooling). The widget approval step may further involve approval or provisioning of the Widget. The present invention provides a set of tools for a static analysis of the widget, which may help the overall approval process. In an example, the technical process of widget approval may include various steps. A first step may be creating a static analysis report for the given widget, indicating any warnings/duplications/conflict. In an example, these indications may relate to an application protocol data unit conflict (APDU), an uniform

resource locator (URL) Conflict, a presence of too many external resources, use of restricted/controlled application programming interfaces (API). A next step may be capturing a manual approval of the widget recording the approval conditions and accepted/ignored warnings. A next step may be creating a widget provisioning profile. The widget provisioning profile may include the hash of the widget binaries and the ACL. A next step may be creating a widget package to be downloaded using a wallet channel.

[00224] In some exemplary and non limiting embodiments, a next phase in the widget life cycle may be the widget delivery and installation. In an example, the widget may be ready for delivery once it is certified and may be available for download from the wallet platform. In some exemplary embodiments, the widget download and installation may be triggered due to various events. The events may include a new card provisioning event by the issuer.

[00225] In some exemplary and non limiting embodiments, the mClient's widget manager may execute various steps at the time of installing the widget. A first step may be to verify the integrity of the widget bundle after the widget bundle has been downloaded. In an example, the verification may be done using secure element services. In case, the integrity of the widget bundle is verified, the widget bundle may be added to the mClient. In case, the integrity of the widget bundle is not verified, an error message will be shown and the widget bundle will not be added to the mClient.

[00226] In some exemplary and non limiting embodiments, a next phase in the widget life cycle may be the widget execution. The Widget's integrity and access profile may be determined by the wallet manager for each run of the widget. The widget execution and integrity check may include a load time integrity check, and a runtime access control. The load time integrity check may check if the widget may be ready for execution post successful installation. The Hash may be verified every time a widget is loaded for execution. This validation may be required to ensure that the widget or its data has not been tempered with during the idle time of the wallet or the widget. The runtime access control may check if the Widget may start its function once it passes through the hash verification step at its load time. The widget access controller may ensure that the widget

trying to access a particular resource may be as per proper authorization i.e. isolation rules.

[00227] The mClient Access Control (also referred to as the access control) - While the Widget Descriptor Lifecycle describes the process of "Establishing the policy", the mClient Access Control mechanism ensures the "Enforcement of the policy". This section describes a high level overview of the Access Control Mechanism.

[00228] FIG. 19 illustrates a high level view of the access control mechanism. As illustrated in FIG. 19, the resource access may be controlled by via an access controller 1902. The access controller 1902 may determine the access rights based on the widget descriptor. The access controller 1902 may avail a particular resource for example, a screen or an action or an applet and the like to the widget. In exemplary and non-limiting embodiments, the access controller 1902 may include a set of design considerations. In an example, the access controller 1902 may require a system for data storage 1904. The data storage may be configured to store widget binaries. In an example, the widget profile provisioning and ACL information may be securely stored in a secure element 1908. The secure element 1908 may be in communication with the access controller 1902. The widget profile provisioning and ACL information may include a wallet security domain 1910, a wallet applet 1912, an issuer security domain 1914, and an issuer applet 1918. The access controller 1902 may be in communication with a widget policy file 1920.

[00229] The access controller 1902 may be in communication with a near field communication (NFC) chip manager 1922, a resource manager 1924, and a communication manager 1928. The NFC chip manager 1922, the resource manager 1924, and the communication manager 1928 may be in communication with a script execution platform 1930 that can be in communication with a screen manager 1932. As illustrated in FIG. 19, the widgets may not have API level access to any native resources and all resource usage may have to be facilitated by the access controller 1902. For example, the system for data storage 1904 may have to communicate with the access controller 1902, which may facilitate access of the NFC chip manager 1922, the resource manager 1924, and the communication manager 1928 for the system for data storage 1904. In an example, the screen manager 1922 may be allowed to execute only one

widget at a particular point of time. In exemplary and non-limiting embodiments, there may be only one access controller 1902 for entire system.

[00230] In an example, a widget may be required to get its hash verified every time it may be loaded or reloaded. In an example, only one widget may be allowed to be executed at.

[00231] FIG. 20 illustrates a method 2000 for loading and execution of a widget with respect to the access controller.

[00232] The method 2000 may require the mClient to load a wallet home at step 2002. The mClient may be in interfacing with the user. The user may select a widget from the wallet home after the wallet home may be loaded onto the mClient. The mClient may be configured to open the widget selected by the user at step 2004 of the method 2000. The mClient may be in communication with an application database. The mClient may be configured for loading the widget bundle of the selected widget from the application database at step 2008. The mClient may be in communication with the secure element. The secure element may verify the integrity of the widget bundle by performing hash verification at step 2010. As mentioned above, the hash may be verified every time a widget may be loaded. In an example, the widget bundle may be contained in a wallet applet for has verification. The hash may be stored at the time of widget provisioning. The hash verification may be vital for the method 2000 and in cases of hash verification failure; the widget execution may be stopped. In cases where hash verification is successful, the mClient may retrieve the widget descriptor for the selected widget from the secure element. The widget descriptor may be stored in the main memory of the MTP at step 2012 of the method 2000. The screen manager may load a next screen onto the mClient after the widget descriptor may be retrieved at step 2014. The mClient may request a platform service execution at step 2018. In an example, he service may be requested by specifying various parameters. The platform service may be exclusive the selected widget (platform services explained by way of FIG.21). The mClient may verify the widget descriptor so as to ensure that the widget may have privileges to access the platform service requested at step 2020. The mClient may execute the platform service requested in case the selected widget may have privileges to access the selected platform service at step 2022 of the method 2000. The method 2000 may require the

widget descriptor verification of the step 2020 for further execution of the widget. For example, if the widget descriptor may be found to not have access to the selected platform service, then the mClient may terminate the method. The mClient may communicate widget execution to the user and the mClient may prompt the user to exit the widget after execution at step 2024 of the method 2000. The mClient may destroy the widget after execution at step 2028. The mClient may be configured to load another wallet home if he user may want to execute another widget.

[00233] FIG.21 illustrates a widget runtime footprint 2100 based on widget access control policy. Each widget may create a logical footprint at runtime based on its access control policy. The access control policy may be defined in the widget provisioning profile. Each logical footprint may contain UI workflow resources 2102, actions/services resources 2104, data resources 2108, online resources or communication resources 2110, and secure element resources 2112. As illustrated by FIG. 21, a boundary of the widget runtime footprint 2100 may described herein.

[00234] Each widget may have its own screen 2114 and content files. A widget A may not use content files from a widget B and vice versa. The widget runtime footprint 2100 would use a native platform 2118 that may be common for all widgets. Every widget may have a set of privileged services 2120 that may be used by a particular widget only. For example, the widget A may use only a widget A privileged service 2120A from the privilege services 2120 and the widget B may only use widget B privileged service 2120B from the privilege services 2120. The widget runtime footprint 2100 has a platform services block 2122 that may be a part of the mClient and therefore, may be used by any of the widget A or widget B or any other widget. Any widget may access a common data 2124. The common data is stored by default in the widget runtime footprint 2100. In an example, the widget may define, its own data store if authorized to do so. For example, the widget A may define a data source 2128A and the widget B may define a data store 2128B.

[00235] The mClient may have a common URL platform 2130. Any widget for example, widget A or widget B may access data from the common URL platform 2130 by default. In an example, each of the widget A and widget B may define their own communication end points 2132A and 2132B. The communication end points may be

domain names. In this example, widget A may use the communication end point 2132A and widget B may use the communication end point 2132B.

[00236] Any of the widget may not have default access to any SE applets. Each applet access that may be allowed for the given widget may be defined explicitly in the widget descriptor. One or more Widgets may have access to the same SE applet, based on the permission granted by the server's signature.

[00237] Detailed design of the widget isolation requirement. In some exemplary and non limiting embodiments, The methods and systems described herein of multi-domain ecosystem secure personalized transactions includes an overall lifecycle of the policy file used for the access control. Various stages describe important files/data-structure and their transformation/processing, which ultimately results in access control implementation at the mClient container. The stages may include a development stage, a provisioning stage, a widget launch stage, and a widget execution stage. These stages may be described herein.

[00238] The development stage may include the widget descriptor file. The widget descriptor file may be provided by a developer (for example, the issuer). The widget descriptor file may describe a set of resource requirements from the developer's perspective. In an example, during the development stage the widget along with the widget descriptor file may be submitted for the widget approval process. In exemplary embodiments, the resource requirements may be justified during the widget approval process. The widget may be signed by a provisioning authority (for example, JVL) after, the widget has been approved.

[00239] The provisioning stage may include a widget provisioning profile. The widget provisioning profile may be created by the provisioning authority as a part of approval process. The widget provisioning profile may be a derivation of the widget descriptor file with some additional information. In an example, the additional information may include a user-ID, a widget hash and the like. The widget provisioning profile may be provided using a trusted service manager (TSM). The widget provisioning profile may be stored inside the secure element (SE), using the wallet authentication / companion applet. The SE may provide a service to the wallet to accept the user ID and the widget hash and return the verification status as well as the ACL.

[00240] The widget launch stage may include at least one widget access record. In an example, the widget launch stage may include more than one widget access record. The widget access record may be created and maintained by the widget manager inside the wallet's main memory. The widget access record may be a simple record structure where the resource permissions may be stored. When a widget is launched, the widget manager may contact the SE to read the ACL and the widget manager loads up the widget access records in the main memory.

[00241] The widget execution stage may include a widget access context object. The widget access context object may be a runtime object which may provide a simple method to check permissions for the widget execution. The widget access records may be used to populate the widget access context object. The widget access context object may internally maintain a simple Map (key-value pair) of permissions. The widget access context object may be used by all access aware resource managers to check the given widget's permissions. The resource managers may internally decide whether the given resource needs to be checked against policy or not.

[00242] In some exemplary and non limiting embodiments the widget descriptor may be a central place where all the access rights for a plurality of resources may be defined. In an example, the plurality of resources may include the action, the screen, at least one URL and SE/Applets. In an exemplary embodiment, construction of a widget may require availing different resources like actions, applets, screens and the like. In exemplary but non limiting embodiments, action may include that the mClient platform may provide 4 different types of actions i.e. system actions, core/infrastructure actions, wallet actions and any other issuer specific custom actions. In an example, actions may be a part of mClient platform and hence are shared across widgets. In an example, any widget may access core and wallet actions. In an example, the widget by default may not be allowed to access any system actions. The widget may need to specify all the system actions required by it in the descriptor. In an example, the widget may specify any widget specific action that may be an action meant for only that widget in particular. In exemplary but non limiting embodiments, every widget may be required to have an individual set of screens and content files that may not reuse the screens and content resources from any other widget. In exemplary but non limiting embodiments, a widget

may be permitted to make a connection to a plurality of URL's. In exemplary but non limiting embodiments, the widget may interface with a plurality of applets.

[00243] Widget Approval Process – FIG. 22 illustrates a widget approval process.

[00244] The widget approval process as illustrated in FIG. 22 may involve a user 2202, a widget certification manager 2204, a widget repository manager 2208, a static verification utility 2210 and a binary package generator 2212. The user 2202 may select a widget bundle 2214 from various widget bundles available to the user. In an example, the widget bundle 2214 may contain an applicaition.xml file, a content.xml file, a theme.xml file and a widget descriptor (these files may be same as described above). The widget bundle 2214 may be sent to the widget certification manager 2204 by the user 2202. The widget certification manager 2204 may also be the widget approval manager. The widget certification manager 2204 may be the main controller class driving approval process. The widget certification manager 2204 may send the widget bundle 2214 to the widget repository manager 2208, the widget repository manager 2208 may manage details about all the certified and uncertified widgets with proper timestamp details. The widget repository manager 2208 may have access to a widget repository 2218. The widget repository 2218 may contain all the certified and uncertified widgets with proper timestamp details. This information may help for audit trailing and for static verification. The widget certification manager 2204 may send the widget bundle 2214 to the static verification utility 2210. The static verification utility 2210 may validate the widget bundle 2214 against static rules. For example, if a client may have already declared client_Check_CardStatus as one of its custom actions, then any other issuer then the client itself may not be allowed to use this particular action. The static verification utility 2210 may check the widget bundle 2214 in the widget repository 2218 for any accidental/incorrect reference to assets that may be owned by other widgets. The static verification utility 2210 may automate a part of the access verification process, but manual verifications may still be required. The widget certification manager 2204 may receive the widget bundle 2214 verified against static rules from the static verification utility 2210. The widget certification manager 2204 may send the widget bundle 2214 to the binary package generator 2212. The binary package generator 2212 may generates the final binary package corresponding to the widget bundle 2214. The binary package

may be ready for over-the-air download and delivery. The binary package generator 2212 may send a binary widget file 2220 to the user 2202.

[00245] Widget Environment Aspects- Few exemplary nut non-limiting environmental assumptions and dependencies for the wallet software are described here. The environmental assumptions and dependencies may include a secure communication, a secure main memory, a secure application environment, a reliable API, a SE access management, a protected database, and a reliable rush registry. In an example, the wallet software may assume that the underlying platform and the APIs offer reliable and secure HTTP and HTTPS connections for Secure Communication. The native platform may manage a root certificate store for the secure sockets layer (SSL) verification. In an example, the wallet software may assume that the main memory allocated to the wallet software by the native OS may be protected from other applications (apps) and may be secure to hold sensitive data elements for transitional scope. The wallet software may assume that the wallet binaries may be stored and maintained securely by the application environment on the phone to provide a secure application environment. The app signature is verified by the environment using trusted content certificates. The Environment may also impose strict signing and access control rules on all apps downloaded on the phone. The wallet software may assume that all Java or similar APIs may be secured by the underlying environment and may be trusted for processing sensitive data for a reliable API. Also, the APIs may be access controlled by the environment. The wallet software may assume that the access to secure element may be managed by the environment infrastructure (for example, JSR 177) and the accesses to various secure element services may be privileged. The privilege management may be implemented by the environment based on the app signatures. The wallet software may assume that the environment may restrict access to persistent data to the owner application only and the applications may not access or modify each other's data. The wallet software may assume that the environment may provide a reliable application push registry, where the events for the wallet app are passed on to the wallet reliably (without corruption or intrusion).

[00246] The methods and systems described herein of multi-domain ecosystem secure personalized transactions includes a computer readable medium containing program

instructions. The execution of the program instructions by one or more processors of a computer system may cause the one or more processors to carry out the steps that include configuring one or more widget modules for execution via a wallet container on a mobile device. The one or more widget modules may comprise a service group of a service provider. In an example, the wallet container may operate in cooperation with an enabling tier of a multi-tier platform for providing secure personalized transactions to provide the one or more widget modules exclusive access to a workflow of the service provider. In an example, the wallet container may interpret the widget module in a runtime environment operating on the mobile device to provide the exclusive access. In an example, service provider may be a widget issuer. In an example, one or more of the widget modules issued by the widget issuer may be provided exclusive access to a workflow of the widget issuer. In an example, the wallet container may operate in cooperation with an enabling tier of a multi-tier platform for providing secure personalized transactions to provide the exclusive access. In an example, the one or more of the widget modules issued by the widget issuer may be provided exclusive access to a user interface of the widget issuer. In an example, the wallet container may operate in cooperation with an enabling tier of a multi-tier platform for providing secure personalized transactions to provide the exclusive access. In an example, the one or more of the widget modules issued by the widget issuer may be prevented from accessing a workflow of another widget issuer. In an example, the wallet container may operate in cooperation with an enabling tier of a multi-tier platform for providing secure personalized transactions to prevent accessing a workflow of another widget issuer. In an example, the one or more of the widget modules issued by the widget issuer may be prevented from accessing a user interface of another widget issuer. In an example, the wallet container may operate in cooperation with an enabling tier of a multi-tier platform for providing secure personalized transactions to prevent accessing a workflow of another widget issuer.

[00247] A multi-domain ecosystem for secure personalized transactions may include various components including server, client, and other types of components. Client components may include various software programs, applications, applets, data sets, interfaces, menus, and the like that may be described herein or elsewhere. In general

client functionality may be derived from various wallets, widgets, applets, containers, and the like that reside and/or are executable on a client device, such as a mobile phone, smart phone, tablet, universal electronic transaction facility, laptop, notebook, netbook, kiosk, appliance, utility device, automobile, plane, and any other electronic or electro-mechanical device.

[00248] In general the methods and systems described herein includes client software referred to herein and elsewhere as mClient, container, and the like which acts as an enabling infrastructure for rapid mobile application development. It provides a container-based

approach to supporting a multi-tenant application architecture. The mClient container provides a phone specific run-time environment for the wallet and widgets. The container handles complexity related to application execution, transactional security, reliability, and the like including device specific implementations. An mClient container may be based on APIs, such as device specific APIs. The container runtime environment facilitates generating a native application screen from widget descriptors. This facilitates ensuring that a touch screen button looks like a native ANDROID, BLACKBERRY, IPHONE, or other device button on the respective devices. The wallet application may be similar to a widget in that it may also be interpreted and executed by the mClient runtime functionality. However, as described herein a wallet will have privileged control over the application to maintain a highly secure and separated environment for use and access to wallet information that may include user account, personal, and transaction-related content, among other things. A wall may also be defined to provide functionality that is domain specific (e.g. financial, retail, healthcare, government, and the like).

[00249] With reference to FIG. 23, there is illustrated an exemplary and non-limiting embodiment of a widget architecture that may facilitate isolating distinct service provider widgets within a wallet container on a client device in a multi-domain ecosystem for secure personalized transactions. A client container 2302 may operate as a client runtime component for facilitating execution of mobile applications on a client device, such as a mobile telephone, computer, PDA, laptop and the like in a multidomain ecosystem for secure personalized transactions. As illustrated, one or more wallets 2304 may comprise applications and data that may provide an overall user digital wallet experience and may

be accessed through one or more secure entry points managed by the container 2302 for one or more widgets 2306. One or more widgets 2306 may comprise service groups developed and/or issued by third party service providers. In accordance with exemplary embodiments, the widgets 2306 may run within the secure environment of the client container 2302 in an independent manner isolated from the one or more wallets 2304. When implemented in accordance with exemplary embodiments as discrete units of executable code, widgets may execute and operate in a manner separate and distinct from the operation of other widgets, wallets and applets or, conversely, may operate in a cooperative manner with one or more other wallets, widgets and/or applets to perform secure transactions.

[00250] As illustrated, the client container 2302 may interface with an API layer 2308 comprising one or more device specific APIs. The API layer 2308 may support various and/or popular device environments. A phone independent API layer 2310 may ensure that applications, including, but not limited to wallets, widgets and applets, such as those developed using a Mobile Transaction Platform (MTP) software development kit (SDK), run across all supported MTP handsets. In accordance with exemplary embodiments, both the wallets 2304 and the widgets 2306 may be nearly identical in structure (e.g., screens, JavaScript, & CSS Styles), except that the wallets 2304 may have access to a wider set of privileged APIs within the API layer 2308 than do the widgets 2306. In some instances, wallets 2304 may be capable of robust user configuration and personalization particularly as compared to widgets in general and service provider derived widgets in particular. As a result, in some instances, wallets may be provided with increased access to APIs exerting control over the functionality of a client device.

[00251] In accordance with exemplary and non-limiting embodiments, when invoked, a wallet 2304 may take application control until a widget 2306 entry point is selected (e.g. by a user, wallet 2304, container 2302, external service, and the like). User selection of a widget 2306 may be via a card image, a menu option, and the like that may be available to the user through the wallet 2304 interface. In response to such selection, the selected widget may be launched within a runtime secure environment (e.g. an environment provided by the container 2302) comprising the wallet 2304 and the execution of the wallet 2304 may be suspended.

[00252] Once launched, the widget 2306 may operate to facilitate secure transactions between, for example, a wallet 2304 of a mobile client device upon which the widget 2306 is executing and a service provider, such as the service provider who issued the widget 2306. In such an instance, a widget 2306 issued by a service provider may be granted exclusive access to the workflow of the issuing service provider, resources on the mobile device, such as wallet resources, secure element resources, as well as to user interfaces of the client device. Likewise, a widget 2306 issued by a service provider may be denied access to the workflows, wallet elements, and the like of other service providers as well as to certain mobile device resources, such as user interfaces of the client device.

[00253] As separately executable units of code, widgets 2306 may be effectively isolated one from the other via the execution separation functions available from a mobile operating system and/or by aspects of the mClient including the runtime environment provided by a container. In addition, widgets 2306 may be developed by a service provider (e.g. an issuer of instruments, and the like) who may configure the widget to incorporate provider-specific security elements that may make it hard for other service-provider developed widgets from accessing its resources, workflows, and the like. A mobile transaction platform that may include such widgets, containers, wallets, service provider software (including without limitation various aspects of a wallet service center described later herein), and the like may enforce strong and diverse security in all such aspects so that, for example, widgets developed by service provider A may not access resources (e.g. server-based) of service provider B. Such strong security may be important for a variety of reasons including keeping domain-specific information separate and confidential from other domains that the user may interact with through the user's mobile device (e.g. financial domains, health domain, legal domains, business domains, personal domains, identity domains, and the like).

[00254] The methods and systems described herein of multi-domain ecosystem secure personalized transactions include a computer readable medium containing program instructions. The execution of the program instructions by one or more processors of a computer system may cause the one or more processors to carry out the steps that may include configuring one or more widget modules in a wallet container on a

mobile device. The wallet container may comprise one or more wallet modules. The steps carried out by computer readable medium may further include, limiting access by the one or more widget modules to the set of elements consisting of the one or more wallet modules, the mobile device, one or more client applications. The steps may further include, providing access to the one or more widget modules of at least one of an issuer-specific security domain and widget-specific issuer security domain.

[00255] The methods and systems described herein of multi-domain ecosystem secure personalized transactions includes a method that includes providing a container runtime environment on a mobile device. The method may further include accessing a mobile-device resident widget module to determine access constraints associated with the widget, In an example, the access constraints may pertain to electronic wallet access, device resource access, device resident ecosystem applications, network resources accessible by the mobile device, issuer-specific security domains, and widget-specific security domains. The method may further include, operating at least one of the widget modules through a container to enforce access constraints associated with the widget module. The method may further include facilitating access by the widget module to at least one of issuer-specific security domains and widget-specific security domains associated with the widget module. In an example, enforcing widget module access constraints may include preventing access to electronic wallet resources on the mobile device. In an example, enforcing widget module access constraints may include preventing access by a first widget to a wallet associated with a second widget. In an example, enforcing widget module access constraints may include preventing access by a widget to a wallet on the mobile device based on a state of the wallet. The state of the wallet may include an active state during which access by the widget may be allowed and an inactive state during which access by the widget may be prevented. In an example, facilitating access to a security domain may comprise allowing access to a digital wallet on the mobile device that may share a security domain with the widget module. In an example, facilitating access to a security domain may comprise allowing access to network resources of an issuer that may share a security domain with the widget module.

[00256] Various aspects of a mobile transaction platform, including client-specific aspects, for isolating and limiting access among domain-specific resources are described

herein above. In addition to these important security and resource access limiting functions, application-related limits on access to wallets, device resources, applications, network resources and the like may also be facilitated by the methods and systems described herein including development, deployment, execution, and transaction methods and systems.

[00257] In accordance with exemplary embodiments, a container may implement one or more triggers to control or otherwise limit access to one or more wallets , devices, client applications and network resources. Limits on access may be based on location (e.g. venue, location, network type, provider accessibility, wallet state, spending limit, credit balance, credit score, and the like). Examples of these and other limits on widget access to resources are described here. A wallet may be managed in a personalized way by a merchant so that, for example, when one enters a venue one may not be given the option of using a widget that enables a form of payment that is not accepted at the venue. In another non-limiting example, access to various widgets may be enhanced by a trigger or triggers when a user travels outside the U.S.. In accordance with yet other examples, a trigger based on credit card balance/limit may further enhance the operation of one or more widgets including, but not limited to, access by one or more widgets to client applications and network resources. In an exemplary embodiment, if a wallet is not active, such as might occur when the client device on which the container and wallet or wallets are resident is shut off or when a user has deactivated a wallet (e.g. for enhanced security of the wallet resources), an associated widget may be limited from accessing client device and network resources, wallet content, and the like normally controlled by or accessed through the wallet.

[00258] The methods and systems described herein of multi-domain ecosystem secure personalized transactions includes a method that includes configuring a widget at least partially by a networked server to provide access to an issuer-specific service group. The method may further include receiving in a secure wallet container operating on a client device, a request from the networked server to deploy the configured widget. The method may further include determining widget isolation requirements for operating the widget on the client device. The method may further include installing the configured

widget based on the isolation requirements on the mobile device through the secure wallet container.

[00259] The methods and systems described herein of multi-domain ecosystem secure personalized transactions include a computer readable medium containing program instructions. The execution of the program instructions by one or more processors of a computer system may cause the one or more processors to carry out the steps that may include deploying on a client device a first issuer-specific widget to facilitate a first trust model-based secure personalized transaction. The steps may further include deploying on the client device a second issuer-specific widget to facilitate a second trust model-based secure personalized transaction. In an example, the first issuer-specific widget and the second issuer-specific widget may be deployed in a wallet container. In an example, at least one of the first trust model and the second trust model may be one of a delegated trust model, direct trust model, and brokered trust model. In an example, at least one of the first trust model-based secure personalized transaction and the second trust model-based secure personalized transaction maybe facilitated based upon a trigger. In an example, the trigger may correspond to a monetary value of at least one of the first trust model-based secure personalized transaction and the second trust model-based secure personalized transaction.

[00260] In accordance with exemplary and non-limiting embodiments, after a widget 2306 has been developed, such as by an issuer who may be a service provider, the operation of the widget 2306 may be verified and approved and may be stored on a server for subsequent downloading to a client device or to more than one client devices in the case of a non-user-specific widget 2306. As noted, widgets 2306 can vary in nature depending upon the functional task to which they are directed. As a result various types of widgets 2306 include, but are not limited to, wallet-specific widgets, user-wallet-specific widgets, offering-specific widgets, marketing-specific widgets, audit-specific widgets, multi-purpose widgets and the like. After a successful installation on the client device, the widget 2306 is ready for execution.

[00261] In an exemplary embodiment, once a widget 2306 is installed, confirmation in the form of an acknowledgement may be sent to the server from whence the widget 2306 was downloaded. Receipt by the server of the acknowledgement may result in the server

storing information regarding the successful download of a widget 2306. For example, the server may store information indicating a version number of the downloaded and installed widget 2306. Such information may be used, for example, to ascertain the existence of a more recent version of a widget 2306 for download than has been previously downloaded. In some exemplary embodiments, the successful download of a widget 2306 at a client device may result in a confirmatory visual indicia, such as a displayed text message, being displayed to a user of the client device. Also, a user may be provided with a means for accepting the downloaded widget including, for example, performing a non-binding trial of the widget. A result of such a widget installation may include placing a new icon in a mobile transaction platform user interface of the client device. The new icon may be differentiated from other icons temporarily to facilitate indicating that it is newer that the others. Alternatively, a new widget may have no visually detectable difference to the user.

[00262] Deployed or installed widget integrity may be provided by a container (e.g. mClient container), which is described herein. One technique for ensuring widget integrity is to ensure that the widget and/or its data are unaltered. In accordance with an exemplary embodiment, in order to make sure that neither the widget 2306 nor its data has not been tampered with during the time between downloading of the widget 2306 and its execution, a hash may be employed. A hash may be computed of the application code comprising the widget 2306 and the hash verified when a widget is installed, accessed for execution, accessed during an audit of client mobile transaction platform resources, loaded for execution (e.g. in an cClient container runtime environment), and the like. In an example of verifying a hash when a widget is loaded for execution, the widget 2306 may start its function once it passes through a hash verification at its load time.

[00263] An mClient container runtime environment may also facilitate strong isolation of widgets so that distinct widgets cannot impact other widget resources, alter such resources or widget application code, and the like that may impact widget integrity.

[00264] As noted above, widgets 2306 may provide one or more secure entry points via which wallet applications may interact with the issuer of the widget 2306. In exemplary embodiments, the issuer of a widget 2306 may also be a service provider whose services are enabled, either in whole or in part, by use of the issued widget 2306 by a user. In an

exemplary and non-limiting embodiment, one or more widgets 2306 may act in concert to perform secure personalized transactions with varying degrees of trust. Specifically, different widgets 2306 may be deployed to facilitate secure personalized transactions in accordance with varying trust models, including among others direct, indirect, brokered, and third-party secured trust models for transaction execution.

[00265] In accordance with exemplary and non-limiting embodiments, the use of more than one trust model may reflect differences in the environment in which the client device and associated widgets 2306 are deployed. As noted elsewhere herein, wallet and widget issuers may select one or more trust models when configuring elements of an ecosystem. For example, an issuer, perhaps comprising a service provider, may deploy a first widget 2306 to perform secure banking transactions over the air wherein the transactions have a first level of trust. The issuer may further, for example, deploy a second widget to perform secure banking transactions at a point of sale (POS) kiosk wherein the transactions have a second higher level of trust owing to the client devices proximity to the kiosk and/or access to independent verification of the user of the device. Either or both of these widgets may be deployed to a single client device and the selection of a widget for execution may be performed by the wallet 2304, such as based on the nature of the transaction as described, the security of the available OTA network, and the like.

[00266] In accordance with other exemplary and non-limiting embodiments, widget deployment may be dependent on a variable. For example, an issuer may configure a plurality of widgets 2306 each operating with a trust model that is financial transaction amount dependent. For example, a widget requiring a relatively low level of trust may be deployed for financial transactions involving less than fifty dollars while a widget requiring a relatively high level of trust may be deployed for financial transactions involving more than fifty dollars.

[00267] The methods and systems described herein of multi-domain ecosystem secure personalized transactions includes a method that includes receiving an alert with a multi-domain secure wallet container at a mobile device that an updated version of an existing widget that is deployed on the mobile device is available. The method may further include confirming that the update is an update of a existing widget that is

deployed on the mobile device. The method may further include configuring the wallet container to receive the updated widget. The method may further include receiving the updated widget with the wallet container. The method may further include dispositioning the existing widget to prevent further use of the widget. The method may further include storing the updated widget on the mobile device.

[00268] The methods and systems described herein of multi-domain ecosystem secure personalized transactions includes a method that includes querying a server with a multi-domain wallet container operating on a mobile device for the existence of an updated widget. The method may further include receiving information from the server indicative of the availability of the updated widget. The method may further include requesting the updated widget. The method may further include receiving the updated widget at the mobile device. The method may further include dispositioning an existing widget that the updated widget supersedes to prevent further use of the existing widget. The method may additionally include storing the updated widget on the mobile device. In an example, querying the server may be performed at predefined time intervals. In an example, querying the server may be performed in response to the execution of a widget on the client device. The method may further include sending a confirmation of storing the updated widget to the server.

[00269] The methods and systems described herein of multi-domain ecosystem secure personalized transactions include a method that includes receiving a query from a mobile device for the existence of an updated widget. The method may further include transmitting information indicative of the availability of the updated widget to the mobile device. The method may further include receiving a request for the updated widget from the mobile device. The method may further include transmitting the updated widget to the mobile device. In an example, the query may be received at predefined time intervals. In an embodiment, the query may be received in response to the execution of a widget on the client device. The method may further include receiving a confirmation of storing the updated widget.

[00270] In accordance with exemplary and non-limiting embodiments, widgets 2306 stored and capable of running within the secure environment of the client container 2302 may be updated for any of a variety of purposes (e.g. periodically, such as weekly, to

provide current pricing or offers, increase a spending limit, in response to a user request, to improve security, to add access to a different trust model, and the like). In one exemplary embodiment, after reviewing and publishing the widget 2306, a widget provider may create and issue an alert indicating the availability of a new or newer version of one or more widgets 2306.

[00271] In one exemplary embodiment, the alert may be issued to only those client devices that have previously successfully downloaded (or otherwise have installed) a version of the widget 2306 to be updated. In such an instance, a message may be sent to the client device to inform the user that an updated widget 2306 is available for download and provide the user with the choice to download the updated widget 2306.

[00272] In another exemplary embodiment, a configuration file, such as might be stored on a wallet server, may contain user customizable information defining the manner in which the user desires for widget updates to occur. For example, a user may customize an associated configuration file to indicate that each time a new version of a previously downloaded widget 2306 becomes available the newer version of the widget is to be downloaded. Conversely, a client device may periodically, such as at start-up or prior to engaging in a commercial transaction, query the wallet server to downloaded newer versions of previously installed widgets 2306. In other embodiments, the configuration file may be customized by or for the widget issuer, a related wallet issuer, and the like.

[00273] In accordance with other exemplary embodiments, a widget 2306 itself may be informed through a workflow of the widget issuer that an update is available and based on aspects of the updated widget (e.g. a security update), the installed widget might take an action (e.g. disable itself, notify the user that it is no longer current, update portions of itself rather than replacing it with a new widget, etc). Another update mechanism may be realized through operation of the wallet. For example, a wallet may be updated and, as a result, may indicate that a widget is out of date, triggering various widget actions (e.g., notify the user, automatically update, request an update, and the like)

[00274] Methods and systems of providing a multi-tier mobile transaction platform for secure personalized transactions in a multi-domain ecosystem may include a method of configuring at least one mobile transaction platform-specific application programming interface on a mobile device for facilitating access to mobile device secure resources by a

wallet container executing on the mobile device. The method may further include disposing a plurality of distinct wallet containers in a memory of the mobile device, wherein each wallet container interfaces with mobile device secure resources via the at least one application programming interface. The method may further include the at least one programming application interface enforcing wallet container isolation via security criteria of a three layer mobile transaction platform. The three layers may include a personalization layer, a service layer, and an enabling layer.

[00275] The methods and systems described herein of multi-domain ecosystem secure personalized transactions include a wallet container operable on a client device. The wallet container may include a runtime component for facilitating management of widgets, wallets, security, and access to a secure element. The wallet container may further include an API-based interface for accessing business specific services provided by the widgets and for accessing client device specific resources. The wallet container may facilitate operation of a plurality of distinct wallets and distinct widgets for accessing at least one of the plurality of distinct wallets.

[00276] The methods and systems described herein of multi-domain ecosystem secure personalized transactions include a computer readable medium containing program instructions. The execution of the program instructions by one or more processors of a computer system causes the one or more processors to carry out the steps that may include configuring one or more widget modules in a wallet container on a mobile device. The wallet container may comprise one or more wallet modules. The steps may further include facilitating access to at least one of the wallet modules by an issuer of at least one of the widgets.

[00277] A wallet container 2302 may form a client-resident portion of a mobile transaction platform (MTP) enabling infrastructure. As such, a container may provide a cross platform service execution environment for client devices, such as mobile handsets, and the like.

[00278] As noted above, the methods and systems described herein generally include client software referred to herein and elsewhere as mClient, container, runtime environment, and the like, which acts as an enabling infrastructure for rapid mobile application development. It provides a container-based approach to supporting a multi-

tenant application architecture. The mClient container provides a phone specific run-time environment for the wallet and widgets. The container handles complexity related to application execution, transactional security, reliability, and the like including device specific implementations. An mClient container may be based on APIs, such as device specific APIs.

[00279] As illustrated, one or more wallets 2304 may comprise applications and data that may provide an overall user digital wallet experience and may be accessed through one or more secure entry points managed by the container 2302 for one or more widgets 2306. One or more widgets 2306 may comprise service groups developed and/or issued by third party service providers. In accordance with exemplary embodiments, the widgets 2306 may run within the secure environment of the client container 2302 in an independent manner isolated from the one or more wallets 2304. When implemented in accordance with exemplary embodiments as discrete units of executable code, widgets may execute and operate in a manner separate and distinct from the operation of other widgets, wallets and applets or, conversely, may operate in a cooperative manner with one or more other wallets, widgets and/or applets to perform secure transactions.

[00280] Referring to FIG. 24, which depicts a high level architecture view of a client-device container (e.g. mClient), detailed exemplary aspects of a container are presented. A container 2402 may facilitate user interface rendering and navigation; workflow execution; access to developer APIs; and may further include application container infrastructure. A container 2402 may facilitate widget management 2430 with a widget lifecycle manager capability 2432 , an access controller 2434, and the like while integrating widget and wallet context. A container may further include a user interface framework 2412 that may include a UI engine 2414 , screens and layout 2418, styles 2420, UI elements 2422, and the like. A container as depicted in FIG. 24 may further include a workflow framework 2424 that may leverage a script engine 2428, service manager, and may include javascript functions, scriptable objects, and the like. Other container capabilities may include database functionality 2440, such as find/search, insert/update, delete, and the like. Communications 2438 may be facilitated by a container for sending (e.g. sms, HTTP/S), encoding, decoding, attaching, detaching, and the like. Near Field Communication functionality 2448 may also be supported by the

container. A container may further facilitate security through interaction with a secure element or other secure framework as may be provided by the client, including applet installation, removal, secure channel setup, APDU exchange, and the like. Further security features provided in associatoin with a container may include cryptography 2442 (e.g. AES, 3DES, RSA, and the like), signing, verification (e.g. HMAC), hash computation and verification (e.g. SHA/1), key generation (e.g. PKCS#5), random number functions (e.g. PRNG), and the like. A container may also interact with client resources, such as a phone function to facilitate making calls, reaching and/or managing contacts, interaction with calendars to facilitate event-based functions, location-based services, and the like.

[00281] The methods and systems described herein of multi-domain ecosystem secure personalized transactions include a method of managing a plurality of transaction protocols that includes disposing a wallet module in a memory of a client device. The method may further include disposing a wallet companion applet in the memory. The method may further include executing the wallet companion applet in association with a secure personalized transaction being conducted over an ecosystem in which the client device is deployed to manage access to selected content of a secure element associated with the client device. In an example, the wallet companion applet may use the accessed secure element content to facilitate authenticating access to the wallet module to perform the transaction.

[00282] The methods and systems described herein of multi-domain ecosystem secure personalized transactions include a computer readable medium containing program instructions. The execution of the program instructions by one or more processors of a computer system causes the one or more processors to carry out the steps that may include configuring one or more widget modules in a wallet container on a mobile device. The wallet container may include one or more wallet modules. The steps may further include configuring at least one applet to execute in conjunction with at least one wallet module to facilitate management of s plurality of transaction protocols.

[00283] In accordance with an exemplary and non-limiting embodiment, a Secure Element (SE) may be utilized as a secure data storage device at the client-end, such as for mobile devices including standard smart phones. In exemplary and non-limiting

embodiments, an SE may be mounted in one or more options including an UICC, a removable MicroSD, embedded in the phone hardware, or mounted as an add-on back plate or in accordance with some other exemplary interface method.

[00284] In accordance with exemplary and non-limiting embodiments, a device application, such as the wallet, if permitted by the controlling and / or issuing entity and rest of the ecosystem, may communicate with the SE. The mode and method of communication may vary depending on the type of the SE from among the options mentioned above.

[00285] In some exemplary embodiments depending on the security model used for controlling the SE, the wallet may be permitted to install, load and manage applets in the SE. It may also be, or only be, allowed to communicate with pre-installed applets through standard methods of applet communication including sending and receiving application protocol data units (APDUs).

[00286] The wallet companion applet, as the name suggests, may have a primary function of working work with a digital wallet (e.g. a wallet application on a client device), and may be used by the client device to carry out a number of tasks including, but not limited to, storing of important and / or critical data required by one or more wallets, or for any operation that is transacted using the wallet, or controlling the functioning of other applets designed to facilitate such a control through a sharable interface and /or through the communication of APDUs designed and implemented for the purpose.

[00287] In accordance with various exemplary and non-limiting embodiments, the wallet application, in coordination with the wallet companion applet, may manage, influence and/or control the other applets so enabled, for managing their transaction protocol.

[00288] Referring to Fig 24A, which depicts a detailed view of mobile device contents, mobile device applications and secure element features are presented. Mobile device applications may include MTP applications as described herein (e.g. containers, wallets, widgets, and the like). Secure element content may include security domains 2450, wallet companion applets 2452, and wallet companion domains 2454. As depicted in Fig

24A, secure elements may be configured in a variety of physical devices including UICCs, microSDs, sleeves, add-ons, embedded elements, and the like.

[00289] A wallet companion domain 2454 may expand on a companion applet described herein to encompass an entire security domain for companion applets and features. A wallet companion domain 2454 or security domain may facilitate loading, installing, managing and deleting multiple applets in the secure element as a group associated with the security domain. Each security domain may have unique security keys that may be used by the mobile transaction platform to perform operations on the group of applets in the secure element. A wallet companion domain 2454 may be managed by adapting current technology secure element key security features to enable separate security keys for the mobile transaction platform to manage the contents of the wallet companion domain 2454 in the secure element.

[00290] Fig. 24B depicts various features, and control and data flows associated with a wallet companion applet. The wallet companion applet in the secure element may provide features such as secure storage, authentication, ID management, activation control, store/get functions, crypto functions, key storage and generation functions, PIN control, and the like. The wallet companion applet may provide a wallet interface function that may facilitate managing access to the wallet by other applets, external mobile transaction platform participants, and the like. The wallet companion applet may include an applet interface for accessing the applet in the secure element. The wallet companion applet may facilitate secure interaction with trusted security manager (TSM) servers, mobile transaction platform (MTP) servers, and other participants in a multi-domain ecosystem for secure personalized transactions.

[00291] The methods and systems described herein of multi-domain ecosystem secure personalized transactions include a method that includes disposing a wallet module in a memory of a client device. The method may further include disposing a wallet companion applet in a secure element of the memory. The method may further include executing the wallet companion applet to facilitate access to one or more additional wallet applets. In an example, the access may be facilitated through a shareable interface to mobile wallet functions. In an example, the wallet companion applet may enable at least two of couponing, security and payment.

[00292] The methods and systems described herein of multi-domain ecosystem secure personalized transactions include a method that includes disposing a container in a memory of a client device. The method may further include disposing a container companion applet in a secure element of the memory. The method may further include executing the container companion applet to facilitate access to one or more additional container companion applets. In an example, the access may be facilitated through a shareable interface to mobile wallet functions and mobile widget functions accessible through the container.

[00293] The methods and systems described herein of multi-domain ecosystem secure personalized transactions include a platform operable on a client device that includes a runtime component that may configure at least one tokenized widget. In an example, the at least one tokenized widget may facilitate at least one transaction in a multi-domain ecosystem for secure personalized transactions.

[00294] The methods and systems described herein of multi-domain ecosystem secure personalized transactions includes a platform operable on a client device that includes a runtime component that may configure at least one service provider widget in a multi-domain ecosystem for secure personalized transactions.

[00295] The methods and systems described herein of multi-domain ecosystem secure personalized transactions includes a platform operable on a client device that includes a runtime component that may configure at least one client container in a multi-domain ecosystem for secure personalized transactions

[00296] The methods and systems described herein of multi-domain ecosystem secure personalized transactions includes a platform operable on a client device that includes a runtime component that may secure account information for a plurality of accounts in a wallet.

[00297] The methods and systems described herein of multi-domain ecosystem secure personalized transactions includes a platform operable on a client device that includes a runtime component that may provide secure access to a mobile wallet by a plurality of service providers

[00298] A mobile wallet for integrating offers with transactions between an external service provide and the mobile wallet in a multi-domain ecosystem for secure

personalized transactions may include a multi-domain digital wallet operable on a mobile device, wherein the digital wallet comprises accounts from at least two distinct service providers. A runtime container for executing widgets that attempt to access the digital wallet may also be included. In addition, a plurality of mobile offer management widgets, operable in the runtime container that interface with distinct mobile offer domains by communicating with external service providers may be included. Lastly a mobile transaction capability provided by cooperative operation of the runtime container, the multi-domain digital wallet, and at least a portion of the plurality of mobile offer management widgets may also be included.

[00299] In accordance with exemplary and non-limiting embodiments, the present disclosure provides coupon applets that may be a part of a suite of applets developed to facilitate a richer, and a more convenient retail shopping experience that may be aimed to enable offer-related flows between a mobile wallet application and service provider, such as in association with a point-of-sale (PoS) terminal.

[00300] The coupon applets may support near-field communication (NFC) transactions in the form of a single tap as well as a multi-tap user experience. The one or more versions of the coupon applets may support features for attaching multiple coupons to a single transaction, presenting the attached coupons in the transaction (e.g. to a PoS terminal), communicating redemption status, managing offers, matching offers to wallets, seeking offers from service providers (e.g. through a service-provider widget), and the like. To facilitate ease of development and integration, the coupon applets may include at least two interface APIs. One API may be used for interfacing one or more offers (e.g. coupons) with a digital wallet application (e.g., a mobile wallet client as described herein), and another API may be used for interfacing the coupon with the service provider, such as through a PoS terminal. The service provider API may be facilitate access by a PoS terminal to the coupon applet for querying coupons, updating the coupon redemption status, and the like.

[00301] The one or more versions of the coupon applet may enable a straight forward flow by which a user can present coupons downloaded from one or more sources in a transaction with a service provider, such as through a PoS terminal, usually through a mobile phone with a Secure Element (SE).

[00302] FIG. 25 illustrates an offer or coupon-related transaction flow diagram. Entities participating in the transaction may include a user 2502, a wallet client 2504, a coupon applet 2508, and a PoS terminal 2510. In an example, the user may contact various coupon services, browse coupons, and decide which coupons to use for a particular purchase. The user may perform these operations via a mobile wallet application that may be associated with the wallet client 2504. At 2512, the wallet client 2504 may present the stored coupons to the user 2502 and at 2514, the user 2502 may select the coupons that may be attached.

[00303] Further, a phase 1 (also referred to herein as a pre-transaction setup phase) may be initiated. As shown in FIG. 25, during the phase 25, the wallet client 2504 may attach the selected coupons to the applet and thereby, the wallet client 2504 may indicate that the transaction can begin.

[00304] After phase 1, phase 2 (also referred to herein as a PoS transaction phase) may be initiated. During phase 2, the authentication identification (e.g. user identity and/or account data from a Secure Element) may be presented to the PoS terminal 2510 to initiate the PoS Transaction Phase. The PoS terminal 2510 then may query the coupon applet 2508 for the attached coupons and the coupon applet 2508 may send back the data pertaining to the attached coupons in response. This could take several queries based on the number of coupons. Once the PoS terminal receives all the coupons, it may process these coupons. Such processing may include validation of coupons, checking applicability, compliance to other business rules to determine applicability of the coupons, etc. At the end of the processing of this phase, redeemed coupons may be identified. The status of these coupons may be communicated to the coupon applet 2508.

[00305] After phase 2, phase 3 (also referred to herein as post-transaction query & clean up) may be initiated. During this phase, the wallet client 2504 may query the redemption status if required and reset all the coupons. The resetting of the coupons may mark the end of phase 3, and reposition the state to the beginning of phase 1.

[00306] FIG. 26 illustrates a state machine diagram of the coupon applet during the transaction flow according to an example. A state 2602 may indicate a READY FOR UPLOADING COUPONS state, indicating that all coupons from the coupon applet may have been reset, and the wallet client can now upload the coupons selected for this

transaction. In an example, as shown in FIG. 26, the applet may remain in the state 2602 if a command GET APPLET STATUS results in an error. The coupon applet may transit to a next state 2604 on receiving a valid ENABLE TRANSACTION command.

[00307] The 2604 state may include a READY FOR COUPON REDEMPTION state, indicating that the wallet client may have attached all the coupons to the coupon applet's memory, and that the PoS terminal can proceed with the phase 2 of the transaction. In an example, as shown in FIG. 26, the coupon applet may remain in the state 2604, if a command READ COUPONS results in an error. The applet may transit to a next state 2608 on receiving a valid SET COUPON REDEMPTION STATUS command. As shown in FIG. 26, the coupon applet may transit to a next state 2610 on receiving a valid ABORT TRANSACTION command.

[00308] The state 2608 may include a TRANSACTION COMPLETED state, indicating that the PoS terminal may have completed phase 2 of the transaction and COUPON REDEMPTION STATUS may have been set. Further, this state 2608 may indicate that the wallet client may now proceed with phase 3 of the transaction. The coupon applet may remain in this state till the applet receives a valid RESET COUPONS command or a valid ABORT TRANSACTION command.

[00309] The state 2610 may include a TRANSACTION ABORTED state indicating that the PoS terminal transaction (relating to the coupon presentation & redemption) may have been aborted either by the wallet client or the PoS terminal. The coupon applet may remain in this state 2610 till the applet receives a valid RESET COUPONS command.

[00310] In accordance with exemplary and non-limiting embodiments, the present disclosure provides commands to support coupon-related functionality that may be relevant for the interaction with the PoS terminal. In an example, these commands may be based on the standard smart card command and response protocol as defined by ISO/IEC 7816 for all communications with the SE. In an example, a table CAT1 may summarize coding of class bytes (CLA) and instruction (INS) bytes.

CLA	**INS**	**Command**
'00'	'A4'	SELECT

'C0'	'01'	GET APPLET STATUS
'C0'	'03'	ABORT TRANSACTION
'C0'	'11'	READ COUPONS
'C0'	'41'	SET COUPON REDEMPTION STATUS

CAT1: Command Summary CLA and INS Command

[00311] In an example, the coupon applet may return status bytes that may be coded as per ISO/IEC 7816-4. The applet may return a set of generic status bytes shown in table CAT2. The table may list status word SW1, status word SW2 and the meaning associated thereof.

SW1	**SW2**	**Meaning**
'90'	'00'	Normal processing
'67'	'00'	Wrong length
'69'	'85'	Conditions of use not satisfied
'6A'	'80'	Incorrect values in command data
'6A'	'82'	Applet not found
'6A'	'84'	Not enough memory
'6A'	'86'	Incorrect parameters P1-P2
'6D'	'00'	Instruction not supported
'6E'	'00'	Class not supported

CAT2: Generic Status Bytes

[00312] In an example, the present disclosure may include an ABORT TRANSACTION command that may move the applet from its existing state to the transaction aborted state. In an example, the ABORT TRANSACTION command may be coded according to a table CAT3. The table may list one or more codes, such as class byte (CLA), instruction byte (INS), parameter P1, parameter P2, and the like. In an example, there may be no data returned from this command.

Code	Value
CLA	‘C0’
INS	‘03’
P1	‘00’
P2	‘00’
Lc	‘00’
data	Absent
Le	‘00’

CAT3: ABORT TRANSACTION Command Message

[00313] Further, a successful execution of the command may be indicated by status bytes '9000'. The ABORT TRANSACTION command may either return a general error condition, such as listed in the table CAT1. In addition, the ABORT TRANSACTION may return one of the following specific error and warning conditions as listed in the table CAT3.

SW1	SW2	Meaning
‘67’	‘00’	Wrong length
‘69’	‘85’	Conditions of use not satisfied
‘6A’	‘86’	Incorrect parameters P1-P2

CAT4: Status Bytes for ABORT TRANSACTION

[00314] In an example, a GET APPLET STATUS command may be used to retrieve the current status of the applet. This command may return the current status of the applet. A table CAT5 may list codes and corresponding values associated with this command.

Code	Value
CLA	‘C0’
INS	‘01’

P1	'00'
P2	'00'
Lc	'00'
data	Absent
Le	'00'

CAT5: GET APPLET STATUS command Message

[00315] In response to this command, a data field may be returned. In an example, the applet status may be returned and coded as 1 byte and details of this data field can be found in table **CAT17** as described below in the description. Further, a successful execution of the command may be indicated by status bytes '9000'. The GET APPLET STATUS command may either return a general error condition, such as listed in the table CAT1. In addition, the GET APPLET STATUS command may return one of the following specific error and warning conditions as listed in table CAT6.

SW1	**SW2**	**Meaning**
'67'	'00'	Wrong length
'6A'	'86'	Incorrect parameters P1-P2

CAT6: Status Bytes for GET APPLET STATUS

[00316] In an example, a READ COUPONS command may be used to retrieve coupon data. Further, the READ COUPONS command may be coded in accordance with a table CAT7.

Code	**Value**
CLA	'C0'
INS	'11'
P1	'00'

P2	See table CAT8
Lc	'00'
data	Absent
Le	'00'

CAT7: READ COUPONS Command Message

[00317] The table CAT7 includes a reference control parameter P2 that may control the number of consecutive READ COUPONS command. Further, the reference control parameter P2 may be coded according to the following table CAT8.

P2	**Meaning**
'00'	Get All Occurrences
'01'	Get Next Occurrences

CAT8: READ COUPONS Reference Control Parameter P2

[00318] In an example, a Read Coupons [Get Next Occurrences] command may be rejected if no prior Read Coupons [Get All Occurrences] is received within the current application session. Further, in response to the READ COUPONS command, multiple occurrences of a coupon data as may be defined in a customer specific coupon specification may be returned.

Tag	**Len**	**Value**
'E1'	Var	Customer-specific coupon data

CAT9: READ COUPONS data Field

[00319] Further, a successful execution of this command may be indicated by status bytes '9000'. The read coupons command may either return a general error condition, such as listed in the table CAT10.

SW1	SW2	Meaning
'63'	'10'	More data available
'67'	'00'	Wrong length
'69'	'85'	Conditions of use not satisfied
'6A'	'86'	Incorrect parameters P1-P2

CAT10: Status bytes for READ COUPONS

[00320] In an example, a SELECT Command may be used for selecting the coupon applet. The SELECT Command may be coded according to a table CAT11.

Code	Value
CLA	'00'
INS	'A4'
P1	'04'
P2	'00'
Lc	Length of AID
data	AID value
Le	'00'

CAT11: SELECT Command Message

[00321] The "data" field of the command may contain an application identifier (AID) of the coupon applet that may be selected, which may be a customized per customer requirements.

[00322] In response to this command, a returned message may be obtained. The returned message may include a data field that may include a file control interface (FCI) and may be coded according to a table CAT12.

Tag	Len	Value						Presence
'6F'	Var	FCI Template						M
		Tag	Len	Value			Presence	
		'84'	Var	DF Name (Applet AID)			M	
		'A5'	Var	Proprietary Data			M	
				Tag	Len	Value	Presence	
				'9F07'	2B	C-SAM Coupon Applet - Version Lx API ID	M	
				'9F09'	2B	C-SAM Coupon Applet - Version Lx API Version	M	
				'9F11'	2B	Coupon Format ID	M	
				'9F13'	2B	Coupon Format Version	M	

CAT12: SELECT Message Response data Field

[00323] The AID (Tag '84') of the service provider applet may be AID as described herein. The coupon applet API ID (Tag '9F07') may indicate the support for the coupon applet API. Further, the coupon applet API version (Tag '9F09') may be '0101' for this version. The coupon format ID may be a customer customizable coupon format ID and the coupon format version may be a customer customizable coupon format version . Further, additional two bytes tags may be listed in the table CAT12.

[00324] Further, a successful execution of the SELECT command may be indicated by status bytes '9000'. On occurrence of an error, the SELECT command may return a general error condition, such as listed in the table CAT1. In addition, the SELECT command may return one of the following specific errors and warning conditions as listed in the table CAT13.

SW1	SW2	Meaning
'67'	'00'	Wrong length
'6A'	'80'	Incorrect values in command data
'6A'	'82'	Applet not found
'6A'	'86'	Incorrect parameters P1-P2

CAT13: Status Bytes for SELECT

[00325] In an example, the SET COUPON REDEMPTION STATUS Command may be used for setting the coupon redemption status. On successful execution of this command, the applet status may change to "Transaction Completed". The SET COUPON REDEMPTION STATUS Command may be coded according to a table CAT14.

Code	**Value**
CLA	'C0'
INS	'41'
P1	'00'
P2	'00'
Lc	'0A'
data	Coupon Redemption Status
Le	'00'

CAT14: SET COUPON REDEMPTION STATUS Command Message

[00326] The execution of this command may not return any data. Further, a successful execution of the SET COUPON REDEMPTION STATUS command may be indicated by status bytes '9000'. On occurrence of an error, the SET COUPON REDEMPTION STATUS command may return a general error condition, such as listed in the table CAT1. In addition, the SET COUPON REDEMPTION STATUS command may return one of the following specific errors and warning conditions as listed in the table CAT15.

SW1	**SW2**	**Meaning**
'67'	'00'	Wrong length
'69'	'85'	Conditions of use not satisfied
'6A'	'86'	Incorrect parameters P1-P2

CAT15: Status bytes for SET COUPON REDEMPTION STATUS

[00327] In accordance with the exemplary and non-limiting embodiments, the present disclosure provides details of the interaction between the applet and the PoS terminal. FIG. 27 illustrates the interaction between the applet 2702 and the PoS terminal 2704.

[00328] At 2708, the PoS terminal 2704 may issue the SELECT command to select the coupon applet 2702. At 2710, the coupon applet 2702 may respond with the File Control Information (FCI). At 2712, the PoS terminal 2704 may issue the GET APPLET STATUS command. At 2714, the coupon applet 2702 may return the application status. At 2718, the PoS terminal 2704 may issue the READ COUPONS command with P2= Get All Occurrences on determining that the application status is "READY FOR COUPON REDEMPTION". At 2720, the coupon applet 2702 may return the coupon data for the available Coupons. If there are multiple coupons, it may include coupon data for multiple coupons in the Response data. If the applet may include all the coupon data, the applet may send an SW = '9000'. If the applet may not include the coupon data for all the coupons, the applet may send a warning SW = '6310', indicating that more data may exist. At 2722, the PoS terminal 2704 may parse the data and the SW. If the SW may include '6310', the PoS terminal may issue another READ COUPONS command with P2 = Get Next Occurrences.

[00329] On receiving the READ COUPONS command with P2 = Get Next Occurrences, at 2724, the coupon applet 2702 may attempt to send the coupon data for the remaining coupons. As with the READ COUPONS command with P2= Get All Occurrences, if there are multiple coupons pending to be sent, the coupon applet 2702 may include coupon data for multiple pending coupons in the response data as shown at 2724 in FIG. 27. Once the applet includes all the pending coupon data, the applet may send an SW = '9000'. If the applet does not include the coupon data for all the pending coupons, the applet may send a warning SW = '6310', indicating that more data may exist.

[00330] On receiving the SW='9000' in response to the READ COUPONS, at 2728, the PoS terminal 2704 may proceed to process the coupons. After processing the coupon data, at 2730, the PoS terminal 2704 may set the coupon Redemption Status by issuing a SET COUPON REDEMPTION STATUS command. The byte Array is always of a

length equal to the maximum number of coupons. The coupon Redemption Status may need to be set to valid coupon states for all the bytes - even if fewer coupons are present. If fewer coupons were received than the Maximum, the coupon Redemption Status Bytes corresponding to the ones that were not received may be set to indicate "No coupon Present". The coupon Redemption Status may be set in the order of receipt of coupon data in the responses to various calls to the READ COUPONS command. Even in the case when there are no coupons redeemed, the PoS terminal 2704 may be required to issue one of the following two commands: SET COUPON REDEMPTION STATUS or ABORT TRANSACTION.

[00331] FIG. 28 illustrates a state machine diagram of the coupon applet during the transaction flow according to an example. A state 2802 may indicate a READY FOR UPLOADING COUPONS state indicating that all coupons from the coupon applet may have been reset, and the wallet client can now upload the coupons selected for this transaction. In an example, as shown in FIG. 28, the applet may remain in the state 2802, if a command GET APPLET STATUS results in an error. The coupon applet may transit to a next state 2804 on receiving a valid ENABLE TRANSACTION command.

[00332] The 2804 state may include a READY FOR COUPON REDEMPTION state indicating that the wallet client may have attached all the coupons to the coupon applet's memory, and that the PoS terminal can proceed with the phase 2 of the transaction. In an example, as shown in FIG. 28, the coupon applet may remain in the state 2804, if a command READ COUPONS results in an error. The applet may transit to a next state 2808 on receiving a valid SET COUPON REDEMPTION STATUS command. As shown in FIG. 28, the coupon applet may transit to a next state 2810 on receiving a valid ABORT TRANSACTION command.

[00333] The state 2808 may include a TRANSACTION COMPLETED state indicating that the PoS terminal may have completed the phase 2 of the transaction and COUPON REDEMPTION STATUS may have been set. Further, this state 2808 may indicate that the wallet client may now proceed with phase 3 of the transaction. The coupon applet may remain in this state till the applet may receive a valid RESET COUPONS command or a valid ABORT TRANSACTION command.

[00334] The state 2810 may include a TRANSACTION ABORTED state indicating that the PoS terminal transaction (relating to the coupon presentation & redemption) may have been aborted either by the wallet client or the PoS terminal. The coupon applet may remain in this state 2810 till the applet may receive a valid RESET COUPONS command.

[00335] In an example, a table CAT16 may list the various applicable conditions to accept a command application data unit. In case of rejections, as shown in the table below, the coupon applet may return a status word SW='6985'.

Command	**Ready for Uploading Coupons**	**Ready for Coupon Redemption**	**Transaction Completed**	**Transaction Aborted**
SELECT	Accepted	Accepted	Accepted	Accepted
GET APPLET STATUS	Accepted	Accepted	Accepted	Accepted
ABORT TRANSACTION	Rejected	Accepted	Rejected	Rejected
READ COUPONS	Rejected	Accepted	Rejected	Rejected
SET COUPON REDEMPTION STATUS	Rejected	Accepted	Rejected	Rejected

CAT16: Command Acceptance Matrix

[00336] In accordance with exemplary and non-limiting embodiments, the present disclosure provides various data objects associated with the coupon applet. In an example, a data object including a coupon applet API ID may be disclosed. The coupon applet API ID may be encoded as a 2 byte structure. In order to determine whether the coupon applet API may be supported, the PoS terminal may perform a logical AND operation on the value with a mask '0102'. The result of the AND operation when equal to '0102', may indicate that coupon applet may be supported.

[00337] In an example, a data object including an applet API version may be disclosed. The coupon applet version may be encoded as a 2 byte structure. In an example, the most significant byte may represent a major release version and the least significant byte may represent the minor release version. In an example, a data object including an applet

status may be disclosed. The applet status may identify the current state of the applet and may be coded as 1 byte. In an example, a table CAT17 may list status values associated with the different statuses of the applet.

Status	Description
'01'	Ready For Uploading Coupons
'02'	Ready For Coupon Redemption
'40'	Transaction Completed
'80'	Transaction Aborted
'FF'	Unknown State

CAT17: Applet Status

[00338] In an example, a data object including a coupon redemption status may be disclosed. The coupon redemption status may be stored in one byte for every attached coupon. The coupon redemption status for the supported maximum number of coupons may be stored in a byte array of the length maximum number coupons. In an example, the redemption status for coupon 1 may be stored in the B1 of the byte array and the redemption status for coupon 2 is stored in the B2 of the byte array. A Table CAT 18 may list the various supported values of the coupon redemption status.

Coupon Status	Description
'00'	Unused
'01'	Redeemed
'05'	Rejected
'FF'	No coupon Present

CAT18: Coupon Redemption Status

[00339] In an example, the data object may include one or more coupons and the coupon format may depend on the type of implementation.

[00340] In accordance with the exemplary and non-limiting embodiments, the present disclosure provides various processing features of the PoS terminal. In an example, the

PoS terminal may be compliant with ISO/IEC 14443-4. While executing a transaction, the PoS terminal may analyze the data objects that may be returned by the applet. For example, the PoS terminal may check for a mandatory data object that may be returned by the applet. If the PoS terminal may not found the mandatory data object, the PoS terminal may abort the transaction. Similarly, if the PoS terminal may recombine incorrect formatting in the data, the PoS terminal may abort the transaction. The PoS terminal may take due care to ensure that the processing exceptions (e.g., incorrect formatting and the like) may be handled in a robust manner.

[00341] In an example, the PoS terminal may perform various processing operations while performing a transaction. The PoS terminal may format the SELECT command as discussed above and may verify the formatting of the FCI that may be included in the response message of the SELECT command. In an example, the PoS terminal may verify the formatting of the FCI in accordance with the table CAT12. If it may be detected that the FCI is not formatted in accordance with the table CAT 12, the PoS terminal may terminate processing and ensure de-selection of the applet.

[00342] Further, the PoS terminal may analyze the DF Name (tag '84') that may be returned in the FCI. If the DF Name is not same as the AID provided in the data field of the SELECT command, then the PoS terminal may terminate processing and ensure de-selection of the applet. In addition, the PoS terminal may verify that it may support the API by checking the API ID (tag '9F07') that may be returned in the FCI. As discussed above, the PoS terminal may perform a logical AND operation on the value with the mask '0102' to determine an availability of the support for the API. If the result of the AND operation is not equal to '0102', then the PoS terminal may terminate processing and ensure de-selection of the applet.

[00343] In an example, the PoS terminal may verify that it may support the API version (tag '9F09') that may be returned in the FCI. If it may not be determined, then the PoS terminal may terminate processing and ensure de-selection of the applet. In an example, the PoS terminal may verify that it may support the handling of the coupon format ID (tag '9F11') that may be returned in the FCI. If this may not be verified, then the PoS terminal may terminate processing and ensure de-selection of the applet. In an example, the PoS terminal may verify that it may support the handling of the coupon format

version (tag '9F13') that may be returned in the FCI. If it may not be verified, then the PoS terminal may terminate processing and ensure de-selection of the applet.

[00344] In an example, the PoS terminal may process the GET APPLET STATUS command. After successfully selecting the coupon applet, the PoS terminal may query the current applet status using GET APPLET STATUS. In an example, the PoS terminal may recognize that the applet Status may not be "READY FOR COUPON REDEMPTION". In such conditions, the PoS terminal may terminate processing and ensure de-selection of the applet. In an example, the PoS terminal may recognize that the applet status may be "READY FOR COUPON REDEMPTION". In such conditions, the PoS terminal may proceed with the normal processing of coupons.

[00345] In an example, the PoS terminal may process the READ COUPONS command. The PoS terminal may issue a READ COUPONS [Get All Occurrences] command to read the coupon data. A successful execution of the command may be indicated by status bytes '9000'. This may indicate that all coupon data may have been sent. Further, the READ COUPONS command may return a warning with SW = '6310' to indicate "More data available". On receipt of response with SW = '6310', the PoS terminal may issue a subsequent READ COUPONS [Get Next Occurrences] to retrieve the additional data. In an example, in a current application session, if the READ COUPONS [Get Next Occurrences] command may be issued before and without issuing the READ COUPONS [Get All Occurrences], the command may be rejected. Further, the PoS terminal may check to ensure that all mandatory data objects are present. In an example, the PoS terminal may encounter more than one occurrence of a Tag in a coupon object. In such conditions, the PoS terminal may reject the coupon.

[00346] In an example, the PoS terminal may process a SET COUPON REDEMPTION STATUS command. On validating the applied coupons, the PoS terminal may issue the SET COUPON REDEMPTION STATUS command to update the coupon redemption status on the applet. The coupon redemption status may be considered only for the coupons that may have been read since the last READ COUPONS [Get All Occurrences]. Further, if the READ COUPONS [Get All Occurrences] was not issued, all the coupon redemption status may be ignored. If all the coupons are not read since the

last call to READ COUPONS [Get All Occurrences], the coupon redemption status for the unread coupons may be ignored.

[00347] In an example, the PoS terminal may process an ABORT TRANSACTION command. The PoS terminal may encounter errors or states where the PoS may not be able to proceed with the processing. In such conditions, the PoS terminal may issue the ABORT TRANSACTION command to discontinue the transaction. On successful processing of the ABORT TRANSACTION command, the coupon redemption status may be lost. Further, on receiving the response to the ABORT TRANSACTION command, the PoS terminal may terminate processing and ensure de-selection of the applet.

[00348] High Level Architecture for a typical deployment –

[00349] FIG. 29 illustrates a high level architecture of a system 2900 for a deployment of the present invention in an exemplary embodiment.

[00350] A MTP 2902 may be set up in an ecosystem provider's domain 2904 and may be integrated into an enterprise service bus (ESB) 2908 of the domain 2904. The domain 2904 may include a wallet server 2918. The wallet server 2918 may include at least one wallet 2920 and the MTP 2902. The wallet 2920 may include at least one widget 2922. In an example, the ESB 2908 may connect the MTP 2902 an ecosystem of service providers 2910. The ESB 2908 may be used for designing and implementing the interaction and communication between mutually interacting software applications in service oriented architecture as illustrated in FIG.29. The ESB 2908 may be required for implementing an ecosystem with multiple wallet applications as illustrated in FIG. 29. The ESB 2908 may monitor and control the routing of a message exchange between various services for example, between the wallet server 2918 and a wallet provider 2924.

[00351] The MTP 2902 may have a plurality of roles. For example, the MTP may perform a wallet and transaction management along with all the relevant business services that may be associated with delivering wallets to a plurality of diverse mobile environments 2912. In an example, the MTP 2902 may provide an establishment of the ecosystem of the service providers 2910 and the third-party developers that build widgets for the wallet without having to worry about handset fragmentation. In an example, the MTP 2902 may establish a plurality of connectors with various ecosystems (for example,

the ecosystem of service providers 2910, plurality of mobile environments). The connectors along with the MTP 2902 may provide an improved user experience and a diverse set of services, which may be accessible to the end user irrespective of time or location.

[00352] The ecosystem provider's domain 2904 may include a trusted service manager (TSM) 2912. The TSM 2912 may be configured for an over the air provisioning and key management of widgets. The TSM 1812 may collaborate with the MTP 2902 to facilitate a near field communication experience. In an example, the wallet 2920 and the provisioning of the widget 2922 may be facilitated by the MTP 2902 (described by way of FIGS. 30-36). In an example, the setup and management of the wallet 2920 may be facilitated by the MTP102 (described by way of FIGS. 30-36). In an example, the applet issuance and personalization may be done by the TSM 2912 (described by way of FIGS. 30-36).

[00353] In an example, the service providers may use an MTP software development kit to build widgets that may run within a MTP wallet container. The widgets may provide both OTA value added services and proximity NFC services. The wallet container may provide the widgets 2922 a secure runtime environment and services to communicate OTA with the server (OTA services) and also communicate with the secure element to manage payment and non-payment applets for proximity NFC transactions. In an example, the widget 2922 may have an associated applet running in the secure element. In an example, the widget 2922 may not have an associated applet running in the secure element. In an exemplary embodiment, the service provider may provide only OTA services.

[00354] The ecosystem provider 2904 may ensure that all NFC readers 2930 at a merchant location 2914 may be equipped to handle the proximity protocols being used by the various applets that may be loaded into the secure element. The NFC readers 2930 may process the transaction through an acquiring network that may switch the transaction to the appropriate ecosystem service provider in the cloud. In an example, the wallet 2920 or the widget 2922 may communicate with an ecosystem cloud to determine service status and various other hygiene requirements like balance, transaction history, stored value top-up, etc.

[00355] The NFC readers 2930 may facilitate finalization of the secure element form factor, a micro SD, an embedded or UICC (as described in detail by way of FIG. 31). The NFC readers 2930 and allied infrastructure may facilitate finalization of devices. The NFC readers 2930 may facilitate payment protocol finalization. The NFC readers 2930 may facilitate reader distribution and support.

[00356] As illustrated in FIG. 29, the mwallet 2920 may interact with various elements from the system 2900. The mwallet 2920 may interact with the wallet providers 2924 and a plurality of widget providers 2928. In an example, the mwallet 2920 may interact with various infrastructure systems like support, message server and other service providers.

[00357] Each Ecosystem may include a plurality of ecosystem actors. In exemplary embodiments, the ecosystem actors may include near field communication (NFC) applets, an application store, a barcode reader, an ecosystem provider, an issuer, a message source, a mobile phone, a mobile wallet, a NFC point of sale terminal, an original equipment manufacturer (OEM) provider, a secure element (SE), a service providers (or widget provider), a telecom operator (telco), a trusted service manager (TSM), a TSM proxy client, an user, wallet administrator, a wallet companion applet (WCAp), a wallet provider, a wallet server, and a widget. In an example, the NFC applets on the secure element may perform transactions with the NFC point of sale readers. In an example, the application stores may refer to the application distribution mechanism that may be set up by device manufacturers such as apple®, blackberry®/rim® and android®/Google®. In an example, the barcode reader may be used to scan the barcode. In an example, the ecosystem aggregator entity that owns and sets up the mobile wallet ecosystem may ties up with the ecosystem operators or partners. The ecosystem aggregator entity may set up infrastructure and aggregate service providers to enable an mPayment and an mCommerce ecosystem. In an example, the issuer may be an entity that may issue cards, coupons, loyalty, gift and other tokens. In an example, the message source may be an entity that may create messages using wallet APIs for wallet users. In an example, the mobile phone may be the one on which the wallet runs. In an example, the mobile wallet may be downloadable client application for mobile handsets to provide users with a seamless payment experience along with a suite

of value added services for both proximity (NFC) and over the air transactions. In an example, the NFC point of sale terminal may interact with the NFC phone and performs a transaction. In an example, the OEM provider may provide devices, micro SDs, SE etc. In an example, the SE may represent the combination of the NFC with smart card technology for secure and trusted transactions, wherein the NFC may provide the RF front end for connectivity and smart card may provide security/crypto engine. In an example, the service providers may be the entity that creates wallet and offers payment and non-payment services to the wallet users. In an example, the telecom operator (Telco) may provide SMS and data services. In an example, the TSM may manage the OTA Issuance of Applet and the Card Data on the Secure Element and may provide Key Management Services. In an example, the TSM Proxy Client may sit on the mobile phone and interacts with the TSM server and the secure element. In an example, the user may be an end user of the mobile wallet. In an example, the Wallet Administrator may manage the backend operation of the wallet including the wallet lifecycle. In an example, the wallet companion applet may enable secure wallet & widget life cycle management that may include wallet & widget authentication, wallet state management and settings, and provides a ticket-based verification for secure authentication with the wallet server. In an example, the wallet provider may be an entity that launches branded wallet application and offers it to the user. In an example, the wallet server may be a robust, scalable, server platform that may manage the wallet ecosystem, the wallet lifecycle, the widget provisioning & integration with other ecosystem partners. In an example, the widget may be a set of business features bundled together and availed as a single unit. The widget may contain screen / UI, event handling, and corresponding business logic.

[00358] Use Cases – Overview

[00359] In various exemplary and non-limiting embodiments, the wallet use-cases may be categorized into various categories. These categories may include wallet download, wallet activation & set up, transaction & widget, life cycle activities, and administration & support. In an example, the wallet download may include use cases and methods for wallet distribution and installation of the wallet software on consumer's mobile phone. These services may be required for introducing the user to the platform and provisioning

the mobile software to the mobile phone. In an example, the wallet activation & set up category may include use cases for authenticating user credentials and establishing user profile and enabling user to use mobile wallet application. In an example, the transaction & widget category may include use cases to enable user to download widgets and get payment & non-payment tokens and transact using an activated mobile wallet application. The transaction use cases may include NFC transactions, barcode based transactions and over-the-air transactions. In an example, the life cycle activities may include use cases and features that may be required to enable users to continuing to use the services of the wallet application. In an example, the administration & support use cases may be related to managing day to day operational and support activities of the mWallet platform and includes use cases such as change device, lock and unlock device etc.

[00360] In various exemplary and non-limiting embodiments, the widget download may be the first use case scenario followed by the wallet activation & set up. The wallet activation & set up may further be followed by the transaction & widget use cases. The transaction & widget use cases may be followed by the life cycle activities and the administration & support use case category may support and assist the wallet download, wallet activation & set up, transaction & widget, life cycle activities categories. Few exemplary use cases may be described herein.

[00361] In an exemplary and non-limiting embodiment, the widget download may be in form of a SMS with Download URL (Push or Pull) use case. In an example, the SMS with Download URL (Push or Pull) use case may allow wallet provider to distribute application over the air using SMS push/pull without publishing the application on the application store. In this example, the mobile wallet may be successfully installed on the mobile phone and the user may see the mobile wallet icon on the main screen. The SMS with Download URL (Push or Pull) use case may use ecosystem actors like the user, the ecosystem provider, the mobile phone, and the telco. A precondition required for this use case may be the mobile phone to be data enabled.

[00362] FIG. 30 illustrates a flow chart describing a method 3000 of installing the mobile wallet on the mobile phone of the user using SMS with Download URL (Push or Pull). The method 3000 includes initiating the SMS by sending a SMS to a short code

at step 3002. In an example, the SMS may be sent by the user to the Telco or wallet provider, also referred to as an SMS push. In an alternative embodiment, the user may receive an SMS from telco or wallet provider, also referred to as an SMS pull. In an alternative embodiment, the user may initiate SMS from a Website. The completion of step 3002 may be dependent on successful initiation of the SMS short code (SMSC). The method 3000 may further include sending the SMS to the user at step 3004. In an example, the Telco may send the SMS to the user. In an alternative embodiment, the user may use integration with the SMSC. The completion of step 3004 may be dependent on the method of SMS integration, which may be with the Telco or the SMSC.

[00363] The method 3000 further includes receiving the URL with a welcome text by the user at step 3008. The user may select the URL provided in the SMS. The completion of step 3008 may be dependent on URL descriptor and welcome message. The method 3000 further includes the mobile phone sending the request to wallet server to provide the downloadable file at step 3010. The wallet server may be further configured to for providing the wallet software to the mobile phone at step 3012 of the method 3000. The mobile phone may be configured for downloading the application on the mobile phone and prompting user for installation of the application at step 3014 of the method 3000. The user may confirm the installation at step 3018 of the method 3000 after the user has been prompted by the mobile phone for installation. The mobile phone may install the application after receiving user confirmation at step 3020 of the method 3000.

[00364] In an exemplary and non limiting embodiment, a wallet activation use case process may enable the user to activate the mobile wallet and establish secure linkages between user & ecosystem actors such as mobile wallet, mobile phone, mobile phone, secure element, wallet server, TSM, and wallet provider. In an example, the wallet activation use case actors like mobile handset, TSM, the user, the mobile wallet, the wallet provider, the wallet server. The wallet activation may take place only when the user may have successfully installed the mobile wallet on the mobile handset (either preloaded or downloaded by the user as described by way of FIG. 30). Further, for wallet activation to take place, the SE may be connected/associated with the mobile handset (for removable SEs like micro SD and UICC). Further, the user may have the

data connectivity for wallet activation to take place. After successful activation of the wallet, the user may be able to use the wallet application.

[00365] FIG. 31 illustrates a method 3100 for wallet activation by the user. The method 3100 includes the mobile wallet showing a registration & activation screen to the user that includes an email address and a personal identification number (PIN) at step 3102. In an example, the PIN can be a 4 digit number. In an example, the PIN may be configured to be numeric and have business rules associated with it, e.g., 4-10 digits, no sequence, no repeating numbers and the like. In an example, the registration screen may be customized to capture additional parameters, as required to create a user profile, for example, username, a user photograph or any other user signature. The method 3100 further includes the user entering all required information on the registration and activation screen at step 3104. The method 3100 further includes the mobile wallet showing latest terms and conditions (T&C) to the user at step 3108. In an example, the T&C may be pulled at run time from server. In an example, the T&C may be different for each wallet provider. In an example, the acceptance of the T&C may be informed to the wallet provider along with a version number. The method 3100 may require the user to accept the T&C at step 3110. In an example, if the user may reject the T&C, then the user may be shown a message indicating that acceptance of T&C may be required to activate the wallet.

[00366] The method 3100 further includes the mobile wallet fetching a secure element ID (CIN), a device ID (IMEI), and a SIM ID (ICCID) from the mobile phone and sends the information (including E-mail ID, PIN and T&C version and a time stamp) to the wallet server at step 3112. In an example, not all information may be available to the mobile wallet in all cases and a separate work flow or interaction with Telco or service provider may be required to capture the required details. For example, if a mobile number may not be available from the SIM or device APIs, it may be captured from HTTP Header coming from the device if a Telco provides it. The method 3100 may require mobile number and SIM information from Telco before proceeding further in wallet activation. The method 3100 further includes the wallet server performing a first level verification to check if any of the information may be duplicate at step 3114. In an

example, if the user record may already be available, then the wallet server may provide appropriate error message to the user.

[00367] The method 3100 further includes the wallet server sending the information to wallet provider for verification at step 3118. At step 3120 the wallet provider may verify the information and send the verification results to the wallet server. In an example, the verification process may include a KYC check by the Telco. The verification process may be required by the method 3100 before proceeding further in wallet activation. At step 3122, the wallet server may create a wallet record with a user status as 'INACTIVE' and send a confirmation to the mobile wallet. At step 3124, the mobile wallet may show a message to the user that wallet may be in process of being personalized for the user. At step 3128, the wallet server may send a request to TSM through ESB with user information (the mobile number, the SE ID, the device ID, the WALLET ID, and the PIN as discussed above).

[00368] The method 3100 further includes the ESB configured for sending the information to TSM at step 3130. At step 3132, the TSM may communicate with the TSM proxy client and initiates the SE personalization process. At step 3134, the TSM may personalize the secure element with wallet companion applet installation and may perform store data on it with the hardware and wallet related information. At step 3138, the TSM may send a notification to the wallet server once the personalization is complete. At step 3140, the wallet server may update the wallet status as "active" and notifies the wallet client. At step 3142 of the method, the mobile wallet may show a confirmation to the user indicating successful wallet activation.

[00369] In an exemplary and non limiting embodiment, the transaction and widget use cases may include a series of widget management use cases. In an example, the widget management use cases may include a series of widget installation uses cases. In an example, the widget installation uses cases may include an in-application use case alert. The in-application use case alert may allow the users to download widgets based on the alerts received by the wallet provider. The wallet provider may send the alerts using C-SAM's alert framework. The in-application use case alert may require the mobile wallet, the service provider, the user, the wallet provider, and the wallet server. The user may install widget as an in-application case after activation of the mobile wallet. Also, the

user may be logged in to the mobile wallet. The service should have created a widget before in-application use case alert. The wallet provider may have reviewed the widget and published it before the in-application use case alert. The wallet provider may have created an alert before the in-application use case alert. After the in-application alert, the user may have successfully downloaded the widget. In the in-application use case alert, the mobile wallet may show an alert to the user. In an example, the alert may be shown to the user in the mobile wallet message inbox or during communication with wallet server. The user may then, selects alert, view the message and initiate a download. After initiation of download, the mobile wallet may install the widget and send confirmation to the wallet server. The wallet server may send confirmation to the mobile wallet. The mobile wallet may show the confirmation to the user.

[00370] In an example, the widget installation uses cases may include a search and download use case. The search and download use case may allow the user to search the widgets and download and install on the mobile wallet. The search and download use case may require the mobile wallet, the wallet server, the service provider, the user and the wallet provider. The search and download use case may require the user to have activated the mobile wallet, user to have logged in to the mobile wallet, service provider to have created a widget, and wallet provider to have reviewed the widget and published it. The user may successfully download the widget as a result of the search and download use case.

[00371] FIG. 32 illustrates a method 3200 as an example of the search and download use case. The method 3200 may include the user to search or browse categories of available widgets at step 3202. In an example, the user may be able to search the widget by selecting widget from a service list or entering or selecting appropriate criteria in the wallet. At step 3204, the method 3200 may require the mobile wallet to send a request based on user criteria to the wallet server. The wallet server may send the mobile wallet a list of all available widget matching search criteria of the user at step 3208. The mobile wallet may show the list of the widgets to the user at step 3210. The user may view the widget description of any widget from the list sent by the wallet server and initiate a download of the desired widget at step 3212. The mobile wallet may install the widget and send a confirmation to the wallet server at step 3214 of the method 3200. The wallet

server may send a confirmation to the mobile wallet at step 3218. The mobile wallet may show the confirmation to the user at step 3220.

[00372] In some exemplary and non-limiting embodiments, the transaction use cases may include an over-the-air (OTA)-transaction use case. The over-the-air transaction use case may describe a process for the OTA-transaction based transaction. The OTA-transaction may include the user, the mobile wallet, the widget, and the service provider. For the OTA-transaction to take place, the user may have to be logged into the mobile wallet, the user may have to have data connectivity, and the user may have loaded the widget for OTA transaction on the mobile wallet. In an example, the OTA-transaction may facilitate in successfully completing a transaction by the user

[00373] FIG. 33 illustrates an exemplary method 3300 for the OTA-transaction. The method 3300 may include the user to select a widget at step 3302. The widget selection may be done by various types of services that may be available to the user, for example, ticketing, bill payment and the like. At step 3304 the method 3300 may require the mobile wallet to load the widget at step 3304 onto the service screens (as also discussed above). The user may browse the service screens and selects product/service to pay at step 3308. The workflow and user experience may differ for different services. A service may require multiple interaction before the user may reach a payment screen. The wallet server may send the information from the user to the service provider at step 3310. The service provider may provide a payment request to the user with a payment amount at step 3312. The user may select a payment card and provides the payment info at step 3314. In an example, user may select a payment method available on wallet and that may allow the OTA transaction. In an example, the user may enter card information while air-travelling as well. The step 3318 of the method 3300 may require the wallet server to provide the payment information to the service provider. At step 3320 the service provider may settle the payment through a payment gateway/banking interface and may provide a transaction confirmation to the wallet server. At step 3322, the wallet server may receive the transaction confirmation and may send the information to the widget and the wallet. At step 3324 of the method 3300, the mobile wallet may show the transaction confirmation and details to the user.

[00374] In exemplary and non-limiting embodiments, the life cycle activities may include a card issuance use case. The card issuance use case may allow a card issuer to issue a card data on the wallet. The card issuance use case may include the issuer, the wallet server, the TSM, the user, and the mobile wallet. For the card issuance to take place, the user may have to activate the mobile wallet, the user may have to log in to the mobile wallet, the service provider may have to create an updated widget, and the wallet provider may have to review the widget and publish it. After, the card issuance is completed successfully; the user may be able to download the widget.

[00375] FIG. 34 illustrates a method 3400 as an example for the card issuance use case. The method 3400 may require the issuer to approve a payment account (for example, a credit card) at step 3402. The issuer may personalize the secure element with the help of the TSM at step 3404. In an example, the TSM may install the payment applet and may personalize with the track data. The card issuance use case may depend upon the payment card and the applet issuance on the SE & the type of applet. In an example, the applet may be a Pay-pass, a zip etc. At step 3408, the method 3400 may require the user to provide soft card data to the wallet server via ESB. The soft card data may contain images, balance, card nick name, and AID to be used for transaction. The method 3400 may further require the wallet server to create a record for a service account for the user at step US610. The wallet server may identify a widget to be sent along with the service account at step 3412. The wallet server may create a hash on the widget and send the Widget ID, the version and the hash to TSM via ESB at step 3414. The TSM may complete the widget hash update on the secure element with the help from the TSM proxy client at step 3418. The TSM may notify wallet server via ESB once the widget hash may be updated at step 3420. The wallet server may make the soft card and widget available for download at step 3422. The mobile wallet may perform synchronization and may download the soft card and the widget on the mobile phone of the user.

[00376] In exemplary and non-limiting embodiments, the support and administration may include a terminate wallet use case. The terminate wallet use case may allow a wallet administrator to terminate a wallet account. The terminate wallet use case may include a mobile wallet, a wallet server, a TSM, a TSM proxy client. For the wallet to terminate, the wallet administrator may have to be registered, the wallet administrator

may have to have privileges to manage a user profile, and the wallet administrator may have to a wallet management console. The wallet administrator may successfully terminate a wallet account after completion of wallet termination use case.

[00377] FIG. 35 illustrates a method 3500 as an example to terminate a wallet. The method 3500 may require the wallet administrator to initiates a suspend wallet activity at step 3502. The wallet server may update the wallet state as "termination_pending" at step 3504 after the suspension activity may be initiated. The wallet server may send a request to ESB at step 3508. The ESB may retrieve a list of service accounts from the wallet server at step 3510. The ESB may send request to the issuers to terminate the service accounts at step 3512. The ESB may check the list of service accounts before sending request to the issuers. The issuer may terminate the wallet account in the backend and also remove the payment applets from SE, with the help of TSM at step 3514. The ESB may send a request to TSM to remove the wallet companion applet at step 3518. The TSM may send a message to the TSM proxy client at step 3520. The TSM proxy client may remove the wallet companion applet from SE at step 3522. The TSM may send a notification to ESB once the update may be done at step 3524. The wallet server may receive the update and may change the wallet status to "terminated" at step 3528 of the method 3500. The wallet may be successfully terminated after step 3528.

[00378] In exemplary and non-limiting embodiments, the widget management includes a publish widget use case. The publish widget use case may allow the ecosystem provider to publish an approved widget. The publish widget use case may include the wallet administrator, the wallet server, and the service provider. Before publishing a widget, the wallet administrator may have to be registered, the service provider may have to be registered, the wallet administrator may have to be logged into the wallet management console, and the widget may have to be validated and approved.

[00379] FIG. 36 illustrates a method 3600 as an example for publishing a widget. The method 3600 may require the wallet administrator to log into the wallet server system management console at step 3602. The wallet administrator may select widget management from the wallet server system management console at step 3604. The wallet administrator may select a link for publishing widgets from the widget management at

step 3608. The wallet server may display all widgets in the "approved" state at step 3610. The wallet administrator may select a widget to publish at step 3612. The wallet administrator may enters any comments or remarks and select publish one the publishing widgets of the wallet server system management console at step 3614. The wallet server may prompt the wallet administrator for a confirmation at step 3618. The wallet administrator may confirm that the widget may be published at step 3620. The wallet server may change the state of the widget to "published" at step 3622. The wallet server may generate a signature for the widget at step 3624. The wallet server may confirm that the widget may have been published at step 3628 of the method 3600.

[00380] Behavioral Model – Add Widget - FIG. 37 is a method 3700 for adding a widget for the first time along with a TSM interaction. The method 3700 may require a user to use a wallet 3702 for selecting a widget 3704 at step 3708. At step 3710, a widget script component 3712. The script component 3712 may communicate with a widget manager 3714 so as to retrieve a latest version of the widget 3704 at step 3718 of the method 3700. In an example, the script component 3712 may communicate with a widget manager 3714 so as to retrieve a widget descriptor. In an example, the script component 3712 may communicate with a widget manager 3714 so as to retrieve the widget binaries. The widget manager 3714 may send widget descriptor data blocks, hash of widget bundle, MDN, and widget instance ID to a trusted service manager (TSM) 3720 at step 3718 of the method 3700. The widget manager 3714 may generate widget response and communicate to the wallet 3702. The wallet 3702 may include a database for storing widget binaries and widget response from the widget manager 3714. The TSM 3720 may add the widget 3704 to a secure element 3722 at step 3728. In an example, the secure element 3722 may be a wallet applet.

[00381] FIG.38 is a behavioral model of a system 3800 for widget access verification with reference to the widget loading time signature, verification, and execution time access verification.

[00382] In an exemplary and non-limiting embodiment, a wallet 1002 may be configured so that a user may select a widget from a list of widgets. The wallet 1002 may be configured so that wallet 1002 may start the widget application selected by the user. The wallet 1002 may interact with a widget manager 3804. The widget manager

1004 may be configured to initialize resource managers, controllers, create default access context and the like. In an example, the widget manager 3804 may include the list of widgets and may facilitate showing of the widget lists to the user through the wallet 3802. The system 3800 may include a cryto-utility 3808 that may be configured to load the widget bundle from database. The cryto-utility 3808 may create a hash, for example a secure hash algorithm 1 (SHA1) hash from the widget bundle. The cryto-utility 3808 may communicate the hash and widget bundle to the widget manager 3804. The system 3800 may include a NFC chip utility 3810 and a secure element or wallet applet 3812. The cryto-utility 3808 may initiate a request to verify the widget bundle and send the request to the NFC chip utility 3810 and the wallet applet 3812. The NFC utility 3810 may facilitate selection of the 3812. The applet 3812 may send the verification request to a widget access context 3814. The widget access context 3814 may verify the widget bundles, the widget descriptor. In an example, the widget manager 3804 may initiate a communication with the widget access context 3814. The widget access context 3814 may be in communication with an access action controller 1018. The access action controller 1018 may check permission and access for permissions of the selected widget. In an example, an appropriate message may be sent to the user when the verification may be passed or failed. The system 3800 may include an action controller 3820 may execute the widget requests after permission verification. An access aware screen controller 3822 may be included in the system 3800, and configured to show a screen if the permission access may be received from the access action controller 1018. The system 3800 may include a screen controller 3824. The screen controller 3824 may be configured to display a screen on the wallet 3802 after checking display permission from the widget access context 3814.

[00383] The methods and systems of mobile transaction platforms in a multi-domain eco system for secure personalized transactions may provide for a method of protocol abstraction for a mobile transaction platform in a dynamic ecosystem for secure commerce and payment transactions. The mobile transaction platform may be a carrier-grade infrastructure platform that facilitates the development and delivery of secure transactional services, both Over-the-Air and through NFC Proximity, to a broad range of mobile phone environments. The method may include taking the information elements of

the original method apart and mapping the information elements of the original method to new fields of information of a common format. These new fields of information may be empty, filled, or translated by tokenization or some other method of translation. The common format may work for different input protocols and deliver a single output protocol. The common format may be a meta-model that is consistent with the data structures of each protocol. The common model may be a services abstraction model that can handle the native sequence flows within an underlying protocol. The input protocols may be point-of-sale protocols, NFC protocols, credit card protocols, debit card protocols, ticketing protocols, EPSON Esc/POS protocol, UTC Standard protocol, AEDEX protocol, ICD 2002 protocol, Ultimate protocol, CD 5220 protocol, DSP-800 protocol, ADM 787/788 protocol, HP protocol, UnifiedPOS protocol, OLE for Point-of-Sale protocol, JavaPOS, cloud-based protocol, and the like.

[00384] The method of protocol abstraction may be performed between a mobile device and a point-of-sale device via a communication link. The mobile device may be a smartphone, tablet computer, laptop computer, or the like. The mobile device may be running an operating system such as iOS, Android, Windows, or the like. The mobile device may be running an application such as a container, widget, wallet, or the like. The application may store information related to soft cards. The soft cards may be credit cards, loyalty cards, gift cards, store value cards, ticket cards, and the like. The soft cards may be issued from multiple service providers from multiple ecosystems through multiple carrier or network providers. The soft cards may be provisioned, maintained, and enabled through and by a provisioning architecture. The provisioning architecture may include a mobile enabling tier, a service tier, and a software development kit. The point-of-sale device could be a real world point-of-sale device operating in a proximity environment or a virtual world point-of-sale device operating in a remote environment. The communication link may be an wired communication link, a wireless communication link, an NFC link, a Bluetooth link, a local area network link, a wide area network link or the like.

[00385] The method of protocol abstraction may include a tokenization capability for a mobile transaction platform in a dynamic ecosystem for secure commerce and payment transactions. The mobile transaction platform may be a carrier-grade infrastructure

platform that facilitates the development and delivery of secure transactional services, both Over-the-Air and through NFC Proximity, to a broad range of mobile phone environments. The tokenization capability may take private information and replace it with a token. The private information may include a pin number, a credit card number, primary account number, other personally identifiable information, and the like. The token may consistent of numeric characters, alphabetic characters, a combination of alphabetical and numeric characters, a truncated primary account number with alphabetic and numeric characters replacing the middle digits of the primary account number, and the like. The token may be a one-time use token. The token may be issued by a trusted entity in the dynamic ecosystem for secure commerce and payment transactions. The trusted entity may be a server located at the payment gateway, acquirer, or issuer. The token may be generated as a mathematically reversible cryptographic function, a one-way non-reversible cryptographic function, or assigned through an index function, sequence number or a randomly generated number.

[00386] With reference to FIG. 39, there is illustrated a dynamic ecosystem for secure commerce and payment transactions according to exemplary and non-limiting embodiments. A dynamic ecosystem for secure commerce and payment transactions 3902, include a point-of-sale terminal 3908, an application running on a mobile device 3910, a communication link between the point-of-sale device 3908 and the mobile application running on a mobile device 3910. A single protocol 3912 associated with a single domain 3914 may connect the point-of-sale device 3908 and the mobile application running on a mobile device 3910.

[00387] With continued reference to FIG. 39, a dynamic ecosystem for secure commerce and payment transactions 3904, a multiple protocols 3916 are associated with multiple domains 3918 may connect the point-of-sale device 3908 and the mobile application running on a mobile device 3910.

[00388] With continued reference to FIG. 39, a dynamic ecosystem for secure commerce and payment transactions 3906, a multiple protocols 3916 are associated with multiple domains 3918 may be abstracted using a wrapper 3920, as described previously in this specification, to connect the point-of-sale device 3908 and the mobile application running on a mobile device 3910.

[00389] The methods and systems described herein of multi-domain ecosystem secure personalized transactions includes a method of abstracting a protocol in a secure personalized transaction through a multi-domain ecosystem. The method includes receiving an electronic transaction message in a secure personalized transaction comprising a plurality of information fields at a networked device in the multi-domain ecosystem. The method may further include determining with the device a type and a content for each of the plurality of information fields. The method may further include mapping at least a portion of the plurality of information fields to common format message fields. The method may further include configuring a common format transaction message based on the mapping. The method may further include transmitting the common format transaction message from the networked device to a second networked device in the multi-domain ecosystem.

[00390] The methods and systems described herein of multi-domain ecosystem secure personalized transactions includes a method of abstracting a protocol in a secure personalized transaction through a multi-domain ecosystem. The method includes receiving an electronic transaction message in a secure personalized transaction comprising a plurality of information fields at a networked device in the multi-domain ecosystem. The method may further includedetermining with the device a type and a content for each of the plurality of information fields. The method may further include mapping at least a portion of the plurality of information fields with a meta-model that provides consistent mapping of the secure personalized transaction protocol to common format message fields. The method may further include configuring a common format transaction message based on the mapping. The method may further include transmitting the common format transaction message from the networked device to a second networked device in the multi-domain ecosystem.

[00391] The methods and systems described herein of multi-domain ecosystem secure personalized transactions includes a method of abstracting a protocol in a secure personalized transaction through a multi-domain ecosystem. The method includes receiving an electronic transaction message in a secure personalized transaction comprising a plurality of information fields at a networked device in the multi-domain ecosystem. The method may further includedetermining with the device a type and a

content for each of the plurality of information fields. The method may further include mapping at least a portion of the plurality of information fields to common format message fields. The method may further include configuring a common format transaction message based on the mapping. The method may further include wrapping the common format transaction in a protocol abstraction content wrapper. The method may further include transmitting the wrapped common format transaction message from the networked device to a second networked device in the multi-domain ecosystem.

[00392] With reference to FIG. 40, there is illustrated a method of protocol abstraction in a dynamic ecosystem 4002 for secure commerce and payment transactions according to exemplary and non-limiting embodiments. A point-of-sale device 4004 may connect to a mobile wallet 4006 running on a mobile device 4008. Multiple mobile wallets may be running on the mobile device. The point-of-sale device 4004 may communicate with the mobile wallet 4006 running on the mobile device 4008 over a communication link 4010. The communication link 4010 may be a wired communication link, a wireless communication link, an NFC link, a Bluetooth link, a local area network link, a wide area network link or the like.

[00393] With continued reference to FIG. 40, the mobile wallet 4006 running on a mobile device 4008 may communicate with service provider 4012 via a protocol through a network or wireless provider 4014 and a network 4022. The network 4022 may be an Internet Protocol based network. A wrapped acquisition channel 4016 may be established to facilitate protocol abstracted communication among the point-of-sale device 4004, the mobile wallet 4006 running on a mobile device 4008, and an acquirer facility 4018. The wrapped acquisition channel 4016 that may wrap the protocol used for communication as described previously in this specification may facilitate exchanging electronic transaction information via a standardized communication protocol between the point-of-sale device 4004 and the mobile wallet 4006 running on a mobile device 4008 that is seamless to a user of the mobile device 4008. A user of the mobile device 4008 may operate the mobile device at a point-of-sale without necessarily being aware of the type of protocol being used during the POS transaction. In this way, legacy POS devices that only support a non-IP based transaction flow and POS devices that are capable of supporting an aggregated protocol transaction flow (e.g. through an IP network) may be seamlessly

operated by the user. The mobile wallet 4006 running on a mobile device 4008 may communicate with service provider 4012 via a protocol through an acquirer 4018 and a network 4016. The network 4016 may be an Internet Protocol based network. The point-of-sale device 4004 may communicate via an existing direct transaction channel 4020. The existing direct transaction channel may be a non-Internet Protocol based channel 4020.

[00394] The methods and systems described herein of multi-domain ecosystem secure personalized transactions include a method of a secure personalized transaction through a multi-domain ecosystem. The method includes receiving an electronic transaction message in a secure personalized transaction comprising a plurality of non-confidential information fields at a networked device in the multi-domain ecosystem. The method may further include configuring a token comprising the non-confidential information and transaction identification information at the networked device. The method may further include electronically transmitting the token to a source of the electronic transaction message and to a service provider server disposed in the multi-domain ecosystem. The method may further include transmitting authentication of the token, along with token identification information from the source of the electronic transaction message to the service provider server for authenticating the transaction identified by the token. In an example, the service provider may be an instrument issuer. In an example, the service provider may access confidential information that is not accessible to the networked device to facilitate authenticating the transaction identified by the token. The confidential information may be associated with a digital wallet disposed on a device from which the authentication of the token is transmitted. In an example, transmitting authentication of the token may be facilitated by a provider-specific widget executing on a device from which the electronic transaction message is sourced.

[00395] With reference to FIG. 41, there is illustrated a method of tokenization of transaction content in a dynamic ecosystem 4102 for secure commerce and payment transactions according to exemplary and non-limiting embodiments.

[00396] A point of sale device 4104 may be activated by near-by presence of a mobile phone that incorporates a mobile wallet 4106 requesting to perform an electronic transaction such as a purchase. The POS device 4104 and mobile wallet 4106 may

exchange non-personal information that describes the requested transaction (item, cost, date/time, POS identity, POS service provider identity, and the like) along with a unique identifier generated by the mobile wallet 4106. The POS 4106 may issue a token that incorporates this information in a secure encoding to the mobile wallet 4106 and may also submit a version of the issued token to a payment authority, such as an issuing bank 4112 that may have been specified by the mobile wallet 4106.

[00397] The mobile wallet 4106 and the issuing bank 4112 may subsequently exchange information related to the transaction as described in the token to complete the transaction. In the example, the mobile wallet may submit a query to the issuing bank that includes a reference to the token (e.g. a token ID) that may signal to the issuing bank that the transaction identified in the token is authorized. Alternatively, the issuing bank may submit a request for authorization of the transaction referenced in the token to the mobile wallet. The transaction information of the token received from the POS terminal 4104 and from the issuing bank 4112 may be compared and presented to the user of the mobile device 4108 for authentication. In this way the transaction may be executed without requiring the POS 4104 or other entities in the POS communication path to have access to any user confidential, personal, or financial information because all confidential, personal, and financial information of the user/mobile wallet may be retained on the backend server (e.g. the issuing bank 4112) and may be accessed indirectly through the token that is exchanged among the mobile wallet 4106, the POS terminal 4104, the acquisition channel 4120, and the payment facilitator, such as the issuing bank 4112.

[00398] A point-of-sale device 4104 may connect to a mobile wallet 4106 running on a mobile device 4108. Multiple mobile wallets may be running on the mobile device. The point-of-sale device 4104 may communicate with the mobile wallet 4106 running on the mobile device 4108 over a communication link 4110 or based on visual information exchange (e.g. a QR code or bar code, and the like). The communication link 4110 may be a wired communication link, a wireless communication link, an NFC link, a Bluetooth link, a local area network link, a wide area network link or the like. The visual information exchange may include receiving an image with a camera and/or displaying an image on an electronic display.

[00399] With continued reference to FIG. 41, the point-of-sale device 4104 may receive private information from the mobile wallet 4106 running on the mobile device 4108, store the private information, then send the private information to an issuing bank 4112 for authorization. The issuing bank may receive the private information directly through an over the air interface 4114, , or through a service provider 4116 and a network 4118. Upon authorization, the issuing bank 4112 may replace the private information with a token. The token may consistent of numeric characters, alphabetic characters, a combination of alphabetical and numeric characters, a truncated primary account number with alphabetic and numeric characters replacing the middle digits of the primary account number, and the like. The token may be a one-time use token. The token may be issued by a trusted entity in the dynamic ecosystem for secure commerce and payment transactions. The trusted entity may be a server located at the payment gateway, acquirer, or issuer. The token may be generated as a mathematically reversible cryptographic function, a one-way non-reversible cryptographic function, or assigned through an index function, sequence number or a randomly generated number. The issuing bank 4112 may then return the token to the point-of-sale device 4102. The point-of-sale device 4102 may then replace the private information with the token and store the token. The token may be contained in a wrapper, as described previously in this specification, and communicated among the mobile wallet 4106 running on a mobile device 4108, point-of-sale device 4104, and acquiring bank 4112, through a wrapped acquisition channel 4120.

[00400] With continued reference to FIG. 41, the mobile wallet 4106 running on the mobile device 4108, may establish a secure channel with the issuing bank 4112. Upon authorization, the issuing bank 4112 may generate a token and map it to the private information stored at the issuing bank 4112. The token will then be sent to the mobile wallet 4106 running on the mobile device 4108 and the mobile wallet 4106 running on the mobile device 4108 will use the token to complete the transaction at the point-of-sale device 4104. The mobile wallet 4106 running on the mobile device 4108, may generate and store a stack of tokens upon personalization of the mobile wallet 4106 and use these tokens to complete the transaction at the point-of-sale device 4104. The mobile wallet

4106 running on the mobile device 4108, may generate, at the time of the transaction, and use a token to complete the transaction at the point-of-sale device 4104.

[00401] With reference to FIG. 42, there is illustrated a method of dynamic tokenization of transaction content in a dynamic ecosystem 4202 for secure commerce and payment transactions according to exemplary and non-limiting embodiments. A point-of-sale device 4204 may connect to a mobile wallet 4206 running on a mobile device 4208. Multiple mobile wallets may be running on the mobile device. The point-of-sale device 4204 may communicate with the mobile wallet 4206 running on the mobile device C408 over a communication link 4210. The communication link 4210 may be a wired communication link, a wireless communication link, an NFC link, a Bluetooth link, a local area network link, a wide area network link or the like. Visual information exchange may include receiving an image with a camera and/or displaying an image on an electronic display.

[00402] With continued reference to FIG. 42, the point-of-sale device 4204 may receive a dynamic token from the mobile wallet 4206 running on the mobile device 4208. The dynamic token may be one of a dynamic token generated by the mobile wallet 4106 running on the mobile device 4108 upon personalization of the mobile wallet 4106, at the time of the transaction, or the like. The point-of-sale device 4204 may then send the dynamic token to the issuing bank 4212 for validation. The issuing bank 4212 may then validate the token via an over-the-air interface 4214 by validating the token and a mobile link ID known by the issuing bank 4212 to be associated with the mobile device 4208 on which the mobile wallet 4206 that generated the token resides. Upon validation of the token and mobile link ID, the issuing bank 4212 then authorized payment on the point-of-sale device 4204 and settlement on the mobile wallet 4206.

[00403] The methods and systems described herein of multi-domain ecosystem secure personalized transactions includes a method of token mobile caching that includes generating with a networked server a token for performing an electronic transaction over a public network with a client device. The method may further include providing the token to the client device over the public network. The method may further include storing the token along with information that uniquely identifies the client to which the token was provided in a memory accessible to the server. The method may further

include comparing the stored information that uniquely identifies the client to which the token was provided with client identification information received with the token to determine if conducting the electronic transaction should be authenticated. The comparing may be performed in response to receiving an electronic message from a client device at the server to conduct an electronic transaction using the token over the public network. In an example, the electronic message may include information derived from a secure element associated with the client to which the token was provided.

[00404] To enhance mobile transaction security, tokenization as described herein may facilitate reducing confidential information exchange among devices. However, even a token may be captured by a third party and attempted to be used for an unauthorized transaction. Therefore, caching of mobile information related to tokenization may ensure that a token is received from a mobile device to which it was delivered. This may be accomplished by sharing token and mobile identification information over a secure acquisition channel between an issuer of the token (e.g. an issuing bank) and a POS / service provider so that when a mobile device provides the token, if the mobile device identification information does not match the token, the transaction can be denied. Likewise, caching of mobile-related information associated with a token may be retained at a payment facilitator such as an issuing bank. This information may be compared to a tokenized acquisition transaction from a POS device. In an example, a mobile device and a bank may exchange information over an IP network so that a token is provided to the mobile device for conducting one or more POS transactions. The mobile device may submit a token to a POS device along with mobile device identification information. This set of information may be optionally further tokenized and/or sent as a set to an issuing bank (e.g. over a secure acquisition channel). The issuing bank may compare the cached mobile related information associated with the token received from the POS device to the mobile device information received from the POS and only authorize the transaction if the information favorably compares.

[00405] The methods and systems described herein of multi-domain ecosystem secure personalized transactions include a method of tokenizing a wrapper in a secure personalized transaction through a multi-domain ecosystem. The method may include receiving an electronic transaction message in a secure personalized transaction

comprising a plurality of information fields at a networked device in the multi-domain ecosystem. The method may further includedetermining with the device a type and a content for each of the plurality of information fields. The method may further include mapping at least a portion of the plurality of information fields to common format message fields. The method may further include configuring a common format transaction message based on the mapping. The method may further include wrapping the common format transaction in a protocol abstraction content wrapper. The method may further include configuring a token comprising the protocol abstraction content wrapper. The method may further include transmitting the wrapped common format transaction message from the networked device to a second networked device in the multi-domain ecosystem.

[00406] With reference to FIG. 43, there is illustrated a method of generating a token for an electronic transaction wrapper for secure content traversing a network in a dynamic ecosystem for secure commerce and payment transactions according to exemplary and non-limiting embodiments. A protocol 4302 for a particular ecosystem may be used to store and communicate protocol private information elements 4304 within a dynamic ecosystem for secure commerce and payment transactions. Protocol private information elements may include a credit card expiration date, primary account number (PAN), card holder's name, card holder's address, and the like. The protocol 4302 may organize the protocol private information elements 4304 in a specific order or configuration. For example the expiration date may be held in a first position, the primary account number in the second position, the card holder's name in a third position and the card holder's address in a fourth position. The protocol 4302 may store protocol private information elements 4304 in a particular configuration. For example, a protocol 4302 may store the credit card expiration date in a MM/YYYY format.

[00407] With continued reference to FIG. 43, a wrapper 4306 may be used to store and communicate wrapper private information elements 4308 for multiple ecosystems among dynamic ecosystems for secure commerce and payment transactions. Wrapper private information elements may include a credit card expiration date, primary account number (PAN), card holder's name, card holder's address, account balance, credit limit, and the like. The wrapper 4306 may organize the wrapper private information elements 4308 in

a specific order or configuration. For example the primary account number may be held in a first position, the expiration date in the second position, the card holder's name in a third position, the card holder's address in a fourth position, the account balance in a fifth position, and the credit limit in a sixth position. The wrapper 4302 may store protocol private information elements 4306 in a particular configuration. For example, a wrapper 4306 may store the credit card expiration date in a YYYY/MM format.

[00408] With continued reference to FIG. 43, a translation process 4310 may be used to map protocol private information elements 4304 to wrapper private information elements 4308. For example the translation process 4314 may take the expiration date protocol private information element from the first position of the protocol 4302 and place it in the second position of the wrapper 4306 and translate the configuration of the private information element, MM/YYYY, to match the configuration of the wrapper information element YYYY/MM.

[00409] With continued reference to FIG. 43, a tokenization process 4314, may be used to replace the wrapper private information elements with a token 4312. The token 4312 may consistent of numeric characters, alphabetic characters, a combination of alphabetical and numeric characters, a truncated primary account number with alphabetic and numeric characters replacing the middle digits of the primary account number, and the like. The token 4312 may be a one-time use token 4312. The token 4312 may be issued by a trusted entity in the dynamic ecosystem for secure commerce and payment transactions. The trusted entity may be a server located at the payment gateway, acquirer, or issuer. The token 4312 may be generated as a mathematically reversible cryptographic function, a one-way non-reversible cryptographic function, or assigned through an index function, sequence number or a randomly generated number.

[00410] The methods and systems described herein of multi-domain ecosystem secure personalized transactions include a method of tokenizing a wrapper in a secure personalized transaction through a multi-domain ecosystem. The method may include receiving a wrapper for an electronic transaction message in a secure personalized transaction at a networked device in the multi-domain ecosystem. The method may further include configuring a token comprising the wrapper and transaction identification information at the networked device. The method may further include electronically

transmitting the token to a source of the wrapper for an electronic transaction message and to a service provider server disposed in the multi-domain ecosystem. The method may further include transmitting authentication of the token, along with token identification information from the source of the wrapper for an electronic transaction message to the service provider server for authenticating the transaction identified by the token. In an example, the service provider may be an instrument issuer. In an example, the service provider may access confidential information that is not accessible to the networked device to facilitate authenticating the transaction identified by the token. In an example, the confidential information may be associated with a digital wallet disposed on a device from which the authentication of the token is transmitted. In an example, transmitting authentication of the token may be facilitated by a provider-specific widget executing on a device from which the wrapper for an electronic transaction message is sourced.

[00411] With reference to FIG. 44, there is illustrated a method of tokenization of transaction wrapper in a dynamic ecosystem 4402 for secure commerce and payment transactions according to exemplary and non-limiting embodiments. A point-of-sale device 4404 may connect to a mobile wallet 4406 running on a mobile device 4408. In exemplarily and non-limiting embodiments, multiple mobile wallets may be running on the mobile device. The point-of-sale device 4404 may communicate with the mobile wallet 4406 running on the mobile device 4408 over a communication link 4410. The communication link 4410 may be a wired communication link, a wireless communication link, an NFC link, a Bluetooth link, a local area network link, a wide area network link or the like.

[00412] With continued reference to FIG. 44, the mobile wallet 4406 running on the mobile device 4408, may establish a secure channel with the issuing bank 4412. The secure channel 4414 may be established directly between the mobile wallet 4406 running on the mobile device 4408 or through an over the air interface 4414, such as through a service provider 4416 and a network 4418. Upon authorization, the issuing bank 4412 may generate a token and map it to the wrapper information stored at the issuing bank 4412. The token will then be sent to the mobile wallet 4106 running on the mobile device 4108 and the mobile wallet 4106 running on the mobile device 4408 will use the

token to complete the transaction at the point-of-sale device 4404. The mobile wallet 4406 running on the mobile device 4408, may generate and store a stack of tokens upon personalization of the mobile wallet 4406 and use these tokens to complete the transaction at the point-of-sale device 4404. The mobile wallet 4406 running on the mobile device 4408, may generate, at the time of the transaction, and use a token to complete the transaction at the point-of-sale device 4404.

[00413] The methods and systems described herein of multi-domain ecosystem secure personalized transactions includes a method for performing tokenized electronic transactions in a multi-domain ecosystem for secure personalized transactions. The method includes generating a token representing an electronic transaction message that identifies at least one transaction and at least one ecosystem participant in the transaction. The method may further include wrapping the token representing the electronic transaction message and providing a wrapper that may include a token representing an electronic transaction message. The method may further include generating a token representing the wrapper and providing a tokenized wrapper for a tokenized representation of an electronic message. The method may further include performing an electronic transaction between two participants in the ecosystem by transmitting the tokenized wrapper between two the two participants. In an example one of the two participants in the ecosystem may be the at least one ecosystem participant.

[00414] With reference to FIG. 45, there is illustrated a method of performing electronic transactions across an open network by transmitting a dynamically generated tokenized wrapper that includes a token representing transaction content in a dynamic ecosystem 4502 for secure commerce and payment transactions according to exemplary and non-limiting embodiments. A point-of-sale device 4504 may connect to a mobile wallet 4506 running on a mobile device 4508. Multiple mobile wallets may be running on the mobile device. The point-of-sale device 4504 may communicate with the mobile wallet 4506 running on the mobile device 4508 over a communication link 4510. The communication link 4510 may be a wired communication link, a wireless communication link, an NFC link, a Bluetooth link, a local area network link, a wide area network link or the like.

[00415] With continued reference to FIG. 45, the point-of-sale device 4504 may receive tokenized information from the mobile wallet 4506 running on the mobile device 4508,

wrap the tokenized information in a tokenized wrapper and send the wrapper information to the mobile device which may communicate the tokenized wrapper that includes tokenized content to an issuing bank 4512 associated with the mobile wallet for authorization. The mobile device 4508 may transmit the tokenized wrapper with tokenized content to the issuing bank directly through an over the air interface 4514 or through a service provider 4516 and a network 4518. The tokenized information and/or the tokenized wrapper may consistent of a numeric characters, alphabetic characters, a combination of alphabetical and numeric characters, a truncated primary account number with alphabetic and numeric characters replacing the middle digits of the primary account number, and the like. The token may be a one-time use token. The token may be issued by a trusted entity in the dynamic ecosystem for secure commerce and payment transactions. The trusted entity may be a server located at the payment gateway, acquirer, or issuer. The token may be generated as a mathematically reversible cryptographic function, a one-way non-reversible cryptographic function, or assigned through an index function, sequence number or a randomly generated number.

[00416] In an example of protocol abstraction in transaction flow that includes multiple acquisition methods with multiple issuing banks a merchant may wish to utilize multiple payment information acquisition devices from multiple issuing banks, such as near-field communication (NFC), a barcode reader, and Bluetooth, in order to offer choices in payment methods to the merchant's customers. In this example, protocol abstraction would be used at the merchant's point-of-sale terminal, in order to allow the multiple payment acquisition devices to communicate to a single point-of-sale terminal, via a single communication protocol and link, such as Internet Protocol, by abstracting the individual acquisition protocols for each acquisition device, into a single protocol understood by the merchant's point-of-sale terminal, which can then be routed to multiple issuing banks, over one or more communication protocols and links.

[00417] In an example of protocol abstraction in transaction flow that includes multiple acquisition methods with a single issuing bank, referring to FIG. 46, a merchant may wish to utilize multiple payment information acquisition devices from a single issuing bank 4602, such as near-field communication (NFC) 4604, a barcode reader 4606, and Bluetooth 4608, in order to offer choices in payment methods to the merchant's

customers. In this example, protocol abstraction would be used at the merchant's point-of-sale terminal 4610, in order to allow the multiple payment acquisition devices to communicate to a single point-of-sale terminal, via multiple protocols and links 4612, by abstracting the multiple acquisition protocols for each acquisition device, into a single protocol understood by the merchant's point-of-sale terminal, which can then be routed to the issuing bank, over a single communication protocol and link 4614.

[00418] In an example of protocol abstraction in transaction flow that includes a single acquisition method with multiple issuing banks, a merchant may wish to utilize a single payment information acquisition device to acquire payment information from multiple issuing banks, such as near-field communication (NFC), a barcode reader, and Bluetooth, in order to offer choices in payment methods to the merchant's customers. In this example, protocol abstraction would be used from the merchant's point-of-sale terminal, in order to allow the single payment acquisition device to communicate with a single point-of-sale terminal, via a single communication protocol and link, such as Internet Protocol, then abstracting the individual acquisition protocol for the single acquisition device into multiple communication protocols for the multiple issuing banks, allowing the appropriate transactions to be routed to the correct issuing banks, over multiple communication protocols and links.

[00419] In an example of protocol abstraction in over-the-air transaction flow that includes multiple acquisition methods and multiple issuing banks, multiple issuing banks may wish to send authorization information to a merchant that utilizes multiple payment information acquisition devices, such as near-field communication (NFC), a barcode reader, and Bluetooth, in order to offer choices in payment methods to the merchant's customers. In this example, protocol abstraction would be used at the merchant's point-of-sale terminal, in order to allow the multiple issuing banks to communicate to a single point-of-sale terminal, via a single communication protocol and link, such as Internet Protocol, by abstracting the individual communication protocols for each issuing bank, into a single protocol understood by the merchant's point-of-sale terminal, which can then be communicated each acquisition device, multiple communication protocols and links.

[00420] In an example of protocol abstraction in over-the-air transaction flow that includes multiple acquisition methods and a single issuing bank, a single issuing bank

may wish to send authorization information to a merchant that utilizes multiple payment information acquisition devices, such as near-field communication (NFC), a barcode reader, and Bluetooth, in order to offer choices in payment methods to the merchant's customers. In this example, protocol abstraction would be used at the merchant's point-of-sale terminal, which connects to the single issuing bank, via a single communication protocol and link, such as Internet Protocol, and abstracting the individual communication protocol for the issuing bank, into multiple protocols understood by each acquisition device, which can then be communicated to one or more acquisition devices, over one or more communication protocols and links.

[00421] In an example of protocol abstraction in over-the-air transaction flow that includes single acquisition method and multiple issuing banks (as illustrated in FIG. 47), multiple issuing banks 4702 may wish to send authorization information to a merchant that utilizes a single payment information acquisition device, such as a near-field communication (NFC) device 4704. In this example, protocol abstraction would be used at the merchant's point-of-sale terminal 4710, in order to allow the multiple issuing banks to communicate to a single point-of-sale terminal, via a single communication protocol and link 4712, by abstracting the multiple communication protocols and communication links 4714 used by multiple issuing banks, into a single protocol understood by the merchant's point-of-sale terminal, which can then be communicated to the single acquisition device, via a single communication protocol and link 4712.

[00422] In an example of tokenized transactions, tokenized protocol abstraction, and tokenized wrapper, referring to FIG. 48, a user may present an NFC-enabled smartphone 4802 to a merchant's NFC reader 4804 at a point-of-sale system 4806. The NFC-enabled smartphone 4802 dynamically generates a token that contains the credit card number and expiration date of the user and passes the token through the NFC reader 4804 to the point-of-sale system 4806. The point-of-sale system 4806 passes the token to the mobile transaction platform server 4808 at the issuing bank 4810. The mobile transaction platform server 4808 at the issuing bank 4810 decrypts the token using a key that resides only on the mobile transaction platform server, authenticates the transaction, and sends an authorization code back to the point-of-sale system 4806. The point-of-sale system 4806 then completes the transaction with the user through the NFC reader 4804.

[00423] Further in this example, multiple issuing banks may wish to send authorization information to a merchant that utilizes multiple payment information acquisition devices, such as near-field communication (NFC), a barcode reader, and Bluetooth, in order to offer choices in payment methods to the merchant's customers. In this example, protocol abstraction could be used at the merchant's point-of-sale terminal, in order to allow the multiple issuing banks to communicate to a single point-of-sale terminal, via a single communication protocol and link, such as Internet Protocol, by abstracting the individual communication protocols for each issuing bank, into a single protocol understood by the point-of-sale terminal, which can then be communicated each acquisition device, via multiple communication protocols and links.

[00424] Also, user A may present an NFC-enabled smartphone to a merchant's NFC reader that uses protocol X and user B may present a bar code displayed on a smartphone to the merchant's bar-code reader that uses protocol Y at the point-of-sale system. The smartphones of both users dynamically generate a tokens that contain the credit card number and expiration date of the respective user and passes the token through the NFC reader for user A and the bar-code reader for user B to the point-of-sale system. The point-of-sale system abstracts the different protocols used by the NFC and bar-code readers to communicate to the point-of-sale into a single protocol that cane used to communicate the information for both tokens to the mobile transaction platform server at the issuing bank via a single protocol and single communication link. The mobile transaction platform server at the issuing bank decrypts the tokens using keys that reside only on the mobile transaction server, authenticates the transactions, and sends authorization codes back to the point-of-sale system. The point-of-sale system then converts the abstracted protocol back into the individual protocols used by the NFC and bar code readers to complete the transactions with User A and User B.

[00425] In an alternate example, an issuing bank may wish to send authorization information to a merchant that utilizes multiple payment information acquisition devices, such as near-field communication (NFC), a barcode reader, and Bluetooth, in order to offer choices in payment methods to the merchant's customers. In this example, a tokenized wrapper could be used at the merchant's point-of-sale terminal, in order to allow the multiple issuing banks to communicate to a single point-of-sale terminal, via a

single communication protocol and link, such as Internet Protocol, by abstracting the individual communication protocols for each issuing bank, into a single protocol understood by the point-of-sale terminal, which can then be communicated each acquisition device, via multiple communication protocols and links. In yet another example, user A may present an NFC-enabled smartphone to a merchant's NFC reader that uses protocol X and user B may present a bar code displayed on a smartphone to the merchant's bar-code reader that uses protocol Y at the point-of-sale system. The smartphones of both users send the credit card number and expiration date of the respective users and pass the credit card numbers and expiration dates through the NFC reader for user A and the bar-code reader for user B to the point-of-sale system. The point-of-sale system wraps the different protocols used by the NFC and bar-code readers to communicate to the point-of-sale system into a single protocol that can be used to communicate the information for both credit card numbers and expiration dates to the mobile transaction platform server at the issuing bank via a single protocol and single communication link. The mobile transaction platform server at the issuing bank replaces the credit card numbers and expiration dates with tokens and sends the tokens back to the point-of-sale system. The point-of-sale system then converts the wrapped tokens back into the individual protocols used by the NFC and bar code readers and passes the tokens back to the NFC and bar-code readers. The NFC and bar-code readers then replace the credit card numbers and expiration dates with the tokens.

[00426] The methods and systems described herein of multi-domain ecosystem secure personalized transactions includes a mobile transaction processing system that includes a plurality of servers and a plurality of client devices deployed in a network. Each server may be configured to facilitate secure personalized transactions among the plurality of client devices and a plurality of service provider servers in the network. Further, at least one of the plurality of client devices may be configured with a multi-tenant capable wallet operable in a container-based execution environment. The wallet may be accessible through a widget that may be disposed on the at least one of the plurality of client devices. Additionally, the plurality of servers may further be configured to facilitate transactions between the at least one of the plurality of client devices through the widget and at least one of the plurality of service provider servers

through any of a plurality of trust models including a single trust domain, a single trust cluster, multiple trust clusters, and a direct trust relationship. In an example, the plurality of distinct trust models may include at least one of a single trust domain, a single trust cluster, multiple trust clusters, and direct trust relationship. In an example, the single trust domain may comprise a central wallet service center server that may interface with an electronic wallet on a user device. The central wallet service center server may facilitate establishing secure transaction channels with trusted service providers. In an example, the single trust cluster may comprise a trusted third party server for securely exchanging confidential wallet information with a personal electronic wallet and a wallet service center server for facilitating direct authentication of the user by a each of a plurality of service providers via the third party server. In an example, the wallet service center server may deliver user identification and security keys to at least one of the plurality of service providers. In an example, multiple trust clusters may comprise a plurality of distinct trusted third parties. Each third party may be configured with at least a server for exchanging confidential wallet information with a personal wallet and further for exchanging user and/or device identification provided by a wallet service center with a plurality of cluster-specific service providers.

[00427] In an example, a direct trust relationship may comprise a home wallet service center server for establishing transaction context with an electronic wallet. The home wallet service center server may facilitate establishing an authenticated transaction channel directly between the electronic wallet and the service provider. In an example, the widget may be configured for secure interaction with a specific provider. In an example, the widget may be configured for secure interaction through an intermediary with a specific provider. In an example, the widget may be configured on the mobile device over the air through an electronic interaction between the mobile device and a networked server associated with the specific provider.

[00428] The mobile transaction platform as described herein may facilitate conducting secure personalized transactions in an ecosystem via a plurality of distinct trust models. Each transaction between a client device in the ecosystem and a service provider server or the like in the ecosystem may be conducted via distinct trust models based on a variety of factors related to the transaction, the ecosystem configuration, and the like. Likewise,

a service provider, or other participant in the ecosystem may determine a preferred or default trust model by which transactions are conducted with the service provider. By providing support for transaction-specific trust model determination, the platform may facilitate dynamic trust model determination and transactional use that can be influenced by a plurality of factors including transactional factors, ecosystem instantiation factors, service-provider preference factors, mobile network provider preferences, and the like.

[00429] Trust models may include single trust domain models, single trust cluster models, multiple trust cluster models, direct trust relationship, and the like. A single trust domain model is characterized by a one-to-one relationship between a mobile device and a mobile transaction platform server (also referred herein as a Wallet Service Center (WSC) server. However, the WSC experiences a one-to-many relationship with implicitly trusted service providers. The device wallet and the WSC handle user authentication, payment authentication, and the like for the service providers who rely on the WSC to ensure transaction and end user integrity. FIG. 49 depicts an embodiment of a single trust domain model. A device wallet 4902 communicates directly with a WSC server 4904 in a one-to-one relationship as described above. Each of a plurality of trusted service provider servers 4908a and 4908b communicate over a network with the WSC server 4904. The WSC server 4904 may perform all wallet and user related authentication, validation, payment approval, and the like, while service providers 4908 implicitly trust the WSC server 4904 for this purpose. A user of the wallet 4902 may conduct transactions through the WSC server 4904 to any of a variety of service providers 4908 without needing to share confidential or personally identifiable information with the service provider.

[00430] Trust models may also include a single trust cluster model as depicted in FIG. 50, which assumes that a trusted third party (TTP) 5002 is accessible in the ecosystem. The service providers 4908 do not maintain a direct trust relationship with the end-user via the device 4902, nor do they maintain an implicit trust relationship with a WSC 4904. Instead, the WSC 4904 performs authentication of the user/mobile device 4902 and may share such authentication via a ticket or token that the mobile device 4902 can explicitly provide to the TTP 5002 where transactions between an end user 4902 and a service provider 4908 may be consummated. An MTP Server 4904 maintains a strong

relationship with the TTP 5002, and acts as a user registration entity. For example, in PKI authentication, the MTP Server 4904 acts as an RA (Registration Authority), while CA (Certification Authority) is considered to be a TTP 5002. As noted, service providers do not necessarily trust the MTP Server 4904 for authentication; however, the MTP Server 104 is trusted as an identification provider (IdP). Each service provider 4908 can authenticate the user identified by this ID, using the credentials forwarded by MTP Server 4904. A transaction is conducted via an explicit trust relationship between a wallet 4902 and a TTP 5002, wherein a wallet and a TTP can identify and authenticate each other securely. The Wallet's credentials need not be shared with the MTP Server.

[00431] Trust models may include a multiple trust cluster model as depicted in FIG. 51, which may include a distinct TTP 5002 for each of a plurality of service provider clusters 5102. Each cluster-specific TTP may establish direct trust with a wallet 4902. The MTP server 4904 performs user registration, identity providing, and the like as depicted and described with respect to FIG. 50.

[00432] Trust models may also include a direct trust relationship as depicted in FIG. 52, in which no two entities are compelled to trust each other. Each service provider 5204 maintains a direct credential with the user / client 4902 and authenticates directly.

[00433] The main characteristics of this model include: wallet/user 4902 maintains a credential for each service provider 5204 separately. For each transaction/service, the wallet has to be authenticated by each entity involved in the service fulfillment. A MTP Server does not control all aspects of a transaction; rather a transaction context is shared between each service provider. A wallet can use any type of authentication with the service provider, as far as it meets the security/risk requirements of the service provider. In an example, for a bill issuer authentication can be as simple as user's zipcode and mother's maiden name. For a payment service provider, this can be a credit card CVV or VbV/SecureCode password. Other authentication methods can be incorporated into direct trust relationship models including without limitation, three-dimensional authentication of the user, device, and domain.

[00434] The methods and systems of transaction processing with a plurality of trust models may include providing a multi-domain ecosystem platform for secure personalized transactions capable of operating with a plurality of distinct trust models. In

such methods and systems a mobile transaction processing system may be configured with a plurality of servers and a plurality of client devices deployed in a network. Each server may be configured to facilitate secure personalized transactions among the plurality of client devices and a plurality of service provider servers in the network. The system may include at least one of the plurality of client devices configured with a multi-tenant capable wallet operable in a container-based execution environment, wherein the wallet is accessible through a widget disposed on the at least one of the plurality of client devices. The plurality of servers in the system may be further configured to facilitate transactions between the at least one of the plurality of client devices via the widget and at least one of the plurality of service provider servers via any of a plurality of trust models including a single trust domain, a single trust cluster, multiple trust clusters, and a direct trust relationship. A single trust domain may comprise a central wallet service center server that interfaces with an electronic wallet on a user device and facilitates establishing secure transaction channels with trusted service providers. A single trust cluster may comprise a trusted third party server for securely exchanging confidential wallet information with a personal electronic wallet and a wallet service center server for facilitating direct authentication of the user by a each of a plurality of service providers via the third party server, wherein the wallet service center server delivers user identification and security keys to at least one of the plurality of service providers. Multiple trust clusters may comprise a plurality of distinct trusted third parties each configured with at least a server for exchanging confidential wallet information with a personal wallet and further for exchanging user and/or device identification provided by a wallet service center with a plurality of cluster-specific service providers. Also, a direct trust relationship comprises a home wallet service center server for establishing transaction context with an electronic wallet that facilitates establishing an authenticated transaction channel directly between the electronic wallet and the service provider. The widget disposed on the client device(s) may be configured for secure interaction with a specific service provider or through an intermediary with a specific provider. The widget may further be configured on the mobile device over the air through an electronic interaction between the mobile device and a networked server associated with the specific provider.

[00435] The methods and systems described herein of multi-domain ecosystem secure personalized transactions include a secure mobile wallet operative on a mobile device. The secure mobile wallet may include a plurality of distinct user accounts. In an example, access to a distinct user account may be facilitated by a provider-specific trust model through a widget disposed on the mobile device. In an example, the widget may be configured for secure interaction with a specific provider. In an example, the widget may be configured for secure interaction through an intermediary with a specific provider. In an example, the widget may be configured on the mobile device over the air through an electronic interaction between the mobile device and a networked server associated with the specific provider.

[00436] The methods and systems described herein of multi-domain ecosystem secure personalized transactions include a method of trust model determination for a mobile transaction in an ecosystem for secure personalized transactions. The method may include configuring a plurality of servers deployed in a network to facilitate transactions between a client device and a provider-specific server through any of a plurality of trust models. The plurality of trust models may include such as a single trust domain, a single trust cluster, multiple trust clusters, and a direct trust relationship. The method may further include analyzing a default trust model for a transaction between the mobile device and the provider-specific server to determine a measure of risk associated with the transaction. The method may further include conducting the transaction through one of the default trust model and another trust model from the plurality of trust models, based on the resulting measure of risk. In an example, conducting the transaction may include accessing a wallet function on the mobile device via a provider-specific widget executing on the mobile device. The provider-specific widget may be selected based on the resulting measure of risk. In an example, the measure of risk may be based on a location of the mobile device. In an example, the measure of risk may be based on a location of the provider-specific server relative to a location of the mobile device. In an example, the measure of risk may be based on aspects of a network over which the mobile device is conducting the transaction.

[00437] The methods and systems of secure personalized transactions in an ecosystem may include a mobile wallet that is capable of operating with a plurality of distinct trust

models. Such a secure mobile wallet may be operative on a mobile or other client device and may comprise a plurality of distinct user account. Access to a distinct user account may be facilitated by a provider-specific trust model via a widget disposed on the mobile device. As noted above the widget disposed on the client device(s) may be configured for secure interaction with a specific service provider or through an intermediary with a specific provider. Also as noted above, the widget may further be configured on the mobile device over the air through an electronic interaction between the mobile device and a networked server associated with the specific provider.

[00438] The mobile transaction processing system as described herein for providing an ecosystem for secure personalized transactions may facilitate trust model determination for a mobile transaction. Model determination may include configuring a plurality of servers deployed in a network to facilitate transactions between a client device and a provider-specific server via any of a plurality of trust models including a single trust domain, a single trust cluster, multiple trust clusters, and a direct trust relationship. Model determination may further include analyzing a default trust model for a transaction between the mobile device and the provider-specific server to determine a measure of risk associated with the transaction. Based on the resulting measure of risk, the transaction may be conducted via the default trust model or another trust model selected from the plurality of trust models that include a single trust domain, a single trust cluster, multiple trust clusters, and a direct trust relationship.

[00439] Conducting the transaction may include accessing a wallet function on the mobile device via a provider-specific widget executing on the mobile device. The provider-specific widget may be selected based on the resulting measure of risk, which may be based on a location of the mobile device or a location of the provider-specific server relative to a location of the mobile device. Also, the measure of risk may be based on aspects of a network over which the mobile device is conducting the transaction.

[00440] The methods and systems described herein of multi-domain ecosystem secure personalized transactions include a domain-specific program language for use in an ecosystem. The ecosystem may include an instrument issuer, a service provider, an acquirer, and a client. The domain-specific program language may include a syntax that may be compliant with such as XML, CSS, and Javascript. The language may further

include an object that may enable a wallet-related function on a client device. In an example, the enabled wallet-related function may be at least one of a wallet container, a wallet, a widget, and a wallet companion applet.

[00441] The mobile transaction platform methods and systems that facilitate secure personalized transactions in a multi-domain ecosystem may include an multi-domain ecosystem domain-specific language that supports XML, CSS, Javascript and the like for creating wallets and widgets that may be configured to operate as a runtime component of a container. Such a language may include one or more objects, tags, metadata elements and the like that may enable a container, wallet, widget, companion applet and the like. The multi-domain ecosystem domain-specific language may provide domain specific Javascript or XML functions to access a variety of modules, functions, and services of the mobile transaction platform. In addition, functions for enabling NFC operations, accessing barcode information, and the like may also be included.

[00442] By creating a domain-specific set of functions within a standardized software pattern (e.g. an official language), the domain-specific language reveals functions through standardized patterns so that users and software-related architects may readily determine that a domain-specific language is being patterned. This domain-specific language is applied to software programming and module configuration of an ecosystem that is described herein to, among other things, simplify generating code to take action for that domain. Each domain-specific function may represent a uniquely novel aspect of the domain-specific language. The software development environment of the MTP may allow use of the domain-specific set of functions as a variant of the domain-specific language. In an example, adapting the domain specific-language with a wallet enabling function may provide a modified language that is particularly suited for development of functions that enable creation of services that support secure, personalized, mobile transaction services. A developer can use the MTP domain-specific language to create services based on the set of functions that may be specific for creating secure mobile transactions. As a domain-specific language, there may be provided native tools to check the language; validate the use of it; and ensure that encryption is properly used.

[00443] The methods and systems described herein of multi-domain ecosystem secure personalized transactions include a platform for serving secure personalized transactions

in a multiple domain ecosystem. The platform may include an interpreter for interpreting domain-specific program language objects for enabling wallet-related functions that are relevant to at least one of the multiple domains. In an example, the interpreter may be a runtime interpreter. In an example, the interpreter may be a runtime container executing on a client device. In an example, the enabled wallet-related function may be at least one of a wallet container, a wallet, a widget, and a wallet companion applet.

[00444] In addition to an XML, CSS, and Javascript compliant language, the MTP may include a domain-specific language interpreter that may operate on the client device (e.g. as part of the runtime client). The runtime itself may be a language interpreter, which interprets the code based on the domain-specific language and executes that functionality on the client (e.g. a mobile phone). A client operable runtime (e.g. container) may facilitate interpreting domain specific language elements that are relevant to a domain, such as the multi-domain ecosystems for secure personalized transactions described herein. Although the description of the domain-specific language is based on a standard language that is Javascript, XML, and CSS compliant, the domain-specific language could be based on nearly any interpreted language (e.g. VB script, visual basic, and the like).

[00445] Such a domain-specific programming language for use in an ecosystem involving an instrument issuer, a service provider, an acquirer, and a client, may include a syntax that is compliant with XML, CSS, and Javascript, and a domain-specific object/element/tag that enables a wallet-related function on a client device. In particular, the enabled wallet-related function may be a wallet container, a wallet, a widget, a wallet companion applet, or a combination thereof.

[00446] A platform for serving secure personalized transactions in a multiple domain ecosystem, such as an MTP may include an interpreter for interpreting domain-specific program language objects for enabling wallet-related functions that are relevant to at least one of the multiple domains. The interpreter may be a runtime interpreter, a runtime container executing on a client device, and the like. Also, the enabled wallet-related function may be at least one of a wallet container, a wallet, a widget, and a wallet companion applet.

[00447] The methods and systems described herein of multi-domain ecosystem secure personalized transactions include a set of tools for securing content in a multi-domain ecosystem. The set of tools may include a software development kit for constructing a mobile transaction ecosystem. The set of tools may further include a set of domain-specific functions for enabling client-side functionality of at least one of a widget, a wallet, a container, and a companion applet. The set of tools may further include device agnostic application programming interfaces. The set of tools may further include device-specific application programming interfaces. The set of tools may further include enabling tier functions for facilitating interoperation with client device software and hardware. The set of tools may further include widget publishing modules to facilitate availability of provider-specific widgets for delivery and downloading to a client device.

[00448] As described herein, a software development kit (SDK) for a mobile transaction processing platform (MTP) may facilitate securing content in a multi-domain ecosystem for secure personalized transactions. The MTP SDK may provide service provider programmers and the like with tools for implementing and configuring transaction workflows, service provider-specific widgets for use on client devices, wallets that include device specific and device agnostic capabilities that may be accessed through APIs, and the like to secure content. The SDK leverages an on-device runtime container that further facilitates isolating widgets, wallets, and the like to facilitate seamless content security. The runtime container, also referred herein and elsewhere as mClient may also include a user interface framework, and a workflow engine that provide service execution, actions, application (e.g. widget) execution control, scripting, and communication support for a variety of communication channels. The MTP tools for securing content in a multi-domain ecosystem for secure personalized transactions may facilitate a development process that includes defining business requirements, creating user interface flows, developing widgets, applets, and server-based applications, building and testing the developed elements.

[00449] The methods and systems described herein of multi-domain ecosystem secure personalized transactions include a network-enabled server for configuring a set of service provider tools for a multi-domain ecosystem for secure personalized transactions. The network-enabled server may include a mobile transaction platform server. The

network-enabled server may further include a wallet service center development environment operable on the platform server. The network-enabled server may further include a wallet service center runtime module operable on the platform server. The network-enabled server may further include a software development kit for constructing and maintaining server-based and client-based elements of the ecosystem. In an example, the elements of the ecosystem may include wallets, widgets, companion applets, and containers.

[00450] The MTP may comprise a platform for configuring a set of service provider tools for a multi-domain eco system for secure personalized transactions. The MTP may comprise a network-enabled server for configuring a set of service provider tools including a wallet service center development environment operable on the network-enabled server, a wallet service center runtime module operable on the server, and a software development kit for constructing and maintaining server-based and client-based elements of the ecosystem. The client-based elements that may be maintained may include wallets, widgets, companion applets, containers, and the like.

[00451] A widget configuration capability of a mobile transaction platform may include one or more applications operable on a client device in an ecosystem. The widget configuration application(s) may include a user interface that initially appears as a gray widget. The configuration application may include a user interface by which widget user interface icons and the like may be positioned in the gray widget, customized in look and feel (e.g. layouts, color, etc) and saved on the client device for use as a validated widget. Such a widget configuration application may enable user-specific customization of widget interfaces.

[00452] The methods and systems described herein of multi-domain ecosystem secure personalized transactions include a mobile application for in-situ customizing of a portion of a widget. The mobile application may include a user interface template for receiving widget functional code and descriptors from a widget development process performed on a networked server. The mobile application may further include a user interface for facilitating access to widget features and functions through icons defined by a widget descriptor. The mobile application may further include an algorithm for configuring a widget downloaded to the device on which the user interface may be

displayed based on user actions detected in the user interface. In an example, the user interface template may be automatically configurable based on the widget descriptor. In an example, the application for in-situ customizing of a portion of a widget may operate on a mobile device. In an example, the application may execute within an mClient container runtime environment.

[00453] The MTP may include a mobile application for in-situ customizing of a portion of a widget that may include a user interface template for receiving widget functional code and descriptors from a widget development process performed on a networked server, a user interface for facilitating access to widget features and functions via icons defined by a widget descriptor, and an algorithm for configuring a widget downloaded to the device on which the user interface is displayed based on user actions detected in the user interface. The application may automatically configure the user interface template based on the widget descriptor. The application for in-situ customizing of a portion of a widget may be adapted to operate on a mobile device. The application may execute within an mClient container runtime environment.

[00454] The methods and systems described herein of multi-domain ecosystem secure personalized transactions include a server-based application that includes a set of reference mobile transaction platform services. The application may further include an automated wizard application for accessing the set of reference services and configuring at least one service from the set of reference services based on user input. In an example, the automated wizard application may facilitate in configuring at least one of client element distribution, issuer environments, settlement environments, aggregation services, transaction medium support, support for handset types, user experience, business model, and regulatory considerations based on the user input.

[00455] The methods and systems described herein of multi-domain ecosystem secure personalized transactions include an ecosystem configuration guidance system that includes a set of reference mobile transaction platform services. The ecosystem configuration guidance system may further include an automated wizard application for receiving user input associated with a desired ecosystem configuration. The automated wizard application may also be operable for recommending a configuration of at least one service from the set of reference services based on the user input. In an example, the

set of reference services may include such as distribution services, continuum use services, issuer services, settlement services, aggregation services, transaction medium requirements, handset and operating system considerations, user experience factors, business modeling, and regulatory considerations. In an example, the user input associated with a desired ecosystem configuration may be a hyperlink. In an example, the wizard may search for information based on the hyperlink and may populate at least one reference service template. In an example, the hyperlink may be a link to a website of a service provider operating the wizard.

[00456] The mobile transaction platform may further include a wizard to facilitate configuring a plurality of aspects of a multi-domain ecosystem for secure personalized transactions. The wizard may be an automated application or the like that accesses a set of reference mobile transaction platform services and configures at least one service from the set of reference services based on user input. In an example, the automated wizard application facilitates configuring at least one of client element distribution, issuer environments, settlement environments, aggregation services, transaction medium support, support for handset types, user experience, business model, and regulatory considerations based on the user input.

[00457] The mobile transaction platform may alternatively include an ecosystem configuration guidance system that may include a set of reference mobile transaction platform services, and an automated wizard application for receiving user input associated with a desired ecosystem configuration and recommending a configuration of at least one service from the set of reference services based on the user input. In an example, the set of reference services may include distribution services, continuum use services, issuer services, settlement services, aggregation services, transaction medium requirements, handset and operating system considerations, user experience factors, business modeling, and regulatory considerations.

[00458] The ecosystem configuration guidance system may receive a hyperlink as user input associated with a desired ecosystem configuration. The wizard may search for information based on the hyperlink and may populate at least one reference service template.

[00459] In another example, the hyperlink may be a link to a website of a service provider operating the wizard.

[00460] A method of type-length variable data communication among participants in an multi-domain ecosystem for secure personalized transactions may include receiving a plain old java object at a producer server that is a participant in the multi-domain ecosystem; converting the received plain old java object to a type length variable stream that complies with a type length variable protocol; and transmitting the converted object via a one-way message bus from producer to a mobile device participant in the multi-domain ecosystem. The plain old java object may conform to type length variable protocol guidelines. The type length variable protocol may support streaming, float support, and prepended runtime class type information.

[00461] The present method may provide for a message bus for a mobile transaction platform in a dynamic ecosystem for secure commerce and payment transactions. The message bus may enable the one-way exchange of ancillary data between components within a mobile transaction platform (MTP). The message bus may be channel agnostic to allow the exchange of data through multiple channels. The multiple channels may be proprietary protocol channels, a short-messaging-service (SMS) channel, the push notification framework of a mobile network operator, and the like. The proprietary protocol channel may be a type length variable (TyLV) protocol channel. The TyLV protocol channel may enable mobile device software to communicate with a transaction platform. The TyLV protocol channel may facilitate the conversion of a plain old java object (POJO) to be converted into a TyLV protocol stream and a TyLV protocol stream to be converted into a POJO. The conversion process may be an automated conversion process. The conversion process may require that the POJO conform to guidelines. The guidelines could require that the POJO: be a publicly defined class, implement the marker interface com.csam.wsc.daf.tylv.TyLVBindObject, have data member values that are all non-final, non-static, non-transient, and public, be a non-negative number, pass null instead of the actual value to indicate an optional value, not be an anonymous class, must provide a default constructor, provide class type information for the decode operation, and the like. The TyLV protocol may also require that member value types be declared. Member value types may be: short and its wrapper Short class, int and its

wrapper Integer class, long and its wrapper Long class, Boolean and its wrapper Boolean class, java.math.BigInteger, byte array,java.util.Map and its sub types, java.util.List and its sub types, Composed user defined types implementing TyLVBindObject, java.lang.String, representing java modified UTF encoding, java.util.Date, C-SAM defined com.csam.wsc.core.util.BitSequence class type, and the like. The TyLV protocol may also support other functions. Other functions may include inheritance support for streaming, float support, runtime class type information prepended to the actual data of that class type, and the like. The one-way exchange may enable the sending of data objects without explicitly requiring network round-trips. The ancillary data may not be the primary payload of a service request. The ancillary data may be received and consumed by a consumer, which may reside within a mobile application. The mobile application may be a wallet application or the like. The ancillary data may be produced and sent by a producer. The producer may be a component within the mobile transaction platform.

[00462] The message bus for a mobile transaction platform in a dynamic ecosystem for secure commerce and payment transactions may provide for a message. The message may be an atomic packet of data that can be transmitted over a communication link. The message may be any business object that could be exchanged in the form of data through the message bus. The message may be comprised of a message ID and payload. The message ID could be the unique identifier of the message determined by a developer of the system. The payload may be the actual data exchanged between the producer and the consumer. The format of the payload data may be a string. The message may be sent by a component on a channel. The message may be received and processed by a receiver component.

[00463] The message bus for a mobile transaction platform in a dynamic ecosystem for secure commerce and payment transactions may provide for a message channel. The message channel may be a virtual pipe that connects a sender component to a receiver component. The sender component may write information to the message channel and the receiver component may read information from the message channel.

[00464] The message bus for a mobile transaction platform in a dynamic ecosystem for secure commerce and payment transactions may provide for a message class. The

message class may be a platform message or a wallet message. The platform message may not be meant for service consumption and may be consumed and acknowledged by the runtime. A developer may receive a notification for the message in JavaScript and the message may be flagged as consumed. The message may not be able to be acknowledged from JavaScript. An attempt to acknowledge a platform message from JavaScript may be ignored. The wallet message may be made for service consumption and acknowledgement from JavaScript. The wallet message may be produced by the server and may be transformed into an equivalent JSON format and passed to a device.

[00465] The message bus for a mobile transaction platform In a dynamic ecosystem for secure commerce and payment transactions may provide for consumers and producers. Consumers may be entities or components responsible for consuming messages. Producers may be entities or components responsible for producing messages. Consumers may be applications running on mobile devices. Producers may be server components. The producer may register against a message ID. The message ID may uniquely identify the message. The consumer of a message may be the runtime if the data is owned by the runtime. The message may be passed to the central message handling function of the container application. The producer may consume an acknowledgement for a message produced by the producer when the message is returned by the consumer to the producer via the message bus.

[00466] With reference to FIG. 53, there is illustrated a message bus communication architecture 5302 for a mobile transaction platform in a dynamic ecosystem for secure commerce and payment transactions according to exemplary and non-limiting embodiments. A message may be consumed when the messages 5304 arrive through a channel 5306 to a message bus 5308. A channel may be a set of messages. A message 5304 in the channel 5306 may be an item, which will be dispatched to a consumer who has registered for it with the message bus 5308. The messages may be dispatched in the sequence of their occurrence on the channel. Multiple messages 5304 may be dispatched in a batch piggybacked to the same response packet.

[00467] With continued reference to FIG. 53, a consumption strategy 5316 may determine the process of dispatch for items. The process may be to dispatch the messages 5304 in the sequence of the messages' occurrence on the channel 5306. Multiple

messages may be disputed in a batch piggybacked to the same response packet. A consumer 5310 may send an acknowledgement when it consumes a message 5304 to indicate the successful consumption of the message 5304. The acknowledgement may hold the value indicating the successful or unsuccessful consumption of the message 5304. The acknowledgment may be sent back to the server when a mobile application sends its next request to the server. The acknowledgement may be sent during the next request in the next new application session. The acknowledgement may be sent instantaneously and in real-time by the consumer 5310 initiating a special response sequence to acknowledge processing of the message (804).

[00468] With continued reference to FIG. 53, a producer 5312 may begin message production when a response is to be sent via the message bus 5308. The registered producers may produce messages and dispatch them on the output channel. The messages 5304 may then be encoded and the entire list may then be sent via the message bus 5308. A server-based producer may produce a java object which may then be converted to an equivalent JSON object by the mobile transaction platform infrastructure. The production strategy 5314 may dispatch the message on the output channel in the order of the producers registered. The mobile transaction platform server may provide a producer registry where developers register their respective producers. The registration order for the producers may be the order in which the producers are invoked when there is an opportunity to deliver a message.

[00469] In accordance with exemplary and non-limiting embodiments, the present disclosure provides a type-mapping scheme and type definition rules for converting any Plain Old Java Object (POJO) to a Type Length Variable (TyLV) stream and vice-versa. The scheme may enable developers to write device based use cases to define POJOs, which may be converted on the fly to the corresponding TyLV stream. In an example, the TyLV stream may be used in accordance with the TyLV protocol which may enable any device based software to communicate with a transaction provider to accomplish a task.

[00470] In an embodiment, on the fly streaming may help the developers in abstracting away non-essential streaming details. This in turn may result in freeing the developers from understanding a myriad of TyLV protocol related details. The type mapping

scheme and the type definition rules of the present invention may be used in defining device based use cases and other interactions based on the TyLV protocol.

[00471] In an example, the mapping and type definition rules may be defined and used as guidelines such as to cause POJOs to be streamed to the TyLV protocol. A POJO which follows these defined set of guidelines may be referred to as a TyLV compliant POJO. In an example, the guidelines may include such as a rule to define a POJO as a publicly defined class. Further, the POJO may implement a marker interface such as com.xyz.wsc.daf.tylv.TyLVBindObject. In an event of failure of such an implementation, runtime errors of the nature of ClassCastException may be generated. in various embodiments, data members of a POJO may be non-final, non-static, non-transient and public (while all other type of objects may be ignored for the purpose of streaming). In an embodiment, POJO members of integral nature may not supply negative values as the failure to do this may result in runtime exceptions being thrown with a message such as "Decoder does not support length computation of negative numbers". The guidelines may include a rule to allow the POJO to declare member variables such as short and its wrapper Short class, integer and its wrapper Integer class, long and its wrapper Long class, boolean and its wrapper boolean class, java.math.BigInteger, bytearray, java.util.Map and its sub types, java.util.List and its sub types, composed user defined types implementing TyLVBindObject, java.lang.String (representing Java modified UTF encoding), java.util.Date, com.csam.wsc.core.util.BitSequence class type, and the like. In an example, specification of a variable of a type such as other than those defined such as other than those mentioned above may result in runtime exceptions. In cases such as where an object is an aggregate object, the object may be declared as a public inner static or non-static class.

[00472] In an example, as per the rules specified for a Java class format, a compiler may generate synthetic fields for non-static inner classes.

[00473] In an example, the key value pairs such as contained within java.util.Map instance maybe of the type java.lang.String, and overriding Object.toString may not be very helpful for such key value pairs.

[00474] In an alternate example, TyLV protocol implementation may allow Boolean or Boolean data type but internally, it may encode or decode this data types as an 8-bit byte. Thus, a Boolean value true may be treated as 1, and a Boolean value false may be treated as 0. Also, the class type com.xyz.wsc.core.util.BitSequence supported by TyLV protocol may be internally treated an integer-32 value. Further, the data types may be stringized in a specific format. Additionally, a map containing zero entries may not be allowed and a null may be used to represent map as an optional value.

[00475] In an example, systems using TyLV protocol may support the Universal Character Set or UCS Transformation Format (UTF) enncoding definition modified by the Java platform, as the java.lang.String types may be Universal Character Set or UCS Transformation Format (UTF) encoded. Further, primitive type members may not be optional.

[00476] In an example, byte may be non-empty and composed types may be declared on the basis of their concrete types rather than on the basis of the interface they implement.

[00477] In an example, TyLV protocol may support a Homogeneous, variable length, List with objects of same type. The type of objects may be any Java class type or any user defined class type, and may include Wrapper types to support primitive type data.

[00478] In an alternate example, additional guidelines related to POJOs may include defining a rule for any POJO or any of its member variables such that passing a null value instead of an actual value may indicate an optional value. Further, a rule may be defined such that a TyLV compliant POJO may not support type inheritance. Also, declaring a POJO or any of its members as an Anonymous Class may result in runtime exceptions. Additionally, a default constructor may need to be provided as the TyLV engine may not be able to use a defined constructor.

[00479] In an example, additional guidelines related to POJOs may include a rule such that a TyLV codec may require a class type information for decode operation, so that the TyLV engine may correctly identify the BindObject interface implementation type or List content object type which may be populated with decoded data appropriately. In an example, the class type information may be provided as comma separated list of fully qualified class name string values. In an example, the class type information is included herein below.

```
Public class Parent implements BindObject {

Public BindObject child1;

Public int child2;

Public List child3;
}
```

[00480] Additionally, an example of a class type declaration as Action Configuration Parameter may be provided in the following manner:

com.xyz.wsc.comms.Parent, com.xyz.wsc.comms.CertificateResponse, com.xyz.wsc.comms.Certificate

[00481] In the above examples, the parent class may resemble the parent class type which may be composed of BindObject type child1, int type child2, and List type child3. Also, in the class type declaration example, the second class type declaration CertificateResponse may be used to identify the run time type information for child1. Similarly, third class type declaration, the Certificate, may identify the child3 content that may be of the type of object collected in list. The list data content may be provided as any of the wrapper types like Integer, Map or may be any other valid Java type. It may result in a runtime exception if class type information is not provided.

[00482] In an example, additional rules may be defined regarding streaming of object graphs using the TyLV protocol. The streaming of object graphs in the TyLV protocol may be similar to Java object serialization protocol. However, in the TyLV protocol, all the objects composing the graph may be TyLV compliant. Further, a root object of the graph may almost always be an object implementing the TyLVBindObject interface.

[00483] In an example, the TyLV compliant POJOs may be defined as plain value aggregation objects that may not contain any functionality. These POJOs which may be defined for the purpose of TyLV streaming may be used only in scenarios where streaming is the prime objective. Further, putting computation in the POJOs may be avoided.

[00484] In an example, probable range of values for an integral data type may be identified before defining it. As an example, defining a long variable for carrying values that may be accommodated in a short type may be avoided if the probable range of values

for the variable is known. In order to achieve this objective, a type may be assigned by such as anticipating the max value that may need to be transferred. However, in other embodiments such as in cases where anticipation of the maximum value is not possible, the most accommodating type may be selected accordingly. Further, it may be advisable to take care of underflows and overflows in code which may involve numeric computation. This may be helpful in avoiding assignment of negative values to bind object members, which may not be supported by the engine.

[00485] In an example, the TyLV protocol may include inheritance support for streaming. Further, as a protocol change, negative values and float support may be enabled. Also, the TyLV protocol may support prepending of runtime class type information to the actual data of that class type. This may be similar to serialization of an object in Java, and may enable the decoding engine to identify a correct actual bind object type or the list content object type.

[00486] An example of a valid POJO may be defined as:

```
Class ValidBindObject implements TyLVBindObject{

    public String member1 ;
    public int member2 ;
public AggregateObject member3 ;
public AnotherAggregateObject member4 ;
public AnotherAnotherAggregateObject member5 ;

public static class AggregateObject  implements TyLVBindObject{
    public int member1 ;
    public String member2 ;
}

public class AnotherAnotherAggregateObject  implements TyLVBindObject{
    public int member1 ;
    public String member2 ;
}
```

}

[00487] An example of an invalid POJO may be defined as:

class InvalidBindObject implements TyLVBindObject{ @1
public final String member1 ; &&1
public static int member2 ; &&2
private int member3 ; &&3

public TyLVBindObject invalidAggregateMember4 = new TyLVBindObject() {@2
};

public InvalidBindObject (String value){ @3
//whatever
}

public TyLVBindObject invalidAggregateMember5; @4

[00488] In the above examples, error points may be indicated such as by @XX and ignored points may be indicated such as by &&. In an example, these errors may be runtime errors.

[00489] In accordance with exemplary and non-limiting embodiments, this document discloses an MTP 5.x wallet and widget infrastructure model for mPayment and mCommerce ecosystems. The MTP 5.X model may address and incorporate customer feedback on various features associated with an MTP 4.x abstraction model which may be related to committing to an abstraction model and reliance on middleware to build phone application. Some of the exemplary features of MTP 4.x that the model 5.x may address may be such as but are not limited to customer desire to such as incorporate new phone features instantly without waiting for support from middleware providers; expensive training on a new, proprietary tool that may include Human Resource (HR)

related concerns; considerable rework and cost required in migrating away from the middleware.

[00490] In an example, the MTP 5.x model may support development simplicity, lower on-going maintenance effort, and an ability to support third party widgets.

[00491] In an example, an MTP 5.x client may provide flexibility to developers to such as use one or more of the capabilities of the model. The architecture may comprise stand-alone libraries that may contain infrastructure components such as Comms, NFC, Crypto and the like. In accordance with the MTP 5.x model, these components may be extracted into stand-alone libraries. Further, parts or the entire mobile application may be built entirely on a native phone operating system. These parts may include user interface components, business logic, or any other components of the mobile application. Further, the widgets provided in the model may be supported through such as a phone independent application programming interface (API) layer. In an example, the widgets may also use application programming interfaces (APIs) that may be offered by the mobile application, which may be in addition to the application programming interfaces (APIs) available from the runtime. In an example, the model architecture may allow client runtime, to be bundled within the mobile application. In an alternate example, the mobile application may be built on the runtime.

[00492] In an example, the modularity of the MTP 5.x architecture may allow three types of deployment for a mobile application. These three deployment models may include such as an independent native application model, a hybrid ecosystem application model, and a cross platform ecosystem application model. These models may be discussed in greater detail in succeeding description of the invention as follows.

[00493] The independent native application model OM1.1 (hereafter referred to as model 1 for simplicity) may be targeted to customers that may not need widgets. This may provide a unique and custom user experience for such customers. The model1 may include a phone operating system environment including such as a plurality of mobile libraries. The libraries may be used by such as a mobile application such as an mWallet client application. The mobile libraries may facilitate the customers to leverage the existing mobile libraries to build mobile application components, such as a Crypto component or a Comms component, instead of having to build the components from

scratch. The libraries may be designed to facilitate mobile transactions and wallet related capabilities. In an example, the mobile libraries may be coupled to the MTP server and may provide a complete out of the box infrastructure solution for the customers of the mobile application.

[00494] As an example of the independent infrastructure support capabilities of the model 1, a crypto component may be built using this model such that it may provide more than just wrappers over the native crypto operations. The component may perform as a security component that may handle functions such as key generation, key management, personal identification number (PIN) management and the like.

[00495] In an example, each of the libraries may be incorporated by itself, without any external dependency. The developers of the mobile application OM214 may integrate best of the breed libraries from third party sources to provide the complete mobile application solution OM214.

[00496] The hybrid ecosystem application deployment model (also referred to as a model 2 hereafter for simplicity of the description) may be targeted to customers that may require a unique, native experience along with a capability of service extensibility through the widgets. The model 2 may include a mobile application and application programming interface which may be built such as over a phone operating system. In accordance with the model 2, the entire client runtime software may be bundled with the mobile application. The phone operating system may include a plurality of libraries which may be used by the mobile application such as the mWallet client application. The model 2 may also include the widgets that may be supported by such as a phone independent API. In an example, the model 2 may be capable to not compromise on user experience, performance or application functionality. The model may have higher development costs as a variable portion of the mobile application such as the mWallet may be custom built for each phone platform.

[00497] In an example, the client runtime software of the model 2 may switch between native screens and screens built using extensible markup language (XML) through a toolkit. The runtime may also allow integration of actions developed such as in JavaScript, that may be through the toolkit, and actions developed through native implementations. In this manner, in an exemplary embodiment a large part (such as 80

percent) of the mobile application may leverage the runtime and toolkit related development efficiencies while the rest may be built native for a differentiable experience. The widgets may be built using a toolkit such as using XML, cascading style sheets (CSS), and JavaScript. This may allow developers a greater flexibility while staying updated with all phone platforms as the developers may bypass the runtime to create functionality that may not be available through the runtime. In an example, the widgets may be able to use development support through the runtime. However, the widgets may also use the mobile application's exposed application programming interfaces (APIs) for development. The widget toolkit may make it easier and smoother for ecosystem service providers to develop the widgets. The application tools may be enhanced to support hybrid assembly and packaging.

[00498] FIG. OM1.3 illustrates a cross platform ecosystem application deployment model in accordance with the MTP 5.x architecture. The cross platform ecosystem application deployment model (also referred to as model 3 hereafter for simplicity) may be a cost effective model that may be targeted to customers such as whose requirements may be fulfilled by the capabilities of the client application runtime. The model may include a mobile application and API built over such as a phone operating system that may include multiple libraries. Further, in accordance with the cross platform ecosystem application deployment model, multiple widgets may be supported by such as a phone independent API. Further, in accordance with this model, both the client application and widgets may be built using toolkits that may support cross-platform development.

[00499] In an example, the model 3 may be similar to a deployment model in accordance with an MTP 4.x version of the MTP architecture with the exception that MTP 5.x may facilitate or support platform extensions as opposed to the MTP 4.x version.

[00500] FIG. 54 illustrates a high level block diagram of various subsystems of a payment facilitator system 5400 as used in the deployment of a service delivery platform ecosystem (hereafter referred to as SDP 5408 for the purpose of simplicity of description and discussed later), in accordance with an exemplary embodiment of the present invention.

[00501] In accordance with an exemplary embodiment, various subsystems within the payment facilitator 5400 may include a point of sale (POS) 5402 terminal, application (APP) 5404 such as a mobile application 5404, SDP 5408, several issuer systems such as including the depicted issuer system 1 5410, issuer system 2 5412, and issuer system 3 5414, without limitations. The various subsystems may be connected over a network such as a payment processing network 5418. The APP 5404 can further include one or more widgets.

[00502] The payment facilitator 5400 of FIG. 54 may facilitate in leveraging mobile phones at a physical merchant POS 5402 such as the POS 5402 to such as enact payment of goods and services. The mobiles phones may thus be utilized to facilitate payment as an alternative to plastic-based physical transaction cards such that a payment transaction can be processed, routed, and settled through the payment facilitator. As such, a merchant may be facilitated with the use of the payment facilitator 5400 as it may reduce the overall costs associated with a transaction. Also, the payment facilitator 5400 may be advantageous for a customer or a consumer as it provides a rewarding shopping or transaction experience that can save money such as through offers, loyalty, or the like promotional schemes or overall reduced prices offered by the merchants due to savings obtained by the merchants from the payment process with the use of the payment facilitator 5400.

[00503] The payment facilitator 5400 may provide consumers a choice with multiple payment products across several issuers like issuers. The payment facilitator 5400 may provide an improved payment experience to the consumers as compared to other transaction modes and may be as simple as pulling out such as a physical leather wallet, selecting a card, and swiping it over a machine. Also, the payment facilitator 5400 may provide a secured platform with better risk management as compared to conventional plastic cards and payment options. The merchants may also be facilitated with the use of the payment facilitator 5400 such as making several available for the merchants with respect to their individual POS equipments. In addition, several promotional schemes or value added services such as coupons, offers, loyalty, and receipts can be integrated into the payment facilitator 5400 to provide a holistic retail experience for the merchants as well as the consumers.

[00504] In accordance with an exemplary embodiment, FIG. 54 depicts the payment facilitator 5400 from a payment perspective. However, as the payment facilitator 5400 scales similar models of the payment facilitator 5400 may be applied for other non-payment instruments such as coupons, loyalty cards, and the like.

[00505] In an embodiment of the present invention, the APP 5404 as shown in FIG. 54 may be a mobile application 5404. The mobile application 5404 can include such as a wallet (also referred to as virtual wallet) as one of its components. A consumer can be associated with one or more such wallets. The wallets may facilitate consumers such as during payment processes for the sake of convenience, compactness, and interoperability provided by the wallets. Each of the wallets may be configured to hold a subset of a consumer's transacting instruments. In accordance with various embodiments, the wallet may securely store and protect consumer payment account information, organizing and managing lifecycle of consumer payment accounts, implement strong authentication as required to access the accounts, provide options for selection of a specific account for a transaction, execute the transaction with the POS 5402 or payment gateway with the selected account, implement and enforce business rules governing access to the accounts and transactions using these accounts. In accordance with several embodiments, several forms and implementations of the virtual wallets may be possible depending upon various parameters such as including without limitations an entity hosting the wallet, a channel through which a consumer transacts and location where payment credentials are stored, and the like.

[00506] In an exemplary scenario, a consumer or a user can create a wallet such as on a shopping store or website for example "merchant.com" and add payment cards using such as a web portal. In an example, such a wallet may be referred to as an on-line wallet that may provide convenience during such as purchase of products only from "merchant.com". It must be appreciated that the payment facilitator 5400 may facilitate the consumers to add their pre-existing payment accounts to the virtual wallets by themselves. However, in some other cases such as merchant issued cards (e.g. gift cards), the payment accounts may be added automatically.

[00507] In some examples, an entity such as a bank or any other financial institution (FI) may establish their own versions of the virtual wallet. In such cases, the entity may take on a significant level of risk and indemnify the merchants from fraud since they may provide wallet functionality such as holding an account, authenticating a user, card selection, and the like, and are issues merely for payment account. In such cases, the wallets may thus be limited to self-issued instruments.

[00508] The wallets and each payment type of the wallets may be configured to comply with such as defined business rules, interchange fees, PCI compliance, and the like such as by mapping various standards and payment pricing models across from plastic to the virtual wallets. In an embodiment, the wallets may be implemented in such as smart phones such as implemented within or using capabilities of mobile handsets or other such handheld devices. This may facilitate extension of the virtual wallets from an on-line deployment to the physical POS 5402.

[00509] In accordance with the embodiments of the present invention, the wallet may be a component or a subsystem of the mobile application 5404 or the container as described above. The mobile application 5404 (as shown in FIG. 54) may include several services and functions such as the wallet without limitations. For example, a mobile application 5404 with the wallet as one of the several mobile services, one or more additional services or functions may be provided such as to create a motivation within a consumer to purchase. Therefore, while the additional services may promote motivation to purchase, the wallet may provide the functionality required for the purchase to be completed or executed such that the focus of the wallet may be on purchases executed at the POS 5402 such as a physical POS 5402.

[00510] FIG. 55 shows various components of the wallet, in an exemplary embodiment of the present invention. The wallet of FIG. 55 can be termed as a software wallet 5500. In accordance with various embodiments and depending on the design of the payment facilitator and the service delivery platform ecosystem 5502, the various components of the wallet may be variously located in any suitable combination of the mobile phone or any other handheld device 5504, SDP 5508, or the issuer system 5510. Therefore, the wallet 5500 can be construed and viewed as a distributed component or a combination of distributed components that can run across multiple systems or subsystems such as

shown in FIG. 55. As shown in FIG. 55, the wallet may include a vault 5512, a reference store 5514, a wallet controller 5518, and a wallet view 5520.

[00511] The vault 5512 may be a secured storage associated with each individual payment credential. In an embodiment, the vault 5512 may be consolidated on a secure element. In another embodiment, the vault 5512 may be distributed across several systems or subsystems such that each subsystem or system such as an issuer may retain control of the payment credential and release of such as single use tokens upon request. In an embodiment, the service delivery platform ecosystem may employ the distributed model, however it must be appreciated that the use the consolidated model may be possible and within the purview of the invention.

[00512] The reference store 5514 may serve as a repository of reference data relating to such as the contents of the wallet. The reference data may include such as but is not limited to artwork and branding, issuer information, routing information (and reference identifiers), masked account data, meta-data, and the like. In an embodiment, a subset of these data elements may be stored on such as the mobile phone such as within a memory of the mobile phone while the complete set or data may be stored on a server associated with the SDP 5508 referred to as SDP server (as discussed later in detail).

[00513] The wallet controller 5518 may be configured to implement such as a wallet specific business logic and lifecycle. The wallet controller 5518 may be configured to control and manage the wallet and provide interfaces such APIs to external systems that may access the wallet. Additionally, the wallet controller 5518 may be configured to implement protocols such as to communicate with the POS. For example, the wallet controller 5518 may reside on the SDP server in a typical deployment of the service delivery platform ecosystem of the present invention.

[00514] The wallet view 5520 may be a user interface associated with the wallet such as to facilitate access, interaction, population, and management of the wallets by respective users through means provided by the wallet view 5520. The user interface provided by the wallet view 5520 may be created separately and preferably consistently for each channel such as including but is not limited a mobile phone, web, IPTV, and the like that may be used to access the wallet. In an embodiment of the present invention, the wallet view 5520 may be included in a container application designed for and associated with

the service delivery platform ecosystem. In an embodiment, the wallet view 5520 may be included in a web interface for the consumers linked through the service delivery platform ecosystem.

[00515] In an exemplary embodiment, the payment facilitator may be deployed in a mobile wallet. The wallet may be deployed such as in mobile payment-based systems such as the service delivery platform ecosystem 5502 or the payment facilitator. The wallet 5500 may be configured so as to facilitate the consumers to choose a certified application to access the wallet such as the wallet associated with the payment facilitator from the cloud to perform payments. In an embodiment, these payments may be made at the POS. In another embodiment, these payments may be made at a remote locate location.

[00516] The wallet architecture may include such as the wallet view 5520, wallet controller 5518, wallet vault 5512, application, reference storage 5514, service delivery platform 5508, a plurality of the issuer systems such as the issuer system 5510-5514. The service delivery platform 5508 may further include a master wallet, and several application wallets such as a first application wallet, a second application wallet, and the like. The application may include such as a first application 1, a second application, and an application for the payment facilitator.

[00517] In an embodiment, the SDP 5508 architecture may include several SDPs 5508 and SDP wallets with such as a hierarchical parent child relationship among them. The SDPs 5508 may be configured to maintain a parent-child hierarchy with multiple wallets. Each of the applications such as mobile applications may be linked to an independent wallet instance. The SDPs 5508 may be configured to maintain a parent-child relationship between such as a parent payment facilitator wallet and child application wallets corresponding to the respective parent wallets. In such a scenario, the architecture may be designed so as to define a single SDP 5508 and single wallet view 5520 to a consumer such that the consumer perceives as if interacting with a single wallet and SDP 5508 that is used from within different applications or different wallet views 5520.

[00518] In a hierarchical architecture as discussed above, all instruments may be provisioned to the parent payment facilitator wallet i.e. to the reference store 5514 of the

parent wallet of the payment facilitator. The actual wallet vault 5512 may be configured to remain with and under control of the issuer systems 5510. However, in an embodiment, non-sensitive information may be provisioned to the SDP 5508. The child wallets may be configured to have their own reference stores 5514 such as to enable a filtered view on the reference store 5514 of the parent wallet of the payment facilitator.

[00519] In embodiments, various wallet lifecycle rules such as PIN length, allowed PIN retried, and the like may be configurable independently for each wallet within guidelines such as set by the payment facilitator. The mobile application may request a token from the SDP 5508 at the time of payment. The SDP 5508 may be configured to use an enterprise service bus (ESB) to retrieve a token for the payment account. The token may be a one-time token such as generated by the issuer system. The request for token retrieval and generation may be associated with a strong authentication process by the SDP 5508 and an optional additional authentication available to the issuer system. The token may then be used at the POS through one of the available channels. In an embodiment, the individual mobile applications may communicate consumer services from the SDP 5508 through standard web services in order to access the corresponding wallet.

[00520] In accordance with some exemplary and non-limiting embodiments of the wallet architecture, the issuers may be involved in safeguarding payment credentials and issuing the tokens. The issuer may be allowed to control format of the token and may include additional security measures as appropriate and relevant. Further, the issuers may have minimal impact on chargeback and other exception scenarios as the issuer may be directly involved in provisioning and payment flows. The wallet architecture may also provide wallet accessibility from multiple mobile applications such as from the multiple wallet views 5520. Also, during authentication, each wallet may be authenticated independently with such as independent passcodes and may be managed or activated or locked or suspended and the like independently while still providing a user with a consistent user experience. The wallet architecture may also enable a configurable rule-based filtering for filtering the reference stores 5514 for each child wallet thereby allowing a merchant to have an allowed subset of all the user's payment cards. Also, the architecture may facilitate default accounts to be easily set as per the

respective wallets separately. The wallet architecture may also enable automatic population of the consumer's branded container of the payment facilitator with payment credentials in the consumer's master wallet. This may facilitate management of all accounts from a single location. Further, through this single location-based management, the users can also identify their other applications that are used to access their wallet of the payment facilitator.

[00521] The SDP 5508 as discussed and referred above, and shown in FIGS. 54 may be configured as a system that may be configured to package and deliver services to the consumers through several channels. The SDP 5508 may facilitate an organized, easy to consume manner packaging and delivery of the services through the multiple channels. The SDP 5508 may be enabled such as in a mobile payment domain. The SDP 5508 may be configured to perform such as enrolment, know-your-customer (KYC), and creation of a user wallet instance. The SDP 5508 may be configured to manage the lifecycle of the wallet along with linkages to various end point devices. The SDP 5508 may be configured to such as provide standardized interface to consumers for service discovery and selection. The SDP 5508 may be configured to such as offer issuer governed enrollment and provisioning of services in a uniform manner. The SDP 5508 may be configured to such as implement strong authentication, security and trust models around service provisioning and consumption. The SDP 5508 may be configured to maintain detailed audit trails, logs, and transaction records for such as the services and wallet events, and the like. The SDP 5508 may be configured to onboard individual service providers, aggregate disparate service hosts, enforce service implementation standards, and manage overall growth of the ecosystem, and perform several other functions. The functions and configurations of the SDP 5508 discussed here are merely exemplary without limitations. In still some embodiments, the SDP 5508 may be configured to perform several other functions and tasks based on such as the nature of the services delivered by the SDP 5508. For example, the services related to the financial transactions may have extra mandatory requirements for security and reliability such that the SDP 5508 may be configured to enhance the security related tasks while delivering the financial services. Furthermore, the SDP 5508 may be configured to further enhance system scalability and redundancy while delivering services to public in large.

[00522] The SDP 5508 may be designed to include various components specifically for the payment facilitator in an exemplary embodiment of the SDP 5508 ecosystem deployment based on the MTP methods and systems described herein including wallets, containers, MTP servers, service providers and the like. In an example, there may be several components working together to fulfill the SDP server functions. The components may all be deployed in a highly secure, highly scalable, high availability enterprise server configuration. In an example, the ESB may standardize and interconnect multiple partners and systems from which services may be consumed and delivered to the mobile container. Each partner service provider may be on-boarded through a series of steps that are modeled in the SDP 5508. These steps may be defined by the payment facilitator and be governed by the nature of the service offering (e.g. payment, offers, loyalty, etc.).

[00523] Various exemplary components of the payment facilitator SDP 5508 may be described herein. In an example, the payment facilitator SDP 5508 may include a communication component. The communication component may deal with the communication protocols, codec, serialization, enveloping, and complexities of packet switched communication with the mobile application over multiple data channels. In an example, the payment facilitator SDP 5508 may include a message bus & notification component. The message bus & notification component may support produce / consumer based model for object exchange, along with message structures, with the mobile client. In an example, the message bus & notification component may also facilitate push notifications, alerts, etc. allowing the SDP 5508 to proactively reach the container application to perform some activity. In an example, the payment facilitator SDP 5508 may include a security/authentication component. The security component may facilitate implementation of the security related designs for strong authentication, data privacy, confidentiality / privacy, data integrity, and the like as they relate to the overall trust models and risk management of the services being delivered. In an example, the payment facilitator SDP 5508 may include an over-the-air (OTA) provisioning component. The OTA provisioning may be configured for dealing with the over-the-air provisioning and delivery of the application, widgets, instruments (payment, offers, loyalty, tickets, and the like) and proxy services for tokenization of the instruments. In

an example, the payment facilitator SDP 5508 may include an OTA TSM proxy component. The OTA trusted service manager (TSM) proxy component may be required only in a secure element (such as illustrated in FIG. 55) based ecosystems for over the air provisioning of applets and managing the secure element security domains. In an example, the OTA TSM proxy may typically interface with an external third party TSM. In an example, the payment facilitator SDP 5508 may include a wallet and widget management component. The wallet and widget management may create and manage the lifecycle of a consumer's wallet, association of the wallet with one or more devices, and personalizing the application with one or more widgets and the lifecycle of those widgets. In an example, the payment facilitator SDP 5508 may include a transaction engine. The transaction engine may be configured for dealing with transactional reliability across the wireless channel and maintaining a complete audit trail of the services being delivered. The transaction engine may facilitate receipt and other associated transactional services. In an example, the payment facilitator SDP 5508 may include an instrument management component. The instrument management component may be configured for managing the lifecycle of individual instruments that a consumer populates in their wallet. In an example, the payment facilitator SDP 5508 may include a location and geo-fencing component. The location and geo-fencing component may assist the service providers in setting up geo-fences and opt in location-based services. In an example, the location and geo-fencing component may generate triggers for the service providers when consumers enter the geo fences based on which targeted services can be delivered. In an example, the payment facilitator SDP 5508 may include an augmented reality component. The augmented reality component may provide unique and advanced user experience infrastructure that may be incorporated within the wallet and widgets.

[00524] In an exemplary embodiment of the SDP 5502 ecosystem deployment based on the MTP methods and systems described herein the message definitions may be created for the ESB that may enforce standards and protocols that may be created for the payment facilitator. The service providers may then implement the ESB connectors such as to expose their services to the platform in a way that the various stakeholders of the payment facilitator infrastructure may consume the services.

[00525] In an exemplary embodiment of the SDP 5502 ecosystem deployment based on the MTP methods and systems described herein, the service provisioning from the SDP 5508 may a set of basic steps. In some embodiments, some steps may be optional depending upon the nature of the service and the preference of the service provider. In an example, a first step may include registration and KYC. At this step, the consumer may request a service (wallet, payment account, etc.) and provide the required personal information in order to perform relevant KYC steps. Further to registration and KYC, a service approval may be required for further processing. The service approval may require a service partner to approve the service request after following relevant internal business processes. The service approval may create relevant credentials for the consumer and any service payloads. In some embodiments, the service approval may be optional. A next step may be service provisioning. At this step a service may be provisioned to the user. The service provisioning may include delivering components of the service to a consumer phone or creating necessary updates on a consumer server. The service provisioning may be followed by service activation. The service activation may be required to confirm that the service is provisioned correctly, to the correct user, and all states are set to "READY" for the service to be consumed. In some embodiments, the service activation may be optional. A next step may be service execution. At this step, the consumer may start using the service from within their application. A next step may be service management, which may include all updates and lifecycle components of the service.

[00526] In an exemplary embodiment of the SDP 5502 ecosystem deployment, the SDP 5508 architecture for payment facilitator may allow the service partners to configure and control their service products in a way that they comply with the payment facilitator specifications (through the ESB interfaces), but control their own process, service data, and business rules. Such a control and management may be essential for an open ecosystem that may attract a critical mass of service providers.

[00527] In an exemplary embodiment of the SDP 5502 ecosystem deployment based on the MTP methods and systems described herein including wallets, containers, MTP servers, service providers and the like, there may be a mobile phone stand-in may conduct a transaction at a POS. A high-level mobile payment scenario flows is described

herein. However, it should be noted that these steps are high level and may be re-arranged to suite a specific user experience, and other steps may be inserted in order to provide additional functionality. The purpose of this disclosure document is to demonstrate various deployment options for the payment facilitator and not specific use cases for payment.

[00528] In exemplary and non-limiting embodiments, for affecting the mobile payment a consumer may have to first arrive at checkout terminal and items to be purchased may be scanned. The consumer may then launch a mobile application on the mobile phone and may be required to enter login credential (for example, PIN, password, or passphrase and the like). The mobile application may connect to the SDP 5508 to validate login and authenticate consumer and device. The mobile application may present a list of payment options that may be stored in the consumer's mobile wallet upon successful authentication of the service and device. The consumer may be required to choose a payment option (for example, a gift card) from the list provided to the consumer.

[00529] After the payment option may be chosen the mobile application may initiate a checkout process with the POS. The payment option may be initiated via multiple implementation options depending upon the channels that may be supported by the POS and the mobile phone. In an example, a tokenized NFC may be used for initiating the checkout process. The POS may be near field communication (NFC) enabled for this option. The mobile application may request the SDP 5508 for a token for the selected payment card. The SDP 5508 may authenticate the mobile application and may route the request to corresponding token generator for a concerned issuer. The consumer may be required to tap the phone at the NFC enabled POS to transmit token. In an example, the NFC may be international organization for standardization (ISO) and the international electrotechnical commission (IEC) standard 18092. In an example, a secure element NFC (SE NFC) may be used for initiating the checkout process. In this case, the consumer may be required to tap the phone at NFC enabled POS with the payment credential stored in the SE on the phone. In an example, the NFC may be international organization for standardization (ISO) and the international electrotechnical commission (IEC) standard 14443. In an example, a QR code scan may be used for initiating the checkout process. The mobile application may request the SDP 5508 for a

token for the selected payment card. The SDP 5508 may authenticate the mobile application and may route the request to corresponding token generator for a concerned issuer. The mobile application may convert token into a 2D bar code that is scanned by the POS. In an example, a virtual reader may be used for initiating the checkout process. The mobile application may capture the merchant, POS, and transaction information by scanning a QR code or tapping an NFC tag. The mobile application may send the data along with selected card OTA to SDP 5508. The SDP 5508 may request token, and send the token along with other information to POS through a back-end process.

[00530] A next step after initiating the checkout process may require the POS to processes the transaction and route the process to the payment facilitator. The payment facilitator may route the transaction to the issuer of the payment card at a next step. The issuer may authorize the transaction and approval may be delivered to the POS at a next step. The POS may confirm the transaction and delivers a receipt. In an example, the receipt may be printed, electronically delivered to the mobile application or e-mailed at a verified email address. The delivery of the receipt may complete the Mpayment process and the consumer may leave the checkout terminal with the items purchased.

[00531] In an exemplary embodiment of the SDP 5502 ecosystem deployment based on the MTP methods and systems described herein including wallets, containers, MTP servers, service providers and the like, various deployment models may be employed by the merchants. The merchants may be at various stages in their individual mobile strategies; and therefore, it may be required to support multiple deployment models for merchants so as to adopt a common payment facilitator. The payment facilitator along with each issuer may approve and certify the entire transaction path from a security and risk management perspective such as to keep the transaction costs low. Therefore, it may be important to have a common platform that may be shared amongst all participating merchants.

[00532] In exemplary and non-limiting embodiments, there can be three models that may be used for deployment. The three models ay co-exist and a merchant may adopt one or more of these models depending upon their individual strategies. In exemplary and non-limiting embodiments, more or less than the three models described herein may

be deployed. In exemplary and non-limiting embodiments, the models may be configured for implementing payments, offers, loyalty offers and the like.

[00533] In an exemplary embodiment of the SDP 5502 ecosystem deployment based on the MTP methods and systems described herein, the deployment model maybe a merchant application with a child payment facilitator wallet. The consumers may be required to pay at a POS using their existing, in-market merchant mobile applications linked to their primary payment facilitator wallet through direct child wallets that are provisioned and managed by the SDP 5508 wallet platform. The child wallets may be independently configured but linked to the master wallet. In an example, the existing merchant mobile applications interface directly with the SDP 5508, through the OTA web service interfaces. The SDP 5508 may provide authenticated, controlled, and filtered access to the consumer's payment facilitator wallet and the merchant application may be used for transaction at the POS. In an example, the SDP 5508 may aggregate multiple payment issuers supporting multiple payment channels and manages both the consumer's parent payment facilitator wallet and child merchant wallet.

[00534] In an exemplary embodiment of the SDP 5502 ecosystem deployment based on the MTP methods and systems described herein, the deployment model maybe a merchant application with a merchant wallet. The merchant may use their independent third party wallet within their mobile application and may access the payment credentials from the master payment facilitator wallet.

[00535] In an exemplary embodiment of the SDP 5502 ecosystem deployment based on the MTP methods and systems described herein, the deployment model maybe based in a payment facilitator container. The consumers may pay using a payment facilitator branded mobile application that may automatically get populated from the SDP 5502 and provide consumers with a uniform view into their parent payment facilitator wallet. The SDP 5502 may aggregate multiple payment issuers supporting multiple payment channels and manage the consumer's payment facilitator wallet. In an example, the payment facilitator mobile application may interact with merchant POS to perform transactions without any additional development by merchants. In an example, the merchants and issuers may optionally create their own widgets to run within the payment facilitator application to create custom experiences.

[00536] FIG. 56 illustrates a high level block diagram of the deployment model 5600 maybe based in a payment facilitator container.

[00537] In accordance with the illustrated embodiment, various subsystems within the payment facilitator may include a point of sale (POS) 5602, application (APP) 5604 such as a mobile application 5604, SDP 5608, several issuer systems such as including the depicted issuer 1 system 5610, issuer 2 system 5612, and issuer 3 system 5614, without limitations. The various subsystems may be connected over a network such as a payment processing network 5618. The APP 5604 may further include a first widget 5620, a second widget 5622, and a wallet 5624.

[00538] The illustrated embodiment may be one example of a payment facilitator payment guideline that may have to be managed across multiple issuer systems such as the issuer systems 5610-5614 and multiple merchants. The consumer may not be aware of complex back end operations of mobile transactions in order to create a simple experience of "Select card and Pay". The payment facilitator brand may be created and marketed through a payment facilitator wallet container application 5604. In an embodiment, the merchants may promote the payment facilitator standard as a payment acceptance brand. Ina n example, multiple issuers 5610-5614 may be supported with diverse payment products and multiple POS 5602 channels giving consumers a choice of payment options that may fit their needs across multiple merchants. The payment facilitator container may require a minimal effort on the part of merchants to accept the payment facilitator payment options. The POS 5602 side-work may be required as widgets may be optional. The payment facilitator container may ensure payment processing in compliance to payment facilitator guidelines and standards for POS 5602 based transaction.

[00539] The payment facilitator container model may leverage the full power of the SDP the back and front-end aggregation of multiple service providers. As illustrated in FIG. 56 a payment facilitator branded mobile application 5604 (container) may be built using the MTP wallet toolkit.

[00540] The container may include a wallet implementation that may incorporate the payment facilitator specifications. The wallet may get automatically populated from the SDP as consumers create their payment facilitator wallet. In an embodiment, the

payment facilitator container may be used to make payments at a payment facilitator compliant POS 5602 without any further development by the merchants. In an embodiment, the merchants may have an ability to create widgets using the MTP toolkit that may be specifically designed to provide a merchant centric user experience at the POS 5602. The widget creation may be at the merchant's discretion and may be implemented or added at any time. The SDP may aggregate and allow for the on boarding of multiple issuers 5610-5614, and may allow the merchants to define their payment facilitator compatible payment products and other value added services (VAS). During such on boarding, the issuers 5610-5614 may select the POS 5602 interaction protocols that may be supported for each payment product. The POS 5602 interaction protocols may determine which merchants may accept which payment products as at least one of the issuer-selected protocols may have to be supported by the merchant's POS 5602 channels, for example, NFC, Bar Code, OTA, and the like.

[00541] In some exemplary embodiments, the SDP may be configured to manage a 3-dimensional matrix. The matrix may be configured for merchant's POS 5602 supported proximity channels, for example, NFC, 2D Bar Code, OTA, and the like. The matrix may be configured for merchant POS 5602 capability. The matrix may be configured for issuer approved channels.

[00542] In an exemplary embodiment of the SDP ecosystem deployment based on the MTP methods and systems described herein, the payment facilitator may be used for mobile payment with a merchant widget. In exemplary and non-limiting embodiments, for affecting the mobile mpayment a consumer may have to first arrive at checkout terminal and items to be purchased may be scanned. The consumer may then launch a payment facilitator mobile application 5604 on the mobile phone and may be required to enter a payment facilitator managed login credential (for example, PIN, password, or passphrase and the like). The consumer may then be required to select a merchant from the dashboard of pay-enabled merchants. The merchant widget may be launched in a "Pay" mode and present a list of payment options accepted at the merchant POS 5602. The list of payment options may be a filtered view on the locally stored options in the consumer's wallet as a part of the payment facilitator container. The consumer may be

required to choose a payment option (for example, a gift card) from the list provided to the consumer.

[00543] The merchant widget may request a transaction payload from the SDP that may be sent to the merchant POS 5602 for the corresponding channel supported by the merchant POS 5602. The SDP may transfer a transaction payload formatted for the merchant POS 5602. In exemplary embodiments, the transaction payload may be in form of coupons, loyalty, and the like. The transaction payload may be attached based on the merchant POS 5602 ability. The merchant widget may initiate a checkout process with POS 5602 upon receipt of the transaction payload. The initiation of the checkout process may be done similar to as described above with reference to example with no merchant widget generation.

[00544] A next step after initiating the checkout process may require the POS 5602 to processes the transaction and route the process to the payment facilitator. The payment facilitator may route the transaction to the issuer of the payment card at a next step. The issuer may authorize the transaction and approval may be delivered to the POS 5602 at a next step. The POS 5602 may confirm the transaction and delivers a receipt. In an example, the receipt may be printed, electronically delivered to the mobile application 5604 or e-mailed at a verified email address. The delivery of the receipt may complete the mpayment process and the consumer may leave the checkout terminal with the items purchased.

[00545] In an exemplary embodiment of the SDP ecosystem deployment based on the MTP methods and systems described herein, a user interface (UI) in the payment facilitator container. The UI concept described herein may be used as reference for an exemplary embodiment only. The UI concept may be subject to changes depending upon user requirement. In an exemplary embodiment, the UI concept may include a wallet controlled mode and a widget controlled mode. The wallet control mode may be configured so that the payment facilitator wallet may control the merchant POS 5602 interaction and payment experience. The wallet control mode may circumvent the need for merchants to create widgets. The widget control mode may be configured so that a merchant may create a widget and the widget may will control the interaction and payment experience at the merchant POS 5602.

[00546] The methods and systems described herein may be deployed in part or in whole through a machine that executes computer software, program codes, and/or instructions on a processor. The processor may be part of a server, client, network infrastructure, mobile computing platform, stationary computing platform, or other computing platform. A processor may be any kind of computational or processing device capable of executing program instructions, codes, binary instructions and the like. The processor may be or include a signal processor, digital processor, embedded processor, microprocessor or any variant such as a co-processor (math co-processor, graphic co-processor, communication co-processor and the like) and the like that may directly or indirectly facilitate execution of program code or program instructions stored thereon. In addition, the processor may enable execution of multiple programs, threads, and codes. The threads may be executed simultaneously to enhance the performance of the processor and to facilitate simultaneous operations of the application. By way of implementation, methods, program codes, program instructions and the like described herein may be implemented in one or more thread. The thread may spawn other threads that may have assigned priorities associated with them; the processor may execute these threads based on priority or any other order based on instructions provided in the program code. The processor may include memory that stores methods, codes, instructions and programs as described herein and elsewhere. The processor may access a storage medium through an interface that may store methods, codes, and instructions as described herein and elsewhere. The storage medium associated with the processor for storing methods, programs, codes, program instructions or other type of instructions capable of being executed by the computing or processing device may include but may not be limited to one or more of a CD-ROM, DVD, memory, hard disk, flash drive, RAM, ROM, cache and the like.

[00547] A processor may include one or more cores that may enhance speed and performance of a multiprocessor. In embodiments, the process may be a dual core processor, quad core processors, other chip-level multiprocessor and the like that combine two or more independent cores (called a die).

[00548] The methods and systems described herein may be deployed in part or in whole through a machine that executes computer software on a server, client, firewall, gateway, hub, router, or other such computer and/or networking hardware. The software program

may be associated with a server that may include a file server, print server, domain server, internet server, intranet server and other variants such as secondary server, host server, distributed server and the like. The server may include one or more of memories, processors, computer readable media, storage media, ports (physical and virtual), communication devices, and interfaces capable of accessing other servers, clients, machines, and devices through a wired or a wireless medium, and the like. The methods, programs or codes as described herein and elsewhere may be executed by the server. In addition, other devices required for execution of methods as described in this application may be considered as a part of the infrastructure associated with the server.

[00549] The server may provide an interface to other devices including, without limitation, clients, other servers, printers, database servers, print servers, file servers, communication servers, distributed servers and the like. Additionally, this coupling and/or connection may facilitate remote execution of program across the network. The networking of some or all of these devices may facilitate parallel processing of a program or method at one or more location without deviating from the scope. In addition, any of the devices attached to the server through an interface may include at least one storage medium capable of storing methods, programs, code and/or instructions. A central repository may provide program instructions to be executed on different devices. In this implementation, the remote repository may act as a storage medium for program code, instructions, and programs.

[00550] The software program may be associated with a client that may include a file client, print client, domain client, internet client, intranet client and other variants such as secondary client, host client, distributed client and the like. The client may include one or more of memories, processors, computer readable media, storage media, ports (physical and virtual), communication devices, and interfaces capable of accessing other clients, servers, machines, and devices through a wired or a wireless medium, and the like. The methods, programs or codes as described herein and elsewhere may be executed by the client. In addition, other devices required for execution of methods as described in this application may be considered as a part of the infrastructure associated with the client.

[00551] The client may provide an interface to other devices including, without limitation, servers, other clients, printers, database servers, print servers, file servers,

communication servers, distributed servers and the like. Additionally, this coupling and/or connection may facilitate remote execution of program across the network. The networking of some or all of these devices may facilitate parallel processing of a program or method at one or more location without deviating from the scope. In addition, any of the devices attached to the client through an interface may include at least one storage medium capable of storing methods, programs, applications, code and/or instructions. A central repository may provide program instructions to be executed on different devices. In this implementation, the remote repository may act as a storage medium for program code, instructions, and programs.

[00552] The methods and systems described herein may be deployed in part or in whole through network infrastructures. The network infrastructure may include elements such as computing devices, servers, routers, hubs, firewalls, clients, personal computers, communication devices, routing devices and other active and passive devices, modules and/or components as known in the art. The computing and/or non-computing device(s) associated with the network infrastructure may include, apart from other components, a storage medium such as flash memory, buffer, stack, RAM, ROM and the like. The processes, methods, program codes, instructions described herein and elsewhere may be executed by one or more of the network infrastructural elements.

[00553] The methods, program codes, and instructions described herein and elsewhere may be implemented on a cellular network having multiple cells. The cellular network may either be frequency division multiple access (FDMA) network or code division multiple access (CDMA) network. The cellular network may include mobile devices, cell sites, base stations, repeaters, antennas, towers, and the like. The cell network may be a GSM, GPRS, 3G, EVDO, mesh, or other networks types.

[00554] The methods, programs codes, and instructions described herein and elsewhere may be implemented on or through mobile devices. The mobile devices may include navigation devices, cell phones, mobile phones, mobile personal digital assistants, laptops, palmtops, netbooks, pagers, electronic books readers, music players and the like. These devices may include, apart from other components, a storage medium such as a flash memory, buffer, RAM, ROM and one or more computing devices. The computing devices associated with mobile devices may be enabled to execute program codes,

methods, and instructions stored thereon. Alternatively, the mobile devices may be configured to execute instructions in collaboration with other devices. The mobile devices may communicate with base stations interfaced with servers and configured to execute program codes. The mobile devices may communicate on a peer to peer network, mesh network, or other communications network. The program code may be stored on the storage medium associated with the server and executed by a computing device embedded within the server. The base station may include a computing device and a storage medium. The storage device may store program codes and instructions executed by the computing devices associated with the base station.

[00555] The computer software, program codes, and/or instructions may be stored and/or accessed on machine readable media that may include: computer components, devices, and recording media that retain digital data used for computing for some interval of time; semiconductor storage known as random access memory (RAM); mass storage typically for more permanent storage, such as optical discs, forms of magnetic storage like hard disks, tapes, drums, cards and other types; processor registers, cache memory, volatile memory, non-volatile memory; optical storage such as CD, DVD; removable media such as flash memory (e.g. USB sticks or keys), floppy disks, magnetic tape, paper tape, punch cards, standalone RAM disks, Zip drives, removable mass storage, off-line, and the like; other computer memory such as dynamic memory, static memory, read/write storage, mutable storage, read only, random access, sequential access, location addressable, file addressable, content addressable, network attached storage, storage area network, bar codes, magnetic ink, and the like.

[00556] The methods and systems described herein may transform physical and/or or intangible items from one state to another. The methods and systems described herein may also transform data representing physical and/or intangible items from one state to another.

[00557] The elements described and depicted herein, including in flow charts and block diagrams throughout the figures, imply logical boundaries between the elements. However, according to software or hardware engineering practices, the depicted elements and the functions thereof may be implemented on machines through computer executable media having a processor capable of executing program instructions stored

thereon as a monolithic software structure, as standalone software modules, or as modules that employ external routines, code, services, and so forth, or any combination of these, and all such implementations may be within the scope of the present disclosure. Examples of such machines may include, but may not be limited to, personal digital assistants, laptops, personal computers, mobile phones, other handheld computing devices, medical equipment, wired or wireless communication devices, transducers, chips, calculators, satellites, tablet PCs, electronic books, gadgets, electronic devices, devices having artificial intelligence, computing devices, networking equipment, servers, routers and the like. Furthermore, the elements depicted in the flow chart and block diagrams or any other logical component may be implemented on a machine capable of executing program instructions. Thus, while the foregoing drawings and descriptions set forth functional aspects of the disclosed systems, no particular arrangement of software for implementing these functional aspects should be inferred from these descriptions unless explicitly stated or otherwise clear from the context. Similarly, it may be appreciated that the various steps identified and described above may be varied, and that the order of steps may be adapted to particular applications of the techniques disclosed herein. All such variations and modifications are intended to fall within the scope of this disclosure. As such, the depiction and/or description of an order for various steps should not be understood to require a particular order of execution for those steps, unless required by a particular application, or explicitly stated or otherwise clear from the context.

[00558] The methods and/or processes described above, and steps thereof, may be realized in hardware, software or any combination of hardware and software suitable for a particular application. The hardware may include a general purpose computer and/or dedicated computing device or specific computing device or particular aspect or component of a specific computing device. The processes may be realized in one or more microprocessors, microcontrollers, embedded microcontrollers, programmable digital signal processors or other programmable device, along with internal and/or external memory. The processes may also, or instead, be embodied in an application specific integrated circuit, a programmable gate array, programmable array logic, or any other device or combination of devices that may be configured to process electronic

signals. It may further be appreciated that one or more of the processes may be realized as a computer executable code capable of being executed on a machine readable medium.

[00559] The computer executable code may be created using a structured programming language such as C, an object oriented programming language such as C++, or any other high-level or low-level programming language (including assembly languages, hardware description languages, and database programming languages and technologies) that may be stored, compiled or interpreted to run on one of the above devices, as well as heterogeneous combinations of processors, processor architectures, or combinations of different hardware and software, or any other machine capable of executing program instructions.

[00560] Thus, in one aspect, each method described above and combinations thereof may be embodied in computer executable code that, when executing on one or more computing devices, performs the steps thereof. In another aspect, the methods may be embodied in systems that perform the steps thereof, and may be distributed across devices in a number of ways, or all of the functionality may be integrated into a dedicated, standalone device or other hardware. In another aspect, the means for performing the steps associated with the processes described above may include any of the hardware and/or software described above. All such permutations and combinations are intended to fall within the scope of the present disclosure.

[00561] While the methods and systems described herein have been disclosed in connection with certain preferred embodiments shown and described in detail, various modifications and improvements thereon may become readily apparent to those skilled in the art. Accordingly, the spirit and scope of the methods and systems described herein is not to be limited by the foregoing examples, but is to be understood in the broadest sense allowable by law.

[00562] All documents referenced herein are hereby incorporated by reference.

CLAIMS

What is claimed is:

1. A platform for secure personalized transactions in a multi-domain ecosystem comprising:

a personalization tier for enabling service provider personalization of one or more ecosystem elements stored on a mobile device;

an enabling tier for facilitating interoperation between the personalization tier and a client device; and

a service tier operating independently of the enabling tier to enable service delivery for a plurality of services.

2. The platform of claim 1, wherein the plurality of services are selected from the group consisting of banking, ticketing and bill payment

3. The platform of claim 1, wherein the one or more ecosystem elements are selected from the group consisting of widgets, wallets, and applets.

4. The platform of claim 1, wherein the platform is operable in a distributed network of ecosystem participants including service providers, instrument issuers, instrument acquirers, and mobile device users.

5. The platform of claim 1, wherein the platform further comprises a plurality of functional modules including one or more module for: communication, notification, security, authentication, over the air operation, and device runtime container to manage wallets and widgets.

6. The platform of claim 1, wherein the platform further comprises a software development environment that includes reference implementations of containers, wallets, and widgets.

7. The platform of claim 1, wherein the platform further comprises a software development environment that includes reference implementations of finance, retail, health, and government domains.

8. The platform of claim 1, wherein the platform is deployed through the use of a native client application.

9. The platform of claim 1, wherein the platform is deployed through the use of a native client application and a set of software development tools.

10. The platform of claim 9, wherein the set of software development tools can be used to develop a set of client functions.

11. The platform of claim 10, wherein the set of client functions can be selected from the group of wallets, widgets, companion applets, and couponing applets.

12. The platform of claim 1, wherein the platform is deployed through the use of a native client application and a set of web services that are exposed by the platform to integrate peripheral systems.

13. The platform of claim 12, wherein the peripheral systems are selected from a group consisting of customer care and user portals.

14. The platform of claim 1, wherein the platform is deployed with tools to develop wallets, widgets, and companion applications.

15. The platform of claim 15, wherein the tools include a wizard-like utility.

16. The platform of claim 15, wherein the tools can be configured to monitor usage and transaction information.

17. The platform of claim 15, wherein the tools can be used to provide intelligent optimization options and solutions.

18. The platform of claim 17, wherein the optimization options and solutions are based on monitored capability and performance criteria.

19. The platform of claim 1, wherein the platform is deployed with a necessary support in an enterprise service delivery environment.

20. The platform of claim 19, wherein the necessary support includes support for scalability, redundancy, inter-operability, service delivery, and message definitions.

21. The platform of claim 1, wherein the platform is deployed through a hosted model for a plurality of providers.

22. The platform of claim 21, wherein the plurality of providers can be selected from the group of service, issue, and acquisition providers.

23. The platform of claim 1, wherein the platform is delivered by implementing rules which will govern the use of services by the ecosystem participants.

24. The platform of claim 23, wherein the rules govern the access of the services by the participants.

25. The platform of claim 23, wherein the rules govern the consumption of the services by the participants.

26. The platform of claim 23, wherein the rules govern how a participant may be removed from the ecosystem.

27. The platform of claim 23, wherein the rules govern how participants may cooperate with each other within the ecosystem.

28. The platform of claim 23, wherein the rules govern how participants may cooperate with entities outside of the ecosystem.

29. The platform of claim 23, wherein the rules govern how participants acquire customers.

30. The platform of claim 23, wherein the rules govern how participants will support customers.

31. The platform of claim 23, wherein the rules govern how participants capture data related to users and transactions.

32. The platform of claim 31, wherein the users and transactions can be static.

33. The platform of claim 31, wherein the users and transactions can be dynamic.

34. The platform of claim 31, wherein the rules govern how participants will use the captured data within the ecosystem.

35. The platform of claim 31, wherein the rules govern how participants will use the captured data outside of the ecosystem.

36. The platform of claim 31, wherein the rules govern how participants will share the captured data within the ecosystem.

37. The platform of claim 31, wherein the rules govern how participants will share the captured data outside of the ecosystem.

38. The platform of claim 31, wherein the rules govern how participants will monetize the captured data within the ecosystem.

39. The platform of claim 31, wherein the rules govern how participants will monetize the captured data outside of the ecosystem.

40. The platform of claim 1, wherein the platform is delivered by leveraging an expert engine to provide customer support to participants.

41. The platform of claim 40, wherein the participants are selected from the group of consumers and providers.

42. The platform of claim 1, wherein the platform is delivered by leveraging an expert engine to provide a simplified user experience to participants.

43. The platform of claim 42, wherein the participants are selected from the group of consumers and providers.

44. The platform of claim 1, wherein the platform is architected using multiple tiers.

45. The platform of claim 44, wherein the multiple tiers are selected from the group of enabling tier, service tier, and personalization tier.

46. The platform of claim 44, wherein each tier is divided into additional tiers.

47. The platform of claim 46, wherein the additional tiers are selected from the group of logical tiers and physical tiers.

48. A method comprising:

analyzing one or more of the set consisting of ecosystem participants, service requirements, customer requirements, regulatory requirements, transaction

security requirements, trust models, and electronic transaction communication channels;

determining, based on the analysis, one or more ecosystem features to configure in a platform comprising at least one of a service tier, an enabling tier, and a personalization tier; and

configuring the one or more ecosystem features in accordance with the determination.

49. The method of claim 48, wherein the one or more ecosystem features are configured to ensure satisfaction of each analyzed requirement within a context of a service provider business model.

50. The method of claim 48, wherein the platform additionally comprises an expert engine tier and an experience framework tier.

51. A method comprising:

in a platform comprising at least one of a service tier, an enabling tier, and a personalization tier, applying an expert engine to analyze one or more mobile device user and transaction environment requirements; and

configuring one or more platform features in accordance with the analysis.

52. The method of claim 51, wherein configuring the one or more ecosystem features includes configuring a user interface based on one or more transaction requirements.

53. The method of claim 51, wherein the expert engine comprises an expert engine tier.

54. The method of claim 53, wherein the expert engine tier interacts with an experience framework to adapt the one or more user requirements.

55. The method of claim 54, wherein applying the expert engine comprises applying the expert engine to analyze a user's habits.

56. The method of claim 51, wherein configuring one or more ecosystem features comprises configuring one or more display features of a mobile device.

57. The method of claim 56, wherein the one or more display features are configured based, at least in part, upon a regional preference associated with a location of the mobile device.

58. The method of claim 56, wherein the one or more display features are configured based, at least in part, upon an identified client base.

59. A platform for enabling multiple types of electronic wallets in an ecosystem, comprising:

a software development kit for developing a provider specific wallet, wherein the software development kit is executable at least in part by a networked server;

a plurality of reference wallets accessible through the software development kit, wherein the plurality of reference wallets comprises a reference wallet for each of a plurality of providers including an instrument issuer, a service provider, and an instrument acquirer; and

a runtime container operable on a client device for receiving different types of provider specific wallets, wherein the container facilitates the distinct operation of each type of provider specific wallet for performing secure transactions between a provider specific wallet and a provider that developed the provider specific wallet.

60. A computer readable medium containing program instructions wherein execution of the program instructions by one or more processors of a computer system causes the one or more processors to carry out the steps of:

configuring one or more widget modules in a wallet container on a mobile device; and

accessing one or more wallet modules in the wallet container via the one or more widget modules.

61. The computer readable medium of claim 60 further comprising authenticating each of the one or more widget modules prior to accessing the one or more wallet modules.

62. The computer readable medium of claim 60, wherein the accessing the one or more wallet modules comprises receiving a communication from a service provider module to a first widget module in the wallet container and enabling authenticated access by the first widget module to the wallet module.

63. The computer readable medium of claim 60, wherein each of the one or more wallet modules is configured and installed in the wallet container over a network by a service provider.

64. The computer readable medium of claim 60, wherein each of the one or more wallet modules is configured and installed in the wallet container over a network by a user.

65. A method comprising:

analyzing at least one requirement; and

configuring a service tier, an enabling tier and a personalization tier based, at least in part, upon the analysis.

66. The method of claim 65, wherein the at least one requirement is selected from the set consisting of business strategies, go-to-market strategies, exit strategies, competitive strategies, product roadmaps, hardware release schedules and feature lists, and software release schedules, feature lists, and application programming interfaces.

67. The method of claim 65, wherein the at least one requirement is analyzed within the context of a service provider business model.

68. The method of claim 65, wherein the enabling tier is configured to include an infrastructure runtime container software built using a specific development environment associated with a mobile device.

69. The method of claim 65, wherein the service tier is configured to include a library of common platform services used to implement a plurality of business use cases.

70. A computer readable medium containing program instructions wherein execution of the program instructions by one or more processors of a computer system causes the one or more processors to carry out the steps of:

configuring one or more widget modules for execution via a wallet container on a mobile device wherein the one or more widget modules comprise a service group of a service provider.

71. The computer readable medium of claim 70, wherein the wallet container provides a runtime environment on the mobile device in which the widget module is interpreted for delivering one or more services of the service group.

72. The computer readable medium of claim 70, wherein the wallet container operates in cooperation with an enabling tier of a multi-tier platform for providing secure personalized transactions to provide the one or more widget modules exclusive access to a workflow of the service provider.

73. The computer readable medium of claim 72, wherein the wallet container interprets the widget module in a runtime environment operating on the mobile device to provide the exclusive access.

74. The computer readable medium of claim 70, wherein the service provider is a widget issuer.

75. The computer readable medium of claim 74, wherein one or more of the widget modules issued by the widget issuer is provided exclusive access to a workflow of the widget issuer.

76. The computer readable medium of claim 75, wherein the wallet container operates in cooperation with an enabling tier of a multi-tier platform for providing secure personalized transactions to provide the exclusive access.

77. The computer readable medium of claim 74, wherein one or more of the widget modules issued by the widget issuer is provided exclusive access to a user interface of the widget issuer.

78. The computer readable medium of claim 77, wherein the wallet container operates in cooperation with an enabling tier of a multi-tier platform for providing secure personalized transactions to provide the exclusive access.

79. The computer readable medium of claim 74, wherein one or more of the widget modules issued by the widget issuer is prevented from accessing a workflow of another widget issuer.

80. The computer readable medium of claim 79, wherein the wallet container operates in cooperation with an enabling tier of a multi-tier platform for providing secure personalized transactions to prevent accessing a workflow of another widget issuer.

81. The computer readable medium of claim 74, wherein one or more of the widget modules issued by the widget issuer is prevented from accessing a user interface of another widget issuer.

82. The computer readable medium of claim 81, wherein the wallet container operates in cooperation with an enabling tier of a multi-tier platform for providing secure personalized transactions to prevent accessing a user interface of another widget issuer.

83. A computer readable medium containing program instructions wherein execution of the program instructions by one or more processors of a computer system causes the one or more processors to carry out the steps of:

configuring one or more widget modules in a wallet container on a mobile device, the wallet container comprising one or more wallet modules;

limiting access by the one or more widget modules to the set of elements consisting of the one or more wallet modules, the mobile device, one or more client applications; and

providing access to the one or more widget modules of at least one of an issuer-specific security domain and widget-specific issuer security domain.

84. A method of managing widget module access, comprising:

providing a container runtime environment on a mobile device;

accessing a mobile-device resident widget module to determine access constraints associated with the widget, wherein the access constraints pertain to electronic wallet access, device resource access, device resident ecosystem applications, network resources accessible by the mobile device, issuer-specific security domains, and widget-specific security domains;

operating at least one of the widget modules via the container to enforce access constraints associated with the widget module; and

facilitating access by the widget module to at least one of issuer-specific security domains and widget-specific security domains associated with the widget module.

85. The method of claim 84, wherein enforcing widget module access constraints includes preventing access to electronic wallet resources on the mobile device.

86. The method of claim 84, wherein enforcing widget module access constraints includes preventing access by a first widget to a wallet associated with a second widget.

87. The method of claim 84, wherein enforcing widget module access constraints includes preventing access by a widget to a wallet on the mobile device based on a state of the wallet.

88. The method of claim 87, wherein the state of the wallet includes an active state during which access by the widget is allowed and an inactive state during which access by the widget is prevented.

89. The method of claim 84, wherein facilitating access to a security domain comprises allowing access to a digital wallet on the mobile device that shares a security domain with the widget module.

90. The method of claim 84, wherein facilitating access to a security domain comprises allowing access to network resources of an issuer that shares a security domain with the widget module.

91. A method comprising:

configuring a widget at least partially by a networked server to provide access to an issuer-specific service group;

receiving in a secure wallet container operating on a client device, a request from the networked server to deploy the configured widget;

determining widget isolation requirements for operating the widget on the client device; and

installing the configured widget based on the isolation requirements on the mobile device through the secure wallet container.

92. A computer readable medium containing program instructions wherein execution of the program instructions by one or more processors of a computer system causes the one or more processors to carry out the steps of:

deploying on a client device a first issuer-specific widget to facilitate a first trust model-based secure personalized transaction; and

deploying on the client device a second issuer-specific widget to facilitate a second trust model-based secure personalized transaction.

93. The computer readable medium of claim 92, wherein the first issuer-specific widget and the second issuer-specific widget are deployed in a wallet container.

94. The computer readable medium of claim 92, wherein at least one of the first trust model and the second trust model is one of a delegated trust model, direct trust model, and brokered trust model.

95. The computer readable medium of claim 92, wherein at least one of the first trust model-based secure personalized transaction and the second trust model-based secure personalized transaction is facilitated based, at least in part, upon a trigger.

95. The computer readable medium of claim 95, wherein the trigger corresponds to a monetary value of at least one of the first trust model-based secure personalized transaction and the second trust model-based secure personalized transaction.

96. A method comprising:

receiving an alert with a multi-domain secure wallet container at a mobile device that an updated version of an existing widget that is deployed on the mobile device is available;

confirming that the update is an update of a existing widget that is deployed on the mobile device;

configuring the wallet container to receive the updated widget;

receiving the updated widget with the wallet container;

dispositioning the existing widget to prevent further use of the widget; and

storing the updated widget on the mobile device.

97. A method comprising:

querying a server with a multi-domain wallet container operating on a mobile device for the existence of an updated widget;

receiving information from the server indicative of the availability of the updated widget;

requesting the updated widget;

receiving the updated widget at the mobile device;

dispositioning an existing widget that the updated widget supersedes to prevent further use of the existing widget; and

storing the updated widget on the mobile device.

98. The method of claim 97, wherein querying the server is performed at predefined time intervals.

99. The method of claim 97, wherein querying the server is performed in response to the execution of a widget on the client device.

100. The method of claim 97 further comprising sending a confirmation of storing the updated widget to the server.

101. A method comprising:

receiving a query from a mobile device for the existence of an updated widget;

transmitting information indicative of the availability of the updated widget to the mobile device;

receiving a request for the updated widget from the mobile device; and

transmitting the updated widget to the mobile device.

102. The method of claim 101, wherein query is received at predefined time intervals.

103. The method of claim 101, wherein query is received in response to the execution of a widget on the client device.

104. The method of claim 101 further comprising receiving a confirmation of storing the updated widget.

105. A method comprising:

configuring at least one mobile transaction platform-specific application programming interface on a mobile device for facilitating access to mobile device secure resources by a wallet container executing on the mobile device;

disposing a plurality of distinct wallet containers in a memory of the mobile device, wherein each wallet container interfaces with mobile device secure resources via the at least one application programming interface; and

wherein the at least one application programming interface enforces wallet container isolation via security criteria of a three layer mobile transaction platform.

106. The method of claim 105, wherein the three layers include a personalization layer, a service layer, and an enabling layer.

107. A wallet container operable on a client device, comprising:

a runtime component for facilitating management of widgets, wallets, security, and access to a secure element;

an API-based interface for accessing business specific services provided by the widgets and for accessing client device specific resources; and

wherein the wallet container facilitates operation of a plurality of distinct wallets and distinct widgets for accessing at least one of the plurality of distinct wallets.

108. A computer readable medium containing program instructions wherein execution of the program instructions by one or more processors of a computer system causes the one or more processors to carry out the steps of:

configuring one or more widget modules in a wallet container on a mobile device the wallet container comprising one or more wallet modules; and

facilitating access to at least one of the wallet modules by an issuer of at least one of the widgets.

109. A method of managing a plurality of transaction protocols, comprising:

disposing a wallet module in a memory of a client device;

disposing a wallet companion applet in the memory;

executing the wallet companion applet in association with a secure personalized transaction being conducted over an ecosystem in which the client device is deployed to manage access to selected content of a secure element associated with the client device; and

wherein the wallet companion applet uses the accessed secure element content to facilitate authenticating access to the wallet module to perform the transaction.

110. A computer readable medium containing program instructions wherein execution of the program instructions by one or more processors of a computer system causes the one or more processors to carry out the steps of:

configuring one or more widget modules in a wallet container on a mobile device the wallet container comprising one or more wallet modules; and

configuring at least one applet to execute in conjunction with at least one wallet module to facilitate management of a plurality of transaction protocols.

111. A method comprising:

disposing a wallet module in a memory of a client device;

disposing a wallet companion applet in a secure element accessible by the client device, wherein the wallet companion applet facilitates access by at least one other applet and at least one wallet module to content that is stored in the secure element via a mobile wallet function sharable interface.

112. The method of claim 111, wherein the at least one other applet performs a function selected from the applet function list consisting of: couponing, security, and payment.

113. A method comprising:

disposing a container for providing a runtime environment for at least one widget and at least one wallet in a memory of a client device, wherein the container provides access to the at least one widget and the at least one wallet through a sharable interface

disposing a container companion applet in a secure element of the client device that, when executed in the runtime environment, facilitates access by at least one other applet to secure wallet resources associated with the at least one wallet.

114. The method of claim 113, wherein the sharable interface facilitates access to mobile wallet functions and mobile widget functions accessible via the container.

115. A platform operable on a client device, comprising:

a runtime component for configuring at least one tokenized widget wherein the at least one tokenized widget facilitates at least one transaction in a multi-domain ecosystem for secure personalized transactions.

116. A platform operable on a client device, comprising:

a runtime component for configuring at least one service provider widget in a multi-domain ecosystem for secure personalized transactions.

117. The platform of claim 116, wherein the service provider widget facilitates access to an electronic wallet associated with the service provider.

118. A platform operable on a client device, comprising:

a mobile transaction platform mobile device runtime component;

at least one mobile transaction platform application programming interface operable by the runtime component, comprising program functions for configuring at least one client container on the mobile device;

wherein the mobile device is a participant in a multi-domain ecosystem for secure personalized transactions.

119. A mobile transaction platform disposed on a mobile device for securing account information for a plurality of accounts in an electronic wallet disposed on the mobile device, comprising:

a wallet container that operates on the mobile device to provide access to the plurality of accounts in a secure manner by interpreting widget applets in a runtime environment and limiting access by the widget applets to accounts in the electronic wallet to which the widget applets are authorized to access.

120. A platform operable on a client device, comprising:

a runtime component for securing account information for a plurality of accounts in a wallet.

121. A wallet container operable on a client device that manages requests by service provider specific widgets to access a mobile wallet to enforce access restrictions to the mobile wallet based on the wallet authentication status of the service provider.

122. A platform operable on a client device, comprising:

a runtime component for providing secure access to a mobile wallet by a plurality of service providers.

123. A mobile wallet for integrating offers with transactions between an external service provide and the mobile wallet in a multi-domain ecosystem for secure personalized transactions, comprising:

a multi-domain digital wallet operable on a mobile device, wherein the digital wallet comprises accounts from at least two distinct service providers;

a runtime container for executing widgets that attempt to access the digital wallet;

a plurality of mobile offer management widgets, operable in the runtime container that interface with distinct mobile offer domains by communicating with external service providers; and

a mobile transaction capability provided by cooperative operation of the runtime container, the multi-domain digital wallet, and at least a portion of the plurality of mobile offer management widgets.

124. A method of abstracting a protocol in a secure personalized transaction via a multi-domain ecosystem, comprising:

receiving an electronic transaction message in a secure personalized transaction comprising a plurality of information fields at a networked device in the multi-domain ecosystem;

determining with the device a type and a content for each of the plurality of information fields;

mapping at least a portion of the plurality of information fields to common format message fields;

configuring a common format transaction message based on the mapping; and

transmitting the common format transaction message from the networked device to a second networked device in the multi-domain ecosystem.

125. A method of abstracting a protocol in a secure personalized transaction via a multi-domain ecosystem, comprising:

receiving an electronic transaction message in a secure personalized transaction protocol comprising a plurality of information fields at a networked device in the multi-domain ecosystem;

determining with the device a type and a content for each of the plurality of information fields;

mapping at least a portion of the plurality of information fields with a meta-model that provides consistent mapping of the secure personalized transaction protocol to common format message fields;

configuring a common format transaction message based on the mapping; and

transmitting the common format transaction message from the networked device to a second networked device in the multi-domain ecosystem.

126. A method of abstracting a protocol in a secure personalized transaction via a multi-domain ecosystem, comprising:

receiving an electronic transaction message in a secure personalized transaction comprising a plurality of information fields at a networked device in the multi-domain ecosystem;

determining with the device a type and a content for each of the plurality of information fields;

mapping at least a portion of the plurality of information fields to common format message fields;

configuring a common format transaction message based on the mapping;

wrapping the common format transaction in a protocol abstraction content wrapper; and

transmitting the wrapped common format transaction message from the networked device to a second networked device in the multi-domain ecosystem.

127. A method of dynamic tokenization of a secure personalized transaction via a multi-domain ecosystem, comprising:

receiving an electronic transaction message in a secure personalized transaction comprising a plurality of non-confidential information fields at a networked device in the multi-domain ecosystem;

configuring a token comprising the non-confidential information and transaction identification information at the networked device;

electronically transmitting the token to a source of the electronic transaction message and to a service provider server disposed in the multi-domain ecosystem; and

transmitting authentication of the token, along with token identification information from the source of the electronic transaction message to the service provider server for authenticating the transaction identified by the token.

128. The method of claim 127, wherein the service provider is an instrument issuer.

129. The method of claim 127, wherein the service provider accesses confidential information that is not accessible to the networked device to facilitate authenticating the transaction identified by the token.

130. The method of claim 129, wherein the confidential information is associated with a digital wallet disposed on a device from which the authentication of the token is transmitted.

131. The method of claim 127, wherein transmitting authentication of the token is facilitated by a provider-specific widget executing on a device from which the electronic transaction message is sourced.

132. A method of token mobile caching, comprising:

generating with a networked server a token for performing an electronic transaction over a public network with a client device;

providing the token to the client device over the public network;

storing the token along with information that uniquely identifies the client to which the token was provided in a memory accessible to the server; and

in response to receiving an electronic message from a client device at the server to conduct an electronic transaction using the token over the public network, comparing the stored information that uniquely identifies the client to which the token was provided with client identification information received with the token to determine if conducting the electronic transaction should be authenticated.

133. The method of claim 132, wherein the electronic message includes information derived from a secure element associated with the client to which the token was provided.

134. A method of tokenizing a wrapper in a secure personalized transaction via a multi-domain ecosystem, comprising:

receiving an electronic transaction message in a secure personalized transaction comprising a plurality of information fields at a networked device in the multi-domain ecosystem;

determining with the device a type and a content for each of the plurality of information fields;

mapping at least a portion of the plurality of information fields to common format message fields;

configuring a common format transaction message based on the mapping;

wrapping the common format transaction in a protocol abstraction content wrapper;

configuring a token comprising the protocol abstraction content wrapper; and

transmitting the wrapped common format transaction message from the networked device to a second networked device in the multi-domain ecosystem.

135. A method of tokenization of a wrapper for a secure personalized transaction via a multi-domain ecosystem, comprising:

receiving a wrapper for an electronic transaction message in a secure personalized transaction at a networked device in the multi-domain ecosystem;

configuring a token comprising the wrapper and transaction identification information at the networked device;

electronically transmitting the token to a source of the wrapper for an electronic transaction message and to a service provider server disposed in the multi-domain ecosystem; and

transmitting authentication of the token, along with token identification information from the source of the wrapper for an electronic transaction message to the service provider server for authenticating the transaction identified by the token.

136. The method of claim 135, wherein the service provider is an instrument issuer.

137. The method of claim 135, wherein the service provider accesses confidential information that is not accessible to the networked device to facilitate authenticating the transaction identified by the token.

138. The method of claim 137, wherein the confidential information is associated with a digital wallet disposed on a device from which the authentication of the token is transmitted.

139. The method of claim 135, wherein transmitting authentication of the token is facilitated by a provider-specific widget executing on a device from which the wrapper for an electronic transaction message is sourced.

140. A method for performing tokenized electronic transactions in a multi-domain ecosystem for secure personalized transactions, comprising:

generating a token representing an electronic transaction message that identifies at least one transaction and at least one ecosystem participant in the transaction;

wrapping the token representing the electronic transaction message, providing a wrapper that includes a token representing an electronic transaction message;

generating a token representing the wrapper, providing a tokenized wrapper for a tokenized representation of an electronic message; and

performing an electronic transaction between two participants in the ecosystem by transmitting the tokenized wrapper between the two participants, wherein one of the two participants in the ecosystem is the at least one ecosystem participant.

141. A mobile transaction processing system, comprising:

a plurality of servers and a plurality of client devices deployed in a network, each server configured to facilitate secure personalized transactions among the plurality of client devices and a plurality of service provider servers in the network;

at least one of the plurality of client devices configured with a multi-tenant capable wallet operable in a container-based execution environment, wherein the wallet is accessible through a widget disposed on the at least one of the plurality of client devices;

the plurality of servers further configured to facilitate transactions between at least one of the plurality of client devices via the widget and at least one of the plurality of service provider servers via any of a plurality of trust models including a single trust domain, a single trust cluster, multiple trust clusters, and a direct trust relationship.

142. The mobile transaction processing system of claim 141, wherein the plurality of distinct trust models includes at least one of a single trust domain, a single trust cluster, multiple trust clusters, and direct trust relationship.

143. The mobile transaction processing system of claim 142, wherein the single trust domain comprises a central wallet service center server that interfaces with an electronic wallet on a user device and facilitates establishing secure transaction channels with trusted service providers.

144. The mobile transaction processing system of claim 142, wherein the single trust cluster comprises a trusted third party server for securely exchanging confidential wallet information with a personal electronic wallet and a wallet service center server for facilitating direct authentication of the user by a each of a plurality of service providers via the third party server, wherein the wallet service center server delivers user identification and security keys to at least one of the plurality of service providers.

145. The mobile transaction processing system of claim 142, wherein multiple trust clusters comprise a plurality of distinct trusted third parties each configured with at least a server for exchanging confidential wallet information with a personal wallet and further for exchanging user and/or device identification provided by a wallet service center with a plurality of cluster-specific service providers.

146. The mobile transaction processing system of claim 142, wherein a direct trust relationship comprises a home wallet service center server for establishing transaction context with an electronic wallet that facilitates establishing an authenticated transaction channel directly between the electronic wallet and the service provider.

147. The mobile transaction processing system of claim 141, wherein the widget is configured for secure interaction with a specific provider.

148. The mobile transaction processing system of claim 141, wherein the widget is configured for secure interaction through an intermediary with a specific provider.

149. The mobile transaction processing system of claim 147, wherein the widget is configured on the mobile device over the air through an electronic interaction between the mobile device and a networked server associated with the specific provider.

150. A secure mobile wallet operative on a mobile device, comprising:

a plurality of distinct user accounts, wherein access to a distinct user account is facilitated by a provider-specific trust model via a widget disposed on the mobile device.

151. The secure mobile wallet of claim 150, wherein the widget is configured for secure interaction with a specific provider.

152. The secure mobile wallet of claim 150, wherein the widget is configured for secure interaction through an intermediary with a specific provider.

153. The secure mobile wallet of claim 151, wherein the widget is configured on the mobile device over the air through an electronic interaction between the mobile device and a networked server associated with the specific provider.

154. A method of trust model determination for a mobile transaction in an ecosystem for secure personalized transactions, comprising:

configuring a plurality of servers deployed in a network to facilitate transactions between a client device and a provider-specific server via any of a plurality of trust models including a single trust domain, a single trust cluster, multiple trust clusters, and a direct trust relationship;

analyzing a default trust model for a transaction between the mobile device and the provider-specific server to determine a measure of risk associated with the transaction; and

based on the resulting measure of risk, conducting the transaction via one of the default trust model and another trust model from the plurality of trust models.

155. The method of claim 154, wherein conducting the transaction includes accessing a wallet function on the mobile device via a provider-specific widget executing on the mobile device.

156. The method of claim 155, wherein the provider-specific widget is selected based on the resulting measure of risk.

157. The method of claim 156, wherein the measure of risk is based on a location of the mobile device.

158. The method of claim 156, wherein the measure of risk is based on a location of the provider-specific server relative to a location of the mobile device.

159. The method of claim 156, wherein the measure of risk is based on aspects of a network over which the mobile device is conducting the transaction.

160. A domain-specific program language for use in an ecosystem involving an instrument issuer, a service provider, an acquirer, and a client, comprising:

a syntax that is compliant with XML, CSS, and Javascript; and

an object that enables a wallet-related function on a client device.

161. The domain-specific program language of claim 160, wherein the enabled wallet-related function is at least one of a wallet container, a wallet, a widget, and a wallet companion applet.

162. A platform for serving secure personalized transactions in a multiple domain ecosystem, comprising:

an interpreter for interpreting domain-specific program language objects for enabling wallet-related functions that are relevant to at least one of the multiple domains.

163. The platform of claim 162, wherein the interpreter is a runtime interpreter.

164. The platform of claim 162, wherein the interpreter is a runtime container executing on a client device.

165. The platform of claim 162, wherein the enabled wallet-related function is at least one of a wallet container, a wallet, a widget, and a wallet companion applet.

166. A set of tools for securing content in a multi-domain ecosystem, comprising:

a software development kit for constructing a mobile transaction ecosystem:

a set of domain-specific functions for enabling client-side functionality of at least one of a widget, a wallet, a container, and a companion applet;

device agnostic application programming interfaces;

device-specific application programming interfaces;

enabling tier functions for facilitating interoperation with client device software and hardware; and

widget publishing modules to facilitate availability of provider-specific widgets for delivery and downloading to a client device.

167. A network-enabled server for configuring a set of service provider tools for a multi-domain ecosystem for secure personalized transactions, comprising:

a mobile transaction platform server;

a wallet service center development environment operable on the platform server;

a wallet service center runtime module operable on the platform server; and

a software development kit for constructing and maintaining server-based and client-based elements of the ecosystem.

168. The network-enabled server of claim 167, wherein the cleint-based elements of the ecosystem comprise wallets, widgets, companion applets, and containers

169. A mobile application for in-situ customizing of a portion of a widget, comprising:

a user interface template for receiving widget functional code and descriptors from a widget development process performed on a networked server;

a user interface for facilitating access to widget features and functions via icons defined by a widget descriptor; and

an algorithm for configuring a widget downloaded to the device on which the user interface is displayed based on user actions detected in the user interface.

170. The mobile application of claim 169, wherein the user interface template is automatically configurable based on the widget descriptor.

171. The mobile application of claim 169, wherein the application for in-situ customizing of a portion of a widget operates on a mobile device.

172. The mobile application of claim 169, wherein the application executes within an mClient container runtime environment.

173. A server-based application, comprising:

a set of reference mobile transaction platform services;

an automated wizard application for accessing the set of reference services and configuring at least one service from the set of reference services based on user input; and

wherein the automated wizard application facilitates configuring at least one of client element distribution, issuer environments, settlement environments, aggregation services, transaction medium support, support for handset types, user experience, business model, and regulatory considerations based on the user input.

174. An ecosystem configuration guidance system, comprising:

a set of reference mobile transaction platform services; and

an automated wizard application for receiving user input associated with a desired ecosystem configuration and recommending a configuration of at least one service from the set of reference services based on the user input.

175. The ecosystem configuration guidance system of claim 174, wherein the set of reference services includes distribution services, continuum use services, issuer services, settlement services, aggregation services, transaction medium requirements, handset and operating system considerations, user experience factors, business modeling, and regulatory considerations.

176. The ecosystem configuration guidance system of claim 174, wherein the user input associated with a desired ecosystem configuration is a hyperlink.

177. The ecosystem configuration guidance system of claim 174, wherein the wizard searches for information based on the hyperlink and populates at least one reference service template.

178. The ecosystem configuration guidance system of claim 176, wherein the hyperlink is a link to a website of a service provider operating the wizard.

179. A method of type-length variable data communication among participants in an multi-domain ecosystem for secure personalized transactions, comprising:

receiving a plain old java object at a producer server that is a participant in the multi-domain ecosystem;

converting the received plain old java object to a type length variable stream that complies with a type length variable protocol; and

transmitting the converted object via a one-way message bus from producer to a mobile device participant in the multi-domain ecosystem.

180. The method of claim 179, wherein the plain old java object conforms to type length variable protocol guidelines.

181. The method of claim 180, wherein the type length variable protocol supports streaming, float support, and prepended runtime class type information.

.

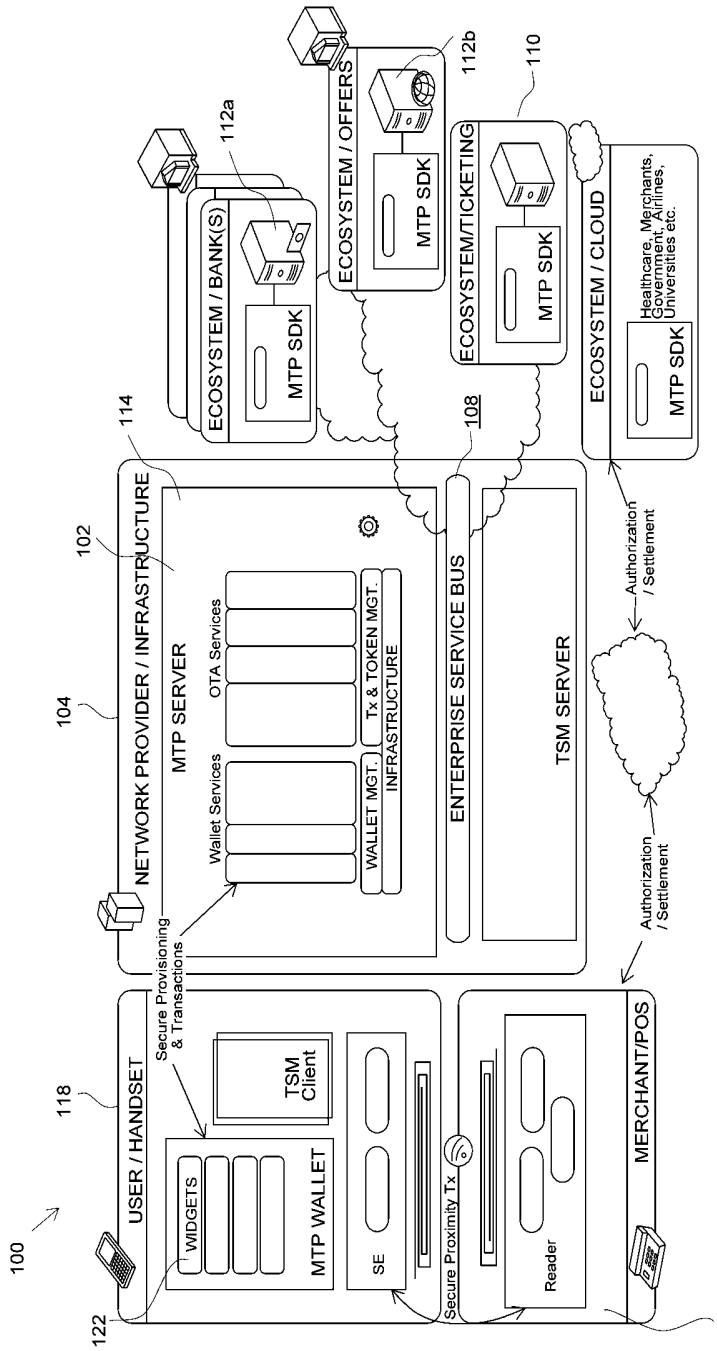

100
USER / HANDSET
118
122
WIDGETS
MTP WALLET
TSM Client
SE
Secure Provisioning & Transactions
Secure Proximity Tx
Reader
MERCHANT/POS
120
NETWORK PROVIDER / INFRASTRUCTURE
102
104
MTP SERVER
Wallet Services
OTA Services
WALLET MGT.
Tx & TOKEN MGT.
INFRASTRUCTURE
114
ENTERPRISE SERVICE BUS
108
TSM SERVER
Authorization / Settlement
Authorization / Settlement
ECOSYSTEM / BANK(S)
MTP SDK
112a
ECOSYSTEM / OFFERS
MTP SDK
112b
ECOSYSTEM/TICKETING
MTP SDK
110
ECOSYSTEM / CLOUD
Healthcare, Merchants, Government, Airlines, Universities etc.
MTP SDK

FIG. 1

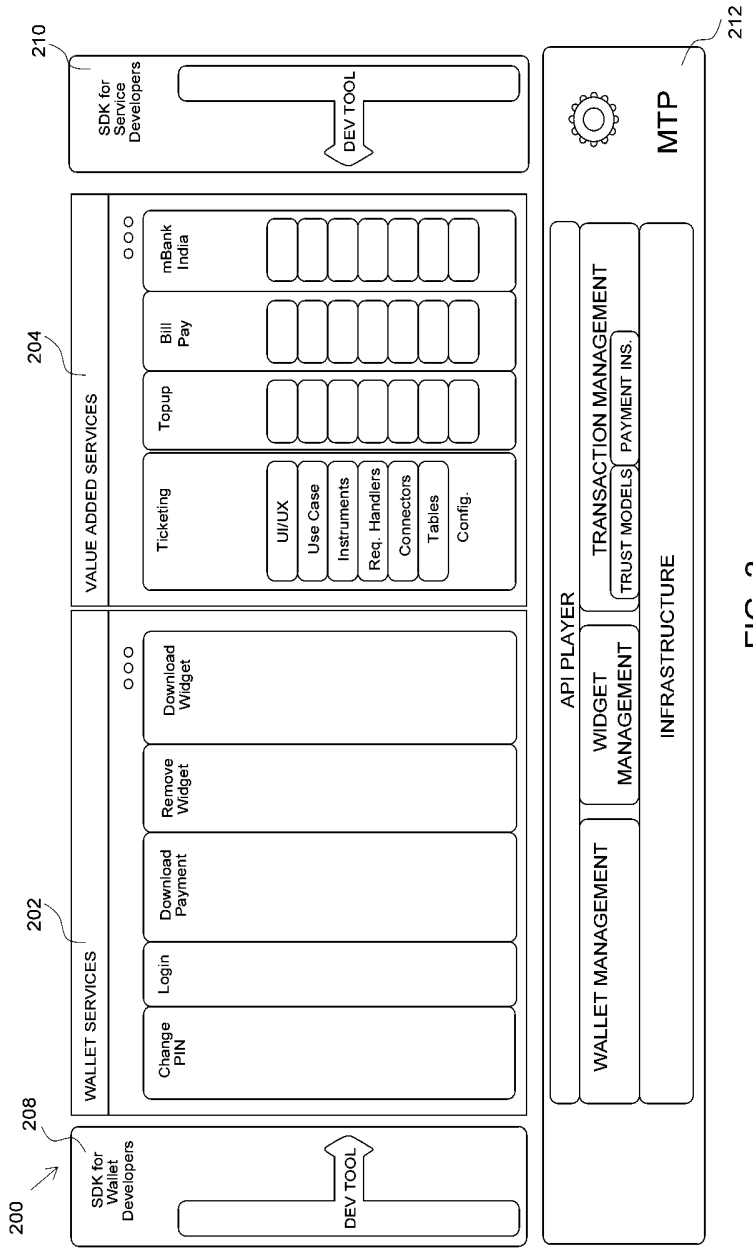

200
208
SDK for Wallet Developers
DEV TOOL
202
WALLET SERVICES
Change PIN
Login
Download Payment
Remove Widget
Download Widget
204
VALUE ADDED SERVICES
Ticketing
UI/UX
Use Case
Instruments
Req. Handlers
Connectors
Tables
Config.
Topup
Bill Pay
mBank India
210
SDK for Service Developers
DEV TOOL
212
MTP
API PLAYER
WALLET MANAGEMENT
WIDGET MANAGEMENT
TRANSACTION MANAGEMENT
TRUST MODELS
PAYMENT INS.
INFRASTRUCTURE

FIG. 2

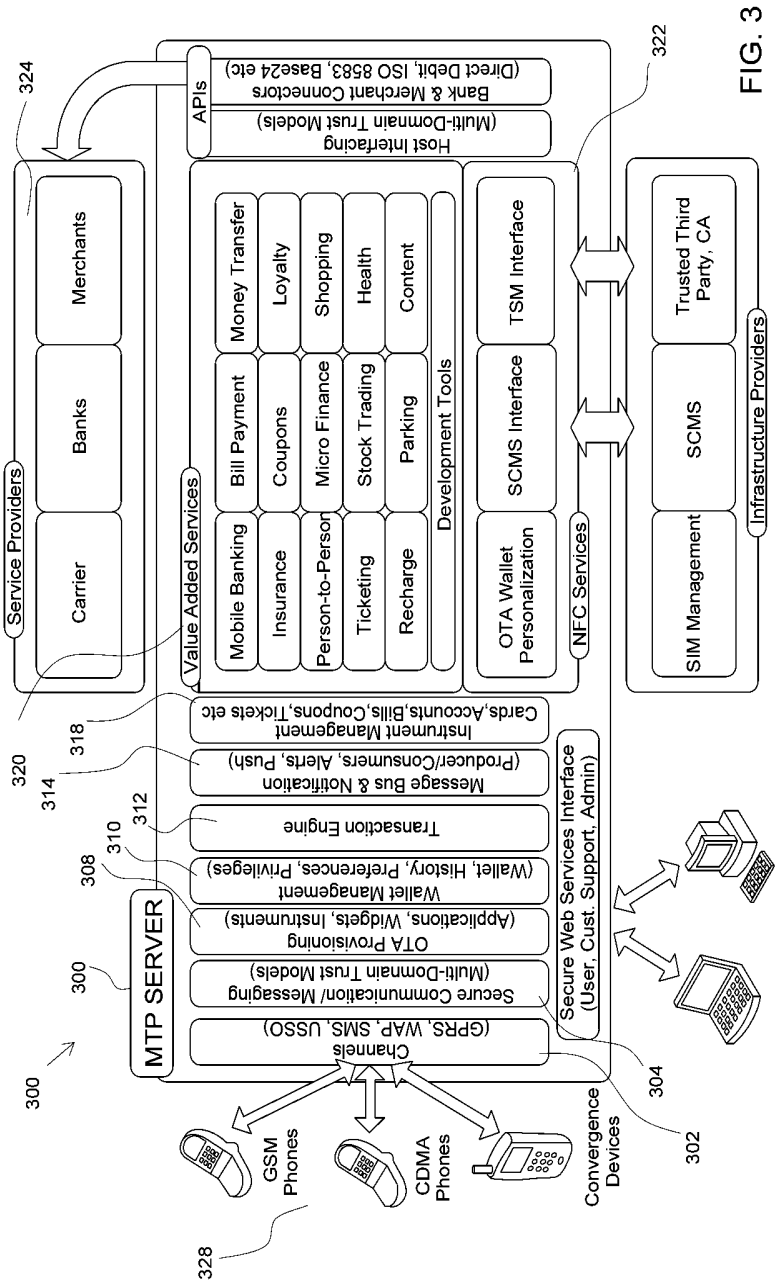

300
MTP SERVER
Channels
(GPRS, WAP, SMS, USSO)
Secure Communication/ Messaging
(Multi-Domain Trust Models)
OTA Provisioning
(Applications, Widgets, Instruments)
Wallet Management
(Wallet, History, Preferences, Privileges)
Transaction Engine
Message Bus & Notification
(Producer/Consumers, Alerts, Push)
Instrument Management
Cards,Accounts,Bills,Coupons,Tickets etc
Value Added Services
Mobile Banking
Insurance
Person-to-Person
Ticketing
Recharge
Bill Payment
Coupons
Micro Finance
Stock Trading
Parking
Money Transfer
Loyalty
Shopping
Health
Content
Development Tools
APIs
Host Interfacing
(Multi-Domain Trust Models)
Bank & Merchant Connectors
(Direct Debit, ISO 8583, Base24 etc)
Service Providers
Carrier
Banks
Merchants
OTA Wallet Personalization
SCMS Interface
TSM Interface
NFC Services
SIM Management
SCMS
Trusted Third Party, CA
Infrastructure Providers
Secure Web Services Interface
(User, Cust. Support, Admin)
Convergence Devices
GSM Phones
CDMA Phones
302
304
308
310
312
314
318
320
322
324
328

FIG. 3

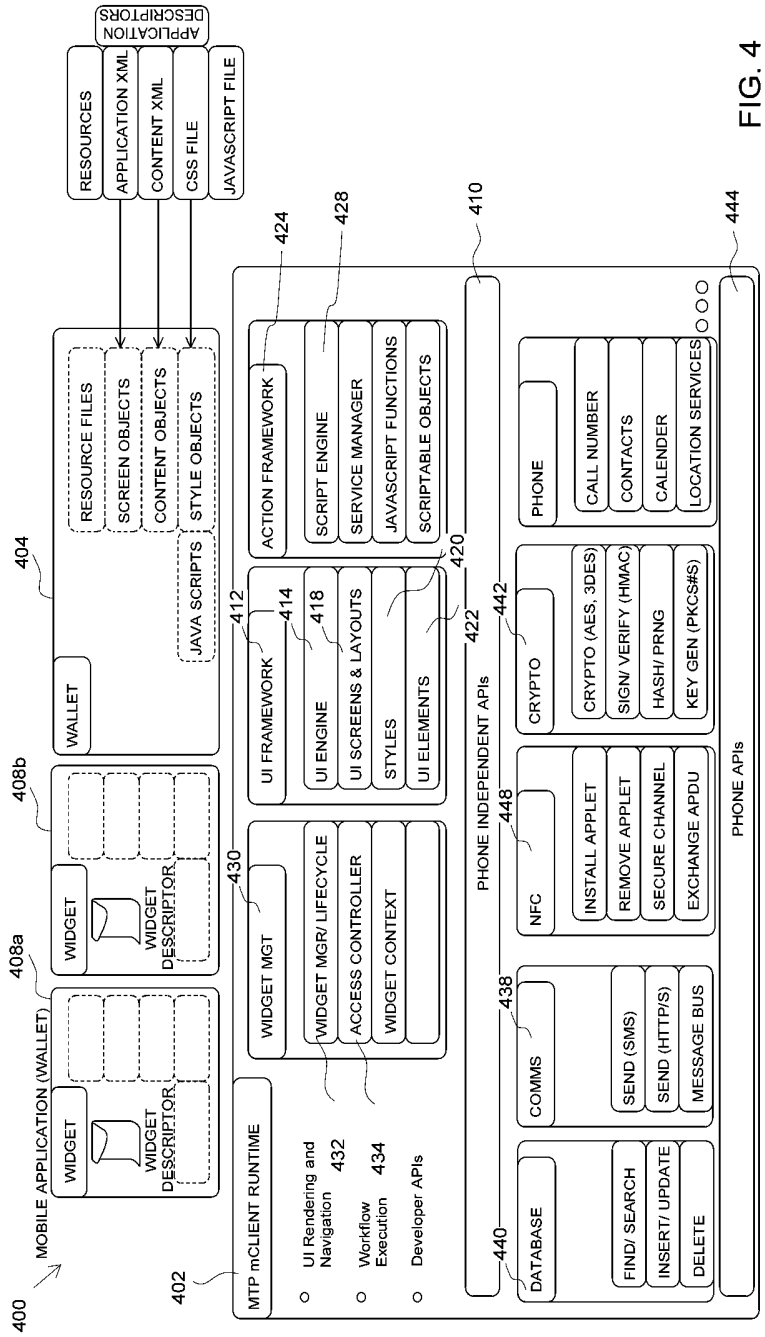

400
MOBILE APPLICATION (WALLET)
402
MTP mCLIENT RUNTIME
UI Rendering and Navigation 432
Workflow Execution 434
Developer APIs
408a
WIDGET
WIDGET DESCRIPTOR
408b
WIDGET
WIDGET DESCRIPTOR
404
WALLET
RESOURCE FILES
SCREEN OBJECTS
CONTENT OBJECTS
STYLE OBJECTS
JAVA SCRIPTS
RESOURCES
APPLICATION XML
CONTENT XML
CSS FILE
JAVASCRIPT FILE
APPLICATION DESCRIPTORS
430
WIDGET MGT
WIDGET MGR/ LIFECYCLE
ACCESS CONTROLLER
WIDGET CONTEXT
412
UI FRAMEWORK
414
UI ENGINE
418
UI SCREENS & LAYOUTS
STYLES
UI ELEMENTS
420
424
ACTION FRAMEWORK
428
SCRIPT ENGINE
SERVICE MANAGER
JAVASCRIPT FUNCTIONS
SCRIPTABLE OBJECTS
410
PHONE INDEPENDENT APIs
422
440
DATABASE
FIND/ SEARCH
INSERT/ UPDATE
DELETE
438
COMMS
SEND (SMS)
SEND (HTTP/S)
MESSAGE BUS
448
NFC
INSTALL APPLET
REMOVE APPLET
SECURE CHANNEL
EXCHANGE APDU
442
CRYPTO
CRYPTO (AES, 3DES)
SIGN/ VERIFY (HMAC)
HASH/ PRNG
KEY GEN (PKCS#S)
PHONE
CALL NUMBER
CONTACTS
CALENDER
LOCATION SERVICES
444
PHONE APIs

FIG. 4

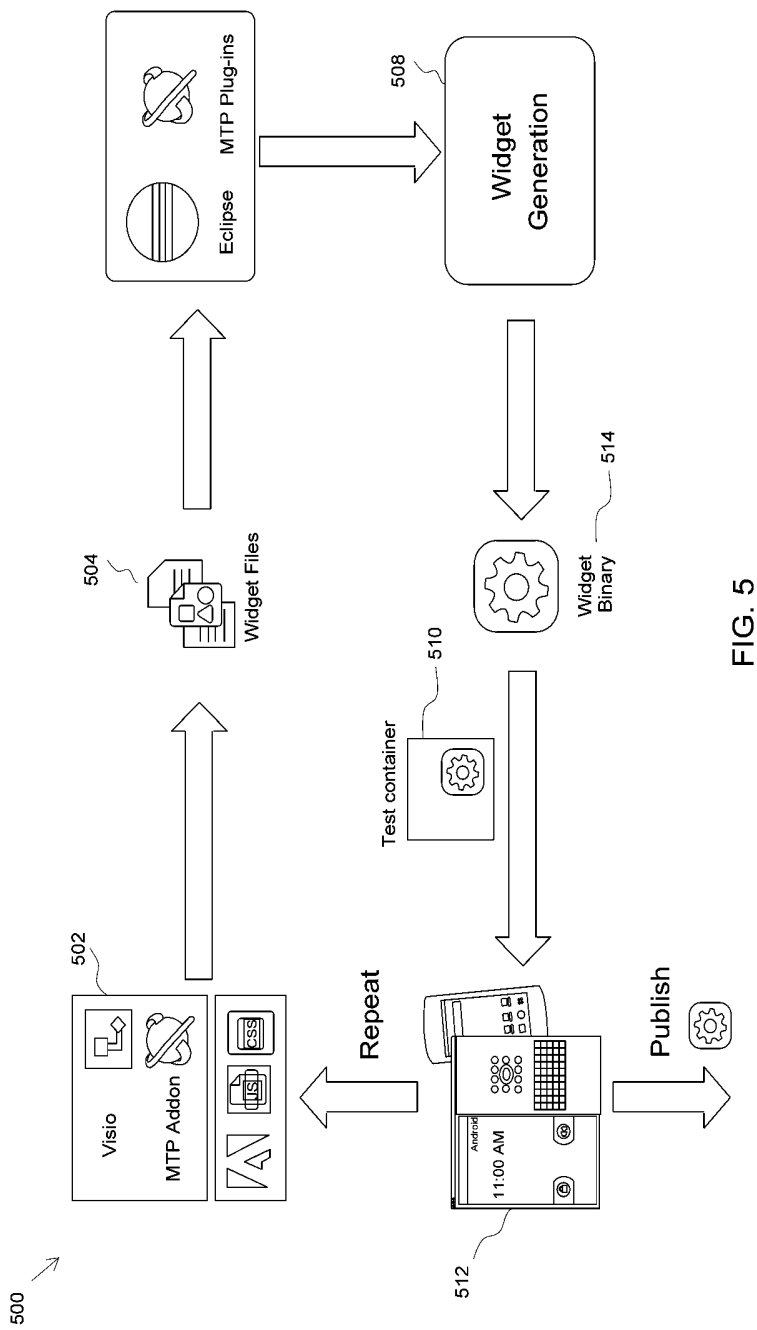

500
502
Visio
MTP Addon
CSS
JS
Repeat
504
Widget Files
Eclipse
MTP Plug-ins
508
Widget Generation
510
Test container
514
Widget Binary
512
Android
11:00 AM
Publish

FIG. 5

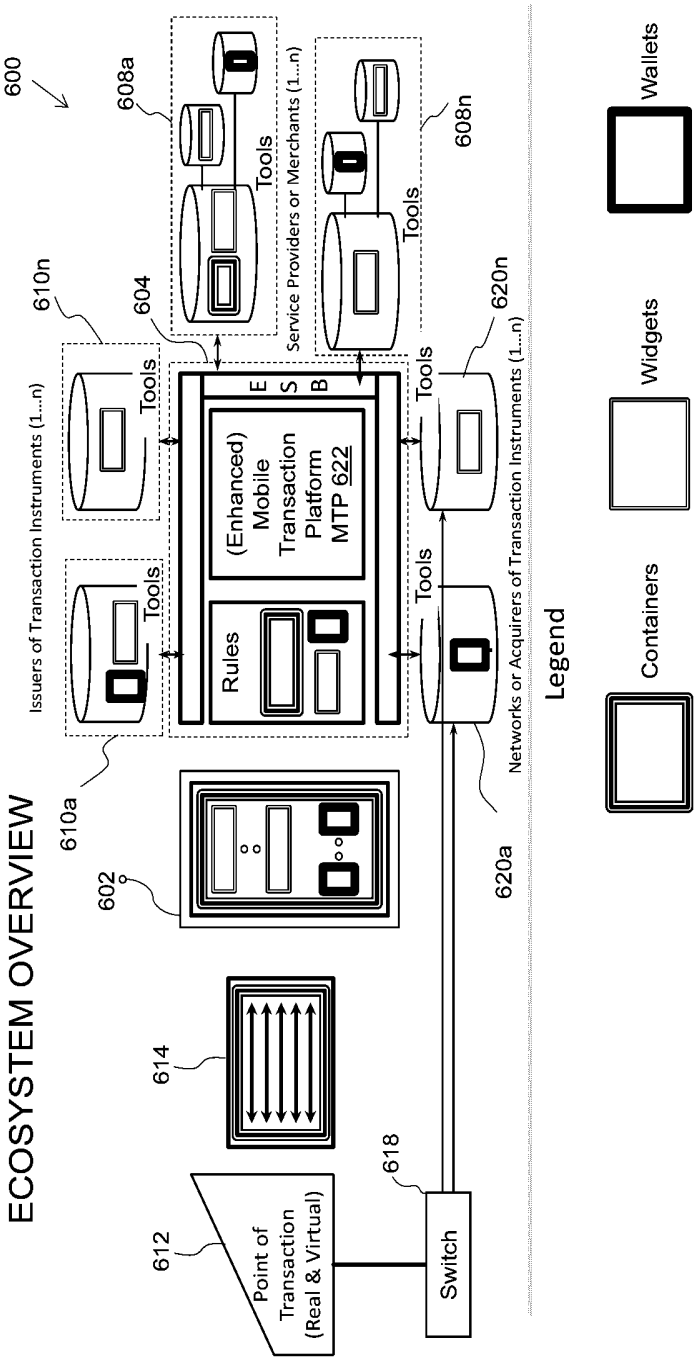

ECOSYSTEM OVERVIEW
600
608a
608n
610n
610a
604
602
614
612
618
620n
620a
Issuers of Transaction Instruments (1...n)
Service Providers or Merchants (1...n)
Networks or Acquirers of Transaction Instruments (1...n)
Tools
Rules
(Enhanced) Mobile Transaction Platform MTP 622
E S B
Point of Transaction (Real & Virtual)
Switch
Legend
Containers
Widgets
Wallets

FIG. 6

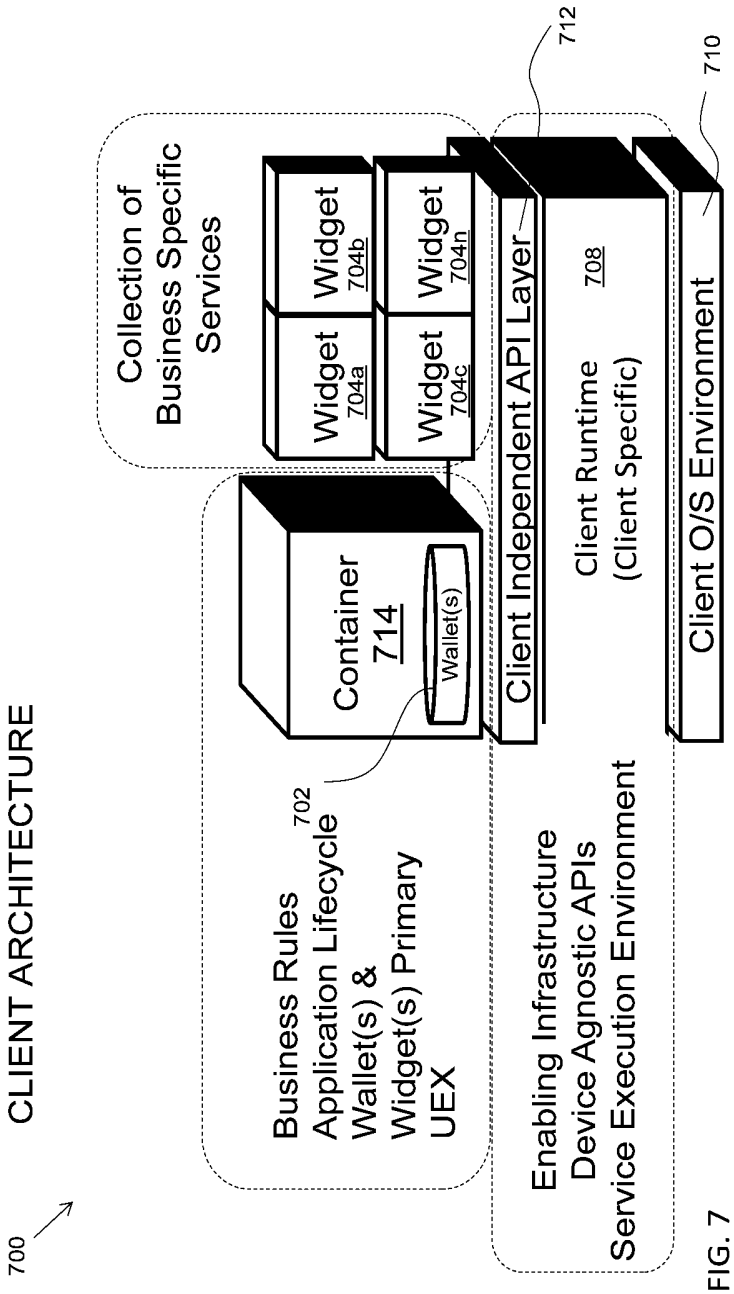

CLIENT ARCHITECTURE
700
Business Rules
Application Lifecycle
Wallet(s) &
Widget(s) Primary
UEX
Container
714
Wallet(s)
702
Collection of
Business Specific
Services
Widget
704a
Widget
704b
Widget
704c
Widget
704n
Client Independent API Layer
712
Client Runtime
(Client Specific)
708
Client O/S Environment
710
Enabling Infrastructure
Device Agnostic APIs
Service Execution Environment

FIG. 7

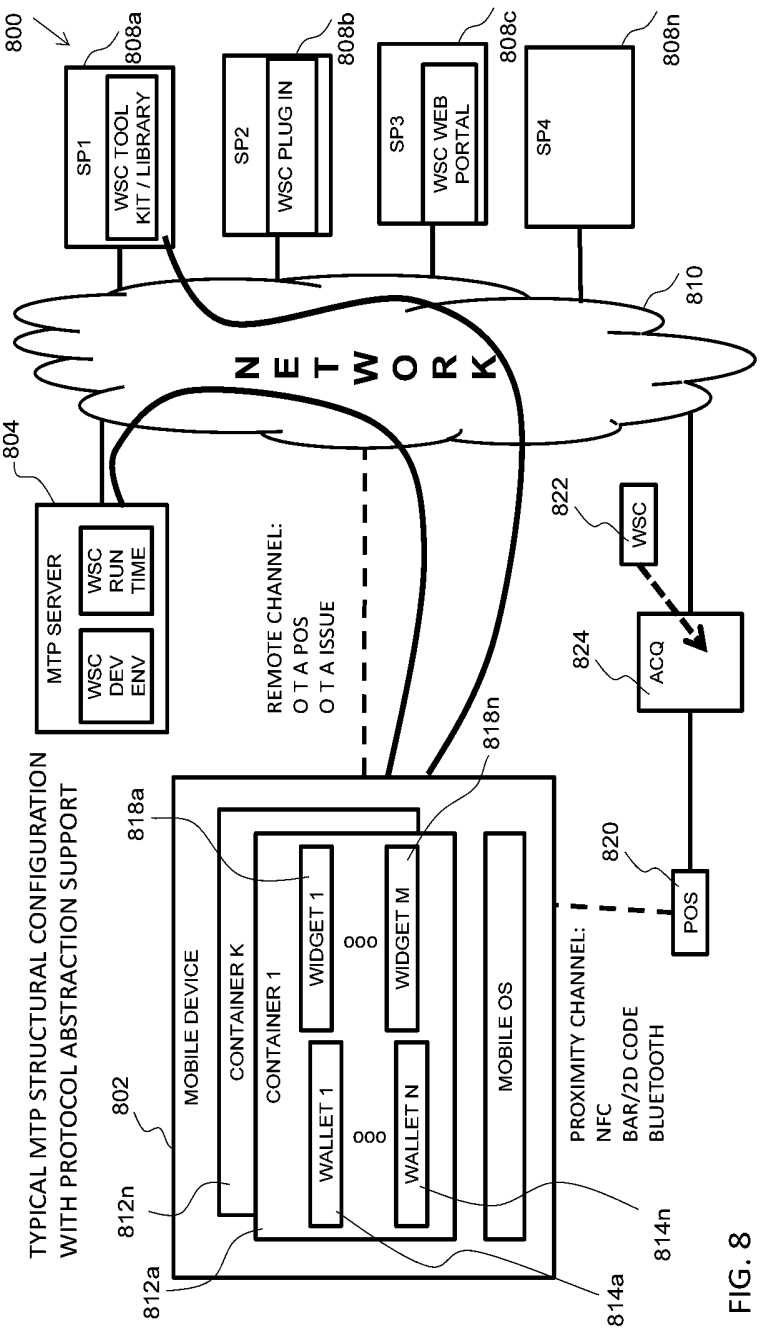

TYPICAL MTP STRUCTURAL CONFIGURATION
WITH PROTOCOL ABSTRACTION SUPPORT
800
808a
SP1
WSC TOOL KIT / LIBRARY
808b
SP2
WSC PLUG IN
808c
SP3
WSC WEB PORTAL
808n
SP4
810
NETWORK
804
MTP SERVER
WSC RUN TIME
WSC DEV ENV
REMOTE CHANNEL:
O T A POS
O T A ISSUE
822
WSC
824
ACQ
818n
818a
802
MOBILE DEVICE
CONTAINER K
CONTAINER 1
WIDGET 1
ooo
WIDGET M
WALLET 1
ooo
WALLET N
MOBILE OS
820
POS
PROXIMITY CHANNEL:
NFC
BAR/2D CODE
BLUETOOTH
812n
812a
814n
814a

FIG. 8

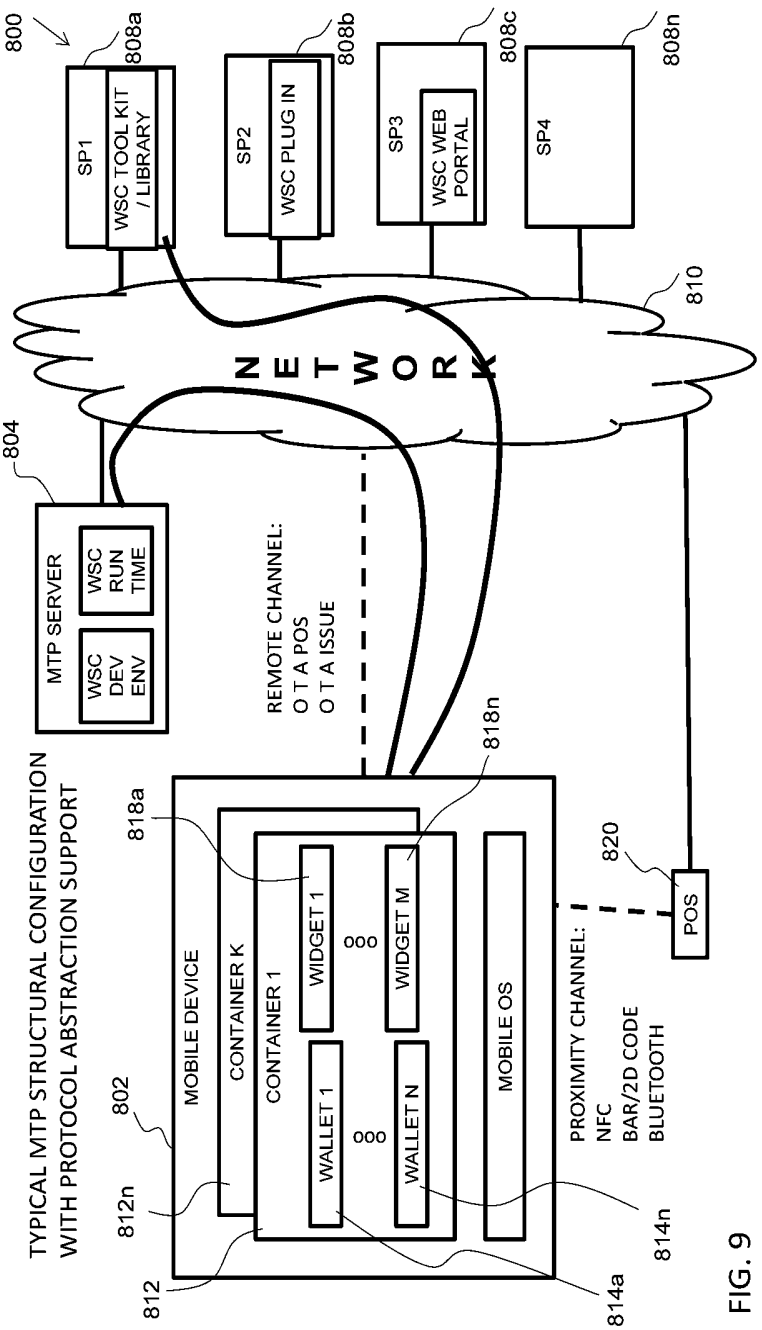

800
808a
SP1
WSC TOOL KIT / LIBRARY
808b
SP2
WSC PLUG IN
808c
SP3
WSC WEB PORTAL
808n
SP4
810
NETWORK
804
MTP SERVER
WSC RUN TIME
WSC DEV ENV
REMOTE CHANNEL:
O T A POS
O T A ISSUE
818n
818a
MOBILE DEVICE
802
CONTAINER K
CONTAINER 1
WIDGET 1
ooo
WIDGET M
WALLET 1
ooo
WALLET N
MOBILE OS
812n
812
814a
814n
820
POS
PROXIMITY CHANNEL:
NFC
BAR/2D CODE
BLUETOOTH
TYPICAL MTP STRUCTURAL CONFIGURATION
WITH PROTOCOL ABSTRACTION SUPPORT

FIG. 9

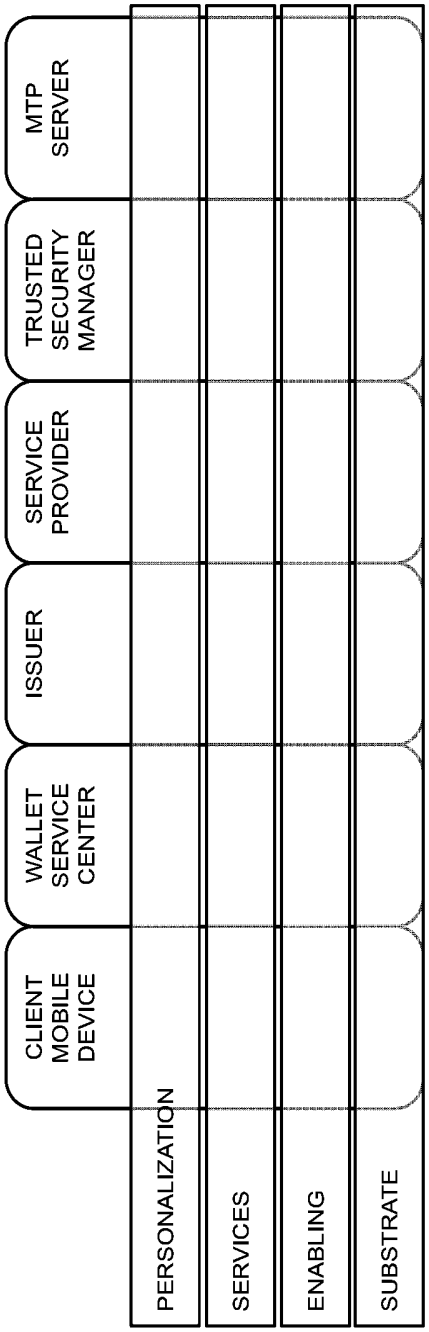

CLIENT MOBILE DEVICE
WALLET SERVICE CENTER
ISSUER
SERVICE PROVIDER
TRUSTED SECURITY MANAGER
MTP SERVER
PERSONALIZATION
SERVICES
ENABLING
SUBSTRATE

FIG. 10

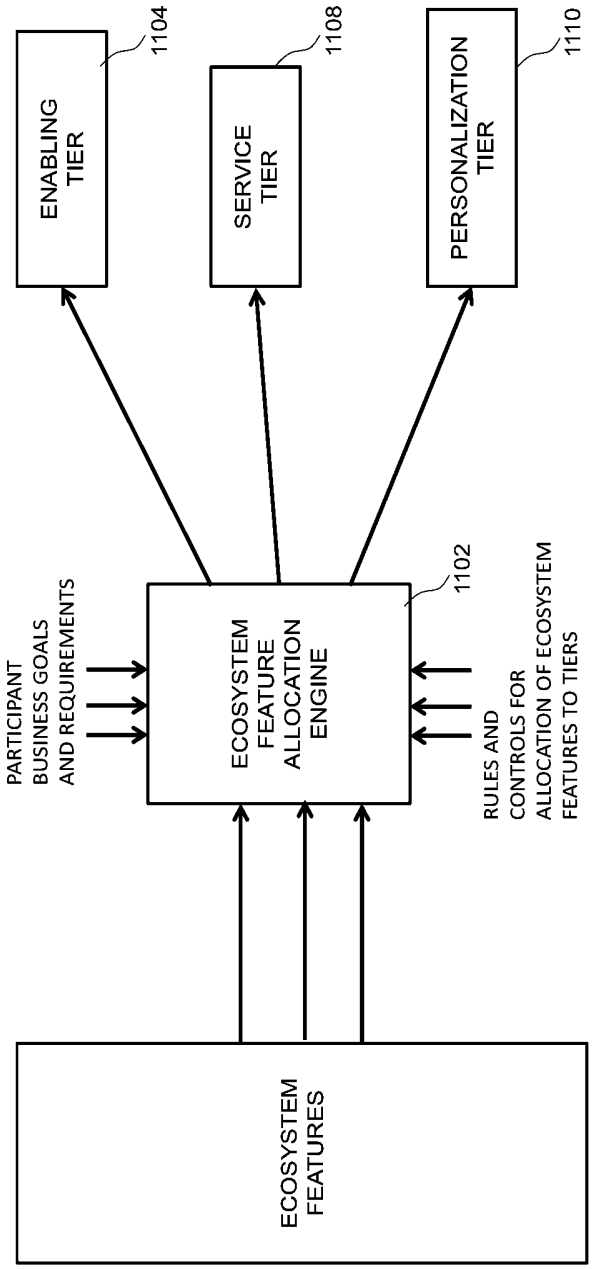

ENABLING TIER
1104
SERVICE TIER
1108
PERSONALIZATION TIER
1110
ECOSYSTEM FEATURE ALLOCATION ENGINE
1102
PARTICIPANT BUSINESS GOALS AND REQUIREMENTS
RULES AND CONTROLS FOR ALLOCATION OF ECOSYSTEM FEATURES TO TIERS
ECOSYSTEM FEATURES

FIG. 11

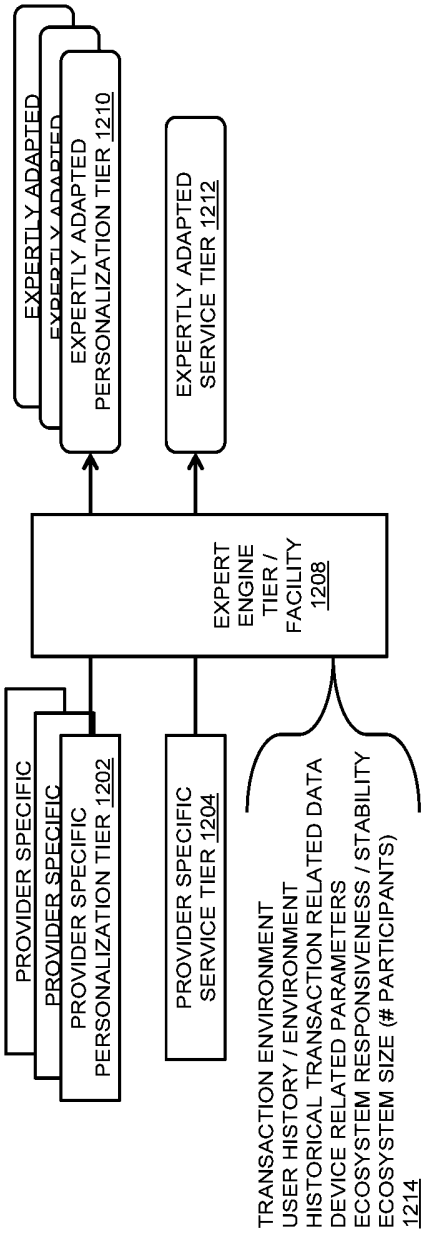

PROVIDER SPECIFIC
PROVIDER SPECIFIC
PROVIDER SPECIFIC PERSONALIZATION TIER 1202
PROVIDER SPECIFIC SERVICE TIER 1204
EXPERT ENGINE TIER / FACILITY 1208
EXPERTLY ADAPTED
EXPERTLY ADAPTED
EXPERTLY ADAPTED PERSONALIZATION TIER 1210
EXPERTLY ADAPTED SERVICE TIER 1212
TRANSACTION ENVIRONMENT
USER HISTORY / ENVIRONMENT
HISTORICAL TRANSACTION RELATED DATA
DEVICE RELATED PARAMETERS
ECOSYSTEM RESPONSIVENESS / STABILITY
ECOSYSTEM SIZE (# PARTICIPANTS)
1214

FIG. 12

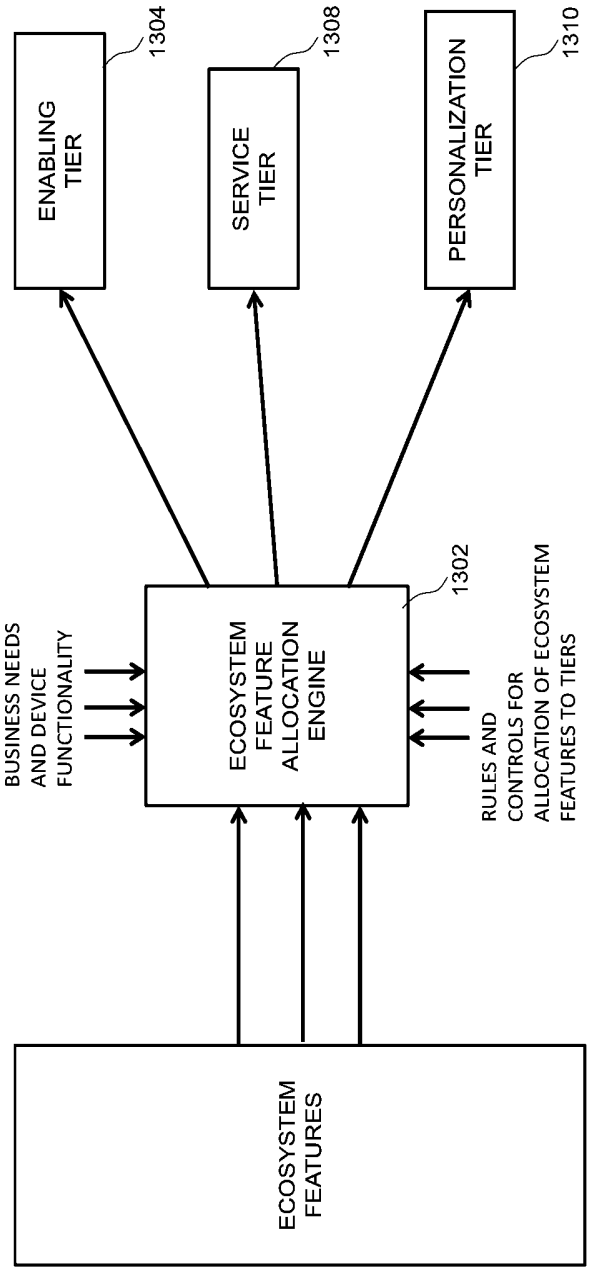

ENABLING TIER
1304
SERVICE TIER
1308
PERSONALIZATION TIER
1310
ECOSYSTEM FEATURE ALLOCATION ENGINE
1302
BUSINESS NEEDS AND DEVICE FUNCTIONALITY
RULES AND CONTROLS FOR ALLOCATION OF ECOSYSTEM FEATURES TO TIERS
ECOSYSTEM FEATURES

FIG. 13

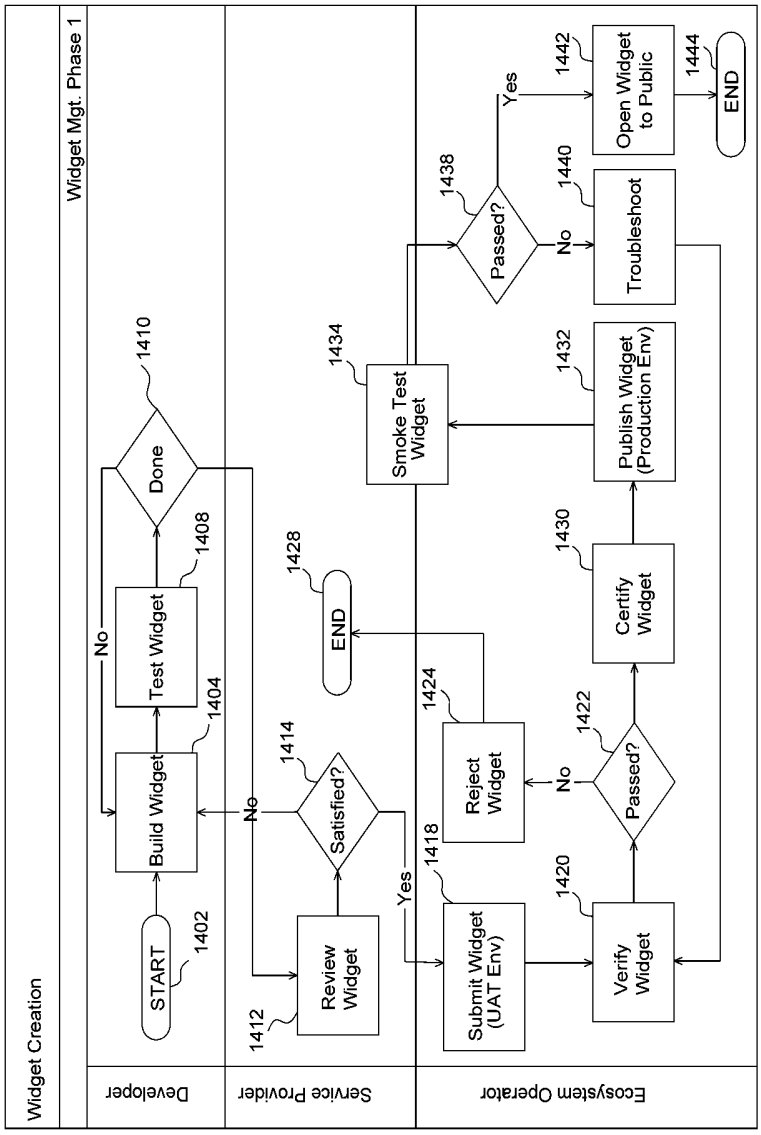

Widget Creation
Widget Mgt. Phase 1
Developer
Service Provider
Ecosystem Operator
START
1402
Build Widget
1404
Test Widget
1408
Done
1410
No
Review Widget
1412
Satisfied?
1414
No
Yes
END
1428
Submit Widget (UAT Env)
1418
Reject Widget
1424
Smoke Test Widget
1434
Verify Widget
1420
Passed?
1422
No
Yes
Certify Widget
1430
Publish Widget (Production Env)
1432
Passed?
1438
No
Troubleshoot
1440
Yes
Open Widget to Public
1442
END
1444

FIG. 14

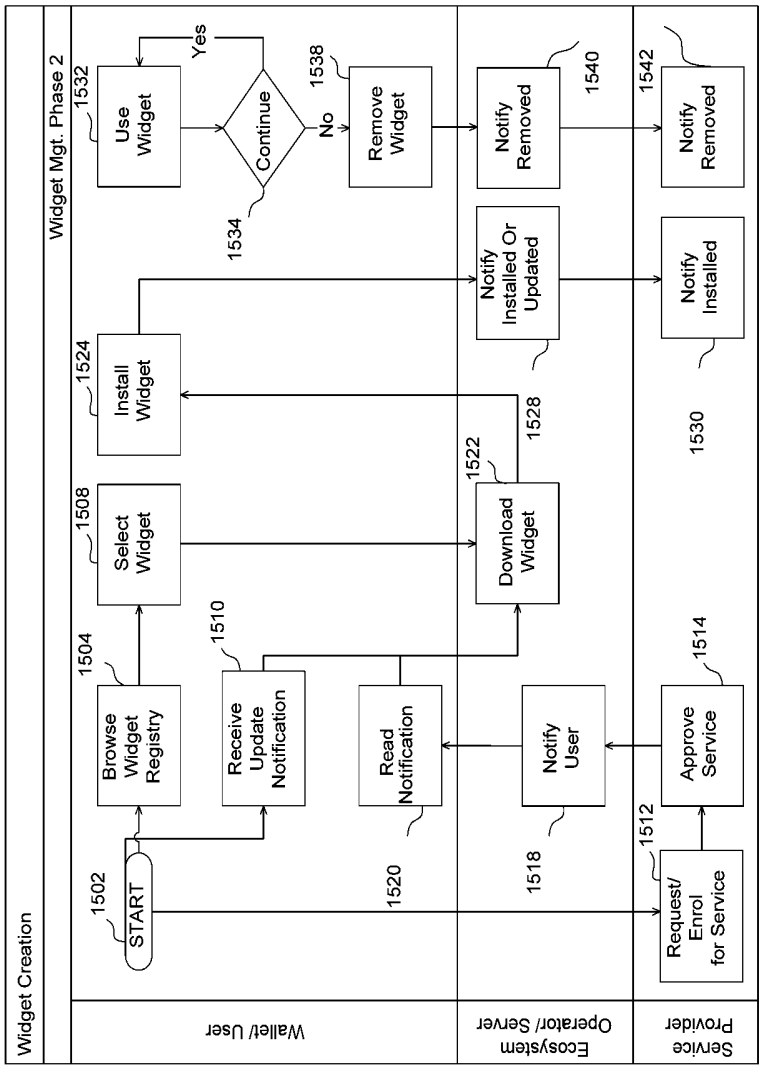

Widget Creation
Widget Mgt. Phase 2
Wallet/ User
Ecosystem Operator/ Server
Service Provider
1502
START
1504
Browse Widget Registry
1508
Select Widget
1510
Receive Update Notification
1512
Request/ Enrol for Service
1514
Approve Service
1518
Notify User
1520
Read Notification
1522
Download Widget
1524
Install Widget
1528
1530
Notify Installed
Notify Installed Or Updated
1532
Use Widget
1534
Continue
Yes
No
1538
Remove Widget
1540
Notify Removed
1542
Notify Removed

FIG. 15

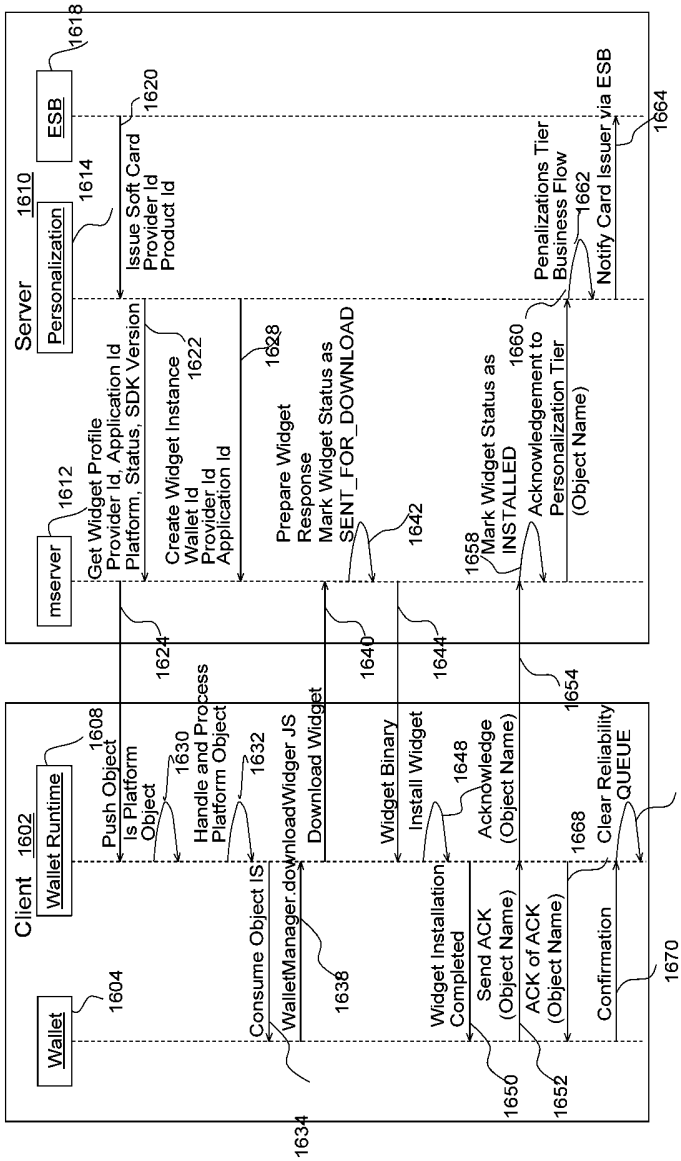

Client 1602
Wallet
1604
Wallet Runtime
1608
Server 1610
mserver
1612
Personalization
1614
ESB
1618
Push Object
Is Platform Object
1630
Handle and Process Platform Object
1632
Consume Object IS
WalletManager.downloadWidger JS
1638
Download Widget
1634
Get Widget Profile
Provider Id, Application Id
Platform, Status, SDK Version
1624
Create Widget Instance
Wallet Id
Provider Id
Application Id
1622
Issue Soft Card
Provider Id
Product Id
1620
Prepare Widget Response
Mark Widget Status as SENT_FOR_DOWNLOAD
1628
1640
Widget Binary
Install Widget
1642
1644
Widget Installation Completed
Send ACK (Object Name)
1648
1650
ACK of ACK (Object Name)
1652
1654
Mark Widget Status as INSTALLED
1658
Acknowledgement to Personalization Tier (Object Name)
1660
Penalizations Tier Business Flow
1662
Notify Card Issuer via ESB
1664
Confirmation
1670
Clear Reliability QUEUE
1668
1672

FIG. 16

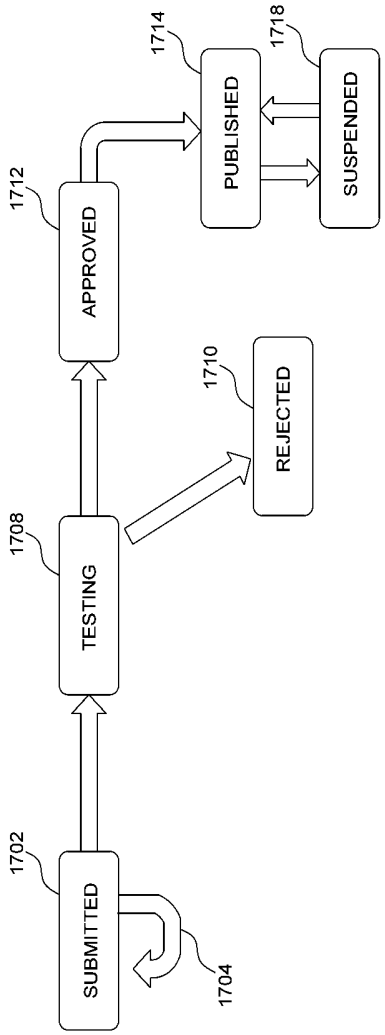

SUBMITTED
1702
1704
TESTING
1708
REJECTED
1710
APPROVED
1712
PUBLISHED
1714
SUSPENDED
1718

FIG. 17

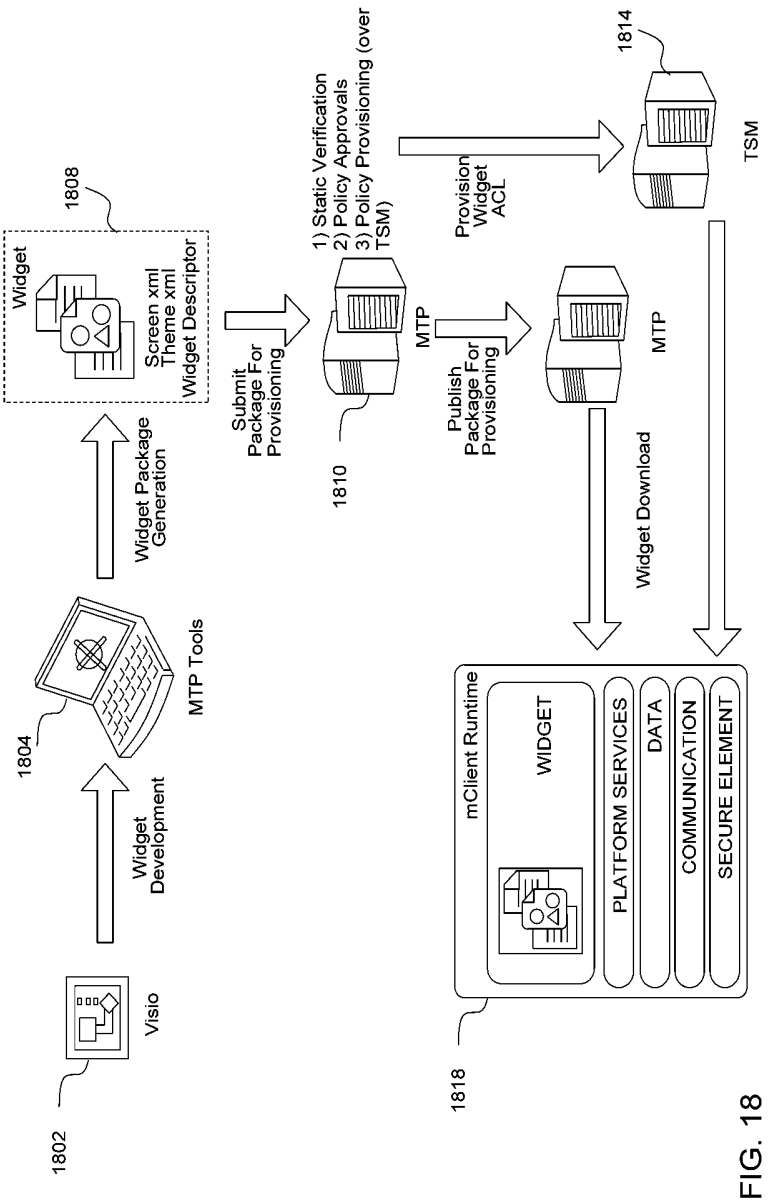

1802
Visio
Widget Development
1804
MTP Tools
Widget Package Generation
1808
Widget
Screen xml
Theme xml
Widget Descriptor
Submit Package For Provisioning
1) Static Verification
2) Policy Approvals
3) Policy Provisioning (over TSM)
1810
MTP
Publish Package For Provisioning
Provision Widget ACL
1814
TSM
MTP
Widget Download
1818
mClient Runtime
WIDGET
PLATFORM SERVICES
DATA
COMMUNICATION
SECURE ELEMENT

FIG. 18

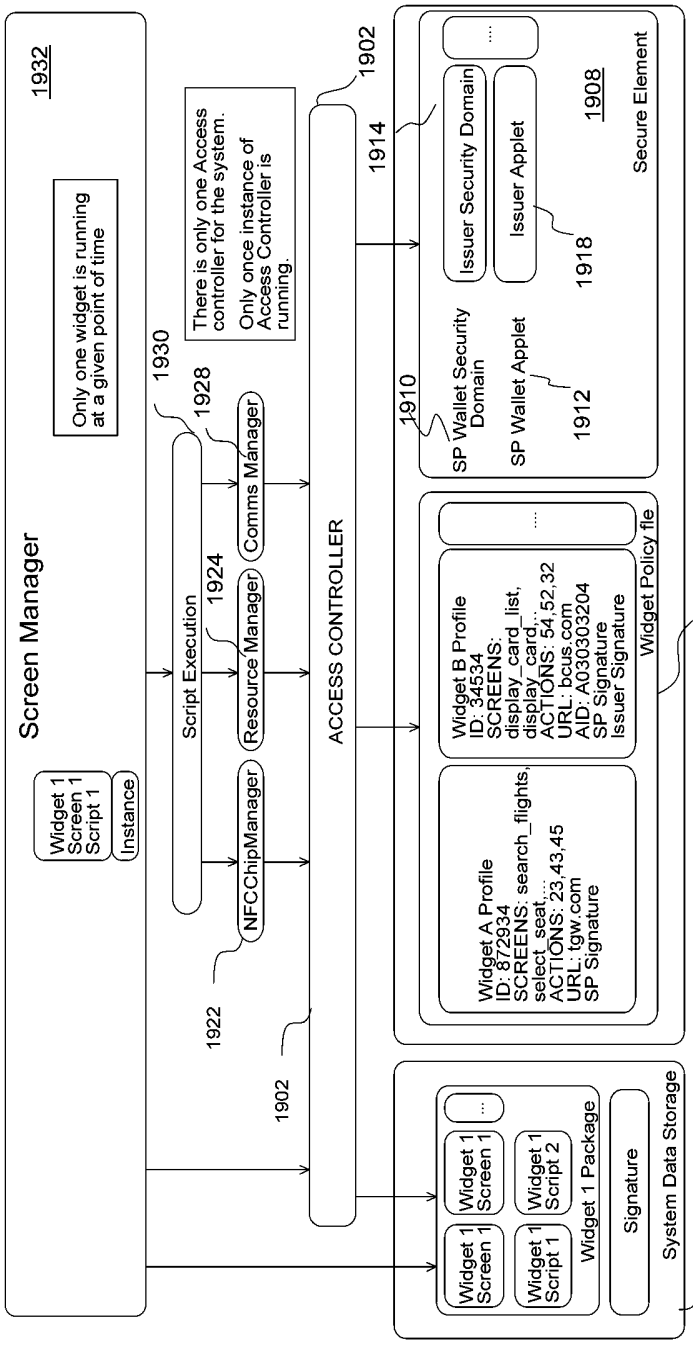

Screen Manager
1932
Only one widget is running at a given point of time
Widget 1 Screen 1 Script 1
Instance
1930
Script Execution
1928
Comms Manager
1924
Resource Manager
NFCChipManager
1922
There is only one Access controller for the system.
Only once instance of Access Controller is running.
1902
ACCESS CONTROLLER
1914
1902
1910
SP Wallet Security Domain
SP Wallet Applet
1912
Issuer Security Domain
Issuer Applet
1918
1908
Secure Element
Widget B Profile
ID: 34534
SCREENS:
display_card_list,
display_card,...
ACTIONS: 54,52,32
URL: bcus.com
AID: A030303204
SP Signature
Issuer Signature
Widget Policy file
Widget A Profile
ID: 872934
SCREENS: search_flights,
select_seat,...
ACTIONS: 23,43,45
URL: tgw.com
SP Signature
1920
Widget 1 Screen 1
Widget 1 Script 1
Widget 1 Screen 1
Widget 1 Script 2
Widget 1 Package
Signature
System Data Storage
1904

FIG. 19

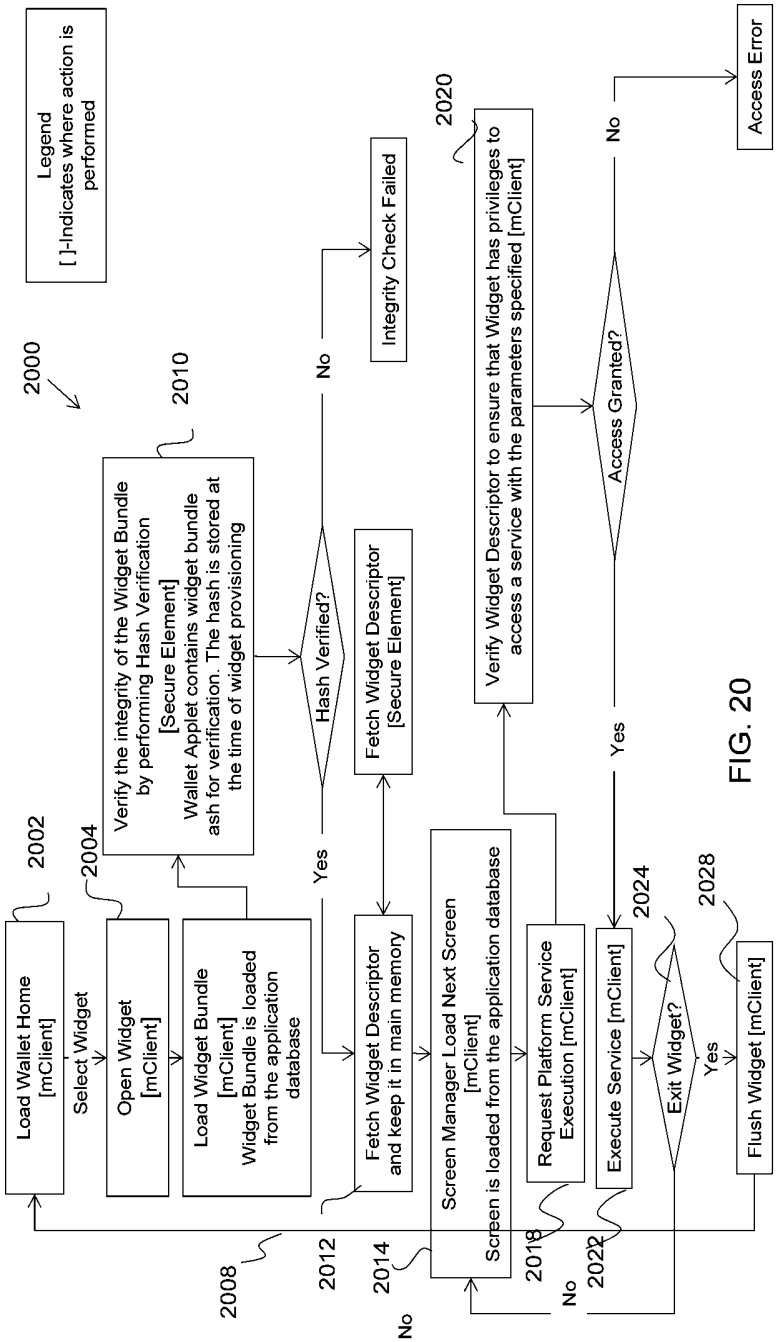

Legend
[ ]-Indicates where action is performed
2000
Load Wallet Home [mClient]
2002
Select Widget
Open Widget [mClient]
2004
Load Widget Bundle [mClient] Widget Bundle is loaded from the application database
Verify the integrity of the Widget Bundle by performing Hash Verification [Secure Element] Wallet Applet contains widget bundle ash for verification. The hash is stored at the time of widget provisioning
2010
Hash Verified?
Yes
No
Integrity Check Failed
Fetch Widget Descriptor [Secure Element]
Fetch Widget Descriptor and keep it in main memory
2012
2008
Screen Manager Load Next Screen [mClient] Screen is loaded from the application database
2014
No
Request Platform Service Execution [mClient]
2018
Verify Widget Descriptor to ensure that Widget has privileges to access a service with the parameters specified [mClient]
2020
Access Granted?
Yes
No
Access Error
Execute Service [mClient]
2024
Exit Widget?
2022
No
Yes
2028
Flush Widget [mClient]

FIG. 20

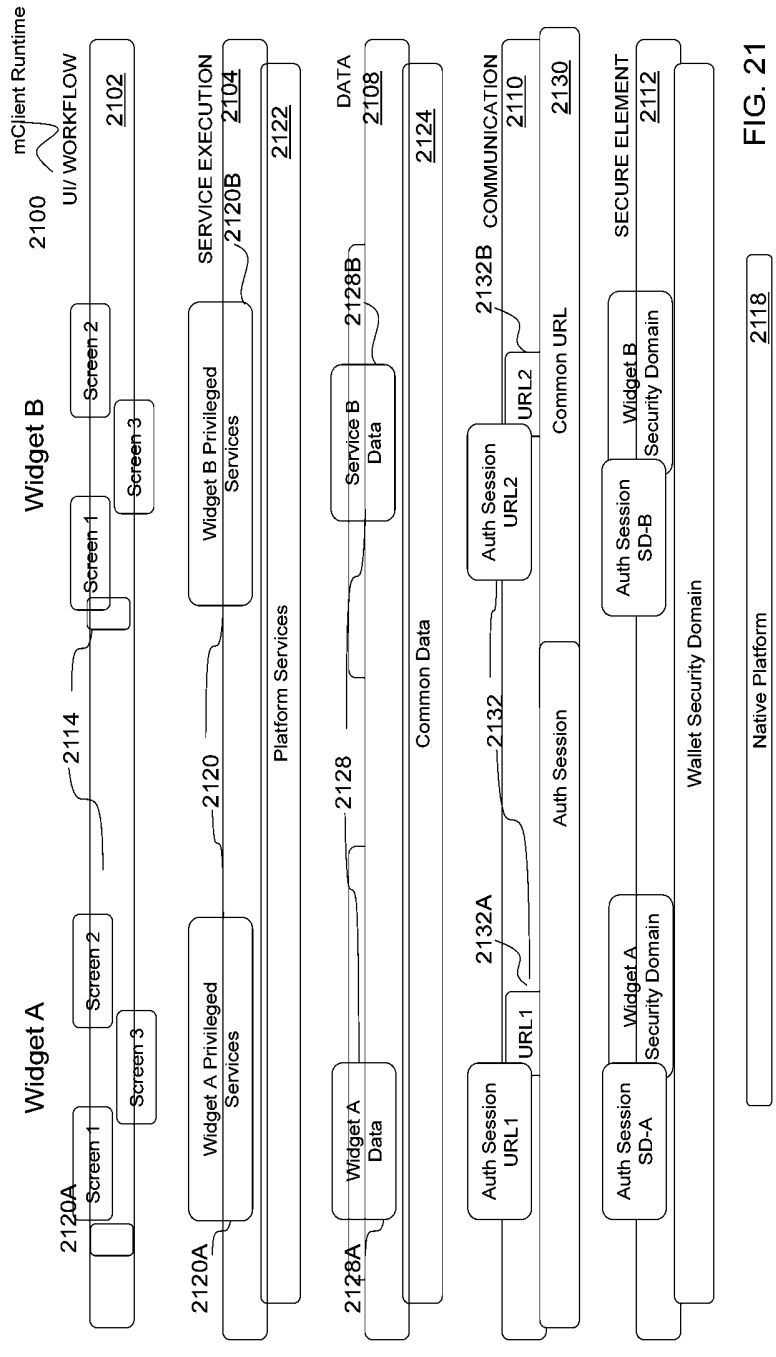

mClient Runtime
2100
UI/ WORKFLOW
2102
Widget A
Screen 1
Screen 2
Screen 3
2120A
2114
Widget B
Screen 1
Screen 2
Screen 3
SERVICE EXECUTION
2104
2120B
Widget A Privileged Services
2120
Widget B Privileged Services
Platform Services
2122
DATA
2108
2128B
Widget A Data
2128
Service B Data
Common Data
2124
COMMUNICATION
2110
2132B
2132A
Auth Session URL1
URL1
2132
Auth Session URL2
URL2
Common URL
Auth Session
2130
SECURE ELEMENT
2112
Auth Session SD-A
Widget A Security Domain
Auth Session SD-B
Widget B Security Domain
Wallet Security Domain
Native Platform
2118

FIG. 21

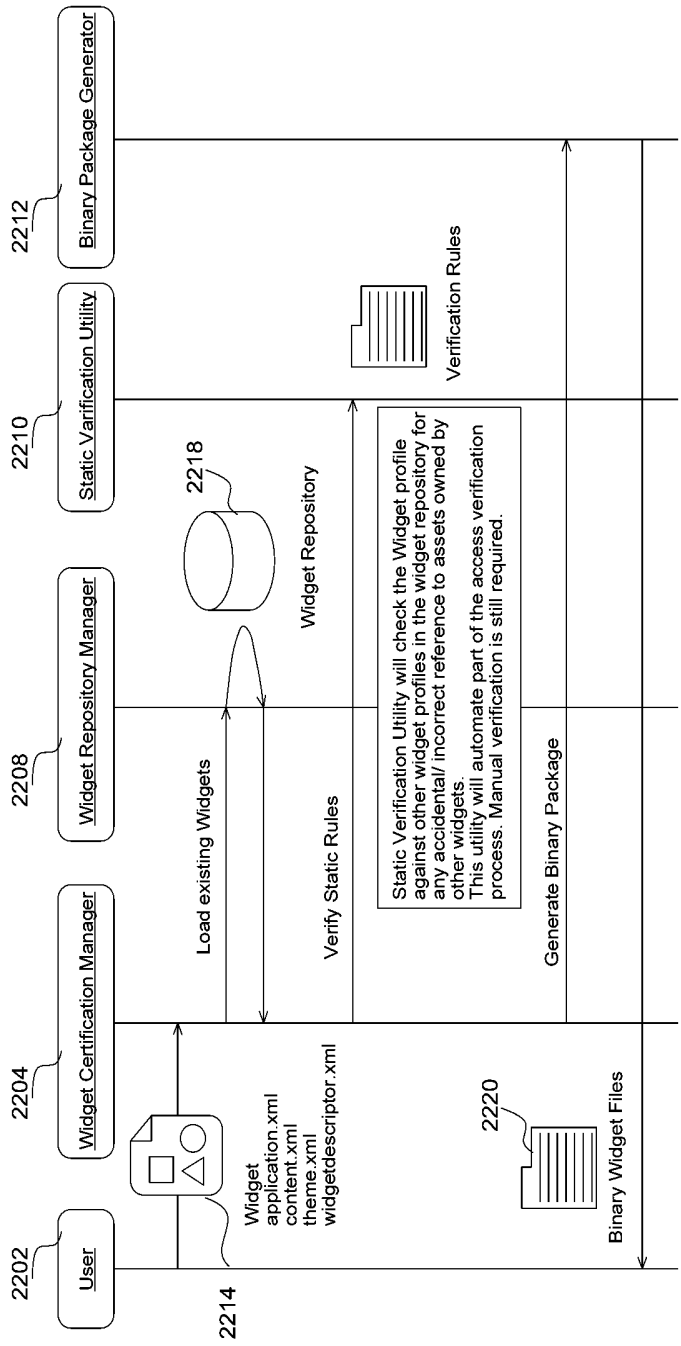

2202
User
2204
Widget Certification Manager
2208
Widget Repository Manager
2210
Static Varification Utility
2212
Binary Package Generator
2214
Widget
application.xml
content.xml
theme.xml
widgetdescriptor.xml
Load existing Widgets
Verify Static Rules
2218
Widget Repository
Verification Rules
Static Verification Utility will check the Widget profile against other widget profiles in the widget repository for any accidental/ incorrect reference to assets owned by other widgets.
This utility will automate part of the access verification process. Manual verification is still required.
Generate Binary Package
2220
Binary Widget Files

FIG. 22

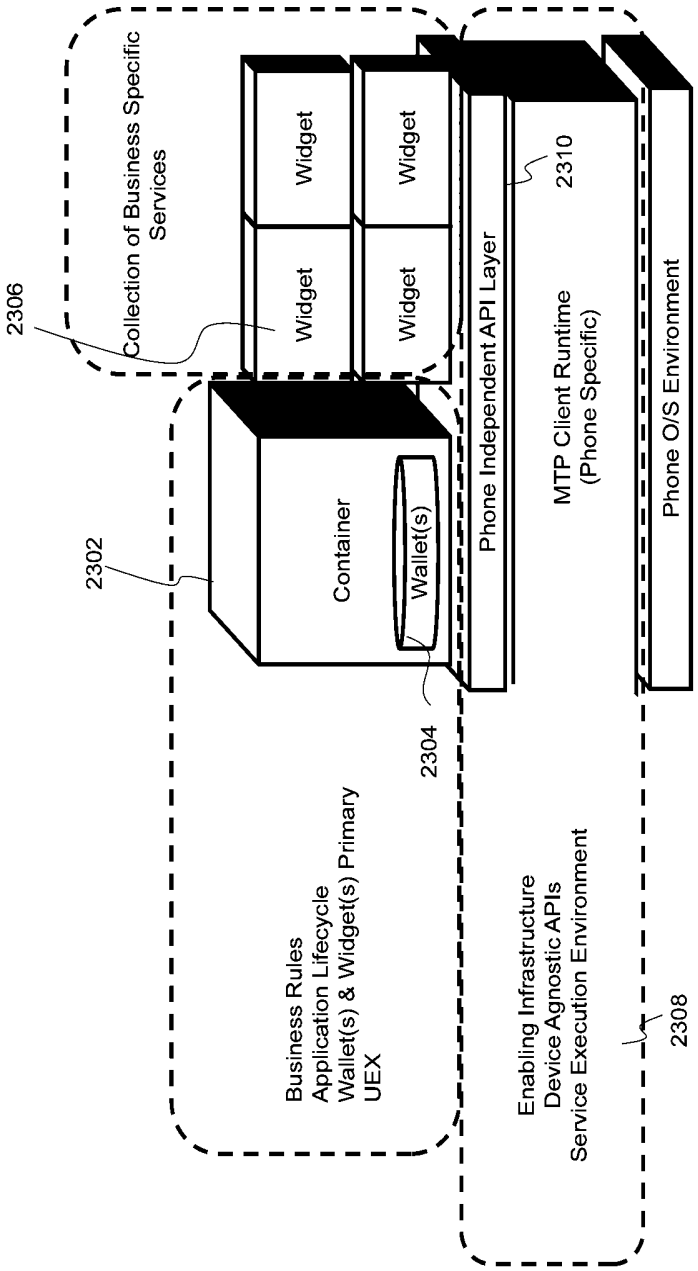

2306
Collection of Business Specific Services
Widget
Widget
Widget
Widget
2302
Container
Wallet(s)
2304
Business Rules
Application Lifecycle
Wallet(s) & Widget(s) Primary
UEX
Phone Independent API Layer
2310
MTP Client Runtime
(Phone Specific)
Phone O/S Environment
Enabling Infrastructure
Device Agnostic APIs
Service Execution Environment
2308

FIG. 23

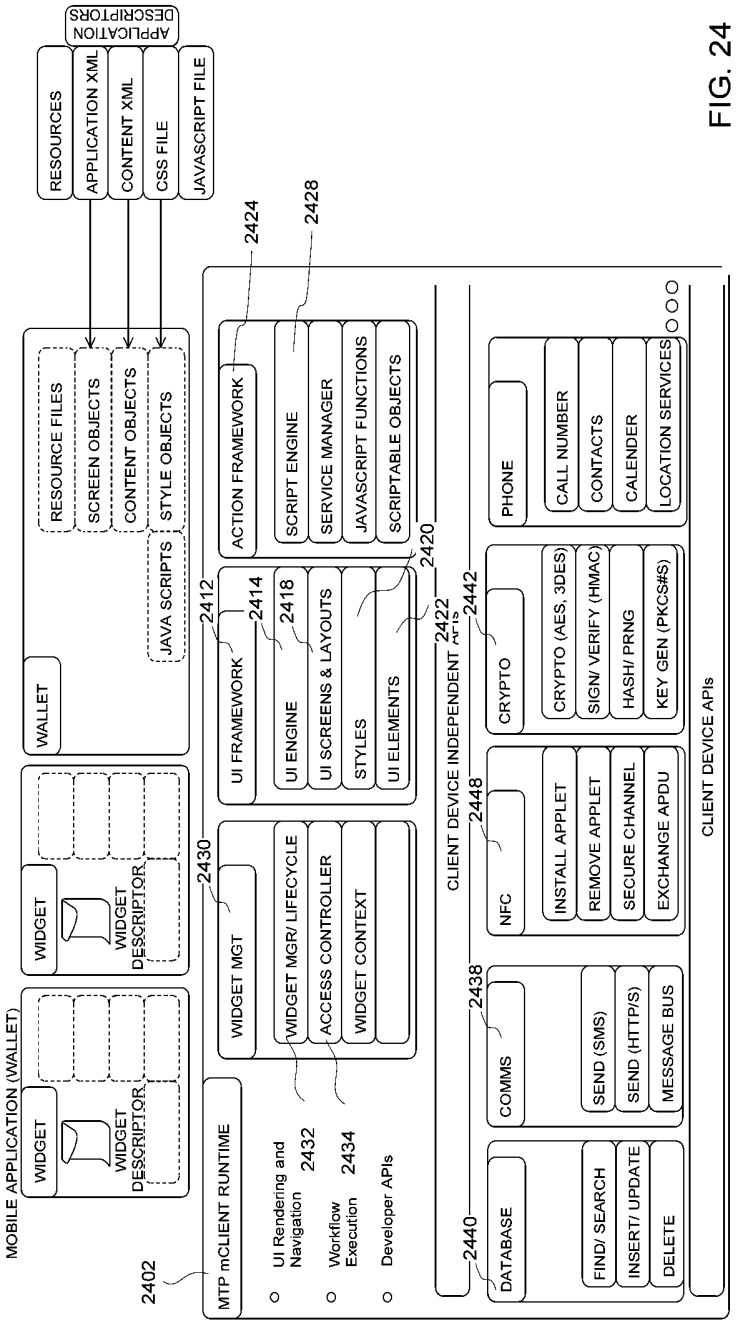

APPLICATION DESCRIPTORS
RESOURCES
APPLICATION XML
CONTENT XML
CSS FILE
JAVASCRIPT FILE
WALLET
RESOURCE FILES
SCREEN OBJECTS
CONTENT OBJECTS
STYLE OBJECTS
JAVA SCRIPTS
WIDGET
WIDGET DESCRIPTOR
WIDGET
WIDGET DESCRIPTOR
MOBILE APPLICATION (WALLET)
2402
MTP mCLIENT RUNTIME
UI Rendering and Navigation 2432
Workflow Execution 2434
Developer APIs
2430
WIDGET MGT
WIDGET MGR/ LIFECYCLE
ACCESS CONTROLLER
WIDGET CONTEXT
2412
UI FRAMEWORK
2414
UI ENGINE
2418
UI SCREENS & LAYOUTS
STYLES
UI ELEMENTS
2420
2422
2424
ACTION FRAMEWORK
2428
SCRIPT ENGINE
SERVICE MANAGER
JAVASCRIPT FUNCTIONS
SCRIPTABLE OBJECTS
CLIENT DEVICE INDEPENDENT APIs
2440
DATABASE
FIND/ SEARCH
INSERT/ UPDATE
DELETE
2438
COMMS
SEND (SMS)
SEND (HTTP/S)
MESSAGE BUS
2448
NFC
INSTALL APPLET
REMOVE APPLET
SECURE CHANNEL
EXCHANGE APDU
2442
CRYPTO
CRYPTO (AES, 3DES)
SIGN/ VERIFY (HMAC)
HASH/ PRNG
KEY GEN (PKCS#S)
PHONE
CALL NUMBER
CONTACTS
CALENDER
LOCATION SERVICES
CLIENT DEVICE APIs

FIG. 24

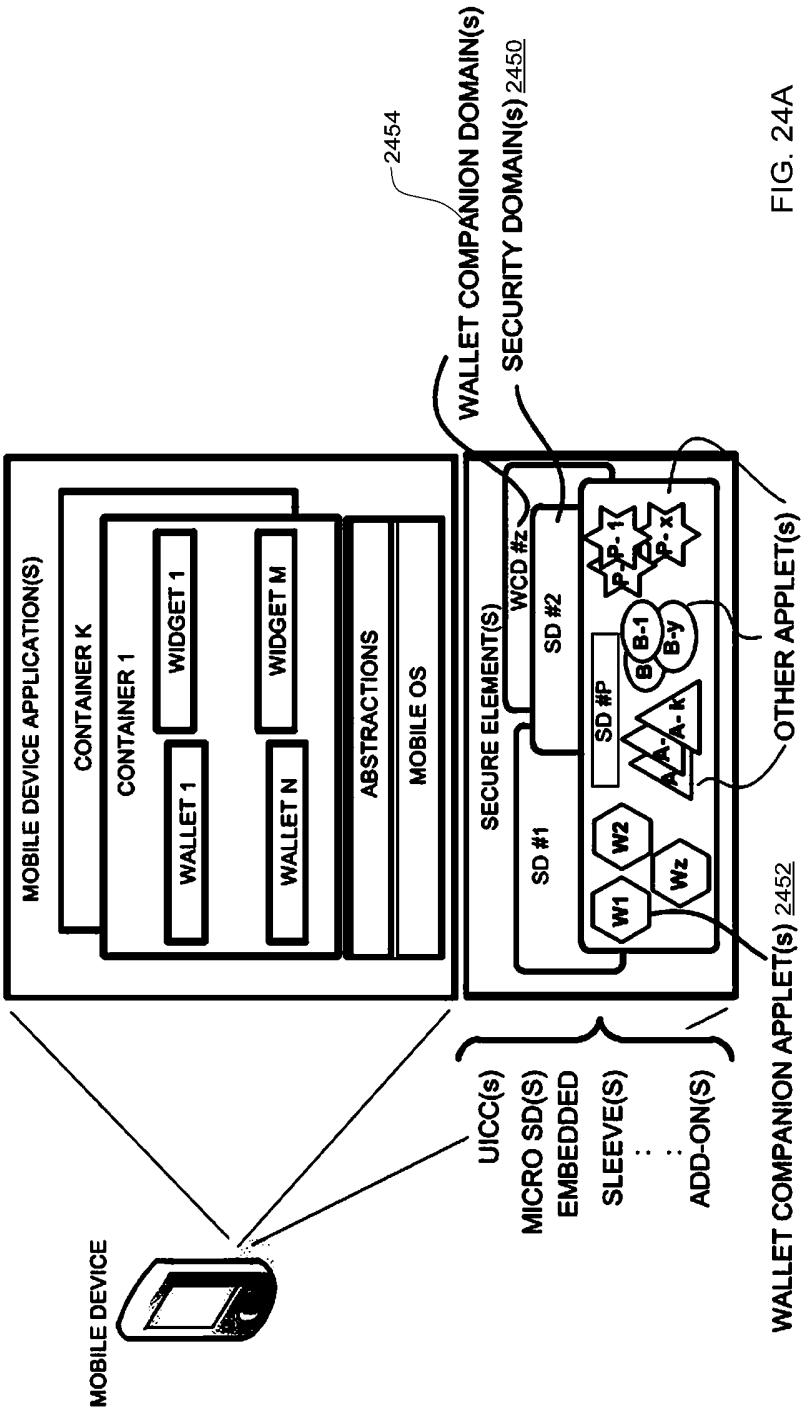

MOBILE DEVICE
MOBILE DEVICE APPLICATION(S)
CONTAINER K
CONTAINER 1
WIDGET 1
WIDGET M
WALLET 1
WALLET N
ABSTRACTIONS
MOBILE OS
2454
WALLET COMPANION DOMAIN(s)
SECURITY DOMAIN(s) 2450
SECURE ELEMENT(S)
WCD #z
SD #1
SD #2
SD #P
W1
W2
Wz
A-1
A-k
B-1
B-y
P-1
P-z
P-x
OTHER APPLET(s)
WALLET COMPANION APPLET(s) 2452
UICC(s)
MICRO SD(S)
EMBEDDED
SLEEVE(S)
ADD-ON(S)

FIG. 24A

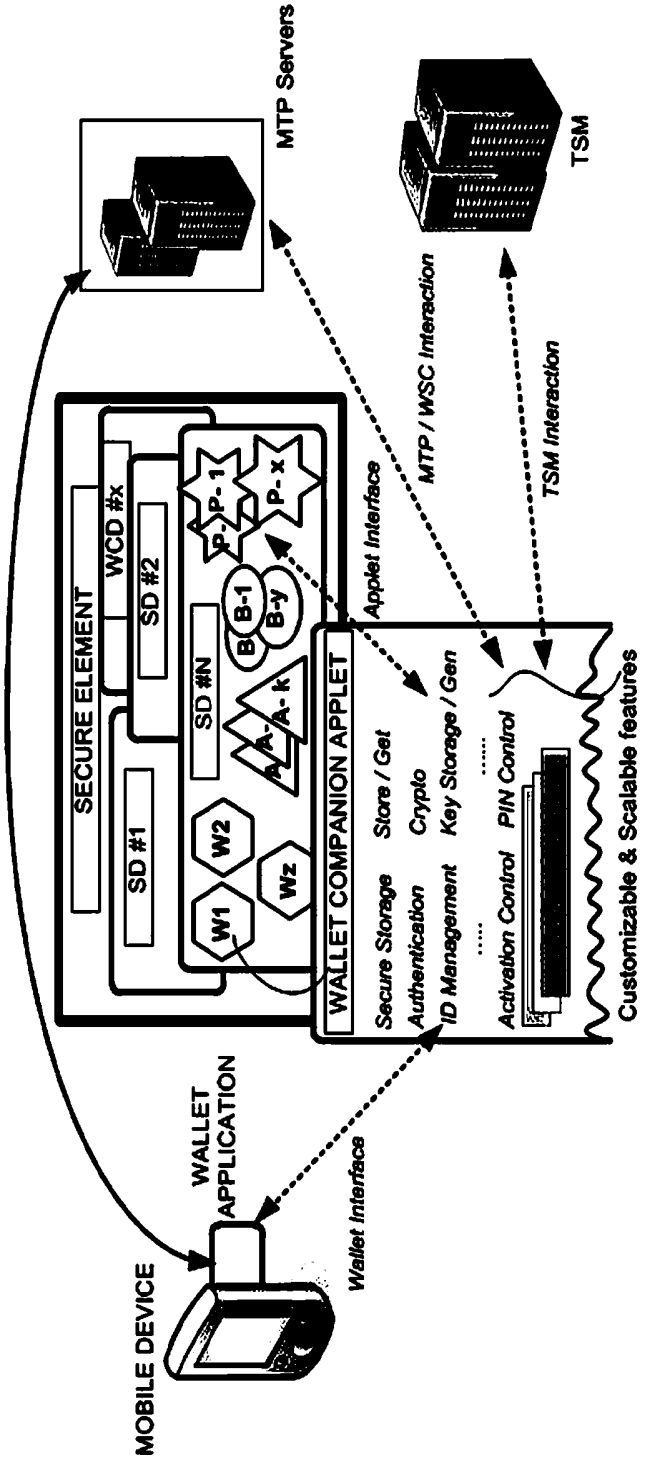

MTP Servers
TSM
MTP / WSC Interaction
TSM Interaction
Applet Interface
SECURE ELEMENT
WCD #x
SD #1
SD #2
SD #N
W1
W2
Wz
A-1
A-k
B-1
B-y
P-1
P-x
WALLET COMPANION APPLET
Secure Storage
Store / Get
Authentication
Crypto
ID Management
Key Storage / Gen
.....
Activation Control
PIN Control
Customizable & Scalable features
WALLET APPLICATION
Wallet Interface
MOBILE DEVICE

FIG. 24B

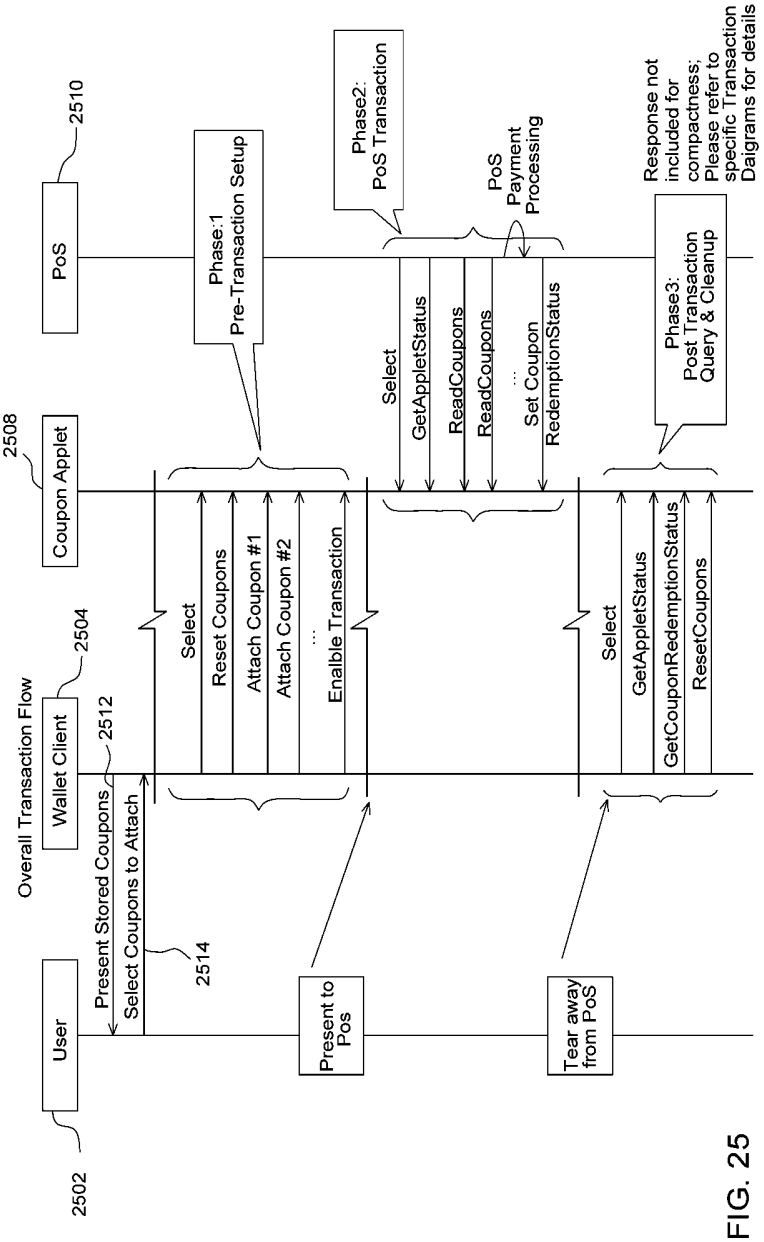

Overall Transaction Flow
User
2502
Wallet Client
2504
Coupon Applet
2508
PoS
2510
2512
Present Stored Coupons
Select Coupons to Attach
2514
Present to Pos
Tear away from PoS
Select
Reset Coupons
Attach Coupon #1
Attach Coupon #2
...
Enalble Transaction
Phase:1 Pre-Transaction Setup
Phase2: PoS Transaction
PoS Payment Processing
Select
GetAppletStatus
ReadCoupons
ReadCoupons
...
Set Coupon RedemptionStatus
Response not included for compactness; Please refer to specific Transaction Daigrams for details
Phase3: Post Transaction Query & Cleanup
Select
GetAppletStatus
GetCouponRedemptionStatus
ResetCoupons

FIG. 25

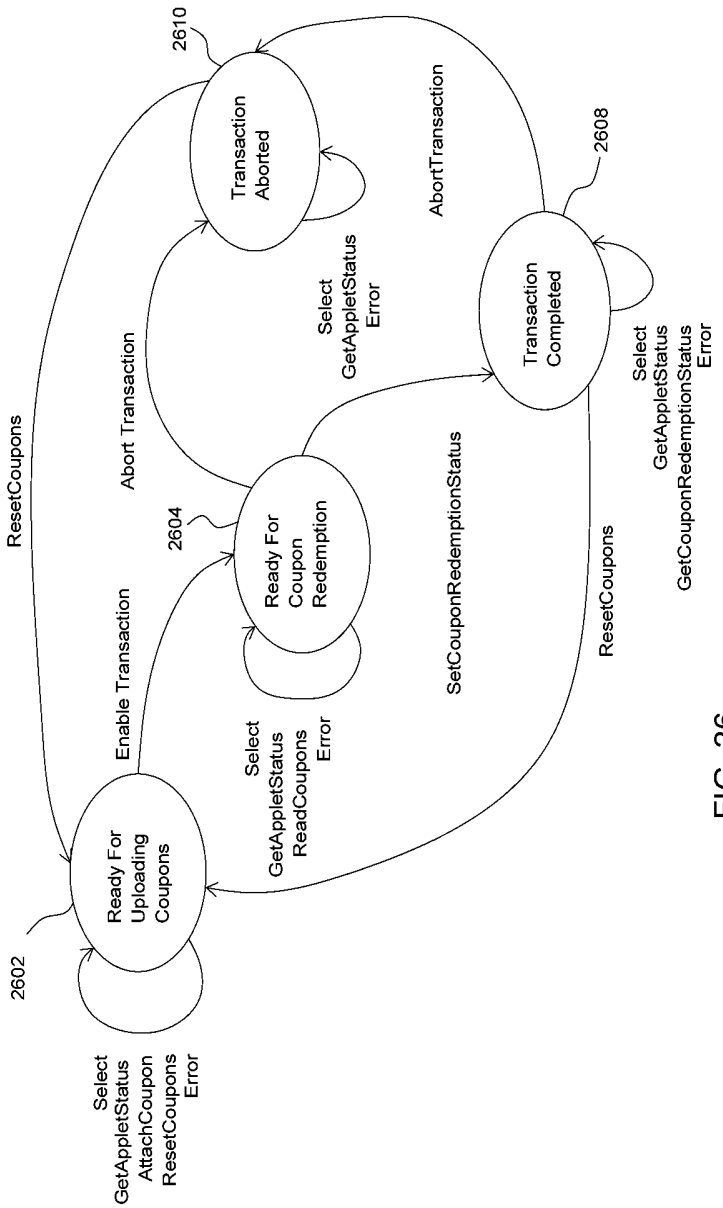

2602
Ready For Uploading Coupons
Select GetAppletStatus AttachCoupon ResetCoupons Error
Enable Transaction
2604
Ready For Coupon Redemption
Select GetAppletStatus ReadCoupons Error
Abort Transaction
2610
Transaction Aborted
Select GetAppletStatus Error
ResetCoupons
SetCouponRedemptionStatus
2608
Transaction Completed
Select GetAppletStatus GetCouponRedemptionStatus Error
AbortTransaction
ResetCoupons

FIG. 26

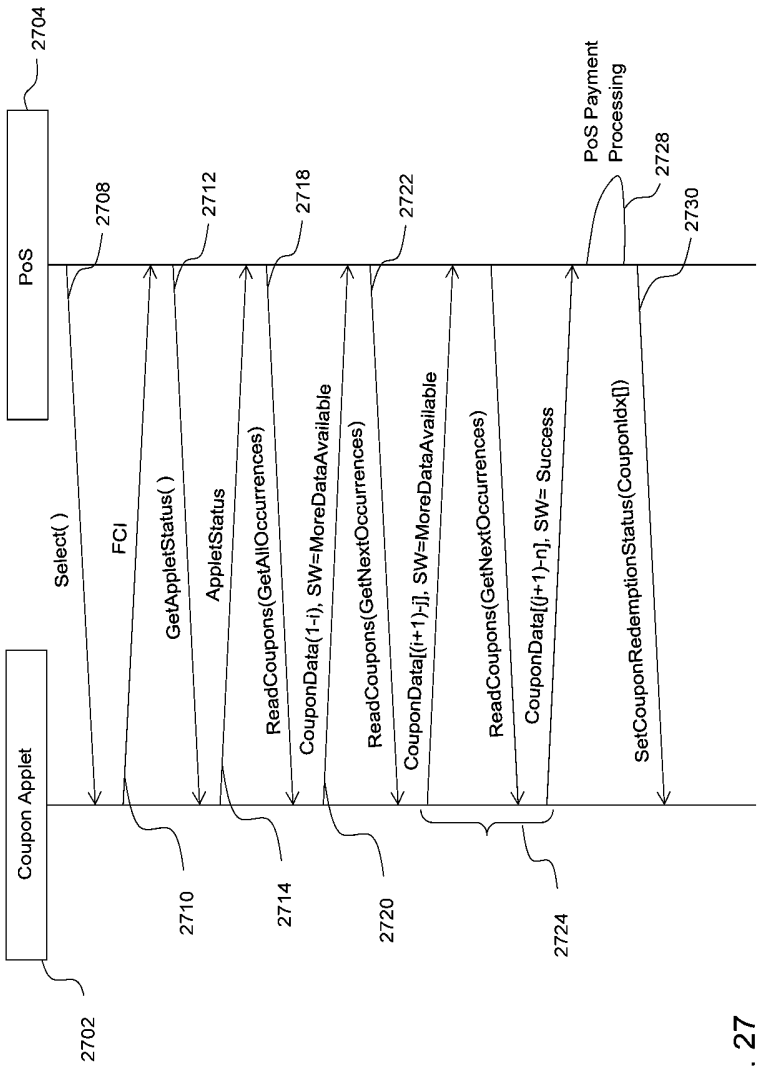

2704
PoS
2708
2712
2718
2722
PoS Payment Processing
2728
2730
Coupon Applet
2702
Select( )
FCI
GetAppletStatus( )
AppletStatus
ReadCoupons(GetAllOccurrences)
CouponData(1-i), SW=MoreDataAvailable
ReadCoupons(GetNextOccurrences)
CouponData[(i+1)-j], SW=MoreDataAvailable
ReadCoupons(GetNextOccurrences)
CouponData[(j+1)-n], SW= Success
SetCouponRedemptionStatus(CouponIdx[])
2710
2714
2720
2724

FIG. 27

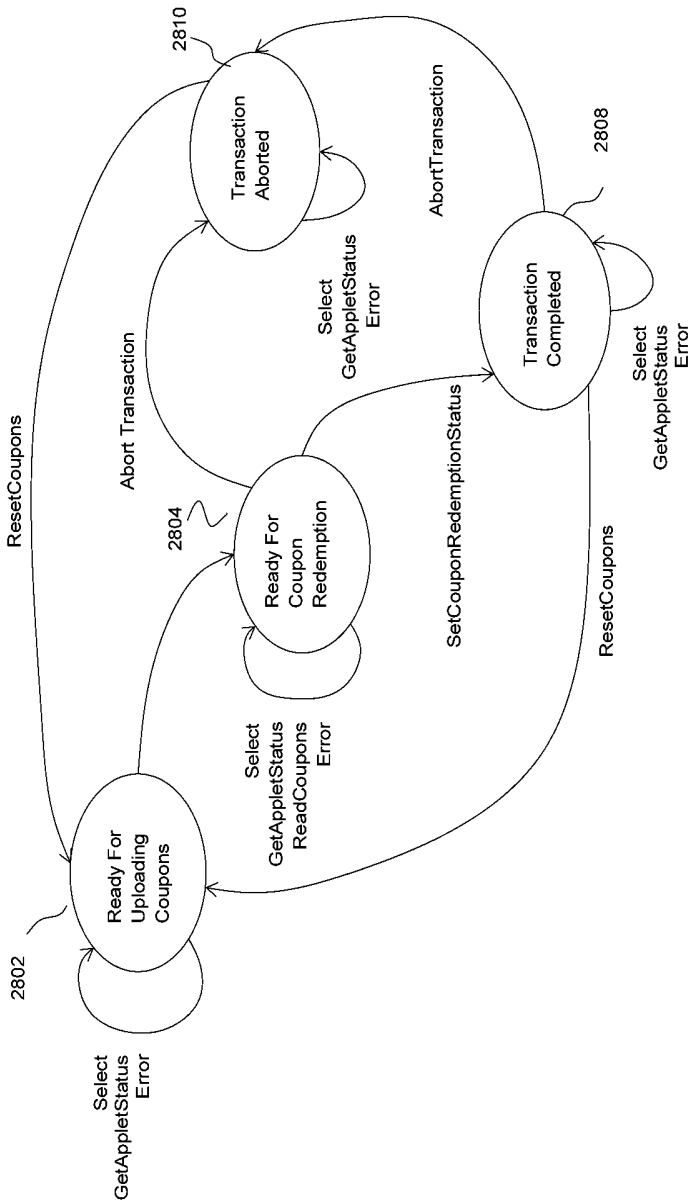

2810
Transaction Aborted
AbortTransaction
2808
Transaction Completed
Select GetAppletStatus Error
Select GetAppletStatus Error
ResetCoupons
Abort Transaction
2804
Ready For Coupon Redemption
SetCouponRedemptionStatus
ResetCoupons
Select GetAppletStatus ReadCoupons Error
Ready For Uploading Coupons
2802
Select GetAppletStatus Error

FIG. 28

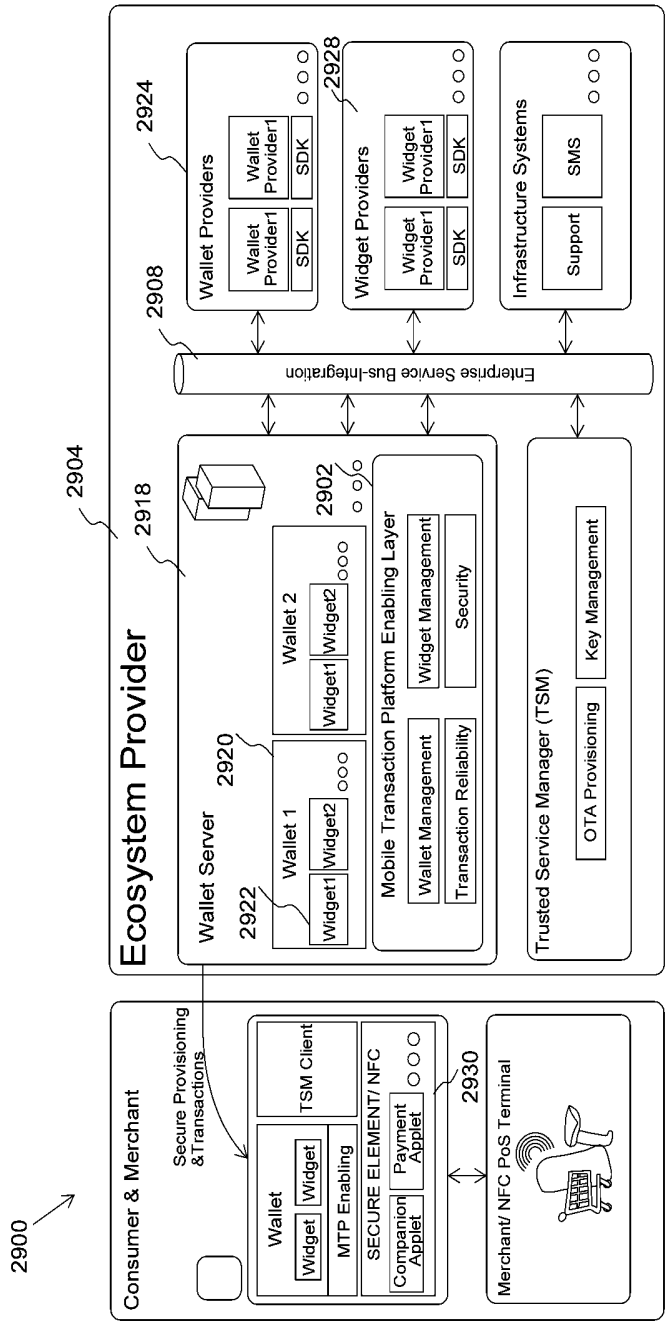

2900
Consumer & Merchant
Secure Provisioning &Transactions
Wallet
Widget
Widget
MTP Enabling
TSM Client
SECURE ELEMENT/ NFC
Companion Applet
Payment Applet
2930
Merchant/ NFC PoS Terminal
2904
Ecosystem Provider
2918
Wallet Server
2920
Wallet 1
Widget1
Widget2
2922
Wallet 2
Widget1
Widget2
2902
Mobile Transaction Platform Enabling Layer
Wallet Management
Transaction Reliability
Widget Management
Security
2908
Enterprise Service Bus-Integration
2924
Wallet Providers
Wallet Provider1
SDK
Wallet Provider1
SDK
2928
Widget Providers
Widget Provider1
SDK
Widget Provider1
SDK
Infrastructure Systems
Support
SMS
Trusted Service Manager (TSM)
OTA Provisioning
Key Management

FIG. 29

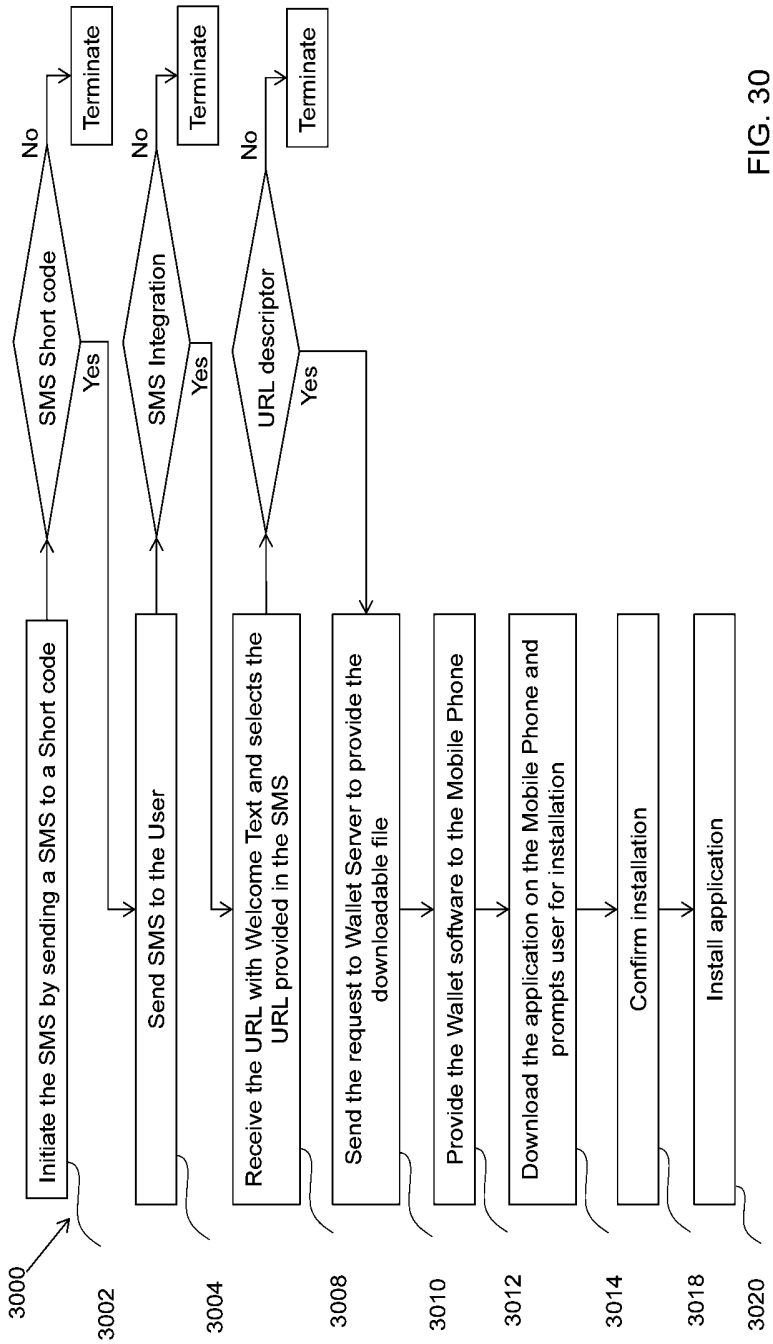

3000
Initiate the SMS by sending a SMS to a Short code
3002
SMS Short code
No
Terminate
Yes
Send SMS to the User
3004
SMS Integration
No
Terminate
Yes
Receive the URL with Welcome Text and selects the URL provided in the SMS
3008
URL descriptor
No
Terminate
Yes
Send the request to Wallet Server to provide the downloadable file
3010
Provide the Wallet software to the Mobile Phone
3012
Download the application on the Mobile Phone and prompts user for installation
3014
Confirm installation
3018
Install application
3020

FIG. 30

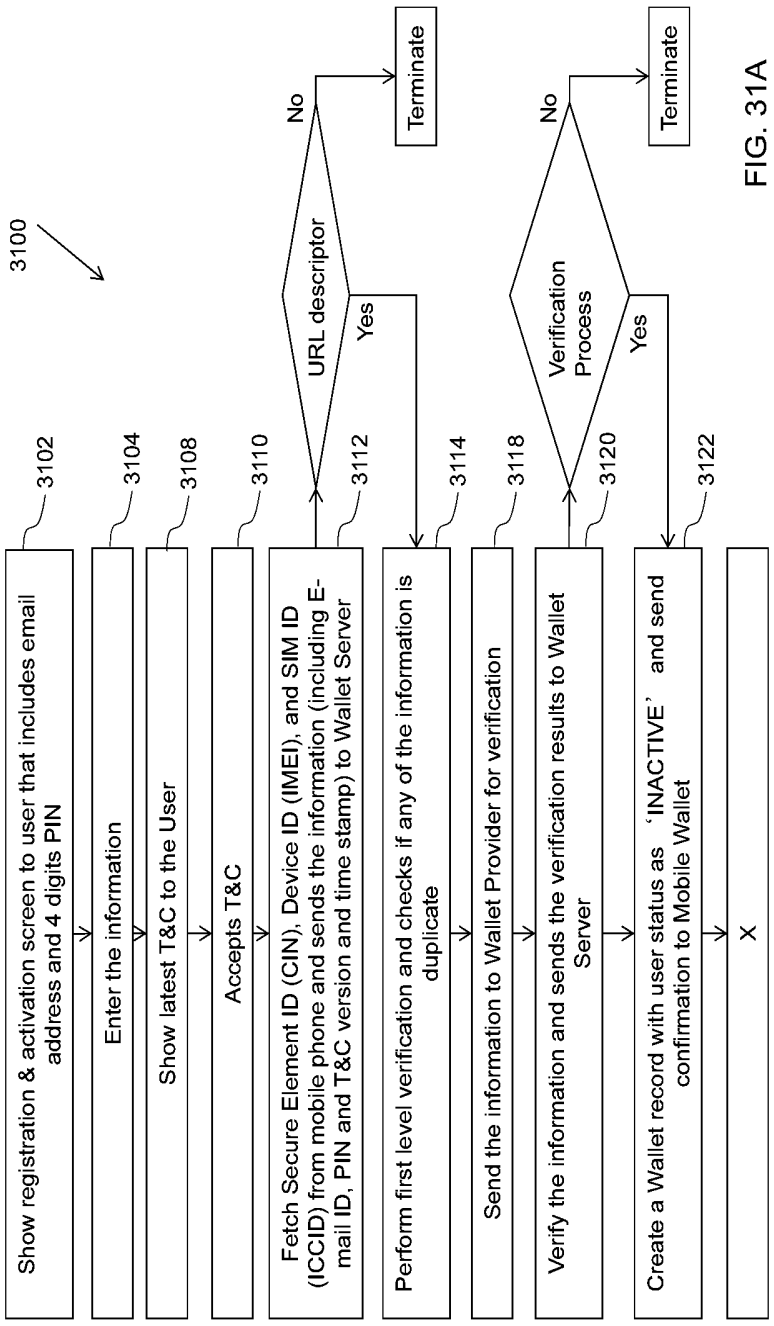

3100
3102
Show registration & activation screen to user that includes email address and 4 digits PIN
3104
Enter the information
3108
Show latest T&C to the User
3110
Accepts T&C
3112
Fetch Secure Element ID (CIN), Device ID (IMEI), and SIM ID (ICCID) from mobile phone and sends the information (including E-mail ID, PIN and T&C version and time stamp) to Wallet Server
URL descriptor
No
Terminate
Yes
3114
Perform first level verification and checks if any of the information is duplicate
3118
Send the information to Wallet Provider for verification
3120
Verify the information and sends the verification results to Wallet Server
Verification Process
No
Terminate
Yes
3122
Create a Wallet record with user status as 'INACTIVE' and send confirmation to Mobile Wallet
X

FIG. 31A

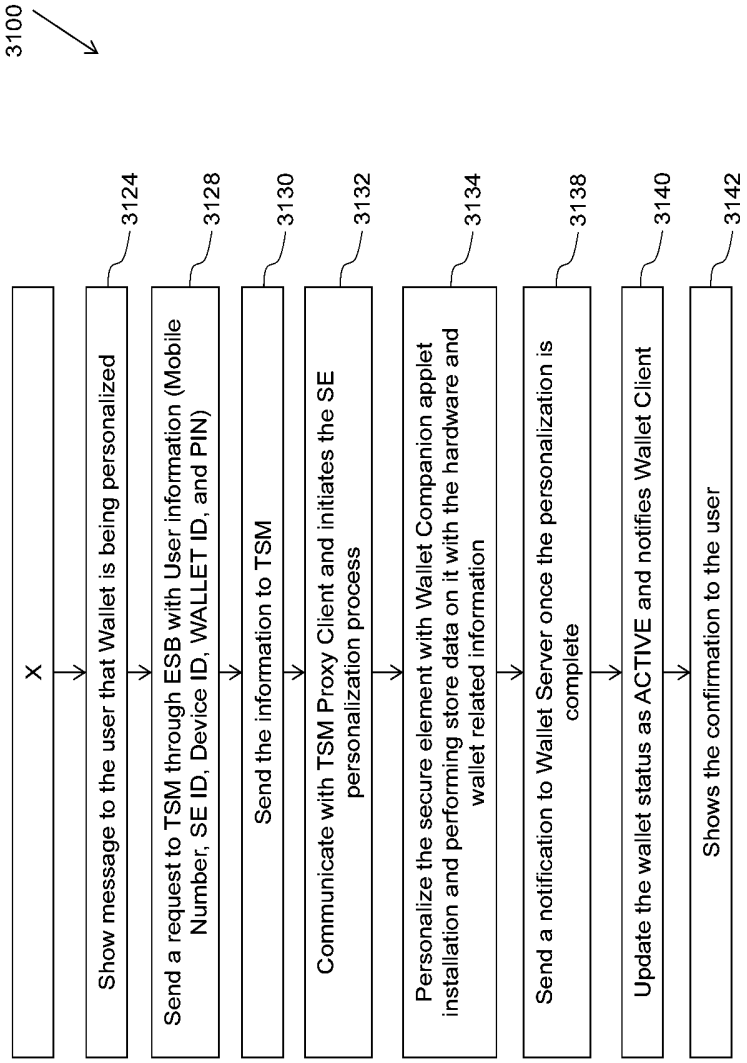

3100
X
Show message to the user that Wallet is being personalized
3124
Send a request to TSM through ESB with User information (Mobile Number, SE ID, Device ID, WALLET ID, and PIN)
3128
Send the information to TSM
3130
Communicate with TSM Proxy Client and initiates the SE personalization process
3132
Personalize the secure element with Wallet Companion applet installation and performing store data on it with the hardware and wallet related information
3134
Send a notification to Wallet Server once the personalization is complete
3138
Update the wallet status as ACTIVE and notifies Wallet Client
3140
Shows the confirmation to the user
3142

FIG. 31B

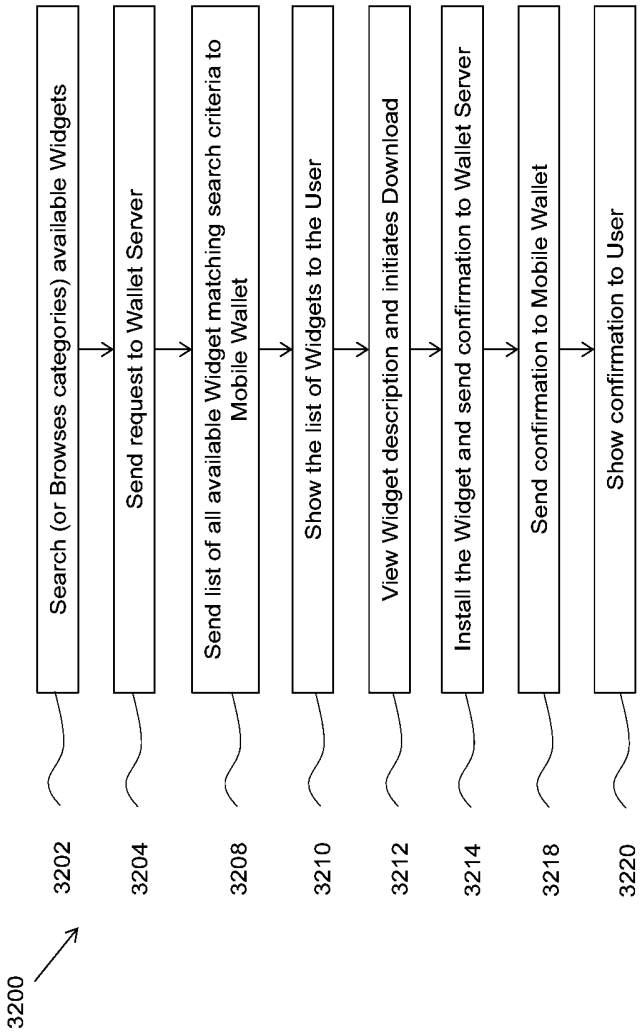

3200
Search (or Browses categories) available Widgets
3202
Send request to Wallet Server
3204
Send list of all available Widget matching search criteria to Mobile Wallet
3208
Show the list of Widgets to the User
3210
View Widget description and initiates Download
3212
Install the Widget and send confirmation to Wallet Server
3214
Send confirmation to Mobile Wallet
3218
Show confirmation to User
3220

FIG. 32

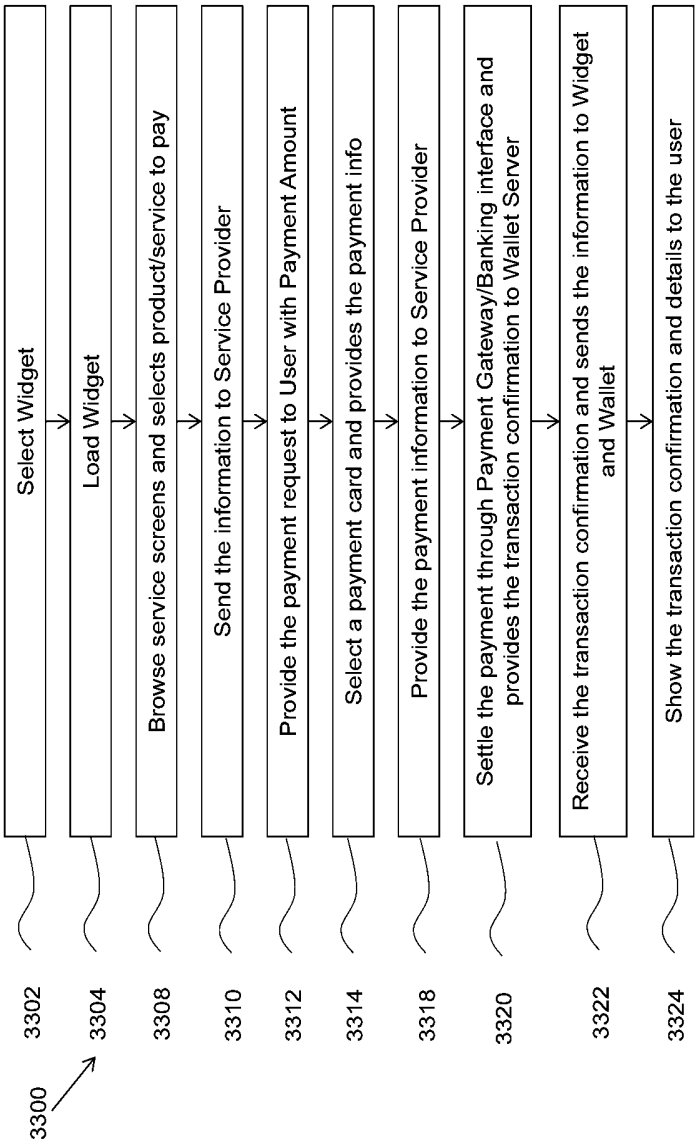

3300
3302
Select Widget
3304
Load Widget
3308
Browse service screens and selects product/service to pay
3310
Send the information to Service Provider
3312
Provide the payment request to User with Payment Amount
3314
Select a payment card and provides the payment info
3318
Provide the payment information to Service Provider
3320
Settle the payment through Payment Gateway/Banking interface and provides the transaction confirmation to Wallet Server
3322
Receive the transaction confirmation and sends the information to Widget and Wallet
3324
Show the transaction confirmation and details to the user

FIG. 33

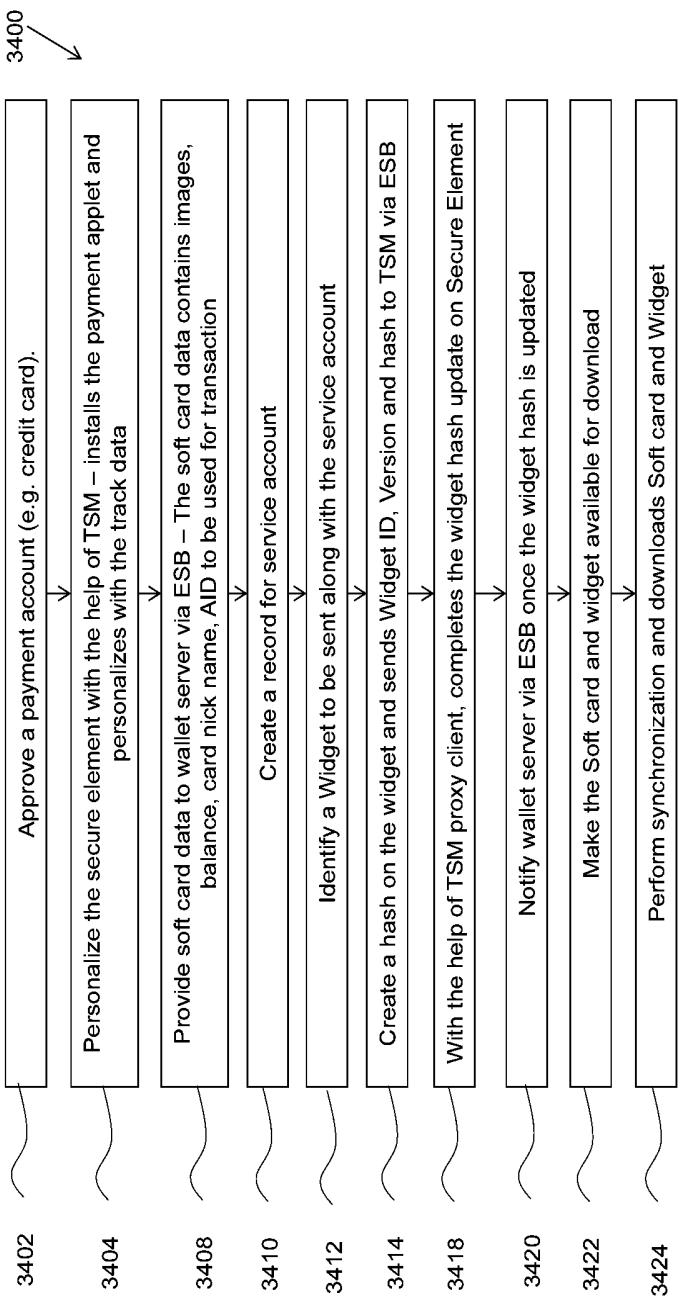

3400
Approve a payment account (e.g. credit card).
Personalize the secure element with the help of TSM – installs the payment applet and personalizes with the track data
Provide soft card data to wallet server via ESB – The soft card data contains images, balance, card nick name, AID to be used for transaction
Create a record for service account
Identify a Widget to be sent along with the service account
Create a hash on the widget and sends Widget ID, Version and hash to TSM via ESB
With the help of TSM proxy client, completes the widget hash update on Secure Element
Notify wallet server via ESB once the widget hash is updated
Make the Soft card and widget available for download
Perform synchronization and downloads Soft card and Widget
3402
3404
3408
3410
3412
3414
3418
3420
3422
3424

FIG. 34

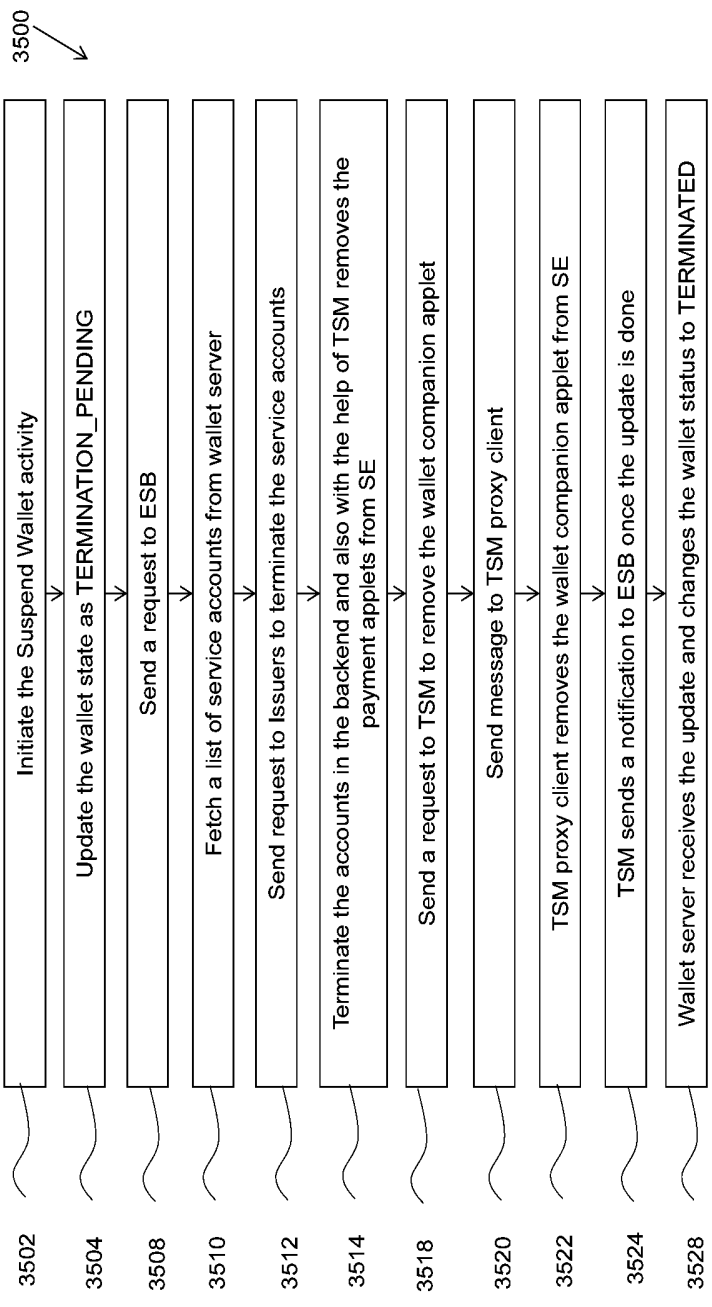

3500
Initiate the Suspend Wallet activity
Update the wallet state as TERMINATION_PENDING
Send a request to ESB
Fetch a list of service accounts from wallet server
Send request to Issuers to terminate the service accounts
Terminate the accounts in the backend and also with the help of TSM removes the payment applets from SE
Send a request to TSM to remove the wallet companion applet
Send message to TSM proxy client
TSM proxy client removes the wallet companion applet from SE
TSM sends a notification to ESB once the update is done
Wallet server receives the update and changes the wallet status to TERMINATED
3502
3504
3508
3510
3512
3514
3518
3520
3522
3524
3528

FIG. 35

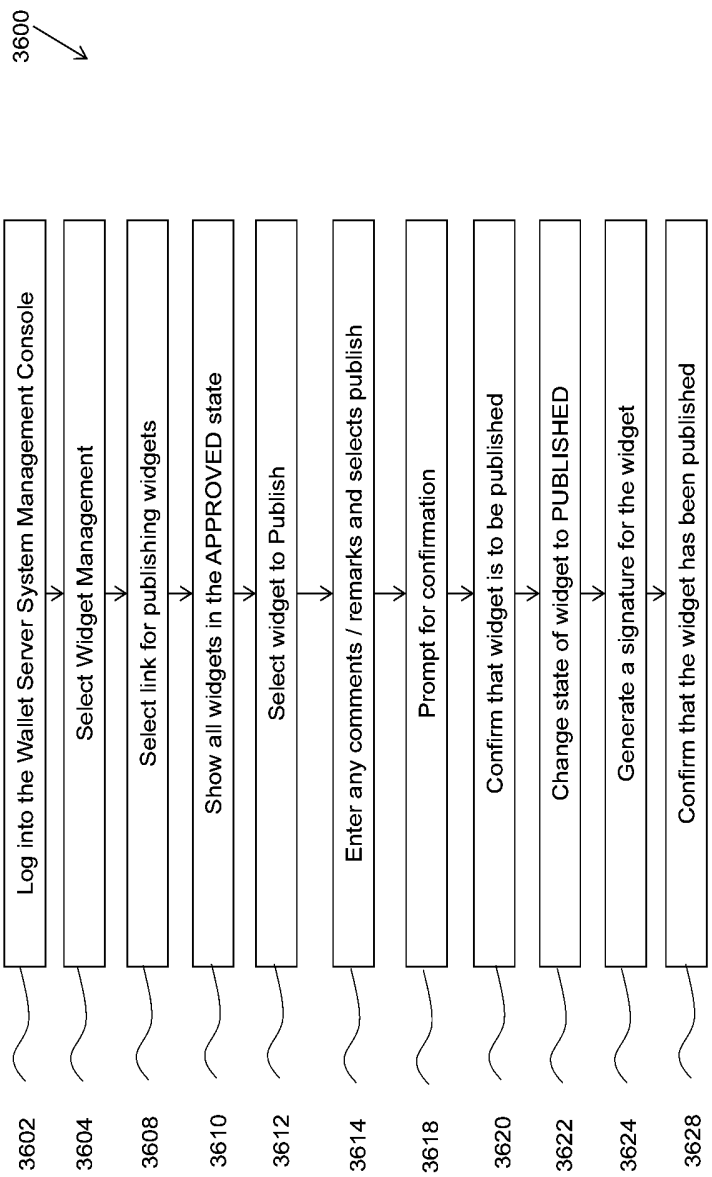

3600
Log into the Wallet Server System Management Console
Select Widget Management
Select link for publishing widgets
Show all widgets in the APPROVED state
Select widget to Publish
Enter any comments / remarks and selects publish
Prompt for confirmation
Confirm that widget is to be published
Change state of widget to PUBLISHED
Generate a signature for the widget
Confirm that the widget has been published
3602
3604
3608
3610
3612
3614
3618
3620
3622
3624
3628

FIG. 36

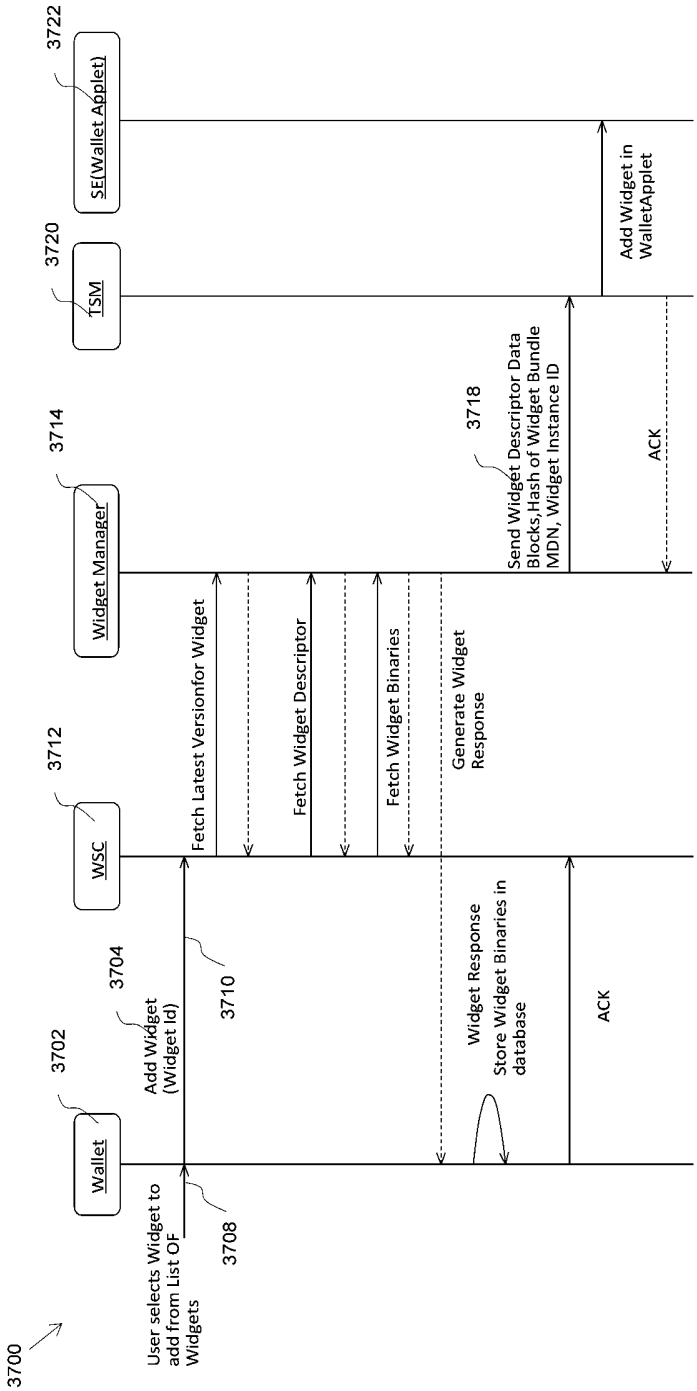

3700
User selects Widget to add from List OF Widgets
3708
Wallet
3702
Add Widget (Widget Id)
3704
3710
WSC
3712
Fetch Latest Versionfor Widget
Fetch Widget Descriptor
Fetch Widget Binaries
Generate Widget Response
Widget Response
Store Widget Binaries in database
Widget Manager
3714
Send Widget Descriptor Data Blocks,Hash of Widget Bundle MDN, Widget Instance ID
3718
ACK
TSM
3720
Add Widget in WalletApplet
SE(Wallet Applet)
3722
ACK

FIG. 37

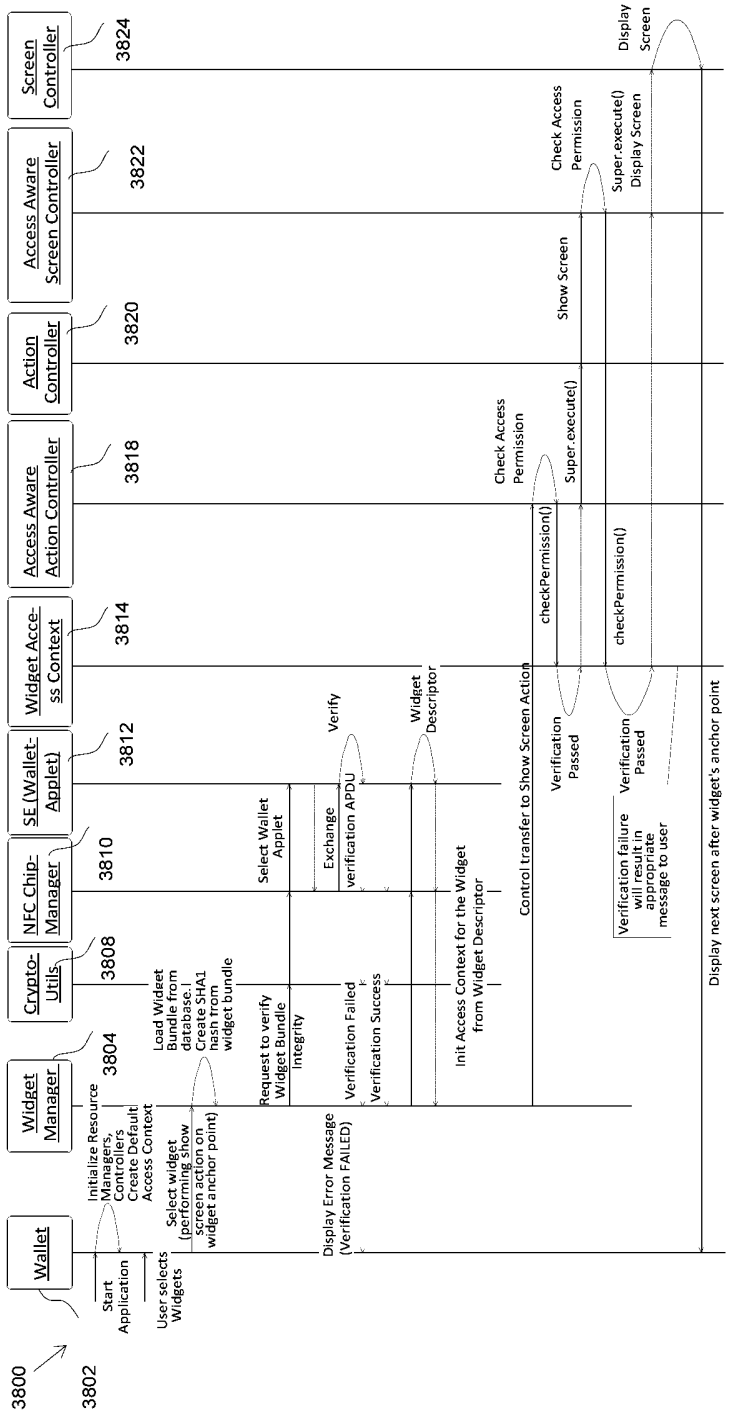

3800
3802
Wallet
Widget Manager
3804
Crypto-Utils
3808
NFC Chip-Manager
3810
SE (Wallet-Applet)
3812
Widget Acce-ss Context
3814
Access Aware Action Controller
3818
Action Controller
3820
Access Aware Screen Controller
3822
Screen Controller
3824
Start Application
Initialize Resource Managers, Controllers Create Default Access Context
User selects Widgets
Select widget (performing show screen action on widget anchor point)
Load Widget Bundle from database. Create SHA1 hash from widget bundle
Request to verify Widget Bundle Integrity
Select Wallet Applet
Exchange
verification APDU
Verify
Verification Failed
Verification Success
Widget Descriptor
Display Error Message (Verification FAILED)
Init Access Context for the Widget from Widget Descriptor
Control transfer to Show Screen Action
checkPermission()
Check Access Permission
Verification Passed
Super.execute()
Show Screen
checkPermission()
Check Access Permission
Verification Passed
Super.execute()
Display Screen
Verification failure will result in appropriate message to user
Display next screen after widget's anchor point

FIG. 38

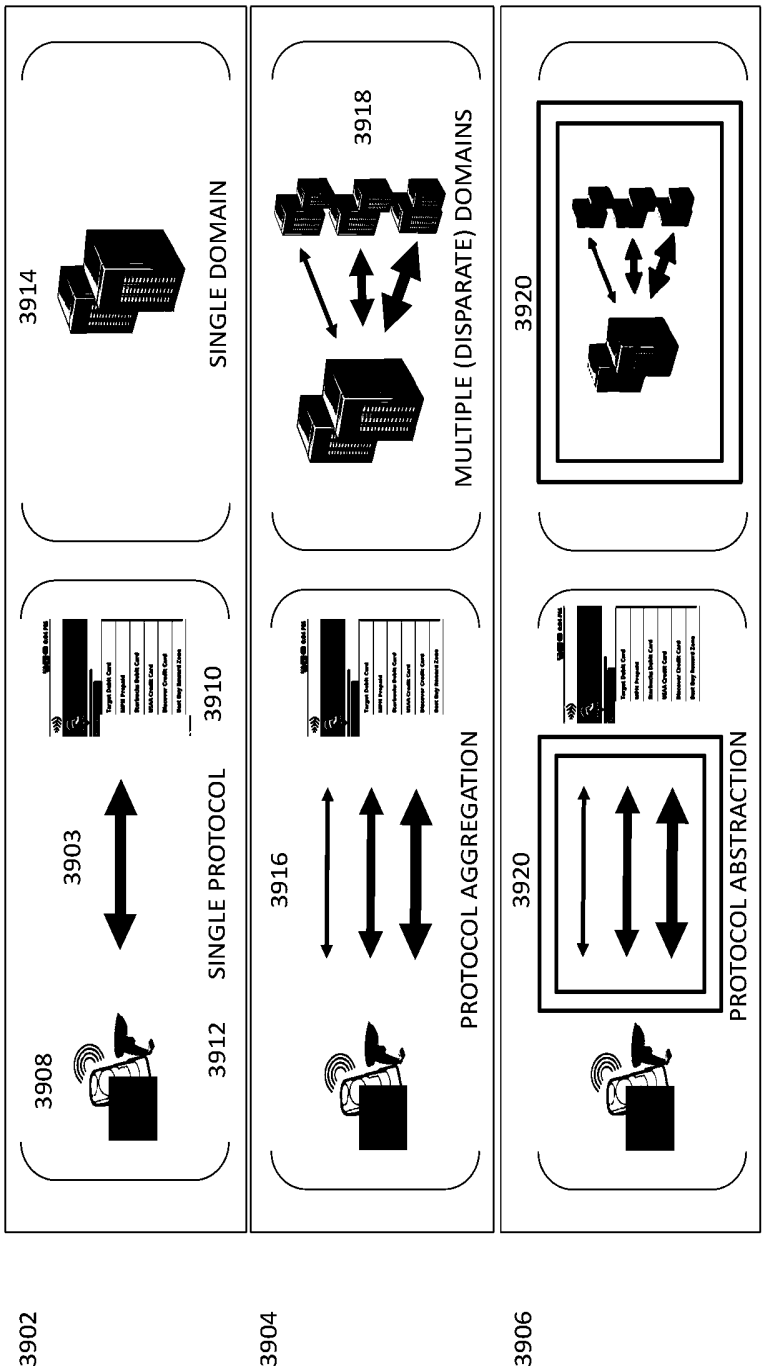

3902
3904
3906
3908
3912
SINGLE PROTOCOL
3903
3910
3914
SINGLE DOMAIN
3916
PROTOCOL AGGREGATION
3918
MULTIPLE (DISPARATE) DOMAINS
3920
PROTOCOL ABSTRACTION

FIG. 39

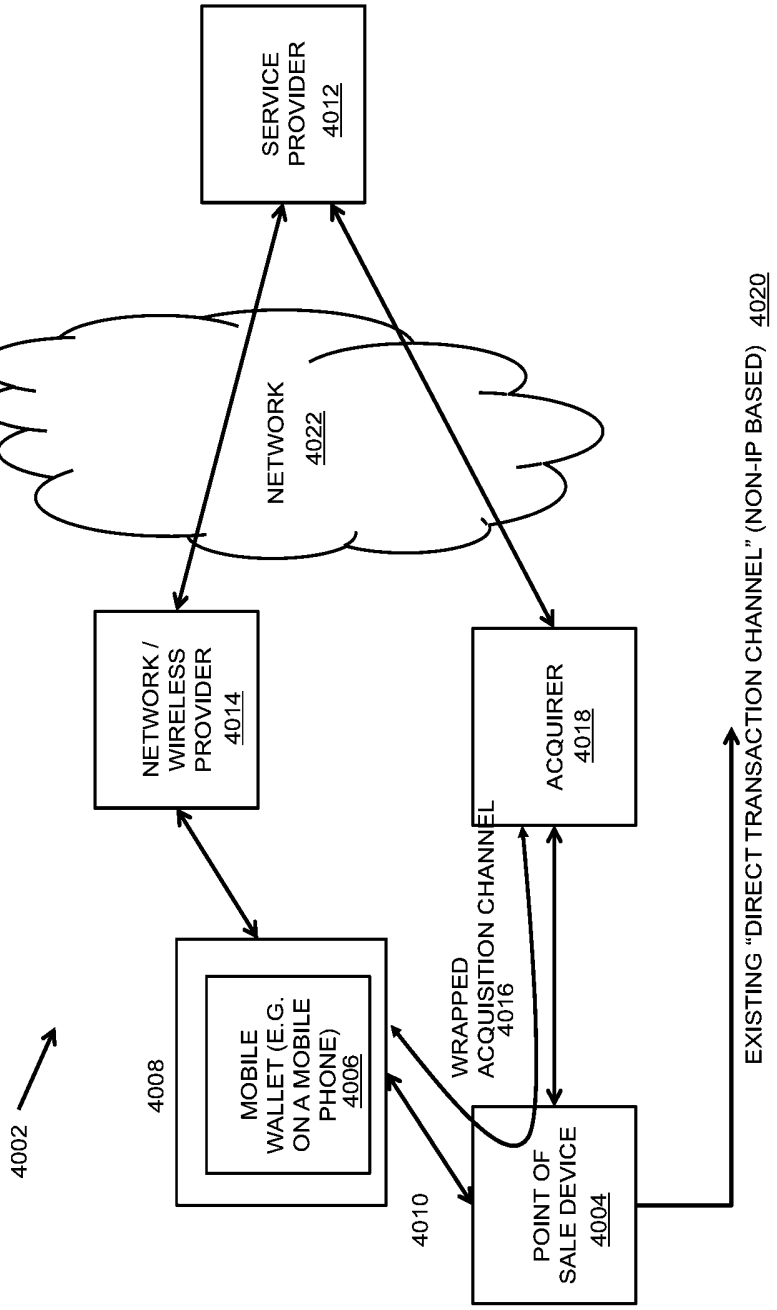

SERVICE PROVIDER 4012
NETWORK 4022
NETWORK / WIRELESS PROVIDER 4014
ACQUIRER 4018
4002
4008
MOBILE WALLET (E.G. ON A MOBILE PHONE) 4006
WRAPPED ACQUISITION CHANNEL 4016
4010
POINT OF SALE DEVICE 4004
EXISTING "DIRECT TRANSACTION CHANNEL" (NON-IP BASED) 4020

FIG. 40

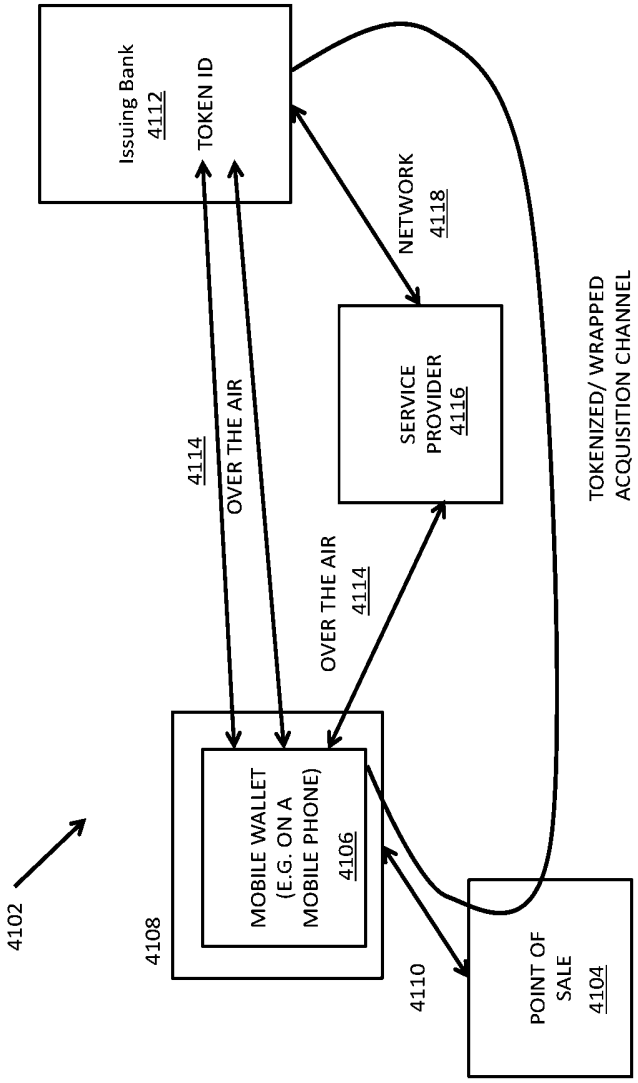

Issuing Bank
4112
TOKEN ID
NETWORK
4118
SERVICE
PROVIDER
4116
4114
OVER THE AIR
OVER THE AIR
4114
TOKENIZED/ WRAPPED
ACQUISITION CHANNEL
4120
4102
4108
MOBILE WALLET
(E.G. ON A
MOBILE PHONE)
4106
4110
POINT OF
SALE
4104

FIG. 41

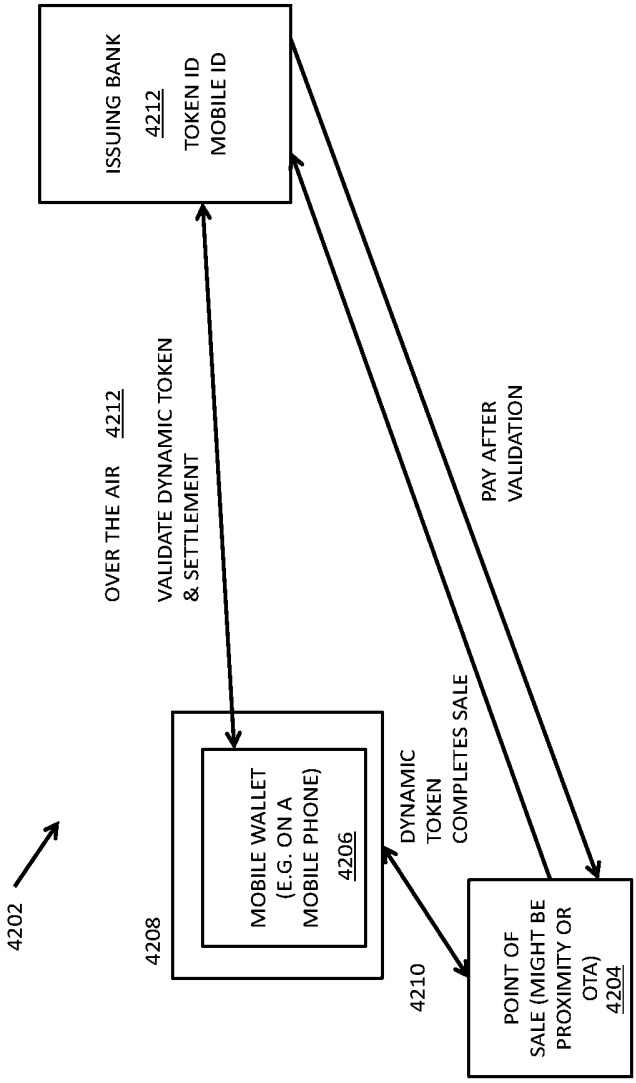

ISSUING BANK
4212
TOKEN ID
MOBILE ID
OVER THE AIR 4212
VALIDATE DYNAMIC TOKEN
& SETTLEMENT
PAY AFTER
VALIDATION
4202
4208
MOBILE WALLET
(E.G. ON A
MOBILE PHONE)
4206
DYNAMIC
TOKEN
COMPLETES SALE
4210
POINT OF
SALE (MIGHT BE
PROXIMITY OR
OTA)
4204

FIG. 42

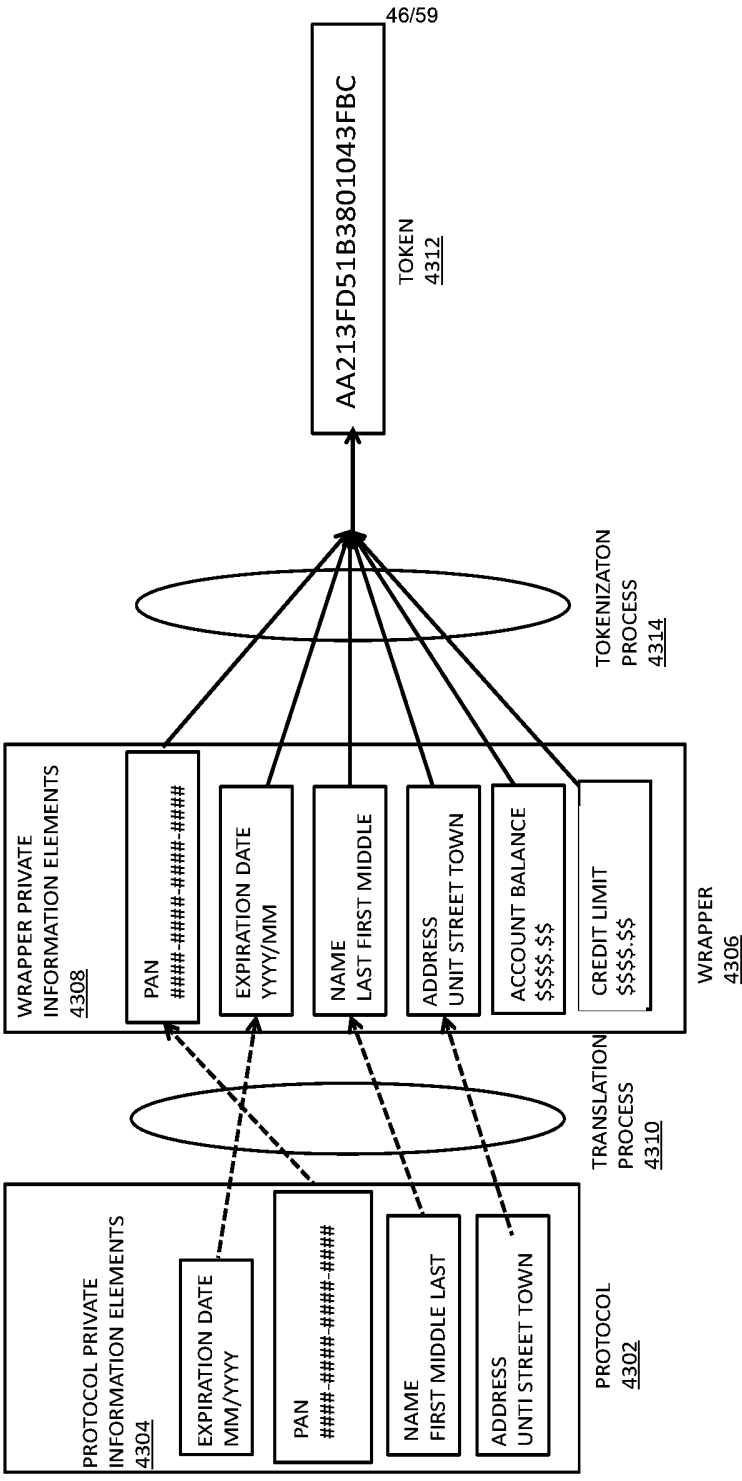

PROTOCOL PRIVATE INFORMATION ELEMENTS 4304
EXPIRATION DATE MM/YYYY
PAN ####-####-####-####
NAME FIRST MIDDLE LAST
ADDRESS UNTI STREET TOWN
PROTOCOL 4302
TRANSLATION PROCESS 4310
WRAPPER PRIVATE INFORMATION ELEMENTS 4308
PAN ####-####-####-####
EXPIRATION DATE YYYY/MM
NAME LAST FIRST MIDDLE
ADDRESS UNIT STREET TOWN
ACCOUNT BALANCE $$$$.$$
CREDIT LIMIT $$$$.$$
WRAPPER 4306
TOKENIZATON PROCESS 4314
AA213FD51B3801043FBC
TOKEN 4312

FIG. 43

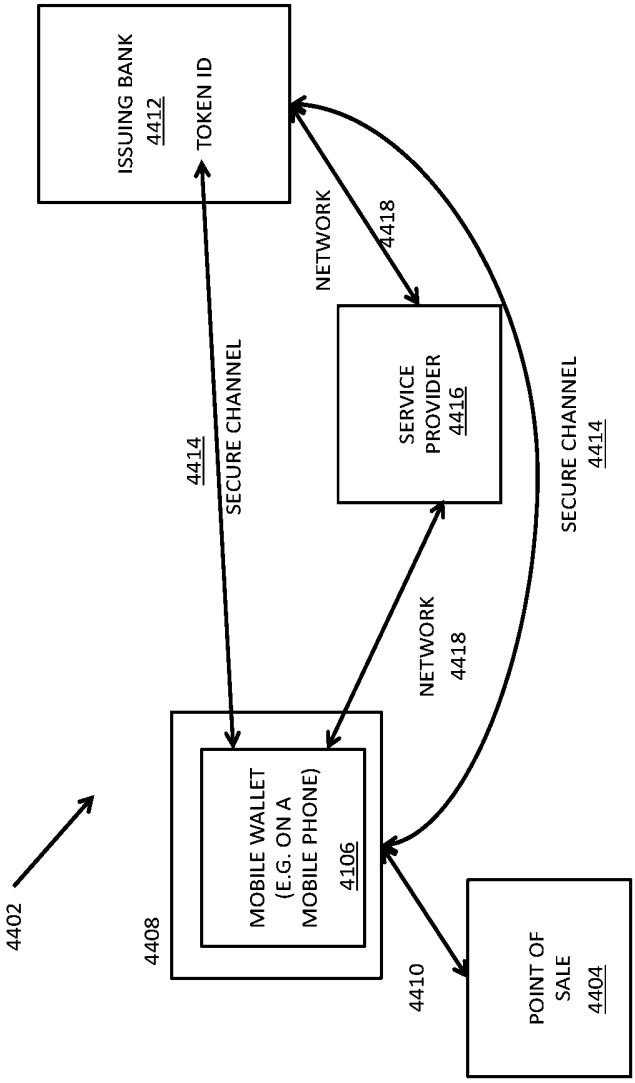

ISSUING BANK
4412
TOKEN ID
NETWORK
4418
SERVICE
PROVIDER
4416
4414
SECURE CHANNEL
SECURE CHANNEL
4414
NETWORK
4418
MOBILE WALLET
(E.G. ON A
MOBILE PHONE)
4106
4402
4408
4410
POINT OF
SALE
4404

FIG. 44

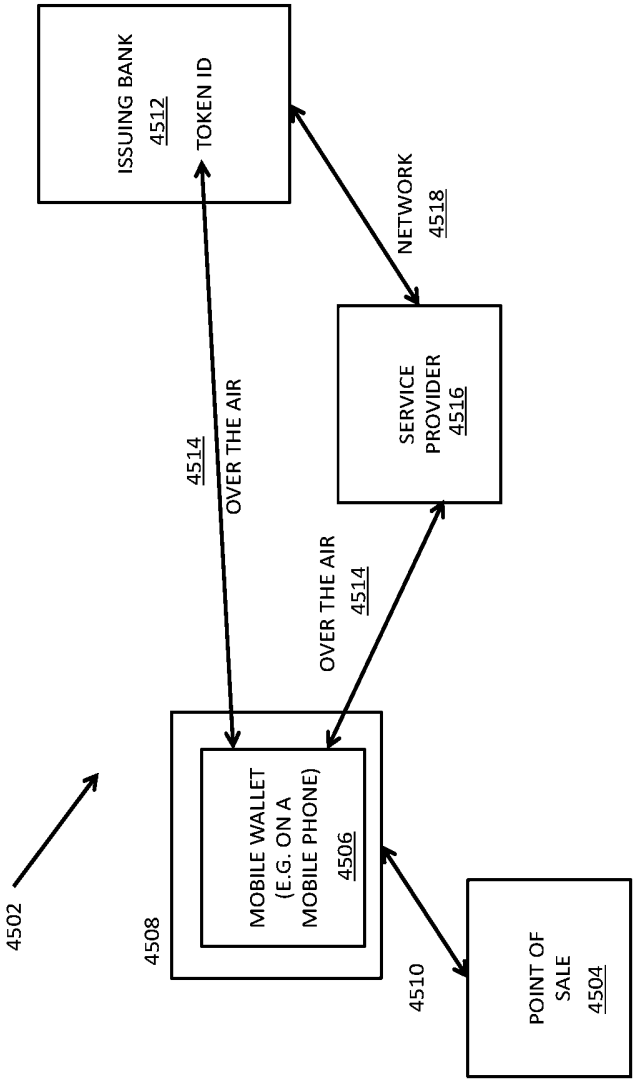

ISSUING BANK
4512
TOKEN ID
NETWORK
4518
SERVICE
PROVIDER
4516
4514
OVER THE AIR
OVER THE AIR
4514
4502
4508
MOBILE WALLET
(E.G. ON A
MOBILE PHONE)
4506
4510
POINT OF
SALE
4504

FIG. 45

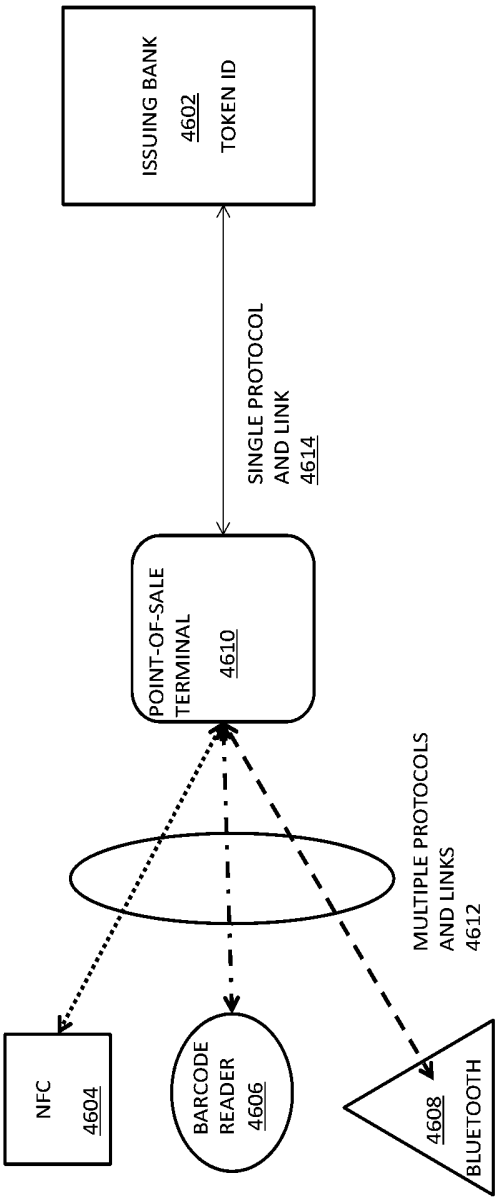

ISSUING BANK
4602
TOKEN ID
SINGLE PROTOCOL
AND LINK
4614
POINT-OF-SALE
TERMINAL
4610
MULTIPLE PROTOCOLS
AND LINKS
4612
NFC
4604
BARCODE
READER
4606
4608
BLUETOOTH

FIG. 46

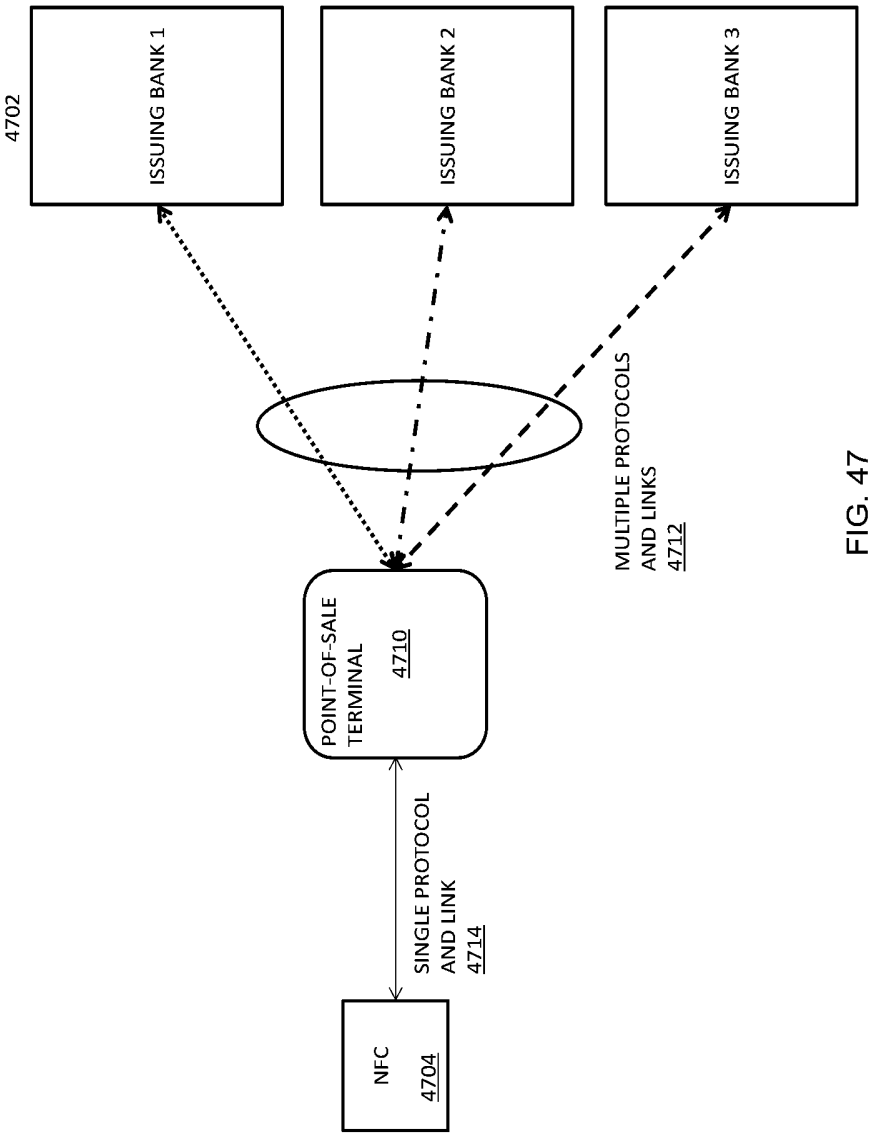

4702
ISSUING BANK 1
ISSUING BANK 2
ISSUING BANK 3
MULTIPLE PROTOCOLS AND LINKS 4712
POINT-OF-SALE TERMINAL 4710
SINGLE PROTOCOL AND LINK 4714
NFC 4704

FIG. 47

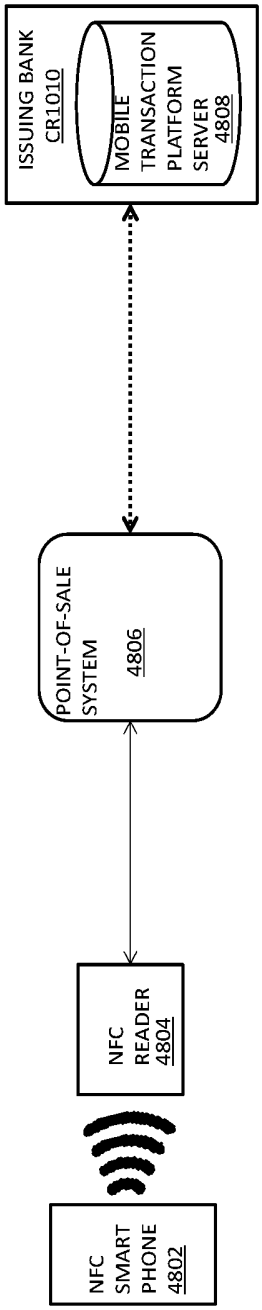

ISSUING BANK
CR1010
MOBILE
TRANSACTION
PLATFORM
SERVER
4808
POINT-OF-SALE
SYSTEM
4806
NFC
READER
4804
NFC
SMART
PHONE
4802

FIG. 48

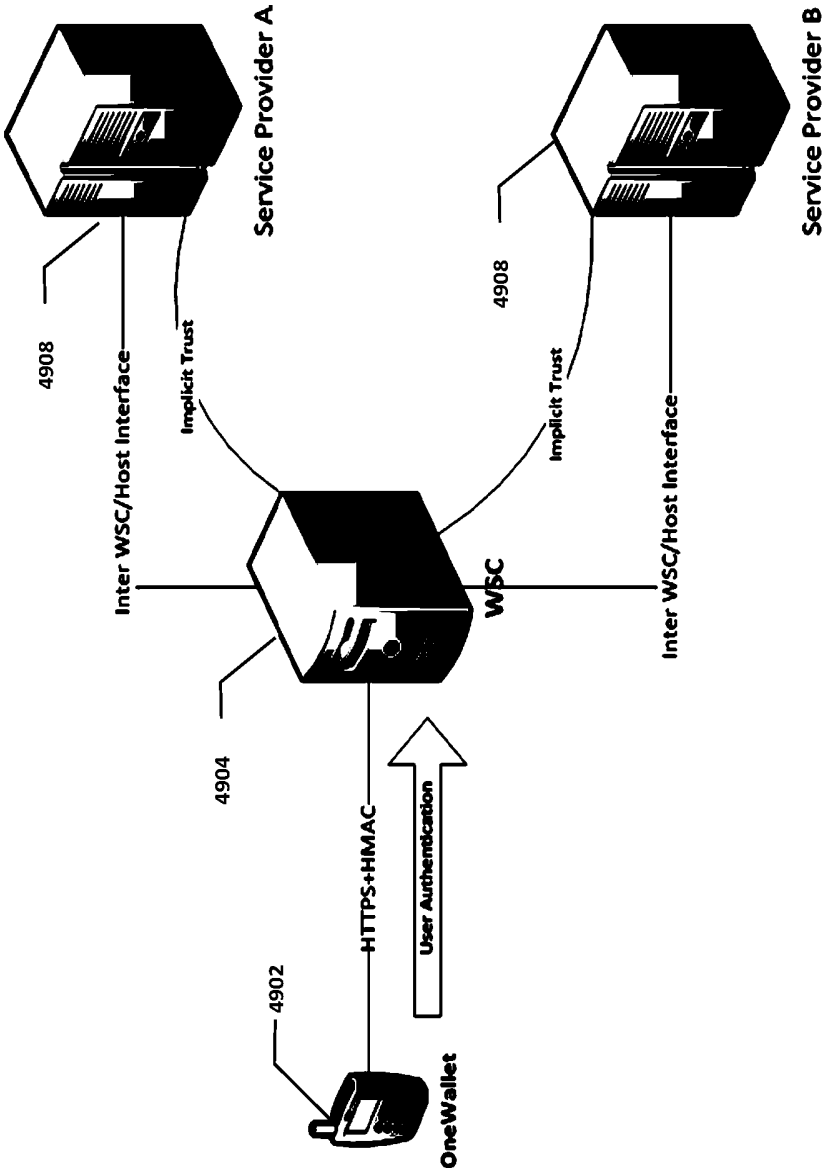

Service Provider A
Service Provider B
4908
4908
Implicit Trust
Implicit Trust
Inter WSC/Host Interface
Inter WSC/Host Interface
WSC
4904
HTTPS+HMAC
User Authentication
4902
OneWallet

FIG. 49

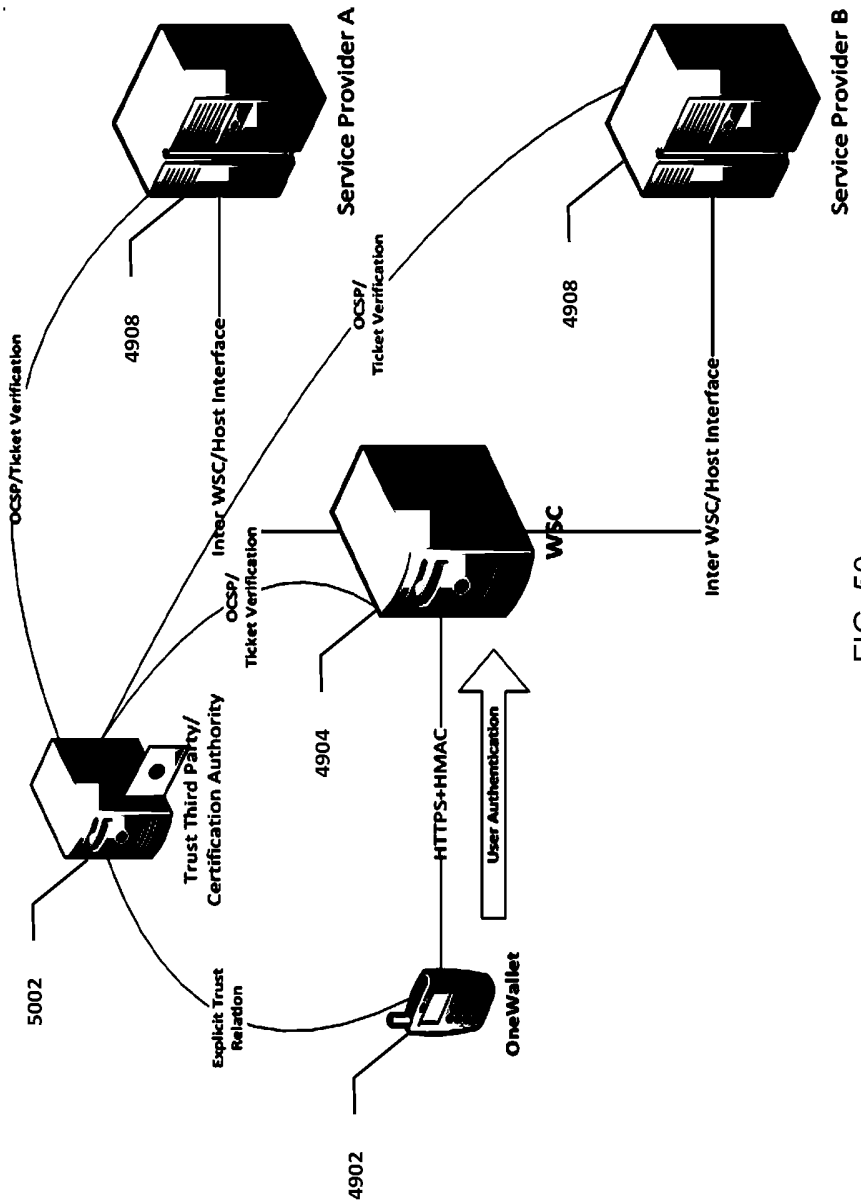

Service Provider A
Service Provider B
WSC
Trust Third Party/
Certification Authority
OneWallet
OCSP/Ticket Verification
Inter WSC/Host Interface
HTTPS+HMAC
User Authentication
Explicit Trust
Relation
4902
4904
4908
5002

FIG. 50

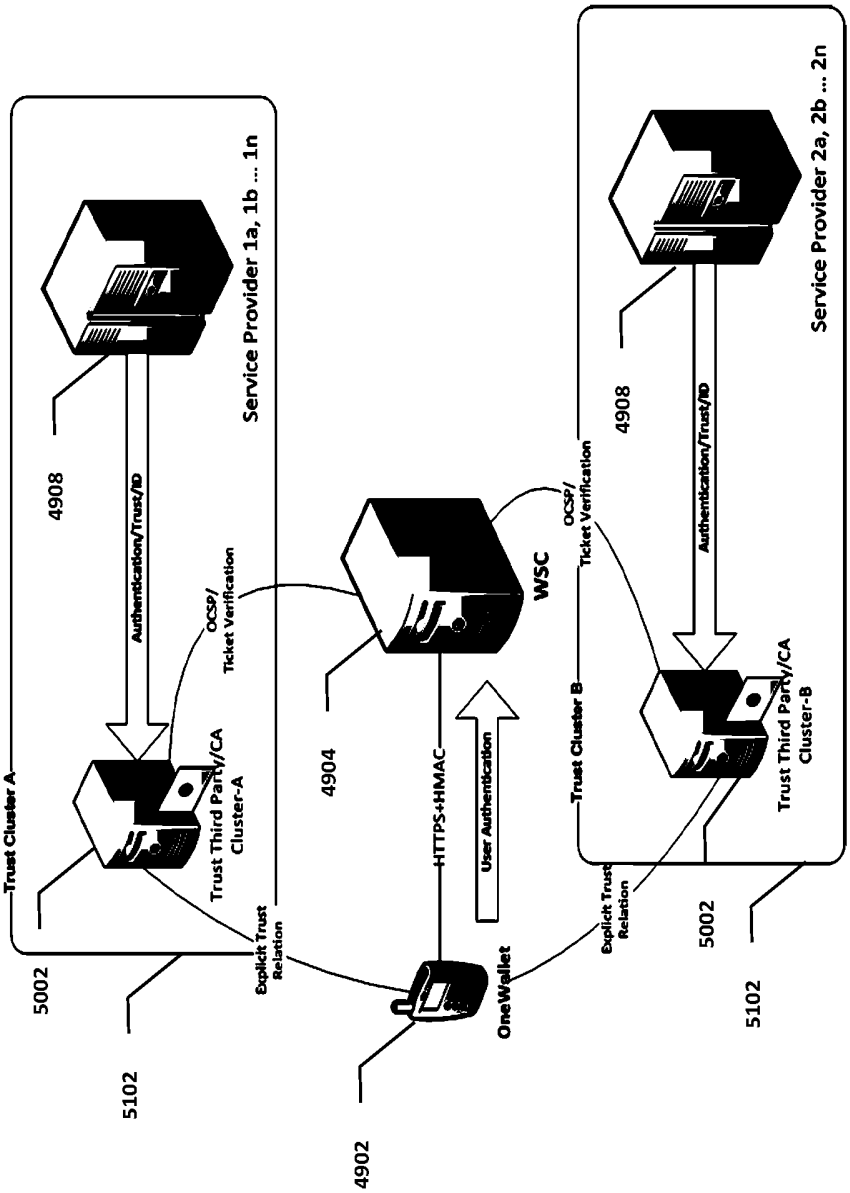

Trust Cluster A
Service Provider 1a, 1b ... 1n
4908
Authentication/Trust/ID
OCSP/
Ticket Verification
Trust Third Party/CA
Cluster-A
4904
WSC
HTTPS+HMAC
User Authentication
Explicit Trust
Relation
5002
5102
4902
OneWallet
Trust Cluster B
Service Provider 2a, 2b ... 2n
4908
Authentication/Trust/ID
OCSP/
Ticket Verification
Trust Third Party/CA
Cluster-B
Explicit Trust
Relation
5002
5102

FIG. 51

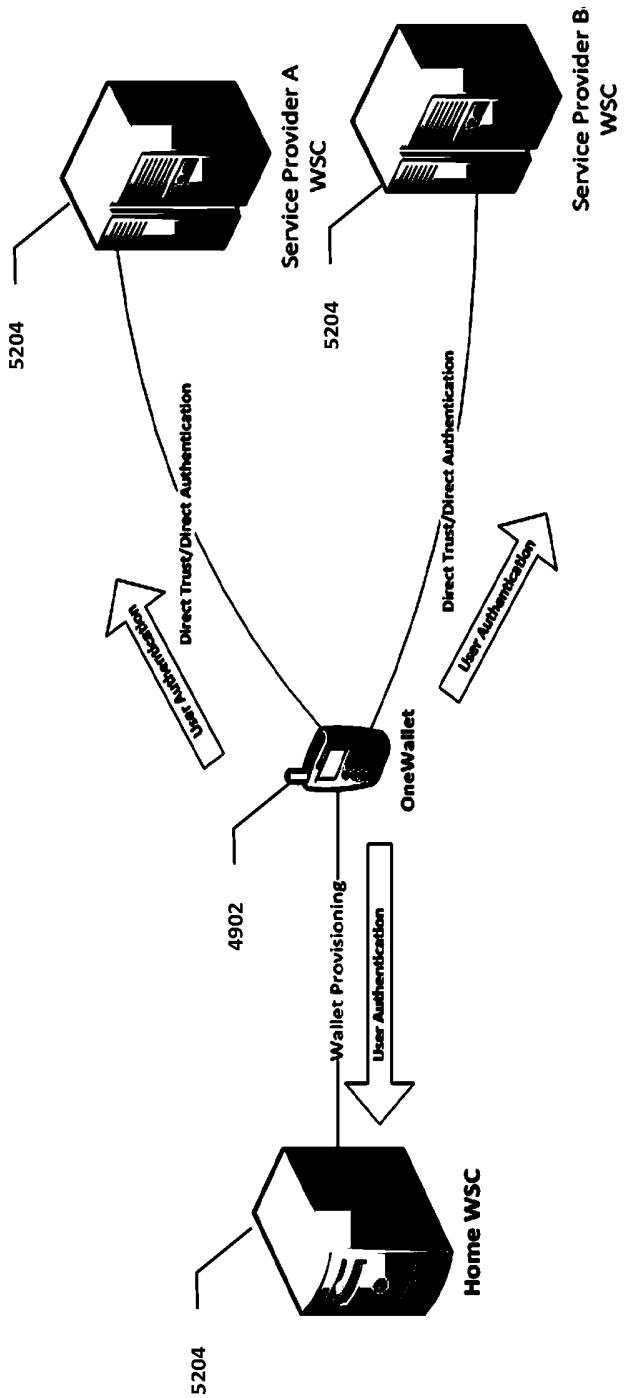

Service Provider A WSC
Service Provider B WSC
5204
5204
Direct Trust/Direct Authentication
Direct Trust/Direct Authentication
User Authentication
User Authentication
OneWallet
4902
Wallet Provisioning
User Authentication
Home WSC
5204

FIG. 52

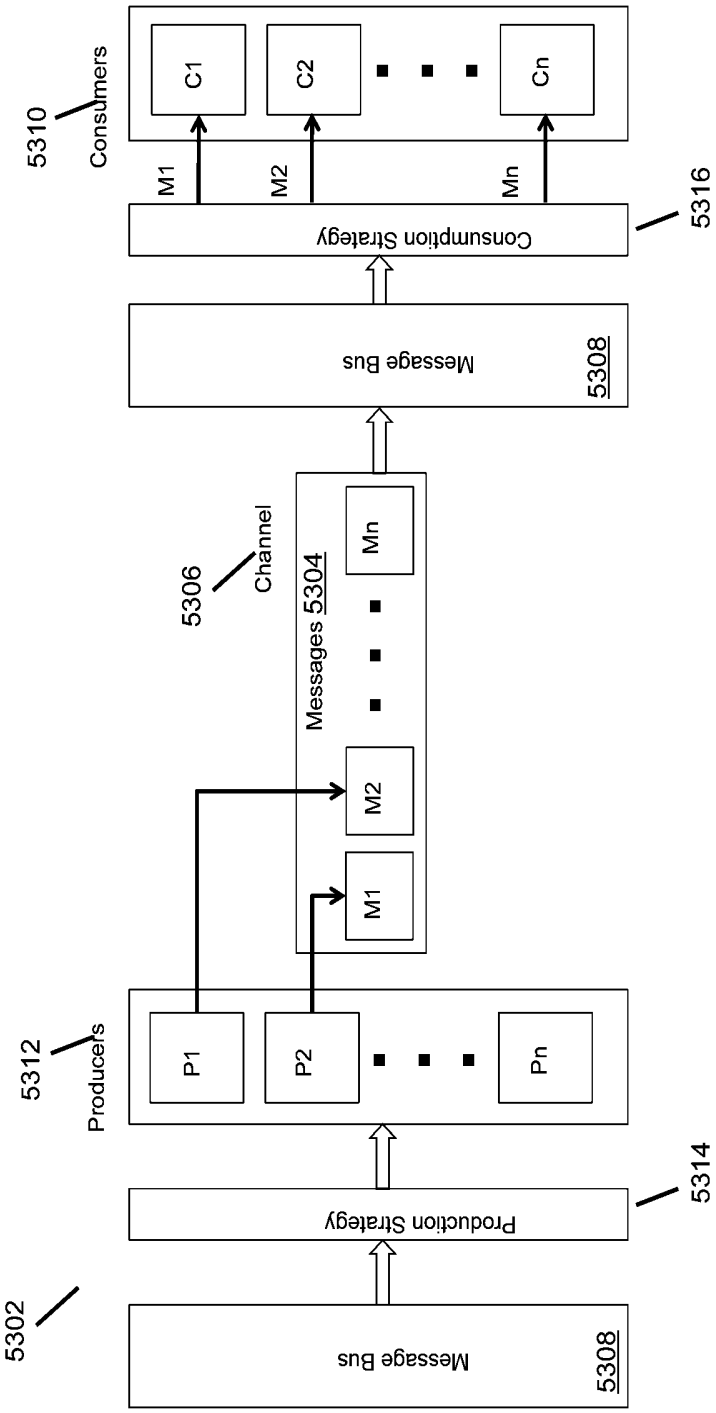

5302
Message Bus
5308
5314
Production Strategy
5312
Producers
P1
P2
Pn
5306
Channel
Messages 5304
M1
M2
Mn
Message Bus
5308
5316
Consumption Strategy
M1
M2
Mn
5310
Consumers
C1
C2
Cn

FIG. 53

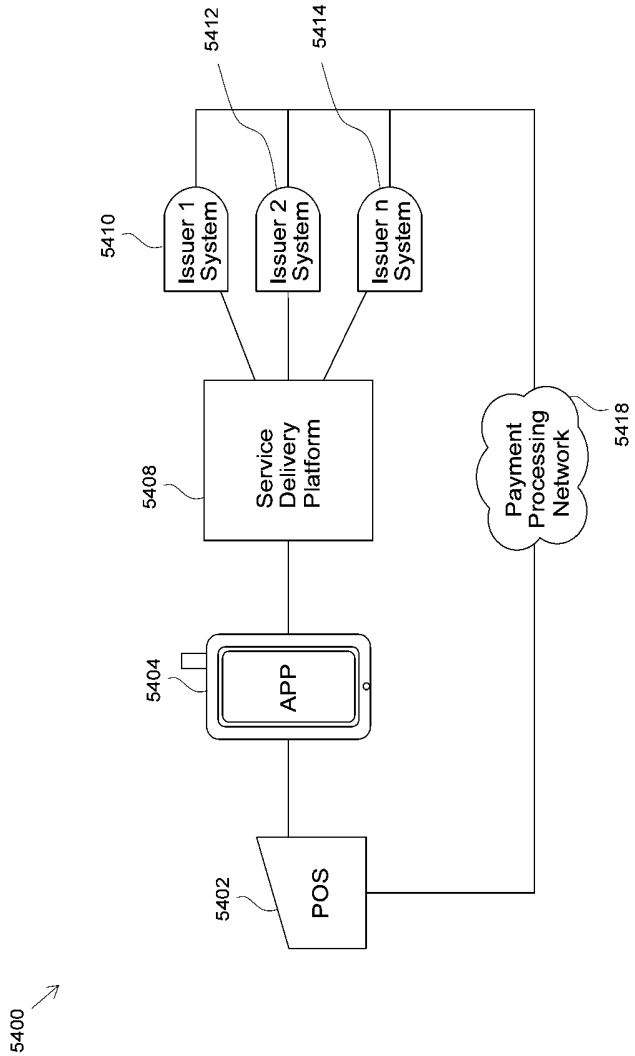

5400
5402
POS
5404
APP
5408
Service Delivery Platform
5410
Issuer 1 System
Issuer 2 System
Issuer n System
5412
5414
Payment Processing Network
5418

FIG. 54

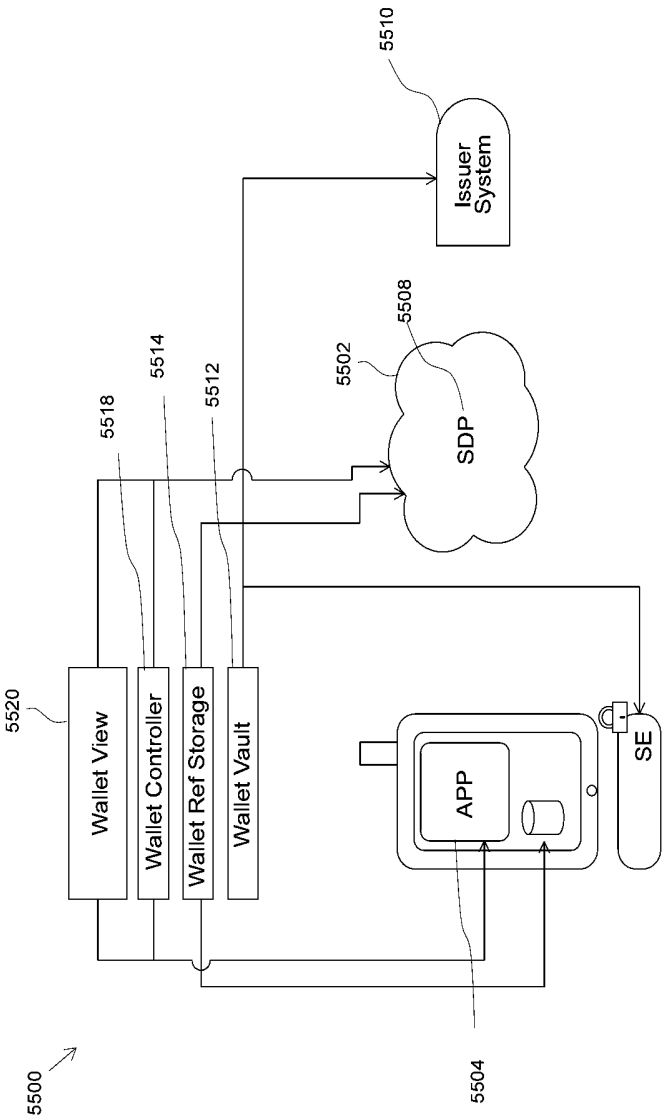

5510
Issuer System
5508
5502
SDP
5514
5512
5518
5520
Wallet View
Wallet Controller
Wallet Ref Storage
Wallet Vault
APP
SE
5500
5504

FIG. 55

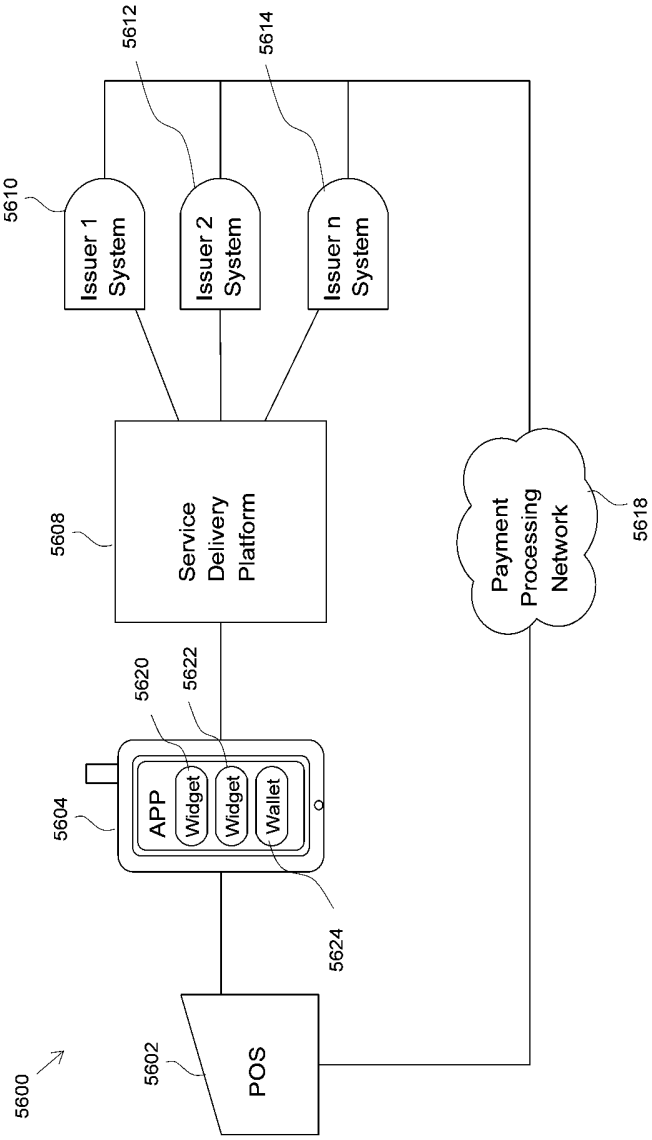

5612
5614
5610
Issuer 1 System
Issuer 2 System
Issuer n System
5608
Service Delivery Platform
Payment Processing Network
5618
5620
5622
5604
APP
Widget
Widget
Wallet
5624
5600
5602
POS

FIG. 56

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2012/060038

A. CLASSIFICATION OF SUBJECT MATTER

H04W 12/00(2009.01)i, G06F 9/44(2006.01)i, G06Q 40/02(2012.01)i, G06Q 20/40(2012.01)i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

H04W 12/00; G06F 15/16; G06F 9/44; G06F 17/30; B62D 39/00; G06F 12/14; G06Q 40/00; H04L 9/00; G06F 9/45

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Korean utility models and applications for utility models
Japanese utility models and applications for utility models

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

eKOMPASS(KIPO internal) & Keywords:secure,transaction,service

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2007-0130252 A1 (CURRIE PETER BOYLE et al.) 07 June 2007	65
A	See the abstract; claim 1, 6, 11; Figure 1-4.	1-64,66-181
X	US 2009-0307300 A1 (ISAAC DAVID GUEDALIA et al.) 10 December 2009	101
A	See the abstract; paragraphs [0005]-[0006]; claim 1, 10.	1-100,102-181
X	US 2008-0059375 A1 (BASIL MUNIR ABIFAKER) 06 March 2008	115-116,118-122
A	See paragraphs [0005]-[0015]; claims 1-5; Figure 1-3.	1-114,117,123-181
X	WO 2006-074258 A2 (REALM SYSTEMS, INC. et al.) 13 July 2006	150
A	See Figure 6.	1-149,151-181
X	US 2010-0269096 A1 (CARLOS ARAYA et al.) 21 October 2010	160
A	See paragraphs [0036]-[0037]; claim 5.	1-159,161-181
X	US 2001-0018746 A1 (ALONG LIN) 30 August 2001	162
A	See paragraphs [0024]-[0025], [0029]-[0030]; claims 1-5.	1-161,163-181

☐ Further documents are listed in the continuation of Box C.

☒ See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents,such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
27 March 2013 (27.03.2013)	**28 March 2013 (28.03.2013)**

Name and mailing address of the ISA/KR

Korean Intellectual Property Office
189 Cheongsa-ro, Seo-gu, Daejeon Metropolitan City, 302-701, Republic of Korea

Facsimile No. 82-42-472-7140

Authorized officer

SUNG, In Gu

Telephone No. 82-42-481-8485

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/US2012/060038

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2007-0130252 A1	07.06.2007	CA 2528866 A1	06.06.2007
		US 8015021 B2	06.09.2011
US 2009-0307300 A1	10.12.2009	None	
US 2008-0059375 A1	06.03.2008	None	
WO 2006-074258 A2	13.07.2006	WO 2006-074258 A3	16.08.2007
US 2010-0269096 A1	21.10.2010	None	
US 2001-0018746 A1	30.08.2001	DE 60015709 D1	16.12.2004
		DE 60015709 T2	10.11.2005
		EP 1118925 A1	25.07.2001
		EP 1118925 B1	10.11.2004
		US 6941471 B2	06.09.2005

(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 104106276 A
(43)申请公布日 2014.10.15

(21)申请号 201280061249.2
(22)申请日 2012.10.12
(30)优先权数据
61/546084 2011.10.12 US
61/619751 2012.04.03 US
(85)PCT国际申请进入国家阶段日
2014.06.12
(86)PCT国际申请的申请数据
PCT/US2012/060038 2012.10.12
(87)PCT国际申请的公布数据
WO2013/056104 EN 2013.04.18
(71)申请人 施萨姆公司
地址 美国伊利诺伊州
(72)发明人 M. 德赛 萨特延·皮特罗达
内哈·马尼亚尔
(74)专利代理机构 中国专利代理(香港)有限公司 72001
代理人 臧永杰 陈岚
(51)Int.Cl.
H04W 12/00(2006.01)
G06F 9/44(2006.01)
G06Q 40/02(2006.01)
G06Q 20/40(2006.01)

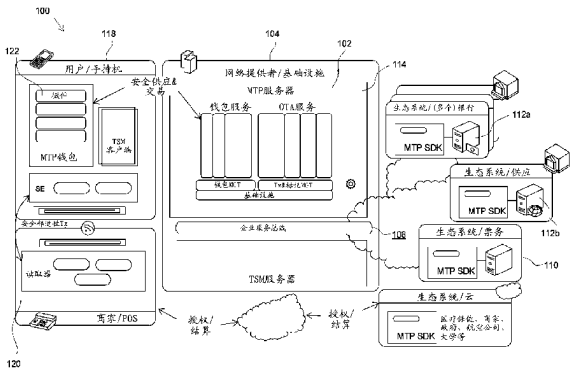
权利要求书15页 说明书92页 附图59页

(54)发明名称

多层安全移动交易使能平台

(57)摘要

一种用于在多域生态系统中执行安全个性化交易的平台包括个性化层，其使能对于存储在移动设备上的一个或多个生态系统元件的服务提供者个性化。此外，该平台包括使能层，用于促进个性化层与客户端设备之间的互操作。该平台此外包括服务层，其可以独立于使能层而操作，并且可以使能对于多个服务的服务递送。



CN 104106276 A

1. 一种用于多域生态系统中的安全个性化交易的平台，包括：

个性化层，用于使能存储在移动设备上的一个或多个生态系统元件的服务提供者个性化；

使能层，用于促进所述个性化层与客户端设备之间的互操作；以及

服务层，其独立于所述使能层而操作，用以使能多个服务的服务递送。

2. 如权利要求 1 所述的平台，其中，所述多个服务是从包括银行业务、票务和账单支付的组中选择的。

3. 如权利要求 1 所述的平台，其中，所述一个或多个生态系统元件是从包括微件、钱包和小程序的组中选择的。

4. 如权利要求 1 所述的平台，其中，所述平台是在生态系统参与者的分布式网络中可操作的，所述生态系统参与者包括服务提供者、器件发行者、器件获取者以及移动设备用户。

5. 如权利要求 1 所述的平台，其中，所述平台此外包括多个功能模块，所述多个功能模块包括用于以下各项的一个或多个模块：通信、通知、安全性、认证、空中操作以及用以管理钱包和微件的设备运行时容器。

6. 如权利要求 1 所述的平台，其中，所述平台此外包括软件开发环境，所述软件开发环境包括容器、钱包和微件的参考实现。

7. 如权利要求 1 所述的平台，其中，所述平台此外包括软件开发环境，所述软件开发环境包括金融、零售、健康和管控域的参考实现。

8. 如权利要求 1 所述的平台，其中，所述平台通过使用原生客户端应用来部署。

9. 如权利要求 1 所述的平台，其中，所述平台通过使用原生客户端应用和一组软件开发工具来部署。

10. 如权利要求 9 所述的平台，其中，该组软件开发工具可以用于开发一组客户端功能。

11. 如权利要求 10 所述的平台，其中，该组客户端功能可以是从钱包、微件、伴随小程序以及优惠券发放小程序的组中选择的。

12. 如权利要求 1 所述的平台，其中，所述平台通过使用原生客户端应用和一组 web 服务来部署，所述 web 服务由所述平台展露用以集成外围系统。

13. 如权利要求 12 所述的平台，其中，所述外围系统是从包括客户关怀和用户门户的组中选择的。

14. 如权利要求 1 所述的平台，其中，所述平台部署有用以开发钱包、微件和伴随应用的工具。

15. 如权利要求 15 所述的平台，其中，所述工具包括向导类实用程序。

16. 如权利要求 15 所述的平台，其中，所述工具可以被配置为监视使用和交易信息。

17. 如权利要求 15 所述的平台，其中，所述工具可以用于提供智能优化选项和解决方案。

18. 如权利要求 17 所述的平台，其中，所述优化选项和解决方案基于所监视的能力和性能准则。

19. 如权利要求 1 所述的平台，其中，所述平台部署有企业服务递送环境中的必要支

持。

20. 如权利要求 19 所述的平台，其中，所述必要支持包括对可伸缩性、冗余性、互操作性、服务递送以及消息定义的支持。

21. 如权利要求 1 所述的平台，其中，所述平台通过用于多个提供者的托管模型来部署。

22. 如权利要求 21 所述的平台，其中，所述多个提供者可以是从服务、发行和获取提供者的组中选择的。

23. 如权利要求 1 所述的平台，其中，所述平台通过实现将管控由生态系统参与者对服务的使用的规则来递送。

24. 如权利要求 23 所述的平台，其中，所述规则管控由参与者对服务的访问。

25. 如权利要求 23 所述的平台，其中，所述规则管控由参与者对服务的消费。

26. 如权利要求 23 所述的平台，其中，所述规则管控如何可以从生态系统中移除参与者。

27. 如权利要求 23 所述的平台，其中，所述规则管控参与者如何可以在生态系统内彼此协作。

28. 如权利要求 23 所述的平台，其中，所述规则管控参与者如何可以与生态系统外部的实体协作。

29. 如权利要求 23 所述的平台，其中，所述规则管控参与者如何获取客户。

30. 如权利要求 23 所述的平台，其中，所述规则管控参与者将如何支持客户。

31. 如权利要求 23 所述的平台，其中，所述规则管控参与者如何捕获与用户和交易相关的数据。

32. 如权利要求 31 所述的平台，其中，所述用户和交易可以是静态的。

33. 如权利要求 31 所述的平台，其中，所述用户和交易可以是动态的。

34. 如权利要求 31 所述的平台，其中，所述规则管控参与者将如何在生态系统内使用所捕获的数据。

35. 如权利要求 31 所述的平台，其中，所述规则管控参与者将如何在生态系统外部使用所捕获的数据。

36. 如权利要求 31 所述的平台，其中，所述规则管控参与者将如何在生态系统内共享所捕获的数据。

37. 如权利要求 31 所述的平台，其中，所述规则管控参与者将如何在生态系统外部共享所捕获的数据。

38. 如权利要求 31 所述的平台，其中，所述规则管控参与者将如何在生态系统内货币化所捕获的数据。

39. 如权利要求 31 所述的平台，其中，所述规则管控参与者将如何在生态系统外部货币化所捕获的数据。

40. 如权利要求 1 所述的平台，其中，所述平台是通过利用专家引擎向参与者提供客户支持来递送的。

41. 如权利要求 40 所述的平台，其中，参与者是从消费者和提供者的组中选择的。

42. 如权利要求 1 所述的平台，其中，所述平台是通过利用专家引擎向参与者提供简化

的用户体验来递送的。

43. 如权利要求 42 所述的平台，其中，所述参与者是从消费者和提供者的组中选择的。

44. 如权利要求 1 所述的平台，其中，所述平台是通过使用多个层来架构的。

45. 如权利要求 44 所述的平台，其中，所述多个层是从使能层、服务层和个性化层的组中选择的。

46. 如权利要求 44 所述的平台，其中，每个层被划分成附加的层。

47. 如权利要求 46 所述的平台，其中，所述附加的层是从逻辑层和物理层的组中选择的。

48. 一种方法，包括 ：

分析包括以下各项的集合中的一个或多个 ：生态系统参与者、服务要求、客户要求、调节要求、交易安全性要求、信任模型以及电子交易通信信道 ；

基于所述分析确定要配置在平台中的一个或多个生态系统特征，所述平台包括服务层、使能层以及个性化层中的至少一个 ；以及

根据所述确定来配置所述一个或多个生态系统特征。

49. 如权利要求 48 所述的方法，其中，所述一个或多个生态系统特征被配置为确保在服务提供者商业模型的上下文内满足每个所分析的要求。

50. 如权利要求 48 所述的方法，其中，所述平台附加地包括专家引擎层和体验框架层。

51. 一种方法，包括 ：

在包括服务层、使能层以及个性化层中至少一个的平台中，应用专家引擎来分析一个或多个移动设备用户和交易环境要求 ；以及

根据所述分析来配置一个或多个平台特征。

52. 如权利要求 51 所述的方法，其中，配置一个或多个生态系统特征包括基于一个或多个交易要求来配置用户接口。

53. 如权利要求 51 所述的方法，其中，所述专家引擎包括专家引擎层。

54. 如权利要求 53 所述的方法，其中，所述专家引擎层与体验框架相交互以适应一个或多个用户要求。

55. 如权利要求 54 所述的方法，其中，应用专家引擎包括应用专家引擎来分析用户的习惯。

56. 如权利要求 51 所述的方法，其中，配置一个或多个生态系统特征包括配置移动设备的一个或多个显示特征。

57. 如权利要求 56 所述的方法，其中，所述一个或多个显示特征是至少部分地基于与移动设备的位置相关联的地区偏好来配置的。

58. 如权利要求 56 所述的方法，其中，所述一个或多个显示特征是至少部分地基于所标识的客户端基础来配置的。

59. 一种用于使能生态系统中的多种类型的电子钱包的平台，包括 ：

软件开发套件，用于开发提供者特定的钱包，其中，所述软件开发套件至少部分地通过联网的服务器是可执行的 ；

多个参考钱包，其是通过所述软件开发套件可访问的，其中，所述多个参考钱包包括用于多个提供者中每一个的参考钱包，所述多个提供者包括器件发行者、服务提供者以及器

件获取者；以及

运行时容器，其在客户端设备上可操作，用于接收不同类型的提供者特定的钱包，其中，所述容器促进每种类型的提供者特定的钱包的不同操作以用于在提供者特定的钱包与开发所述提供者特定的钱包的提供者之间执行安全交易。

60. 一种包含程序指令的计算机可读介质，其中，通过计算机系统的一个或多个处理器执行所述程序指令使得所述一个或多个处理器实行以下步骤：

在移动设备上的钱包容器中配置一个或多个微件模块；以及

经由所述一个或多个微件模块来访问所述钱包容器中的一个或多个钱包模块。

61. 如权利要求 60 所述的计算机可读介质，此外包括：在访问所述一个或多个钱包模块之前认证所述一个或多个微件模块中的每一个。

62. 如权利要求 60 所述的计算机可读介质，其中，访问一个或多个钱包模块包括接收从服务提供者模块到钱包容器中第一微件模块的通信，并使能由第一微件模块对钱包模块的经认证访问。

63. 如权利要求 60 所述的计算机可读介质，其中，所述一个或多个钱包模块中的每一个通过网络由服务提供者配置和安装在钱包容器中。

64. 如权利要求 60 所述的计算机可读介质，其中，所述一个或多个钱包模块中的每一个通过网络由用户配置和安装在钱包容器中。

65. 一种方法，包括：

分析至少一个要求；以及

至少部分地基于所述分析来配置服务层、使能层和个性化层。

66. 如权利要求 65 所述的方法，其中，所述至少一个要求是从包括以下各项的集合中选择的：商业策略、进入市场策略、退出策略、竞争策略、产品路线图、硬件发布安排和特征列表，和软件发布安排、特征列表以及应用编程接口。

67. 如权利要求 65 所述的方法，其中，所述至少一个要求是在服务提供者商业模型的上下文内被分析的。

68. 如权利要求 65 所述的方法，其中，所述使能层被配置为包括通过使用与移动设备相关联的特定开发环境所构建的基础设施运行时容器软件。

69. 如权利要求 65 所述的方法，其中，所述服务层被配置为包括用于实现多个商业用例的公共平台服务的库。

70. 一种包含程序指令的计算机可读介质，其中，通过计算机系统的一个或多个处理器执行所述程序指令使得所述一个或多个处理器实行以下步骤：

经由移动设备上的钱包容器来配置一个或多个微件模块以供执行，其中，所述一个或多个微件模块包括服务提供者的服务群组。

71. 如权利要求 70 所述的计算机可读介质，其中，所述钱包容器在移动设备上提供运行时环境，其中，微件模块被解译以用于递送所述服务群组的一个或多个服务。

72. 如权利要求 70 所述的计算机可读介质，其中，所述钱包容器与用于提供安全个性化交易的多层平台的使能层协作地操作，以向一个或多个微件模块提供对服务提供者的工作流的排他访问。

73. 如权利要求 72 所述的计算机可读介质，其中，所述钱包容器在操作于移动设备上

的运行时环境中解译微件模块以提供排他访问。

74. 如权利要求 70 所述的计算机可读介质，其中，所述服务提供者是微件发行者。

75. 如权利要求 74 所述的计算机可读介质，其中，向所述微件发行者所发行的一个或多个微件模块提供对所述微件发行者的工作流的排他访问。

76. 如权利要求 75 所述的计算机可读介质，其中，所述钱包容器与用于提供安全个性化交易的多层平台的使能层协作地操作，以提供所述排他访问。

77. 如权利要求 74 所述的计算机可读介质，其中，向所述微件发行者所发行的一个或多个微件模块提供对所述微件发行者的用户接口的排他访问。

78. 如权利要求 77 所述的计算机可读介质，其中，所述钱包容器与用于提供安全个性化交易的多层平台的使能层协作地操作，以提供所述排他访问。

79. 如权利要求 74 所述的计算机可读介质，其中，防止所述微件发行者所发行的一个或多个微件模块访问另一微件发行者的工作流。

80. 如权利要求 79 所述的计算机可读介质，其中，所述钱包容器与用于提供安全个性化交易的多层平台的使能层协作地操作，以防止访问另一微件发行者的工作流。

81. 如权利要求 74 所述的计算机可读介质，其中，防止所述微件发行者所发行的一个或多个微件模块访问另一微件发行者的用户接口。

82. 如权利要求 81 所述的计算机可读介质，其中，所述钱包容器与用于提供安全个性化交易的多层平台的使能层协作地操作，以防止访问另一微件发行者的用户接口。

83. 一种包含程序指令的计算机可读介质，其中，通过计算机系统的一个或多个处理器执行所述程序指令使得所述一个或多个处理器实行以下步骤：

在移动设备上的钱包容器中配置一个或多个微件模块，所述微件容器包括一个或多个钱包模块；

限制由所述一个或多个微件模块对包括以下各项的元件集合的访问：所述一个或多个钱包模块、所述移动设备、一个或多个客户端应用；以及

提供对发行者特定的安全性域和微件特定的发行者安全性域中的至少一个的一个或多个微件模块的访问。

84. 一种管理微件模块访问的方法，包括：

在移动设备上提供容器运行时环境；

访问移动设备驻留的微件模块以确定与微件相关联的访问约束，其中，所述访问约束关于：电子钱包访问、设备资源访问、设备驻留的生态系统应用、由移动设备可访问的网络资源、发行者特定的安全性域以及微件特定的安全性域；

经由所述容器来操作微件模块中至少一个以施加与微件模块相关联的访问约束；以及

促进由微件模块对与微件模块相关联的发行者特定的安全性域和微件特定的安全性域中至少一个的访问。

85. 如权利要求 84 所述的方法，其中，施加微件模块访问约束包括防止对移动设备上的电子钱包资源的访问。

86. 如权利要求 84 所述的方法，其中，施加微件模块访问约束包括防止由第一微件对与第二微件相关联的钱包的访问。

87. 如权利要求 84 所述的方法，其中，施加微件模块访问约束包括基于钱包的状态而

防止由微件对移动设备上的钱包的访问。

88. 如权利要求 87 所述的方法，其中，所述钱包的状态包括活动状态和非活动状态，在所述活动状态期间允许所述微件进行的访问，在所述非活动状态期间防止所述微件进行的访问。

89. 如权利要求 84 所述的方法，其中，促进对安全性域的访问包括允许对移动设备上的数字钱包的访问，所述数字钱包与所述微件模块共享安全性域。

90. 如权利要求 84 所述的方法，其中，促进对安全性域的访问包括允许对发行者的网络资源的访问，所述网络资源与所述微件模块共享安全性域。

91. 一种方法，包括：

至少部分地通过联网的服务器来配置微件以提供对发行者特定的服务群组的访问；

在操作于客户端设备上的安全钱包容器中从所述联网的服务器接收部署所配置的微件的请求；

确定用于在所述客户端设备上操作微件的微件隔离要求；以及

基于所述隔离要求在移动设备上通过所述安全钱包容器来安装所配置的微件。

92. 一种包含程序指令的计算机可读介质，其中，通过计算机系统的一个或多个处理器执行所述程序指令使得所述一个或多个处理器实行以下步骤：

在客户端设备上部署第一发行者特定的微件以促进基于第一信任模型的安全个性化交易；以及

在所述客户端设备上部署第二发行者特定的微件以促进基于第二信任模型的安全个性化交易。

93. 如权利要求 92 所述的计算机可读介质，其中，所述第一发行者特定的微件和第二发行者特定的微件被部署在钱包容器中。

94. 如权利要求 92 所述的计算机可读介质，其中，第一信任模型和第二信任模型中的至少一个是委派信任模型、直接信任模型和代理信任模型之一。

95. 如权利要求 92 所述的计算机可读介质，其中，所述基于第一信任模型的安全个性化交易和所述基于第二信任模型的安全个性化交易中的至少一个是至少部分地基于触发来促进的.

95 如权利要求 95 所述的计算机可读介质，其中，所述触发对应于所述基于第一信任模型的安全个性化交易和所述基于第二信任模型的安全个性化交易中至少一个的货币值。

96. 一种方法，包括：

在移动设备处用多域安全钱包容器来接收警报，即部署在移动设备上的现有微件的更新版本是可用的；

确认所述更新是部署在移动设备上的现有微件的更新；

配置所述钱包容器以接收更新的微件；

用所述钱包容器来接收更新的微件；

处置所述现有微件以防止对该微件的进一步使用；以及

将更新的微件存储在移动设备上。

97. 一种方法，包括：

用操作于移动设备上的多域钱包容器向服务器查询更新的微件的存在；

从所述服务器接收指示经更新的微件的可用性的信息；

请求所述更新的微件；

在移动设备处接收所述更新的微件；

处置所述更新的微件所取代的现有微件以防止对所述现有微件的进一步使用；以及

将所述更新的微件存储在移动设备上。

98. 如权利要求 97 所述的方法，其中，查询服务器是以预定时间间隔而执行的。

99. 如权利要求 97 所述的方法，其中，查询服务器是响应于客户端设备上微件的执行而执行的。

100. 如权利要求 97 所述的方法，此外包括将存储了所述更新的微件的确认发送到所述服务器。

101. 一种方法，包括：

从移动设备接收针对存在更新的微件的查询；

将指示所述更新的微件的可用性的信息传输到移动设备；

从所述移动设备接收针对所述更新的微件的请求；以及

将所述更新的微件传输到所述移动设备。

102. 如权利要求 101 所述的方法，其中，查询是以预定时间间隔而接收的。

103. 如权利要求 101 所述的方法，其中，查询是响应于客户端设备上微件的执行而接收的。

104. 如权利要求 101 所述的方法，此外包括接收存储了所述更新的微件的确认。

105. 一种方法，包括：

在移动设备上配置至少一个移动交易平台特定的应用编程接口以用于促进由执行在移动设备上的钱包容器对移动设备安全资源的访问；

将多个不同的钱包容器布置在移动设备的存储器中，其中，每个钱包容器经由所述至少一个应用编程接口与移动设备安全资源相接口；以及

其中，所述至少一个应用编程接口经由三层次 移动交易平台的安全性准则来施加钱包容器隔离。

106. 如权利要求 105 所述的方法，其中，所述三层次包括个性化层次、服务层次和使能层次。

107. 一种在客户端设备上可操作的钱包容器，包括：

运行时组件，用于促进管理微件、钱包、安全性以及对安全元件的访问；

基于 API 的接口，用于访问由所述微件提供的商业特定的服务以及用于访问客户端设备特定的资源；以及

其中，所述钱包容器促进多个不同钱包和不同微件的操作以用于访问所述多个不同钱包中的至少一个。

108. 一种包含程序指令的计算机可读介质，其中，通过计算机系统的一个或多个处理器执行所述程序指令使得所述一个或多个处理器实行以下步骤：

在移动设备上的钱包容器中配置一个或多个微件模块，所述钱包容器包括一个或多个钱包模块；以及

促进由微件中至少一个的发行者对钱包模块中至少一个的访问。

109. 一种管理多个交易协议的方法，包括：

将钱包模块布置在客户端设备的存储器中；

将钱包伴随小程序布置在所述存储器中；

执行与通过其中部署了所述客户端设备的生态系统进行的安全个性化交易相关联的钱包伴随小程序，以管理对与所述客户端设备相关联的安全元件的所选内容的访问；以及

其中，所述钱包伴随小程序使用所访问的安全元件内容来促进认证对所述钱包模块的访问以执行交易。

110. 一种包含程序指令的计算机可读介质，其中，通过计算机系统的一个或多个处理器执行所述程序指令使得所述一个或多个处理器实行以下步骤：

在移动设备上的钱包容器中配置一个或多个微件模块，所述钱包容器包括一个或多个钱包模块；以及

配置至少一个小程序以与至少一个钱包模块结合执行来促进对多个交易协议的管理。

111. 一种方法，包括：

将钱包模块布置在客户端设备的存储器中；

将钱包伴随小程序布置在由客户端设备可访问的安全元件中，其中，所述钱包伴随小程序促进由至少一个其它小程序和至少一个钱包模块经由移动钱包功能可共享接口而对所述安全元件中存储的内容的访问。

112. 如权利要求 111 所述的方法，其中，所述至少一个其它小程序执行从包括优惠券发放、安全性和支付的小程序功能列表中选择的功能。

113. 一种方法，包括：

将用于提供至少一个微件和至少一个钱包的运行时环境的容器布置在客户端设备的存储器中，其中，所述容器通过可共享接口来提供对所述至少一个微件和所述至少一个钱包的访问；

将容器伴随小程序布置在客户端设备的安全元件中，当在所述运行时环境中执行时，所述容器伴随小程序促进由至少一个其它小程序对与所述至少一个钱包相关联的安全钱包资源的访问。

114. 如权利要求 113 所述的方法，其中，所述可共享接口促进对经由所述容器可访问的移动钱包功能和移动微件功能的访问。

115. 一种在客户端设备上可操作的平台，包括：

运行时组件，用于配置至少一个标记化的微件，其中，所述至少一个标记化的微件促进用于安全个性化交易的多域生态系统中的至少一个交易。

116. 一种在客户端设备上可操作的平台，包括：

运行时组件，用于配置用于安全个性化交易的多域生态系统中的至少一个服务提供者微件。

117. 如权利要求 116 所述的平台，其中，所述服务提供者微件促进对与所述服务提供者相关联的电子钱包的访问。

118. 一种在客户端设备上可操作的平台，包括：

移动交易平台移动设备运行时组件；

通过所述运行时组件可操作的至少一个移动交易平台应用编程接口，包括用于在移动

设备上配置至少一个客户端容器的程序功能；

其中，所述移动设备是用于安全个性化交易的多域生态系统中的参与者。

119. 一种布置在移动设备上的移动交易平台，用于保护布置在所述移动设备上的电子钱包中的多个账户的账户信息，包括：

钱包容器，其在所述移动设备上操作以通过以下方式来提供以安全方式对所述多个账户的访问：在运行时环境中解译微件小程序并限制由所述微件小程序对所述电子钱包中账户的访问，所述微件小程序被授权访问所述电子钱包。

120. 一种在客户端设备上可操作的平台，包括：

运行时组件，用于保护钱包中多个账户的账户信息。

121. 一种在客户端设备上可操作的钱包容器，其管理由服务提供者特定的微件访问移动钱包的请求，以基于所述服务提供者的钱包认证状态来对所述移动钱包施加访问限制。

122. 一种在客户端设备上可操作的平台，包括：

运行时组件，用于提供通过多个服务提供者对移动钱包的安全访问。

123. 一种用于将供应与在外部服务提供和用于安全个性化交易的多域生态系统中的移动钱包之间的交易集成在一起的移动钱包，包括：

在移动设备上可操作的多域数字钱包，其中，所述数字钱包包括来自至少两个不同服务提供者的账户；

运行时容器，用于执行尝试访问所述数字钱包的微件；

多个移动供应管理微件，其在所述运行时容器中可操作，所述运行时容器通过与外部服务提供者通信来与不同移动供应域相接口；以及

移动交易能力，其通过所述多个移动供应管理微件的至少一部分、所述多域数字钱包以及所述运行时容器的协作操作来提供。

124. 一种经由多域生态系统来抽象安全个性化交易中的协议的方法，包括：

在所述多域生态系统中的联网设备处接收安全个性化交易中的电子交易消息，其包括多个信息字段；

用所述设备来确定所述多个信息字段中每一个的类型和内容；

将所述多个信息字段的至少一部分映射到公共格式的消息字段；

基于所述映射来配置公共格式的交易消息；以及

将所述公共格式的交易消息从所述联网设备传输到所述多域生态系统中的第二联网设备。

125. 一种经由多域生态系统来抽象安全个性化交易中的协议的方法，包括：

在所述多域生态系统中的联网设备处接收安全个性化交易协议中的电子交易消息，其包括多个信息字段；

用所述设备来确定所述多个信息字段中每一个的类型和内容；

用元模型将所述多个信息字段的至少一部分映射到公共格式的消息字段，所述元模型提供安全个性化交易协议的一致映射；

基于所述映射来配置公共格式的交易消息；以及

将所述公共格式的交易消息从所述联网设备传输到所述多域生态系统中的第二联网设备。

126. 一种经由多域生态系统来抽象安全个性化交易中的协议的方法，包括：

在所述多域生态系统中的联网设备处接收安全个性化交易中的电子交易消息，其包括多个信息字段；

用所述设备来确定所述多个信息字段中每一个的类型和内容；

将所述多个信息字段的至少一部分映射到公共格式的消息字段；

基于所述映射来配置公共格式的交易消息；

将所述公共格式的交易封装在协议抽象内容封装器中；以及

将经封装的公共格式交易消息从所述联网设备传输到所述多域生态系统中的第二联网设备。

127. 一种经由多域生态系统对安全个性化交易进行动态标记化的方法，包括：

在所述多域生态系统中的联网设备处接收安全个性化交易中的电子交易消息，其包括多个非机密信息字段；

在所述联网设备处配置包括所述非机密信息和交易标识信息的标记；

将所述标记电子地传输到所述电子交易消息的源以及传输到布置在所述多域生态系统中的服务提供者服务器；以及

将所述标记的认证连同标记标识信息一起从所述电子交易消息的源传输到所述服务提供者服务器以用于认证由所述标记标识的交易。

128. 如权利要求 127 所述的方法，其中，所述服务提供者是器件发行者。

129. 如权利要求 127 所述的方法，其中，所述服务提供者访问所述联网设备不可访问的机密信息以促进认证由所述标记标识的交易。

130. 如权利要求 129 所述的方法，其中，所述机密信息与布置在从其传输所述标记认证的设备上的数字钱包相关联。

131. 如权利要求 127 所述的方法，其中，通过在所述电子交易消息所源自的设备上执行的提供者特定的微件来促进传输所述标记认证。

132. 一种标记移动缓存的方法，包括：

用联网的服务器生成用于通过公共网络与客户端设备执行电子交易的标记；

通过所述公共网络将所述标记提供给所述客户端设备；

将所述标记连同唯一地标识所述标记被提供到的客户端的信息一起存储在所述服务器可访问的存储器中；以及

响应于在所述服务器处从客户端设备接收到电子消息以使用所述标记通过所述公共网络进行电子交易，将所存储的唯一地标识所述标记被提供到的客户端的信息与用所述标记接收的客户端标识信息相比较以确定进行所述电子交易是否应被认证。

133. 如权利要求 132 所述的方法，其中，所述电子消息包括从与所述标记被提供到的客户端相关联的安全元件得到的信息。

134. 一种经由多域生态系统来对安全个性化交易中的封装器进行标记化的方法，包括：

在所述多域生态系统中的联网设备处接收安全个性化交易中的电子交易消息，其包括多个信息字段；

用所述设备来确定所述多个信息字段中每一个的类型和内容；

将所述多个信息字段的至少一部分映射到公共格式的消息字段；

基于所述映射来配置公共格式的交易消息；

将所述公共格式的交易封装在协议抽象内容封装器中；

配置包括所述协议抽象内容封装器的标记；以及

将经封装的公共格式交易消息从所述联网设备传输到所述多域生态系统中的第二联网设备。

135. 一种经由多域生态系统来对用于安全个性化交易的封装器进行标记化的方法，包括：

在所述多域生态系统中的联网设备处接收用于安全个性化交易中的电子交易消息的封装器；

在所述联网设备处配置包括所述封装器和交易标识信息的标记；

将所述标记电子地传输到用于电子交易消息的封装器的源和传输到布置在所述多域生态系统中的服务提供者服务器；以及

将所述标记的认证连同标记标识信息一起从用于电子交易消息的封装器的源传输到所述服务提供者服务器以用于认证由所述标记标识的交易。

136. 如权利要求 135 所述的方法，其中，所述服务提供者是器件发行者。

137. 如权利要求 135 所述的方法，其中，所述服务提供者访问所述联网设备不可访问的机密信息以促进认证由所述标记标识的交易。

138. 如权利要求 137 所述的方法，其中，所述机密信息与布置在从其传输所述标记认证的设备上的数字钱包相关联。

139. 如权利要求 135 所述的方法，其中，通过在用于电子交易消息的封装器所源自的设备上执行的提供者特定的微件来促进传输所述标记的认证。

140. 一种在用于安全个性化交易的多域生态系统中执行标记化的电子交易的方法，包括：

生成表示电子交易消息的标记，其标识至少一个交易和所述交易中的至少一个生态系统参与者；

封装表示电子交易消息的所述标记，从而提供包括表示电子交易消息的标记的封装器；

生成表示所述封装器的标记，从而提供用于标记化表示电子消息的标记化的封装器；以及

通过在所述生态系统中的两个参与者之间传输所述标记化的封装器来执行所述两个参与者之间的电子交易，其中，所述生态系统中的两个参与者之一是至少一个生态系统参与者。

141. 一种移动交易处理系统，包括：

部署在网络中的多个服务器和多个客户端设备，每个服务器被配置为促进所述网络中的所述多个客户端设备和多个服务提供者服务器间的安全个性化交易；

所述多个客户端设备中的至少一个配置有在基于容器的执行环境中可操作的可容纳多租户的钱包，其中，所述钱包是通过布置在所述多个客户端设备的所述至少一个上的微件可访问的；

所述多个服务器此外被配置为促进在经由所述微件的所述多个客户端设备中至少一个与经由多个信任模型中任一个的所述多个服务提供者服务器中至少一个之间的交易，所述信任模型包括单个信任域、单个信任集群、多个信任集群以及直接信任关系。

142. 如权利要求 141 所述的移动交易处理系统，其中，所述多个不同信任模型包括以下各项中的至少一个：单个信任域、单个信任集群、多个信任集群以及直接信任关系。

143. 如权利要求 142 所述的移动交易处理系统，其中，所述单个信任域包括中央钱包服务中心服务器，其与用户设备上的电子钱包相接口，并且促进建立与可信服务提供者的安全交易信道。

144. 如权利要求 142 所述的移动交易处理系统，其中，所述单个信任集群包括可信第三方服务器和钱包服务中心服务器，所述可信第三方服务器用于与个人电子钱包安全地交换机密钱包信息，所述钱包服务中心服务器用于促进由多个服务提供者中的每一个经由所述第三方服务器对用户的直接认证，其中，所述钱包服务中心服务器将用户标识和安全性密钥递送到所述多个服务提供者中的至少一个。

145. 如权利要求 142 所述的移动交易处理系统，其中，多个信任集群包括多个不同的可信第三方，其各自至少配置有用于以下各项的服务器：与个人钱包交换机密钱包信息，并且此外与多个集群特定的服务提供者交换由钱包服务中心提供的用户和 / 或设备标识。

146. 如权利要求 142 所述的移动交易处理系统，其中，直接信任关系包括归属钱包服务中心服务器，用于建立与电子钱包的交易上下文，其促进直接在电子钱包与服务提供者之间建立认证的交易信道。

147. 如权利要求 141 所述的移动交易处理系统，其中，所述微件被配置用于与特定提供者的安全交互。

148. 如权利要求 141 所述的移动交易处理系统，其中，所述微件被配置用于通过中介与特定提供者的安全交互。

149. 如权利要求 147 所述的移动交易处理系统，其中，通过在所述移动设备和与特定提供者相关联的联网的服务器之间的电子交互而在空中将所述微件配置在所述移动设备上。

150. 一种在移动设备上操作的安全移动钱包，包括：

多个不同的用户账户，其中，对不同用户账户的访问是经由布置在所述移动设备上的微件由提供者特定的信任模型来促进的。

151. 如权利要求 150 所述的安全移动钱包，其中，所述微件被配置用于与特定提供者的安全交互。

152. 如权利要求 150 所述的安全移动钱包，其中，所述微件被配置用于通过中介与特定提供者的安全交互。

153. 如权利要求 151 所述的安全移动钱包，其中，所述微件在空中通过在所述移动设备和与特定提供者相关联的联网的服务器之间的电子交互而配置在所述移动设备上。

154. 一种在用于安全个性化交易的生态系统中用于移动交易的信任模型确定的方法，包括：

配置部署在网络中的多个服务器以经由多个信任模型中的任一个来促进客户端设备与提供者特定的服务器之间的交易，所述信任模型包括单个信任域、单个信任集群、多个信

任集群以及直接信任关系；

分析用于所述移动设备与所述提供者特定的服务器之间的交易的默认信任模型，以确定与交易相关联的风险度量；以及

基于产生的风险度量，经由所述默认信任模型和来自所述多个信任模型的另一信任模型之一来进行交易。

155. 如权利要求 154 所述的方法，其中，进行交易包括经由在所述移动设备上执行的提供者特定的微件来访问所述移动设备上的钱包功能。

156. 如权利要求 155 所述的方法，其中，基于产生的风险度量来选择所述提供者特定的微件。

157. 如权利要求 156 所述的方法，其中，所述风险度量基于所述移动设备的位置。

158. 如权利要求 156 所述的方法，其中，所述风险度量基于所述提供者特定的服务器相对于所述移动设备的位置的位置。

159. 如权利要求 156 所述的方法，其中，所述风险度量基于所述移动设备正通过其进行交易的网络的各方面。

160. 一种用于使用在生态系统中的域特定的程序语言，所述生态系统牵涉器件发行者、服务提供者、获取者以及客户端，所述编程语言包括：

顺从 XML、CSS 和 Javascript 的语法；以及

在客户端设备上使能钱包相关功能的对象。

161. 如权利要求 160 所述的域特定的程序语言，其中，所使能的钱包相关功能是钱包容器、钱包、微件和钱包伴随小程序中的至少一个。

162. 一种用于服务多域生态系统中的安全个性化交易的平台，包括：

解译器，用于解译用于使能钱包相关功能的域特定的程序语言对象，所述钱包相关功能与多个域中的至少一个相关。

163. 如权利要求 162 所述的平台，其中，所述解译器是运行时解译器。

164. 如权利要求 162 所述的平台，其中，所述解译器是在客户端设备上执行的运行时容器。

165. 如权利要求 162 所述的平台，其中，所使能的钱包相关功能是钱包容器、钱包、微件和钱包伴随小程序中的至少一个。

166. 一种用于保护多域生态系统中的内容的一组工具，包括：

软件开发套件，用于构造移动交易生态系统；

一组域特定的功能，用于使能微件、钱包、容器和伴随小程序中的至少一个的客户端侧功能性；

设备不可知的应用编程接口；

设备特定的应用编程接口；

使能层功能，用于促进与客户端设备软件和硬件的互操作；以及

微件公布模块，用以促进提供者特定的微件用于递送和下载到客户端设备的可用性。

167. 一种网络使能的服务器，用于配置用于安全个性化交易的多域生态系统的一组服务提供者工具，包括：

移动交易平台服务器；

在所述平台服务器上可操作的钱包服务中心开发环境；

在所述平台服务器上可操作的钱包服务中心运行时模块；以及

软件开发套件，用于构造和维护生态系统的基于服务器和基于客户端的元件。

168. 如权利要求 167 所述的网络使能的服务器，其中，生态系统的基于客户端的元件包括钱包、微件、伴随小程序以及容器。

169. 一种用于现场定制微件的一部分的移动应用，包括：

用户接口模板，用于从在联网的服务器上执行的微件开发过程接收微件功能代码和描述符；

用户接口，用于促进经由微件描述符所定义的图标对微件特征和功能进行访问；以及

用于配置下载到设备的微件的算法，所述用户接口基于所述用户接口中检测到的用户动作而显示在所述设备上。

170. 如权利要求 169 所述的移动应用，其中，所述用户接口模板是基于微件描述符而自动可配置的。

171. 如权利要求 169 所述的移动应用，其中，用于现场定制微件的一部分的应用在移动设备上操作。

172. 如权利要求 169 所述的移动应用，其中，所述应用在移动客户端容器运行时环境内执行。

173. 一种基于服务器的应用，包括：

一组参考移动交易平台服务；

自动化的向导应用，用于访问该组参考服务并基于用户输入来配置来自该组参考服务的至少一个服务；以及

其中，所述自动化的向导应用促进基于用户输入来配置以下各项中的至少一个：客户端元件分布、发行者环境、结算环境、聚集服务、交易介质支持、对手持机类型的支持、用户体验、商业模型以及调节考虑。

174. 一种生态系统配置引导系统，包括：

一组参考移动交易平台服务；以及

自动化的向导应用，用于接收与期望的生态系统配置相关联的用户输入，并基于用户输入来推荐来自该组参考服务的至少一个服务的配置。

175. 如权利要求 174 所述的生态系统配置引导系统，其中，该组参考服务包括分布服务、连续体使用服务、发行者服务、结算服务、聚集服务、交易介质要求、手持机和操作系统考虑、用户体验因素、商业建模以及调节考虑。

176. 如权利要求 174 所述的生态系统配置引导系统，其中，与期望的生态系统配置相关联的用户输入是超链接。

177. 如权利要求 174 所述的生态系统配置引导系统，其中，所述向导基于超链接来搜索信息，并且填充至少一个参考服务模板。

178. 如权利要求 176 所述的生态系统配置引导系统，其中，所述超链接是去往操作所述向导的服务提供者的网站的链接。

179. 一种在用于安全个性化交易的多域生态系统中的参与者间的类型 - 长度可变数据通信的方法，包括：

在作为多域生态系统中的参与者的生产者服务器处接收纯旧式 java 对象；

将接收到的纯旧式 java 对象转换成顺从类型长度可变协议的类型长度可变流；以及

经由单向消息总线将经转换的对象从生产者传输到多域生态系统中的移动设备参与者。

180. 如权利要求 179 所述的方法，其中，所述纯旧式 java 对象遵照类型长度可变协议指南。

181. 如权利要求 180 所述的方法，其中，所述类型长度可变协议支持流式传输、浮点支持，以及预先考虑的运行时类类型信息。

多层安全移动交易使能平台

[0001] 相关申请的交叉引用

本申请是目前未决的2006年10月5日提交的、标题为“Transactional Services”的美国专利申请11539024的部分继续。该申请要求2005年10月5日提交的、标题为“Methods and Systems for Transactional Services”的美国临时专利申请号60/724,066的权益。

[0002] 本申请要求2011年10月12日提交的美国临时专利申请61546084和2012年4月3日提交的美国临时专利申请61619751的权益。

[0003] 本申请与以下美国专利申请相关:2007年10月31日提交的、标题为“System and methods for servicing electronic transactions”的美国专利申请号11/931,872;和2002年10月31日提交的、标题为“System and methods for servicing electronic transactions”的美国专利申请号10/284,676。

[0004] 上述申请中的每一个通过引用以其整体被并入本文。

[0005] 本申请与以下美国专利相关:2000年6月6日提交的、标题为“Point of sale and display adapter for electronic transaction device”的美国专利号6,769,607;1999年11月15日提交的、标题为“Point of sale adapter for electronic transaction device”的美国专利6,705,520;1994年6月20日提交的、标题为“Universal electronic transaction card including receipt storage and system and methods of conducting electronic transactions”的美国专利号 5,590,038;1996年9月6日提交的、标题为“Device, system and methods of conducting paperless transactions device”的美国专利号5,884,271,1999年3月10日提交的、标题为“Device, system and methods of conducting paperless transactions”的美国专利号 6,925,439;1999年8月11日提交的、标题为“System and methods for servicing electronic transactions”的美国专利号 7,308,426;以及2001年1月19日提交的、标题为“Method and system for managing user activities and information using a customized computer interface”的美国专利号 7,366,990,这些美国专利中的每一个通过引用以其整体被并入本文。

背景技术

[0006] 领域:

用于多域生态系统(ecosystem)中的安全个性化交易的多层移动交易平台的方法和系统一般地涉及移动交易处理安全性。

[0007] 相关领域的描述:

移动交易正变成销售点支付的最普遍的形式。用于移动交易处理的现有系统不具有解决日益增长的对安全个性化交易服务和客户联系的需求以满足将来业务需求所需的可配置性的可伸缩性和灵活性。

发明内容

[0008] 移动交易平台(MTP)可以包括运营商等级的基础设施平台,其可以促进安全交

易服务的开发，以及将安全交易服务(移动商务(例如在空中)和移动支付(例如，近场通信(NFC)邻近性)二者)递送到广泛的移动电话环境。MTP 可以包括增强的软件开发方法，用以通过提供一致、公共开发环境来进一步简化开发并从开发者隐藏电话环境特定的复杂性。还可以在开发期间展露原生电话特征以允许开发者定义原生电话体验等，同时仍然受益于公共且鲁棒的开发方法和环境。使用 MTP 的一个益处是用于广泛的移动手持机的快速移动服务开发，用以快速构建有利的移动交易生态系统。

[0009] 另外，本文公开的是端对端安全交易环境，其中，应用信任模型，其不仅确保安全电子交易，而且通过使用对每个用户、用户操作的每个设备以及每个交易和 / 或服务域的动态认证而在用户、支付设施以及商家间提供安全且安稳的电子商业操作。所公开的信任模型例如确保用户和商家可以在不使用“中间人”的情况下进行交易来确保安全且可靠的商务。通过经由一个或多个主服务设施认证用户并且通过经由此类主服务设施认证商家和其它提供者，对于每个商家而言，有可能通过用户已经在其上向主服务设施经认证的任何设备而具有与每个用户的唯一而安全的关系。然而，通过使用用于认证的主服务设施和使能鲁棒的信任模型，多个商家(例如，产品、金融服务以及本文描述的许多其它服务的提供者)可以安全并直接地与单个用户交易。以这种方式，多个经认证的商家可以在聚集的选集(portfolio) 中对用户是可用的，这可以大大促进改进的安全电子商务。

[0010] 电子商业环境的各方面可以包括提供者域(例如，银行、商家等)，其可以通过钱包平台来访问并且提供增建钱包和微件能力的能力以使能从风险管理视角查看电子交易环境。在示例中，跨生态系统的交叉销售可以利用本文描述的方法和系统，从而允许将风险管理映射到电子商务过程中。

[0011] 安全的多域生态系统环境的各方面可以将环境延伸至客户。一旦客户被认证到此类环境中，其就可以具有建立信任模型的能力，其可以使能通过单个信任模型来递送服务的特定实例。可以通过使用其它信任模型来递送其它产品 / 服务。

[0012] 而且，本文的方法和系统可以促进将安全性隧道延伸到商家，其可以具体化为隧道内的隧道，这可以使能核心安全性架构。以这种方式，商业可以避免随着安全性而陷入困境，而非集中于将提供可持续商业模型的应用。在示例中，本文的方法和系统可以使商业人员能够查看应用的使用频率(例如，3x/ 天与 3x/ 月)，以帮助针对不同用户交互模型的范围中的每个预想显著不同的信任模型。

[0013] 根据对优选实施例和附图的以下详细描述，本发明的这些及其它系统、方法、目的、特征和优点对于本领域技术人员将是显而易见的。本文提及的所有文档通过引用以其全部被并入于此。

附图说明

[0014] 可以通过参照以下附图来理解本发明及对其某些实施例的以下详细描述：

图 1 图示了用于安全移动商务(mCommerce)和移动支付(mPayment)交易的动态生态系统 100。

[0015] 图 2 图示了移动交易平台(MTP) 高层级结构。

[0016] 图 3 图示了移动交易平台(MTP) 架构。

[0017] 图 4 图示了具有 MTP 移动客户端(mClient)运行时的钱包应用的高层级架构视图。

[0018] 图 5 参照开发周期图示了微件(widget)开发过程的概观。

[0019] 图 6 图示了用于安全个性化交易的生态系统的概观。

[0020] 图 7 图示了用于使用在用于安全个性化交易的生态系统中的示例性客户端架构。

[0021] 图 8 图示了用于安全个性化交易的生态系统,其包括具有协议抽象支持能力的 MTP 结构配置。

[0022] 图 9 图示了如图 8 中所图示的用于安全个性化交易的生态系统的可替换配置。

[0023] 图 10 图示了可以跨越一系列生态系统元件的分层平台的概念。

[0024] 图 11 图示了选择生态系统特征以用于包括在用于安全个性化交易的多域生态系统平台的使能 (enabling) 层、服务层和个性化层的每一个中的实施例。

[0025] 图 12 图示了用于在用于安全个性化交易的多域生态系统中促进针对多个服务提供者的每一个修改个性化和服务层中的每一个的专家引擎。

[0026] 图 13 图示了选择业务需求和设备功能性以用于包括在用于安全个性化交易的多域生态系统平台的使能层、服务层和个性化层的每一个中的实施例。

[0027] 图 14 图示了开发微件的方法。

[0028] 图 15 图示了微件使用的方法。

[0029] 图 16 图示了用于安装微件的交易流程图。

[0030] 图 17 图示了描绘 MTP 系统中微件的生命周期的状态图。

[0031] 图 18 图示了从开发到执行的微件生命周期。

[0032] 图 19 图示了访问控制机制的高层级视图。

[0033] 图 20 图示了用于相对于访问控制器而加载和执行微件的方法。

[0034] 图 21 图示了基于微件访问控制策略的微件运行时覆盖区(footprint)。

[0035] 图 22 图示了微件批准过程。

[0036] 图 23 图示了微件架构的示例性且非限制性实施例。

[0037] 图 24 图示了客户端设备容器的高层级架构视图。

[0038] 图 24A 图示了移动设备的移动交易平台相关元件的详细视图。

[0039] 图 24B 图示了钱包伴随域特征,以及控制和数据流。

[0040] 图 25 图示了供应或优惠券相关的交易流程图。

[0041] 图 26 图示了根据示例的在交易流期间优惠券小程序的状态机图。

[0042] 图 27 图示了在小程序与销售点终端之间的交互。

[0043] 图 28 图示了根据示例的在交易流期间优惠券小程序(applet)的状态机图。

[0044] 图 29 图示了示例性实施例中用于部署本发明的系统的高层级架构。

[0045] 图 30 图示了描述通过使用具有下载 URL 的 SMS 而在用户的移动电话上安装移动钱包的方法的流程图。

[0046] 图 31A 和 31B 图示了由用户进行的用于钱包激活的方法。

[0047] 图 32 图示了作为搜索和下载用例的示例的方法。

[0048] 图 33 图示了用于 OTA 交易的示例性方法。

[0049] 图 34 图示了作为用于发卡用例的示例的方法。

[0050] 图 35 图示了作为终止钱包的示例的方法。

[0051] 图 36 图示了作为用于发布微件的示例的方法。

[0052] 图 37 是用于第一次添加微件连同 TSM 交互的方法。

[0053] 图 38 是参照微件加载时间签名、核实和执行时间访问核实的用于微件访问核实的系统的行为模型。

[0054] 图 39 图示了用于安全商务和支付交易的动态生态系统。

[0055] 图 40 图示了用于安全商务和支付交易的动态生态系统中的协议抽象的方法。

[0056] 图 41 图示了在用于安全商务和支付交易的动态生态系统中对交易内容标记化(tokenization)的方法。

[0057] 图 42 图示了用以从在移动设备上运行的移动钱包接收动态标记(token)的销售点设备。

[0058] 图 43 图示了在用于安全商务和支付交易的动态生态系统中生成用于穿过网络的安全内容的电子交易封装器的标记的方法。

[0059] 图 44 图示了在用于安全商务和支付交易的动态生态系统中对交易封装器标记化的方法。

[0060] 图 45 图示了通过传输动态生成的标记化的封装器来执行跨开放网络的电子交易的方法,所述封装器包括表示在用于安全商务和支付交易的动态生态系统中的交易内容的标记。

[0061] 图 46 图示了在交易流中的协议抽象的方法,其包括具有单个开证银行的多个采集方法。

[0062] 图 47 图示了在空中交易流中的协议抽象的方法,其包括单个采集方法和多个开证银行。

[0063] 图 14 图示了标记化的交易、标记化的协议抽象以及标记化的封装器的方法。

[0064] 图 49 图示了单个信任域模型。

[0065] 图 50 图示了如图 50 中所图示的单个信任集群模型,其可以假设可信第三方在生态系统中是可访问的。

[0066] 图 51 图示了多个信任集群模型。

[0067] 图 52 图示了直接信任关系,其中不强迫任何两个实体信任彼此。

[0068] 图 53 图示了消息总线通信架构。

[0069] 图 54 图示了支付促进器系统的各种子系统的高层级框图。

[0070] 图 55 示出了钱包的各种组件。

[0071] 图 56 图示了可以基于支付促进器容器的部署模型 5600 的高层级框图。

具体实施方式

[0072] 尽管本文参照各种实施例进行了描述,但是理解的是,在所有情况中,除非另外指定,否则对“实施例”或“多个实施例”的引用是指一个或多个示例性并且非限制性实施例。同样,理解的是,在本文的所有描述中,除非另外指定,否则即使当不是明确引用“实施例”或“多个实施例”时也是指一个或多个示例性并且非限制性实施例。

[0073] 多域生态系统可以被配置成经由多层安全移动交易使能平台而用于安全个性化交易,所述多层安全移动交易使能平台可以包括使能层、服务层、个性化层、专家层和经验框架层。多域生态系统可以包括多个服务提供者、器件(instrument)发行者、移动交易处理

器、交易获取器、第三方认证器以及经由各种各样的电子设备访问生态系统的多个用户。如本文所描述的多层安全移动交易生态系统使能平台可以促进任何生态系统参与者配置生态系统的各方面以满足一组独特的业务模型需求。多层安全移动交易生态系统使能平台可以提供软件开发和元件配置工具，用以准备和个性化目标特定业务模型和 / 或信任模型所需要的电子钱包、活动的设备微件、后端基础设施运行时代码等。通过促进安全、可信的生态系统操作(例如，电子钱包的发行、维护和使用)，该平台可以支持经由单个移动设备与大量不同域进行安全交易。通过提供工具来个性化功能元件(例如，微件)以用于部署在移动设备中，该平台可以促进多个不同域(例如，不同服务提供者)与诸如客户之类的用户之间的安全的、定制的交易相关通信，这可以使用于与客户交互的机会增加到 10 倍或更多倍。

[0074] 可以可替换地铸造为移动交易平台或 MTP 的多层安全移动交易使能平台可以为用户递送安全交易功能性，从而使能参与者的生态系统，所述参与者除其它之外尤其包括服务提供者、器件和服务的发行者 / 获取者、相关支持和增值服务提供者以及分布网络。可以通过良好结构化的集中于递送能力的软件和 / 或硬件模块来递送多层安全移动交易使能平台，所述递送能力诸如对用户通信的直接服务提供者、消息传送和通知、安全性和认证、空中(OTA)供应、OTA 可信服务管理器(TSM)代理操作、基于客户端的安全生态系统容器操作、钱包和微件管理、交易引擎操作、数字器件管理、位置和 / 或地理围栏(geo-fencing)支持、增强现实等。

[0075] 在多域生态系统中用于安全个性化交易的多层移动交易平台的方法和系统可以促进建立和操作市场地点或交易相关证书的交换。这些证书可以被标记化或者可以不被标记化，这服从于它们各自的发行者和获取者的、与支付和非支付服务相关的等等要求。这些证书可以在空中并且在邻近环境中通过诸如销售点终端等的信道主机来进行交易。通过采用可以使能这样的交易相关证书交换的这些方法和系统，可以解决与伸缩这样的生态系统相关的关键商业问题。交易相关证书的这种交换可以完全消除对于单个服务提供者将它们自己嵌入作为这样的交易证书的有效交换的需要，并且相反地，允许现有发行者、获取者和服务提供者(和 / 或这样的证书的消费者)继续利用它们的现有系统、商业和风险管理实践。这样的交换将通过利用本文描述的移动交易平台的各方面来增值，这些方面包括各种部署模型、可配置生态系统规则、底层聚集安全性架构和可配置安全性信任模型、必要工具、实用程序、web 服务、移动设备和服务器应用(例如，钱包、容器、微件、钱包伴随域)等。结果可以是在高度可伸缩企业环境中的高质量服务递送，以用于确保不同生态系统参与者可以被整合并且可以添加新参与者，而没有繁重开销。

[0076] 图 1 图示了用于安全移动商务和移动支付交易的动态生态系统 100。本文可以描述生态系统 100。如下文所描述的，MTP 114 可以被配置为对用于安全移动商务和移动支付交易的动态生态系统供以动力。MTP 可以设置在网络提供者的域 102 中并且集成到网络提供者 104 的系统中。在示例中，网络提供者 104 的企业服务总线(ESB) 108 可以用于集成。MTP 114 可以连接到服务和安全性提供者 110 的生态系统，其可以包括一个或多个可信服务管理者(TSM)，例如 112a、112b。网络提供者 104 可以使用 MTP 114 来创建以特定商业垂直面(vertical)为目标的品牌移动应用。在示例中，移动应用可以是用于零售 / 金融域的移动钱包(mWallet)或者用于医疗保健垂直面的移动健康(mHealth)应用。在示例中，移动应用可以向用户提供特定于一域的核心服务集，其可能针对该域而已经被创建。出于本发

明的目的，移动应用可以是假设零售域应用的钱包。在示例中，钱包还可以促进应用生命周期和安全性、标准化用户体验以及微件管理职责。微件可以提供特定的商业服务并且遵照钱包指南。移动应用可以下载到用户手持机(handset)118上。服务110的生态系统可以被配置为对手持机118上进行的任何交易进行认证。在示例中，可以在商家终端120处使用用户手持机118。商家终端120可以被配置为从生态系统云110接收通信，其关于用户手持机118微件122当用在商家终端120附近以用于实现交易时的微件认证。

[0077] 在示例中，单独、独立的服务提供者可以使用MTP SDK来构建可以加载到钱包中的微件。这些微件可以提供增量式功能性，其可以增强移动应用的整体吸引力。钱包和微件可以为用户提供OTA增值服务(VAS)和邻近NFC服务。容器(或MTP移动客户端运行时)可以为钱包和微件提供安全运行时环境和服务，以与服务器(OTA服务)进行OTA通信，并且还与安全元件通信来为邻近NFC交易管理支付和非支付小程序。

[0078] 在示例中，微件可以具有在安全元件中运行的相关联的小程序。在示例中，微件可以没有在安全元件中运行的相关联的小程序。在示例中，服务提供者可以仅选择仅提供OTA服务。在示例中，MTP和TSM可以一起协作来提供整个NFC体验。在示例中，钱包和微件(包括软卡)供应、设置和管理可以由MTP操控，而小程序发行和个性化可以由TSM完成。在示例中，网络提供者可以确保可以装备商家位置处的NFC读取器以操控邻近协议，所述邻近协议可以由可能已经加载到电话的SE中的各种小程序所使用。这些读取器然后可以通过获取网络来处理交易，所述获取网络可以将交易切换到云中的合适生态系统服务提供者。钱包和/或微件可以与生态系统云通信，以确定服务状态和各种其它VAS要求，像余额、交易历史、存储的充值等。

[0079] 在示例性且非限制性实施例中，MTP可以被配置用于提供钱包、微件，并且管理交易连同可以与向总体的各种移动环境递送钱包相关联的所有相关商业服务。在示例性且非限制性实施例中，MTP可以被配置用于提供可以帮助构建服务提供者的生态系统和钱包的改进的SDK，其可以创建具有精密特征集的微件而不必担心手持机分片(fragmentation)。在示例性且非限制性实施例中，MTP可以被配置用于建立与各种生态系统的连接器，以向终端用户递送无论何时及何地理位置都可以访问的动态用户体验和各种服务集。

[0080] MTP可以针对来自各个国家的客户进行部署，并且可以被使得适合于特定国家要求。例如，MTP可以使用在美国(USA)、日本、中国、印度、新加坡、墨西哥和玻利维亚，用于各种移动商务和移动支付服务。在示例性且非限制性实施例中，MTP可以被配置用于提供银行服务、票务服务和支付服务。

[0081] 在示例性且非限制性实施例中，MTP可以部署在云中，并且可以被配置为聚集各种不同的服务提供者，以向终端用户有效地提供一套个性化安全交易服务。在示例中，网络提供者可以部署MTP的使能层(例如，钱包和交易管理连同基础设施组件)，其对于服务和信道可以是不可知的。MTP的使能层可以被配置为支持例如银行、商家、医疗保健提供者等的服务提供者的生态系统。MTP的使能层可以被配置为支持可以使用SDK来构建它们的例如安全交易应用的单独个性化应用的所有第三方开发者。

[0082] 图2图示了MTP高层级结构200。MTP 212可以是客户端服务器平台。客户端可以在各种移动手持机上操作，并且服务器可以在服务提供者的“云”中操作。客户端和服务器可以遵循如图1中所示的高层级架构。该高层级架构可以具有增建(build out)生

态系统可能需要的三个不同组件。所述不同组件可以包括移动使能层、服务层和软件开发套件(SDK)。不同组件中的每一个可以与客户端和服务器交互。在示例性且非限制性实施例中,移动使能层可以包括开发移动应用可能必需的基础设施或商业独立组件。在示例性且非限制性实施例中,服务层可以包括用于钱包生态系统和增值服务的参考服务实现的集(collection)。在示例性且非限制性实施例中,SDK 可以包括开发(本文档中描述的)钱包和微件服务以及构建综合生态系统可能必需的一组工具和 API。

[0083] 在示例性且非限制性实施例中,移动使能层可以包括用于客户端和用于服务器的不同组件。例如,用于客户端的移动使能层可以包括通过使用移动电话的特定开发环境所构建的基础设施运行时容器软件(SW)。在示例中,移动使能层可以利用公共开发者 API、UI 解译和渲染引擎、商业执行环境、微件和钱包生命周期管理、用于 OTA/ 邻近交易的模式等来向客户端提供对底层电话能力的公共抽象。在示例中,移动使能层可以提供在网络提供者域中部署的服务器基础设施软件,其可以被配置用于建立和管理安全移动交易生态系统。在示例中,移动使能层可以向服务器提供软件,所述软件被装备用于钱包管理、状态性交易引擎、对象生命周期管理、信道特定通信、多域安全性策略和风险管理、OTA 供应等。

[0084] 在示例性且非限制性实施例中,服务层可以包括用于客户端和用于服务器的不同组件。在示例中,用于客户端的服务层可以包括公共平台服务库,其可以用于实现多个商业用例。公共平台服务可以在由使能层所提供的商业执行环境内运行。在示例中,服务层可以包括三个基本单元。第一单元可以是用例,例如,为各种交易服务所构建的客户端侧工作流和模板。第二单元可以是器件,例如,可以促进到真实世界器件的映射的对应单个钱包器件。在示例中,器件可以提供用于可视化或显示的数据。在示例中,器件可以提供用于空中(OTA)交易的数据。在示例中,器件可以提供用于邻近交易的数据。在示例中,器件可以提供用于监管的数据。第三单元可以包括至少一个信任模型。信任模型可以促进认证方案的实现,其可以包括牵引(tow)因素认证(2FA)。2FA 可以是由不同微件提供者所需的并且可以是与生态系统相关的。

[0085] 在示例中,用于服务器的服务层可以包括相关联的服务模板和商业特定组件的库,其可以部分或整体地重用来开发客户为中心的提供。在示例中,服务可以包括钱包服务 202 和增值服务 204。钱包服务 202 可以包括管理钱包和支付标记的生命周期。增值服务可以包括提供商业服务。服务层可以包括三个单元。第一单元可以是用例,例如,可以通过使用移动使能层的交易框架来构建服务器侧模板以用于实现特定用例。在示例中,可以存在用于商业服务(例如,账单支付、电影票务等)的一个或多个不同的模板。第二单元可以包括至少一个器件或标记。器件可以包括商业特定数据对象(例如,信用卡、银行帐户、账单、优惠券等),其可以跨各种服务被重用并且可以提供针对对象的共同理解定义以及管控对象的生命周期与交易的规则的实现。第三组件可以包括至少一个信任模型。该信任模型可以促进用于服务的不同认证方案的实现,包括对在具有它们的独立 TSM 的单独信任集群内操作的支持。

[0086] 在示例性且非限制性实施例中,SDK 可以包括用于客户端和用于服务器的不同组件。在示例中,用于客户端的 SDK 可以划分成 2 部分。第一部分可以是钱包 SDK 208,并且第二部分可以是微件 SDK 210。在示例中,钱包 SDK 208 可以提供对钱包服务 202、特许 API 以及必要组件的访问以构建钱包容器应用或用户体验。在示例中,微件 SDK 210 可以促进

生态系统合作伙伴构建它们的可以在钱包容器内操作的微件。在示例性且非限制性实施例中，SDK 可以基本上促进客户端侧配置和用户体验。在示例中，SDK 可以通过促进对于由客户端容器展露的参数的可定制性来促进配置。这可以包括设备特定参数、变量（PIN 大小、退出确认、选择模块的包含物等）。在示例中，SDK 可以通过促进跨手持机的差异化用户体验的开发来提供用户体验。

[0087] 在示例中，用于服务器的 SDK 可以包括至少一个库，其可以允许任何生态系统服务器与钱包通信、格式化微件的内容，以及向钱包应用展露服务。该库可以包括配置模块和接口模块。配置模块可以允许对于由服务器的移动使能和服务层所展露的参数的可定制性。可以在属性文件和资源束中配置参数。接口模块可以提供被开发来促进与各种后端系统的通信的连接器，所述后端系统可以协作来履行完整的端对端服务。接口模块还可以包括由平台所展露的 web 服务以集成外围系统，像客户关怀、用户门户等。

[0088] SDK 视觉—在示例中，SDK 可以指代工具和底层平台的 API、实用程序、模式、接口、XML 定义、抽象等的集体集合，其可以被使用和 / 或遵循来促进在 MTP 212 内运行的移动客户端应用或微件的创建。SDK 的可以促进应用的开发，而无需对 MTP 或底层电话环境的深度知识。在示例中，安全、可靠的移动交易的复杂性可以被设计以使得是移动使能层的“隐藏在罩（hood）下”。

[0089] SDK 可以支持对于第三方合作伙伴和声音幻觉（sound illusion，SI）团队的开放工具驱动开发。SDK 可以授权开发者基于由钱包管理 / 使能层提供者（典型为网络提供者）所发行的指南来独立构建和分布它们自己的个性化安全交易微件。SDK 可以开放各种层（基本上顶至底）。SDK 可以相对于支持多手持机 OS、鲁棒消息传送和通信、安全性和认证以及用于支付和非支付服务的基于 NFC 的安全邻近交易，提供对平台的横向能力的访问。SDK 可以简化钱包和微件的开发。在示例中，MTP 还可以具有用于直接构建原生（native）钱包的供应，以利用可能还未通过 SDK 展露的底层功能性。这可以给予开发者 SDK 和 MTP 二者中的最佳的，从而具有足够的灵活性以构建改进的用户体验。

[0090] 图 3 图示了 MTP 服务器 300 的详细架构。MTP 服务器可以是用于构建钱包生态系统的使能基础设施，所述钱包生态系统可以支持移动商务（例如通过 OTA）和移动支付（例如通过 NFC 邻近）交易服务。服务器可以提供前端网关，从而扩展由生态系统合作伙伴或服务提供者 324（例如，商家、银行和运营商）通过移动钱包应用而提供给多个移动手持机 328 的服务。在示例性且非限制性实施例中，服务器可以包括组件集合，其可以协作来提供钱包管理、交易管理和生态系统集成。服务器可以用 Java 来开发并且可以支持各种应用服务器和数据库。在示例中，服务器可以是用于正由 MTP 提供的服务的容器。在示例中，服务器可以托管在集群化设置中的安全 DMZ（非军事化（demilitarized）区）中，用于具有至生态系统云的连接性的关键任务应用。

[0091] 如图 3 中所图示，MTP 架构可以包括服务器基础设施组件、服务模板和参考实现、NFC 基础设施组件和主机集成框架。

[0092] 在示例中，MTP 服务器可以包括多个信道 302、通信模块 304、OTA 组件 308、钱包管理模块 310、交易引擎 312、通知和服务总线 314、指令管理模块 318、至少一个增值服务（VAS）模块 320、NFC 基础设施 322 以及主机集成框架。在示例中，信道 302 可以提供信道特定适配，以用于要通过多个信道来接收的请求，诸如移动数据、SMS、USSD 等。在示例中，

通信模块 304 可以提供用于支持安全移动客户端服务器交互的框架。通信模块 304 可以促进来自单独商业组件的通信、认证、授权等。灵活架构可以促进服务开发者通过添加附加层来扩展安全性或者引入新要求。在示例中，OTA 组件 308 可以提供对于促进可以在交易期间使用的应用、微件和个性化标记的 OTA 递送可能需要的支持。在示例中，OTA 供应可以包括设备发现并确保可以将合适的版本递送给设备。在示例中，钱包管理 310 模块可以提供对于钱包状态管理、生命周期服务、版本管理、钱包历史等的实现。支付服务(例如软卡和个性化小程序的供应)也可以被包括作为钱包管理的一部分。在示例中，交易引擎 312 可以提供用于开发 OTA 交易服务的框架。可以通过遵循由交易框架所提供的模板来开发交易服务实现。可以通过交易框架来透明地管理可靠性、安全性、日志记录等的要求。交易框架可以提供对与内部虚拟帐户和外部支付网关的接口的集中式访问。在示例中，通知和消息总线 314 可以提供与客户端应用的对象交换总线。在示例中，通知和消息总线 314 可以包括对于向钱包推送警报和消息的支持。在示例中，器件管理模块 318，可以存在若干组件，所述组件可以提供各种钱包器件(例如，信用卡、账单、票据等)并且管理与平台内的这些器件相关联的各种生命周期服务和配置。在示例中，VAS 模块 320 中的至少一个可以包括用于公共商业服务的服务模板和实现方式的库，其可能已经通过使用 MTP 平台而被构建和部署。在示例中，NFC 基础设施 322 组件是这样的一组组件，所述组件可以与 NFC 生态系统主机集成以用于精心安排(orchestrate)软卡和小程序的发行和个性化。所述集成可以是对于使能来自生态系统的全局平台顺从的 NFC 交易服务所需的支持。在示例中，主机集成框架可以是包括各种连接器的层，所述连接器可以插接到服务器中以与后端系统通信以用于促进交易服务。主机集成框架可以包括一系列接口定义，所述接口定义可能必须由服务开发者以连接器的形式来实现，以与外部主机系统通信。

[0093] 在本发明的各种示例性且非限制性实施例中，MTP 服务器的主要职责可以包括为钱包用户社群提供钱包管理。在示例中，钱包管理可以包括钱包生命周期服务、支付、微件管理等。在本发明的各种示例性且非限制性实施例中，MTP 服务器的主要职责可以包括以安全和可靠方式构建服务生态系统和精心安排来自钱包应用的与生态系统服务提供者的交易。在本发明的各种示例性且非限制性实施例中，MTP 服务器的主要职责可以包括在构建和分布交易微件并且管理微件发现和递送中支持第三方开发者社群。在本发明的各种示例性且非限制性实施例中，MTP 服务器的主要职责可以包括提供用于实现移动金融交易的所定义的模式和模板，其从服务开发者隐藏安全性、可靠性等的复杂性。

[0094] 图 4 图示了具有 MTP 移动客户端运行时 400 的钱包应用的高层级架构视图。出于该图的目的，钱包 404 和微件 408a 与 408b 可以与上面描述的模块类似，但是在功能性和职责上不同。在本发明的各种示例性且非限制性实施例中，MTP 移动客户端 402 可以是用于快速移动应用开发的使能基础设施。移动客户端可以提供基于容器的途径来支持多租户(multi-tenant)应用架构。在示例中，移动客户端容器可以是电话特定运行时环境，用于可以通过使用 MTP SDK 所开发的钱包 404 和微件 408a 与 408b。移动客户端容器可以促进应用执行、交易安全性、可靠性、设备特定实现等，从而允许开发者更多地集中于服务的独特功能。移动客户端容器可以构建在设备特定 API 的顶部并且可用于流行设备环境，例如，电话 API 444。移动客户端运行时可以从微件描述符生成原生应用屏幕。该屏幕可以确保屏幕上的按钮可以看起来好像相应设备上的原生 Android®、Blackberry® 或 iPhone® 按

钮。移动客户端服务器可以包括至少一个电话独立的 API 410。电话独立 API 410 可以确保通过使用 SDK 开发的微件 408a 与 408b 跨所有支持的手持机而运行。电话独立 API 410 可以特别地设计来提供开发者创建改进的用户体验可能需要的所有支持。可以配置移动客户端服务器 402 以便促进底层电话 O/S 内的原生实现，因为其可以通过运行时和 API 来抽象。在示例中，钱包应用可以像微件一样，因为钱包应用可以由移动客户端运行时 402 来解译和执行。在示例中，钱包 404 可以具有在移动应用上的特许控制。钱包 404 还可以为微件开发者定义附加 API，其可以特定于应用的上下文(例如，有关于金融行业、零售行业、医疗保健行业、政府)。

[0095] 在本发明的各种示例性且非限制性实施例中，移动客户端运行时可以具有可以包裹所有主要电话平台开发者 API 的组件。在示例中，UI 框架和工作流引擎可以渲染用户接口并操控本地商业功能的基于事件的执行。

[0096] 在本发明的各种示例性且非限制性实施例中，移动客户端运行时可以包括以下组件。移动客户端运行时可以包括 UI 框架 412。在示例中，UI 框架 412 可以提供必要的基础设施来渲染用户接口。在示例中，UI 框架 412 可以包括 UI 引擎 414、UI 屏幕和布局 418、样式集 420 以及多个 UI 元素 422。UI 框架 412 可以被配置为管理各种各样的手持机和屏幕能力。在示例中，UI 引擎 414 可以基于在应用 XML 中提供的屏幕配置来管理屏幕 418 的生成和显示。移动客户端运行时 402 可以包括行动框架 424。行动框架 424 可以提供必要的支持来执行商业特定逻辑(行动)。行动框架 424 可以以定制 Java 脚本(JavaScript)或预先存在的平台服务的形式。行动框架可以包括脚本引擎 428。脚本引擎 428 可以控制行动的执行以及与用户接口的交互。移动客户端运行时 402 可以包括微件管理组件 430。微件管理组件 430 可以促进管理微件生命周期服务 432，诸如安装、加载、卸载、移除、更新等。微件管理组件可以包括访问控制器 434，用以管理访问控制以仅准许微件访问它们在其微件描述符中具有针对其的准许的资源。移动客户端运行时 402 可以包括通信组件 438。通信组件 438 可以被配置用于提供在客户端与服务器应用之间的基本通信服务。通信组件 438 可以提供对于 HTTP (&HTTPS) 以及 SMS 信道的通信支持。在示例中，通信组件 438 可以包括分组构造、编码 / 解码和基本 HTTP 差错处理。在示例中，通信组件 438 可以支持至用于像警报和广告一样的非必需商业对象的通信分组的附接。移动客户端运行时 402 可以包括数据库组件 440。数据库 440 可以提供服务来存储、更新、移除和寻找来自本地数据库的数据。在示例中，数据库 440 可以提供可能从原生电话平台不可获得的 API。移动客户端运行时 402 可以包括安全性组件 442。安全性组件 442 可以提供加密 API，用于加密、解密、基于散列的消息认证码(HMAC)、安全随机散列法等。移动客户端运行时可以包括 NFC 组件。NFC 448 可以实现全局平台顺从 NFC 服务，并且提供用于与电话的安全元件上的小程序交互的 API。NFC 448 可以包含用于 NFC 标签和对等(P2P)模式的 API。移动客户端运行时可以包括电话组件。电话可以提供平台服务以与电话组件相交互，所述电话组件像呼叫、SMS、LED、振动命令、日历功能、联系人管理等。移动客户端运行时可以包括微件 408a 与 408b。微件可以是服务束，其可以封装用于商业服务集的屏幕、内容、资源、Java 脚本和样式。微件可以通过使用 MTP SDK 来构建，所述 MTP SDK 可以包括在 XML 文件的预定义集上操作的工具集。

[0097] 在本发明的各种示例性且非限制性实施例中，移动客户端可以是在其之内可以安

装和运行微件的容器。在本发明的各种示例性且非限制性实施例中，移动客户端可以为整个移动应用来简化端对端可视工作流的开发。在本发明的各种示例性且非限制性实施例中，移动客户端可以提供一组良好测试的、标准化的 API 和平台服务，以用于应用开发者快速使用来构建他们的跨移动手持机平台的应用，而没有设备分片问题的风险。在本发明的各种示例性且非限制性实施例中，移动客户端可以生成改进的用户接口，其可以利用原生智能电话能力，但是仍然可以在有限能力设备上是响应性的。在本发明的各种示例性且非限制性实施例中，移动客户端可以提供用于实现移动金融交易的所定义的模式，其可以被配置使得具有安全性、可靠性等的复杂性，如后端操作那样。

[0098] 图 5 图示了参照开发周期的微件开发过程的概观。

[0099] 在示例性且非限制性实施例中，MTP SDK 可以被配置用于促进整体微件开发 500 并且向微件开发者提供更大程度的灵活性和独立性。微件开发中的第一阶段可以是设计阶段 502。在示例中，开发者可以通过使用 Microsoft office Visio® 或类似工具 & 任何 XML 及 Java 脚本编辑器来创建移动应用。对开发进一步地，微件或微件文件 504 可以被封装。在示例中，开发者可以使用 eclipse 插件来封装微件工作空间并创建微件包（package）。可以将微件包发送到自动化构建服务器以用于微件生成 508。微件文件 504 可以不以微件二进制 514 的形式。在示例中，自动化构建服务器可以生成包括移动客户端运行时以及微件包的应用二进制。对封装进一步地，可以使用测试容器 510 来测试微件。在示例中，开发者可以利用任何现成设备模拟器来核实微件流和脚本执行。对测试进一步地，微件可以被发布到 MTP 服务器 512。在示例中，钱包用户可以能够发现微件并从 MTP 服务器 512 下载它们。

[0100] 在示例性且非限制性实施例中，多个工具可以用于微件开发，例如，MTP-SDK、MTP Microsoft Visio 外接附件（add-on）以及 MTP eclipse 插件。

[0101] 在示例中，MTP SDK 用于服务器侧和客户端侧开发。客户端侧 SDK（也称为移动客户端 SDK）对于开发任何移动客户端微件可以是基本的。服务器 SDK 可以提供一组有用的协议和合约，其可以促进开发者在 MTP 的被证明的使能框架上构建。在示例中，还可以在不使用 MTP 服务器的情况下开发用于微件的独立 web 服务。

[0102] 在示例性且非限制性实施例中，MTP 移动客户端 SDK 可以包括一组 API，以访问各种原生电话平台服务，像数据库、无接触、通信等。所述 API 可以被提供作为 Java 脚本函数并且可以由微件开发者利用。在示例性且非限制性实施例中，MTP 服务器 SDK 可以包括有用的库和合约，其可以使能与移动客户端微件的利用 MTP 的用于通信、安全性、可靠性、失效备援（failover）等的底层实现的快速集成。

[0103] 在示例中，诸如用 SDK 进行的 MTP 的元件的开发可以由可视化开发能力（例如，Microsoft Visio®）来支持，其促进开发者通过在 MTP 模板、型版（stencil）等的情况下执行拖放动作来定义用户接口工作流。MTP 外接附件可以促进工作流至 MTP 中、电话无关、XML 格式（例如，MTP 应用描述符）的导出。

[0104] 在示例性且非限制性实施例中，MTP eclipse 插件、MTP SDK 可以由可以用于从应用描述符生成钱包应用或微件的 Eclipse 插件所支持。MTP eclipse 插件可以为所有所支持的设备平台提供单击构建选项。

[0105] 移动交易平台可以包括用于诸如容器、钱包、微件之类的元件和用于诸如金融、零

售、健康、政府等之类的应用的参考实现。移动交易平台的特征和功能可以通过多个部署模型来递送，其可以包括使用可以由 MTP 工具支持的原生客户端应用(例如，容器或移动客户端)，提供者可以使用所述 MTP 工具来开发客户端功能，诸如钱包、微件、伴随小程序、优惠券发放小程序等。类似地，MTP 可以促进(可以为每个提供者提供和 / 或定制的)web 服务的部署，其可以促进展露 MTP 的不同特征等。诸如本文描述的多层安全移动交易使能平台之类的 MTP 可以包括可以用于开发钱包、微件、伴随小程序等的工具。用于配置和递送钱包、微件等的这些工具和方法还可以配置有向导(wizard)类实用程序以加快生态系统开发。附加的 MTP 工具可以被配置为监视使用和交易信息，并且此外基于各种能力和性能准则等来贡献用于优化的智能选项和解决方案。

[0106] 多层安全移动交易使能平台可以以对于可伸缩性、冗余性、互操作性、服务递送、消息定义等的必要支持来满足企业服务递送环境预期和需求。可以通过托管的用于服务 / 发行 / 获取提供者的模型和 / 或通过受管理的服务模型来递送平台。MTP 促进跨如本文所描述的生态系统来实现规则(例如，商业、交易、信任等)，其可以管控各种生态系统操作，除其它外尤其包括 :参与者登上到生态系统中、参与者如何通过生态系统来访问和消费各种服务、如何可以从生态系统移除参与者、生态系统内的不同参与者如何可以彼此协作、不同参与者如何可以与生态系统外部的实体协作、参与者如何获取客户、参与者将如何支持客户、参与者如何可以捕捉与用户和交易相关的数据(静态和动态)、参与者将如何在生态系统内和外使用这样的数据、共享这样的数据和将这样的数据货币化。

[0107] MTP 可以与专家引擎相关联，用于向生态系统中的客户以及提供者 / 参与者提供客户支持。可以利用专家引擎以通过本文描述的方法和系统来向客户以及提供者 / 参与者递送简化的用户体验。MTP 可以架构有如本文所描述的若干层或小面(facet)，并且为了描述而被标签为使能、服务、个性化等，其中每个层潜在地进一步被划分成其它层，诸如逻辑和物理层或方面。这样的多层架构或平台可以独特地帮助在如本文所描述的多域(和可选地多租户)生态系统中递送服务和功能性，而同时确保在服务提供者之中的独立所有权并支持差异化能力的演进，尤其是在多域生态系统环境中。

[0108] 可以通过可以促进生态系统宽泛能力、钱包和微件的容器来递送 MTP，所述钱包可以促进差异化器件实现相关的能力和特征(例如，NFC、具有安全元件(SE)的 NFC、标记化、通过 QR 码 / 条形码所展露的标记化等)，所述微件可以促进提供者可以在生态系统内不经媒介(disintermediation)的情况下直接向外展露给终端用户的差异化能力和特征。

[0109] 诸如本文描述的多层安全移动交易使能平台之类的 MTP 可以基于生态系统参与者、规则等的后端要求来确保跨诸如本文描述的多域生态系统之类的生态系统的不同前端协议的聚集。可以此外通过提供抽象或包裹器(wrapper)能力和 / 或包裹器能力的标记化来优化不同前端协议的聚集。

[0110] MTP 可以通过本文描述的方法和系统在企业服务递送环境上使得生态系统能够将协议层级功能性逐渐递送到前端中。可以此外通过提供本文描述的包裹器的标记化或聚集能力或者通过使用生态系统规则来增强协议层级功能性递送。

[0111] MTP 可以设置在网络提供者的域中并且其部分集成到它们的系统中。MTP 可以附加地连接到服务和安全性提供者的生态系统，包括一个或多个可信服务管理器(TSM)。网络提供者可以使用 MTP 来创建以特定商业垂直面为目标的品牌移动应用。这样的品牌移动应

用可以是用于例如零售 / 金融域的钱包，或者例如用于医疗保健垂直面的钱包。这样的钱包应用可以向用户提供核心服务集，所述核心服务集特定于其被创建所针对的域。除了提供核心域服务之外，钱包还可以操控应用生命周期和安全性、标准化的用户体验以及钱包管理职责。微件可以提供特定商业服务并遵照钱包指南。

[0112] 图 6 描绘了如本文所描述的用于安全个性化交易的生态系统的概观。生态系统 600 可以包括多个参与者(诸如客户端设备的用户 602、器件的发行者 610、交易器件的获取者 620、服务提供者或商家 608)以及包括交易和处理资源 604 的各种基础设施元件，所述交易和处理资源 604 可以包括移动交易平台(MTP)622、规则、与工具和参与者的接口，其可以具体化为一个或多个企业服务总线(ESB)、销售点设施 612、安全销售点转换功能性 618、用于抽象交易协议的功能性 614 等。客户端设备 602 可以包括多个容器，其可以包括用于执行针对一个或多个服务 / 产品的交易(例如，支付交易)的多个钱包、钱包伴随小程序和多个微件，所述一个或多个服务 / 产品可以由诸如服务提供者 608 之类的服务提供者所供应。

[0113] 图 6 中描绘的 MTP 622 可以设置在网络提供者的域中并且集成在网络提供者的系统中(可选地通过它们的 ESB)。如所示出的，MTP 可以被配置为连接在服务提供者的生态系统 600 内。单独、独立的服务提供者或商家 608 可以使用 MTP 软件开发套件和环境来构建可以加载到在客户端设备 602 上可访问的钱包或钱包容器中的微件。这些微件可以提供增量式功能性，其可以增强钱包应用的整体吸引力。此外，MTP 基础设施 604 可以包括规则，诸如用以提供钱包、微件和交易管理连同与向总体的(a universe of)各种移动环境递送钱包相关联的所有相关商业服务。此外，MTP 可以建立与各种生态系统的连接，以向终端用户递送可以随时随地访问的动态用户体验和各种服务集。

[0114] 生态系统 600 此外可以包括多个交易器件发行者 610。发行者 610 可以提供各种交易器件，诸如虚拟信用卡、礼品卡、忠诚卡、票据等。客户端的用户 602 可以通过使用一个或多个钱包和微件来选择这些器件以执行交易。在示例中，多个发行者 610 可以使用与 MTP 622 相关联地提供的工具来开发定制的微件以增强用户体验、提供安全性、遵照发行者工作流等。此外，生态系统 600 可以包括数字钱包(例如，其可以安装在移动客户端 602 上)，其可以包括用于存储这些器件、执行针对这些器件的交易等的功能性。

[0115] 生态系统 600 此外可以通过以下能力和特征来表征，所述能力和特征诸如：对于个性化交易的三维安全性(例如，设备、域和用户)、多个功能性层(例如，具有个性化、服务、使能层、专家和体验框架层或层次的架构)、对于客户端设备 602 上的多个独立容器的支持、容器内的多个不同微件、容器内多个不同且独立的钱包、对多个钱包的单个微件访问、通过多个微件的单个钱包访问、对于多个信任模型的支持等。生态系统 600 此外可以包括销售点设施或功能性 612，其可以被配置为执行真实(例如，基于邻近)和 / 或虚拟(例如，在线)交易。生态系统 600 可以包括协议抽象组件 614，其可以包括包裹器，诸如用以跨多种类型的协议在客户端 602 与销售点设施 612 之间执行交易等。例如，协议抽象组件 614 可以使客户端设备 602 能够独立于销售点设备 612 或聚集器交换机(aggregator switch)618 通过其而进行操作的协议来执行交易。

[0116] 生态系统 600 此外可以包括多个获取者 620。获取者 620 中的每一个可以是与 MTP 622 相关联地提供的使用工具，用于开发定制的微件、钱包等。在示例中，生态系统 600 可以包括交换机 618，用于从多个获取者选择获取者 608 用于执行交易。

[0117] 如图 6 中所描绘的，发行者 610、服务提供者 608、获取者 620 等可以以生态系统 600 功能性的仅部分而具有对用于安全个性化交易的生态系统的其部分的访问或者可以配置所述部分。在示例中，获取者 620A 包括钱包功能性，而获取者 620n 包括微件功能性。同样地，发行者 610a 包括钱包和微件功能性，而发行者 610n 仅包括微件。类似地，服务提供者 608a 包括微件、钱包和容器，而服务提供者 608n 仅包括微件和钱包。

[0118] 图 7 描绘了用于使用在用于安全个性化交易的生态系统中的示例性客户端架构 700。客户端架构 700 可以包括基于容器的途径，诸如用以支持多租户应用架构。客户端架构 700 可以被配置成可以由设备独立 API、一个或多个设备特定的操作系统环境等支持的商业特定服务、钱包功能性、使能基础设施等。钱包功能性可以是基于容器的，用以提供包括商业规则附着、钱包和微件应用生命周期管理、用户体验管理的功能性，并且此外可以包括一个或多个容器 714，其可以包括一个或多个钱包 702。商业特定服务可以由一个或多个微件提供，诸如微件 704a、微件 704b、微件 704c……，以及微件 704n(本文总称为微件 704)。通过客户端运行时环境 708 中的设备不可知 API 和服务执行环境可以促进使能基础设施。

[0119] 此外，客户端架构 700 可以包括移动客户端运行时容器 708，其可以是用于通过使用 MTP 软件开发套件(SDK)所开发的钱包 702 和微件 704 的电话特定运行时环境。该容器可以操控与应用执行、交易安全性、可靠性等相关的复杂性，从而这可以允许开发者更多地集中于服务的独特功能并且不担心包括设备特定实现的大部分复杂性。移动客户端容器 708 可以构建在设备特定 API 的顶部并且可用于流行的设备操作系统环境 710。运行时容器 708 可以从与微件 704 相关联的描述符生成原生应用屏幕。这确保按钮在相应设备上看起来像原生 ANDROID、BLACKBERRYOS 或 IPHONEOS 按钮。

[0120] 客户端架构 700 此外可以包括设备独立 API 层次 710，其可以确保通过使用 SDK 所开发的微件可以跨所有支持的手持机而运行。这些 API 特别设计用于提供开发者创建丰富用户体验所需要的所有支持。然而，开发者可以不需要担心底层电话 O/S 内的原生实现，因为这通过客户端运行时容器 708 和 API 层次 712 而被抽象出。钱包 702 可以像微件那样执行，因为其可以由客户端运行时 708 解译和执行。然而，钱包 702 可以具有对应用的特许控制。钱包 702 还可以为微件开发者定义可以特定于应用的上下文(金融、零售、医疗保健、政府等)的附加 API。

[0121] 图 8 描绘了包括具有协议抽象支持能力的 MTP 结构配置的用于安全个性化交易的生态系统。系统 800 可以包括通信地耦合到网络 810 的移动设备 802、MTP 服务器 804 和多个服务提供者，诸如服务提供者 808a、服务提供者 808b、服务提供者 808c……，以及服务提供者 808n。移动设备 802 可以包括多个容器，诸如容器 812a 和容器 812n。容器中的每一个可以包括多个钱包，诸如钱包 814a 和钱包 814n (本文中总称为钱包 814)。容器中的每一个可以包括多个微件，诸如微件 818a 和微件 818n (本文中总称为微件 818)。

[0122] 不同的服务提供者可以利用如本文所描述的移动交易平台和 / 或钱包服务中心(WSC)的不同特征。例如，服务提供者 808a 可以访问 MTP 的钱包服务中心工具套件能力，服务提供者 808b 可以访问钱包服务中心插件能力，服务提供者 808c 可以访问钱包服务中心 web 门户服务，以及服务提供者 808n 可以提供不需要任何专用 WSC 或 MTP 驻留软件的服务包。各种服务提供者或商家可以通过网络 810 来访问 WSC 开发环境和 WSC 运行时。例如，服务提供者 808n 可以不被要求运行工具套件，并且可以直接访问如在 MTP 服务器 804 上可

用的WSC工具套件。类似地，MTP服务器804可以向各种服务提供者提供对库的访问，其中，所述库可以包括用于编码／解码的编解码器、认证／安全性方案等。不是每个服务提供者被要求运行WSC库。

[0123] 在示例中，MTP服务器804可以被配置为向多个服务提供者提供开发环境以及运行时环境。在任何时刻，服务提供者可以具有插件、库，或者可以从MTP服务器804直接访问这些组件。结果，该系统允许服务提供者或商家按照所期望的要求来定制配置。可替换地，MTP服务器804可以与诸如移动设备802之类的客户端设备直接连接以执行交易，诸如代表服务提供者(例如，PL808n)。

[0124] 生态系统可以包括诸如通过各种销售点(PoS)终端和通过空中交易的获取功能性。可以通过获取模块824可选地促进与交易获取相关联的协议抽象，所述获取模块824可以合并MTP和／或WSC获取功能性822，其可以向系统添加更多功能性以诸如使得交易更可靠、鲁棒和安全。PoS终端820可以经由各种获取协议中的任一种、通过一个或多个邻近通信信道来与移动设备802通信，所述信道包括但不限于NFC、蓝牙、条形码读取器等。可以经由包裹移动设备802与PoS终端820之间的这样的通信来操控协议抽象。在示例中，包裹器可以被实现用于抽象这样的协议：所述协议具体化钱包824与PoS终端820之间的交易。对于空中交易(例如，其不直接使用PoS终端802)，可以经由包裹器、标记、标记化的包裹器等类似地生成协议抽象，所述包裹器、标记、标记化的包裹器等可以用于通过公共网络810(例如，因特网)来交换信息以实现交易。

[0125] 图9描绘了如图8中所描绘的用于安全个性化交易的生态系统的可替换配置，其不使用获取设施(例如，图8中的824)。

[0126] 本文描述的多域生态系统安全个性化交易的方法和系统包括在多域生态系统中用于安全个性化交易的诸如平台的系统。所述平台可以包括个性化层，用于使能存储在移动设备上的一个或多个生态系统元件的服务提供者个性化。可以从包括微件、钱包和小程序的组中选择所述一个或多个生态系统元件。所述平台此外可以包括使能层，用于促进个性化层与客户端设备之间的互操作。所述平台此外可以包括独立于使能层而操作的服务层，用以使能针对多个服务的服务递送。在示例中，可以从包括银行业务、票务和账单支付的组中选择所述多个服务。所述平台在生态系统参与者的分布式网络中可以是可操作的。生态系统参与者可以包括诸如服务提供者、器件发行者、器件获取者以及移动设备用户。所述平台此外可以包括多个功能模块。所述功能模块可以包括用于诸如通信、通知、安全性、认证、空中操作之类的一个或多个模块以及用以管理钱包和微件的设备运行时容器。所述平台此外可以包括软件开发环境，其可以包括诸如容器、钱包和微件的参考实现。软件开发环境此外可以包括诸如金融、零售、健康和政府域的参考实现。在示例中，可以通过使用原生客户端应用来部署所述平台。在示例中，可以通过使用原生客户端应用和一组软件开发工具来部署所述平台。该组软件开发工具可以用于开发客户端功能。可以从钱包、微件、伴随小程序和优惠券发放小程序的组中选择该组客户端功能。在示例中，可以通过使用原生客户端应用和一组web服务来部署所述平台，该组web服务可以由平台展露以集成外围系统。所述外围系统可以包括诸如客户关怀和用户门户。在示例中，可以用这样的工具来部署所述平台：所述工具可以用于诸如开发钱包、微件和伴随应用。所述工具可以包括向导类实用程序。在示例中，所述工具可以用于监视使用和交易信息。在示例中，所述工具可以用

于提供智能优化选项和解决方案。优化解决方案可以基于所监视的能力和性能准则。在示例中，所述平台可以被部署有企业服务递送环境中的必要支持。所述必要支持可以包括诸如对可伸缩性、冗余性、互操作性、服务递送以及消息定义的支持。在示例中，可以通过用于多个提供者的托管的模型来部署所述平台。可以从包括诸如服务、发行和获取提供者的组中选择所述多个提供者。在示例中，可以通过实现规则来递送所述平台，所述规则可以管控由生态系统参与者对服务的使用。在示例中，所述规则可以管控由参与者对服务的访问。在示例中，所述规则可以管控由参与者对服务的消费。在示例中，所述规则可以管控如何可以从生态系统中移除参与者。在示例中，所述规则可以管控在生态系统内参与者如何可以与彼此协作。在示例中，所述规则可以管控参与者如何可以与生态系统外部的实体协作。在示例中，所述规则可以管控参与者如何可以获取客户。在示例中，所述规则可以管控参与者如何可以支持客户。在示例中，所述规则可以管控参与者如何可以捕获与用户和交易相关的数据。在示例中，用户和交易可以是静态的。在示例中，用户和交易可以是动态的。在示例中，所述规则可以管控参与者如何可以在生态系统内使用所捕获到的数据。在示例中，所述规则可以管控参与者如何可以在生态系统外部使用所捕获到的数据。在示例中，所述规则可以管控参与者如何可以在生态系统内共享所捕获到的数据。在示例中，所述规则可以管控参与者如何可以在生态系统外部共享所捕获到的数据。在示例中，所述规则可以管控参与者如何可以在生态系统内将所捕获到的数据货币化。在示例中，所述规则可以管控参与者如何可以在生态系统外部将所捕获到的数据货币化。在示例中，可以通过利用专家引擎来向参与者提供客户支持以使得可以从客户和提供者的组中选择参与者来递送所述平台。在示例中，可以通过利用专家引擎来向参与者提供简化的用户体验以使得可以从客户和提供者的组中选择参与者来递送所述平台。在示例中，可以通过使用多层来架构所述平台。可以从包括诸如使能层、服务层和个性化层的组中选择所述多层。每层可以进一步被划分成附加层。可以从包括诸如逻辑层和物理层的组中选择所述附加层。

[0127] 用于安全个性化交易的多域生态系统平台可以包括用于生态系统元件的服务提供者个性化的个性化层、独立于服务层而操作的用于促进与客户端设备软件和硬件的互操作的使能层，所述服务层促进针对诸如银行业务、票务、账单支付等之类的各种服务的服务递送。这样的多域生态系统平台可以促进包括安全个性化交易的一般和专门的交易，可以在逻辑上和／或物理上分割成多个功能独立块、分段、单元、层次或层，使得它们结合通过接口使用数据或信息的交换来分离地操作，或者以预定或可配置顺序来与彼此无缝地操作，以履行较大功能或单元的交易或完整交易。作为示例，如本文所描述的多层生态系统使能平台可以由三层构成，即，个性化层、服务层和使能层。

[0128] 此外，作为示例，个性化层可以用于实现特定品牌化元素(诸如商标、徽标、主题、颜色等)、用户交互元素、工作流(例如，存储的帐户和交易数据，用以使能高效完成重复交易、购物车、单击交易以及许多其它)、用户接口效果(例如，过渡、内容格式、动态或任何其它行动)、可能由服务提供者对于特定要求所需要的原因和／或效果。个性化层次可以促进服务提供者的协作设计或交互哲学的部署，以及为消费者给出空间来调整至它们的个人需要和偏好，包括针对视觉上受损等。

[0129] 使能层可以促进与底层层次、平台或基底的交互，所述底层层次、平台或基底可以以软件或硬件或二者的组合的形式以与其通信和交互。使能层可以或者可以不充当将高于

其的层次或层与低于使能层的那些有效隔离的抽象。

[0130] 服务层可以迎合诸如工作流、商业逻辑、规则、智能之类的特定要求以及服务垂直面的需要，所述服务垂直面包括诸如但不限于以下各项的服务垂直面：银行业务、票务、账单支付、优惠券发放、忠诚及积分卡程序、停车以及其它生态系统服务。

[0131] 由这样的服务层所使能的服务可以包括移动银行业务、转账、账单支付、票务、优惠券发放、忠诚程序、市场和广告相关服务、教育相关服务、城市服务、移动健康服务、移动保险、诸如停车优惠券之类的运输服务、旅行服务、紧急情况或 911 服务、零售支付、后勤支持服务、商业智能解决方案、品牌化、购物、产品认证服务、调节服务、记录管理、交互电视服务、文本到语音服务、基于位置的服务等。

[0132] 尽管本文使用的示例说明了多层生态系统使能平台的概念，但是本文描述的方法和系统实际上涵盖更一般化的平台架构和 / 或结构概念，其包括多个功能独立的块、分段、单元、层次或层。图 10 描绘了可以跨越一系列生态系统元件的分层平台的概念，所述元件包括客户端设备(例如，移动设备)、钱包服务中心功能性、可信安全性管理器、移动交易平台服务器、钱包发行者、商家(亦称服务提供者)等。每个这样的生态系统元件可以至少包括使能、服务和个性化层。如本文中所注意到的，还可以在多层安全移动交易生态系统使能平台的部署中并入附加层，诸如专家引擎和体验框架层。

[0133] 本文描述的多域生态系统安全个性化交易的方法和系统包括一种方法，其包括分析包括以下各项的集合中的一个或多个：生态系统参与者、服务要求、客户要求、调节要求、交易安全性要求、信任模型以及电子交易通信信道。所述方法此外可以包括基于分析来确定一个或多个生态系统特征以配置在包括服务层、使能层和个性化层中的至少一个的平台中。所述方法此外可以包括根据所述确定来配置所述一个或多个生态系统特征。在实施例中，所述一个或多个生态系统特征可以被配置以确保在服务提供者商业模型的上下文内满足每个所分析的要求。所述平台可以附加地包括专家引擎层和体验框架层。

[0134] 用于选择生态系统特征以用于包括在使能层、服务层和个性化层的每一个中的方法可以包括分析生态系统参与者、服务要求、客户要求、调节要求、交易安全性要求、信任模型以及电子交易通信信道，以确定生态系统特征以配置在服务层、使能层和个性化层中以确保在服务提供者商业模型的上下文内满足每个所分析的要求。可以对于诸如专家引擎和体验框架层之类的附加层类似地确定特征。

[0135] 将整体交易、服务或过程的部分分解成多层生态系统平台的方法或过程可以基于分析法、启发法和要求的组合，可选地通过包括专家引擎的工具基于人工智能概念来提供帮助。这样的方法或过程可以考虑生态系统中所有参与(目前和预期的将来)实体的视角和要求，所述实体包括但不限于：服务提供者；服务要求；客户要求；终端用户要求；调节委托 & 指南；安全性委托、指南和要求；信任模型；经济模型；商业逻辑和工作流；与其它服务的公共性；操作因素；设计 & 技术因素；目标设备 & 平台；规划的技术演进；其它相关 & 不相关服务；以及操作约束。

[0136] 在大多数情况中，上面提及的方面和参数可以导致服务要求在逻辑上被分割或分区成三层，即，个性化、服务和使能层。

[0137] 参照图 11，选择生态系统特征以用于包括在用于安全个性化交易的多域生态系统平台的使能层、服务层和个性化层的每一个中的实施例。基于参与者商业目标和要求，生

态系统特征分配引擎 1102 可以处理规则和控制，以诸如将生态系统的特征分配给使能层 1104、服务层 1108 和个性化层 1110。

[0138] 在示例中，使能层 1104 可以包括对于开发移动应用可能必需的基础设施或商业独立组件。服务层 1108 可以包括对于钱包生态系统和增值服务的参考服务实现集。个性化层 1110 可以包括一组工具和 API，其可能对于开发(本文档中描述的)钱包和微件服务以及构建综合生态系统是必需的。

[0139] 在示例中，使能层 1104 可以包括对于客户端和服务器不同的特征。在使能层 1104 中，客户端可以被配置为包括基础设施运行时容器软件(SW)，其可以通过使用与移动设备相关联的特定开发环境来构建。在示例中，使能层 1104 可以向客户端提供对底层设备能力的公共抽象，所述底层设备能力具有公共开发者 API、UI 解译和渲染引擎、商业执行环境、微件和钱包生命周期管理、用于 OTA/ 邻近交易的模式等。在服务器侧，使能层 1104 可以向服务器提供可以在网络提供者的域中部署的基础设施软件，其可以被配置用于建立和管理安全移动交易生态系统。在示例中，使能层 1104 可以配置服务器特征，诸如钱包管理、状态交易引擎、对象生命周期管理、信道特定通信、多域安全性策略和风险管理、OTA 供应等。

[0140] 在示例中，服务层 1108 可以包括用于客户端的多个特征。例如，客户端可以被配置为包括可以用于实现多个商业用例的公共平台服务库。公共平台服务可以在由使能层 1104 提供的商业执行环境内运行。在示例中，服务层 1108 可以包括三个基本单元。第一单元可以是用例，例如，为各种交易服务所构建的一组客户端侧工作流和模板。第二单元可以是器件，例如，可以促进到真实世界器件的映射的对应单个钱包器件。在示例中，器件可以提供数据用于可视化或显示。在示例中，器件可以提供数据用于空中(OTA)交易。在示例中，器件可以提供数据用于邻近交易。在示例中，器件可以提供数据用于监管。第三单元可以包括至少一个信任模型。信任模型可以促进对于认证方案的实现，其可以包括两因素认证(2FA)。2FA 可以是由不同微件提供者所需的并且可以是与生态系统相关的。

[0141] 在示例中，服务层 1108 可以包括用于服务器的多个特征。服务器可以被配置为包括商业特定组件和相关联的服务模板的库，其可以部分或整体重用来开发客户为中心的供应。在示例中，服务可以包括钱包服务和增值服务。钱包服务可以包括管理钱包和支付标记的生命周期。增值服务可以包括提供商业服务。服务层 1108 可以包括三个单元。第一单元可以是用例，例如，可以通过使用用于实现特定用例的移动使能层 1104 的交易框架来构建服务器侧模板。在示例中，可以存在用于商业服务(例如，账单支付、电影票务等)的一个或多个不同的模板。第二单元可以包括至少一个器件或标记。器件可以包括商业特定数据对象(例如，信用卡、银行帐户、账单、优惠券等)，其可以跨各种服务被重用并且可以提供对于对象的共同理解定义以及管控对象的生命周期与交易的规则的实现。第三组件可以包括至少一个信任模型。该信任模型可以促进用于服务的不同认证方案的实现，包括对在具有它们的独立 TSM 的单独信任集群内操作的支持。

[0142] 在示例性且非限制性实施例中，个性化层 1110 (也可以称为 SDK 层)可以包括用于客户端和用于服务器的不同特征。在示例中，与 SDK 层中的客户端相关联的特征可以划分成 2 部分。第一部分可以是钱包 SDK，并且第二部分可以是微件 SDK。在示例中，钱包 SDK 可以提供对钱包服务、特许 API 以及对于构建钱包容器应用或用户体验所必需的组件的访问。在示例中，微件 SDK 可以促进生态系统合作伙伴构建它们的可以在钱包容器内操作的

微件。在示例性且非限制性实施例中，SDK 可以基本上促进客户端侧配置和用户体验。在示例中，SDK 可以通过促进对于由客户端容器所展露的参数的可定制性来促进配置。这可以包括设备特定参数、变量(像 PIN 大小、退出确认、选择模块的包含物等)。在示例中，SDK 可以通过促进跨手持机的差异化用户体验的开发来提供用户体验。

[0143] 在示例中，用于服务器的 SDK 可以包括至少一个库，其可以允许任何生态系统服务器与钱包通信、格式化对于微件的内容，以及向钱包应用展露服务。该库可以包括配置模块和接口模块。配置模块可以允许对于由服务器的移动使能层 1104 和服务层 1108 展露的参数的可定制性。可以在属性文件和资源束中配置参数。接口模块可以提供开发来促进与各种后端系统的通信的连接器，所述后端系统可以协作来履行完整的端对端服务。接口模块还可以包括由平台所展露的 web 服务以集成外围系统，像客户关怀、用户门户等。

[0144] 本文描述的多域生态系统安全个性化交易的方法和系统包括一种在平台中的方法，所述平台包括诸如服务层、使能层和个性化层中的至少一个。该方法包括应用专家引擎来分析一个或多个移动设备用户和交易环境要求。在示例中，专家引擎可以包括专家引擎层。专家引擎层可以与体验框架相交互以适应所述一个或多个用户要求。在示例中，应用专家引擎可以包括应用专家引擎来分析用户的习惯。该方法此外可以包括根据分析来配置一个或多个平台特征。在实施例中，配置所述一个或多个生态系统特征可以包括基于一个或多个交易要求来配置用户接口。在示例中，配置所述一个或多个生态系统特征可以包括配置移动设备的一个或多个显示特征。在示例中，所述一个或多个显示特征可以基于与移动设备的位置相关联的地区偏好来配置。在示例中，所述一个或多个显示特征可以基于所识别的客户端基础来配置。

[0145] 多层安全移动交易使能平台可以促进将用于安全个性化交易的多域生态系统伸缩至数千万用户，包括应用专家引擎以用于分析用户和交易环境要求以及配置生态系统特征来简化交易，诸如基于交易要求来配置用户接口。专家引擎可以与体验框架相交互以促进对用户要求的适应。

[0146] 预期的是，当提到使用设备和应用时，用户将具有变化的偏好、习惯、感知、视角、要求和风格。除了上面提及的参数之外，用户对于它们操作的结果的呈现具有单独的偏好、品味和要求。当前可获得的许多或大多数应用约束用户遵从特定生活规则(regimen)来实行交易和从系统获得与查看结果。

[0147] 本文档中描述的专家引擎层可以学习上面提及的用户习惯、偏好和其它参数，并且可以调整系统和 / 或应用以用最合适于单独用户的方式呈现其自身，包括地区偏好(“大部分法国人喜欢其是这种方式，而美国人喜欢其是那种方式”)、移动设备能力(屏幕尺寸、键盘可用或者仅在屏幕上等)以及客户端基础中的差异(低端相对于奢侈客户)。

[0148] 专家引擎层还可以管理生态系统中的变幻莫测(vagary)，诸如通信中的中断、服务器响应中的延迟、诸如耗尽电池或信号下降之类的异常操作条件等，以确保对于用户的不中断或中断管理的体验。专家引擎可以确保操作故障被管理，并且随着和当常态已经被还原时，用户能够无缝地恢复操作。

[0149] 当我们向上面提及的问题添加具有大量不同用户的维度时，所述问题进一步加剧。估计数量多达数千万，其中数千个交易同时进行。

[0150] 专家引擎可以基本上如早前所概述的那样管理所有用户，并且还对交易、使用模

式以及将帮助推理引擎的“学习”过程的其它参数实行分析。

[0151] 参照图 12，如本文所描述的专家引擎被示例性描绘用于促进针对用于安全个性化交易的多域生态系统中的多个服务提供者中的每一个修改个性化和服务层中的每一个。

[0152] 专家引擎层 / 设施 1208 可以占据提供者特定个性化层 1202，并且通过分析生态系统的各种方面 1214 来适配它们以提供经适配的个性化层 1210，所述方面包括 ：交易环境方面、用户历史方面、用户环境方面、历史交易相关数据方面、设备相关方面、生态系统状态 (例如，响应性和 / 或稳定性等)、如可以从登记的参与者和 / 或生态系统交易活动的数量所确定的生态系统大小、提供者特定方面等。类似地，提供者特定服务层数据 1204 可以由专家引擎层 / 设施 1208 进行专家适配以产生一个或多个经专家适配的服务层 1212。

[0153] 本文描述的多域生态系统安全个性化交易的方法和系统包括用于在生态系统中使能多种类型的电子钱包的平台。该平台可以包括用于开发提供者特定钱包的软件开发套件。该软件开发套件可以至少部分地通过联网服务器可执行。该平台此外可以包括通过软件开发套件可访问的多个参考钱包。所述多个参考钱包可以包括诸如用于多个提供者中的每一个的参考钱包，所述多个提供者包括器件发行者、服务提供者以及器件获取者。该平台此外可以包括运行时容器，其在客户端设备上可操作用于接收不同类型的提供者特定钱包。所述容器可以促进每种类型的提供者特定钱包的不同操作以用于执行提供者特定钱包与开发该提供者特定钱包的提供者之间的安全交易。

[0154] 本文描述的多域生态系统安全个性化交易的方法和系统包括包含程序指令的计算机可读介质。通过计算机系统的一个或多个处理器执行程序指令可以使得所述一个或多个处理器实行下面的步骤，所述步骤包括在移动设备上的钱包容器中配置一个或多个微件模块。所述步骤此外可以包括通过一个或多个微件模块来访问钱包容器中的一个或多个钱包模块。在示例中，访问一个或多个钱包模块可以包括接收从服务提供者模块到钱包容器中的第一微件模块的通信，和使能通过第一微件模块对钱包模块的经认证的访问。在示例中，可以通过网络由服务提供者在钱包容器中配置和安装一个或多个钱包模块中的每一个。在示例中，可以通过网络由用户在钱包容器中配置和安装一个或多个钱包模块中的每一个。通过计算机可读介质实行的所述步骤此外可以包括在访问一个或多个钱包模块之前对一个或多个微件模块中的每一个进行认证。

[0155] 可以在发行者、服务提供者和器件获取者的多域生态系统中使能多种类型的电子钱包的安全生态系统基础设施可以包括客户端侧功能性，其包括在移动设备上可操作的钱包容器，所述钱包容器促进多个不同钱包的隔离并且此外促进通过被配置到钱包容器中且安全地维护为分离模块的微件模块来访问每个不同钱包，其中，每个微件必须被认证以访问任何钱包。这样的生态系统基础设施还可以包括在移动设备远程操作并且彼此不同的个性化服务提供者模块，其中，服务提供者模块与钱包容器中的微件通信，并且钱包容器促进通过微件对钱包的经认证的访问。另外，生态系统基础设施元件可以包括用于通过网络由服务提供者、发行者、代理等在钱包容器中配置和安装不同钱包中的每一个的能力。用于使能多种类型的电子钱包的安全生态系统基础设施的这些和其它特征此外在图 8 和 9 中进行了描绘和描述。

[0156] 本文描述的多域生态系统安全个性化交易的方法和系统包括这样的方法，所述方法包括分析至少一个要求。在示例中，所述至少一个要求可以选自包括以下各项的集合 ：

商业策略、进入市场策略、退出策略、竞争策略、产品路线图(roadmap)、硬件发布安排和特征列表、软件发布安排、特征列表以及应用编程接口。在示例中,可以在服务提供者商业模型的上下文内分析所述至少一个要求。该方法此外可以包括至少部分基于分析来配置服务层、使能层和个性化层。在示例中,使能层可以被配置为包括通过使用与移动设备相关联的特定开发环境所构建的基础设施运行时容器软件。在示例中,服务层可以被配置为包括可以用于实现多个商业用例的公共平台服务的库。

[0157] 用于选择新业务需求和设备功能性以用于包括在使能层、服务层和个性化层的每一个中的方法可以包括分析与商业策略、进入市场策略、退出策略、竞争策略、产品路线图、硬件发布安排和特征列表以及软件发布安排、特征列表和应用编程接口相关的要求,以在服务层、使能层和个性化层中进行配置以确保在服务提供者商业模型的上下文内满足每个分析的要求。可以为诸如专家引擎和体验框架层之类的附加层类似地确定特征。

[0158] 将整体交易、服务或过程的部分分解成多层生态系统平台的方法或过程可以基于分析法、启发法和要求的组合,可选地通过包括专家引擎的工具基于人工智能概念来提供帮助。这样的方法或过程可以考虑与以下各项相关的视角和要求(目前的和预期在将来的这二者):商业策略、进入市场策略、退出策略、竞争策略、产品路线图、硬件发布安排和特征列表以及软件发布安排、特征列表和应用编程接口。

[0159] 在大多数情况中,上面提及的方面和参数可以导致服务要求在逻辑上被分割或分区成三层,即,个性化、服务和使能层。

[0160] 参照图 13,选择业务需求和设备功能性以用于包括在用于安全个性化交易的多域生态系统平台的使能层、服务层和个性化层的每一个中的实施例。基于参与者业务需求和设备功能性,生态系统特征分配引擎 1302 可以处理规则和控制,以诸如将生态系统的特征分配给使能层 1304、服务层 1308 和个性化层 1310。

[0161] 在示例中,使能层 1304 可以包括开发移动应用可能必需的基础设施或商业独立组件。服务层 1308 可以包括用于钱包生态系统和增值服务的参考服务实现的集。个性化层 1310 可以包括一组工具和 API,其可能是开发(本文档中描述的)钱包和微件服务以及构建综合生态系统所必需的。

[0162] 在示例中,使能层 1304 可以包括对于客户端和服务器不同的特征。在使能层 1304 中,客户端可以被配置为包括基础设施运行时容器软件(SW),其可以通过使用与移动设备相关联的特定开发环境来构建。在示例中,使能层 1304 可以向客户端提供对底层设备能力的公共抽象,所述底层设备能力具有公共开发者 API、UI 解译和渲染引擎、商业执行环境、微件和钱包生命周期管理、用于 OTA/ 邻近交易的模式等。在服务器侧,使能层 1304 可以向服务器提供可以在网络提供者的域中部署的基础设施软件,其可以被配置用于建立和管理安全移动交易生态系统。在示例中,使能层 1304 可以配置服务器特征,诸如钱包管理、状态交易引擎、对象生命周期管理、信道特定通信、多域安全性策略和风险管理、OTA 供应等。

[0163] 在示例中,服务层 1308 可以包括用于客户端的多个特征。例如,客户端可以被配置为包括可以用于实现多个商业用例的公共平台服务的库。公共平台服务可以在由使能层 1304 提供的商业执行环境内运行。在示例中,服务层 1308 可以包括三个基本单元。第一单元可以是用例,例如,针对各种交易服务所构建的一组客户端侧工作流和模板。第二单元可以是器件,例如,可以促进到真实世界器件的映射的对应单个钱包器件。在示例中,器件

可以提供用于可视化或显示的数据。在示例中，器件可以提供用于空中(OTA)交易的数据。在示例中，器件可以提供用于邻近交易的数据。在示例中，器件可以提供用于监管的数据。第三单元可以包括至少一个信任模型。信任模型可以促进认证方案的实现，其可以包括两因素认证(2FA)。2FA 可以由不同微件提供者所要求并且可以是与生态系统相关的。

[0164] 在示例中，服务层 1308 可以包括用于服务器的多个特征。服务器可以被配置为包括商业特定组件和相关联的服务模板的库，其可以部分或整体重用来开发客户为中心的供应。在示例中，服务可以包括钱包服务和增值服务。钱包服务可以包括管理钱包和支付标记的生命周期。增值服务可以包括提供商业服务。服务层 1308 可以包括三个单元。第一单元可以是用例，例如，可以通过使用移动使能层 1304 的用于实现特定用例的交易框架来构建服务器侧模板。在示例中，可以存在用于商业服务(例如，账单支付、电影票务等)的一个或多个不同模板。第二单元可以包括至少一个器件或标记。器件可以包括商业特定数据对象(例如，信用卡、银行帐户、账单、优惠券等)，其可以跨各种服务被重用并且可以提供对于对象的共同理解定义和管控对象的生命周期与交易的规则的实现。第三组件可以包括至少一个信任模型。该信任模型可以促进用于服务的不同认证方案的实现，包括对在具有它们的独立 TSM 的单独信任集群内操作的支持。

[0165] 在示例性且非限制性实施例中，个性化层 1310 (也可以称为 SDK 层)可以包括用于客户端和用于服务器的不同特征。在示例中，与 SDK 层中的客户端相关联的特征可以划分成 2 部分。第一部分可以是钱包 SDK，并且第二部分可以是微件 SDK。在示例中，钱包 SDK 可以提供对钱包服务、特许 API 以及构建钱包容器应用或用户体验必需的组件的访问。在示例中，微件 SDK 可以促进生态系统合作伙伴构建它们的可以在钱包容器内操作的微件。在示例性且非限制性实施例中，SDK 可以基本上促进客户端侧配置和用户体验。在示例中，SDK 可以通过促进对于由客户端容器所展露的参数的可定制性来促进配置。这可以包括设备特定参数、变量(像 PIN 大小、退出确认、选择模块的包含物等)。在示例中，SDK 可以通过促进跨手持机的差异化用户体验的开发来提供用户体验。

[0166] 在示例中，用于服务器的 SDK 可以包括至少一个库，其可以允许任何生态系统服务器与钱包通信、格式化对于微件的内容，以及向钱包应用展露服务。该库可以包括配置模块和接口模块。配置模块可以允许对于由服务器的移动使能层 1304 和服务层 1308 所展露的参数的可定制性。可以在属性文件和资源束中配置参数。接口模块可以提供开发来促进与各种后端系统的通信的连接器，所述后端系统可以协作来履行完整的端对端服务。接口模块还可以包括由平台所展露的 web 服务以集成外围系统，像客户关怀、用户门户等。

[0167] 在示例中，设备制造者可以确定 NFC 功能性将被提供作为新设备上的新通信选项，以便支持公司的商业目标。在检测到新 NFC 功能性的可用性时，业务需求和设备功能性分配引擎 1302 可以复查设备制造者的要求并且向使能层 1304 分配设备功能性。使能层 1304 于是可以使 NFC 设备功能性对于向其客户提供该设备的服务提供者可用。服务提供者于是可以通过服务层 1308 在设备上使能 NFC 功能性，并且通过使用由个性化层 1310 提供的 SDK 将 NFC 图标添加到移动微件来使终端用户能够激活 NFC 功能性作为支付方法。

[0168] 根据示例性且非限制性实施例，本公开提供可以构建用于移动支付和移动商务的安全且鲁棒的生态系统的移动交易平台(MTP)。在示例中，MTP 可以并入微件架构，其可以允许服务提供者构建它们自己的可以被下载并安装在移动钱包应用内的独立微件。在示例

中，MTP 可以向服务提供者提供综合软件开发套件（SDK）。SDK 可以允许服务提供者快速构建它们的用于所有 MTP 手持机的微件，而不必编写电话特定的代码。尽管移动钱包应用可以由生态系统操作者分布和推广，但是服务提供者可以通过微件来构建和提供它们自己的独特服务并且利用由生态系统操作者创建的分布信道。移动钱包应用可以向微件提供安全且隔离的运行时环境。

[0169] 在示例中，MTP 可以通过使用微件管理组件来管理一个或多个微件，所述微件管理组件可以管理 MTP 内的微件的用例、设计、架构、数据结构、生命周期、版本等。

[0170] MTP 可以包括移动客户端以诸如使能用于快速移动应用开发的基础设施。移动客户端可以提供基于容器的途径来诸如支持多租户应用架构。在示例中，移动客户端运行时可以是用于运行移动钱包应用的容器。钱包应用可以是支配的和主要的应用，其可以提供整体用户体验和服务器作为至微件的入口点。在示例中，微件可以包括由第三方开发的服务群组，并且在移动客户端容器内运行，独立且隔离于钱包应用。移动客户端可以操控与应用执行、安全性、可靠性等相关的复杂性，并且从而可以允许开发者更多地集中于服务的独特功能。移动客户端可以使得开发者能够不担心大部分复杂性，包括设备特定的实现。在示例中，移动客户端容器可以构建在设备特定应用编程接口（API）的顶部并且可用于流行设备环境。设备独立 API 可以确保可以通过使用 SDK 开发的应用可以跨所有支持的 MTP 手持机而运行。这些 API 可以被设计为提供开发者创建动态用户体验可能需要的所有支持。在示例中，钱包应用和微件可以通过使用 MTP SDK 来构建。在示例中，钱包应用和微件在结构上（屏幕、Java 脚本、&CSS 样式）可以是相同的，除了钱包应用可以访问相对更广的特许 API 集合之外。当钱包应用可以首先被调用时，钱包应用可以取得控制，直到用户可以选择到特定微件（卡图像、菜单选项等）中的入口点为止。作为整体 UI 设计的一部分的钱包开发者可以确定对于微件的一个或多个入口点。在选择了微件的入口点时，微件可以被发动（在相同运行时应用内）并且钱包可以被挂起。在示例中，钱包可以停止执行。一旦完成微件执行，钱包就可以恢复，并且用户从中离开钱包的屏幕可以返回到显示器。

[0171] 根据示例性且非限制性实施例，微件开发者可以取决于可以由服务提供者提供的要求来开发微件并进行测试。服务提供者可以向 MTP 服务器提交开发的微件。此外，生态系统提供者可以证明和发布微件，使得该微件对于钱包用户可以是可访问的。在示例中，钱包用户可以发现并下载微件以访问由服务提供者提供的移动服务。在示例中，微件可以已经安装在钱包用户的移动电话中。钱包用户可以发现微件的更高版本，并且相应地可以用更新的版本来更新微件的现有版本。在示例中，钱包用户可能对使用已安装的微件不感兴趣，并相应地，钱包用户可以终止并从移动设备移除（即，卸载）微件。

[0172] 在示例中，微件开发者可以被培训以使用 SDK 和理解服务来构建微件。另外，微件开发者可以具有可以由生态系统提供者所指定的指南，以诸如确保微件可以遵照指南。

[0173] 在示例中，服务提供者可以是可以发行或拥有服务产品的实体。服务可以包括用于移动支付的信用卡、电影票务服务、移动银行服务等。在示例中，服务提供者可以委托微件开发并且与生态系统提供者签订商业协定。

[0174] 在示例中，生态系统操作者可以是可以在地区 / 地理或商业域内设立移动支付和移动商务生态系统的实体。生态系统可以不一定是支付或商务生态系统。在示例中，生态系统可以是医疗保健或其它复杂商业域。

[0175] 在示例中，钱包用户可以是可能已经在他们的电话上下载并激活了移动应用(或钱包，如果关于支付域的话)的消费者。

[0176] 根据示例性且非限制性实施例，本公开提供用于微件的多个用例。在示例中，用于微件的一个或多个用例可以包括开发微件用例，其中，开发者可以按照由服务提供者提供的规范来构建微件。在示例中，用于微件的一个或多个用例可以包括测试微件用例，其中，开发者可以通过使用虚设容器和可能的沙盒钱包应用二者来测试微件。在示例中，用于微件的一个或多个用例可以包括批准微件用例，其中，服务提供者内的商业发起人可以复查和批准微件。这样的用例可以不在 MTP 服务器或客户端上。

[0177] 在示例中，用于微件的一个或多个用例可以包括提交微件用例，其中，服务提供者可以将微件连同一组测试案例一起发送给生态系统操作者。微件可以安装在生态系统提供者的测试 /UAT 环境上。在示例中，用于微件的一个或多个用例可以包括验证微件用例，其中，生态系统操作者可以复查微件测试案例、测试微件、验证微件的所有功能性，以及对着所有定义的指南来检查微件。在示例中，用于微件的一个或多个用例可以包括批准微件用例，其中，生态系统操作者可以批准微件并准许微件被包括在钱包应用内。

[0178] 在示例中，用于微件的一个或多个用例可以包括拒绝微件用例，其中，生态系统操作者可以出于某种原因拒绝或否决微件。一旦问题被解决，就可以重复目前为止的整个过程，并且可以提交新微件。在示例中，用于微件的一个或多个用例可以包括发布微件用例，其中，生态系统操作者可以发布微件并将微件加载到 MTP 服务器，使得钱包用户可以下载所发布的微件。在示例中，用于微件的一个或多个用例可以包括挂起微件用例，其中，生态系统操作者可以由于某种问题而挂起已经发布的微件直到该问题可以得到纠正为止。在示例中，用于微件的一个或多个用例可以包括发现微件用例，其中，可以通过警报或者通过对可用微件的主动搜索来向钱包用户警报微件的可用性。在示例中，用于微件的一个或多个用例可以包括下载微件用例，其中，微件可以被供应到用户的 MTP 移动客户端应用，并且可以被安装和就绪以供使用。在示例中，用于微件的一个或多个用例可以包括升级微件用例，其中，钱包用户可以将他们的微件升级为来自移动钱包应用内的更新的版本。在示例中，用于微件的一个或多个用例可以包括移除微件用例，其中，用户可以决定微件可能不再被期望。钱包用户可以从钱包应用移除微件和所有相关联的数据存储。

[0179] 在示例中，微件可以包括多个服务，其可以向钱包用户提供有意义的、域特定的功能性，所述功能性可以增大可以由钱包用户使用的移动应用的能力。微件可以称为微件束或服务舱(service capsule)。在示例中，微件束可以包括微件简档数据、微件用户接口、微件商业逻辑以及微件认可。

[0180] 在示例中，微件简档数据可以包括关于微件、服务提供者、版本等的信息。在示例中，微件用户接口可以包括可以由钱包容器用来渲染微件的用户接口的屏幕、内容和样式。在示例中，微件商业逻辑可以包含可以封装微件的商业逻辑的脚本，所述微件的商业逻辑在微件可以加载到移动设备上时可以被执行。在示例中，微件认可可以包括访问控制列表，所述访问控制列表可以包括在运行在移动应用内时生态系统操作者可能已经提供给微件的认可。

[0181] 根据示例性且非限制性实施例，本公开提供可以分组成两个大阶段的对微件的管理。第一阶段可以包括微件开发，并且第二阶段可以包括微件使用。微件开发阶段可以包

括逐渐使得微件就绪以供使用的所有步骤，并且微件使用可以包括与微件的使用及其生命周期相关的所有步骤。

[0182] 根据示例性且非限制性实施例，本公开提供根据示例的如图 14 中所图示的开发微件的方法 1400。在 1402 处，方法 1400 可以开始，并且在 1404 处，方法 1400 可以允许构建微件。在示例中，开发者可以使用由 MTP 提供的 SDK 来根据由服务提供者指定的要求构建微件。在 1408 处，方法 1400 可以测试微件。在示例中，系统可以对微件的 Java 脚本执行静态代码核实。在 1410 处，可以做出决定，以诸如完成微件开发。如果微件开发未完成，则方法 1400 可以移动到 1404，其中，开发者可以对微件进一步工作。如果微件开发完成，则方法 1400 可以移动到 1412，其中，服务提供者可以复查微件。如果在 1414 处确定服务提供者对微件不满意，则方法 1400 可以移动到 1404，其中，开发者可以并入服务提供者的反馈来构建微件。

[0183] 在服务提供者满意时，在 1418 处，可以将微件提交到生态系统操作者。在示例中，服务提供者的雇员可以通过在线提交模式而将微件提交到生态系统操作者。在 1420 处，生态系统操作者可以核实微件，以诸如识别微件中的任何错误。在 1422 处，生态系统操作者可以确定微件的通过或失败。如果微件未通过生态系统操作者的要求，则在 1424 处，生态系统操作者可以拒绝微件，并且方法 1400 可以在 1428 处结束。

[0184] 如果微件已经满足生态系统操作者的要求，则在 1430 处，可以证明微件，并且在 1432 处，可以发布微件。此外，在 1434 处，可以对微件执行发烟测试（smoke test）。在 1438 处，可以确定如微件是否已通过了发烟测试。如果微件可能未通过发烟测试，则在 1440 处，可以发起故障检修，并且方法 1400 可以移动到 1420 以重新核实微件。否则，在 1442 处，生态系统操作者可以向公众开放微件，使得公众可以下载并使用微件。在 1444 处，方法 1400 可以结束。

[0185] 根据示例性且非限制性实施例，本公开提供根据如图 15 中所图示的示例的微件使用的方法 1500。在 1502 处，方法 1500 可以开始并分支成三个不同步骤。在 1504 处，钱包用户可以浏览微件注册，以诸如识别一个或多个微件。在 1508 处，钱包用户可以选择需要的微件，并且在 1522 处，方法 1500 可以允许钱包用户下载微件。

[0186] 在示例中，在 1510 处，钱包用户可以接收对于可以在钱包用户的设备中安装的微件的更新通知。钱包用户可能期望下载微件的更新，并且相应地，在 1522 处，方法 1500 可以允许微件用户下载在 MTP 服务器处可以是可获得的微件的最新版本。在示例中，在 1512 处，服务提供者可以利用用于向终端用户提供一个或多个服务的 MTP 服务器来登记。在 1514 处，可以批准一个或多个服务，并且在 1518 处，生态系统操作者可以向钱包用户通知关于服务提供者的经批准的服务。在 1520 处，钱包用户可以读取通知，并且在 1522 处，方法 1500 可以允许钱包用户下载与服务提供者的通知的服务相关联的微件。

[0187] 在 1524 处，钱包用户可以安装微件，并且相应地，在 1528 处，可以向生态系统操作者通知关于由钱包用户进行的微件的安装或更新。在 1530 处，可以向服务提供者通知关于微件的安装。

[0188] 根据示例性且非限制性实施例，本公开提供用于向服务提供者通知关于已安装微件的移除的方法 1500。在 1532 处，钱包用户可以使用可以安装在钱包用户的设备中的微件。在 1534 处，可以做出关于钱包用户是否对已安装微件的连续使用感兴趣的确定。如果

用户可能想要在将来使用微件，则方法 1500 可以移动到 1532，其中，用户可以继续使用该微件。否则，方法 1500 可以移动到 1538，其中，可以从设备中移除该微件。在 1540 处，可以向服务器通知关于微件的移除，并且在 1542 处，可以向服务提供者通知关于微件的移除。

[0189] 根据示例性且非限制性实施例，本公开描述了其中可以创建服务提供者的用例。服务提供者也可以称为微件提供者。如上所讨论的，微件提供者可能已经与生态系统操作者签订了商业关系，以诸如创建和推广他们的微件。在示例中，在可以在系统中创建微件之前，可以要求服务提供者实体在系统中更新。例如，生态系统操作者的雇员可以登录系统管理控制台，并且可以选择微件管理的选项。此外，所述雇员可以选择添加服务提供者的链接，填充与服务提供者相关联的对应细节，并创建服务提供者记录。在示例中，系统可以向服务提供者发送关于他们的登记细节连同他们的唯一标识符的电子邮件。

[0190] 根据示例性且非限制性实施例，本公开描述了用例提交微件，其中，微件可以存储在生态系统操作者的域内的 MTP 测试台 /UAT 系统上。在示例中，可能已经通过使用 MTP SDK 工具的正确版本开发和创建了微件。此外，可以由服务提供者对微件进行内部测试和批准以供提交。在示例中，MTP 系统可以允许以两种模式提交微件 :在线和离线。

[0191] 在离线提交微件的示例中，服务提供者的雇员可以将与微件相关联的微件束发送到生态系统的经授权雇员，该经授权雇员可以从 MTP 系统管理控制台中选择微件管理。此外，生态系统的雇员可以选择提交微件和上传微件束的链接。MTP 系统可以验证微件束，并且系统可以创建微件记录并将微件的状态设置为已提交。因此，系统可以向服务提供者的雇员确认微件已经成功提交。

[0192] 在在线提交微件的示例中，服务提供者的雇员可以配置 Eclipse 插件工具以指向正确的 MTP 服务器。此外，服务提供者的雇员可以配置他们的唯一微件提供者 ID 并且选择“提交微件”选项。Eclipse 工具可以创建微件提交请求以将微件上传到 MTP 服务器，MTP 服务器可以验证微件束、创建微件记录并将微件的状态设置为已提交。MTP 服务器可以返回微件已经成功提交的确认，并且可以向生态系统的雇员发送新微件已经提交的通知。

[0193] 根据示例性且非限制性实施例，本公开描述了用例验证微件，其中，可以成功测试微件。在示例中，微件已经提交到 MTP 并且可以就绪以供测试。生态系统的雇员 (EOE) 可以从 MTP 系统控制台中选择微件管理。此外，EOE 可以选择用于微件列表的链接，该微件列表可以示出已提交状态中的所有微件。EOE 可以选择要测试的微件，并且 MTP 可以提示确认，其利用可以执行的所有所要求核实的检查表。EOE 可以选择测试选项，并且系统可以将微件的状态从已提交改变为正在测试。在示例中，系统可以对微件的 Java 脚本执行静态代码核实，并且可以确认静态代码核实已通过。系统可以显示核实结果，并且提示 EOE 对其进行复查和确认。EOE 可以确认核实结果批准，并且可以按照生态系统指南来执行所有其它测试。

[0194] 根据示例性且非限制性实施例，本公开描述了用例批准微件，其中，可以批准和发布微件。EOE 可以登录 MTP 系统控制台并在微件管理区段中选择用于批准微件的链接。系统可以示出处于正在测试状态中的所有微件，并从而，EOE 可以选择微件以批准。因此，系统可以提示确认，其利用要执行的所有所要求核实的检查表，并且 EOE 可以确认并检查所有已经执行的测试。此外，EOE 可以输入任何评价 / 备注，并且选择批准选项。之后，系统可以将微件的状态改变为已批准，并且可以确认微件已被批准。

[0195] 根据示例性且非限制性实施例，本公开描述了用例发布微件，其中，微件可以被发

布并且就绪以被下载到用户的钱包应用。EOE 可以登录 MTP 系统控制台，并且在微件管理区段中选择用于发布微件的链接。系统可以示出处于已批准状态中的所有微件，并且 EOE 可以选择要发布的微件。EOE 可以输入任何评价 / 备注并且选择发布选项。系统可以提示以供确认，并且 EOE 可以确认可以发布微件。因此，系统可以将微件的状态改变为已发布，并且生成用于微件的签名。在示例中，系统可以创建要安全存储在 SE 中的钱包上的微件束的散列，并且用 HMAC 来对微件束进行签名并具有与钱包共享的密钥。在示例中，系统可以用 RSA 密钥对微件束进行签名，并且在钱包内包括证书。之后，系统可以确认微件已被发布。

[0196] 根据示例性且非限制性实施例，本公开描述了用例挂起微件，其中，微件状态可以转变为已挂起，并且其不再可以被主动下载。在示例中，EOE 可以登录 MTP 系统管理控制台，并且在微件管理区段中选择用于挂起微件的链接。系统可以示出处于已发布状态中的所有微件，并且 EOE 可以选择微件以将该微件挂起。EOE 可以输入任何评价 / 备注并选择挂起发布。此外，系统可以提示以供确认，并且 EOE 可以确认微件的挂起。因此，系统可以将微件的状态改变为已挂起，并且可以确认微件已经挂起。在示例中，系统可以向服务提供者生成微件已经挂起的电子邮件通知。

[0197] 图 16 图示了根据示例的用于安装微件的交易流程图 1600。如图 16 中所示，客户端 1602 可以包括钱包 1604 和钱包运行时 1608。服务器 1610 的一个或多个组件可以包括移动服务器 1612、个性化模块 1614 以及企业服务总线（ESB）1618。

[0198] 在示例中，在 1620 处，服务器 1608 可以使用 ESB 1618 向个性化模块 1614 发行软卡、服务提供者 ID 和产品 ID。在 1622 处，可以向移动服务器 1612 发送请求，诸如得到微件简档、诸如提供者 ID、产品 ID 之类的标识、平台、状态和 SDK 版本。在 1624 处，对象被推向钱包运行时 1608。在 1628 处，个性化模块 1614 可以向移动服务器 1612 传送请求以创建微件实例，其可以包括钱包 ID、提供者 ID 和小程序 ID。在 1630 处，可以确定对象是平台对象。在 1632 处，可以操控和处理平台对象。在 1634 处，钱包运行时 1608 可以传送消费对象 JS 的请求。在 1638 处，钱包 1604 可以以用于下载对象 JS 的对象做出响应。

[0199] 在 1640 处，钱包运行时 1608 可以向移动服务器 1612 传送下载微件的请求。在 1642 处，移动服务器 1612 可以准备微件响应并将微件状态标记为 SENT_FOR_DOWNLOAD（发送 _ 用于 _ 下载）。在 1644 处，移动服务器 1612 可以向钱包运行时 1608 发送微件二进制。在 1648 处，钱包运行时 1608 可以安装微件，并且相应地，在 1650 处，钱包运行时 1608 可以向钱包 1604 传达微件安装完成。在 1652 处，钱包 1604 可以向钱包运行时 1608 发送应答，钱包运行时 1608 进而在 1654 处可以向移动服务器 1612 应答。在 1658 处，移动服务器 1612 可以将微件状态标记为已安装。在 1660 处，移动服务器 1612 可以此外向个性化模块 1614 传送应答，个性化模块 1614 进而在 1662 处可以执行商业流。在 1664 处，个性化模块 1614 可以经由 ESB 1618 通知卡发行者。另外，在 1668 处，钱包运行时 1608 可以反过来对钱包 1604 的应答进行应答。在 1670 处，钱包 1604 可以发送确认，并且在 1672 处，钱包运行时 1608 可以清理可靠性队列。

[0200] 根据示例性且非限制性实施例，本公开描述了器件和钱包的关联。在示例中，平台（即，MTP）可以得到诸如用以在平台中推送器件（例如，软卡）之类的指令。可以使用和 / 或发布各种 API，以诸如将发行者系统与 MTP 进行集成。MTP 可以基于标识参数来从主记录得到服务产品。在示例中，标识参数可以包括提供者 ID、服务产品的类型、网络提供者类型以

及互相关身份(例如,移动号码)等。平台可以通过使用互相关身份来取出钱包帐户信息。平台可以创建可以与钱包帐户相关联的器件(即,服务帐户参考)记录。平台可以将器件记录状态标记为已注册,其可以由发行者系统来更新。此外,可以通过使用诸如WAITING_FOR_ACTIVATION(等待_激活)、激活(ACTIVE)、关闭(CLOSED)等之类的状态来控制其生命周期。平台可以通过使用推送通知(PUSH)或背负信道(piggyback channel)触发钱包来递送器件。

[0201] 图17图示了描绘MTP系统中微件的生命周期的示例状态图。在1702处,可以利用MTP来提交微件。在示例中,EOE可以利用MTP来提交微件。在另一示例中,服务提供者的雇员可以使用在线提交特征来利用MTP提交微件。如图17中所示,在1704处,当状态为已提交(SUBMITTED)时,可以提交微件任意次数。这可以允许开发者在测试可以开始之前更新微件。在示例中,当微件处于不同于已提交状态的状态中时,不可以重新提交微件。

[0202] 在1708处,微件状态可以更新为正在测试(TESTING)。在该状态期间,系统可以对微件执行多个测试(例如,静态代码核实)。在该状态处,取决于测试结果,系统可以将微件的状态改变为已批准(APPROVED)或已拒绝(REJECTED)。在1710处,微件的已拒绝状态可以不允许开发者更新已拒绝的微件。换言之,一旦系统可以拒绝微件,就可以提交(具有相同微件ID&版本号的)新微件束,并且可以重复该过程。结果,可以创建微件表中的新记录。此外,如果具有相同微件ID和版本号的微件已经存在于除已提交或已拒绝之外的任何状态中,则微件不可以进入已提交状态。在1712处,微件状态可以更新为已批准,描绘了微件可能已经清理了正在测试的阶段。在1714处,已批准的微件可以被发布,并且对于下载可以是可获得的。此外,可以可能的是,已发布的微件可能需要更新或者系统可能需要挂起已发布的微件。因此,在1718处,微件可以被挂起,描绘了微件对于下载不可以是可获得的。在示例中,只有当所有之前的版本号处于已批准、已发布、已拒绝或已挂起状态中,具有数学上较大版本号的微件才可以是已提交的。在示例中,系统可以用日志记录微件的状态转变,连同时间戳、动作者的唯一标识以及已经由动作者输入的评价的审查跟踪(audit trail)。

[0203] 系统可以被配置为包括用于服务帐户管理的各种API。例如,系统可以被配置为包括创建相关API,其中,所述API可以创建服务产品记录、服务提供者记录、服务帐户参考记录等。在示例中,系统可以被配置为包括更新相关API,其中,所述API可以更新服务帐户参考记录、服务帐户参考状态(以诸如控制服务帐户参考生命周期)等。在示例中,系统可以被配置为包括列表相关API,其中,所述API可以对系统中可用的所有已注册的服务产品、系统中可用的所有已注册的服务提供者等进行列表。在示例中,系统可以被配置为包括检索相关API,其中,所述API可以检索基于钱包id过滤准则的服务帐户参考记录、基于微件ID过滤准则的服务帐户参考记录、基于产品ID过滤准则的服务帐户参考记录、基于状态过滤准则的服务帐户参考记录、基于服务帐户参考号的服务帐户参考记录、基于服务产品提供者id过滤准则的服务帐户参考记录、基于微件实例id过滤准则的服务帐户参考记录、基于服务产品id过滤准则的服务产品、通过提供者id过滤准则的服务产品提供者记录,等等。在示例中,状态值可以是已注册(REGISTERED)、WAITING_FOR_ACTIVATION(等待_激活)以及激活(ACTIVE)。所述状态值可以被扩展以用于进行定制,或者用以引入更细粒度水平的服务帐户参考状态。在示例中,服务帐户参考号可以是由发行者系统共享的唯一标识符,其

可以充当钱包平台和发行者系统内的互相关 id。在示例中，系统可以被配置为包括更新相关 API，其中，所述 API 可以更新服务产品记录、服务产品提供者记录等。

[0204] 根据示例性且非限制性实施例，本公开描述了各种数据映射的细节。在示例中，系统可以包括用于与服务提供者相关联的数据的各种数据字段。ID 可以是服务产品提供者的唯一系统标识符，并且可以自动生成。名称字段可以用于显示服务提供者的名称，并且当创建服务提供者的入口时，可以输入与服务提供者相关联的数据。类型字段可以标识服务提供者的服务类型，并且种类字段可以定义可以由服务提供者提供的服务产品的种类。电子邮件字段可以包括服务提供者的电子邮件地址。此外，系统可以包括一个或多个属性，其可以用键值对（例如，联系人号码）的形式存储并且可以用于向服务提供者引入任何定制属性。

[0205] 在示例中，系统可以包括用于与服务产品相关联的数据的各种数据字段。ID 可以是服务产品的唯一系统标识符，并且 ID 可以由系统自动生成。如上文所讨论的，提供者 ID 可以是服务产品提供者的唯一系统标识符。产品 ID 可以是产品的唯一名称（例如，银行金卡、高端信用卡等），并且系统可以包括与服务产品相关联的应用（例如，银行业务）的应用 ID。类型字段可以标识服务产品的类型（例如，信用卡），并且种类字段可以标识服务产品的种类（例如，支付）。名称字段可以标识服务产品的显示名称，并且描述可以标识服务产品的描述。此外，系统可以包括以键值对的形式存储的属性，并且属性可以用于向服务产品实体引入任何定制属性。

[0206] 在示例中，系统可以包括用于与器件（服务帐户参考）相关联的数据的各种数据字段。ID 可以标识器件的唯一系统标识符，并且系统可以自动生成 ID。钱包 ID 可以标识钱包的唯一标识符。发行者 ID 可以标识发行者的唯一标识符，并且产品 ID 可以标识服务产品的唯一标识符。微件实例 ID 可以标识微件实例的唯一标识符，并且帐户参考号可以标识器件在发行者端的唯一标识符。状态可以标识器件的状态（例如，已创建、已激活、已挂起等）。类似地，系统可以包括钱包平台状态。供应时间戳可以标识器件的供应日期（时间戳），并且下载时间戳可以标识器件的下载日期（时间戳）。激活时间戳可以标识器件的激活日期（时间戳），并且中断时间戳可以标识器件的中断日期（时间戳）。最后活动时间戳可以标识器件的最后活动日期（时间戳），并且最后状态改变时间戳可以标识最后状态改变的时间戳。此外，系统可以包括昵称字段（诸如用以标识用户提供的器件名称），其它字段（诸如 OTA TX 数据、邻近 TX 数据等）。

[0207] 有效期（expiry date）可以标识器件的有效期，并且到期日可以标识器件的到期日。系统可以包括额外属性，其可以用于容纳任何附加信息。这些属性可以不由平台使用，并且可以主要用于微件。此外，徽标图像可以标识可以用于在钱包应用上显示软卡的图像。该图像按照发行者的品牌化定义可以是联合品牌的。卡图像可以标识可以用于在钱包应用上显示软卡的图像。该图像按照发行者的品牌化定义可以是联合品牌。产品类型可以标识在设置期间在钱包平台处配置的服务产品的唯一标识符。AID（例如，HEXSTRING（十六进制字符串））可以标识小程序 ID，其可以是在个性化期间安装在安全元件上的小程序的唯一标识符。OTA 认可（OTA Permission）可以标识可以由发行者管理来控制 OTA 服务认可的空中服务认可标志，并且邻近认可（Proximity Permission）字段可以标识可以由发行者管理来控制邻近交易或服务的邻近服务认可标志。软卡品牌 ID 可以包括品牌化信息 ID，并且网

络操作者 ID 可以标识移动网络操作者。

[0208] 在示例中，系统可以包括用于与微件提供者相关联的数据的各种数据字段。ID 可以标识微件提供者的唯一系统标识符，并且系统可以自动生成 ID。名称文件可以标识微件提供者的显示名称，并且描述可以标识微件提供者的描述。电子邮件可以标识微件提供者的电子邮件地址，并且状态可以指示微件提供者的状态。创建日期时间戳可以标识微件提供者的创建日期(时间戳)，并且批准日期时间戳可以标识当微件提供者已被批准时的批准日期(时间戳)。

[0209] 在示例中，系统可以包括用于与微件提供者相关联的数据的各种数据字段。ID 可以标识微件简档的唯一系统标识符，并且系统可以自动生成 ID。应用 ID 可以标识微件应用 ID，并且应用名称字段可以标识微件可以与其相关联的应用名称。提供者 ID 可以标识微件提供者的唯一标识符，并且提供者名称可以标识微件提供者的名称。最小运行时版本可以标识最小所需钱包平台版本，并且版本可以标识微件的版本。状态字段可以标识微件简档的状态(例如，已创建、已发布等)。

[0210] 在示例中，系统可以包括用于与微件实例相关联的数据的各种数据字段。ID 可以标识微件实例的唯一系统标识符，并且系统可以自动生成 ID。微件束 ID 可以将区外(foreign)键标识为 wsc_widget_bundle。微件简档 ID 可以标识微件的唯一标识符，并且系统可以定义区外键 :微件简档的 ID。钱包 ID 可以标识钱包的唯一标识符。供应时间戳可以标识微件供应日期(以时间戳)，并且安装时间戳可以标识微件安装日期(以时间戳)。激活时间戳是可选字段，并且移除时间戳可以标识微件实例移除日期。状态字段可以标识微件实例的状态。

[0211] 在示例中，系统可以包括用于与微件认可相关联的数据的各种数据字段。ID 可以标识微件实例的唯一系统标识符，并且系统可以自动生成 ID。微件简档 ID 可以标识对微件简档 ID 的参考，并且微件提供者 ID 可以标识对微件提供者 ID 的参考。

[0212] 在示例中，系统可以包括用于与微件束相关联的数据的各种数据字段。ID 可以标识微件实例的唯一系统标识符，并且系统可以自动生成该 ID。微件束可以标识微件文件(.wgt)的二进制数据，并且设备软件类可以标识与微件束相关联的设备软件类。微件简档 ID 可以标识对微件简档 ID 的参考，并且堤岸(bund)散列可以标识微件束文件的散列。

[0213] 在一些示例性且非限制性实施例中，本文描述的多域生态系统安全个性化交易的方法和系统包括可以支持安装和管理多个微件的移动钱包(也称为钱包)容器。这些微件中的每一个可以由不同发行者和服务提供者开发和拥有。钱包可以被配置为履行各种安全性要求。这里描述的示例性关键安全性环境中的一些包括微件完整性、微件访问控制和微件隔离。这些如以下所述 :

在示例中，可以通过使用像钱包服务器这样的受保护的供应服务器来将微件安全地供应到钱包运行时。可以将微件安装在手持机上。当在手持机上时，对微件的任何篡改可以被自动检测，并且可以采取必要行动。

[0214] 在示例中，对发行者安全性域小程序的访问可以专为发行者微件所独有。系统可以防止任何发行者微件访问另一发行者的安全性域。系统可以防止微件访问可能未被明确准予给定微件的系统动作。系统可以防止任何发行者微件访问不同于所明确准予给定微件的 URL/ 域。

[0215] 在示例中，任何发行者微件可以具有对发行者的工作流代码的排他访问。系统可以被配置以使得任何发行者微件不可以访问任何其它发行者的工作流代码。系统可以被配置以使得任何发行者微件可以具有对发行者的用户接口(UI)代码的排他访问。系统可以被配置以使得任何发行者微件不可以访问任何其它发行者的 UI 代码。系统可以被配置以使得任何发行者微件不能够访问任何其它微件的持久性数据。系统可以被配置以使得任何发行者微件不可以访问任何其它微件的静态资源(图像、文件等)。

[0216] 在一些示例性且非限制性实施例中，本文描述的多域生态系统安全个性化交易的方法和系统包括：访问控制机制(ACM)可以被配置为对着给定“策略”来管理“资源”。在一些示例性且非限制性实施例中，本文描述的多域生态系统安全个性化交易的方法和系统包括：被管理的少数示例性资源可以包括微件工作流代码、微件数据、微件 UI 代码(屏幕)、安全元件、服务器通信。在示例中，为了确定“策略”，本公开可以具有访问控制器，其可以将给定微件的“微件描述符”用作策略文件。微件描述符可以是简单的键值对结构，其由微件开发者创建(以描述必要资源)，并且由移动交易平台(MTP)服务器基于钱包操作者的处理来签名。该访问控制机制可以牵涉在对着给定策略来管理资源的整个过程中的微件描述符生命周期和移动客户端访问控制。过程的这两个方面可以在下面进行描述。

[0217] 图 18 图示了从开发到执行的微件生命周期。在一些示例性且非限制性实施例中，微件生命周期(由 1800 表示)可以具有在发行者与 JVL 的子系统之间具有职责划分的阶段。所述阶段可以包括微件开发、微件核实和批准、微件递送、微件安装以及微件执行。微件生命周期的微件开发阶段可以包括工具套件 1802。在示例中，工具套件 1802 可以基于 Microsoft visio®。移动交易平台工具(MTP)1804 可以支持工具套件 1802 以促进微件开发。微件开发之后可以是微件包生成阶段。在微件包生成期间，可以由 MTP 工具 1804 形成微件束 1808。微件束 1808 可以包括微件描述符、屏幕支持文件和主题支持文件(这些可以在下文进行更详细的解释)。可以由 MTP 工具 1804 将微件束 1808 发送到移动交易平台(MTP)1810 以用于静态核实、策略批准或策略供应。在示例中，可以通过使用可信服务管理器(TSM)1814 来提供策略供应和微件供应简档。在示例中，如作为稍后的章节所解释的，可以通过使用访问控制列表(ACL)来提供微件供应。MTP 1810 可以被配置用于在核实之后发布微件束 1808。在示例中，微件供应和发布可以被同步并在 MTP 1810 和 TSM 1816 上同时发生。MTP 1810 和 TSM 1816 可以在核实和发布之后将微件束 1808 发送到移动客户端 1818。移动客户端 1818 可以被配置为包括微件束 1808、用于平台服务的块、用于数据服务的块、用于通信服务的块以及用于安全元件的块。移动客户端 1818 可以定义用于微件束 1808 的运行时覆盖区。稍后可以通过图 21 来解释移动客户端。

[0218] 在一些示例性且非限制性实施例中，微件生命周期中的第一阶段可以是微件开发。在示例中，移动交易平台(MTP)可以支持基于 Microsoft Visio 的微件开发工具套件来使微件开发自动化。微件开发可以在发行者域中发生。微件可以由多个微件工件(artifact)构成。在示例中，微件可以由 screen.xml 文件、content.xml 文件、theme.xml 文件、多个脚本文件、多个图像以及微件描述符构成。在示例中，微件开发工具套件可以生成像屏幕、内容等的微件工件，连同微件描述符。微件描述符可以捕获开发者对于像动作 / URL/ 小程序等的不同资源的微件访问要求(或请求)。

[0219] 在一些示例性且非限制性实施例中，微件生命周期中的下一阶段可以是微件核实

和批准。在示例中，微件批准步骤可以涉及微件的验证。在示例中，微件核实可以在钱包平台(例如，JVL 钱包平台 - 证明工具)处发生。微件批准步骤此外可以涉及微件的批准或供应。本发明提供了用于微件的静态分析的一组工具，其可以帮助整个批准过程。在示例中，微件批准的技术过程可以包括各种步骤。第一步骤可以是创建对于给定微件的静态分析报告，其指示任何警告 / 复制 / 冲突。在示例中，这些指示可以涉及应用协议数据单元冲突(APDU)、统一资源定位符(URL)冲突、存在太多外部资源、使用了受限 / 受控的应用编程接口(API)。下一步骤可以是捕获对微件的手动批准，其记录批准条件和所接受 / 忽略的警告。下一步骤可以是创建微件供应简档。微件供应简档可以包括 ACL 和微件二进制的散列。下一步骤可以是创建微件包以通过使用钱包信道来下载。

[0220] 在一些示例性且非限制性实施例中，微件生命周期中的下一阶段可以是微件递送和安装。在示例中，微件一旦其被证明就可以就绪以供递送并且可以对于从钱包平台下载是可获得的。在一些示例性实施例中，可以由于各种事件而触发微件下载和安装。所述事件可以包括由发行者的新卡供应事件。

[0221] 在一些示例性且非限制性实施例中，移动客户端的微件管理者可以在安装微件的时候执行各种步骤。第一步骤可以是在已经下载了微件束之后核实微件束的完整性。在示例中，可以通过使用安全元件服务来进行核实。在微件束的完整性得到核实的情况下，可以将该微件束添加到移动客户端。在微件束的完整性未得到核实的情况下，将示出错误消息，并且该微件束将不被添加到移动客户端。

[0222] 在一些示例性且非限制性实施例中，微件生命周期中的下一阶段可以是微件执行。可以由钱包管理器为微件的每次运行确定微件的完整性和访问简档。微件执行和完整性检查可以包括加载时间完整性检查和运行时访问控制。加载时间完整性检查可以检查微件对于执行后成功安装是否可以就绪。每次加载微件以供执行时可以核实散列。可以要求该验证以确保微件或其数据在钱包或微件的空闲时间期间未被篡改。运行时访问控制可以检查微件是否一旦它通过在其加载时候的散列核实步骤就可以开始其功能。微件访问控制器可以确保微件尝试访问特定资源可以是按照恰当的授权，即隔离规则。

[0223] 移动客户端访问控制(也称为访问控制)—当微件描述符生命周期描述“建立策略”的过程时，移动客户端访问控制机制确保“策略的施行”。本章节描述了访问控制机制的高层级概观。

[0224] 图 19 图示了访问控制机制的高层级视图。如图 19 中所图示，可以通过经由访问控制器 1902 来控制资源访问。访问控制器 1902 可以基于微件描述符来确定访问权限。访问控制器 1902 可以使例如屏幕或动作或小程序等的特定资源有益于微件。在示例性且非限制性实施例中，访问控制器 1902 可以包括一组设计考虑事项。在示例中，访问控制器 1902 可能需要用于数据存储的系统 1904。数据存储装置可以被配置为存储微件二进制。在示例中，微件简档供应和 ACL 信息可以安全地存储在安全元件 1908 中。安全元件 1908 可以与访问控制器 1902 进行通信。微件简档供应和 ACL 信息可以包括钱包安全性域 1910、钱包小程序 1912、发行者安全性域 1914 以及发行者小程序 1918。访问控制器 1902 可以与微件策略文件 1920 进行通信。

[0225] 访问控制器 1902 可以与近场通信(NFC)芯片管理器 1922、资源管理器 1924 以及通信管理器 1928 进行通信。NFC 芯片管理器 1922、资源管理器 1924 和通信管理器 1928 可

以与脚本执行平台 1930 进行通信，脚本执行平台 1930 可以与屏幕管理器 1932 进行通信。如图 19 中所图示，微件可能没有对任何原生资源的 API 层级访问，并且所有资源使用可能必须由访问控制器 1902 来促进。例如，用于数据存储的系统 1904 可能必须与访问控制器 1902 通信，其可以促进 NFC 芯片管理器 1922、资源管理器 1924 和通信管理器 1928 对于数据存储系统 1904 的访问。在示例中，可以允许屏幕管理器 1922 在特定时间点仅执行一个微件。在示例性且非限制性实施例中，对于整个系统，可以仅存在一个访问控制器 1902。

[0226] 在示例中，可以要求微件在每次其可以被加载或重载时使其散列被核实。在示例中，仅一个微件可以被允许执行。

[0227] 图 20 图示了用于相对于访问控制器而加载和执行微件的方法 2000。

[0228] 方法 2000 可以在步骤 2002 处要求移动客户端加载钱包主页。移动客户端可以与用户在相接口。在钱包主页可以加载到移动客户端上之后，用户可以从钱包主页选择微件。移动客户端可以被配置为在方法 2000 的步骤 2004 处打开由用户选择的微件。移动客户端可以与应用数据库进行通信。移动客户端可以被配置用于在步骤 2008 处加载从应用数据库所选的微件的微件束。移动客户端可以与安全元件进行通信。在步骤 2010 处，安全元件可以通过执行散列核实来核实微件束的完整性。如上文所提及的，每当可以加载微件时，可以核实散列。在示例中，微件束可以包含在钱包小程序中用于具有核实。可以在微件供应的时候存储散列。散列核实对于方法 2000 可以是至关重要的，并且在散列核实失败的情况下，可以停止微件执行。在散列核实成功的情况下，移动客户端可以从安全元件中检索用于所选微件的微件描述符。在方法 2000 的步骤 2012 处，可以将微件描述符存储在 MTP 的主存储器中。在步骤 2014 处，在可以检索到微件描述符之后，屏幕管理器可以将下一屏幕加载到移动客户端上。在步骤 2018 处，移动客户端可以请求平台服务执行。在示例中，可以通过指定各种参数来请求他服务。平台服务对于所选微件可以是排他的(通过图 21 来解释平台服务)。在步骤 2020 处，移动客户端可以核实微件描述符，以使得确保微件可以具有对访问所请求的平台服务的特许。在方法 2000 的步骤 2022 处，在所选微件可以具有对访问所选平台服务的特许的情况下，移动客户端可以执行所请求的平台服务。方法 2000 可以要求步骤 2020 的微件描述符核实以用于进一步执行微件。例如，如果可能发现微件描述符不可以访问所选平台服务，则移动客户端可以终止该方法。在方法 2000 的步骤 2024 处，移动客户端可以向用户传送微件的执行，并且移动客户端可以提示用户在执行之后退出微件。在步骤 2028 处，移动客户端可以在执行后销毁微件。如果他用户可能想要执行另一微件，则移动客户端可以被配置为加载另一钱包主页。

[0229] 图 21 图示了基于微件访问控制策略的微件运行时覆盖区 2100。每个微件可以基于其访问控制策略来在运行时创建逻辑覆盖区。访问控制策略可以在微件供应简档中定义。每个逻辑覆盖区可以包含 UI 工作流资源 2102、动作 / 服务资源 2104、数据资源 2108、在线资源或通信资源 2110 以及安全元件资源 2112。如由图 21 所图示的，可以在本文中描述微件运行时覆盖区 2100 的边界。

[0230] 每个微件可以具有其自己的屏幕 2114 和内容文件。微件 A 不可以使用来自微件 B 的内容文件，并且反之亦然。微件运行时覆盖区 2100 将会使用可以对于所有微件是公共的原生平台 2118。每个微件可以具有可以仅由特定微件使用的一组特许服务 2120。例如，微件 A 可以仅使用来自特许服务 2120 的微件 A 特许服务 2120A，并且微件 B 可以仅使用来自

特许服务 2120 的微件 B 特许服务 2120B。微件运行时覆盖区 2100 具有平台服务块 2122，其可以是移动客户端的一部分，并且因此可以由微件 A 或微件 B 或任何其它微件中的任何一个使用。任何微件都可以访问公共数据 2124。公共数据默认存储在微件运行时覆盖区 2100 中。在示例中，微件可以定义其自己的数据存储，如果被授权这样做的话。例如，微件 A 可以定义数据源 2128A，并且微件 B 可以定义数据存储 2128B。

[0231] 移动客户端可以具有公共 URL 平台 2130。默认情况下，例如微件 A 或微件 B 的任何微件都可以访问来自公共 URL 平台 2130 的数据。在示例中，微件 A 和微件 B 中的每一个可以定义它们自己的通信端点 2132A 和 2132B。通信端点可以是域名。在该示例中，微件 A 可以使用通信端点 2132A，并且微件 B 可以使用通信端点 2132B。

[0232] 微件中的任何一个可以没有对任何 SE 小程序的默认访问。可以在微件描述符中明确定义针对给定微件可以允许的每个小程序访问。基于由服务器的签名所准予的认可，一个或多个微件可以访问相同 SE 小程序。

[0233] 详述了微件隔离要求的设计。在一些示例性且非限制性实施例中，本文描述的多域生态系统安全个性化交易的方法和系统包括用于访问控制的策略文件的整个生命周期。各种阶段描述了重要文件 / 数据结构及它们的变换 / 处理，其最终引起移动客户端容器处的访问控制实现。所述阶段可以包括开发阶段、供应阶段、微件发动阶段以及微件执行阶段。可以在本文中描述这些阶段。

[0234] 开发阶段可以包括微件描述符文件。微件描述符文件可以由开发者(例如，发行者)提供。微件描述符文件可以从开发者视角描述一组资源要求。在示例中，在开发阶段期间，微件连同微件描述符文件一起可以被提交用于微件批准过程。在示例性实施例中，可以在微件批准过程期间证明资源要求是合理的。在微件已经被批准之后，可以由供应管理机构(例如，JVL)对微件签名。

[0235] 供应阶段可以包括微件供应简档。微件供应简档可以由供应管理机构创建，作为批准过程的一部分。微件供应简档可以是具有一些附加信息的微件描述符文件的衍生。在示例中，所述附加信息可以包括用户 ID、微件散列等。可以通过使用可信服务管理器(TSM)来提供微件供应简档。可以通过使用钱包认证 / 伴随小程序来将微件供应简档存储在安全元件(SE)内部。SE 可以向钱包提供服务以接受用户 ID 和微件散列，并返回核实状态以及 ACL。

[0236] 微件发动阶段可以包括至少一个微件访问记录。在示例中，微件发动阶段可以包括多于一个微件访问记录。可以通过微件管理器在钱包的主存储器内部创建和维持微件访问记录。微件访问记录可以是简单的记录结构，其中可以存储资源认可。当发动微件时，微件管理器可以联系 SE 以读取 ACL，并且微件管理器在主存储器中装载微件访问记录。

[0237] 微件执行阶段可以包括微件访问上下文对象。微件访问上下文对象可以是运行时对象，其可以提供用以检查对于微件执行的认可的简单方法。微件访问记录可以用于填充微件访问上下文对象。微件访问上下文对象可以在内部维护认可的简单映射(键值对)。微件访问上下文对象可以由所有访问知晓(access aware)资源管理器用于检查给定微件的认可。资源管理器可以在内部决定是否需要对着策略来检查给定资源。

[0238] 在一些示例性且非限制性实施例中，微件描述符可以是中心位置，其中可以定义对于多个资源的所有访问权限。在示例中，所述多个资源可以包括动作、屏幕、至少一个 URL

和 SE/ 小程序。在示例性实施例中，微件的构造可能要求有益于不同的资源，像动作、小程序、屏幕等。在示例性但非限制性实施例中，动作可以包括移动客户端平台可以提供 4 种不同类型的动作，即，系统动作、核心 / 基础设施动作、钱包动作以及任何其它发行者特定的定制动作。在示例中，动作可以是移动客户端平台的一部分，并由此跨微件被共享。在示例中，任何微件可以访问核心和钱包动作。在示例中，在默认情况下，可以不允许微件访问任何系统动作。微件可能需要在描述符中指定其所需的所有系统动作。在示例中，微件可以指定任何微件特定的动作，其可以是意欲仅特定地用于该微件的动作。在示例性但非限制性实施例中，每个微件可以被要求具有单独的屏幕和内容文件集合，其不可以重用来自任何其它微件的屏幕和内容资源。在示例性但非限制性实施例中，可以准许微件做出至多个 URL 的连接。在示例性但非限制性实施例中，微件可以与多个小程序相接口。

[0239] 微件批准过程 - 图 22 图示了微件批准过程。

[0240] 如图 22 中所图示的微件批准过程可以涉及用户 2202、微件证明管理器 2204、微件贮存库管理器 2208、静态核实实用程序 2210 以及二进制包生成器 2212。用户 2202 可以从对于用户可用的各种微件束中选择微件束 2214。在示例中，微件束 2214 可以包含 application.xml 文件、content.xml 文件、theme.xml 文件以及微件描述符(这些文件可以与上文描述的相同)。微件束 2214 可以由用户 2202 发送到微件证明管理器 2204。微件证明管理器 2204 也可以是微件批准管理器。微件证明管理器 2204 可以是驱动批准过程的主控制器类。微件证明管理器 2204 可以将微件束 2214 发送到微件贮存库管理器 2208，微件贮存库管理器 2208 可以管理关于所有已证明和未证明微件的细节与恰当时间戳细节。微件贮存库管理器 2208 可以具有对微件贮存库 2218 的访问。微件贮存库 2218 可以包含所有已证明和未证明微件与恰当时间戳细节。该信息可以帮助审查跟踪和静态核实。微件证明管理器 2204 可以将微件束 2214 发送到静态核实实用程序 2210。静态核实实用程序 2210 可以对着静态规则来验证微件束 2214。例如，如果客户端可能已经宣告 client_Check_CardStatus (客户端 _ 检查 _ 卡状态) 作为其定制动作之一，则可以不允许该客户端自身以外的任何其它发行者使用该特定动作。静态核实实用程序 2210 可以针对对可能由其它微件拥有的资产的任何意外 / 不正确引用而检查微件贮存库 2218 中的微件束 2214。静态核实实用程序 2210 可以自动化访问核实过程的一部分，但是仍然可以要求手动核实。微件证明管理器 2204 可以从静态核实实用程序 2210 接收对着静态规则所核实的微件束 2214。微件证明管理器 2204 可以将微件束 2214 发送到二进制包生成器 2212。二进制包生成器 2212 可以生成与微件束 2214 对应的最终二进制包。二进制包可以就绪以供空中下载和递送。二进制包生成器 2212 可以将二进制微件文件 2220 发送到用户 2202。

[0241] 微件环境方面 - 这里描述对于钱包软件的少数示例性但非限制性环境假设和依赖性。环境假设和依赖性可以包括安全通信、安全主存储器、安全应用环境、可靠 API、SE 访问管理、受保护数据库以及可靠匆促注册 (rush registry)。在示例中，钱包软件可以假设底层平台和 API 提供用于安全通信的可靠和安全的 HTTP 与 HTTPS 连接。原生平台可以管理用于安全套接层 (SSL) 核实的根证书存储。在示例中，钱包软件可以假设由原生 OS 分配给钱包软件的主存储器可以受保护以免受其它应用 (app) 影响，并且可以安全地持有对于过渡范围的敏感数据元素。钱包软件可以假设钱包二进制可以由应用环境安全地存储和维护在电话上以提供安全的应用环境。通过使用可信内容证书的环境来核实 app 签名。该环

境还可以在下载到电话上的所有app上强加严格的签名和访问控制规则。钱包软件可以假设所有Java或类似API可以受底层环境保护，并且对于处理用于可靠API的敏感数据可以是可信的。而且，所述API可以是受环境访问控制的。钱包软件可以假设对安全元件的访问可以由环境基础设施(例如，JSR 177)管理，并且对各种安全元件服务的访问可以是特许的。可以通过基于app签名的环境来实现特许管理。钱包软件可以假设该环境可以将对持久性数据的访问限制于仅拥有者应用，并且所述应用不可以访问或修改彼此的数据。钱包软件可以假设该环境可以提供可靠的应用推送注册，其中对于钱包app的事件被可靠地传递给钱包(而没有讹误或入侵)。

[0242] 本文描述的多域生态系统安全个性化交易的方法和系统包括包含程序指令的计算机可读介质。通过计算机系统的一个或多个处理器执行程序指令可以使得所述一个或多个处理器实行以下步骤，所述步骤包括经由移动设备上的钱包容器来配置一个或多个微件模块以供执行。所述一个或多个微件模块可以包括服务提供者的服务群组。在示例中，钱包容器可以与用于提供安全个性化交易的多层平台的使能层协作地操作来为一个或多个微件模块提供对服务提供者的工作流的排他访问。在示例中，钱包容器可以解译在移动设备上操作的运行时环境中的微件模块以提供排他访问。在示例中，服务提供者可以是微件发行者。在示例中，可以向由微件发行者发行的微件模块中的一个或多个提供对微件发行者的工作流的排他访问。在示例中，钱包容器可以与用于提供安全个性化交易的多层平台的使能层协作地操作来提供排他访问。在示例中，可以向由微件发行者发行的微件模块中的一个或多个提供对微件发行者的用户接口的排他访问。在示例中，钱包容器可以与用于提供安全个性化交易的多层平台的使能层协作地操作来提供排他访问。在示例中，可以防止由微件发行者发行的微件模块中的一个或多个访问另一微件发行者的工作流。在示例中，钱包容器可以与用于提供安全个性化交易的多层平台的使能层协作地操作来防止访问另一微件发行者的工作流。在示例中，可以防止由微件发行者发行的微件模块中的一个或多个访问另一微件发行者的用户接口。在示例中，钱包容器可以与用于提供安全个性化交易的多层平台的使能层协作地操作来防止访问另一微件发行者的工作流。

[0243] 用于安全个性化交易的多域生态系统可以包括各种组件，包括服务器、客户端及其它类型的组件。客户端组件可以包括可以在本文或别处描述的各种软件程序、应用、小程序、数据集、接口、菜单等。一般而言，客户端功能性可以从在客户端设备上驻留和/或可执行的各种钱包、微件、小程序、容器等得到，所述客户端设备诸如移动电话、智能电话、平板、通用电子交易设施、膝上型计算机、笔记本计算机、上网本、自助服务机、器具、公用设备、汽车、飞机以及任何其它电子或机电设备。

[0244] 一般地，本文描述的方法和系统包括在本文及别处称为移动客户端、容器等的客户端软件，其充当用于快速移动应用开发的使能基础设施。其提供基于容器的途径来支持多租户应用架构。移动客户端容器提供用于钱包和微件的电话特定运行时环境。该容器操控与应用执行、交易安全性、可靠性等相关的复杂性，包括设备特定实现。移动客户端容器可以基于API，诸如设备特定的API。容器运行时环境促进从微件描述符生成原生应用屏幕。这促进确保触摸屏按钮看起来像相应设备上的原生ANDROID、BLACKBERRY、IPHONE或其它设备按钮。钱包应用可以与微件类似，因为它也可以由移动客户端运行时功能性来解译和执行。然而，如本文所描述的，钱包将具有对应用的特许控制以维护高度安全和分离的环

境来使用和访问钱包信息，所述钱包信息除其它外尤其可以包括用户帐户、个人和交易相关内容。壁（wall）还可以被定义为提供域特定（例如，金融、零售、医疗保健、政府等）的功能性。

[0245] 参照图 23，图示了微件架构的示例性且非限制性实施例，其可以促进在用于安全个性化交易的多域生态系统中的客户端设备上的钱包容器内隔离不同的服务提供者微件。客户端容器 2302 可以操作为客户端运行时组件，以用于促进在用于安全个性化交易的多域生态系统中的客户端设备（诸如移动电话、计算机、PDA、膝上型计算机等）上的移动应用的执行。如所图示的，一个或多个钱包 2304 可以包括应用和数据，其可以提供整体用户数字钱包体验并且可以通过由用于一个或多个微件 2306 的容器 2302 管理的一个或多个安全入口点来访问。一个或多个微件 2306 可以包括由第三方服务提供者开发和 / 或发行的服务群组。根据示例性实施例，微件 2306 可以以独立方式与一个或多个钱包 2304 隔离地运行在客户端容器 2302 的安全环境内。当根据示例性实施例实现为分立的可执行代码单元时，微件可以以与其它微件、钱包和小程序的操作分离和不同的方式来执行和操作，或者相反地，可以以与一个或多个其它钱包、微件和 / 或小程序协作的方式来操作，以执行安全交易。

[0246] 如所图示的，客户端容器 2302 可以与包括一个或多个设备特定的 API 的 API 层次 2308 相接口。API 层次 2308 可以支持各种和 / 或流行的设备环境。电话独立 API 层次 2310 可以确保应用跨所有支持的 MTP 手持机而运行，所述应用包括但不限于钱包、微件和小程序，诸如通过使用移动交易平台（MTP）软件开发套件（SDK）开发的那些。根据示例性实施例，钱包 2304 和微件 2306 二者可以在结构上（例如，屏幕、Java 脚本 &CSS 样式）几乎相同，除了与微件 2306 所做的相比，钱包 2304 可以访问 API 层次 2308 内的更宽的特许 API 集合。在一些情况下，钱包 2304 可以能够有鲁棒的用户配置和个性化，在与一般的微件和特别是服务提供者得到的微件相比时尤其如此。结果，在一些情况下，钱包可以提供有对在客户端设备的功能性上施加控制的 API 的增加的访问。

[0247] 根据示例性且非限制性实施例，当被调用时，钱包 2304 可以取得应用控制，直到（例如由用户、钱包 2304、容器 2302、外部服务等）选择了微件 2306 入口点为止。微件 2306 的用户选择可以经由可以通过钱包 2304 接口而对用户可用的卡图像、菜单选项等来进行。响应于这样的选择，可以在包括钱包 2304 的运行时安全环境（例如，由容器 2302 提供的环境）内发动所选微件，并且可以挂起钱包 2304 的执行。

[0248] 一旦被发动，微件 2306 就可以操作来促进例如微件 2306 正在其上执行的移动客户端设备的钱包 2304 与服务提供者之间的安全交易，所述服务提供者诸如发行微件 2306 的服务提供者。在这样的情况下，由服务提供者发行的微件 2306 可以被准予排他访问发行服务提供者的工作流、移动设备上的资源（诸如钱包资源、安全元件资源）以及客户端设备的用户接口。同样地，由服务提供者发行的微件 2306 可以被拒绝访问其它服务提供者的工作流、钱包元件等以及诸如客户端设备的用户接口之类的某些移动设备资源。

[0249] 作为分离地可执行的代码单元，微件 2306 可以经由从移动操作系统可获得的执行分离功能和 / 或通过包括由容器提供的运行时环境的移动客户端的各方面而有效地相互隔离。另外，可以由服务提供者（例如，器件的发行者等）开发微件 2306，所述服务提供者可以配置微件以合并提供者特定的安全性元件，其可以使得其它服务提供者开发的微件难

以访问其资源、工作流等。可以包括这样的微件、容器、钱包、服务提供者软件(没有限制地包括本文稍后描述的钱包服务中心的各种方面)等的移动交易平台可以在所有这样的方面中施加强且不同的安全性,使得例如由服务提供者 A 开发的微件不可以访问服务提供者 B 的(例如,基于服务器的)资源。这样的强安全性出于各种原因可以是重要的,包括保持域特定的信息自用户可以通过用户的移动设备与之交互的其它域(例如,金融域、健康域、法律域、商业域、个人域、身份域等)是分离且机密的。

[0250] 本文描述的多域生态系统安全个性化交易的方法和系统包括包含程序指令的计算机可读介质。通过计算机系统的一个或多个处理器执行程序指令可以使得所述一个或多个处理器实行以下步骤,所述步骤可以包括在移动设备上的钱包容器中配置一个或多个微件模块。钱包容器可以包括一个或多个钱包模块。由计算机可读介质实行的步骤此外可以包括限制由一个或多个微件模块对包括一个或多个钱包模块、移动设备、一个或多个客户端应用的元素集的访问。所述步骤此外可以包括提供对发行者特定的安全性域和微件特定的发行者安全性域中的至少一个的一个或多个微件模块的访问。

[0251] 本文描述的多域生态系统安全个性化交易的方法和系统包括一种方法,所述方法包括在移动设备上提供容器运行时环境。所述方法此外可以包括访问移动设备驻留微件模块以确定与微件相关联的访问约束,在示例中,访问约束可以关于电子钱包访问、设备资源访问、设备驻留生态系统应用、由移动设备可访问的网络资源、发行者特定的安全性域以及微件特定的安全性域。所述方法此外可以包括通过容器来操作微件模块中的至少一个,以施加与微件模块相关联的访问约束。所述方法此外可以包括促进由微件模块对与该微件模块相关联的发行者特定的安全性域和微件特定的安全性域中的至少一个的访问。在示例中,施加微件模块访问约束可以包括防止对移动设备上的电子钱包资源的访问。在示例中,施加微件模块访问约束可以包括防止由第一微件对与第二微件相关联的钱包的访问。在示例中,施加微件模块访问约束可以包括基于钱包的状态来防止由微件对移动设备上的钱包的访问。钱包的状态可以包括活动状态和非活动状态,在活动状态期间,可以允许由微件的访问,并且在非活动状态期间,可以防止由微件的访问。在示例中,促进对安全性域的访问可以包括允许对移动设备上的可以与微件模块共享安全性域的数字钱包的访问。在示例中,促进对安全性域的访问可以包括允许对发行者的可以与微件模块共享安全性域的网络资源的访问。

[0252] 本文在上面描述了移动交易平台的用于隔离和限制域特定的资源之中的访问的各种方面,包括客户端特定的方面。除了这些重要的安全性和资源访问限制功能之外,关于对钱包、设备资源、应用、网络资源等的访问的应用相关限制也可以通过本文描述的方法和系统来促进,所述方法和系统包括开发、部署、执行以及交易方法和系统。

[0253] 根据示例性实施例,容器可以实现一个或多个触发来控制或以其它方式限制对一个或多个钱包、设备、客户端应用和网络资源的访问。关于访问的限制可以基于定位(例如,场所、位置、网络类型、提供者可访问性、钱包状态、花费限制、信用余额、信用评分等)。这里描述了关于对资源的微件访问的这些和其它限制的示例。可以由商家以个性化方式来管理钱包,使得例如当某人进入一场所时,可以不向那个人给出使用使能在该场所不接受的支付形式的微件的选项。在另一非限制性示例中,当用户在美国以外旅行时,可以通过一个触发或多个触发来增强对各种微件的访问。根据又其它的示例,基于信用卡余额 / 限制的触

发可以此外增强一个或多个微件的操作，包括但不限于由一个或多个微件对客户端应用和网络资源的访问。在示例性实施例中，如果钱包不是活动的(诸如可能在容器和一个钱包或多个钱包驻留在其上的客户端设备被关断时或者当用户已经去活钱包(例如为了钱包资源的增强的安全性)时发生)，则可以限制相关联的微件访问客户端设备以及通常由钱包控制或通过钱包访问的网络资源、钱包内容等。

[0254] 本文描述的多域生态系统安全个性化交易的方法和系统包括一种方法，所述方法包括至少部分通过联网服务器来配置微件以通过对发行者特定的服务群组的访问。所述方法此外可以包括在操作于客户端设备上的安全钱包容器中接收来自联网服务器的部署所配置的微件的请求。所述方法此外可以包括确定用于在客户端设备上操作微件的微件隔离要求。所述方法此外可以包括基于隔离要求而在移动设备上通过安全钱包容器来安装所配置的微件。

[0255] 本文描述的多域生态系统安全个性化交易的方法和系统包括包含程序指令的计算机可读介质。通过计算机系统的一个或多个处理器执行程序指令可以使得所述一个或多个处理器实行以下步骤，所述步骤可以包括在客户端设备上部署第一发行者特定的微件以促进基于第一信任模型的安全个性化交易。所述步骤此外可以包括在客户端设备上部署第二发行者特定的微件以促进基于第二信任模型的安全个性化交易。在示例中，第一发行者特定的微件和第二发行者特定的微件可以部署在钱包容器中。在示例中，第一信任模型和第二信任模型中的至少一个可以是委派(delegated)信任模型、直接信任模型以及代理(brokered)信任模型之一。在示例中，基于第一信任模型的安全个性化交易和基于第二信任模型的安全个性化交易中的至少一个可以基于触发而被促进。在示例中，触发可以对应于基于第一信任模型的安全个性化交易和基于第二信任模型的安全个性化交易中的至少一个的货币值。

[0256] 根据示例性且非限制性实施例，在微件 2306 已经诸如由可以是服务提供者的发行者开发之后，微件 2306 的操作可以被核实和批准并且可以存储在服务器上以供随后下载到一个客户端设备或者在非用户特定的微件 2306 的情况下下载到多于一个客户端设备。如所注意到的，微件 2306 可以取决于它们所指向的功能任务而在性质上变化。作为结果，各种类型的微件 2306 包括但不限于钱包特定的微件、用户 - 钱包特定的微件、供应 - 特定的微件、市场特定的微件、审查特定的微件、多用途微件等等。在成功安装到客户端设备上之后，微件 2306 就绪以供执行。

[0257] 在示例性实施例中，一旦微件 2306 被安装，以应答形式的确认就可以从下载微件 2306 的根源发送到服务器。由服务器对应答的接收可以导致服务器存储与微件 2306 的成功下载相关的信息。例如，服务器可以存储指示下载和安装的微件 2306 的版本号的信息。这样的信息可以例如用于确定存在比之前已下载的更新近版本的微件 2306 供下载。在一些示例性实施例中，客户端设备处微件 2306 的成功下载可以导致显示给客户端设备的用户的确认可视记号(indicia)，诸如显示的文本消息。同样，用户可以提供有用于接受下载的微件、例如包括执行微件的非绑定性试验的装置。这样的微件安装的结果可以包括将新的图标放置在客户端设备的移动交易平台用户接口中。新的图标可以暂时区别于其它图标以促进指示其比其它图标更新。可替换地，新的微件可以不具有对于用户而言的视觉上可检测的差异。

[0258] 所部署或安装的微件完整性可以由本文所描述的容器(例如,移动客户端容器)来提供。用于确保微件完整性的一种技术是确保微件和/或其数据是不变的。根据示例性实施例,为了确保在微件 2306 的下载和其执行之间的时间期间微件 2306 或其数据都没有被篡改,可以采用散列。可以从包括微件 2306 的应用代码来计算散列,并且当微件被安装时,散列被核实、被访问以供执行、在客户端移动交易平台资源的审查期间被访问、被加载以供执行(例如,在 cClient 容器运行时环境中)等。在当微件被加载以供执行时核实散列的示例中,微件 2306 一旦其通过在其加载时候的散列核实就可以开始其功能。

[0259] 移动客户端容器运行时环境还可以促进微件的强隔离,使得不同的微件不能影响其它微件资源、变更这样的资源或微件应用代码,以及类似可能影响微件完整性的。

[0260] 如上面所注意到的,微件 2306 可以提供一个或多个安全入口点,经由所述安全入口点,钱包应用可以与微件 2306 的发行者交互。在示例性实施例中,微件 2306 的发行者还可以是其服务通过由用户使用发行的微件 2306 而在整体上或部分上被使能的服务提供者。在示例性且非限制性实施例中,一个或多个微件 2306 可以一致地行动来以不同的信任度来执行安全个性化交易。特别地,可以部署不同的微件 2306 来促进根据不同信任模型的安全个性化交易,所述不同信任模型除了其它以外尤其包括用于交易执行的直接、间接、代理以及第三方安全信任模型。

[0261] 根据示例性且非限制性实施例,多于一个信任模型的使用可以反映其中部署了客户端设备和相关联的微件 2306 的环境中的差异。如本文在别处所注意到的,当配置生态系统的元件时,钱包和微件发行者可以选择一个或多个信任模型。例如,也许包括服务提供者的发行者可以部署第一微件 2306 以在空中执行安全银行业务交易,其中,交易具有第一信任级别。发行者此外可以例如部署第二微件以在销售点(POS)自助服务机(kiosk)处执行安全银行业务交易,其中,由于客户端设备邻近于自助服务机和/或访问设备的用户的独立核实,所以交易具有更高的第二信任级别。这些微件中的任一个或二者可以部署到单个客户端设备,并且可以由钱包 2304 诸如基于如所述的交易性质、可用 OTA 网络的安全性等来执行对用于执行的微件的选择。

[0262] 根据其它示例性且非限制性实施例,微件部署可以取决于变量。例如,发行者可以配置多个微件2306,其各自以金融交易量相关的信任模型来操作。例如,要求相对低信任级别的微件可以被部署用于涉及少于五十美元的金融交易,而要求相对高信任级别的微件可以被部署用于涉及多于五十美元的金融交易。

[0263] 本文描述的多域生态系统安全个性化交易的方法和系统包括一种方法,所述方法包括在移动设备处用多域安全钱包容器接收警报,即在移动设备上部署的现有微件的更新版本是可获得的。所述方法此外可以包括确认所述更新是对在移动设备上部署的现有微件的更新。所述方法此外可以包括配置钱包容器以接收经更新的微件。所述方法此外可以包括用钱包容器接收经更新的微件。所述方法此外可以包括处置现有微件以防止该微件的进一步使用。所述方法此外可以包括在移动设备上存储经更新的微件。

[0264] 本文描述的多域生态系统安全个性化交易的方法和系统包括一种方法,所述方法包括以操作于移动设备上的多域钱包容器来向服务器查询更新的微件的存在。所述方法此外可以包括从服务器接收指示更新的微件的可用性的信息。所述方法此外可以包括请求更新的微件。所述方法此外可以包括在移动设备处接收更新的微件。所述方法此外可以包括

处置更新的微件所取代的现有微件以防止进一步使用现有微件。所述方法可以附加地包括将更新的微件存储在移动设备上。在示例中，可以以预定时间间隔执行对服务器的查询。在示例中，可以响应于微件在客户端设备上的执行而执行对服务器的查询。所述方法此外可以包括向服务器发送存储更新的微件的确认。

[0265] 本文描述的多域生态系统安全个性化交易的方法和系统包括一种方法，所述方法包括从移动设备接收对于存在更新的微件的查询。所述方法此外可以包括向移动设备传输指示更新的微件的可用性的信息。所述方法此外可以包括从移动设备接收对于更新的微件的请求。所述方法此外可以包括向移动设备传输更新的微件。在示例中，可以以预定时间间隔接收查询。在实施例中，可以响应于微件在客户端设备上的执行而接收查询。所述方法此外可以包括接收对存储更新的微件的确认。

[0266] 根据示例性且非限制性实施例，可以出于各种目的中的任何一个来更新在客户端容器 2302 的安全环境内被存储并且能够运行的微件 2306（例如，周期地、诸如每周地以提供当前定价或供应（offer）、增加花费限制，响应于用户请求以改善安全性、添加对不同信任模型的访问等等）。在一个示例性实施例中，在复查和发布微件 2306 之后，微件提供者可以创建和发出警报，指示一个或多个微件 2306 的新或更新版本的可用性。

[0267] 在一个示例性实施例中，可以仅向之前已经成功下载了（或以其它方式已安装了）要更新的微件 2306 的版本的那些客户端设备发出警报。在这样的情况下，消息可以被发送到客户端设备以告知用户更新的微件 2306 是可获得以供下载的，并且向用户提供下载更新的微件 2306 的选择。

[0268] 在另一示例性实施例中，诸如可以存储在钱包服务器上的配置文件可以包含用户可定制信息，其定义了用户期望用于微件更新发生的方式。例如，用户可以定制相关联的配置文件以指示每当之前下载的微件 2306 的新版本变得可用时，微件的较新版本要被下载。相反地，客户端设备可以周期地，诸如在启动时或者在参与商业交易之前，查询钱包服务器以下载之前安装的微件 2306 的较新版本。在其它实施例中，可以由或者针对微件发行者、相关钱包发行者等定制配置文件。

[0269] 根据其它示例性实施例，可以通过微件发行者的工作流向微件 2306 自身告知更新是可获得的，并且基于更新的微件的各方面（例如，安全性更新），所安装的微件可以采取行动（例如，禁用其自身，向用户通知其不再是当前的，更新其自身的部分而非用新微件替代它等）。可以通过钱包的操作来实现另一更新机制。例如，可以更新钱包，并且作为结果，可以指示微件过时，触发各种微件行动（例如，通知用户、自动更新、请求更新等等）。

[0270] 在多域生态系统中提供用于安全个性化交易的多层移动交易平台的方法和系统可以包括一种方法，所述方法在移动设备上配置至少一个移动交易平台特定的应用编程接口以用于促进由移动设备上执行的钱包容器对移动设备安全资源的访问。所述方法此外可以包括将多个不同钱包容器布置在移动设备的存储器中，其中，每个钱包容器经由所述至少一个应用编程接口来与移动设备安全资源相接口。所述方法此外可以包括所述至少一个编程应用接口经由三层次移动交易平台的安全性准则来施加钱包容器隔离。这三个层次可以包括个性化层次、服务层次和使能层次。

[0271] 本文描述的多域生态系统安全个性化交易的方法和系统包括在客户端设备上可操作的钱包容器。钱包容器可以包括用于促进微件、钱包、安全性和对安全元件的访问的管

理的运行时组件。钱包容器此外可以包括基于 API 的接口，用于访问由微件提供的商业特定服务和用于访问客户端设备特定资源。钱包容器可以促进多个不同钱包以及不同微件的用于访问所述多个不同钱包中至少一个的操作。

[0272] 本文描述的多域生态系统安全个性化交易的方法和系统包括包含程序指令的计算机可读介质。通过计算机系统的一个或多个处理器执行程序指令使得所述一个或多个处理器实行以下步骤，所述步骤可以包括在移动设备上的钱包容器中配置一个或多个微件模块。钱包容器可以包括一个或多个钱包模块。所述步骤此外可以包括促进由微件中至少一个的发行者对钱包模块中至少一个的访问。

[0273] 钱包容器 2302 可以形成移动交易平台(MTP)使能基础设施的客户端驻留部分。如这样，容器可以为诸如移动手持机等的客户端设备提供跨平台服务执行环境。

[0274] 如上文所注意到的，本文描述的方法和系统一般包括在本文和别处称为移动客户端、容器、运行时环境等的客户端软件，其充当用于快速移动应用开发的使能基础设施。其提供基于容器的途径以支持多租户应用架构。移动客户端容器为钱包和微件提供电话特定的运行时环境。容器操控与应用执行、交易安全性、可靠性等相关的复杂性，包括设备特定实现。移动客户端容器可以基于诸如设备特定的 API 之类的 API。

[0275] 如所图示出的，一个或多个钱包 2304 可以包括应用和数据，所述应用和数据可以提供整体用户数字钱包体验，并且可以通过由用于一个或多个微件 2306 的容器 2302 管理的一个或多个安全入口点来访问。一个或多个微件 2306 可以包括由第三方服务提供者开发和 / 或发行的服务群组。根据示例性实施例，微件 2306 可以以与一个或多个钱包 2304 隔离的独立方式运行在客户端容器 2302 的安全环境内。当根据示例性实施例而实现为分立的可执行代码单元时，微件可以以与其它微件、钱包和小程序的操作分离和不同的方式执行与操作，或者相反地，可以以与一个或多个其它钱包、微件和 / 或小程序协作的方式操作以执行安全交易。

[0276] 参照图 24，其描绘了客户端设备容器(例如，移动客户端)的高层级架构视图，呈现了容器的详细示例性方面。容器 2402 可以促进用户接口渲染和导航 ；工作流执行 ；对开发者 API 的访问 ；以及此外可以包括应用容器基础设施。容器 2402 可以促进具有微件生命周期管理器能力 2432、访问控制器 2434 等的微件管理 2430，同时集成了微件和钱包上下文。容器此外可以包括用户接口框架 2412，其可以包括 UI 引擎 2414、屏幕和布局 2418、样式 2420、UI 元素 2422 等。如图 24 中所描绘的容器此外可以包括工作流框架 2424，其可以利用脚本引擎 2428、服务管理器，并且可以包括 java 脚本功能、可编写脚本的对象等。其它容器能力可以包括数据库功能性 2440，诸如寻找 / 搜索、插入 / 更新、删除等。可以通过用于发送(例如，sms、HTTP/S)、编码、解码、附接、拆卸等的容器来促进通信 2438。还可以通过容器来支持近场通信功能性 2448。容器此外可以通过与安全元件或如可以由客户端提供的其它安全框架的交互来促进安全性，所述交互包括小程序安装、移除、安全信道设置、APDU 交换等。与容器相关联地提供的另外的安全性特征可以包括密码学 2442(例如，AES、3DES、RSA 等)、签名、核实(例如，HMAC)、散列计算和核实(例如，SHA/1)、密钥生成(例如，PKCS#5)、随机数功能(例如，PRNG)等。容器还可以与诸如电话功能之类的客户端资源交互以促进进行呼叫、到达和 / 或管理联系人，与日历交互以促进基于事件的功能、基于位置的服务等。

[0277] 本文描述的多域生态系统安全个性化交易的方法和系统包括管理多个交易协议

的方法，其包括在客户端设备的存储器中布置钱包模块。所述方法此外可以包括在存储器中布置钱包伴随小程序。所述方法此外可以包括执行与在生态系统上进行的安全个性化交易相关联的钱包伴随小程序，在所述生态系统中，客户端设备被部署来管理对与客户端设备相关联的安全元件的所选内容的访问。在示例中，钱包伴随小程序可以使用所访问的安全元件内容来促进对访问钱包模块的认证以执行交易。

[0278] 本文描述的多域生态系统安全个性化交易的方法和系统包括包含程序指令的计算机可读介质。通过计算机系统的一个或多个处理器执行程序指令使得所述一个或多个处理器实行以下步骤，所述步骤可以包括在移动设备上的钱包容器中配置一个或多个微件模块。钱包容器可以包括一个或多个钱包模块。所述步骤此外可以包括配置至少一个小程序来与至少一个钱包模块相结合地执行，以促进多个交易协议的管理。

[0279] 根据示例性且非限制性实施例，安全元件(SE)可以用作客户端处的安全数据存储设备，诸如用于包括标准智能电话的移动设备。在示例性且非限制性实施例中，SE 可以以一个或多个选项来安装，包括 UICC、可移除微 SD，嵌入在电话硬件中，或者安装为外接背板或者根据一些其它示例性接口方法来安装。

[0280] 根据示例性且非限制性实施例，如果由控制和 / 或发行实体和生态系统的剩余部分认可，则诸如钱包之类的设备应用可以与 SE 进行通信。通信的模式和方法可以取决于来自上面提及的选项之中的 SE 的类型而变化。

[0281] 在一些示例性实施例中，取决于用于控制 SE 的安全性模型，可以认可钱包在 SE 中安装、加载和管理小程序。还可以或者仅可以允许通过包括发送和接收应用协议数据单元(APDU)的小程序通信的标准方法来与预先安装的小程序进行通信。

[0282] 如名称所暗示的，钱包伴随小程序可以具有与数字钱包(例如，客户端设备上的钱包应用)一起工作的工作主功能，并且可以由客户端设备用于实行多种任务，包括但不限于：存储由一个或多个钱包所需的重要和 / 或关键数据，或者用于通过使用钱包而进行交易的任何操作，或者控制被设计用以通过可共享接口和 / 或通过为该目的所设计和实现的 APDU 的通信来促进这样的控制的其它小程序的运转。

[0283] 根据各种示例性且非限制性实施例，与钱包伴随小程序协调的钱包应用可以管理、影响和 / 或控制如此被使能的其它小程序，以用于管理它们的交易协议。

[0284] 参照图 24A，其描绘了所呈现的移动设备内容、移动设备应用和安全元件特征的详细视图。移动设备应用可以包括如本文所描述的 MTP 应用(例如，容器、钱包、微件等)。安全元件内容可以包括安全性域 2450、钱包伴随小程序 2452 以及钱包伴随域 2454。如图 24A 中所描绘的，安全元件可以被配置在各种物理设备中，包括 UICC、微 SD、封套、外接附件、嵌入式元件等。

[0285] 钱包伴随域 2454 可以在本文描述的伴随小程序上扩展以包含用于伴随小程序和特征的整个安全性域。钱包伴随域 2454 或安全性域可以促进加载、安装、管理和删除安全元件中多个小程序，作为与安全性域相关联的群组。每个安全性域可以具有唯一的安全性密钥，其可以由移动交易平台用于在安全元件中的小程序群组上执行操作。钱包伴随域 2454 可以通过以下方式来管理：适配当前技术的安全元件密钥安全性特征以使能用于移动交易平台的分离的安全性密钥，以管理安全元件中的钱包伴随域 2454 的内容。

[0286] 图 24B 描绘了与钱包伴随小程序相关联的各种特征和控制以及数据流。安全元件

中的钱包伴随小程序可以提供诸如以下的特征：安全存储、认证、ID 管理、激活控制、存储 / 取得功能、加密功能、密钥存储和生成功能、PIN 控制等。钱包伴随小程序可以提供钱包接口功能，其可以促进管理由其它小程序、外部移动交易平台参与者等对钱包的访问。钱包伴随小程序可以包括小程序接口，用于访问安全元件中的小程序。钱包伴随小程序可以促进与可信安全性管理器（TSM）服务器、移动交易平台（MTP）服务器及用于安全个性化交易的多域生态系统中的其它参与者的安全交互。

[0287] 本文描述的多域生态系统安全个性化交易的方法和系统包括一种方法，所述方法包括在客户端设备的存储器中布置钱包模块。所述方法此外可以包括在存储器的安全元件中布置钱包伴随小程序。所述方法此外可以包括执行钱包伴随小程序以促进对一个或多个附加钱包小程序的访问。在示例中，可以通过至移动钱包功能的可共享接口来促进所述访问。在示例中，钱包伴随小程序可以使能优惠券发放、安全性和支付中的至少两个。

[0288] 本文描述的多域生态系统安全个性化交易的方法和系统包括一种方法，所述方法包括在客户端设备的存储器中布置容器。所述方法此外可以包括在存储器的安全元件中布置容器伴随小程序。所述方法此外可以包括执行容器伴随小程序以促进对一个或多个附加容器伴随小程序的访问。在示例中，可以通过至经由容器可访问的移动微件功能及移动钱包功能的可共享接口来促进访问。

[0289] 本文描述的多域生态系统安全个性化交易的方法和系统包括在客户端设备上可操作的平台，其包括可以配置至少一个标记化微件的运行时组件。在示例中，所述至少一个标记化微件可以促进用于安全个性化交易的多域生态系统中的至少一个交易。

[0290] 本文描述的多域生态系统安全个性化交易的方法和系统包括在客户端设备上可操作的平台，其包括可以配置用于安全个性化交易的多域生态系统中的至少一个服务提供者微件的运行时组件。

[0291] 本文描述的多域生态系统安全个性化交易的方法和系统包括在客户端设备上可操作的平台，其包括可以配置用于安全个性化交易的多域生态系统中的至少一个客户端容器的运行时组件。

[0292] 本文描述的多域生态系统安全个性化交易的方法和系统包括在客户端设备上可操作的平台，其包括可以保护钱包中多个帐户的帐户信息的运行时组件。

[0293] 本文描述的多域生态系统安全个性化交易的方法和系统包括在客户端设备上可操作的平台，其包括可以提供由多个服务提供者对移动钱包的安全访问的运行时组件。

[0294] 用于将供应与在外部服务提供和用于安全个性化交易的多域生态系统中的移动钱包之间的交易进行集成的移动钱包可以包括在移动设备上可操作的多域数字钱包，其中，数字钱包包括来自至少两个不同服务提供者的帐户。还可以包括运行时容器，用于执行尝试访问数字钱包的微件。另外，可以包括在运行时容器中可操作的多个移动供应管理微件，其通过与外部服务提供者通信而与不同的移动供应域相接口。最后，还可以包括通过运行时容器、多域数字钱包以及多个移动供应管理微件的至少一部分的协作操作所提供的移动交易能力。

[0295] 根据示例性且非限制性实施例，本公开提供可以作为一套小程序的一部分的优惠券小程序，该套小程序被开发来促进更丰富且更便利的零售购物体验，该体验的目标可以是使能移动钱包应用与服务提供者之间的供应相关的流，诸如相关联于销售点（PoS）终端。

[0296] 优惠券小程序可以支持以单敲(single tap)以及多敲(multi-tap)用户体验形式的近场通信(NFC)交易。优惠券小程序的一个或多个版本可以支持用于以下各项的特征:将多个优惠券附到单个交易、在交易中呈现所附的优惠券(例如,向 PoS 终端)、传送赎回(redemption)状态、管理供应、将供应匹配到钱包、(例如,通过服务提供者微件)从服务提供者寻求供应等。为了促进易于开发和集成,优惠券小程序可以包括至少两个接口 API。一个 API 可以用于将一个或多个供应(例如,优惠券)与数字钱包应用(例如,如本文所描述的移动钱包客户端)相接口,并且另一个 API 可以用于将优惠券与服务提供者相接口,诸如通过 PoS 终端。服务提供者 API 可以促进由 PoS 终端对优惠券小程序的访问以用于查询优惠券、更新优惠券赎回状态等。

[0297] 优惠券小程序的一个或多个版本可以使能直接前向流,通过该直接前向流,用户可以诸如通过 PoS 终端、通常通过具有安全元件(SE)的移动电话而在与服务提供者的交易中呈现从一个或多个源下载的优惠券。

[0298] 图 25 图示了供应或优惠券相关交易流程图。参与交易的实体可以包括用户 2502、钱包客户端 2504、优惠券小程序 2508 以及 PoS 终端 2510。在示例中,用户可以接触各种优惠券服务、浏览优惠券并决定要使用哪些优惠券用于特定购买。用户可以经由可以与钱包客户端 2504 相关联的移动钱包应用来执行这些操作。在 2512 处,钱包客户端 2504 可以向用户 2502 呈现存储的优惠券,并且在 2514 处,用户 2502 可以选择可以附着的优惠券。

[0299] 此外,可以发起阶段 1 (本文也称为预交易设置阶段)。如图 25 中所示,在阶段 25 期间,钱包客户端 2504 可以将所选优惠券附到小程序,并因而,钱包客户端 2504 可以指示交易可以开始。

[0300] 在阶段 1 之后,可以发起阶段 2(本文中也称为 PoS 交易阶段)。在阶段 2 期间,可以向 PoS 终端 2510 呈现认证标识(例如,来自安全元件的用户身份和 / 或帐户数据)以发起 PoS 交易阶段。PoS 终端 2510 然后可以向优惠券小程序 2508 查询附着的优惠券,并且优惠券小程序 2508 作为响应可以发送回与附着的优惠券相关的数据。这基于优惠券的数目可以进行若干查询。一旦 PoS 终端接收了所有优惠券,其就可以处理这些优惠券。这样的处理可以包括验证优惠券、检查适用性、遵照其它商业规则以确定优惠券的适用性等。在该阶段的处理结束时,可以标识赎回的优惠券。可以将这些优惠券的状态传送到优惠券小程序 2508。

[0301] 在阶段 2 之后,可以发起阶段 3(本文也称为后交易查询 & 清理)。在该阶段期间,如果需要的话,钱包客户端 2504 可以查询赎回状态并重置所有优惠券。优惠券的重置可以标定阶段 3 的结束,和将状态复位为阶段 1 的开始。

[0302] 图 26 图示了根据示例的优惠券小程序在交易流期间的状态机图。状态 2602 可以指示 READY FOR UPLOADING COUPONS (就绪以供上传优惠券)状态,其指示来自优惠券小程序的所有优惠券可能已经被重置,并且钱包客户端现在可以上传为该交易选择的优惠券。在示例中,如图 26 中所示,如果命令 GET APPLET STATUS(取得小程序状态)导致错误,则小程序可以保持在状态 2602 中。在接收到有效的 ENABLE TRANSACTION (使能交易)命令时,优惠券小程序就可以转变到下一状态 2604。

[0303] 2604 状态可以包括 READY FOR COUPON REDEMPTION (就绪以供优惠券赎回)状态,其指示钱包客户端可能已经将所有优惠券附着到优惠券小程序的存储器,并且 PoS 终端可

以继续进行交易的阶段 2。在示例中，如图 26 中所示，如果命令 READ COUPONS(读取优惠券)导致错误，则优惠券小程序可以保持在状态 2604。在接收到有效的 SET COUPON REDEMPTION STATUS（设置优惠券赎回状态）命令时，小程序就可以转变到下一状态 2608。如图 26 中所示，在接收到有效的 ABORT TRANSACTION（中止交易）命令时，优惠券小程序就可以转变到下一状态 2610。

[0304] 状态 2608 可以包括 TRANSACTION COMPLETED（交易完成）状态，其指示 PoS 终端可能已经完成了交易的阶段 2，并且可能已经设置了 COUPON REDEMPTION STATUS（优惠券赎回状态）。此外，该状态 2608 可以指示钱包客户端现在可以继续进行交易的阶段 3。优惠券小程序可以保持在该状态中，直到小程序接收到有效的 RESET COUPONS（重置优惠券）命令或有效的 ABORT TRANSACTION（中止交易）命令。

[0305] 状态 2610 可以包括 TRANSACTION ABORTED（交易中止）状态，其指示（与优惠券呈现 & 赎回相关的）PoS 终端交易可能已经由钱包客户端或 PoS 终端中止。优惠券小程序可以保持在该状态 2610 中，直到小程序接收到有效的 RESET COUPONS（重置优惠券）命令。

[0306] 根据示例性且非限制性实施例，本公开提供支持优惠券相关功能性的命令，其可以相关于与 PoS 终端的交互。在示例中，这些命令可以基于如由 ISO/IEC 7816 所定义的用于与 SE 的所有通信的标准智能卡命令和响应协议。在示例中，表 CAT1 可以概括类字节（CLA）和指令（INS）字节的编码。

CLA	INS	命令
‘00’	‘A4’	选择
‘C0’	‘01’	取得小程序状态
‘C0’	‘03’	中止交易
‘C0’	‘11’	读取优惠券
‘C0’	‘41’	设置优惠券赎回状态

CAT1 :命令概括 CLA 和 INS 命令。

[0307] 在示例中，优惠券小程序可以返回可以按照 ISO/IEC 7816-4 编码的状态字节。小程序可以返回在表 CAT2 中示出的一组一般状态字节。该表可以列出状态字 SW1、状态字 SW2 及其相关联的含义。

SW1	SW2	含义
‘90’	‘00’	正常处理
‘67’	‘00’	错误长度
‘69’	‘85’	使用条件未满足
‘6A’	‘80’	命令数据中的不正确值
‘6A’	‘82’	未找到小程序
‘6A’	‘84’	存储器不够
‘6A’	‘86’	不正确参数 P1-P2
‘6D’	‘00’	指令不被支持
‘6E’	‘00’	类不被支持

CAT2 :一般状态字节。

[0308] 在示例中，本公开可以包括 ABORT TRANSACTION（中止交易）命令，其可以将小程序从其现有状态移动到交易中止状态。在示例中，ABORT TRANSACTION（中止交易）命令可以根据表 CAT3 来编码。该表可以列出一个或多个代码，诸如类字节（CLA）、指令字节（INS）、参数 P1、参数 P2 等等。在示例中，可能不存在从该命令返回的数据。

代码	值

CLA	‘C0’
INS	‘03’
P1	‘00’
P2	‘00’
Lc	‘00’
数据	不存在
Le	‘00’

CAT3 :ABORT TRANSACTION (中止交易) 命令消息。

[0309] 此外,命令的成功执行可以由状态字节‘9000’指示。ABORT TRANSACTION (中止交易)命令可以返回诸如在表 CAT1 中列出的一般错误条件。另外,ABORT TRANSACTION (中止交易) 可以返回如表 CAT3 中所列出的以下特定错误和警告条件中的一个。

SW1	SW2	含义
‘67’	‘00’	错误长度
‘69’	‘85’	使用条件未满足
‘6A’	‘86’	不正确参数 P1-P2

CAT4 :用于 ABORT TRANSACTION (中止交易) 的状态字节。

[0310] 在示例中,GET APPLET STATUS (取得小程序状态) 命令可以用于检索小程序的当前状态。该命令可以返回小程序的当前状态。表 CAT5 可以列出与该命令相关联的代码和对应值。

代码	值
CLA	‘C0’
INS	‘01’
P1	‘00’
P2	‘00’
Lc	‘00’
数据	不存在
Le	‘00’

CAT5 :GET APPLET STATUS (取得小程序状态) 命令消息。

[0311] 响应于该命令,可以返回数据字段。在示例中,小程序状态可以被返回并编码为 1 字节,并且该数据字段的细节可以在本描述中如下所述的表 CAT17 中找到。此外,命令的成功执行可以由状态字节‘9000’指示。GET APPLET STATUS (取得小程序状态) 命令可以要么返回诸如在表 CAT1 中列出的一般错误条件。另外,GET APPLET STATUS (取得小程序状态) 命令可以返回如表 CAT6 中所列出的以下特定错误和警告条件中的一个。

SW1	SW2	含义
‘67’	‘00’	错误长度
‘6A’	‘86’	不正确参数 P1-P2

CAT6 :用于 GET APPLET STATUS (取得小程序状态) 的状态字节。

[0312] 在示例中,READ COUPONS(读取优惠券)命令可以用于检索优惠券数据。此外,READ COUPONS (读取优惠券) 命令可以根据表 CAT7 来编码。

代码	值
CLA	‘C0’
INS	‘11’
P1	‘00’
P2	见表 CAT8
Lc	‘00’
数据	不存在

Le	‘00’

CAT7 :READ COUPONS (读取优惠券) 命令消息。

[0313] 表 CAT7 包括参考控制参数 P2, 其可以控制相继 READ COUPONS (读取优惠券) 命令的数量。此外,参考控制参数 P2 可以根据下表 CAT8 来编码

P2	含义
‘00’	取得所有出现
‘01’	取得下一出现

CAT8 :READ COUPONS (读取优惠券) 参考控制参数 P2。

[0314] 在示例中,如果在当前应用会话内没有接收到之前的 Read Coupons(读取优惠券) [取得所有出现],则可以拒绝 Read Coupons (读取优惠券)[取得下一出现] 命令。此外,响应于 READ COUPONS (读取优惠券) 命令,如可以在客户特定优惠券规范中定义的优惠券数据的多个出现可以被返回。

标签	Len	值
‘E1’	Var	客户特定优惠券数据

CAT9 :READ COUPONS (读取优惠券) 数据字段。

[0315] 此外,该命令的成功执行可以由状态字节 ‘9000’ 指示。读取优惠券命令可以要么返回诸如在表 CAT10 中列出的一般错误条件。

SW1	SW2	含义
‘63’	‘10’	更多数据可用
‘67’	‘00’	错误长度
‘69’	‘85’	使用条件未满足
‘6A’	‘86’	不正确参数 P1-P2

CAT10 :用于 READ COUPONS (读取优惠券) 的状态字节。

[0316] 在示例中,SELECT (选择) 命令可以用于选择优惠券小程序。SELECT (选择) 命令可以根据表 CAT11 来编码。

代码	值
CLA	‘00’
INS	‘A4’
P1	‘04’
P2	‘00’
Lc	AID 的长度
数据	AID 值
Le	‘00’

CAT11 :SELECT (选择) 命令消息。

[0317] 命令的“数据”字段可以包含可以选择的优惠券小程序的应用标识符(AID),其可以按客户要求来定制。

[0318] 响应于该命令,可以获得返回的消息。返回的消息可以包括数据字段,该数据字段可以包括文件控制接口(FCI) 并且可以根据表 CAT12 来编码。

标签	Len	值					存在
'6F'	Var	FCI 模板					M
		标签	Len	值			存在
		'84'	Var	DF 名称（小程序 AID）			M
		'A5'	Var	专有数据			M
				标签	Len	值	存在
				'9F07'	2B	C-SAM 优惠券小程序-版本 Lx API ID	M
				'9F09'	2B	C-SAM 优惠券小程序-版本 Lx API 版本	M
				'9F11'	2B	优惠券格式 ID	M
				'9F13'	2B	优惠券格式版本	M

CAT12 :SELECT（选择）消息响应数据字段。

[0319] 服务提供者小程序的 AID（标签'84'）可以是如本文所描述的 AID。优惠券小程序 API ID（标签'9F07'）可以指示对优惠券小程序 API 的支持。此外，优惠券小程序 API 版本（标签'9F09'）对于该版本可以是'0101'。优惠券格式 ID 可以是客户可定制的优惠券格式 ID，并且优惠券格式版本可以是客户可定制的优惠券格式版本。此外，可以在表 CAT12 中列出附加的两个字节标签。

[0320] 此外，SELECT（选择）命令的成功执行可以由状态字节'9000'指示。在出现错误时，SELECT（选择）命令就可以返回诸如在表 CAT1 中列出的一般错误条件。另外，SELECT（选择）命令可以返回如表 CAT13 中所列出的以下特定错误和警告条件中的一个。

SW1	SW2	含义
'67'	'00'	错误长度
'6A'	'80'	命令数据中的不正确值
'6A'	'82'	未找到小程序
'6A'	'86'	不正确参数 P1-P2

CAT13 :用于 SELECT（选择）的状态字节。

[0321] 在示例中，SET COUPON REDEMPTION STATUS（设置优惠券赎回状态）命令可以用于设置优惠券赎回状态。在该命令成功执行时，小程序状态就可以改变为"交易完成"。SET COUPON REDEMPTION STATUS（设置优惠券赎回状态）命令可以根据表 CAT14 来编码。

代码	值
CLA	'C0'
INS	'41'
P1	'00'
P2	'00'
Lc	'0A'
数据	优惠券赎回状态
Le	'00'

CAT14 :SET COUPON REDEMPTION STATUS（设置优惠券赎回状态）命令消息。

[0322] 该命令的执行可以不返回任何数据。此外，SET COUPON REDEMPTION STATUS（设置优惠券赎回状态）命令的成功执行可以由状态字节'9000'指示。在出现错误时，SET COUPON REDEMPTION STATUS（设置优惠券赎回状态）命令就可以返回诸如在表 CAT1 中列出的一般

错误条件。另外,SET COUPON REDEMPTION STATUS (设置优惠券赎回状态)命令可以返回如表 CAT15 中所列出的以下特定错误和警告条件中的一个。

SW1	SW2	含义
'67'	'00'	错误长度
'69'	'85'	使用条件未满足
'6A'	'86'	不正确参数 P1-P2

CAT15 :用于 SET COUPON REDEMPTION STATUS (设置优惠券赎回状态) 的状态字节。

[0323] 根据示例性且非限制性实施例,本公开提供在小程序与 PoS 终端之间的交互的细节。图 27 图示了在小程序 2702 与 PoS 终端 2704 之间的交互。

[0324] 在 2708 处,PoS 终端 2704 可以发出 SELECT(选择)命令以选择优惠券小程序 2702。在 2710 处,优惠券小程序 2702 可以以文件控制信息(FCI) 做出响应。在 2712 处,PoS 终端 2704 可以发出 GET APPLET STATUS (取得小程序状态) 命令。在 2714 处,优惠券小程序 2702 可以返回应用状态。在 2718 处,在确定了应用状态是"READY FOR COUPON REDEMPTION (就绪以供优惠券赎回)"时,PoS 终端 2704 可以发出 READ COUPONS (读取优惠券) 命令,其中 P2=Get All Occurrences(取得所有出现)。在 2720 处,优惠券小程序 2702 可以返回针对可用优惠券的优惠券数据。如果存在多个优惠券,则可以在响应数据中包括针对多个优惠券的优惠券数据。如果小程序可以包括所有优惠券数据,则小程序可以发送 SW='9000'。如果小程序不可以包括针对所有优惠券的优惠券数据,则小程序可以发送警告 SW='6310',其指示可以存在更多数据。在 2722 处,PoS 终端 2704 可以解析数据和 SW。如果 SW 可以包括'6310',则 PoS 终端可以发出另一 READ COUPONS (读取优惠券) 命令,其中 P2=Get Next Occurrences (取得下一出现)。

[0325] 在接收到其中 P2=Get Next Occurrences (取得下一出现) 的 READ COUPONS (读取优惠券) 命令时,在 2724 处,优惠券小程序 2702 就可以尝试发送针对剩余优惠券的优惠券数据。至于其中 P2=Get All Occurrences (取得所有出现)的 READ COUPONS (读取优惠券)命令,如果存在多个优惠券待决以发送,则优惠券小程序 2702 可以在响应数据中包括针对多个待决优惠券的优惠券数据,如图 27 中 2724 处所示。一旦小程序包括所有待决优惠券数据,小程序就可以发送 SW= '9000'。如果小程序不包括针对所有待决优惠券的优惠券数据,则小程序可以发送警告 SW= '6310',其指示可以存在更多数据。

[0326] 在响应于 READ COUPONS (读取优惠券)而接收到 SW='9000'时,在 2728 处,PoS 终端 2704 就可以继续进行以处理优惠券。在处理优惠券数据之后,在 2730 处,PoS 终端 2704 可以通过发出 SET COUPON REDEMPTION STATUS (设置优惠券赎回状态) 命令来设置优惠券赎回状态。字节阵列始终为等于优惠券最大数量的长度。可能需要针对所有字节将优惠券赎回状态设置为有效优惠券状态——即使存在较少优惠券。如果接收到比最大值少的优惠券,则对应于未接收到的那些的优惠券赎回状态字节可以被设置以指示"不存在优惠券"。响应于对 READ COUPONS (读取优惠券) 命令的各种调用,可以以优惠券数据的接收顺序设置优惠券赎回状态。即使在不存在赎回的优惠券的情况下,PoS 终端 2704 也可能被要求发出以下两个命令之一 :SET COUPON REDEMPTION STATUS (设置优惠券赎回状态) 或 ABORT TRANSACTION (中止交易)。

[0327] 图 28 图示了根据示例的在交易流期间优惠券小程序的状态机图。状态 2802 可以指示 READY FOR UPLOADING COUPONS (就绪以供上传优惠券) 状态,其指示来自优惠券小程

序的所有优惠券可能已经被重置，并且钱包客户端现在可以上传为该交易选择的优惠券。在示例中，如图 28 中所示，如果命令 GET APPLET STATUS(取得小程序状态)导致错误，则小程序可以保持在状态 2802 中。在接收到有效的 ENABLE TRANSACTION（使能交易）命令时，优惠券小程序就可以转变到下一状态 2804。

[0328] 2804 状态可以包括 READY FOR COUPON REDEMPTION（就绪以供优惠券赎回）状态，其指示钱包客户端可能已经将所有优惠券附着到优惠券小程序的存储器，并且 PoS 终端可以继续进行交易的阶段 2。在示例中，如图 28 中所示，如果命令 READ COUPONS（读取优惠券)导致错误，则优惠券小程序可以保持在状态 2804 中。在接收到有效的 SET COUPON REDEMPTION STATUS（设置优惠券赎回状态）命令时，小程序就可以转变到下一状态 2808。如图 28 中所示，在接收到有效的 ABORT TRANSACTION（中止交易）命令时，优惠券小程序就可以转变到下一状态 2810。

[0329] 状态 2808 可以包括 TRANSACTION COMPLETED（交易完成）状态，其指示 PoS 终端可能已完成了交易的阶段 2，并且可能已经设置了 COUPON REDEMPTION STATUS（优惠券赎回状态)。此外，该状态 2808 可以指示钱包客户端现在可以继续进行交易的阶段 3。优惠券小程序可以保持在该状态中，直到小程序可以接收有效的 RESET COUPONS（重置优惠券）命令或有效的 ABORT TRANSACTION（中止交易）命令。

[0330] 状态 2810 可以包括 TRANSACTION ABORTED（交易中止）状态，其指示(与优惠券呈现 & 赎回相关的）PoS 终端交易可能已经由钱包客户端或 PoS 终端中止。优惠券小程序可以保持在该状态 2810 中，直到小程序可以接收有效的 RESET COUPONS（重置优惠券）命令为止。

[0331] 在示例中，表 CAT16 可以列出接受命令应用数据单元的各种可适用条件。在拒绝的情况中，如下表中所示，优惠券小程序可以返回状态字 SW=’6985’。

命令	就绪以供上传优惠券	就绪以供优惠券赎回	交易完成	交易中止
选择	接受	接受	接受	接受
取得小程序状态	接受	接受	接受	接受
中止交易	拒绝	接受	拒绝	拒绝
读取优惠券	拒绝	接受	拒绝	拒绝
设置优惠券赎回状态	拒绝	接受	拒绝	拒绝

CAT16 :命令接受矩阵。

[0332] 根据示例性且非限制性实施例，本公开提供与优惠券小程序相关联的各种数据对象。在示例中，可以公开包括优惠券小程序 API ID 的数据对象。优惠券小程序 API ID 可以编码为 2 字节结构。为了确定是否可以支持优惠券小程序 API，PoS 终端可以利用掩码‘0102’对值执行逻辑与操作。与操作的结果当等于‘0102’时可以指示可以支持优惠券小程序。

[0333] 在示例中，可以公开包括小程序 API 版本的数据对象。优惠券小程序版本可以编码为 2 字节结构。在示例中，最高有效字节可以表示主发布版本，并且最低有效字节可以表示次发布版本。在示例中，可以公开包括小程序状态的数据对象。小程序状态可以标识小程序的当前状态，并且可以编码为 1 字节。在示例中，表 CAT17 可以列出与小程序的不同状态相关联的状态值。

状态	描述
‘01’	就绪以供上传优惠券

‘02’	就绪以供优惠券赎回
‘40’	交易完成
‘80’	交易中止
‘FF’	未知状态

CAT17 :小程序状态。

[0334] 在示例中,可以公开包括优惠券赎回状态的数据对象。优惠券赎回状态可以存储在每个所附优惠券的一个字节中。对于所支持的最大数量的优惠券的优惠券赎回状态可以存储在长度为最大数量优惠券的字节阵列中。在示例中,对于优惠券 1 的赎回状态可以存储在字节阵列的 B1 中,并且对于优惠券 2 的赎回状态存储在字节阵列的 B2 中。表 CAT18 可以列出优惠券赎回状态的各种受支持的值。

优惠券状态	描述
‘00’	未使用
‘01’	赎回
‘05’	拒绝
‘FF’	不存在优惠券

CAT18 :优惠券赎回状态。

[0335] 在示例中,数据对象可以包括一个或多个优惠券,并且优惠券格式可以取决于实现的类型。

[0336] 根据示例性且非限制性实施例,本公开提供了 PoS 终端的各种处理特征。在示例中,PoS 终端可以遵照 ISO/IEC 14443-4。当执行交易时,PoS 终端可以分析可以由小程序返回的数据对象。例如,PoS 终端可以检查可以由小程序返回的强制性数据对象。如果 PoS 终端可能未找到强制性数据对象,则 PoS 终端可以中止交易。类似地,如果 PoS 终端可能在数据中重组了不正确的格式化,则 PoS 终端可以中止交易。PoS 终端可以采取应有关注以确保可以以鲁棒方式操控处理例外(例如,不正确格式化等)。

[0337] 在示例中,PoS 终端可以在执行交易时执行各种处理操作。PoS 终端可以如上面所讨论那样格式化 SELECT (选择)命令,并且可以核实可以包括在 SELECT (选择)命令的响应消息中的 FCI 的格式化。在示例中,PoS 终端可以根据表 CAT12 来核实 FCI 的格式化。如果可能检测到 FCI 未根据表 CAT12 来格式化,则 PoS 终端可以终止处理并且确保取消选择小程序。

[0338] 此外,PoS 终端可以分析可以在 FCI 中返回的 DF 名称(标签‘84’)。如果 DF 名称与 SELECT (选择)命令的数据字段中所提供的 AID 不相同,则 PoS 终端可以终止处理并且确保取消选择小程序。另外,PoS 终端可以通过检查可以在 FCI 中返回的 API ID(标签‘9F07’)来核实其可以支持 API。如上文所讨论的,PoS 终端可以利用掩码‘0102’来对值执行逻辑与操作以确定对 API 的支持的可用性。如果与操作的结果不等于‘0102’,则 PoS 终端可以终止处理并确保取消选择小程序。

[0339] 在示例中,PoS 终端可以核实其可以支持可以在 FCI 中返回的 API 版本(标签‘9F09’)。如果其不可以被确定,则 PoS 终端可以终止处理并确保取消选择小程序。在示例中,PoS 终端可以核实其可以支持对可以在 FCI 中返回的优惠券格式 ID (标签‘9F11’)的操控。如果这不可以被核实,则 PoS 终端可以终止处理并确保取消选择小程序。在示例中,PoS 终端可以核实其可以支持对可以在 FCI 中返回的优惠券格式版本(标签‘9F13’)的操控。如果这不可以被核实,则 PoS 终端可以终止处理并确保取消选择小程序。

[0340] 在示例中，PoS 终端可以处理 GET APPLET STATUS（取得小程序状态）命令。在成功选择优惠券小程序之后，PoS 终端可以通过使用 GET APPLET STATUS（取得小程序状态）来查询当前小程序状态。在示例中，PoS 终端可以识出小程序状态可以不是“就绪以供优惠券赎回”。在这样的条件中，PoS 终端可以终止处理并且确保取消选择小程序。在示例中，PoS 终端可以识出小程序状态可以是“就绪以供优惠券赎回”。在这样的条件中，PoS 终端可以继续进行对优惠券的正常处理。

[0341] 在示例中，PoS 终端可以处理 READ COUPONS（读取优惠券）命令。PoS 终端可以发出 READ COUPONS(读取优惠券)[取得所有出现] 命令以读取优惠券数据。该命令的成功执行可以由状态字节‘9000’指示。这可以指示所有优惠券数据可能已经被发送。此外，READ COUPONS（读取优惠券）命令可以返回其中 SW= ‘6310’的警告以指示“更多数据可用”。在接收到其中 SW= ‘6310’的响应时，PoS 终端就可以发出后续的 READ COUPONS（读取优惠券）[取得下一出现] 以检索附加数据。在示例中，在当前应用会话中，如果 READ COUPONS（读取优惠券)[取得下一出现] 命令可以在之前并且在没有发出 READ COUPONS（读取优惠券）[取得所有出现] 的情况下发出，则可以拒绝该命令。此外，PoS 终端可以检查以确保所有强制性数据对象存在。在示例中，PoS 终端可能遇到标签(Tag)在优惠券对象中多于一次的出现。在这样的条件中，PoS 终端可以拒绝优惠券。

[0342] 在示例中，PoS 终端可以处理 SET COUPON REDEMPTION STATUS（设置优惠券赎回状态）命令。在验证了所应用的优惠券时，PoS 终端就可以发出 SET COUPON REDEMPTION STATUS（设置优惠券赎回状态）命令以更新小程序上的优惠券赎回状态。可以仅针对自上一次 READ COUPONS（读取优惠券）[取得所有出现] 后可能已经读取的优惠券考虑优惠券赎回状态。此外，如果未发出 READ COUPONS（读取优惠券）[取得所有出现]，则可以忽略所有优惠券赎回状态。如果自上一次对 READ COUPONS（读取优惠券）[取得所有出现] 的调用后未读取所有优惠券，则可以忽略针对未读取的优惠券的优惠券赎回状态。

[0343] 在示例中，PoS 终端可以处理 ABORT TRANSACTION(中止交易)命令。PoS 终端可能遇到错误或者其中 PoS 可能不能够继续进行处理的状态。在这样的条件中，PoS 终端可以发出 ABORT TRANSACTION（中止交易)命令以中断交易。在成功处理 ABORT TRANSACTION（中止交易）命令时，就可能失去优惠券赎回状态。此外，在接收到对 ABORT TRANSACTION（中止交易）命令的响应时，PoS 终端就可以终止处理并且确保取消选择小程序。

[0344] 对于典型部署的高层级架构——

图 29 图示了在示例性实施例中本发明的部署的系统 2900 的高层级架构。

[0345] MTP 2902 可以设置在生态系统提供者的域 2904 中，并且可以集成到域 2904 的企业服务总线(ESB) 2908 中。域 2904 可以包括钱包服务器 2918。钱包服务器 2918 可以包括至少一个钱包 2920 和 MTP 2902。钱包 2920 可以包括至少一个微件 2922。在示例中，ESB 2908 可以连接 MTP 2902 服务提供者 2910 的生态系统。ESB 2908 可以用于设计和实现面向服务的架构中的互相交互的软件应用之间的交互和通信，如图 29 中所图示。可能需要 ESB 2908 以用于实现具有多个钱包应用的生态系统，如图 29 中所图示。ESB 2908 可以监视和控制在各种服务之间(例如在钱包服务器 2918 与钱包提供者 2924 之间)的消息交换的路由。

[0346] MTP 2902 可以具有多个角色。例如，MTP 可以执行钱包和交易管理，连同可以与将

钱包递送给多个不同移动环境 2912 相关联的所有相关商业服务。在示例中，MTP 2902 可以提供构建用于钱包的微件的第三方开发者和服务提供者 2910 的生态系统的建立，而不必担心手持机分片。在示例中，MTP 2902 可以建立与各种生态系统(例如，服务提供者 2910 的生态系统、多个移动环境)的多个连接器。连接器连同 MTP 2902 一起可以提供改善的用户体验和各种服务集合，所述服务集合无论时间或位置都可以对于终端用户是可访问的。

[0347] 生态系统提供者的域 2904 可以包括可信服务管理器(TSM)2912。TSM 2912 可以被配置用于微件的空中供应和密钥管理。TSM 1812 可以与 MTP 2902 协同以促进近场通信体验。在示例中，微件 2922 的供应和钱包 2920 可以通过 MTP 2902 来促进(通过图 30-36 来描述)。在示例中，钱包 2920 的设置和管理可以通过 MTP 102 来促进(通过图 30-36 来描述)。在示例中，可以通过 TSM 2912 来进行小程序发行和个性化(通过图 30-36 来描述)。

[0348] 在示例中，服务提供者可以使用 MTP 软件开发套件来构建可以在 MTP 钱包容器内运行的微件。微件可以提供 OTA 增值服务和邻近 NFC 服务二者。钱包容器可以向微件 2922 提供安全运行时环境和服务以与服务器进行 OTA 通信(OTA 服务)，并且还与安全元件通信以管理用于邻近 NFC 交易的支付和非支付小程序。在示例中，微件 2922 可以具有在安全元件中运行的相关联的小程序。在示例中，微件 2922 可以没有在安全元件中运行的相关联的小程序。在示例性实施例中，服务提供者可以仅提供 OTA 服务。

[0349] 生态系统提供者 2904 可以确保在商家位置 2914 处的所有 NFC 读取器 2930 都可以被配备以操控由可以加载到安全元件中的各种小程序所使用的邻近协议。NFC 读取器 2930 可以通过获取网络来处理交易，所述获取网络可以将交易转换到云中的适当生态系统服务提供者。在示例中，钱包 2920 或微件 2922 可以与生态系统云进行通信以确定服务状态和各种其它保健(hygiene)要求，像余额、交易历史、存储的充值等。

[0350] NFC 读取器 2930 可以促进安全元件形状因数、微 SD、嵌入式或 UICC 的最终化(如通过图 31 所详细描述的)。NFC 读取器 2930 和联合基础设施可以促进设备的最终化(finalization)。NFC 读取器 2930 可以促进支付协议最终化。NFC 读取器 2930 可以促进读取器分布和支持。

[0351] 如图 29 中所图示，移动钱包 2920 可以与来自系统 2900 的各种元件交互。移动钱包 2920 可以与钱包提供者 2924 以及多个微件提供者 2928 交互。在示例中，移动钱包 2920 可以与像支持、消息服务器和其它服务提供者那样的各种基础设施系统交互。

[0352] 每个生态系统可以包括多个生态系统行动者(actor)。在示例性实施例中，生态系统行动者可以包括近场通信(NFC)小程序、应用商店、条形码读取器、生态系统提供者、发行者、消息源、移动电话、移动钱包、NFC 销售点终端、原始装备制造商(OEM)提供者、安全元件(SE)、服务提供者(或微件提供者)、电信运营商(telco)、可信服务管理器(TSM)、TSM 代理客户端、用户、钱包管理员、钱包伴随小程序(WCAp)、钱包提供者、钱包服务器以及微件。在示例中，安全元件上的 NFC 小程序可以与 NFC 销售点读取器执行交易。在示例中，应用商店可以指代应用分布机制，其可以由诸如 apple®、blackberry®/rim® 以及 android®/Google® 之类的设备制造者所设立。在示例中，条形码读取器可以用于扫描条形码。在示例中，拥有和设立移动钱包生态系统的生态系统聚集器实体可以与生态系统操作者或合作伙伴协作。生态系统聚集器实体可以设立基础设施和聚集服务提供者以使能移动支付和移动商务生态系统。在示例中，发行者可以是可以发行卡、优惠券、忠诚度、礼品以及其它标记的实体。

在示例中，消息源可以是可以使用钱包 API 创建用于钱包用户的消息的实体。在示例中，移动电话可以是钱包在其上运行的那个。在示例中，移动钱包可以是用于移动手持机的可下载客户端应用，用以向用户提供无缝支付体验连同一套用于邻近（NFC）和空中交易二者的增值服务。在示例中，NFC 销售点终端可以与 NFC 电话交互并执行交易。在示例中，OEM 提供者可以提供设备、微 SD、SE 等。在示例中，SE 可以表示 NFC 与用于安全和可信交易的智能卡技术的组合，其中，NFC 可以提供用于连接性的 RF 前端，并且智能卡可以提供安全性 / 加密引擎。在示例中，服务提供者可以是创建钱包并向钱包用户提供支付和非支付服务的实体。在示例中，电信运营商（Telco）可以提供 SMS 和数据服务。在示例中，TSM 可以管理安全元件上的小程序和卡数据的 OTA 发行，并且可以提供密钥管理服务。在示例中，TSM 代理客户端可以位于移动电话上并且与 TSM 服务器和安全元件交互。在示例中，用户可以是移动钱包的终端用户。在示例中，钱包管理员可以管理钱包的后端操作，包括钱包生命周期。在示例中，钱包伴随小程序可以使能安全钱包 & 微件生命周期管理，其可以包括钱包 & 微件认证、钱包状态管理和设置，并利用钱包服务器提供用于安全认证的基于票据的核实。在示例中，钱包提供者可以是发动有品牌的钱包应用并将其提供给用户的实体。在示例中，钱包服务器可以是鲁棒的、可伸缩的服务器平台，其可以管理钱包生态系统、钱包生命周期、微件供应 & 与其它生态系统合作伙伴的集成。在示例中，微件可以是绑定在一起并利用为单个单元的一组商业特征。微件可以包含屏幕 /UI、事件操控以及对应的商业逻辑。

[0353] 用例 - 概观

在各种示例性且非限制性实施例中，钱包用例可以分类成各种种类。这些种类可以包括钱包下载、钱包激活 & 设置、交易 & 微件、生命周期活动以及监管 & 支持。在示例中，钱包下载可以包括用于钱包分布以及在消费者的移动电话上安装钱包软件的用例和方法。可以要求这些服务以用于将用户引入到平台并且向移动电话供应移动软件。在示例中，钱包激活 & 设置种类可以包括用于认证用户证书并建立用户简档以及使用户能够使用移动钱包应用的用例。在示例中，交易 & 微件种类可以包括用以使用户能够下载微件并得到支付 & 非支付标记，以及通过使用激活的移动钱包应用来进行交易的用例。交易用例可以包括 NFC 交易、基于条形码的交易以及空中交易。在示例中，生命周期活动可以包括使用户能够继续使用钱包应用的服务可能需要的用例和特征。在示例中，监管 & 支持用例可以有关于管理移动钱包平台的日常操作和支持活动，并且包括诸如改变设备、锁定和解锁设备等的用例。

[0354] 在各种示例性且非限制性实施例中，微件下载可以是第一用例场景，其后是钱包激活 & 设置。进一步在钱包激活 & 设置之后可以是交易 & 微件用例。交易 & 微件用例之后可以是生命周期活动，并且监管 & 支持用例种类可以支持和帮助钱包下载、钱包激活 & 设置、交易 & 微件、生命周期活动种类。可以在本文中描述少数示例性用例。

[0355] 在示例性且非限制性实施例中，微件下载可以是以具有下载 URL（推送或拉取）的 SMS 用例的形式。在示例中，具有下载 URL（推送或拉取）的 SMS 用例可以允许钱包提供者通过使用 SMS 推送 / 拉取在空中分布应用，而不在应用商店发布应用。在该示例中，移动钱包可以成功地安装在移动电话上，并且用户可以在主屏幕上看到移动钱包图标。具有下载 URL（推送或拉取）的 SMS 用例可以使用像用户、生态系统提供者、移动电话和 telco（电信运营商）的生态系统行动者。对于该用例所需的前提条件可以是要数据使能的移动电话。

[0356] 图 30 图示了描述通过使用具有下载 URL（推送或拉取）的 SMS 在用户的移动电话

上安装移动钱包的方法 3000 的流程图。方法 3000 包括在步骤 3002 处通过向短代码发送 SMS 来发起 SMS。在示例中，SMS 可以由用户发送给 Telco（电信运营商）或钱包提供者，这也称为 SMS 推送。在可替换实施例中，用户可以从 telco（电信运营商）或钱包提供者接收 SMS，这也称为 SMS 拉取。在可替换实施例中，用户可以从网站发起 SMS。步骤 3002 的完成可以取决于 SMS 短代码（SMSC）的成功发起。方法 3000 此外可以包括在步骤 3004 处将 SMS 发送给用户。在示例中，Telco（电信运营商）可以将 SMS 发送给用户。在可替换实施例中，用户可以使用与 SMSC 的整合。步骤 3004 的完成可以取决于 SMS 整合的方法，其可以利用 Telco（电信运营商）或 SMSC。

[0357] 方法 3000 此外 包括在步骤 3008 处由用户接收具有欢迎文本的 URL。用户可以选择在 SMS 中提供的 URL。步骤 3008 的完成可以取决于 URL 描述符和欢迎消息。方法 3000 此外包括在步骤 3010 处移动电话向钱包服务器发送请求以提供可下载文件。在方法 3000 的步骤 3012 处，钱包服务器可以此外被配置为用于将钱包软件提供给移动电话。在方法 3000 的步骤 3014 处，移动电话可以被配置用于在移动电话上下载应用并提示用户以安装应用。在已经由移动电话向用户提示了安装之后，在方法 3000 的步骤 3018 处，用户可以确认安装。在方法 3000 的步骤 3020 处，移动电话可以在接收到用户确认之后安装应用。

[0358] 在示例性且非限制性实施例中，钱包激活用例处理可以使用户能够激活移动钱包，并在用户 & 诸如移动钱包、移动电话、移动电话、安全元件、钱包服务器、TSM 和钱包提供者之类的生态系统行动者之间建立安全链接。在示例中，钱包激活用例行动者像移动手持机、TSM、用户、移动钱包、钱包提供者、钱包服务器。钱包激活仅当用户可能已经在移动手持机上成功地安装了移动钱包（如通过图 30 所描述的由用户预加载或下载）时才可以发生。此外，为了钱包激活发生，SE 可以与移动手持机相连接 / 相关联（用于像微 SD 和 UICC 的可移除 SE）。此外，用户可以具有用于钱包激活发生的数据连接性。在钱包的成功激活之后，用户可以能够使用钱包应用。

[0359] 图 31 图示了由用户进行的钱包激活的方法 3100。方法 3100 包括在步骤 3102 处移动钱包向用户示出注册 & 激活屏幕，其包括电子邮件地址和个人标识号（PIN）。在示例中，PIN 可以是 4 数位号码。在示例中，PIN 可以被配置为是数字的，并且具有与其相关联的商业规则，例如，4-10 数位、无序列、无重复数字等。在示例中，可以定制注册屏幕以捕获附加参数，如创建用户简档所需要的，例如，用户名、用户照片或任何其它用户签名。方法 3100 此外包括在步骤 3104 处用户在注册和激活屏幕上键入所有需要的信息。方法 3100 此外包括在步骤 3108 处移动钱包向用户示出最新项目和条件（T&C）。在示例中，可以在运行时从服务器拉取 T&C。在示例中，对于每个钱包提供者，T&C 可以不同。在示例中，对 T&C 的接受可以连同版本号一起被告知给钱包提供者。方法 3100 可以在步骤 3110 处要求用户接受 T&C。在示例中，如果用户可能拒绝 T&C，则可以向用户示出消息，指示可能需要对 T&C 的接受以激活钱包。

[0360] 方法 3100 此外包括在步骤 3112 处移动钱包从移动电话取出安全元件 ID（CIN）、设备 ID（IMEI）和 SIM ID（ICCID），并将信息（包括电子邮件 ID、PIN 和 T&C 版本以及时间戳）发送给钱包服务器。在示例中，并非在所有情况中所有信息都可以对移动钱包是可用的，并且可能需要分离的工作流或与 Telco（电信运营商）或服务提供者的交互来捕获所需细节。例如，如果移动号码不能从 SIM 或设备 API 是可获得的，则其可以从来自设备的 HTTP

标头中捕获，如果 Telco（电信运营商）提供它的话。方法 3100 在钱包激活中进一步继续进行之前可能需要来自 Telco（电信运营商）的移动号码和 SIM 信息。方法 3100 此外包括在步骤 3114 处钱包服务器执行第一级别核实以检查是否有任何信息可以是复制。在示例中，如果用户记录可以已经是可用的，则钱包服务器可以向用户提供适当的错误消息。

[0361] 方法 3100 此外包括在步骤 3118 处钱包服务器向钱包提供者发送信息以用于核实。在步骤 3120 处，钱包提供者可以核实该信息并向钱包服务器发送核实结果。在示例中，核实过程可以包括由 Telco（电信运营商）进行的 KYC 检查。核实过程可能是在进一步在钱包激活中继续进行之前由方法 3100 要求的。在步骤 3122 处，钱包服务器可以创建其中用户状态为“不活动”的钱包记录，并向移动钱包发送确认。在步骤 3124 处，移动钱包可以向用户示出消息，即钱包可能处于针对用户而被个性化的过程中。在步骤 3128 处，钱包服务器可以通过 ESB 向 TSM 发送请求与用户信息(如上所讨论的移动号码、SE ID、设备 ID、钱包 ID 以及 PIN)。

[0362] 方法 3100 此外包括在步骤 3130 处 ESB 被配置用于向 TSM 发送信息。在步骤 3132 处，TSM 可以与 TSM 代理客户端通信，并且发起 SE 个性化过程。在步骤 3134 处，TSM 可以利用钱包伴随小程序安装来使安全元件个性化，并且可以利用硬件和钱包相关信息来在其上执行存储数据。在步骤 3138 处，一旦个性化完成，TSM 就可以向钱包服务器发送通知。在步骤 3140 处，钱包服务器可以更新钱包状态为“活动”，并且通知钱包客户端。在方法的步骤 3142 处，移动钱包可以向用户示出确认，其指示成功的钱包激活。

[0363] 在示例性且非限制性实施例中，交易和微件用例可以包括一系列微件管理用例。在示例中，微件管理用例可以包括一系列微件安装用例。在示例中，微件安装用例可以包括应用内用例警报。应用内用例警报可以允许用户基于由钱包提供者所接收的警报来下载微件。钱包提供者可以通过使用 C-SAM 的警报框架来发送警报。应用内用例警报可能需要移动钱包、服务提供者、用户、钱包提供者以及钱包服务器。在激活移动钱包之后，用户可以安装微件作为应用内案例。而且，用户可以登录到移动钱包。服务应当在应用内用例警报之前已经创建了微件。钱包提供者可以在应用内用例警报之前已经复查了微件并且发布了它。钱包提供者可以在应用内用例警报之前已经创建了警报。在应用内警报之后，用户可以已经成功下载了微件。在应用内用例警报中，移动钱包可以向用户示出警报。在示例中，可以在移动钱包消息收件箱中或者在与钱包服务器的通信期间向用户示出警报。然后，用户可以选择警报、查看消息并发起下载。在发起下载之后，移动钱包可以安装微件并且向钱包服务器发送确认。钱包服务器可以向移动钱包发送确认。移动钱包可以向用户示出确认。

[0364] 在示例中，微件安装用例可以包括搜索和下载用例。搜索和下载用例可以允许用户搜索微件并在移动钱包上下载和安装。搜索和下载用例可能需要移动钱包、钱包服务器、服务提供者、用户以及钱包提供者。搜索和下载用例可能需要用户已经激活了移动钱包、用户已经登录到移动钱包、服务提供者已经创建了微件，以及钱包提供者已经复查了微件并发布了它。作为搜索和下载用例的结果，用户可以成功地下载微件。

[0365] 图 32 图示了作为搜索和下载用例的示例的方法 3200。方法 3200 可以包括在步骤 3202 处用户搜索或浏览可用微件的种类。在示例中，用户可以能够通过从服务列表选择微件或者在钱包中键入或选择适当准则来搜索微件。在步骤 3204 处，方法 3200 可以要求移动钱包将基于用户准则的请求发送给钱包服务器。在步骤 3208 处，钱包服务器可以向移动

钱包发送与用户的搜索准则相匹配的所有可用微件的列表。在步骤 3210 处，移动钱包可以向用户示出微件的列表。在步骤 3212 处，用户可以查看来自由钱包服务器发送的列表的任何微件的微件描述，并且发起所期望微件的下载。在方法 3200 的步骤 3214 处，移动钱包可以安装微件并向钱包服务器发送确认。在步骤 3218 处，钱包服务器可以向移动钱包发送确认。在步骤 3220 处，移动钱包可以向用户示出确认。

[0366] 在一些示例性且非限制性实施例中，交易用例可以包括空中（OTA）交易用例。空中交易用例可以描述基于 OTA 交易的交易过程。OTA 交易可以包括用户、移动钱包、微件以及服务提供者。为了 OTA 交易发生，用户可能必须登录到移动钱包中，用户可能必须具有数据连接性，并且用户可能已经在移动钱包上加载了用于 OTA 交易的微件。在示例中，OTA 交易可以促进由用户成功完成交易。

[0367] 图 33 图示了用于 OTA 交易的示例性方法 3300。方法 3300 可以包括在步骤 3302 处用户选择微件。微件选择可以按对于用户可以可用的各种类型的服务来进行，所述服务例如票务、账单支付等。在步骤 3304 处，方法 3300 可以要求移动钱包在步骤 3304 处将微件加载到服务屏幕上（如也在上文讨论的）。在步骤 3308 处，用户可以浏览服务屏幕并选择要支付的产品 / 服务。工作流和用户体验对于不同服务可以不同。服务可以要求在用户可以到达支付屏幕之前的多个交互。在步骤 3310 处，钱包服务器可以将信息从用户发送到服务提供者。在步骤 3312 处，服务提供者可以向用户提供具有支付量的支付请求。在步骤 3314 处，用户可以选择支付卡并提供支付信息。在示例中，用户可以选择在钱包上可用的支付方法，并且其可以允许 OTA 交易。在示例中，用户可以键入卡信息，同时空中行进也在进行。方法 3300 的步骤 3318 可以要求钱包服务器向服务提供者提供支付信息。在步骤 3320 处，服务提供者可以通过支付网关 / 银行业务接口来结算支付，并且可以向钱包服务器提供交易确认。在步骤 3322 处，钱包服务器可以接收交易确认，并且可以向微件和钱包发送信息。在方法 3300 的步骤 3324 处，移动钱包可以向用户示出交易确认和细节。

[0368] 在示例性且非限制性实施例中，生命周期活动可以包括卡发行用例。卡发行用例可以允许卡发行者在钱包上发行卡数据。卡发行用例可以包括发行者、钱包服务器、TSM、用户以及移动钱包。为了卡发行发生，用户可能必须激活移动钱包，用户可能必须登录到移动钱包，服务提供者可能必须创建更新的微件，以及钱包提供者可能必须复查微件和发布其它。之后，卡发行成功完成；用户可以能够下载微件。

[0369] 图 34 图示了作为卡发行用例的示例的方法 3400。方法 3400 可以在步骤 3402 处要求发行者批准支付帐户（例如，信用卡）。在步骤 3404 处，发行者可以在 TSM 的帮助下对安全元件进行个性化。在示例中，TSM 可以安装支付小程序，并且可以用跟踪数据进行个性化。卡发行用例可以取决于支付卡和 SE 上的小程序发行 & 小程序的类型。在示例中，小程序可以是 Pay-pass、zip 等。在步骤 3408 处，方法 3400 可以要求用户经由 ESB 向钱包服务器提供软卡数据。软卡数据可以包含要用于交易的图像、余额、卡昵称以及 AID。方法 3400 在步骤 US610 处此外可以要求钱包服务器创建用户的服务账户的记录。在步骤 3412 处，钱包服务器可以标识要连同服务账户一起发送的微件。在步骤 3414 处，钱包服务器可以在微件上创建散列，并经由 ESB 将微件 ID、版本和散列发送给 TSM。在步骤 3418 处，TSM 可以在来自 TSM 代理客户端的帮助下在安全元件上完成微件散列更新。在步骤 3420 处，一旦微件散列可能被更新，TSM 就可以经由 ESB 通知钱包服务器。在步骤 3422 处，钱包服务器可以

使软卡和微件可用于下载。移动钱包可以执行同步，并且可以在用户的移动电话上下载软卡和微件。

[0370] 在示例性且非限制性实施例中，支持和监管可以包括终止钱包用例。终止钱包用例可以允许钱包管理员终止钱包账户。终止钱包用例可以包括移动钱包、钱包服务器、TSM、TSM 代理客户端。为了钱包终止，钱包管理员可能必须被注册，钱包管理员可能必须具有管理用户简档的特许，以及钱包管理员可能必须至钱包管理控制台。在钱包终止用例完成之后，钱包管理员可以成功终止钱包账户。

[0371] 图 35 图示了作为用以终止钱包的示例的方法 3500。方法 3500 在步骤 3502 处可以要求钱包管理员发起挂起钱包活动。在可能发起挂起活动之后，在步骤 3504 处，钱包服务器可以更新钱包状态为“termination_pending”(终止 _ 待决)。在步骤 3508 处，钱包服务器可以向 ESB 发送请求。在步骤 3510 处，ESB 可以从钱包服务器检索服务账户列表。在步骤 3512 处，ESB 可以向发行者发送请求以终止服务账户。在向发行者发送请求之前，ESB 可以检查服务账户列表。在步骤 3514 处，发行者可以在 TSM 的帮助下在后端终止钱包账户，并且还从 SE 移除支付小程序。在步骤 3518 处，ESB 可以向 TSM 发送请求以移除钱包伴随小程序。在步骤 3520 处，TSM 可以向 TSM 代理客户端发送消息。在步骤 3522 处，TSM 代理客户端可以从 SE 移除钱包伴随小程序。在步骤 3524 处，一旦更新可能完成，TSM 就可以向 ESB 发送通知。在方法 3500 的步骤 3528 处，钱包服务器可以接收更新，并且可以将钱包状态改变为“已终止”。在步骤 3528 之后，可以成功终止钱包。

[0372] 在示例性且非限制性实施例中，微件管理包括发布微件用例。发布微件用例可以允许生态系统提供者发布经批准的微件。发布微件用例可以包括钱包管理员、钱包服务器和服务提供者。在发布微件之前，钱包管理员可能必须被注册，服务提供者可能必须被注册，钱包管理员可能必须登录到钱包管理控制台中，以及微件可能必须被验证和批准。

[0373] 图 36 图示了作为用于发布微件的示例的方法 3600。方法 3600 在步骤 3602 处可以要求钱包管理员登录到钱包服务器系统管理控制台中。在步骤 3604 处，钱包管理员可以从钱包服务器系统管理控制台选择微件管理。在步骤 3608 处，钱包管理员可以从微件管理选择用于发布微件的链接。在步骤 3610 处，钱包服务器可以显示“已批准”状态下的所有微件。在步骤 3612 处，钱包管理员可以选择要发布的微件。在步骤 3614 处，钱包管理员可以键入任何评价或标注并选择发布钱包服务器系统管理控制台的一个正在发布的微件。在步骤 3618 处，钱包服务器可以提示钱包管理员进行确认。在步骤 3620 处，钱包管理员可以确认微件可以被发布。在步骤 3622 处，钱包服务器可以将微件的状态改变为“已发布”。在步骤 3624 处，钱包服务器可以生成用于微件的签名。在方法 3600 的步骤 3628 处，钱包服务器可以确认微件可以已经被发布。

[0374] 行为模型 - 添加微件 - 图 37 是用于第一次添加微件连同 TSM 交互的方法 3700。方法 3700 在步骤 3708 处可以要求用户使用钱包 3702 以用于选择微件 3704。在步骤 3710 处，微件脚本组件 3712。在方法 3700 的步骤 3718 处，脚本组件 3712 可以与微件管理器 3714 通信以便检索微件 3704 的最新版本。在示例中，脚本组件 3712 可以与微件管理器 3714 通信以便检索微件描述符。在示例中，脚本组件 3712 可以与微件管理器 3714 通信以便检索微件二进制。在方法 3700 的步骤 3718 处，微件管理器 3714 可以将微件描述符数据块、微件束的散列、MDN 和微件实例 ID 发送给可信服务管理器(TSM)3720。微件管理器 3714 可以

生成微件响应并传送给钱包 3702。钱包 3702 可以包括用于存储来自微件管理器 3714 的微件二进制和微件响应的数据库。在步骤 3728 处，TSM 3720 可以向安全元件 3722 添加微件 3704。在示例中，安全元件 3722 可以是钱包小程序。

[0375] 图 38 是参照微件加载时间签名、核实和执行时间访问核实的用于微件访问核实的系统 3800 的行为模型。

[0376] 在示例性且非限制性实施例中，钱包 1002 可以被配置使得用户可以从微件列表中选择微件。钱包 1002 可以被配置使得钱包 1002 可以启动由用户选择的微件应用。钱包 1002 可以与微件管理器 3804 交互。微件管理器 1004 可以被配置为初始化资源管理器、控制器，创建默认访问上下文等。在示例中，微件管理器 3804 可以包括微件列表，并且可以促进通过钱包 3802 向用户示出微件列表。系统 3800 可以包括加密解密实用程序(cryto-utility) 3808，其可以被配置为从数据库加载微件束。加密解密实用程序 3808 可以从微件束创建散列，例如，安全散列算法 1 (SHA1)散列。加密解密实用程序 3808 可以将散列和微件束传送到微件管理器 3804。系统 3800 可以包括 NFC 芯片实用程序 3810 和安全元件或钱包小程序 3812。加密解密实用程序 3808 可以发起请求来核实微件束，并且将请求发送给 NFC 芯片实用程序 3810 和钱包小程序 3812。NFC 实用程序 3810 可以促进 3812 的选择。小程序 3812 可以向微件访问上下文 3814 发送核实请求。微件访问上下文 3814 可以核实微件束、微件描述符。在示例中，微件管理器 3804 可以发起与微件访问上下文 3814 的通信。微件访问上下文 3814 可以与访问动作控制器 1018 进行通信。访问动作控制器 1018 可以检查认可和访问以用于所选微件的认可。在示例中，当核实可能通过或失败时，可以向用户发送适当消息。系统 3800 可以包括动作控制器 3820，其可以在认可核实之后执行微件请求。访问知晓屏幕控制器 3822 可以包括在系统 3800 中，并且被配置为如果可以从访问动作控制器 1018 接收到认可访问则示出屏幕。系统 3800 可以包括屏幕控制器 3824。屏幕控制器 3824 可以被配置为在检查到来自微件访问上下文 3814 的显示认可之后在钱包 3802 上显示屏幕。

[0377] 用于安全个性化交易的多域生态系统中的移动交易平台的方法和系统可以提供用于安全商务和支付交易的动态生态系统中的移动交易平台的协议抽象的方法。所述移动交易平台可以是运营商等级(carrier-grade)基础设施平台，其促进安全交易服务的开发以及在空中和通过 NFC 邻近性二者将所述安全交易服务递送到广泛的移动电话环境。所述方法可以包括使原始方法的信息元素分开，并将原始方法的信息元素映射到公共格式的新信息字段。这些新信息字段可以是空的、被填充的，或者通过标记化或一些其它转化方法被转化。该公共格式可以工作用于不同的输入协议，并且递送单个输出协议。该公共格式可以是与每个协议的数据结构一致的元模型。公共模型可以是服务抽象模型，其可以操控底层协议内的原生序列流。输入协议可以是销售点协议、NFC 协议、信用卡协议、借记卡协议、票务协议、EPSON Esc/POS 协议、UTC 标准协议、AEDEX 协议、ICD 2002 协议、最终协议(Ultimate protocol)、CD 5220 协议、DSP-800 协议、ADM 787/788 协议、HP 协议、统一 POS 协议、用于销售点的 OLE 协议、JavaPOS、基于云的协议等等。

[0378] 可以经由通信链路在移动设备与销售点设备之间执行协议抽象的方法。移动设备可以是智能电话、平板计算机、膝上型计算机等等。移动设备可以运行操作系统，诸如 iOS、Android、Windows 等等。移动设备可以运行应用，诸如容器、微件、钱包等等。应用可以存

储与软卡相关的信息。软卡可以是信用卡、忠诚卡、礼品卡、储值卡、票据卡等等。可以从来自多个生态系统的多个服务提供者通过多个运营商或网络提供者来发行软卡。可以通过并由供应架构来供应、维护和使能软卡。供应架构可以包括移动使能层、服务层和软件开发套件。销售点设备可以是在邻近环境中操作的真实世界销售点设备，或者是在远程环境中操作的虚拟世界销售点设备。通信链路可以是有线通信链路、无线通信链路、NFC 链路、蓝牙链路、局域网链路、广域网链路等等。

[0379] 协议抽象的方法可以包括用于安全商务和支付交易的动态生态系统中的移动交易平台的标记化能力。该移动交易平台可以是运营商等级基础设施平台，其促进安全交易服务的开发以及在空中和通过 NFC 邻近性二者将安全交易服务递送到广泛的移动电话环境。标记化能力可以取得私人信息并用标记来替代它。私人信息可以包括 pin 号码、信用卡号、主账户号、其它个人可识别信息等等。标记可以由以下各项一致 :数字字符、字母字符、字母和数字字符的组合、其中字母和数字字符替代主账户号的中间数位的截取的主账户号等等。标记可以是一次性使用标记。标记可以由用于安全商务和支付交易的动态生态系统中的可信实体发行。可信实体可以是位于支付网关、获取者或发行者处的服务器。标记可以生成为数学可逆加密函数、单向不可逆加密函数，或者通过索引函数、序列号或随机生成的号码来指派。

[0380] 参照图 39，图示了根据示例性且非限制性实施例的用于安全商务和支付交易的动态生态系统。用于安全商务和支付交易的动态生态系统 3902 包括销售点终端 3908、在移动设备上运行的应用 3910、销售点设备 3908 与移动设备上运行的移动应用 3910 之间的通信链路。与单个域 3914 相关联的单个协议 3912 可以连接销售点设备 3908 和在移动设备上运行的移动应用 3910。

[0381] 继续参照图 39 的用于安全商务和支付交易的动态生态系统 3904，与多个域 3918 相关联的多个协议 3916 可以连接销售点设备 3908 和在移动设备上运行的移动应用 3910。

[0382] 继续参照图 39 的用于安全商务和支付交易的动态生态系统 3906，与多个域 3918 相关联的多个协议 3916 可以通过使用封装器 3920 来抽象，如本说明书中之前所描述的，用以连接销售点设备 3908 和在移动设备上运行的移动应用 3910。

[0383] 本文描述的多域生态系统安全个性化交易的方法和系统包括通过多域生态系统在安全个性化交易中抽象协议的方法。该方法包括在多域生态系统中的联网设备处接收安全个性化交易中的电子交易消息，其包括多个信息字段。该方法此外可以包括用设备来确定多个信息字段中每一个的类型和内容。该方法此外可以包括将多个信息字段的至少一部分映射到公共格式消息字段。该方法此外可以包括基于映射来配置公共格式交易消息。该方法此外可以包括将公共格式交易消息从联网设备传输到多域生态系统中的第二联网设备。

[0384] 本文描述的多域生态系统安全个性化交易的方法和系统包括通过多域生态系统在安全个性化交易中抽象协议的方法。该方法包括在多域生态系统中的联网设备处接收安全个性化交易中的电子交易消息，其包括多个信息字段。该方法此外可以包括用设备来确定多个信息字段中每一个的类型和内容。该方法此外可以包括用元模型将多个信息字段的至少一部分映射到公共格式消息字段，所述元模型提供安全个性化交易协议的一致映射。该方法此外可以包括基于映射来配置公共格式交易消息。该方法此外可以包括将公共格式

交易消息从联网设备传输到多域生态系统中的第二联网设备。

[0385] 本文描述的多域生态系统安全个性化交易的方法和系统包括通过多域生态系统在安全个性化交易中抽象协议的方法。该方法包括在多域生态系统中的联网设备处接收安全个性化交易中的电子交易消息，其包括多个信息字段。该方法此外可以包括用设备来确定多个信息字段中每一个的类型和内容。该方法此外可以包括将多个信息字段的至少一部分映射到公共格式消息字段。该方法此外可以包括基于映射来配置公共格式交易消息。该方法此外可以包括在协议抽象内容封装器中封装公共格式交易。该方法此外可以包括将经封装的公共格式交易消息从联网设备传输到多域生态系统中的第二联网设备。

[0386] 参照图 40，图示了根据示例性且非限制性实施例的在用于安全商务和支付交易的动态生态系统 4002 中的协议抽象的方法。销售点设备 4004 可以连接到运行在移动设备 4008 上的移动钱包 4006。多个移动钱包可以在移动设备上运行。销售点设备 4004 可以通过通信链路 4010 与在移动设备 4008 上运行的移动钱包 4006 通信。通信链路 4010 可以是有线通信链路、无线通信链路、NFC 链路、蓝牙链路、局域网链路、广域网链路等等。

[0387] 继续参照图 40，在移动设备 4008 上运行的移动钱包 4006 可以经由协议通过网络或无线提供者 4014 和网络 4022 与服务提供者 4012 通信。网络 4022 可以是基于网际协议的网络。经封装的获取信道 4016 可以被建立来促进在销售点设备 4004、运行在移动设备 4008 上的移动钱包 4006 以及获取者设施 4018 之中的协议抽象的通信。如之前在本说明书中所描述的可以封装用于通信的协议的经封装的获取信道 4016 可以促进经由标准化通信协议在销售点设备 4004 与运行在移动设备 4008 上的移动钱包 4006 之间交换电子交易信息，该交换对于移动设备 4008 的用户是无缝的。移动设备 4008 的用户可以在销售点处操作移动设备，而不必知道在 POS 交易期间使用的协议类型。以这种方式，仅支持非基于 IP 的交易流的传统 POS 设备和能够支持聚集的协议交易流(例如，通过 IP 网络)的 POS 设备可以由用户无缝地操作。在移动设备 4008 上运行的移动钱包 4006 可以经由协议通过获取者 4018 和网络 4016 与服务提供者 4012 通信。网络 4016 可以是基于网际协议的网络。销售点设备 4004 可以经由现有的直接交易信道 4020 来通信。现有的直接交易信道可以是非基于网际协议的信道 4020。

[0388] 本文描述的多域生态系统安全个性化交易的方法和系统包括通过多域生态系统进行安全个性化交易的方法。该方法包括在多域生态系统中的联网设备处接收安全个性化交易中的电子交易消息，其包括多个非机密信息字段。该方法此外可以包括在联网设备处配置包括非机密信息和交易标识信息的标记。该方法此外可以包括将标记电子地传输到电子交易消息的源和传输到布置于多域生态系统中的服务提供者服务器。该方法此外可以包括将标记的认证连同标记标识信息一起从电子交易消息的源传输到服务提供者服务器以用于认证由标记所标识的交易。在示例中，服务提供者可以是器件发行者。在示例中，服务提供者可以访问联网设备不可访问的机密信息，以促进认证由标记标识的交易。机密信息可以与数字钱包相关联，该数字钱包布置于从其传输标记认证的设备上。在示例中，传输标记认证可以由在电子交易消息源自的设备上执行的提供者特定的微件来促进。

[0389] 参照图 41，图示了根据示例性且非限制性实施例的在用于安全商务和支付交易的动态生态系统 4102 中的使交易内容标记化的方法。

[0390] 销售点设备 4104 可以由并入了请求执行诸如购买之类的电子交易的移动钱包

4106 的移动电话的附近存在来激活。POS 设备 4104 和移动钱包 4106 可以交换描述所请求的交易的非个人信息(项目、花费、日期 / 时间、POS 身份、POS 服务提供者身份等)连同由移动钱包 4106 生成的唯一标识符。POS 4106 可以向移动钱包 4106 发行在安全编码中并入该信息的标记,并且还可以向支付管理机构提交发行的标记的版本,所述支付管理机构诸如可能已经由移动钱包 4106 指定的发行银行 4112。

[0391] 移动钱包 4106 和发行银行 4112 可以随后交换如标记中所描述的与交易相关的信息以完成交易。在示例中,移动钱包可以向发行银行提交包括对标记的引用(例如,标记 ID)的查询,其可以向发行银行信号通知在标记中标识的交易是授权的。可替换地,发行银行可以向移动钱包提交针对标记中所引用交易的授权的请求。从 POS 终端 4104 和从发行银行 4112 接收的标记的交易信息可以相比较并呈现给移动设备 4108 的用户以进行认证。以这种方式,可以执行交易,而无需 POS 4104 或 POS 通信路径中的其它实体具有对任何用户机密、个人或金融信息的访问,因为用户 / 移动钱包的所有机密、个人和金融信息可以保留在后端服务器(例如,发行银行 4112)上,并且可以通过在移动钱包 4106、POS 终端 4104、获取信道 4120 以及诸如发行银行 4112 之类的支付促进者之中交换的标记来间接访问。

[0392] 销售点设备 4104 可以连接到在移动设备 4108 上运行的移动钱包 4106。多个移动钱包可以在移动设备上运行。销售点设备 4104 可以通过通信链路 4110 或基于可视信息交换(例如,QR 码或条形码等)来与运行在移动设备 4108 上的移动钱包 4106 通信。通信链路 4110 可以是有线通信链路、无线通信链路、NFC 链路、蓝牙链路、局域网链路、广域网链路等等。可视信息交换可以包括用相机接收图像和 / 或在电子显示器上显示图像。

[0393] 继续参照图 41,销售点设备 4104 可以从运行在移动设备 4108 上的移动钱包 4106 接收私人信息,存储私人信息,然后将私人信息发送给发行银行以供授权。发行银行可以直接通过空中接口 4114 或者通过服务提供者 4116 和网络 4118 来接收私人信息。在授权时,发行银行 4112 就可以用标记来替代私人信息。标记可以由以下各项一致 :数字字符、字母字符、字母和数字字符的组合、其中字母和数字字符替代主账户号的中间数位的截取的主账户号等等。标记可以是一次性使用标记。标记可以由用于安全商务和支付交易的动态生态系统中的可信实体发行。可信实体可以是位于支付网关、获取者或发行者处的服务器。标记可以生成为数学可逆加密函数、单向不可逆加密函数,或者通过索引函数、序列号或随机生成的号码来指派。然后,发行银行 4112 可以将标记返回到销售点设备 4102。然后,销售点设备 4102 可以用标记来替代私人信息,并且存储标记。标记可以包含在封装器中,如之前在本说明书中所描述的,并且可以通过经封装的获取信道 4120 在运行在移动设备 4108 上的移动钱包 4106、销售点设备 4104 以及获取银行 4112 之中传送。

[0394] 继续参照图 41,在移动设备 4108 上运行的移动钱包 4106 可以建立与发行银行 4112 的安全信道。在授权时,发行银行 4112 可以生成标记并将其映射到存储在发行银行 4112 处的私人信息。标记然后将被发送到在移动设备 4108 上运行的移动钱包 4106,并且在移动设备 4108 上运行的移动钱包 4106 将使用标记来在销售点设备 4104 处完成交易。在移动设备 4108 上运行的移动钱包 4106 可以在个性化移动钱包 4106 时生成并存储标记栈,并且使用这些标记来在销售点设备 4104 处完成交易。在移动设备 4108 上运行的移动钱包 4106 可以在交易时生成并使用标记来在销售点设备 4104 处完成交易。

[0395] 参照图 42,图示了根据示例性且非限制性实施例的在用于安全商务和支付交易的

动态生态系统 4202 中的交易内容的动态标记化的方法。销售点设备 4204 可以连接到在移动设备 4208 上运行的移动钱包 4206。多个移动钱包可以在移动设备上运行。销售点设备 4204 可以通过通信链路 4210 来与在移动设备 C408 上运行的移动钱包 4206 通信。通信链路 4210 可以是有线通信链路、无线通信链路、NFC 链路、蓝牙链路、局域网链路、广域网链路等等。可视信息交换可以包括用相机接收图像和 / 或在电子显示器上显示图像。

[0396] 继续参照图 42，销售点设备 4204 可以从在移动设备 4208 上运行的移动钱包 4206 接收动态标记。动态标记可以是在个性化移动钱包 4106 时、在交易时等等由运行在移动设备 4108 上的移动钱包 4106 生成的动态标记之一。然后，销售点设备 4204 可以向发行银行 4212 发送动态标记以供验证。然后，发行银行 4212 可以经由空中接口 4214 通过验证标记和发行银行 4212 所知道的移动链路 ID 来验证标记，所述移动链路 ID 与生成该标记的移动钱包 4206 所驻留在的移动设备 4208 相关联。在验证了标记和移动链路 ID 时，发行银行 4212 于是授权销售点设备 4204 上的支付和移动钱包 4206 上的结算。

[0397] 本文描述的多域生态系统安全个性化交易的方法和系统包括标记移动缓存的方法，其包括用联网的服务器生成用于通过公共网络与客户端设备执行电子交易的标记。该方法此外可以包括通过公共网络向客户端设备提供标记。该方法此外可以包括在服务器可访问的存储器中存储标记，连同唯一地标识标记被提供给的客户端的信息。该方法此外可以包括将所存储的唯一地标识标记被提供给的客户端的信息与用标记接收的客户端标识信息进行比较以确定进行电子交易是否应当被认证。该比较可以在服务器处响应于从客户端设备接收电子消息而执行，以通过公共网络提供使用标记来进行电子交易。在示例中，电子消息可以包括从与标记被提供到的客户端相关联的安全元件得到的信息。

[0398] 为了增强移动交易安全性，如本文描述的标记化可以促进减少在设备之中的机密信息交换。然而，即便是标记也可能被第三方捕获并尝试用于未授权的交易。因此，对与标记化相关的移动信息的缓存可以确保标记是从其被递送到的移动设备接收的。这可以通过以下来实现：通过标记发行者（例如，发行银行）与 POS/ 服务提供者之间的安全获取信道共享标记和移动标识信息，使得当移动设备提供标记时，如果移动设备标识信息不匹配标记，则可以拒绝交易。同样地，对与标记相关联的移动相关信息的缓存可以保留在诸如发行银行之类的支付促进者处。该信息可以与来自 POS 设备的标记化的获取交易相比较。在示例中，移动设备和银行可以通过 IP 网络交换信息，使得将标记提供给移动设备以用于进行一个或多个 POS 交易。移动设备可以将标记连同移动设备标识信息一起提交给 POS 设备。该组信息可以可选地进一步被标记化和 / 或作为集合发送给发行银行（例如，通过安全获取信道）。发行银行可以将所缓存的与从 POS 设备接收的标记相关联的移动相关信息与从 POS 接收的移动设备信息相比较，并且只有信息顺利地比得上才授权交易。

[0399] 本文描述的多域生态系统安全个性化交易的方法和系统包括通过多域生态系统在安全个性化交易中将封装器标记化的方法。该方法可以包括在多域生态系统中的联网设备处接收安全个性化交易中的电子交易消息，其包括多个信息字段。该方法此外可以包括用设备来确定多个信息字段中每一个的类型和内容。该方法此外可以包括将多个信息字段的至少一部分映射到公共格式消息字段。该方法此外可以包括基于映射来配置公共格式交易消息。该方法此外可以包括在协议抽象内容封装器中封装公共格式交易。该方法此外可以包括配置包括协议抽象内容封装器的标记。该方法此外可以包括将经封装的公共格式交

易消息从联网设备传输到多域生态系统中的第二联网设备。

[0400] 参照图 43，图示了根据示例性且非限制性实施例的在用于安全商务和支付交易的动态生态系统中生成用于穿过网络的安全内容的电子交易封装器的标记的方法。用于特定生态系统的协议 4302 可以用于在用于安全商务和支付交易的动态生态系统内存储和传送协议私有信息元素 4304。协议私有信息元素可以包括信用卡有效期、主账户号(PAN)、卡持有者姓名、卡持有者地址等等。协议 4302 可以以特定顺序或配置来组织协议私有信息元素 4304。例如，有效期可以保持在第一位置中、主账户号在第二位置中、卡持有者姓名在第三位置中，并且卡持有者地址在第四位置中。协议 4302 可以以特定配置来存储协议私有信息元素 4304。例如，协议 4302 可以以 MM/YYYY 格式存储信用卡有效期。

[0401] 继续参照图 43，封装器 4306 可以用于在用于安全商务和支付交易的动态生态系统之中存储和传送用于多个生态系统的封装器私有信息元素 4308。封装器私有信息元素可以包括信用卡有效期、主账户号(PAN)、卡持有者姓名、卡持有者地址、账户余额、信用限制等等。封装器 4306 可以以特定顺序或配置来组织封装器私有信息元素 4308。例如，主账户号可以保持在第一位置中、有效期在第二位置中、卡持有者姓名在第三位置中、卡持有者地址在第四位置中、账户余额在第五位置中，以及信用限制在第六位置中。封装器 4302 可以以特定配置来存储协议私有信息元素 4306。例如，封装器 4306 可以以 YYYY/MM 格式存储信用卡有效期。

[0402] 继续参照图 43，转换(translation)过程 4310 可以用于将协议私有信息元素 4304 映射到封装器私有信息元素 4308。例如，转换过程 4314 可以从协议 4302 的第一位置取得有效期协议私有信息元素，并将其放置在封装器 4306 的第二位置中，并转换私有信息元素的配置 MM/YYYY 以匹配封装器信息元素 YYYY/MM 的配置。

[0403] 继续参照图 43，标记化过程 4314 可以用于用标记 4312 来替代封装器私有信息元素。标记 4312 可以由以下各项一致 :数字字符、字母字符、字母和数字字符的组合、其中字母和数字字符替代主账户号的中间数位的截取的主账户号等等。标记 4312 可以是一次性使用标记 4312。标记 4312 可以由用于安全商务和支付交易的动态生态系统中的可信实体发行。可信实体可以是位于支付网关、获取者或发行者处的服务器。标记 4312 可以生成为数学可逆加密函数、单向不可逆加密函数，或者通过索引函数、序列号或随机生成的号码来指派。

[0404] 本文描述的多域生态系统安全个性化交易的方法和系统包括通过多域生态系统在安全个性化交易中将封装器标记化的方法。该方法可以包括在多域生态系统中的联网设备处接收用于安全个性化交易中的电子交易消息的封装器。该方法此外可以包括在联网设备处配置包括封装器和交易标识信息的标记。该方法此外可以包括将标记电子地传输到用于电子交易消息的封装器的源，以及传输到布置于多域生态系统中的服务提供者服务器。该方法此外可以包括将标记的认证连同标记标识信息一起从用于电子交易消息的封装器的源传输到服务提供者服务器以用于认证由标记标识的交易。在示例中，服务提供者可以是器件发行者。在示例中，服务提供者可以访问联网设备不可访问的机密信息，以促进认证由标记标识的交易。在示例中，机密信息可以与数字钱包相关联，该数字钱包布置于从其传输标记认证的设备上。在示例中，传输标记认证可以由在用于电子交易消息的封装器源自的设备上执行的提供者特定的微件来促进。

[0405] 参照图 44，图示了根据示例性且非限制性实施例的在用于安全商务和支付交易的动态生态系统 4402 中对交易封装器进行标记化的方法。销售点设备 4404 可以连接到在移动设备 4408 上运行的移动钱包 4406。在示例性且非限制性实施例中，多个移动钱包可以在移动设备上运行。销售点设备 4404 可以通过通信链路 4410 与在移动设备 4408 上运行的移动钱包 4406 通信。通信链路 4410 可以是有线通信链路、无线通信链路、NFC 链路、蓝牙链路、局域网链路、广域网链路等等。

[0406] 继续参照图 44，在移动设备 4408 上运行的移动钱包 4406 可以建立与发行银行 4412 的安全信道。安全信道 4414 可以直接建立在运行在移动设备 4408 上的移动钱包 4406 之间，或者通过空中接口 4414（诸如通过服务提供者 4416 和网络 4418）来建立。在授权时，发行银行 4412 可以生成标记并将其映射到存储在发行银行 4412 处的封装器信息。标记然后将被发送到在移动设备 4108 上运行的移动钱包 4106，并且在移动设备 4408 上运行的移动钱包 4106 将使用标记来在销售点设备 4404 处完成交易。在移动设备 4408 上运行的移动钱包 4406 可以在个性化移动钱包 4406 时生成并存储标记栈，并且使用这些标记来在销售点设备 4404 处完成交易。在移动设备 4408 上运行的移动钱包 4406 可以在交易时生成并使用标记来在销售点设备 4404 处完成交易。

[0407] 本文描述的多域生态系统安全个性化交易的方法和系统包括用于在用于安全个性化交易的多域生态系统中执行标记化的电子交易的方法。该方法包括生成表示电子交易消息的标记，该电子交易消息标识至少一个交易和交易中的至少一个生态系统参与者。该方法此外可以包括封装表示电子交易消息的标记，并提供可以包括表示电子交易消息的标记的封装器。该方法此外可以包括生成表示封装器的标记，并提供用于电子消息的标记化表示的标记化封装器。该方法此外可以包括通过在两个参与者之间传输标记化的封装器来执行生态系统中两个参与者之间的电子交易。在示例中，生态系统中两个参与者之一可以是至少一个生态系统参与者。

[0408] 参照图 45，图示了根据示例性且非限制性实施例的通过传输动态生成的标记化封装器来执行跨开放网络的电子交易的方法，所述封装器包括表示用于安全商务和支付交易的动态生态系统 4502 中的交易内容的标记。销售点设备 4504 可以连接到在移动设备 4508 上运行的移动钱包 4506。多个移动钱包可以在移动设备上运行。销售点设备 4504 可以通过通信链路 4510 与在移动设备 4508 上运行的移动钱包 4506 通信。通信链路 4510 可以是有线通信链路、无线通信链路、NFC 链路、蓝牙链路、局域网链路、广域网链路等等。

[0409] 继续参照图 45，销售点设备 4504 可以从在移动设备 4508 上运行的移动钱包 4506 接收标记化的信息，在标记化的封装器中封装标记化的信息，以及将封装器信息发送到移动设备，该移动设备可以将包括标记化内容的标记化封装器传送到与移动钱包相关联的发行银行 4512 以供授权。移动设备 4508 可以直接通过空中接口 4514 或者通过服务提供者 4516 和网络 4518 将具有标记化内容的标记化封装器传输到发行银行。标记化的信息和 / 或标记化的封装器可以由以下各项一致 :数字字符、字母字符、字母和数字字符的组合、其中字母和数字字符替代主账户号的中间数位的截取的主账户号等等。标记可以是一次性使用标记。标记可以由用于安全商务和支付交易的动态生态系统中的可信实体发行。可信实体可以是位于支付网关、获取者或发行者处的服务器。标记可以生成为数学可逆加密函数、单向不可逆加密函数，或者通过索引函数、序列号或随机生成的号码来指派。

[0410] 在包括多个获取方法与多个发行银行的交易流中的协议抽象的示例中，商家可能希望利用来自多个发行银行的多个支付信息获取设备，诸如近场通信(NFC)、条形码读取器和蓝牙，以便为商家的客户提供支付方法上的选择。在该示例中，将会在商家的销售点终端处使用协议抽象，以便通过将用于每一个获取设备的单独获取协议抽象成由商家的销售点终端理解的单个协议来允许多个支付获取设备经由诸如网际协议之类的单个通信协议和链路与单个销售点终端通信，所述单个协议然后可以通过一个或多个通信协议和链路而路由到多个发行银行。

[0411] 在包括多个获取方法与单个发行银行的交易流中的协议抽象的示例中，参照图46，商家可能希望利用来自单个发行银行 4602 的多个支付信息获取设备，诸如近场通信(NFC)4604、条形码读取器 4606 和蓝牙 4608，以便为商家的客户提供支付方法上的选择。在该示例中，将会在商家的销售点终端 4610 处使用协议抽象，以便通过将用于每一个获取设备的多个获取协议抽象成由商家的销售点终端理解的单个协议来允许多个支付获取设备经由多个协议和链路 4612 与单个销售点终端通信，所述单个协议然后可以通过单个通信协议和链路 4614 而路由到发行银行。

[0412] 在包括单个获取方法与多个发行银行的交易流中的协议抽象的示例中，商家可能希望利用单个支付信息获取设备从多个发行银行获取支付信息，诸如近场通信(NFC)、条形码读取器和蓝牙，以便为商家的客户提供支付方法上的选择。在该示例中，将会从商家的销售点终端使用协议抽象，以便允许单个支付获取设备经由诸如网际协议之类的单个通信协议和链路与单个销售点终端通信，然后将用于单个获取设备的单独获取协议抽象成用于多个发行银行的多个通信协议，从而允许适当的交易通过多个通信协议和链路而路由到正确的发行银行。

[0413] 在包括多个获取方法和多个发行银行的空中交易流中的协议抽象的示例中，多个发行银行可能希望向利用诸如近场通信(NFC)、条形码读取器和蓝牙之类的多个支付信息获取设备的商家发送授权信息，以便为商家的客户提供支付方法上的选择。在该示例中，将会在商家的销售点终端处使用协议抽象，以便通过将用于每一个发行银行的单独通信协议抽象成由商家的销售点终端理解的单个协议来允许多个发行银行经由诸如网际协议之类的单个通信协议和链路与单个销售点终端通信，所述单个协议然后可以通过多个通信协议和链路传送到每个获取设备。

[0414] 在包括多个获取方法和单个发行银行的空中交易流中的协议抽象的示例中，单个发行银行可能希望向利用诸如近场通信(NFC)、条形码读取器和蓝牙之类的多个支付信息获取设备的商家发送授权信息，以便为商家的客户提供支付方法上的选择。在该示例中，将会在商家的销售点终端处使用协议抽象，该销售点终端经由诸如网际协议之类的单个通信协议和链路而连接到单个发行银行，并且将用于发行银行的单独通信协议抽象成由每个获取设备理解的多个协议，所述多个协议然后可以通过一个或多个通信协议和链路传送到一个或多个获取设备。

[0415] 在包括单个获取方法和多个发行银行的空中交易流中的协议抽象的示例中(如图47 中所图示)，多个发行银行 4702 可能希望向利用诸如近场通信(NFC)设备 4704 之类的单个支付信息获取设备的商家发送授权信息。在该示例中，将会在商家的销售点终端 4710 处使用协议抽象，以便通过将多个发行银行所使用的多个通信协议和通信链路 4714 抽象成

由商家的销售点终端理解的单个协议来允许多个发行银行经由单个通信协议和链路 4712 来与单个销售点终端通信，所述单个协议然后可以经由单个通信协议和链路 4712 而传送到单个获取设备。

[0416] 在标记化交易、标记化协议抽象和标记化封装器的示例中，参照图 48，用户可以在销售点系统 4806 处向商家的 NFC 读取器 4804 呈现 NFC 使能的智能电话 4802。NFC 使能的智能电话 4802 动态地生成包含用户的信用卡号和有效期的标记，并通过 NFC 读取器 4804 将标记传递到销售点系统 4806。销售点系统 4806 将标记传递到在发行银行 4810 处的移动交易平台服务器 4808。发行银行 4810 处的移动交易平台服务器 4808 通过使用仅驻留在移动交易平台服务器上的密钥来解密标记，对交易进行认证，以及将授权码发送回销售点系统 4806。然后，销售点系统 4806 通过 NFC 读取器 4804 完成与用户的交易。

[0417] 此外，在该示例中，多个发行银行可能希望向利用诸如近场通信（NFC）、条形码读取器和蓝牙之类的多个支付信息获取设备的商家发送授权信息，以便为商家的客户提供支付方法上的选择。在该示例中，可以在商家的销售点终端处使用协议抽象，以便通过将用于每一个发行银行的单独通信协议抽象成由销售点终端理解的单个协议来允许多个发行银行经由诸如网际协议之类的单个通信协议和链路与单个销售点终端通信，所述单个协议然后可以经由多个通信协议和链路而传送到每个获取设备。

[0418] 而且，用户 A 可以向使用协议 X 的商家 NFC 读取器呈现 NFC 使能的智能电话，并且用户 B 可以在销售点系统处向使用协议 Y 的商家条形码读取器呈现在智能电话上显示的条形码。这两个用户的智能电话动态地生成包含相应用户的信用卡号和有效期的标记，并且通过用于用户 A 的 NFC 读取器和用于用户 B 的条形码读取器而将标记传递到销售点系统。销售点系统将由 NFC 和条形码读取器用于与销售点通信的不同协议抽象成单个协议，所述单个协议可以用于经由单个协议和单个通信链路将用于这两个标记的信息传送到发行银行处的移动交易平台服务器。发行银行处的移动交易平台服务器通过使用仅驻留在移动交易服务器上的密钥来解密标记，对交易进行认证，以及将授权码发送回销售点系统。然后，销售点系统将经抽象的协议转换回由 NFC 和条形码读取器使用的单独协议以完成与用户 A 和用户 B 的交易。

[0419] 在可替换示例中，发行银行可能希望向利用诸如近场通信（NFC）、条形码读取器和蓝牙之类的多个支付信息获取设备的商家发送授权信息，以便为商家的客户提供支付方法上的选择。在该示例中，可以在商家的销售点终端处使用标记化的封装器，以便通过将用于每一个发行银行的单独通信协议抽象成由销售点终端理解的单个协议来允许多个发行银行经由诸如网际协议之类的单个通信协议和链路与单个销售点终端通信，所述单个协议然后可以经由多个通信协议和链路传送到每个获取设备。在又一示例中，用户 A 可以向使用协议 X 的商家 NFC 读取器呈现 NFC 使能的智能电话，并且用户 B 可以在销售点系统处向使用协议 Y 的商家条形码读取器呈现在智能电话上显示的条形码。这两个用户的智能电话发送相应用户的信用卡号和有效期，并且通过用于用户 A 的 NFC 读取器和用于用户 B 的条形码读取器而将信用卡号和有效期传递到销售点系统。销售点系统将由 NFC 和条形码读取器用于与销售点系统通信的不同协议封装成单个协议，所述单个协议可以用于经由单个协议和单个通信链路将用于信用卡号和有效期二者的信息传送到发行银行处的移动交易平台服务器。发行银行处的移动交易平台服务器用标记替代信用卡号和有效期，并将标记发送

回销售点系统。然后，销售点系统将经封装的标记转换回由 NFC 和条形码读取器使用的单独协议并将标记传递回 NFC 和条形码读取器。然后，NFC 和条形码读取器用标记替代信用卡号和有效期。

[0420] 本文描述的多域生态系统安全个性化交易的方法和系统包括移动交易处理系统，其包括部署在网络中的多个服务器和多个客户端设备。每个服务器可以被配置为促进网络中的多个客户端设备和多个服务提供者服务器之中的安全个性化交易。此外，多个客户端设备中的至少一个可以配置有在基于容器的执行环境中可操作的可容纳多租户(multi-tenant capable)的钱包。钱包可以是通过微件可访问的，该微件可以布置于多个客户端设备的至少一个上。另外，多个服务器可以此外被配置为促进在通过微件的多个客户端设备中至少一个与通过多个信任模型中任一个的多个服务提供者服务器中至少一个之间的交易，所述信任模型包括单个信任域、单个信任集群、多个信任集群以及直接信任关系。在示例中，多个不同信任模型可以包括单个信任域、单个信任集群、多个信任集群以及直接信任关系中的至少一个。在示例中，单个信任域可以包括中央钱包服务中心服务器，其可以与用户设备上的电子钱包相接口。中央钱包服务中心服务器可以促进建立与可信服务提供者的安全交易信道。在示例中，单个信任集群可以包括可信第三方服务器和钱包服务中心服务器，所述可信第三方服务器用于与个人电子钱包安全地交换机密钱包信息，所述钱包服务中心服务器用于促进由多个服务提供者中的每一个经由第三方服务器对用户的直接认证。在示例中，钱包服务中心服务器可以向多个服务提供者中的至少一个递送用户标识和安全性密钥。在示例中，多个信任集群可以包括多个不同的可信第三方。每个第三方可以至少配置有用于以下的服务器：与个人钱包交换机密钱包信息，并且此外与多个集群特定的服务提供者交换由钱包服务中心提供的用户和 / 或设备标识。

[0421] 在示例中，直接信任关系可以包括归属钱包服务中心服务器，用于建立与电子钱包的交易上下文。归属钱包服务中心服务器可以促进直接在电子钱包与服务提供者之间建立经认证的交易信道。在示例中，微件可以被配置用于与特定提供者的安全交互。在示例中，微件可以被配置用于通过中介与特定提供者的安全交互。在示例中，微件可以在空中通过移动设备与相关联于特定提供者的联网服务器之间的电子交互而配置在移动设备上。

[0422] 如本文所描述的移动交易平台可以促进经由多个不同信任模型在生态系统中进行安全个性化交易。在生态系统中的客户端设备与生态系统中的服务提供者服务器等之间的每个交易可以基于与交易、生态系统配置等相关的各种因素经由不同信任模型来进行。同样地，生态系统中的服务提供者或其他参与者可以确定通过其来进行与服务提供者的交易的优选或默认信任模型。通过提供对交易特定的信任模型确定的支持，平台可以促进可能受多个因素影响的动态信任模型确定和交易使用，所述多个因素包括交易因素、生态系统实例化因素、服务提供者偏好因素、移动网络提供者偏好等。

[0423] 信任模型可以包括单个信任域模型、单个信任集群模型、多个信任集群模型、直接信任关系等。单个信任域模型由移动设备与移动交易平台服务器(本文中也称为钱包服务中心(WSC)服务器)之间的一对一关系来表征。然而，WSC 经历与隐式可信服务提供者的一对多关系。设备钱包和 WSC 为依赖于 WSC 的服务提供者操控用户认证、支付认证等，以确保交易和终端用户完整性。图 49 描绘了单个信任域模型的实施例。设备钱包 4902 以一对一关系直接与 WSC 服务器 4904 通信，如上文所描述的。多个可信服务提供者服务器 4908a 和

4908b 中的每一个通过网络与 WSC 服务器 4904 通信。WSC 服务器 4904 可以执行所有钱包和用户相关认证、验证、支付批准等，而服务提供者 4908 出于该目的而隐式地信任 WSC 服务器 4904。钱包 4902 的用户可以通过 WSC 服务器 4904 来与各种服务提供者 4908 中任一个进行交易，而无需与服务提供者共享机密或个人可识别的信息。

[0424] 信任模型还可以包括如图 50 中所描绘的单个信任集群模型，其假设可信第三方(TTP)5002 在生态系统中是可访问的。服务提供者 4908 既不经由设备 4902 维持与终端用户的直接信任关系，也不维持与 WSC 4904 的隐式信任关系。替代地，WSC 4904 执行用户 / 移动设备 4902 的认证，并且可以经由移动设备 4902 可以显式提供给 TTP 5002 的票据或标记来共享这样的认证，在 TTP 5002 中，可以完成终端用户 4902 与服务提供者 4908 之间的交易。MTP 服务器 4904 维持与 TTP 5002 的强关系，并且充当用户注册实体。例如，在 PKI 认证中，MTP 服务器 4904 充当 RA (注册管理机构)，而 CA (证书管理机构)被认为是 TTP 5002。如所注意到的，服务提供者不一定信任 MTP 服务器 4904 用于认证；然而，MTP 服务器 104 被信任为标识提供者(IdP)。每个服务提供者 4908 可以通过使用由 MTP 服务器 4904 转发的证书来认证由该 ID 标识的用户。经由钱包 4902 与 TTP 5002 之间的显式信任关系来进行交易，其中，钱包和 TTP 可以安全地彼此标识和认证。钱包的证书不需要与 MTP 服务器共享。

[0425] 信任模型可以包括如图 51 中所描绘的多个信任集群模型，其可以包括用于多个服务提供者集群 5102 中每一个的不同 TTP 5002。每个集群特定的 TTP 可以与钱包 4902 建立直接信任。MTP 服务器 4904 执行用户注册、身份提供等，如关于图 50 所描绘和描述的。

[0426] 信任模型还可以包括如图 52 中所描绘的直接信任关系，其中，没有两个实体被强制彼此信任。每个服务提供者 5204 维持与用户 / 客户端 4902 的直接证书，并且直接认证。

[0427] 该模型的主要特性包括：钱包 / 用户 4902 分离地维护用于每个服务提供者 5204 的证书。对于每个交易 / 服务，钱包必须由服务履行中涉及的每个实体所认证。MTP 服务器不控制交易的所有方面；相反，在每个服务提供者之间共享交易上下文。钱包可以使用与服务提供者的任何类型的认证，只要其满足服务提供者的安全性 / 风险要求即可。在示例中，对于账单发行者，认证可以如用户的邮编和母亲的婚前姓氏一样简单。对于支付服务提供者，这可以是信用卡 CVV 或 VbV/ 安全码口令。可以在直接信任关系模型中并入其它认证方法，无限制地包括用户、设备和域的三维认证。

[0428] 用多个信任模型进行交易处理的方法和系统可以包括提供能够在多个不同信任模型的情况下操作的用于安全个性化交易的多域生态系统平台。在这样的方法和系统中，移动交易处理系统可以配置有部署在网络中的多个服务器和多个客户端设备。每个服务器可以被配置为促进网络中的多个客户端设备和多个服务提供者服务器之中的安全个性化交易。所述系统可以包括多个客户端设备中的至少一个，所述客户端设备配置有在基于容器的执行环境中可操作的可容纳多租户的钱包，其中，钱包是通过微件可访问的，所述微件布置于多个客户端设备的至少一个上。系统中的多个服务器可以此外被配置为促进在经由微件的多个客户端设备中至少一个与经由多个信任模型中任一个的多个服务提供者服务器中至少一个之间的交易，所述信任模型包括单个信任域、单个信任集群、多个信任集群以及直接信任关系。单个信任域可以包括中央钱包服务中心服务器，其与用户设备上的电子钱包相接口，并且促进建立与可信服务提供者的安全交易信道。单个信任集群可以包括可信第三方服务器和钱包服务中心服务器，所述可信第三方服务器用于与个人电子钱包安全

地交换机密钱包信息，所述钱包服务中心服务器用于促进由多个服务提供者中每一个经由第三方服务器对用户的直接认证，其中，钱包服务中心服务器向多个服务提供者中至少一个递送用户标识和安全性密钥。多个信任集群可以包括多个不同的可信第三方，所述可信第三方各自至少配置有用于以下的服务器：与个人钱包交换机密钱包信息，并且此外与多个集群特定的服务提供者交换由钱包服务中心提供的用户和/或设备标识。同样，直接信任关系包括用于建立与电子钱包的交易上下文的归属钱包服务中心服务器，其促进直接在电子钱包与服务提供者之间建立经认证的交易信道。布置于(多个)客户端设备上的微件可以被配置用于与特定服务提供者的安全交互，或通过中介与特定提供者的安全交互。微件此外可以在空中通过在移动设备与相关联于特定提供者的联网服务器之间的电子交互而配置在移动设备上。

[0429] 本文描述的多域生态系统安全个性化交易的方法和系统包括在移动设备上操作的安全移动钱包。安全移动钱包可以包括多个不同的用户账户。在示例中，可以由提供者特定的信任模型通过布置于移动设备上的微件来促进对不同用户账户的访问。在示例中，微件可以被配置用于与特定提供者的安全交互。在示例中，微件可以被配置用于通过中介与特定提供者的安全交互。在示例中，微件可以在空中通过移动设备与相关联于特定提供者的联网服务器之间的电子交互而配置在移动设备上。

[0430] 本文描述的多域生态系统安全个性化交易的方法和系统包括用于安全个性化交易的生态系统中的用于移动交易的信任模型确定方法。该方法可以包括配置部署在网络中的多个服务器以通过多个信任模型中的任一个来促进客户端设备与提供者特定的服务器之间的交易。多个信任模型可以包括诸如单个信任域、单个信任集群、多个信任集群以及直接信任关系。该方法此外可以包括分析用于移动设备与提供者特定的服务器之间的交易的默认信任模型，以确定与交易相关联的风险度量。该方法此外可以包括基于产生的风险度量、通过默认信任模型和来自多个信任模型的另一信任模型之一来进行交易。在示例中，进行交易可以包括经由在移动设备上执行的提供者特定的微件来访问移动设备上的钱包功能。可以基于产生的风险度量来选择提供者特定的微件。在示例中，风险度量可以基于移动设备的位置。在示例中，风险度量可以基于提供者特定的服务器相对于移动设备的位置的位置。在示例中，风险度量可以基于移动设备正通过其进行交易的网络的各方面。

[0431] 生态系统中的安全个性化交易的方法和系统可以包括能够在多个不同信任模型的情况下操作的移动钱包。这样的安全移动钱包可以在移动或其它客户端设备上操作，并且可以包括多个不同的用户账户。可以由提供者特定的信任模型经由布置于移动设备上的微件来促进对不同用户账户的访问。如上文所注意到的，布置于(多个)客户端设备上的微件可以被配置用于与特定服务提供者的安全交互，或者通过中介与特定提供者的安全交互。还如上文所注意到的，微件此外可以在空中通过移动设备与相关联于特定提供者的联网服务器之间的电子交互而配置在移动设备上。

[0432] 如本文所描述的用于提供用于安全个性化交易的生态系统的移动交易处理系统可以促进用于移动交易的信任模型确定。模型确定可以包括配置部署在网络中的多个服务器以经由多个信任模型中任一个来促进客户端设备与提供者特定的服务器之间的交易，所述信任模型包括单个信任域、单个信任集群、多个信任集群以及直接信任关系。模型确定此外可以包括分析用于移动设备与提供者特定的服务器之间的交易的默认信任模型，以确定

与交易相关联的风险度量。基于产生的风险度量，可以经由默认信任模型或从多个信任模型选择的另一信任模型来进行交易，所述信任模型包括单个信任域、单个信任集群、多个信任集群以及直接信任关系。

[0433] 进行交易可以包括经由在移动设备上执行的提供者特定的微件来访问移动设备上的钱包功能。可以基于产生的风险度量来选择提供者特定的微件，所述风险度量可以基于移动设备的位置，或基于提供者特定的服务器相对于移动设备的位置的位置。同样，风险度量可以基于移动设备正通过其进行交易的网络的各方面。

[0434] 本文描述的多域生态系统安全个性化交易的方法和系统包括用于使用在生态系统中的域特定的程序语言。生态系统可以包括器件发行者、服务提供者、获取者以及客户端。域特定的程序语言可以包括可以顺从诸如 XML、CSS 和 Java 脚本的语法。所述语言此外可以包括可以在客户端设备上使能钱包相关功能的对象。在示例中，所使能的钱包相关功能可以是钱包容器、钱包、微件和钱包伴随小程序中的至少一个。

[0435] 促进多域生态系统中的安全个性化交易的移动交易平台方法和系统可以包括多域生态系统的域特定的语言，其支持 XML、CSS、Java 脚本等，以用于创建可以被配置以作为容器的运行时组件而操作的钱包和微件。这样的语言可以包括一个或多个对象、标签、元数据元素等，其可以使能容器、钱包、微件、伴随小程序等。多域生态系统的域特定的语言可以提供域特定的 Java 脚本或 XML 功能，以访问移动交易平台的各种模块、功能和服务。另外，还可以包括用于使能 NFC 操作、访问条形码信息等的功能。

[0436] 通过在标准化软件模式(例如，官方语言)内创建域特定的功能集合，域特定的语言通过标准化模式揭示功能，使得用户和软件相关的架构师可以容易地确定域特定的语言正被模式化(patterned)。该域特定的语言应用于本文描述的生态系统的软件编程和模块配置，用以除其它外尤其简化生成用以对该域采取行动的代码。每个域特定的功能可以表示域特定的语言的独特新颖方面。MTP 的软件开发环境可以允许将域特定的功能集合用作域特定的语言的变量。在示例中，用钱包使能功能来适配域特定的语言可以提供经修改的语言，所述经修改的语言特别适于使得能够创建支持安全、个性化、移动交易服务的服务的功能开发。开发者可以使用 MTP 域特定的语言来基于可以专用于创建安全移动交易的功能集合来创建服务。作为域特定的语言，可以提供有原生工具来检查语言；验证其使用；以及确保加密被适当使用。

[0437] 本文描述的多域生态系统安全个性化交易的方法和系统包括用于服务多域生态系统中的安全个性化交易的平台。该平台可以包括解译器，用于解译用于使能钱包相关功能的域特定的程序语言对象，所述钱包相关功能与多个域中至少一个相关。在示例中，解译器可以是运行时解译器。在示例中，解译器可以是在客户端设备上执行的运行时容器。在示例中，所使能的钱包相关功能可以是钱包容器、钱包、微件和钱包伴随小程序中至少一个。

[0438] 除了 XML、CSS 和 Java 脚本顺从的语言以外，MTP 可以包括域特定的语言解译器，其可以在客户端设备上操作(例如作为运行时客户端的部分)。运行时自身可以是语言解译器，其基于域特定的语言来解释代码并在客户端(例如，移动电话)上执行该功能性。客户端可操作的运行时(例如，容器)可以促进解译域特定的语言元素，其与域相关，所述域诸如本文描述的用于安全个性化交易的多域生态系统。尽管对域特定的语言的描述基于标准语言，所述标准语言是 Java 脚本、XML 和 CSS 顺从的，但是域特定的语言可以基于几乎任何解

译的语言(例如,VB 脚本、visual basic 等)。

[0439] 用于在涉及器件发行者、服务提供者、获取者以及客户端的生态系统中使用的此类域特定的编程语言可以包括顺从 XML、CSS 和 Java 脚本的语法以及在客户端设备上使能钱包相关功能的域特定的对象 / 元素 / 标签。具体地,所使能的钱包相关功能可以是钱包容器、钱包、微件、钱包伴随小程序或其组合。

[0440] 用于服务多域生态系统中的安全个性化交易的平台,诸如 MTP,可以包括解译器,用于解译用于使能钱包相关功能的域特定的程序语言对象,所述钱包相关功能与多个域中至少一个相关。解译器可以是运行时解释器、在客户端设备上执行的运行时容器等。同样,所使能的钱包相关功能可以是钱包容器、钱包、微件和钱包伴随小程序中的至少一个。

[0441] 本文描述的多域生态系统安全个性化交易的方法和系统包括用于保护多域生态系统中的内容的一组工具。该组工具可以包括用于构造移动交易生态系统的软件开发套件。该组工具此外可以包括一组域特定的功能,用于使能微件、钱包、容器和伴随小程序中至少一个的客户端侧功能性。该组工具此外可以包括设备不可知应用编程接口。该组工具此外可以包括设备特定的应用编程接口。该组工具此外可以包括用于促进与客户端设备软件和硬件的互操作的使能层功能。该组工具此外可以包括微件公布模块,用以促进提供者特定的微件以供递送和下载到客户端设备的可用性。

[0442] 如本文所描述的,用于移动交易处理平台(MTP)的软件开发套件(SDK)可以促进保护用于安全个性化交易的多域生态系统中的内容。MTP SDK 可以为服务提供者编程人员等提供用于实现和配置以下各项的工具来保护内容 :交易工作流、用于在客户端设备上使用的服务提供者特定的微件、包括可以通过 API 访问的设备特定的且设备不可知的能力的钱包等。SDK 利用设备上的运行时容器,其进一步促进隔离微件、钱包等,以促进无缝内容安全性。本文中和别处也称为移动客户端的运行时容器还可以包括用户接口框架和工作流引擎,其提供服务执行、动作、应用(例如,微件)执行控制、脚本化以及用于各种通信信道的通信支持。用于保护用于安全个性化交易的多域生态系统中的内容的 MTP 工具可以促进开发过程,开发过程包括定义业务需求,创建用户接口流,开发微件、小程序和基于服务器的应用,构建和测试所开发的元件。

[0443] 本文描述的多域生态系统安全个性化交易的方法和系统包括网络使能的服务器,用于配置用于安全个性化交易的多域生态系统的一组服务提供者工具。网络使能的服务器可以包括移动交易平台服务器。网络使能的服务器此外可以包括在平台服务器上可操作的钱包服务中心开发环境。网络使能的服务器此外可以包括在平台服务器上可操作的钱包服务中心运行时模块。网络使能的服务器此外可以包括软件开发套件,用于构造和维护生态系统的基于服务器和基于客户端的元件。在示例中,生态系统的元件可以包括钱包、微件、伴随小程序以及容器。

[0444] MTP 可以包括用于配置用于安全个性化交易的多域生态系统的一组服务提供者工具的平台。MTP 可以包括网络使能的服务器,用于配置一组服务提供者工具,包括在网络使能的服务器上可操作的钱包服务中心开发环境、在服务器上可操作的钱包服务中心运行时模块,以及软件开发套件,所述套件用于构造和维护生态系统的基于服务器和基于客户端的元件。可以被维护的基于客户端的元件可以包括钱包、微件、伴随小程序、容器等等。

[0445] 移动交易平台的微件配置能力可以包括在生态系统中的客户端设备上可操作的

一个或多个应用。(多个)微件配置应用可以包括初始表现为灰色微件的用户接口。所述配置应用可以包括用户接口,通过该用户接口,微件用户接口图标等可以被置于灰色微件中、在外观和感觉(例如,布局、颜色等)上进行定制,并且保存在客户端设备上以供用作经验证的微件。这样的微件配置应用可以使能微件接口的用户特定的定制。

[0446] 本文描述的多域生态系统安全个性化交易的方法和系统包括用于现场定制微件的一部分的移动应用。该移动应用可以包括用户接口模板,用于从在联网服务器上执行的微件开发过程接收微件功能代码和描述符。该移动应用此外可以包括用于促进通过由微件描述符定义的图标而对微件特征和功能进行访问的用户接口。该移动应用此外可以包括用于配置已下载到设备的微件的算法,用户接口可以基于用户接口中检测到的用户动作而显示在该设备上。在示例中,用户接口模板可以是基于微件描述符而自动可配置的。在示例中,用于现场定制微件的一部分的应用可以在移动设备上操作。在示例中,应用可以在移动客户端容器运行时环境内执行。

[0447] MTP 可以包括用于现场定制微件的一部分的移动应用,其可以包括 :用户接口模板,用于从在联网服务器上执行的微件开发过程接收微件功能代码和描述符 ;用于促进经由微件描述符定义的图标而对微件特征和功能进行访问的用户接口 ;以及用于配置已下载到设备的微件的算法,用户接口基于用户接口中检测到的用户动作而显示在该设备上。应用可以基于微件描述符来自动配置用户接口模板。用于现场定制微件的一部分的应用可以被适配以在移动设备上操作。所述应用可以在移动客户端容器运行时环境内执行。

[0448] 本文描述的多域生态系统安全个性化交易的方法和系统包括基于服务器的应用,其包括一组参考移动交易平台服务。所述应用此外可以包括自动化的向导应用,用于访问该组参考服务并基于用户输入来配置来自该组参考服务的至少一个服务。在示例中,自动化的向导应用可以促进基于用户输入来配置以下各项中的至少一个 :客户端元件分布、发行者环境、结算环境、聚集服务、交易介质支持、对手持机类型的支持、用户体验、商业模型以及调节考虑事项。

[0449] 本文描述的多域生态系统安全个性化交易的方法和系统包括生态系统配置引导系统,其包括一组参考移动交易平台服务。该生态系统配置引导系统此外可以包括自动化的向导应用,用于接收与期望的生态系统配置相关联的用户输入。自动化的向导应用还可以可操作用于基于用户输入来推荐来自该组参考服务的至少一个服务的配置。在示例中,该组参考服务可以包括诸如分布服务、连续体使用服务、发行者服务、结算服务、聚集服务、交易介质要求、手持机和操作系统考虑、用户体验因素、商业建模以及调节考虑。在示例中,与期望的生态系统配置相关联的用户输入可以是超链接。在示例中,向导可以基于超链接来搜索信息,并且可以填充至少一个参考服务模板。在示例中,超链接可以是去往操作该向导的服务提供者的网站的链接。

[0450] 移动交易平台此外可以包括向导,用以促进配置用于安全个性化交易的多域生态系统的多个方面。该向导可以是自动化应用等,其访问一组参考移动交易平台服务并基于用户输入来配置来自该组参考服务的至少一个服务。在示例中,自动化的向导应用促进基于用户输入来配置以下各项中的至少一个 :客户端元件分布、发行者环境、结算环境、聚集服务、交易介质支持、对手持机类型的支持、用户体验、商业模型以及调节考虑。

[0451] 移动交易平台可以可替换地包括生态系统配置引导系统,其可以包括一组参考移

动交易平台服务，以及自动化的向导应用，其用于接收与期望的生态系统配置相关联的用户输入并基于用户输入来推荐来自该组参考服务的至少一个服务的配置。在示例中，该组参考服务可以包括分布服务、连续体使用服务、发行者服务、结算服务、聚集服务、交易介质要求、手持机和操作系统考虑、用户体验因素、商业建模以及调节考虑。

[0452] 该生态系统配置引导系统可以接收超链接作为与期望的生态系统配置相关联的用户输入。向导可以基于超链接来搜索信息，并且可以填充至少一个参考服务模板。

[0453] 在另一示例中，超链接可以是去往操作该向导的服务提供者的网站的链接。

[0454] 在用于安全个性化交易的多域生态系统中的参与者之中的类型－长度可变的数据通信的方法可以包括：在作为多域生态系统中的参与者的生产者服务器处接收纯旧式java对象；将接收到的纯旧式java对象转换成遵照类型长度可变协议的类型长度可变流；以及经由单向消息总线将经转换的对象从生产者传输到多域生态系统中的移动设备参与者。纯旧式java对象可以遵从类型长度可变协议指南。类型长度可变协议可以支持流式传输(streaming)、浮点支持(float support)，以及预先考虑的运行时类的类型信息。

[0455] 本方法可以提供用于安全商务和支付交易的动态生态系统中的移动交易平台的消息总线。消息总线可以使能在移动交易平台(MTP)内的组件之间的辅助数据的单向交换。消息总线可以是信道不可知的，以允许通过多个信道的数据交换。多个信道可以是专有协议信道、短消息服务(SMS)信道、移动网络运营商的推送通知框架等。专有协议信道可以是类型长度可变(TyLV)协议信道。TyLV协议信道可以使移动设备软件能够与交易平台通信。TyLV协议信道可以促进纯旧式java对象(POJO)转换成TyLV协议流以及TyLV协议流转换成POJO的转换。转换过程可以是自动化的转换过程。转换过程可以要求POJO遵从指南。指南可以要求POJO：是公共定义的类，实现标记接口com.csam.wsc.daf.tylv.TyLVBindObject，具有均为非最终、非静态、非暂态且公共的数据成员值，为非负数，传递空值而非实际值来指示可选值，不是匿名类，必须提供默认构造器，提供用于解码操作的类的类型信息等等。TyLV协议还可以要求成员值类型被声明。成员值类型可以是：short(短)及其wrapper Short(封装器短)类，int及其wrapper Integer(封装器整数)类，long(长)及其wrapper Long(封装器长)类，Boolean(布尔)及其wrapper Boolean(封装器布尔)类，java.math.BigInteger，字节数组，java.util.Map及其子类型，java.util.List及其子类型，实现TyLVBindObject的合成的用户定义的类型，表示java修改的UTF编码的java.lang.String，java.util.Date，C-SAM定义的com.csam.wsc.core.util.BitSequence类的类型等等。TyLV协议还可以支持其它功能。其它功能可以包括对预先考虑为该类类型的实际数据的运行时类的类型信息、浮点支持、流式传输的继承支持等等。单向交换可以使能数据对象的发送，而不显式要求网络环程(round-trip)。辅助数据可以不是服务请求的主有效载荷。辅助数据可以由可以驻留在移动应用内的消费者接收和消费。移动应用可以是钱包应用等。辅助数据可以由生产者产生和发送。生产者可以是移动交易平台内的组件。

[0456] 用于安全商务和支付交易的动态生态系统中的移动交易平台的消息总线可以提供消息。消息可以是可以通过通信链路传输的数据原子分组。消息可以是可以以数据形式通过消息总线交换的任何商业对象。消息可以由消息ID和有效载荷构成。消息ID可以是由系统的开发者确定的消息的唯一标识符。有效载荷可以是在生产者与消费者之间交换的实际数据。有效载荷数据的格式可以是字符串。消息可以由信道上的组件发送。消息可以

由接收器组件接收和处理。

[0457] 用于安全商务和支付交易的动态生态系统中的移动交易平台的消息总线可以提供消息信道。消息信道可以是将发送器组件连接到接收器组件的虚拟管道。发送器组件可以将信息写到消息信道，并且接收器组件可以从消息信道读取信息。

[0458] 用于安全商务和支付交易的动态生态系统中的移动交易平台的消息总线可以提供消息类。消息类可以是平台消息或钱包消息。平台消息可以不意味着服务消费，并且可以由运行时消费和应答。开发者可以在 Java 脚本中接收针对消息的通知，并且消息可以被标志为已消费。消息可能不能够从 Java 脚本得到应答。可以忽略从 Java 脚本对平台消息进行应答的尝试。钱包消息可以产生以用于服务消费和从 Java 脚本进行应答。钱包消息可以由服务器产生，并且可以变换成等效的 JSON 格式并传递给设备。

[0459] 用于安全商务和支付交易的动态生态系统中的移动交易平台的消息总线可以提供用于消费者和生产者。消费者可以是负责消费消息的实体或组件。生产者可以是负责产生消息的实体或组件。消费者可以是在移动设备上运行的应用。生产者可以是服务器组件。生产者可以对着消息 ID 进行注册。消息 ID 可以唯一地标识消息。如果数据是由运行时拥有的，则消息的消费者可以是运行时。可以将消息传递到容器应用的中央消息操控功能。当消息由消费者经由消息总线返回给生产者时，生产者可以消费对于由生产者产生的消息的应答。

[0460] 参照图 53，图示了根据示例性且非限制性实施例的用于安全商务和支付交易的动态生态系统中的移动交易平台的消息总线通信架构 5302。当消息 5304 通过信道 5306 到达消息总线 5308 时，可以消费消息。信道可以是一组消息。信道 5306 中的消息 5304 可以是项目，其将被分发给已经用消息总线 5308 针对其进行注册的消费者。消息可以按它们在信道上的出现顺序来被分发。多个消息 5304 可以以批背负（batch piggybacked）而分发给相同响应分组。

[0461] 继续参照图 53，消费策略 5316 可以确定用于项目的分发过程。该过程可以是按消息在信道 5306 上的出现顺序来分发消息 5304。多个消息可以以批背负而争给相同响应分组。消费者 5310 可以当其消费消息 5304 时发送应答以指示消息 5304 的成功消费。应答可以保持指示消息 5304 的成功或未成功消费的值。当移动应用将其下一请求发送到服务器时，应答可以发送回服务器。可以在下一请求期间在下一新应用会话中发送应答。可以由消费者 5310 即时地和实时地发送应答，从而发起专门响应序列以对消息的处理进行应答（804）。

[0462] 继续参照图 53，当要经由消息总线 5308 发送响应时，生产者 5312 可以开始消息产生。经注册的生产者可以产生消息并在输出信道上分发它们。然后，消息 5304 可以被编码，并且然后可以经由消息总线 5308 发送整个列表。基于服务器的生产者可以产生 java 对象，java 对象然后可以由移动交易平台基础设施转换成等效的 JSON 对象。产生策略 5314 可以以生产者注册的次序在输出信道上分发消息。移动交易平台服务器可以提供生产者注册处，其中，开发者注册它们各自的生产者。生产者的注册次序可以是当存在递送消息的机会时调用生产者的次序。

[0463] 根据示例性且非限制性实施例，本公开提供类型映射方案和类型定义规则，用于将任何纯旧式 Java 对象（POJO）转换成类型长度可变（TyLV）流，并且反之亦然。该方案可

以使开发者能够写出基于设备的用例以定义 POJO，POJO 可以在运行时（on the fly）被转换成对应的 TyLV 流。在示例中，TyLV 流可以根据 TyLV 协议来使用，该 TyLV 协议可以使任何基于设备的软件能够与交易提供者通信以完成任务。

[0464] 在实施例中，运行中的流式传输可以帮助开发者抽象出非本质的流式传输细节。这继而可以使得开发者免于理解大量 TyLV 协议相关细节。本发明的类型映射方案和类型定义规则可以基于 TyLV 协议而用在定义基于设备的用例和其它交互中。

[0465] 在示例中，映射和类型定义规则可以定义和使用为指南，诸如用以使得 POJO 被流式传输（streamed）至 TyLV 协议。遵循这些定义的指南集合的 POJO 可以称为 TyLV 顺从的 POJO。在示例中，指南可以包括诸如用以将 POJO 定义为公共定义的类的规则。此外，POJO 可以实现标记接口，诸如 com.xyz.wsc.daf.tylv.TyLVBindObject。在这样的实现失败的情况下，可以生成 ClassCastException（类转换异常）的性质的运行时错误。在各种实施例中，POJO 的数据成员可以是非最终的、非静态的、非暂态的和公共的（而出于流式传输的目的可以忽略所有其它类型的对象）。在实施例中，整数性质的 POJO 成员可以不提供负值，因为未能这样做可能导致运行时异常，其与诸如“解码器不支持负数的长度计算”的消息一起抛出。指南可以包括用以允许 POJO 声明诸如以下各项的成员变量的规则：short（短）及其 wrapper Short（封装器短）类，integer（整数）及其 wrapper Integer（封装器整数）类，long（长）及其 wrapper Long（封装器长）类，Boolean（布尔）及其 wrapper Boolean（封装器布尔）类，java.math.BigInteger，字节数组，java.util.Map 及其子类型，java.util.List 及其子类型，实现 TyLVBindObject 的合成的用户定义的类型，java.lang.String（表示 Java 修改的 UTF 编码），java.util.Date， com.csam.wsc.core.util.BitSequence 类的类型等等。在示例中，诸如不同于定义的那些，诸如不同于上面提及的那些的类型的变量的规范可能导致运行时异常。在诸如其中对象是聚集对象的情况中，该对象可以被声明为公共内部静态或非静态类。

[0466] 在示例中，按照为 Java 类格式所指定的规则，编译器可以生成用于非静态内部类的合成字段。

[0467] 在示例中，诸如包含在 java.util.Map 实例内的键值对可以为类型 java.lang.String，并且覆盖 Object.toString 对于这样的键值对可能不是非常有帮助。

[0468] 在可替换示例中，TyLV 协议实现可以允许布尔或布尔数据类型，但是在内部，其可以将该数据类型编码或解码为 8 比特类型。因此，布尔值真可以看作 1，并且布尔值假可以看作 0。同样，由 TyLV 协议支持的类的类型 com.xyz.wsc.core.util.BitSequence 可以在内部看作整型 -32（integer-32）值。此外，数据类型可以以特定格式来字符串化。另外，不可以允许包含零条目的映射，并且空值可以用来将映射表示为可选值。

[0469] 在示例中，使用 TyLV 协议的系统可以支持由 Java 平台修改的通用字符集或 USC 变换格式（UTF）编码定义，因为 java.lang.String 类型可以是通用字符集或 UCS 变换格式（UTF）编码的。此外，原语类型成员可以不是可选的。

[0470] 在示例中，字节可以是非空的，并且合成类型可以基于它们的具体类型而非基于它们实现的接口来被声明。

[0471] 在示例中，TyLV 协议可以支持同构的、可变长度的、具有相同类型对象的列表。对象类型可以是任何 Java 类的类型或任何用户定义的类类型，并且可以包括封装器类型以

支持原语类型数据。

[0472] 在可替换示例中，与 POJO 相关的附加指南可以包括定义用于任何 POJO 或任何其成员变量的规则，使得传递空值而非实际值可以指示可选值。此外，可以定义规则以使得 TyLV 顺从的 POJO 可以不支持类型继承。同样，将 POJO 或任何其成员声明为匿名类可能导致运行时异常。另外，可能需要提供默认构造器，因为 TyLV 引擎可能不能使用所定义的构造器。

[0473] 在示例中，与 POJO 相关的附加指南可以包括规则使得 TyLV 编解码器可能需要用于解码操作的类类型信息，使得 TyLV 引擎可以正确地标识可以用解码的数据适当填充的 BindObject 接口实现类型或列表内容对象类型。在示例中，类的类型信息可以被提供为逗号隔开的完全限定(fully qualified)类名字符串值列表。在示例中，下面在此处包括了类的类型信息。

```
Public class Parent implements BindObject {

Public BindObject child1;

Public int child2;

Public List child3;
}
```

[0474] 另外，可以以以下方式提供作为动作配置参数的类类型声明的示例：

```
com.xyz.wsc.comms.Parent, com.xyz.wsc.comms.
CertificateResponse, com.xyz.wsc.comms.Certificate
```

在以上示例中，父类可以类似于可以由 BindObject 类型子级(child)1、int 类型子级 2 和列表(List)类型子级 3 构成的父类类型。同样，在类类型声明示例中，第二类类型声明 CertificateResponse (证书响应)可以用于标识子级 1 的运行时类型信息。类似地，第三类类型声明 Certificate (证书)可以标识可以为列表中收集的对象的类型的子级 3 内容。该列表数据内容可以被提供为像整型、映射那样的任何封装器类型，或者可以是任何其它有效 Java 类型。如果未提供类类型信息，则可能导致运行时异常。

[0475] 在示例中，可以定义关于使用 TyLV 协议来流式传输对象图的附加规则。TyLV 协议中的对象图的流式传输可以类似于 Java 对象串行化协议。然而，在 TyLV 协议中，构成该图的所有对象可以是 TyLV 顺从的。此外，图的根对象可以几乎始终是实现 TyLVBindObject 接口的对象。

[0476] 在示例中，TyLV 顺从的 POJO 可以定义为纯值聚集对象，其可以不包含任何功能性。可以出于 TyLV 流式传输的目的而定义的这些 POJO 可以仅在其中流式传输是主要目标的场景中使用。此外，可以避免将计算放在 POJO 中。

[0477] 在示例中，整型数据类型的值的可能范围可以在定义它之前被标识。作为示例，如果对于变量的值的可能范围是已知的，则可以避免定义长变量用于承载可以用短类型容纳的值。为了实现该目的，可以通过诸如预期可能需要传递的最大值来指派类型。然而，在其它实施例中，诸如在最大值的预期是不可能的情况中，可以相应地选择最有适应性的类型。

此外，可能明智的是关注代码中的下溢和上溢，这可能涉及数值计算。这可以帮助避免将负值指派给绑定对象成员，该指派可能不被引擎支持。

[0478] 在示例中，TyLV 协议可以包括对流式传输的继承支持。此外，当协议改变时，可以使能负值和浮点支持。同样，TyLV 协议可以支持运行时类类型信息到该类类型的实际数据的预先考虑。这可以类似于 Java 中对象的串行化，并且可以使解码引擎能够标识正确的实际绑定对象类型或列表内容对象类型。

[0479] 有效 POJO 的示例可以定义为：

```
Class ValidBindObject implements TyLVBindObject{

    public String member1 ;
    public int member2 ;
public AggregateObject member3 ;
public AnotherAggregateObject member4 ;
public AnotherAnotherAggregateObject member5 ;

public static class AggregateObject  implements TyLVBindObject{
    public int member1 ;
    public String member2 ;
}

public class AnotherAnotherAggregateObject  implements TyLVBindObject{
    public int member1 ;
    public String member2 ;
}
```

无效 POJO 的示例可以定义为：

```
class InvalidBindObject implements TyLVBindObject{ @1
public final String member1 ; &&1
public static int member2 ; &&2
private int member3 ; &&3

public TyLVBindObject invalidAggregateMember4 = new TyLVBindObject() {@2
};

public InvalidBindObject (String value){ @3
    //whatever
}

public TyLVBindObject invalidAggregateMember5; @4
```

在以上示例中，可以诸如通过 @XX 来指示错误点，并且可以诸如通过 && 来指示被忽略点。在示例中，这些错误可以是运行时错误。

[0480] 根据示例性且非限制性实施例，本文档公开了用于移动支付和移动商务生态系统的 MTP 5.x 钱包和微件基础设施模型。MTP 5.x 模型可以在与 MTP 4.x 抽象模型相关联的各种特征上解决和并入客户反馈，MTP 4.x 抽象模型可以涉及交付抽象模型和依赖于中间件来构建电话应用。模型 5.x 可以解决的 MTP 4.x 的示例性特征中的一些可以诸如但不限于客户期望诸如立即并入新电话特征，而不用等待来自中间件提供者的支持；对新专有工具的昂贵培训，这可以包括人力资源（HR）相关关注；从中间件迁移出时所需的大量返工和成本。

[0481] 在示例中，MTP 5.x 模型可以支持开发简单性、降低进行中的维护努力以及支持第三方微件的能力。

[0482] 在示例中，MTP 5.x 客户端可以为开发者提供灵活性，以诸如使用模型能力中的一个或多个。架构可以包括独立库，所述库可以包含基础设施组件，诸如 Comms、NFC、加密等。根据 MTP 5.x 模型，这些组件可以被提取到独立库中。此外，部分或整个移动应用可以整体构建在原生电话操作系统上。这些部分可以包括用户接口组件、商业逻辑，或移动应用的任何其它组件。此外，可以通过诸如电话独立的应用编程接口（API）层次来支持模型中提供的微件。在示例中，微件还可以使用可以由移动应用提供的应用编程接口（API），其可以附加于从运行时可获得的应用编程接口（API）。在示例中，模型架构可以允许客户端运行时被捆绑在移动应用内。在可替换示例中，移动应用可以构建在运行时上。

[0483] 在示例中，MTP 5.x 架构的模块性可以允许用于移动应用的三种类型的部署。这三种部署模型可以包括诸如独立的原生应用模型、混合生态系统应用模型，以及跨平台生

态系统应用模型。可以在本发明的后续描述中更详细地讨论这些模型如下。

[0484] 独立的原生应用模型 OM 1.1 (为了简单,后文中称为模型 1)可以以可能不需要微件的客户为目标。这可以为这样的客户提供独特和定制的用户体验。模型 1 可以包括电话操作系统环境,其包括诸如多个移动库。这些库可以由诸如移动应用所使用,所述移动应用诸如移动钱包客户端应用。移动库可以促进客户利用现有移动库来构建移动应用组件,诸如加密组件或 Comms 组件,而非必须从头开始构建组件。库可以被设计为促进移动交易和钱包相关能力。在示例中,移动库可以耦合到 MTP 服务器,并且可以为移动应用的客户提供完全开箱即用(out of box)的基础设施解决方案。

[0485] 作为模型 1 的独立基础设施支持能力的示例,可以通过使用该模型来构建加密组件,使得其可以提供多于仅仅在原生加密操作上的封装器。该组件可以执行为安全性组件,其可以操控诸如以下的功能 :密钥生成、密钥管理、个人标识号(PIN)管理等等。

[0486] 在示例中,每个库可以通过其自身来并入,而没有任何外部依赖性。移动应用 OM214 的开发者可以集成来自第三方源的最佳的种子库以提供完整的移动应用解决方案 OM214。

[0487] 混合生态系统应用部署模型(为了描述的简单,后文中也称为模型 2)可以以这样的客户为目标 :所述客户可能需要独特、原生的体验,连同通过微件的服务扩展性能力。模型 2 可以包括可以诸如通过电话操作系统来构建的应用编程接口和移动应用。根据模型 2,整个客户端运行时软件可以与移动应用捆绑在一起。电话操作系统可以包括可以由诸如移动钱包客户端应用之类的移动应用使用的多个库。模型 2 还可以包括可以由诸如电话独立的 API 支持的微件。在示例中,模型 2 可以能够不在用户体验、性能或应用功能性上折衷。该模型可能具有更高的开发成本,因为诸如移动钱包之类的移动应用的可变部分可以针对每个电话平台而定制地构建。

[0488] 在示例中,模型 2 的客户端运行时软件可以在原生屏幕与使用可扩展标记语言(XML)通过工具套件所构建的屏幕之间切换。运行时还可以允许集成诸如在 Java 脚本中开发的动作(其可以通过工具套件)以及通过原生实现而开发的动作。以这种方式,在示例性实施例中,移动应用的大部分(诸如百分之八十)可以利用运行时和工具套件相关的开发效率,而剩余部分可以针对可区分的体验而构建成原生的。可以通过使用工具套件,诸如使用 XML、层叠样式表(CSS)以及 Java 脚本,来构建微件。这可以允许开发者有更大灵活性,同时保持与所有电话平台的更新,因为开发者可以绕过运行时来创建通过运行时可能不是可获得的功能性。在示例中,微件可以能够通过运行时来使用开发支持。然而,微件也可以使用移动应用的展露的应用编程接口(API)来用于开发。微件工具套件可以使生态系统服务提供者更容易且更平滑地开发微件。可以增强应用工具以支持混合组装和包装。

[0489] 图 OM1.3 图示了根据 MTP 5.x 架构的跨平台生态系统应用部署模型。跨平台生态系统应用部署模型(为了简单,后文中也称为模型 3)可以是成本有效的模型,其可以以诸如他们的要求可以通过客户端应用运行时的能力来实现的客户为目标。该模型可以包括通过诸如可以包括多个库的电话操作系统所构建的 API 和移动应用。此外,根据跨平台生态系统应用部署模型,多个微件可以由诸如电话独立的 API 支持。此外,根据该模型,客户端应用和微件二者可以通过使用可以支持跨平台开发的工具套件来构建。

[0490] 在示例中,模型 3 可以类似于根据 MTP 架构的 MTP 4.x 版本的部署模型,除了与

MTP 4.x 版本相对地，MTP 5.x 可以促进或支持平台扩展。

[0491] 图 54 图示了根据本发明的示例性实施例的如在服务递送平台生态系统(出于描述简单的目的，在后文中称为 SDP 5408 并稍后讨论)的部署中使用的支付促进器系统 5400 的各种子系统的高层级框图。

[0492] 根据示例性实施例，支付促进器 5400 内的各种子系统可以包括销售点(POS)5402 终端、诸如移动应用 5404 之类的应用(APP)5404、SDP 5408、若干发行者系统，诸如无限制地包括所描绘的发行者系统 1 5410、发行者系统 2 5412 以及发行者系统 3 5414。各种子系统可以通过诸如支付处理网络 5418 之类的网络连接。APP 5404 此外可以包括一个或多个微件。

[0493] 图 54 的支付促进器 5400 可以促进在诸如 POS 5402 之类的物理商家 POS 5402 处利用移动电话来诸如进行商品和服务的支付。因此，移动电话可以用于促进支付，作为对基于塑料的物理交易卡的可替换方案，使得可以通过支付促进器来处理、路由和结算支付交易。因而，可以通过使用支付促进器 5400 来促进商家，因为其可以减少与交易相关联的整体成本。同样，支付促进器 5400 可以有利于客户或消费者，因为其提供有奖励的购物或可以节省金钱的交易体验，这诸如通过供应、忠诚度等促销方案或者由于商家从通过使用支付促进器 5400 的支付过程获得的节省而由商家提供的整体降低的价格。

[0494] 支付促进器 5400 可以向消费者提供具有跨像发行者的若干发行者的多个支付产品的选择。支付促进器 5400 可以向消费者提供与其它交易模式相比时改进的支付体验，并且可以与拔出诸如物理皮革钱包、选择卡以及在机器上刷卡一样简单。同样，与常规塑料卡和支付选项相比时，支付促进器 5400 可以提供具有更好的风险管理的安全平台。通过使用支付促进器 5400 还可以促进商家，诸如关于它们单独的 POS 装备使得若干可用于商家。另外，诸如优惠券、供应、忠诚度和收据的若干促销方案或增值服务可以集成到支付促进器 5400 中，用以为商家以及消费者提供整体零售体验。

[0495] 根据示例性实施例，图 54 从支付视角描绘了支付促进器 5400。然而，当支付促进器 5400 缩放时，支付促进器 5400 的类似模型可以应用于其它非支付器件，诸如优惠券、忠诚卡等。

[0496] 在本发明的实施例中，如图 54 中所示的 APP 5404 可以是移动应用 5404。移动应用 5404 可以包括诸如钱包(也称为虚拟钱包)作为其组件之一。消费者可以与一个或多个这样的钱包相关联。出于由钱包提供的便利性、紧凑性和互操作性的缘故，钱包可以诸如在支付过程期间促进消费者。每个钱包可以被配置为持有消费者的交易器件的子集。根据各种实施例，钱包可以安全地存储和保护消费者支付账户信息、组织和管理消费者支付账户的生命周期、实现如访问账户所要求的强认证，提供用于选择用于交易的特定账户的选项、用所选账户与 POS 5402 或支付网关执行交易，实现并执行商业规则来管控对账户的访问和使用这些账户的交易。根据若干实施例，虚拟钱包的若干形式和实现可以是可能的，这取决于各种参数，诸如无限制地包括托管钱包的实体、消费者通过其来进行交易的信道，以及支付证书所存储的位置等。

[0497] 在示例性场景中，消费者或用户可以创建钱包，诸如在购物商店或例如“merchant.com”的网站上，并且使用诸如 web 门户来添加支付卡。在示例中，这样的钱包可以称为在线钱包，其可以在诸如仅从“merchant.com”购买产品期间提供便利性。必须意

识到，支付促进器 5400 可以促进消费者自己向虚拟钱包添加他们预先存在的支付账户。然而，在诸如商家发行的卡（例如，礼品卡）之类的一些其它情况中，可以自动添加支付账户。

[0498] 在一些示例中，诸如银行或任何其它金融机构（FI）之类的实体可以建立它们自己的虚拟钱包版本。在这样的情况中，该实体可以承担显著的风险级别，并且保障商家不受欺诈，因为它们可以提供诸如持有账户、对用户进行认证、卡选择等钱包功能性，并且是仅仅用于支付账户的发行。在这样的情况中，钱包因此可以限于自发行的器件。

[0499] 钱包和钱包的每种支付类型可以被配置为诸如通过从塑料钱包跨至虚拟钱包映射各种标准和支付定价模型来遵照诸如定义的商业规则、交换费、PCI 顺从性等。在实施例中，钱包可以实现在诸如智能电话中，诸如实现在移动手持机或其它此类手持设备内或使用它们的能力来实现。这可以促进虚拟钱包从在线部署至物理 POS 5402 的扩展。

[0500] 根据本发明的实施例，如上所述，钱包可以是容器或移动应用 5404 的组件或子系统。移动应用 5404（如图 54 中所示）可以包括若干服务和功能，诸如但不限于钱包。例如，可以提供具有作为若干移动服务、一个或多个附加服务或功能之一的钱包的移动应用 5404，诸如以在消费者内产生购买激励。因此，虽然附加服务可以提升购买激励，钱包可以提供对于完成或执行购买所需的功能性，使得钱包的焦点可以在于在诸如物理 POS 5402 之类的 POS 5402 处所执行的购买。

[0501] 图 55 示出了本发明的示例性实施例中的钱包的各种组件。图 55 的钱包可以称为软件钱包 5500。根据各种实施例并且取决于支付促进器和服务递送平台生态系统 5502 的设计，钱包的各种组件可以不同地位于移动电话或任何其它手持设备 5504、SDP 5508 或发行者系统 5510 的任何合适组合中。因此，钱包 5500 可以被构造并视为分布式组件或分布式组件的组合，其可以跨诸如图 55 中示出的多个系统或子系统而运行。如图 55 中所示，钱包可以包括电子仓库（vault）5512、参考存储装置 5514、钱包控制器 5518 以及钱包视图 5520。

[0502] 电子仓库 5512 可以是与每个单独支付证书相关联的安全存储装置。在实施例中，电子仓库 5512 可以合并在安全元件上。在另一实施例中，电子仓库 5512 可以跨若干系统或子系统而分布，使得诸如发行者之类的每个子系统或系统可以保留对支付证书的控制以及在请求时对诸如单个使用标记的释放。在实施例中，服务递送平台生态系统可以采用分布式模型，然而，必须意识到，使用合并的模型可以是可能的并且在本发明的范围内。

[0503] 参考存储装置 5514 可以用作与诸如钱包的内容相关的参考数据的贮存库。参考数据可以包括诸如但不限于原图（artwork）和品牌化、发行者信息、路由信息（和参考标识符）、掩蔽的账户数据、元数据等。在实施例中，这些数据元素的子集可以存储在诸如移动电话上，诸如在移动电话的存储器内，而完整集合或数据可以存储在与 SDP 5508 相关联的服务器上，该服务器称为 SDP 服务器（如稍后所详细讨论的）。

[0504] 钱包控制器 5518 可以被配置为实现诸如钱包特定的商业逻辑和生命周期。钱包控制器 5518 可以被配置为控制和管理钱包，并且向可以访问钱包的外部系统提供诸如 API 的接口。另外，钱包控制器 5518 可以被配置为实现协议诸如用以与 POS 通信。例如，在本发明的服务递送平台生态系统的典型部署中，钱包控制器 5518 可以驻留在 SDP 服务器上。

[0505] 钱包视图 5520 可以是与钱包相关联的用户接口，诸如用以促进由相应用户通过由钱包视图 5520 提供的装置而对钱包的访问、交互、填充和管理。由钱包视图 5520 提供的

用户接口可以针对每个可以用于访问钱包的信道分离地并且优选一致地被创建，所述信道诸如包括但不限于移动电话、web、IPTV 等。在本发明的实施例中，钱包视图 5520 可以被包括在被设计用于和关联于服务递送平台生态系统的容器应用中。在实施例中，钱包视图 5520 可以被包括在通过服务递送平台生态系统所链接的用于消费者的 web 接口中。

[0506] 在示例性实施例中，支付促进器可以部署在移动钱包中。钱包可以部署在诸如基于移动支付的系统中，所述基于移动支付的系统诸如服务递送平台生态系统 5502 或支付促进器。钱包 5500 可以被配置以便促进消费者从云中选择经证明的应用来访问诸如与支付促进器相关联的钱包之类的钱包以执行支付。在实施例中，这些支付可以在 POS 处进行。在另一实施例中，这些支付可以在远程定位位置处进行。

[0507] 钱包架构可以包括诸如钱包视图 5520、钱包控制器 5518、钱包电子仓库 5512、应用、参考存储装置 5514、服务递送平台 5508、诸如发行者系统 5510-5514 之类的多个发行者系统。服务递送平台 5508 此外可以包括主钱包和若干应用钱包，诸如第一应用钱包、第二应用钱包等等。应用可以包括诸如第一应用 1、第二应用以及用于支付促进器的应用。

[0508] 在实施例中，SDP 5508 架构可以包括若干 SDP 5508 和 SDP 钱包，它们间具有诸如分层父子关系。SDP 5508 可以被配置为维护具有多个钱包的父子分层。诸如移动应用之类的每个应用可以链接到独立钱包实例。SDP 5508 可以被配置为维护在诸如父支付促进器钱包与对应于相应父钱包的子应用钱包之间的父子关系。在这样的场景中，该架构可以被设计以便为消费者定义单个 SDP 5508 和单个钱包视图 5520，使得消费者感觉好像在与从不同应用或不同钱包视图 5520 内所使用的 SDP 5508 和单个钱包相交互。

[0509] 在如上所讨论的分层架构中，所有器件可以被供应到父支付促进器钱包，即，供应到支付促进器的父钱包的参考存储装置 5514。实际钱包电子仓库 5512 可以被配置为属于发行者系统 5510 并且在其控制之下。然而，在实施例中，可以将非敏感信息供应到 SDP 5508。子钱包可以被配置为具有它们自己的参考存储装置 5514，以诸如使能在支付促进器的父钱包的参考存储装置 5514 上的经过滤的视图。

[0510] 在实施例中，诸如 PIN 长度、所允许的 PIN 重试等的各种钱包生命周期规则在诸如由支付促进器设置的指南内针对每个钱包可以是独立可配置的。在支付时，移动应用可以从 SDP 5508 请求标记。SDP 5508 可以被配置为使用企业服务总线（ESB）来检索用于支付帐户的标记。标记可以是诸如由发行者系统生成的一次性标记。对标记检索和生成的请求可以与通过 SDP 5508 进行的强认证过程以及对发行者系统可用的可选附加认证相关联。然后，可以在 POS 处通过可用信道之一来使用标记。在实施例中，单独移动应用可以通过标准 web 服务来传送来自 SDP 5508 的消费者服务以便访问对应钱包。

[0511] 根据钱包架构的一些示例性且非限制性实施例，在安全防护支付证书和发行标记中可以牵涉发行者。可以允许发行者控制标记的格式，并且可以包括如适当和相关的附加安全性措施。此外，发行者可以对退款和其它异常场景具有最小影响，因为发行者可以直接牵涉在供应和支付流中。该钱包架构还可以提供从多个移动应用（诸如从多个钱包视图 5520）的钱包可访问性。同样，在认证期间，每个钱包可以用诸如独立口令被独立认证，并且可以被独立地管理或激活或锁定或挂起等，同时仍然向用户提供一致的用户体验。该钱包架构还可以使能可配置的基于规则的过滤，以用于为每个子钱包过滤参考存储装置 5514，从而允许商家具有所有用户的支付卡的获允许子集。同样，该架构可以促进默认帐户容易

地按照相应钱包被分别设置。该钱包架构还可以使能用消费者的主钱包中的支付证书来自动填充支付促进器的消费者有品牌的容器。这可以促进从单个位置对所有帐户的管理。此外，通过该基于单个位置的管理，用户还可以标识他们的用于访问支付促进器的他们的钱包的其它应用。

[0512] 如上文所讨论和提到并且在图 54 中示出的 SDP 5508 可以被配置为这样的系统：其可以被配置为包装服务并且通过若干信道将服务递送给消费者。SDP 5508 可以促进有组织的、易于消费方式的服务包装以及通过多个信道的服务递送。可以诸如在移动支付域中使能 SDP 5508。SDP 5508 可以被配置为执行诸如用户钱包实例的登记、了解你的客户(KYC)以及创建。SDP 5508 可以被配置为管理钱包的生命周期，连同去往各种端点设备的链接。SDP 5508 可以被配置为诸如向消费者提供标准化接口以用于服务发现和选择。SDP 5508 可以被配置为诸如以统一方式提供经发行者管控的服务登记和供应。SDP 5508 可以被配置为诸如围绕服务供应和消费而实现强认证、安全性和信任模型。SDP 5508 可以被配置为针对诸如服务和钱包事件等维护详细的审查跟踪、日志和交易记录。SDP 5508 可以被配置为机载(onboard)单独服务提供者、聚集不同的服务主机、执行服务实现标准以及管理生态系统的整体成长和执行若干其它功能。这里讨论的 SDP 5508 的功能和配置仅仅是示例性的而没有限制。在仍一些实施例中，SDP 5508 可以被配置为基于诸如由 SDP 5508 递送的服务的性质来执行若干其它功能和任务。例如，与金融交易相关的服务可以具有对于安全性和可靠性的额外强制性要求，使得 SDP 5508 可以被配置为在递送金融服务的同时增强安全性相关的任务。此外，SDP 5508 可以被配置为在大规模向公众递送服务时进一步增强系统可伸缩性和冗余性。

[0513] 在基于本文描述的 MTP 方法和系统的 SDP 5508 生态系统部署的示例性实施例中，SDP 5508 可以被设计为包括专用于支付促进器的各种组件，其中所述生态系统部署包括钱包、容器、MTP 服务器、服务提供者等等。在示例中，可以存在若干组件一起工作来履行 SDP 服务器功能。这些组件均可以以高度安全、高度可伸缩、高度可用性的企业服务器配置来部署。在示例中，ESB 可以标准化和互连多个合作伙伴和系统，从所述合作伙伴和系统可以消费服务和将服务递送给移动容器。每个合作伙伴服务提供者可以通过在 SDP 5508 中建模的一系列步骤被机载。这些步骤可以由支付促进器来定义，并且通过服务提供的性质(例如，支付、供应、忠诚等)来管控。

[0514] 本文可以描述支付促进器 SDP 5508 的各种示例性组件。在示例中，支付促进器 SDP 5508 可以包括通信组件。通信组件可以处理通信协议、编解码、串行化、包封(enveloping)，以及通过多个数据信道与移动应用进行分组交换通信的复杂性。在示例中，支付促进器 SDP 5508 可以包括消息总线 & 通知组件。消息总线 & 通知组件可以支持基于生产 / 消费者的模型，用于与移动客户端进行对象连同消息结构的交换。在示例中，消息总线 & 通知组件还可以促进推送通知、警报等，从而允许 SDP 5508 主动地到达容器应用以执行某活动。在示例中，支付促进器 SDP 5508 可以包括安全性 / 认证组件。安全性组件可以促进用于强认证、数据保密、机密性 / 保密、数据完整性等的安全性相关的设计的实现，因为它们涉及整体信任模型以及对正递送的服务的风险管理。在示例中，支付促进器 SDP 5508 可以包括空中(OTA)供应组件。OTA 供应可以被配置用于处理应用、微件、器件(支付、供应、忠诚度、票据等)以及代理服务的空中供应和递送以用于器件的标记化。在示例中，支

付促进器 SPD 5508 可以包括 OTA TSM 代理组件。OTA 可信服务管理器(TSM)代理组件可以仅在基于安全元件(诸如图 55 中所图示的)生态系统中被需要,以用于小程序的空中供应和管理安全元件安全性域。在示例中,OTA TSM 代理可以典型地与外部第三方 TSM 相接口。在示例中,支付促进器 SDP 5508 可以包括钱包和微件管理组件。钱包和微件管理可以创建和管理消费者的钱包的生命周期、钱包与一个或多个设备的关联,以及用一个或多个微件和那些微件的生命周期来使应用个性化。在示例中,支付促进器 SDP 5508 可以包括交易引擎。交易引擎可以被配置用于处理跨无线信道的交易可靠性,并维护正被递送的服务的完整审查跟踪。交易引擎可以促进收到和其它相关联的交易服务。在示例中,支付促进器 SDP 5508 可以包括器件管理组件。器件管理组件可以被配置用于管理消费者在其钱包中所填充的单独器件的生命周期。在示例中,支付促进器 SDP 5508 可以包括位置和地理围栏组件。位置和地理围栏组件可以帮助服务提供者在基于位置的服务中设立地理围栏和选择。在示例中,当消费者进入地理围栏时,位置和地理围栏组件可以生成用于服务提供者的触发,基于此,目标服务可以被递送。在示例中,支付促进器 SDP 5508 可以包括增强现实组件。增强现实组件可以提供可以在钱包和微件内并入的独特和先进的用户体验基础设施。

[0515] 在基于本文描述的 MTP 方法和系统的 SDP 5502 生态系统部署的示例性实施例中,可以针对 ESB 创建消息定义,其可以执行可以针对支付促进器而创建的标准和协议。然后,服务提供者可以实现 ESB 连接器,以诸如以支付促进器基础设施的各种利益相关者可以消费服务的方式向平台展露其服务。

[0516] 在基于本文描述的 MTP 方法和系统的 SDP 5502 生态系统部署的示例性实施例中,来自 SDP 5508 的服务供应可以是一组基本步骤。在一些实施例中,取决于服务的性质和服务提供者的偏好,一些步骤可以是可选的。在示例中,第一步骤可以包括注册和 KYC。在该步骤处,消费者可以请求服务(钱包、支付帐户等),并提供所需的个人信息以便执行相关 KYC 步骤。对注册和 KYC 进一步地,可以要求服务批准以用于进行进一步处理。服务批准可以要求服务合作伙伴在遵循相关内部商业过程之后批准服务请求。服务批准可以创建用于消费者的相关证书以及任何服务有效载荷。在一些实施例中,服务批准可以是可选的。下一步骤可以是服务供应。在该步骤处,可以将服务供应给用户。服务供应可以包括向消费者电话递送服务的组件,或者在消费者服务器上创建必要的更新。服务供应之后可以是服务激活。可以要求服务激活以确认服务被正确供应给正确用户,并且对于要消费的服务,所有状态被设置为“就绪”。在一些实施例中,服务激活可以是可选的。下一步骤可以是服务执行。在该步骤处,消费者可以开始从其应用内使用服务。下一步骤可以是服务管理,其可以包括服务的所有更新和生命周期组件。

[0517] 在 SDP 5502 生态系统部署的示例性实施例中,用于支付促进器的 SDP 5508 架构可以允许服务合作伙伴以如下方式配置和控制它们的服务产品 :它们遵照支付促进器规范(通过 ESB 接口),但是控制它们自己的过程、服务数据以及商业规则。这样的控制和管理对于可以吸引关键性大量服务提供者的开放式生态系统可能是必需的。

[0518] 在基于本文描述的 MTP 方法和系统的包括钱包、容器、MTP 服务器、服务提供者等的 SDP 5502 生态系统部署的示例性实施例中,可以存在可以在 POS 处进行交易的移动电话替代物。本文描述了高层级移动支付场景的流。然而,应当注意,这些步骤是高层级的,并且可以被重新安排以适合特定用户体验,并且可以插入其它步骤以便提供附加功能性。本

公开文档的目的是展示用于支付促进器的各种部署选项，而非支付的特定用例。

[0519] 在示例性且非限制性实施例中，为了对移动支付起作用，消费者可能必须首先到达结账终端并且可以扫描要购买的物品。然后，消费者可以发动移动电话上的移动应用，并且可以被要求输入登录证书(例如，PIN、口令或口令短语等)。移动应用可以连接到 SDP 5508 以验证登录并对消费者和设备进行认证。在服务和设备成功认证时，移动应用可以呈现可以存储在消费者的移动钱包中的支付选项列表。可以要求消费者从提供给消费者的列表中选择支付选项(例如，礼品卡)。

[0520] 在可能选择了支付选项之后，移动应用可以发起与 POS 的结账过程。取决于 POS 和移动电话可以支持的信道，可以经由多个实现选项来发起支付选项。在示例中，标记化的 NFC 可以用于发起结账过程。针对该选项，POS 可以被近场通信(NFC)使能。移动应用可以针对用于所选支付卡的标记而请求 SDP 5508。SDP 5508 可以认证移动应用，并且可以将该请求路由到相关发行者的对应标记生成器。可以要求消费者在 NFC 使能的 POS 处轻敲电话以传输标记。在示例中，NFC 可以是国际标准化组织(ISO)和国际电工技术委员会(IEC)标准 18092。在示例中，安全元件 NFC (SE NFC)可以用于发起结账过程。在该情况中，可以要求消费者在 NFC 使能的 POS 处轻敲电话，其中支付证书存储在电话上的 SE 中。在示例中，NFC 可以是国际标准化组织(ISO)和国际电工技术委员会(IEC)标准 14443。在示例中，QR 码扫描可以用于发起结账过程。移动应用可以针对用于所选支付卡的标记而请求 SDP 5508。SDP 5508 可以认证移动应用，并且可以将该请求路由到相关发行者的对应标记生成器。移动应用可以将标记转换成通过 POS 扫描的 2D 条形码。在示例中，虚拟读取器可以用于发起结账过程。移动应用可以通过扫描 QR 码或轻敲 NFC 标签来捕获商家、POS 和交易信息。移动应用可以将数据连同所选卡通过 OTA 发送给 SDP 5508。SDP 5508 可以请求标记，并且通过后端过程将标记连同其它信息一起发送给 POS。

[0521] 在发起结账过程之后的下一步骤可以要求 POS 处理交易并且将所述处理路由到支付促进器。在下一步骤处，支付促进器可以将交易路由到支付卡的发行者。在下一步骤处，发行者可以授权交易并且可以将批准递送给 POS。POS 可以确认交易并递送收据。在示例中，收据可以被打印、电子地递送给移动应用或者通过电子邮件发送到经核实的电子邮件地址。收据的递送可以使移动支付过程完成，并且消费者可以带着购买的物品离开结账终端。

[0522] 在基于本文描述的 MTP 方法和系统的包括钱包、容器、MTP 服务器、服务提供者等的 SDP 5502 生态系统部署的示例性实施例中，商家可以采用各种部署模型。商家可以处于其单独移动策略中的各种阶段；并因此，可以要求其支持用于商家的多个部署模型以便采用公共支付促进器。支付促进器连同每个发行者一起可以从安全性和风险管理视角批准并证明整个交易路径，以诸如保持低交易成本。因此，具有可以在所有参与商家间共享的公共平台可以是重要的。

[0523] 在示例性且非限制性实施例中，可以存在可以用于部署的三个模型。这三个模型是共存的，并且商家可以取决于其单独策略来采用这些模型中的一个或多个。在示例性且非限制性实施例中，可以部署比本文描述的三个模型更多或更少的模型。在示例性且非限制性实施例中，这些模型可以被配置用于实现支付、供应、忠诚度供应等等。

[0524] 在基于本文描述的 MTP 方法和系统的 SDP 5502 生态系统部署的示例性实施例

中，部署模型可以是具有子支付促进器钱包的商家应用。可以要求消费者通过使用他们现有的、市场上的商家移动应用来在 POS 处支付，所述商家移动应用通过由 SDP 5508 钱包平台供应和管理的直接子钱包而链接到他们的主支付促进器钱包。子钱包可以被独立配置，但是链接到主钱包。在示例中，现有的商家移动应用通过 OTA web 服务接口而直接与 SDP 5508 相接口。SDP 5508 可以提供对消费者的支付促进器钱包的经认证的、受控制的并且经过滤的访问，并且商家应用可以用于 POS 处的交易。在示例中，SDP 5508 可以聚集支持多个支付信道的多个支付发行者，并且管理消费者的父支付促进器钱包和子商家钱包二者。

[0525] 在基于本文描述的 MTP 方法和系统的 SDP 5502 生态系统部署的示例性实施例中，部署模型可以是具有商家钱包的商家应用。商家可以在其移动应用内使用其独立第三方钱包，并且可以访问来自主支付促进器钱包的支付证书。

[0526] 在基于本文描述的 MTP 方法和系统的 SDP 5502 生态系统部署的示例性实施例中，部署模型可以基于支付促进器容器。消费者可以通过使用支付促进器品牌化的移动应用来支付，该移动应用可以从 SDP 5502 自动得以填充并且向消费者提供到其父支付促进器钱包中的统一视图。SDP 5502 可以聚集支持多个支付信道的多个支付发行者，并且管理消费者的支付促进器钱包。在示例中，支付促进器移动应用可以与商家 POS 交互以执行交易，而不用商家进行任何附加开发。在示例中，商家和发行者可以可选地创建其自己的微件以在支付促进器应用内运行来创建定制体验。

[0527] 图 56 图示了可以基于支付促进器容器的部署模型 5600 的高层级框图。

[0528] 根据所图示的实施例，支付促进器内的各种子系统可以包括销售点（POS）5602、诸如移动应用 5604 之类的应用（APP）5604、SDP 5608、若干发行者系统，诸如无限制地包括所描绘的发行者 1 系统 5610、发行者 2 系统 5612 以及发行者 3 系统 5614。各种子系统可以通过诸如支付处理网络 5618 之类的网络来连接。APP 5604 此外可以包括第一微件 5620、第二微件 5622 以及钱包 5624。

[0529] 所图示的实施例可以是支付促进器支付指南的一个示例，其可能必须跨诸如发行者系统 5610-5614 之类的多个发行者系统和多个商家来管理。消费者可以不知道移动交易的复杂后端操作，以便创建“选择卡并支付”的简单体验。可以通过支付促进器钱包容器应用 5604 来创建和市场推广支付促进器品牌。在实施例中，商家可以将支付促进器标准提升为支付接受品牌。在示例中，可以用不同的支付产品和多个 POS 5602 信道来支持多个发行者 5610-5614，从而给予消费者对支付选项的选择，该选择可以跨多个商家而适合他们的需要。支付促进器容器可以要求在商家一方用以接受支付促进器支付选项的最小努力。可以要求 POS 5602 侧工作，因为微件可以是可选的。支付促进器容器可以确保支付处理顺从用于基于 POS 5602 的交易的支付促进器指南和标准。

[0530] 支付促进器容器模型可以利用 SDP 的全部力量，多个服务提供者的后和前端聚集。如图 56 中所图示，支付促进器品牌化的移动应用 5604（容器）可以通过使用 MTP 钱包工具套件来构建。

[0531] 容器可以包括可以并入支付促进器规范的钱包实现。当消费者创建其支付促进器钱包时，钱包可以从 SDP 得以自动填充。在实施例中，支付促进器容器可以用于在支付促进器顺从的 POS 5602 处进行支付，而不用商家进行任何进一步开发。在实施例中，商家可以有能力通过使用 MTP 工具套件来创建微件，MTP 工具套件可以专门设计用于在 POS 5602 处

提供以商家为中心的用户体验。微件创建可以任凭商家决定，并且可以在任何时候被实现或添加。SDP 可以聚集并允许机载多个发行者 5610-5614，并且可以允许商家定义其支付促进器兼容的支付产品和其它增值服务(VAS)。在这样的机载期间，发行者 5610-5614 可以选择针对每个支付产品可以受支持的 POS 5602 交互协议。POS 5602 交互协议可以确定哪些商家可以接受哪些支付产品，因为商家的 POS 5602 信道可能必须支持发行者选择的协议中至少一个，例如，NFC、条形码、OTA 等等。

[0532] 在一些示例性实施例中，SDP 可以被配置为管理 3 维矩阵。该矩阵可以被配置用于商家的 POS 5602 支持的邻近信道，例如，NFC、2D 条形码、OTA 等等。该矩阵可以被配置用于商家 POS 5602 能力。该矩阵可以被配置用于发行者批准的信道。

[0533] 在基于本文描述的 MTP 方法和系统的 SDP 生态系统部署的示例性实施例中，支付促进器可以用于利用商家微件的移动支付。在示例性且非限制性实施例中，为了使移动支付起作用，消费者可能必须首先到达结账终端并且可以扫描要购买的物品。然后，消费者可以发动移动电话上的支付促进器移动应用 5604，并且可以被要求输入支付促进器管理的登录证书(例如，PIN、口令或口令短语等)。然后，可以要求消费者从支付使能的商家的操纵盘(dashboard)选择商家。商家微件可以被发动于“支付”模式中，并且呈现在商家 POS 5602 处接受的支付选项列表。支付选项列表可以是消费者的钱包中本地存储的选项上经过滤的视图，以作为支付促进器容器的一部分。可以要求消费者从提供给消费者的列表中选择支付选项(例如，礼品卡)。

[0534] 商家微件可以从 SDP 请求交易有效载荷，其可以发送到商家 POS 5602 以用于由商家 POS 5602 支持的对应信道。SDP 可以传递针对商家 POS 5602 格式化的交易有效载荷。在示例性实施例中，交易有效载荷可以用优惠券、忠诚度等形式。可以基于商家 POS 5602 能力来附接交易有效载荷。在接收到交易有效载荷时，商家微件可以发起与 POS 5602 的结账过程。该结账过程的发起可以类似于上面参照没有商家微件生成的示例所描述的那样来进行。

[0535] 在发起结账过程之后的下一步骤可以要求 POS 5602 处理交易并将所述处理路由到支付促进器。在下一步骤处，支付促进器可以将交易路由到支付卡的发行者。在下一步骤处，发行者可以授权交易并且可以将批准递送给 POS 5602。POS 5602 可以确认交易并递送收据。在示例中，收据可以被打印、电子地递送给移动应用 5604 或者通过电子邮件发送到经核实的电子邮件地址。收据的递送可以使移动支付过程完成，并且消费者可以带着购买的物品离开结账终端。

[0536] 在基于本文描述的 MTP 方法和系统的 SDP 生态系统部署的示例性实施例中，用户接口(UI)在支付促进器容器中。本文描述的 UI 概念可以仅仅用作对示例性实施例的参考。取决于用户需要，UI 概念可以服从于改变。在示例性实施例中，UI 概念可以包括钱包控制的模式和微件控制的模式。钱包控制模式可以被配置，使得支付促进器钱包可以控制商家 POS 5602 交互和支付体验。钱包控制模式可以绕过商家创建微件的需要。微件控制模式可以被配置，使得商家可以创建微件，并且微件可以将控制商家 POS 5602 处的交互和支付体验。

[0537] 本文描述的方法和系统可以部分地或整体地通过在处理器上执行计算机软件、程序代码和 / 或指令的机器来部署。处理器可以是服务器、客户端、网络基础设施、移动计算

平台、固定计算平台或其它计算平台的部分。处理器可以是能够执行程序指令、代码、二进制指令等的任何种类的计算或处理设备。处理器可以是或者包括单个处理器、数字处理器、嵌入式处理器、微处理器或者诸如协处理器(数学协处理器、图形协处理器、通信协处理器等)的任何变体等,其可以直接或间接促进存储在其上的程序代码或程序指令的执行。另外,处理器可以使能多个程序、线程和代码的执行。线程可以同时执行以增强处理器的性能,并且促进应用的同时操作。作为实现方式,本文描述的方法、程序代码、程序指令等可以在一个或多个线程中实现。线程可以繁衍(spawn)其它线程,所述其它线程可以具有所指派的与它们相关联的优先级;处理器可以基于优先级或者任何其它次序基于在程序代码中提供的指令来执行这些线程。处理器可以包括存储器,其存储如本文及别处所描述的方法、代码、指令和程序。处理器可以通过接口来访问存储介质,其可以存储如本文及别处所描述的方法、代码和指令。与处理器相关联的用于存储能够由计算或处理设备执行的方法、程序、代码、程序指令或其它类型的指令的存储介质可以包括但不限于以下各项中的一个或多个:CD-ROM、DVD、存储器、硬盘、闪速驱动器、RAM、ROM、高速缓存等。

[0538] 处理器可以包括可以增强多处理器的速度和性能的一个或多个核。在实施例中,处理器可以是组合两个或多个独立核心(称为管芯)的双核处理器、四核处理器、其它芯片层级多处理器等。

[0539] 本文描述的方法和系统可以部分地或整体地通过在服务器、客户端、防火墙、网关、集线器、路由器或其它此类计算机和/或联网硬件上执行计算机软件的机器来部署。软件程序可以与服务器相关联,服务器可以包括文件服务器、打印服务器、域服务器、互联网服务器、内联网服务器以及诸如辅助服务器、主机服务器、分布式服务器等的其它变体。服务器可以包括以下中的一个或多个:存储器、处理器、计算机可读介质、存储介质、端口(物理的和虚拟的)、通信设备以及接口等,接口能够通过有线或无线介质来访问其它服务器、客户端、机器以及设备。如本文和别处所描述的方法、程序或代码可以由服务器执行。另外,对于执行如本申请中所描述的方法所需的其它设备可以认为是与服务器相关联的基础设施的一部分。

[0540] 服务器可以提供至其它设备的接口,所述其它设备无限制地包括:客户端、其它服务器、打印机、数据库服务器、打印服务器、文件服务器、通信服务器、分布式服务器等。另外,该耦合和/或连接可以促进跨网络的程序的远程执行。一些或所有这些设备的联网可以促进程序或方法在一个或多个位置处的并行处理,而不偏离本范围。另外,通过接口附接到服务器的任何设备可以包括能够存储方法、程序、代码和/或指令的至少一个存储介质。中央贮存库可以提供要在不同设备上执行的程序指令。在该实现中,远程贮存库可以充当用于程序代码、指令和程序的存储介质。

[0541] 软件程序可以与客户端相关联,客户端可以包括文件客户端、打印客户端、域客户端、互联网客户端、内联网客户端以及诸如辅助客户端、主机客户端、分布式客户端等的其它变体。客户端可以包括以下中的一个或多个:存储器、处理器、计算机可读介质、存储介质、端口(物理的和虚拟的)、通信设备以及接口等,接口能够通过有线或无线介质来访问其它客户端、服务器、机器以及设备。如本文和别处所描述的方法、程序或代码可以由客户端执行。另外,对于执行如本申请中所描述的方法所需的其它设备可以认为是与客户端相关联的基础设施的一部分。

[0542] 客户端可以提供至其它设备的接口，所述其它设备无限制地包括：服务器、其它客户端、打印机、数据库服务器、打印服务器、文件服务器、通信服务器、分布式服务器等。另外，该耦合和/或连接可以促进跨网络的程序的远程执行。一些或所有这些设备的联网可以促进程序或方法在一个或多个位置处的并行处理，而不偏离本范围。另外，通过接口附接到客户端的任何设备可以包括能够存储方法、程序、应用、代码和/或指令的至少一个存储介质。中央贮存库可以提供要在不同设备上执行的程序指令。在该实现中，远程贮存库可以充当用于程序代码、指令和程序的存储介质。

[0543] 本文描述的方法和系统可以部分地或整体地通过网络基础设施来部署。网络基础设施可以包括诸如以下的元件：计算设备、服务器、路由器、集线器、防火墙、客户端、个人计算机、通信设备、路由设备以及如本领域所已知的其它有源和无源设备、模块和/或组件。与网络基础设施相关联的(多个)计算和/或非计算设备可以包括与其它组件分离的存储介质，诸如闪速存储器、缓冲器、栈、RAM、ROM 等。本文和别处描述的过程、方法、程序代码、指令可以由一个或多个网络基础设施元件来执行。

[0544] 本文和别处描述的方法、程序代码和指令可以在具有多个小区的蜂窝网络上实现。蜂窝网络可以是频分多址(FDMA)网络或码分多址(CDMA)网络。蜂窝网络可以包括移动设备、小区站点、基站、中继器、天线、塔等。小区网络可以是 GSM、GPRS、3G、EVDO、网格或其它网络类型。

[0545] 本文和别处描述的方法、程序代码和指令可以在移动设备上或通过移动设备来实现。移动设备可以包括导航设备、手机、移动电话、移动个人数字助理、膝上型计算机、掌上型计算机、上网本、寻呼机、电子书阅读器、音乐播放器等。这些设备可以包括与其它组件分离的存储介质，诸如闪速存储器、缓冲器、RAM、ROM 以及一个或多个计算设备。与移动设备相关联的计算设备可以被使能来执行其上存储的程序代码、方法和指令。可替换地，移动设备可以被配置为与其它设备协同来执行指令。移动设备可以与基站通信，所述基站与服务器相接口且被配置为执行程序代码。移动设备可以在对等网络、网格网络或其它通信网络上通信。程序代码可以存储在与服务器相关联的存储介质上，并且由嵌入在服务器内的计算设备来执行。基站可以包括计算设备和存储介质。存储设备可以存储由与基站相关联的计算设备执行的程序代码和指令。

[0546] 计算机软件、程序代码和/或指令可以在机器可读介质上存储和/或访问，所述机器可读介质可以包括：计算机组件、设备和记录介质，其保留用于在某时间间隔内进行计算的数字数据；称为随机存取存储器(RAM)的半导体存储装置；典型地用于较永久存储的大容量存储装置，诸如光学盘、像硬盘、磁带、磁鼓、卡那样的磁存储形式及其它类型；处理器寄存器、高速缓冲存储器、易失性存储器、非易失性存储器；诸如 CD、DVD 的光学存储装置；可移动介质，诸如闪速存储器(例如，USB 棒或 U 盾(USB key))、软盘、磁性带、纸带、穿孔卡、独立 RAM 盘、Zip 驱动器、可移动大容量存储装置、离线装置等；其它计算机存储器，诸如动态存储器、静态存储器、读/写存储装置、易变存储装置(mutable storage)、只读存储装置、随机存取存储装置、顺序存取存储装置、位置可寻址存储装置、文件可寻址存储装置、内容可寻址存储装置、网络附接的存储装置、存储区域网络、条形码、磁性墨等。

[0547] 本文描述的方法和系统可以将物理和/或无形项目从一个状态变换到另一个状态。本文描述的方法和系统还可以将表示物理和/或无形项目的数据从一个状态变换到另

一个状态。

[0548] 本文描述和描绘的、包括在贯穿附图的流程图和框图中的元件暗含这些元件之间的逻辑边界。然而，根据软件或硬件工程实践，所描绘的元件及其功能可以通过计算机可执行介质而实现在机器上，所述机器具有能够执行程序指令的处理器，所述程序指令在其上存储为单片软件结构、独立软件模块或采用外部例程、代码、服务等的模块或这些的任意组合，并且所有这样的实现可以在本公开的范围内。这样的机器的示例可以包括但是可以不限于：个人数字助理、膝上型计算机、个人计算机、移动电话、其它手持计算设备、医疗装备、有线或无线通信设备、换能器、芯片、计算器、卫星、平板 PC、电子书、小机件、电子设备、具有人工智能的设备、计算设备、联网装备、服务器、路由器等。此外，在流程图和框图中描绘的元件或任何其它逻辑组件可以实现在能够执行程序指令的机器上。因此，尽管前述附图和描述阐述了所公开系统的功能方面，但是除非明确声明或者从上下文另外清楚的，否则不应从这些描述中推断出用于实现这些功能方面的软件的任何特定布置。类似地，可以意识到，上文标识和描述的各种步骤可以变化，并且步骤顺序可以适应于本文公开的技术的特定应用。所有这样的变型和修改意图落入本公开的范围内。因而，除非由特定应用要求或者明确声明或从上下文另外清楚的，否则针对各个步骤的顺序的描绘和 / 或描述不应理解为要求特定顺序来执行那些步骤。

[0549] 以上描述的方法和 / 或过程及其步骤可以实现在适于特定应用的硬件、软件或者硬件和软件的任何组合中。硬件可以包括通用计算机和 / 或专用计算设备或特定计算设备，或者特定计算设备的具体方面或组件。过程可以实现在一个或多个微处理器、微控制器、嵌入式微控制器、可编程数字信号处理器或其它可编程设备，连同内部和 / 或外部存储器中。过程还可以或者替代地具体化在以下各项中：专用集成电路、可编程门阵列、可编程阵列逻辑，或者可以被配置为处理电子信号的任何其它设备或设备的组合。此外可以意识到，一个或多个过程可以实现为能够在机器可读介质上执行的计算机可执行代码。

[0550] 计算机可执行代码可以通过使用诸如 C 的结构化编程语言、诸如 C++ 的面向对象编程语言或任何其它高级或低级编程语言（包括汇编语言、硬件描述语言以及数据库编程语言和技术）来创建，所述计算机可执行代码可以被存储、编译或解译以在以上设备以及处理器的异构组合、处理器架构或者不同硬件和软件的组合或者能够执行程序指令的任何其它机器之一上运行。

[0551] 因此，在一个方面中，上文描述的每种方法及其组合可以具体化在计算机可执行代码中，所述计算机可执行代码当在一个或多个计算设备上执行时执行方法的步骤。在另一方面中，这些方法可以具体化在执行其步骤的系统中，并且可以以多种方式跨设备分布，或者所有功能性可以集成在专用的独立设备或其它硬件中。在另一方面中，用于执行与上文描述的过程相关联的步骤的装置可以包括上文描述的硬件和 / 或软件中的任何一个。所有这样的置换和组合意图落入本公开的范围内。

[0552] 尽管已经结合详细示出和描述的某些优选实施例公开了本文描述的方法和系统，但是对其的各种修改和改进对于本领域技术人员可以容易变得显而易见。因此，本文描述的方法和系统的精神和范围将不受前述示例的限制，而是要在通过法律可允许的最广的意义上理解。

[0553] 据此通过引用并入本文中引用的所有文档。

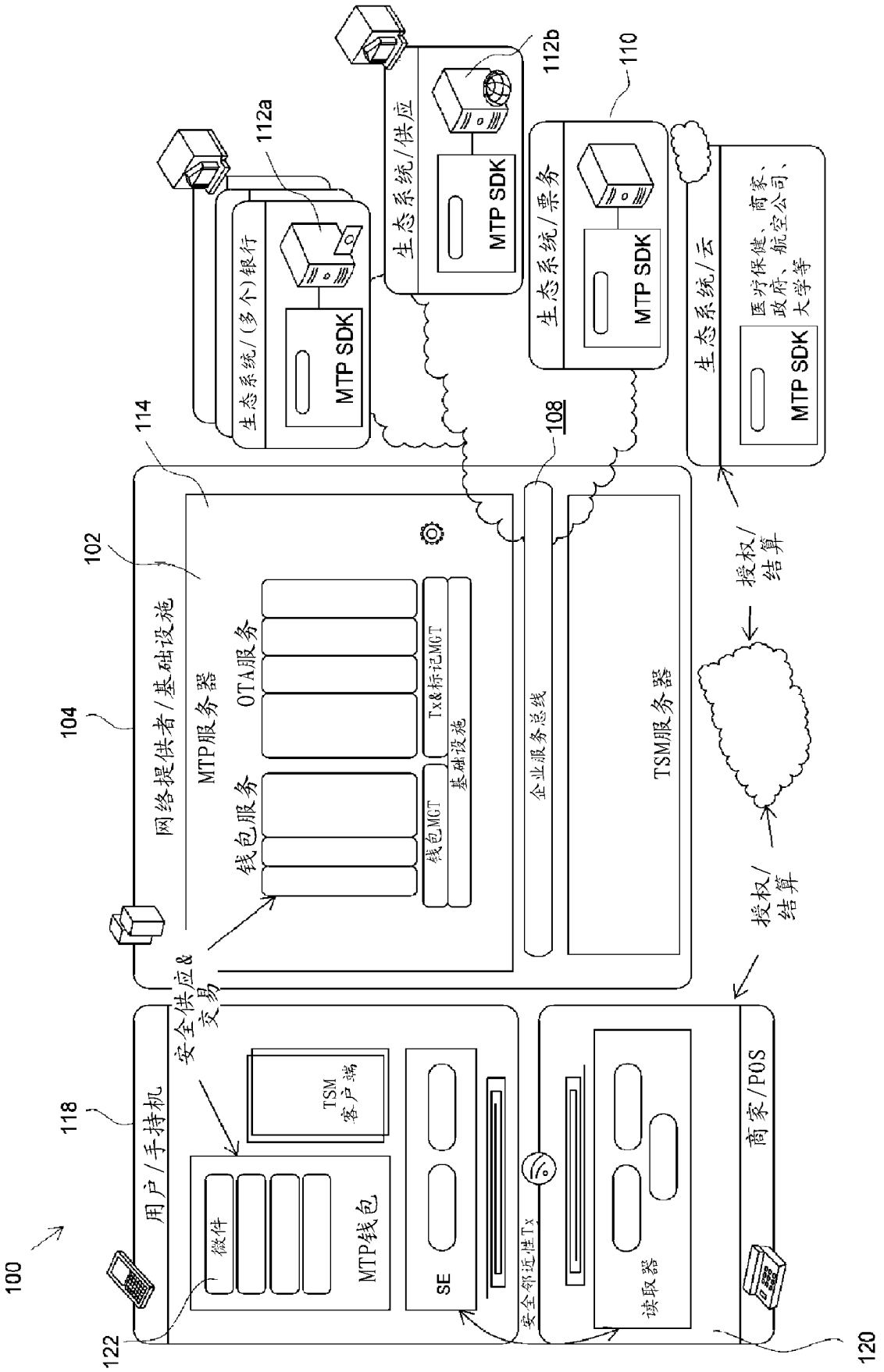

100
用户/手持机
118
微件
MTP钱包
TSM客户端
安全供应&交易
122
SE
安全邻近性Tx
读取器
商家/POS
120
网络提供者/基础设施
102
MTP服务器
OTA服务
钱包服务
钱包MGT
Tx&标记MGT
基础设施
104
114
企业服务总线
108
TSM服务器
授权/结算
授权/结算
生态系统/(多个)银行
MTP SDK
112a
生态系统/供应
MTP SDK
112b
生态系统/票务
MTP SDK
110
生态系统/云
MTP SDK
医疗保健、商家、政府、航空公司、大学等

图 1

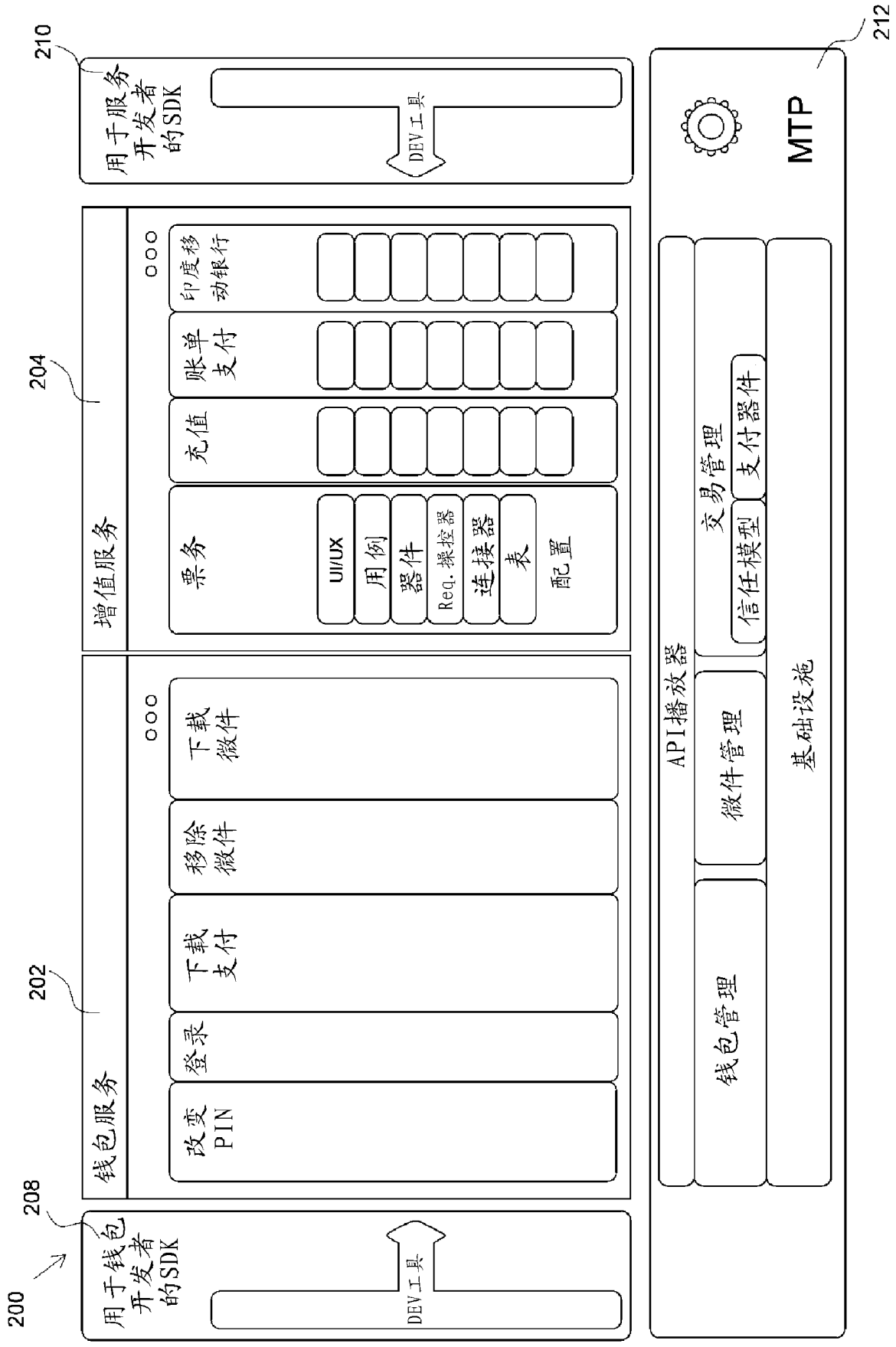

200
208
用于钱包开发者的SDK
DEV工具
钱包服务
202
改变PIN
登录
下载支付
移除微件
下载微件
ooo
增值服务
204
票务
UI/UX
用例
器件
Req.操控器
连接器
表
配置
充值
账单支付
印度移动银行
ooo
210
用于服务开发者的SDK
DEV工具
212
MTP
API播放器
钱包管理
微件管理
交易管理
信任模型
支付器件
基础设施

图 2

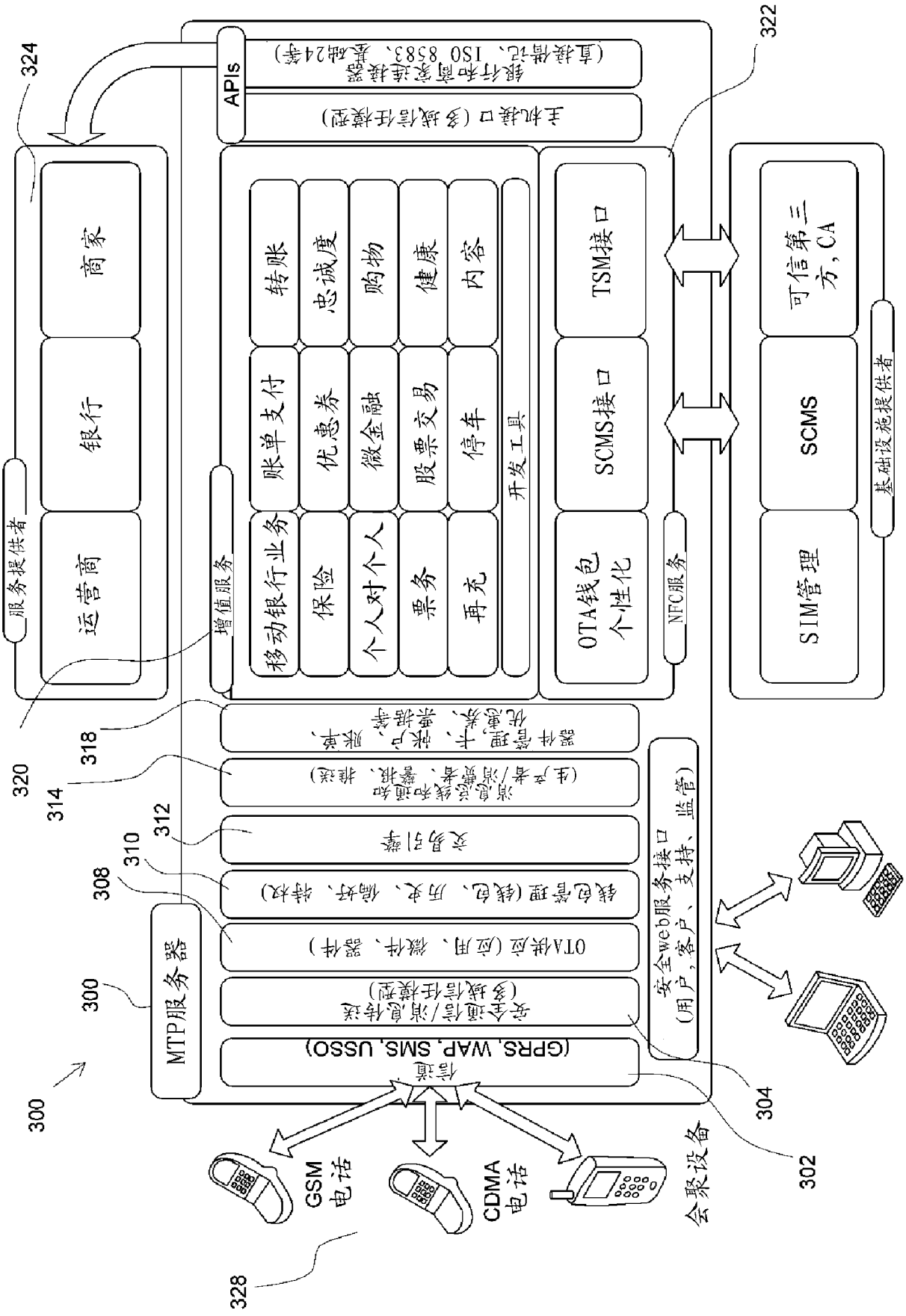

300
MTP服务器
302
304
会聚设备
安全web服务接口
(用户，客户、支持、监管)
信道
(GPRS, WAP, SMS, USSD)
安全通信/消息传送
(多域信任模型)
OTA供应 (应用、微件、器件)
钱包管理 (钱包、历史、偏好、特权)
交易引擎
消息/警报线索和通知
(生产者/消费者、警报、推送)
器件管理 (卡、条款、票据等)
APIs
主机接口 (多域信任模型)
银行和商家准则
(直接借记、ISO 8583、基础24等)
TSM接口
SCMS接口
OTA钱包
个性化
NFC服务
SCMS
SIM管理
可信第三方，CA
基础设施提供者
转账
忠诚度
购物
健康
内容
账单支付
优惠券
微金融
股票交易
停车
移动银行业务
保险
个人对个人
票务
再充
开发工具
增值服务
商家
银行
运营商
服务提供者
GSM电话
CDMA电话
328
320
314
318
308
310
312
322
324

图 3

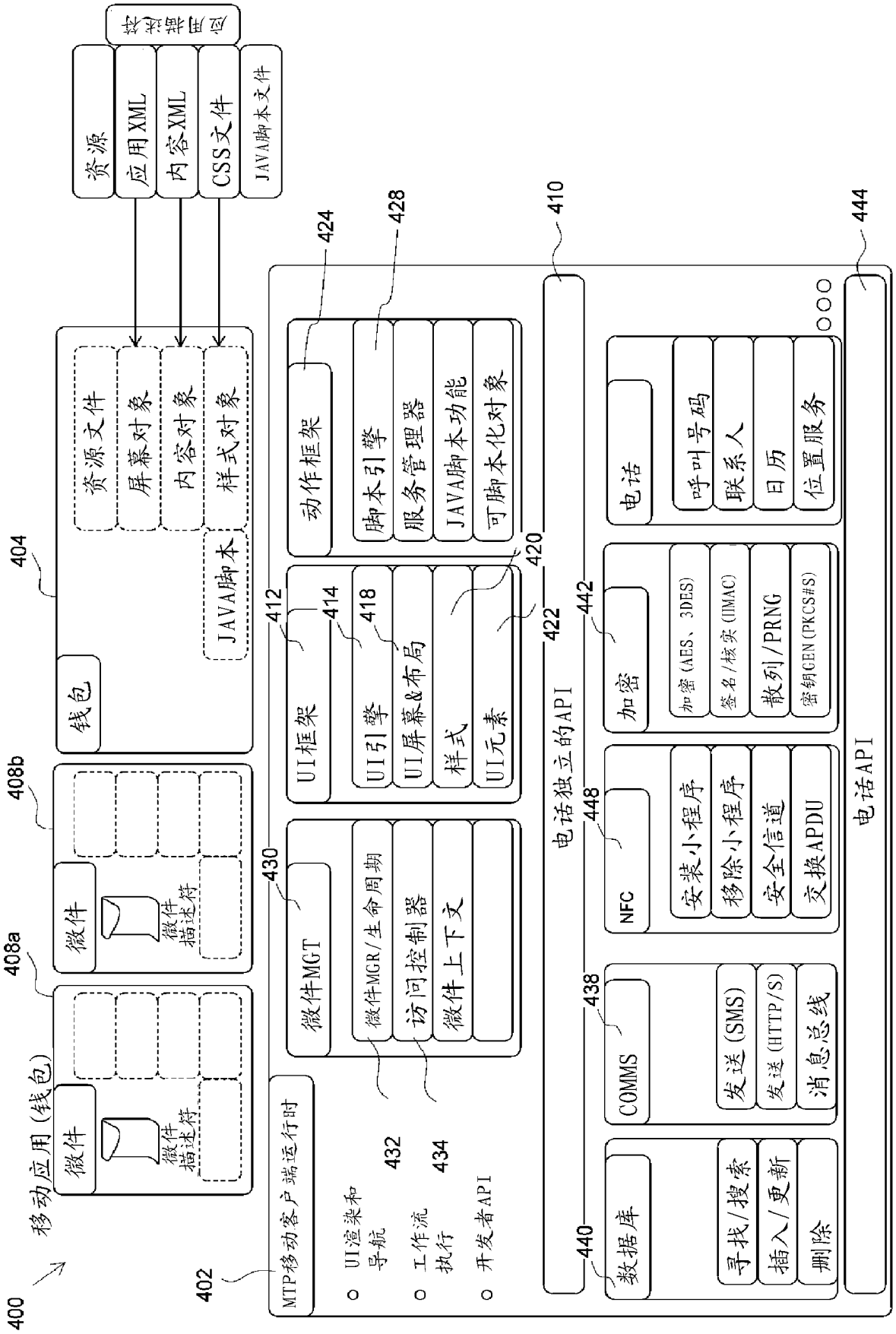

400
移动应用(钱包)
402
MTP移动客户端运行时
UI渲染和导航
工作流执行
开发者API
432
434
408a
微件
微件描述符
408b
微件
微件描述符
404
钱包
资源文件
屏幕对象
内容对象
样式对象
JAVA脚本
资源
应用XML
内容XML
CSS文件
JAVA脚本文件
430
微件MGT
微件MGR/生命周期
访问控制器
微件上下文
412
UI框架
414
UI引擎
418
UI屏幕&布局
样式
UI元素
420
422
电话独立的API
424
动作框架
428
脚本引擎
服务管理器
JAVA脚本功能
可脚本化对象
410
442
加密
加密(AES、3DES)
签名/核实(IMAC)
散列/PRNG
密钥GEN(PKCS#5)
448
NFC
安装小程序
移除小程序
安全信道
交换APDU
438
COMMS
发送(SMS)
发送(HTTP/S)
消息总线
440
数据库
寻找/搜索
插入/更新
删除
444
电话API
电话
呼叫号码
联系人
日历
位置服务

图 4

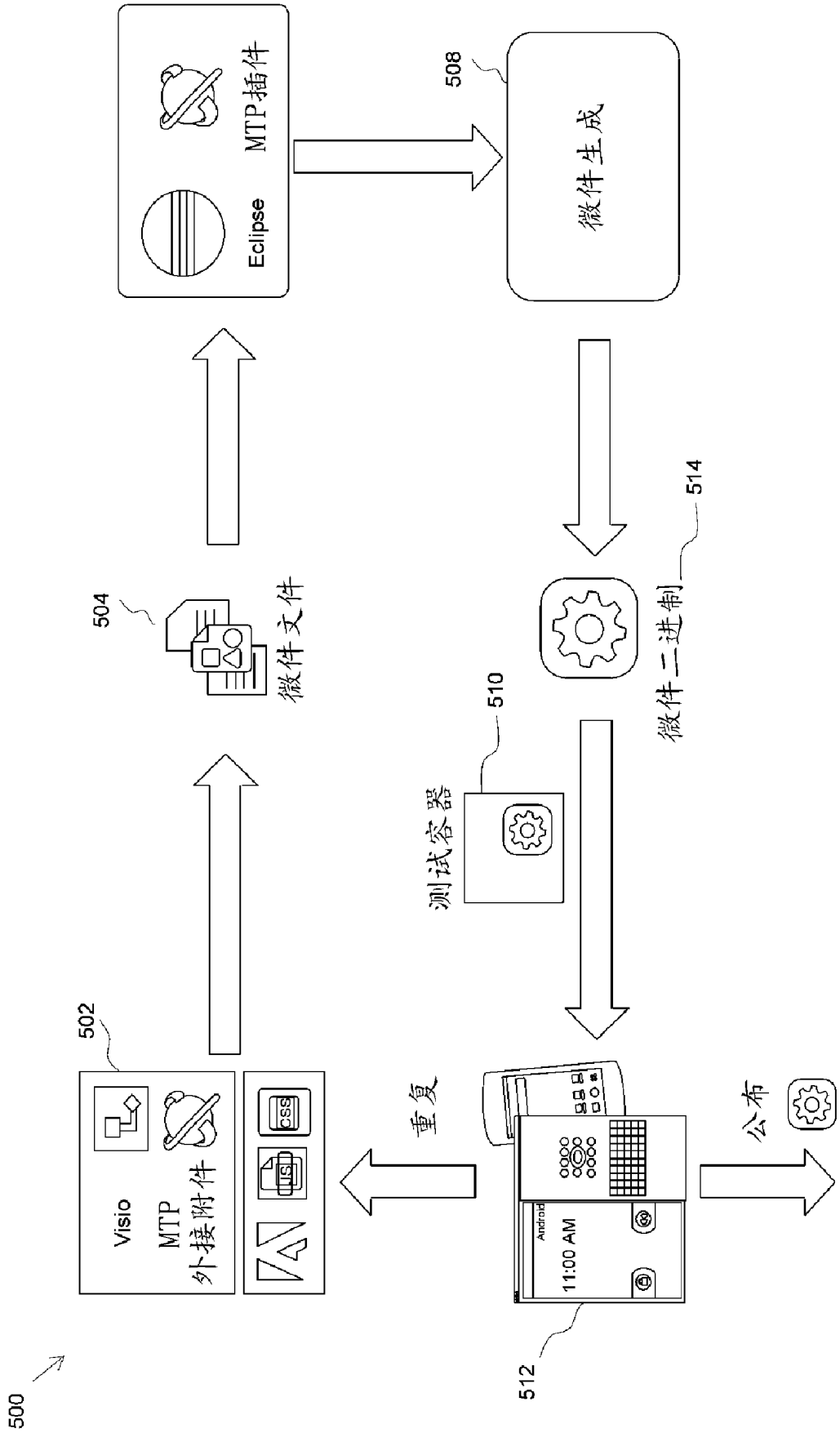

500
502
Visio
MTP
外接附件
CSS
504
微件文件
Eclipse
MTP插件
508
微件生成
514
微件二进制
510
测试容器
重复
512
Android
11:00 AM
公布

图 5

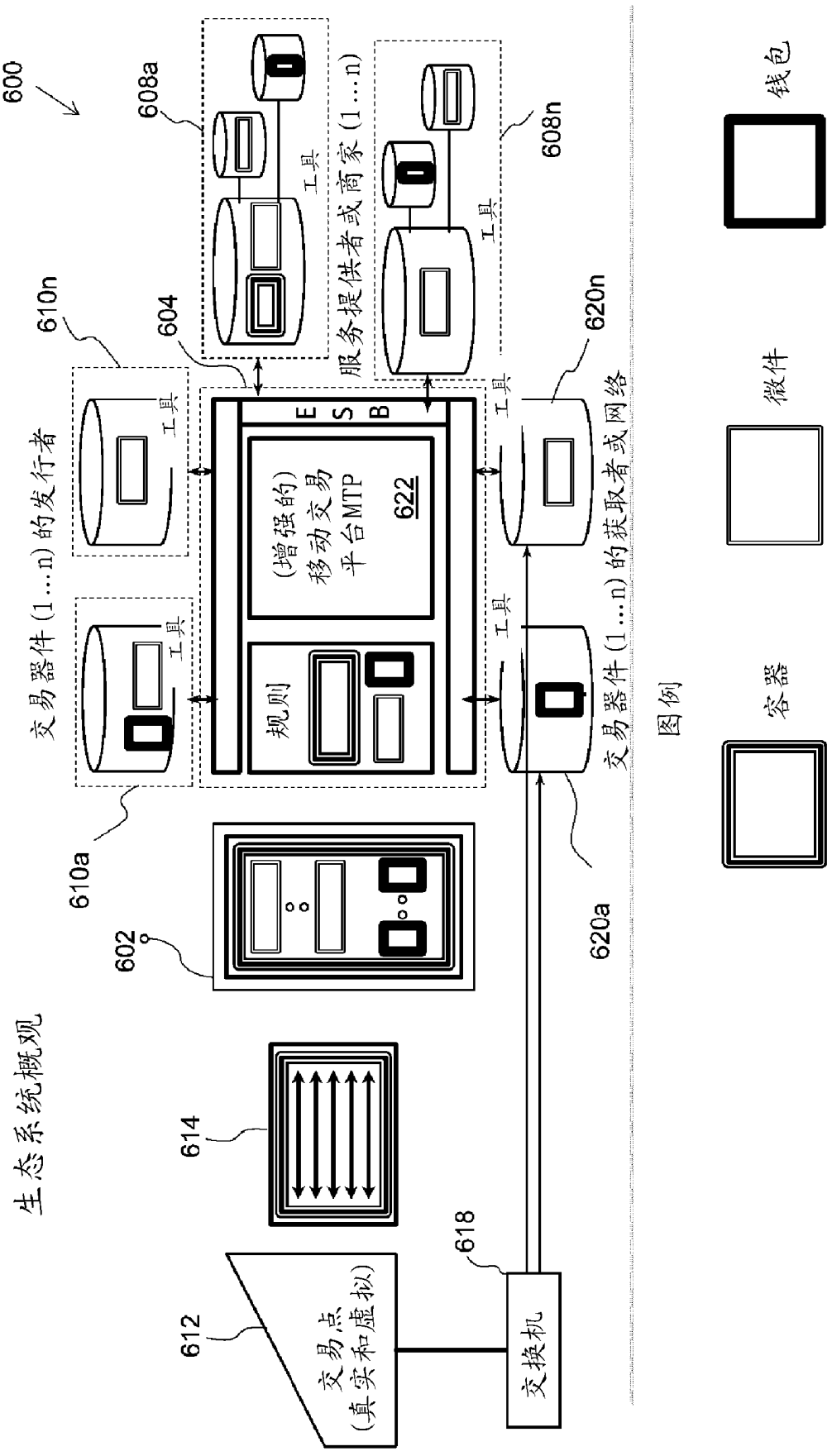

600
608a
608n
610n
604
620n
610a
602
620a
614
618
612
生态系统概观
交易器件(1...n)的发行者
服务提供者或商家(1...n)
交易器件(1...n)的获取者或网络
工具
E S B
(增强的)
移动交易
平台MTP
622
规则
交易点
(真实和虚拟)
交换机
图例
钱包
微件
容器

图 6

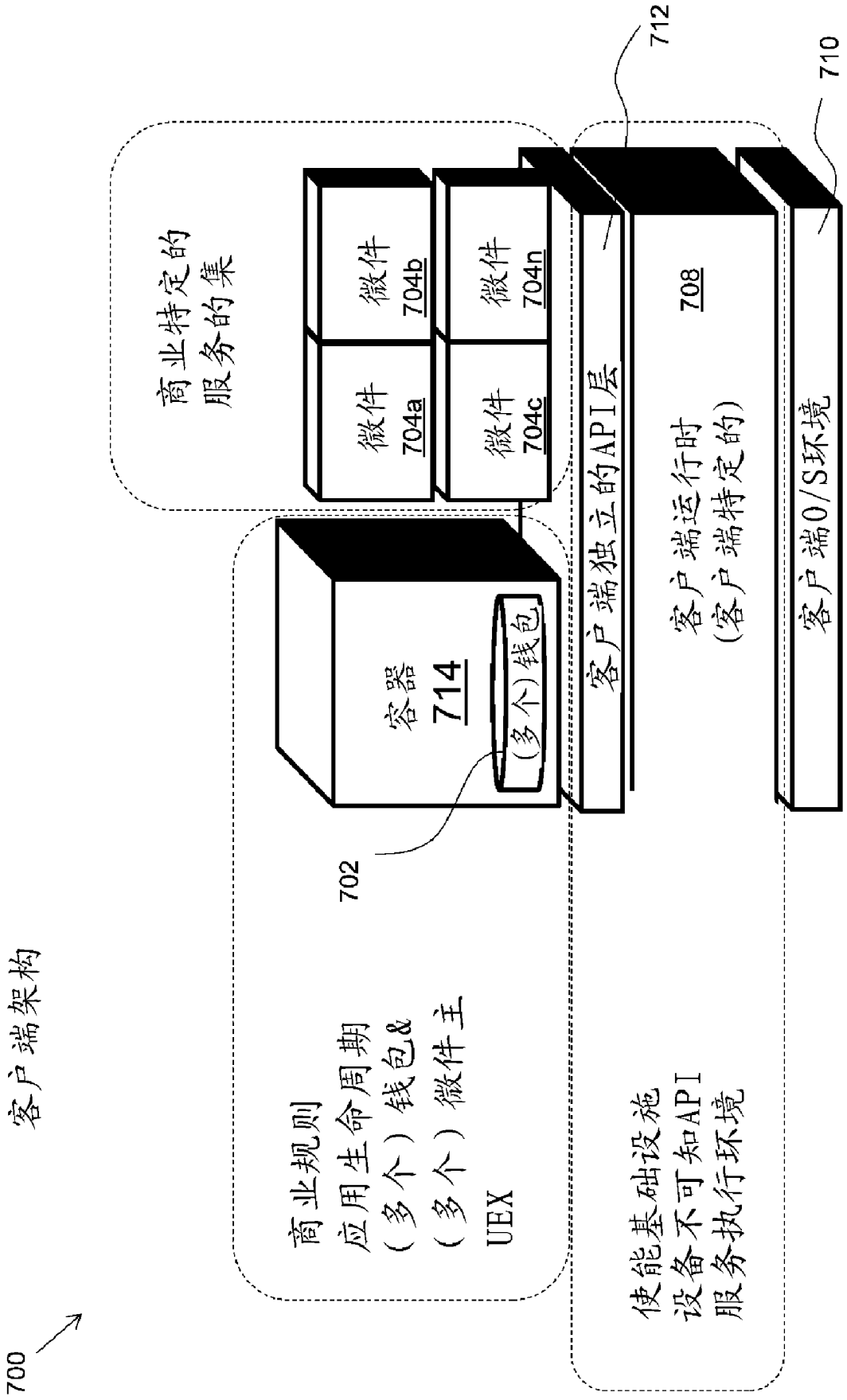

700
客户端架构
商业规则
应用生命周期
（多个）钱包&
（多个）微件主
UEX
商业特定的
服务的集
微件
704a
微件
704b
微件
704c
微件
704n
容器
714
（多个）钱包
702
客户端独立的API层
712
客户端运行时
（客户端特定的）
708
客户端O/S环境
710
使能基础设施
设备不可知API
服务执行环境

图 7

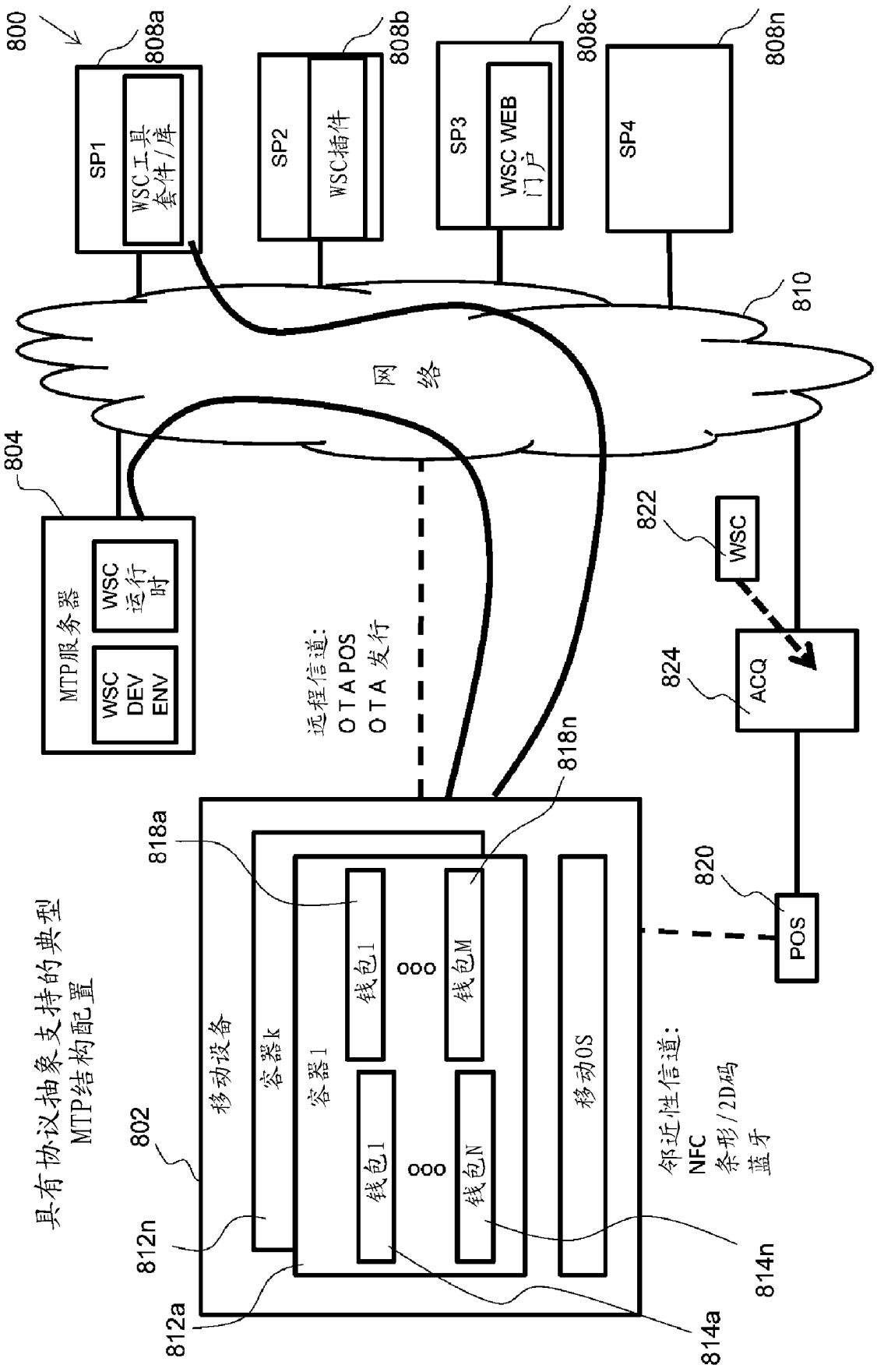

具有协议抽象支持的典型
MTP结构配置
800
808a
SP1
WSC工具
套件/库
808b
SP2
WSC插件
808c
SP3
WSC WEB
门户
808n
SP4
810
网 络
804
MTP服务器
WSC
运行
时
WSC
DEV
ENV
远程信道:
OTA POS
OTA 发行
822
WSC
824
ACQ
818n
818a
820
POS
802
移动设备
容器k
容器1
钱包1
ooo
钱包M
钱包1
ooo
钱包N
移动OS
邻近性信道:
NFC
条形/2D码
蓝牙
812n
812a
814n
814a

图 8

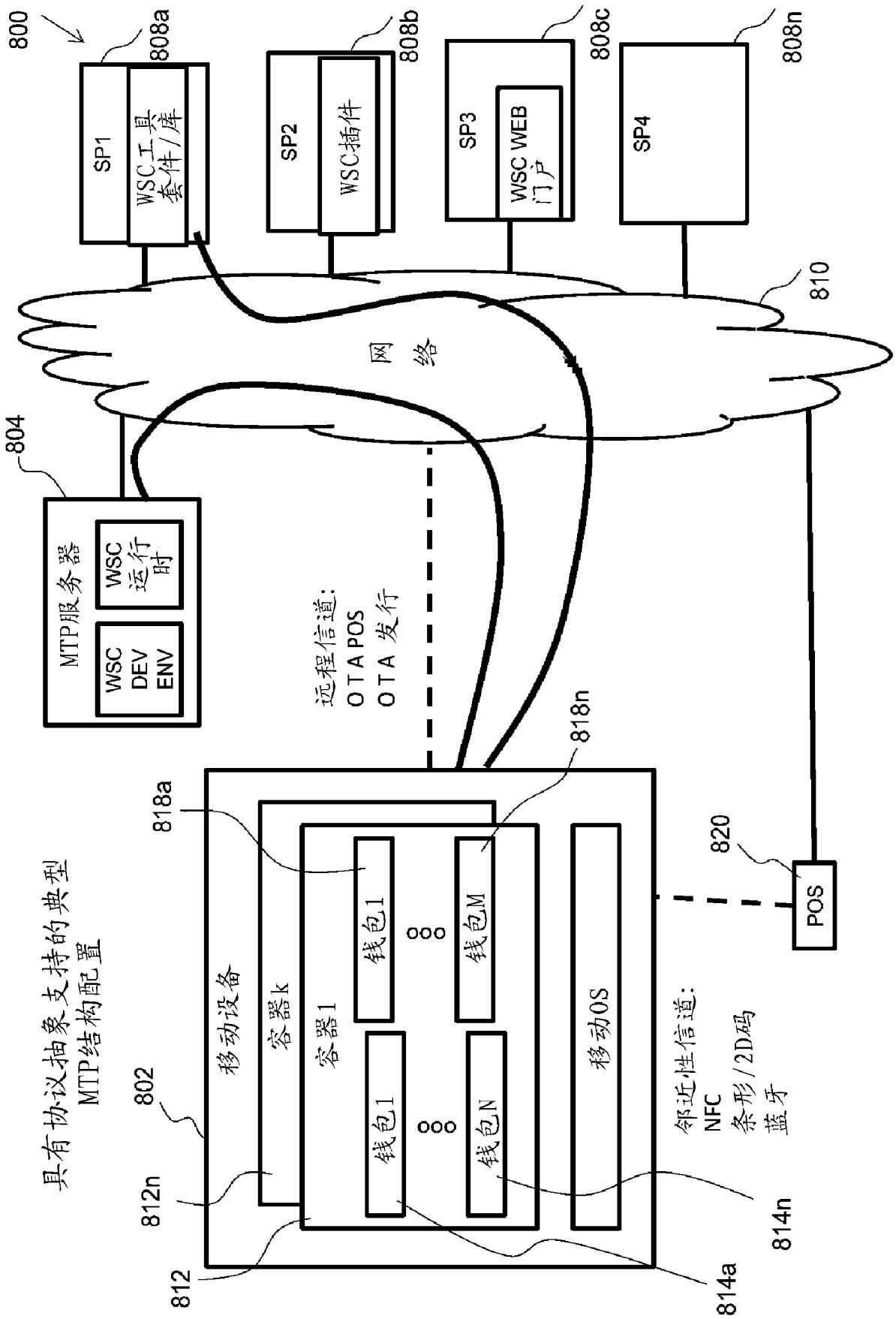

具有协议抽象支持的典型
MTP结构配置
800
808a
SP1
WSC工具
套件/库
808b
SP2
WSC插件
808c
SP3
WSC WEB
门户
808n
SP4
810
网 络
804
MTP服务器
WSC
运行
时
WSC
DEV
ENV
远程信道:
OTA POS
OTA 发行
818n
818a
820
POS
802
移动设备
容器k
容器1
钱包1
钱包M
钱包1
钱包N
移动OS
812n
812
814n
814a
邻近性信道:
NFC
条形/2D码
蓝牙

图 9

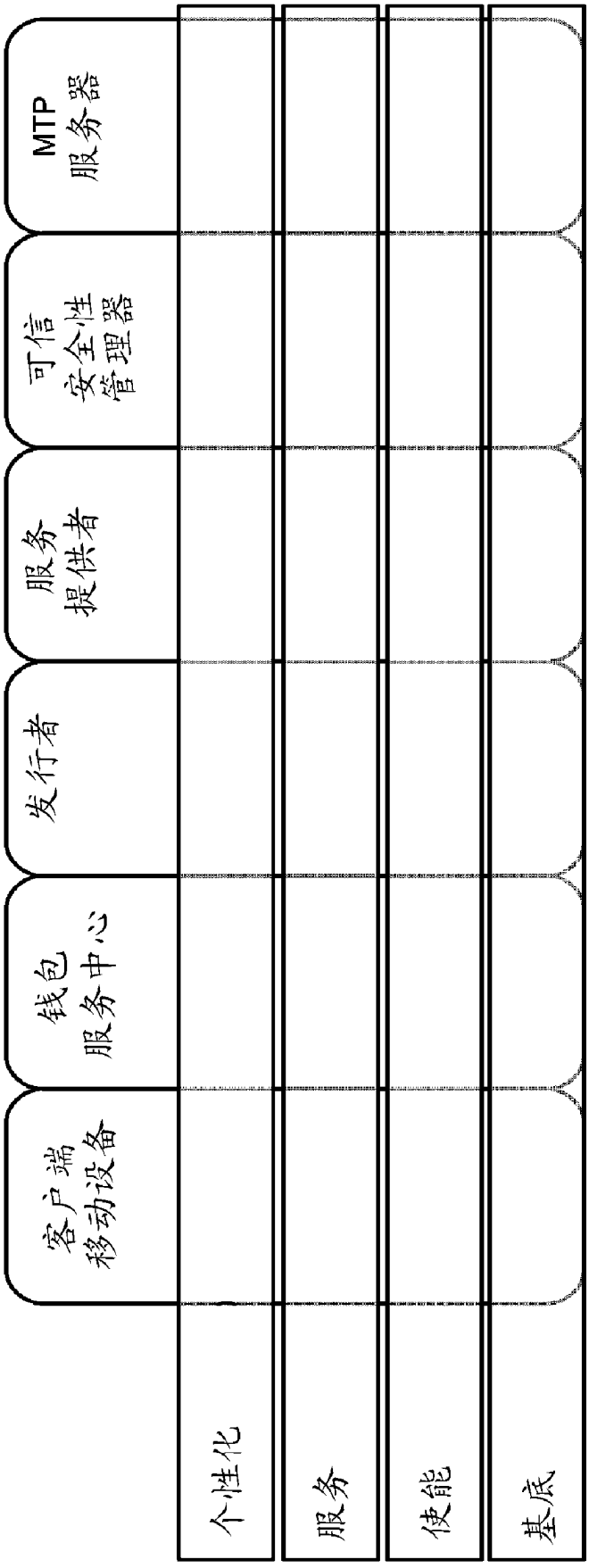


图 10

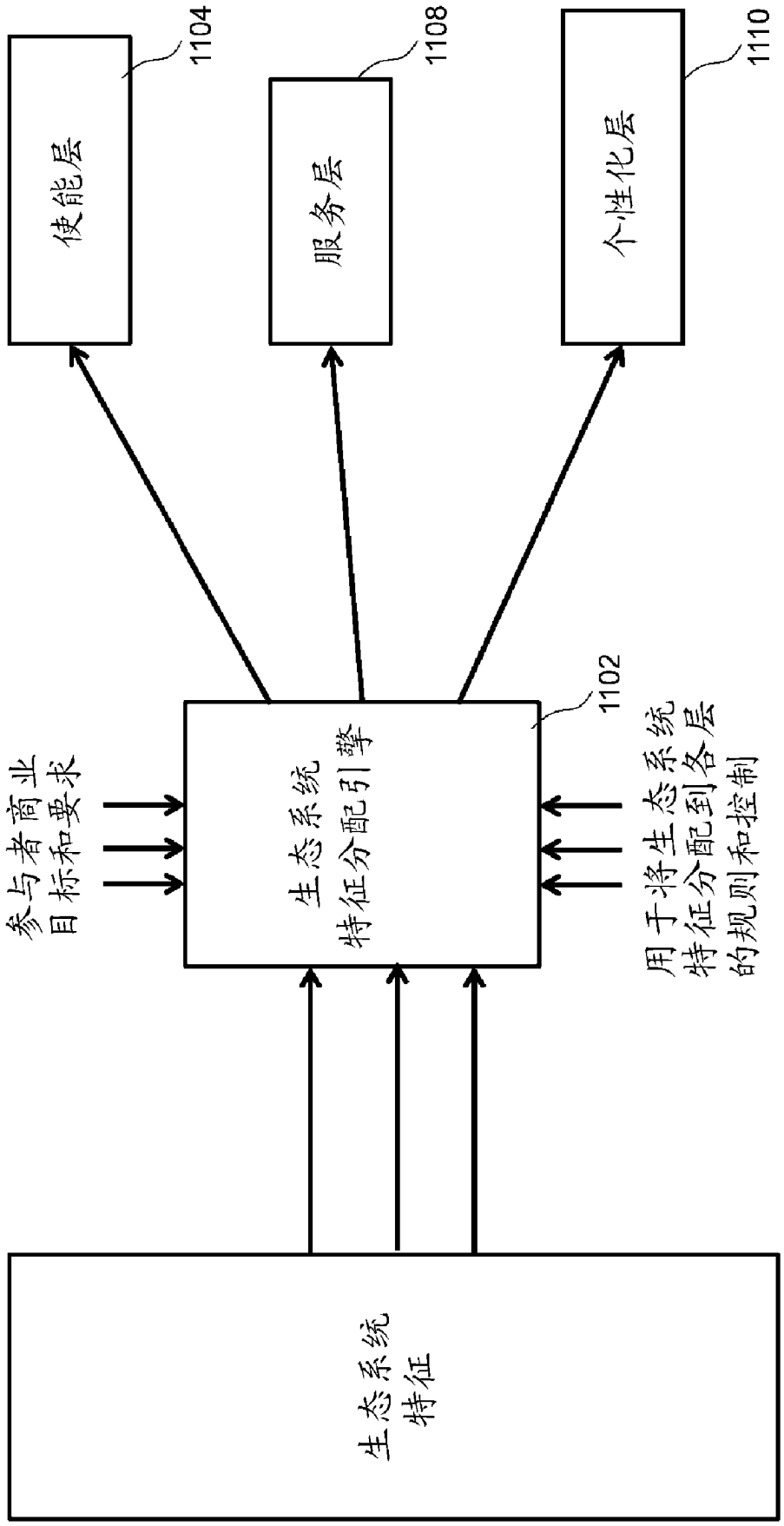

使能层
1104
服务层
1108
个性化层
1110
生态系统
特征分配引擎
1102
参与者商业
目标和要求
用于将生态系统
特征分配到各层
的规则和控制
生态系统
特征

图 11

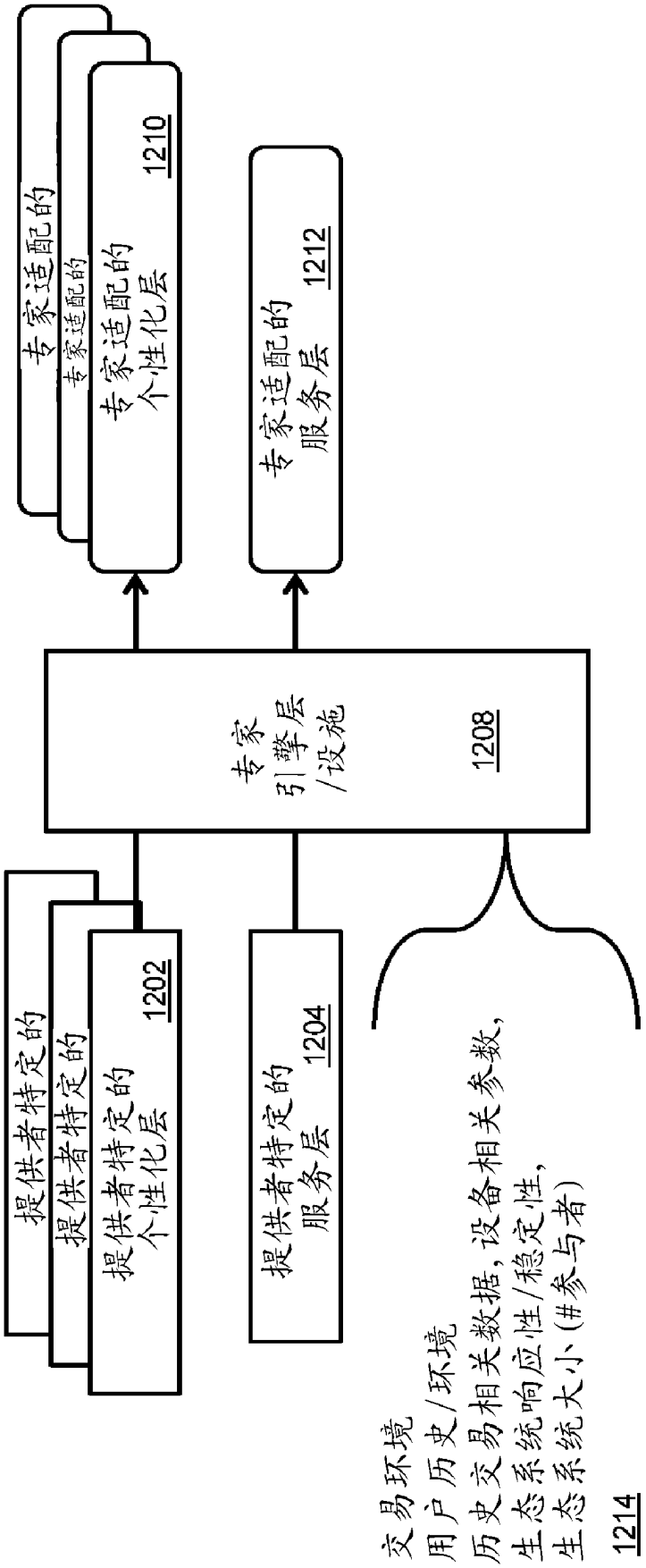

提供者特定的
提供者特定的
提供者特定的
个性化层
1202
提供者特定的
服务层
1204
专家
引擎层
/设施
1208
专家适配的
专家适配的
专家适配的
个性化层
1210
专家适配的
服务层
1212
交易环境
用户历史/环境
历史交易相关数据，设备相关参数，
生态系统响应性/稳定性，
生态系统大小(#参与者)
1214

图 12

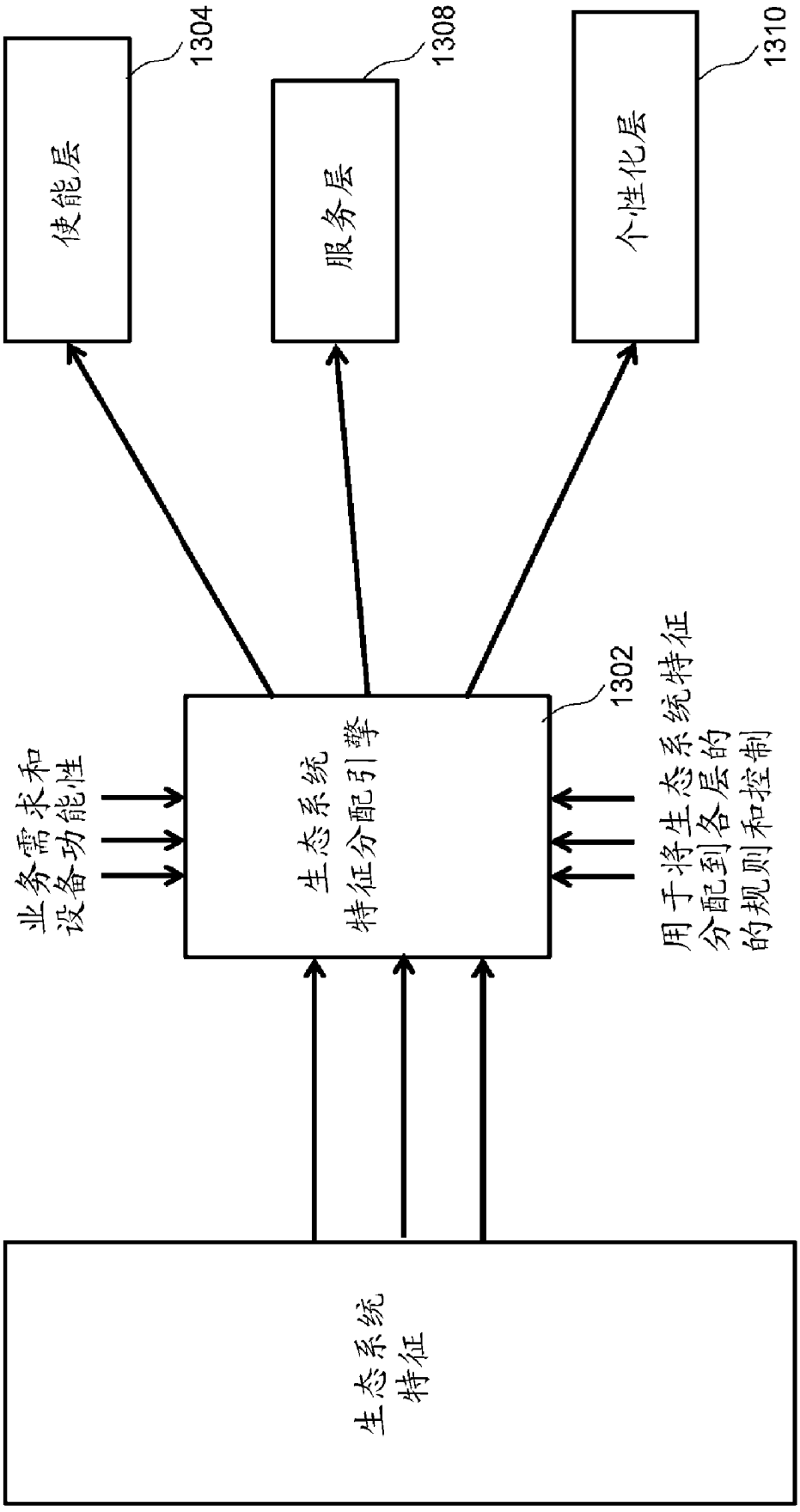

使能层
1304
服务层
1308
个性化层
1310
生态系统
特征分配引擎
1302
业务需求和
设备功能性
用于将生态系统特征
分配到各层的
的规则和控制
生态系统
特征

图 13

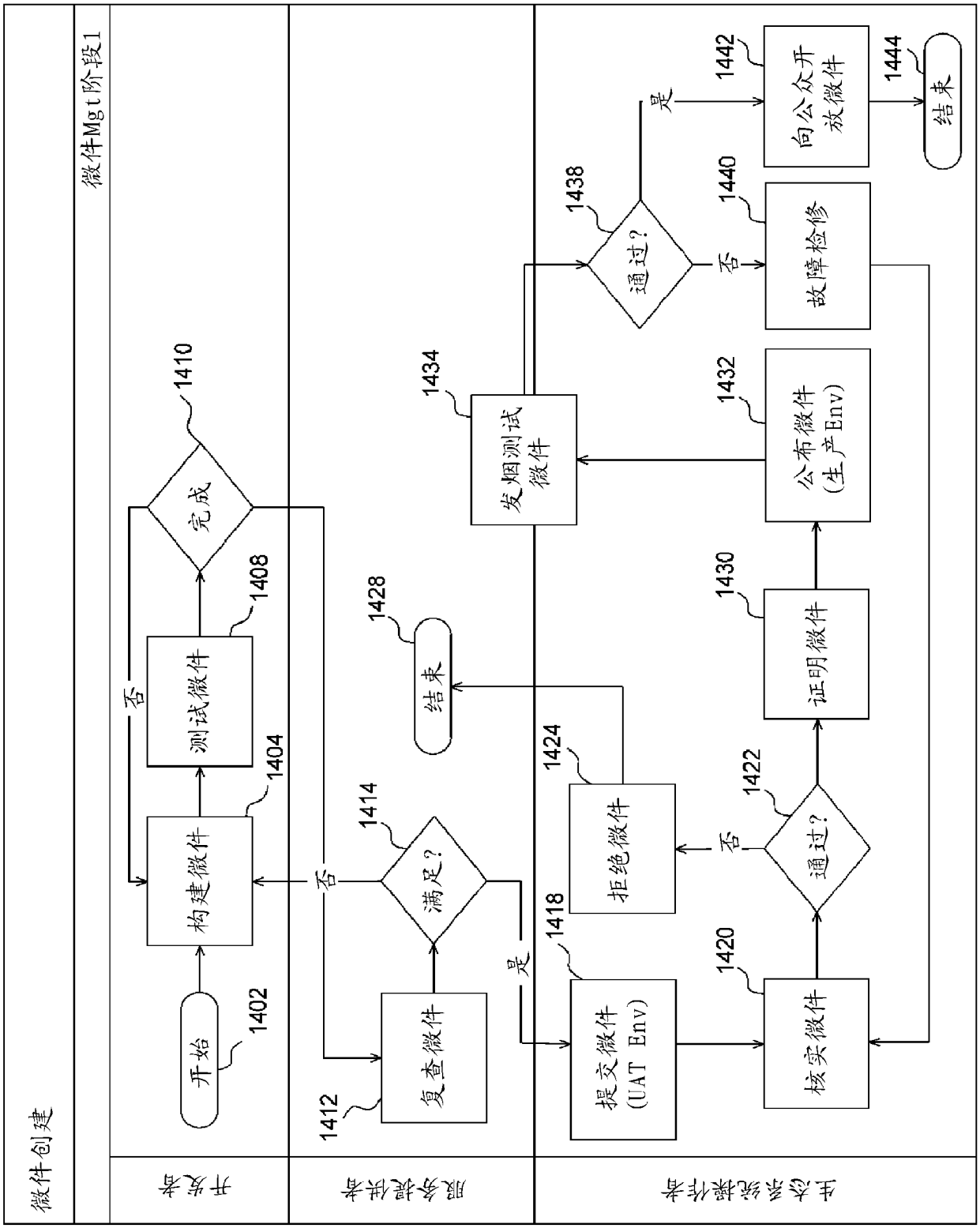

微件创建
微件Mgt阶段1
开发者
服务提供者
生态系统操作者
开始
1402
构建微件
1404
测试微件
1408
完成
1410
否
复查微件
1412
满足？
1414
否
是
结束
1428
提交微件 (UAT Env)
1418
拒绝微件
1424
发烟测试微件
1434
通过？
1438
否
是
故障检修
1440
核实微件
1420
通过？
1422
否
证明微件
1430
公布微件 (生产Env)
1432
向公众开放微件
1442
结束
1444

图 14

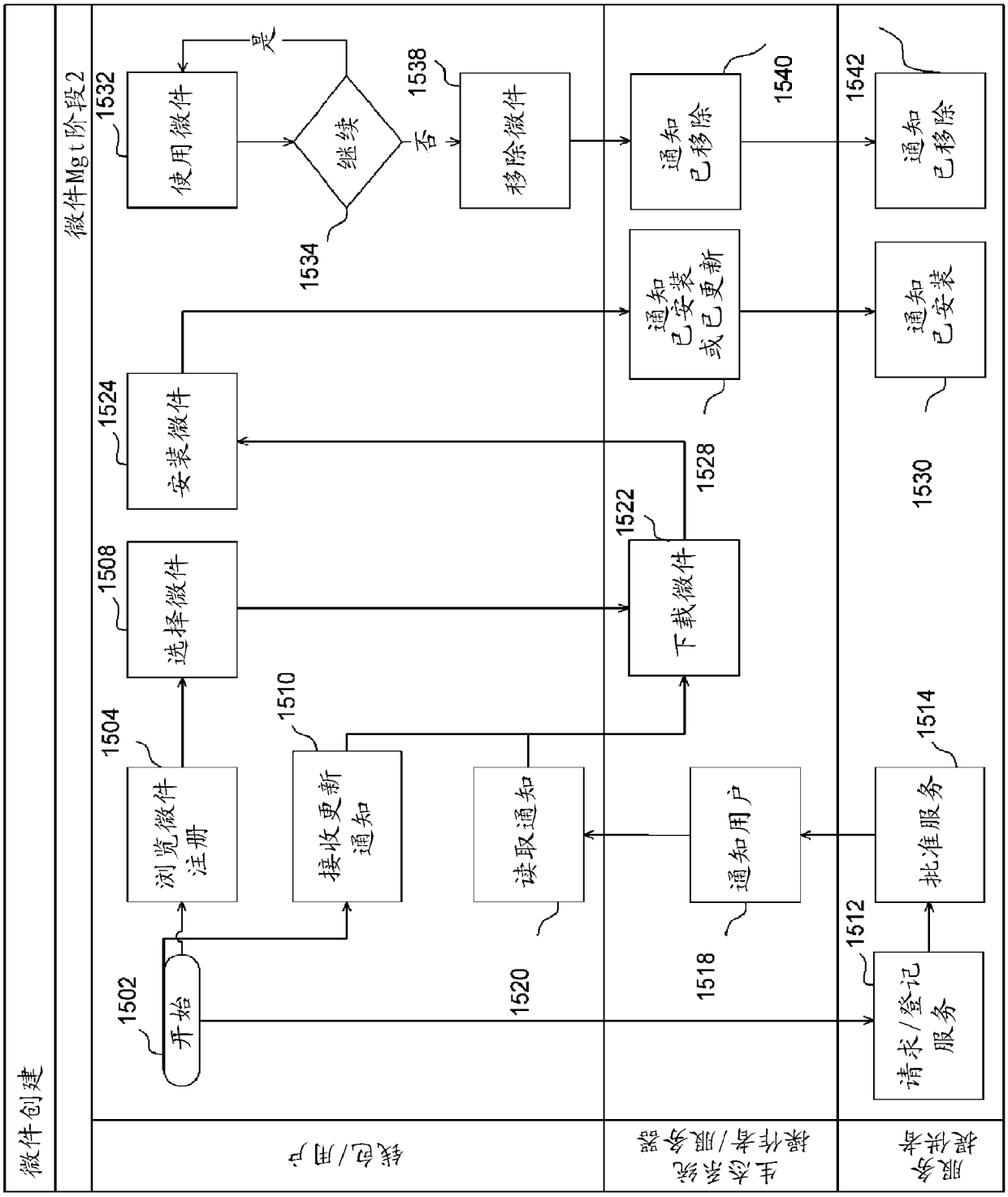

微件创建
微件Mgt阶段2
线包/用户
主体系统/服务器
服务/主体
开始
1502
浏览微件注册
1504
选择微件
1508
接收更新通知
1510
读取通知
1520
安装微件
1524
使用微件
1532
继续
1534
是
否
移除微件
1538
通知用户
1518
下载微件
1522
1528
通知已安装或已更新
1540
通知已移除
1542
1530
请求/登记服务
1512
批准服务
1514

图 15

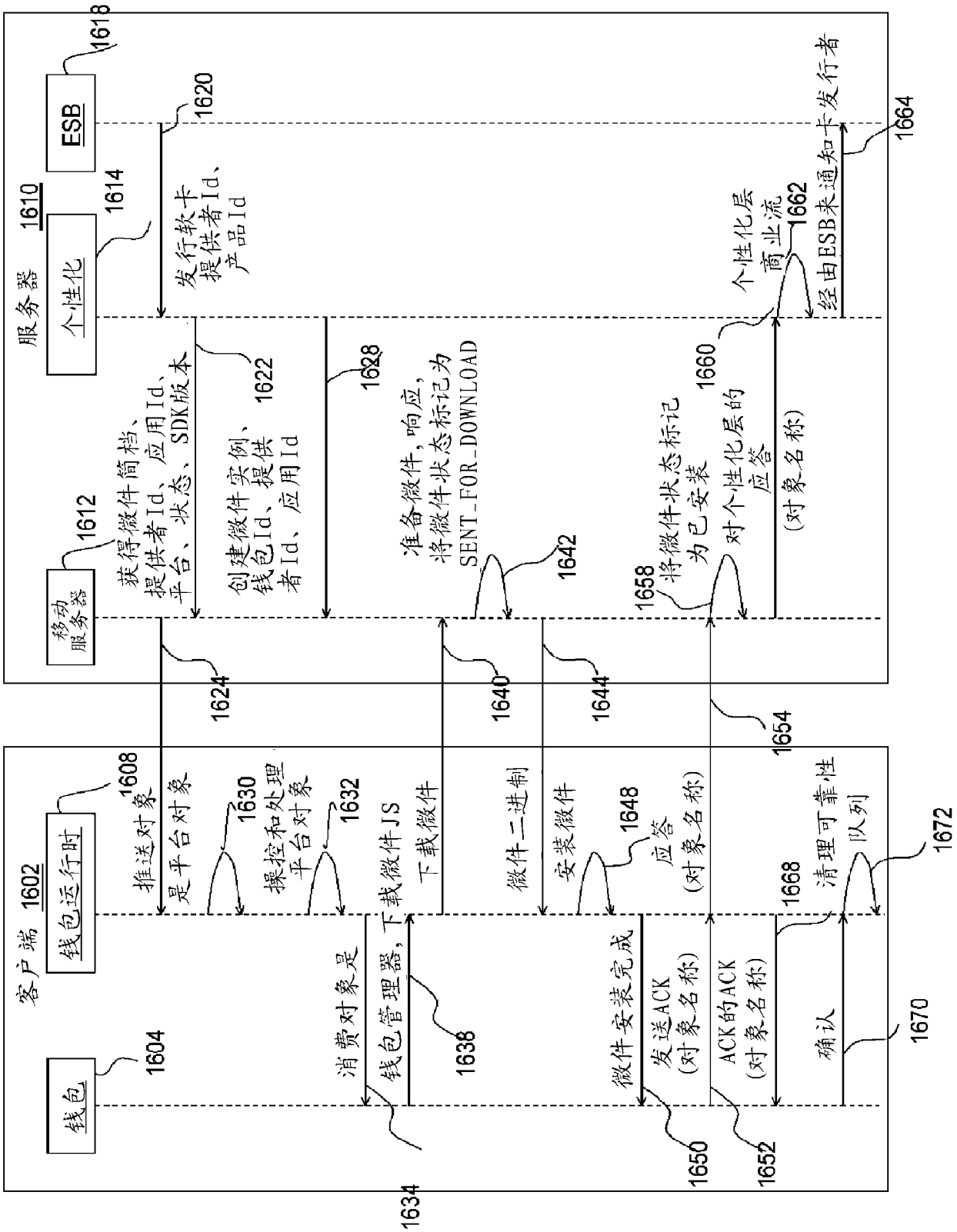

客户端 1602
钱包 1604
钱包运行时 1608
服务器 1610
移动服务器 1612
个性化 1614
ESB 1618
发行软卡
提供者Id、
产品Id 1620
获得微件简档、
提供者Id、应用Id、
平台、状态、SDK版本 1622
创建微件实例、
钱包Id、提供
者Id、应用Id 1628
准备微件，响应，
将微件状态标记为
SENT_FOR_DOWNLOAD 1642
将微件状态标记
为已安装 1658
对个性化层的
应答
(对象名称) 1660
个性化层
商业流 1662
经由ESB来通知卡发行者 1664
推送对象 1624
是平台对象 1630
操控和处理
平台对象 1632
下载微件JS 1640
微件二进制 1644
安装微件 1648
应答
(对象名称) 1654
消费对象是
钱包管理器，下载微件 1638
微件安装完成
发送ACK
(对象名称) 1650
ACK的ACK
(对象名称) 1652
确认 1670
清理可靠性
队列 1668
1672
1634

图 16

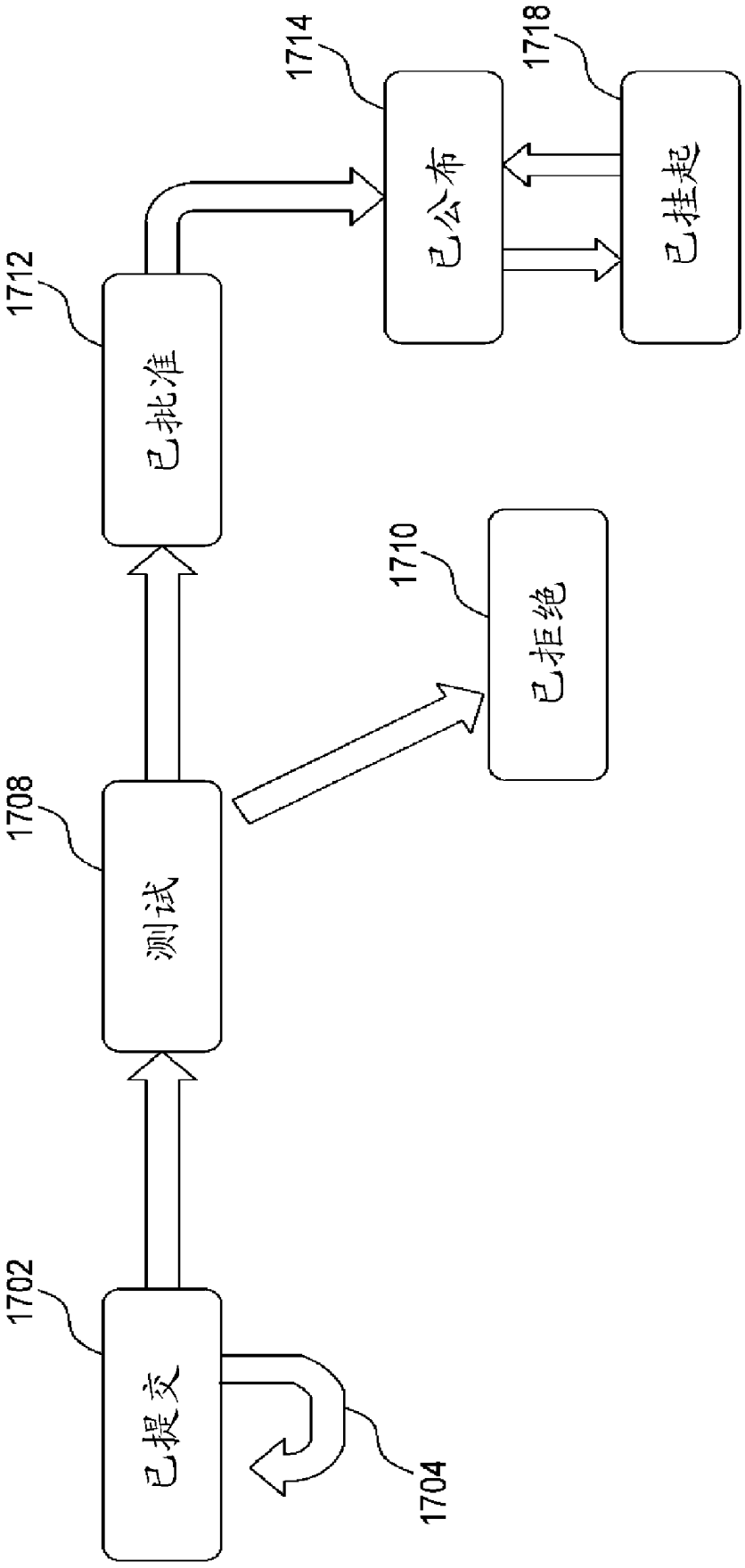


图 17

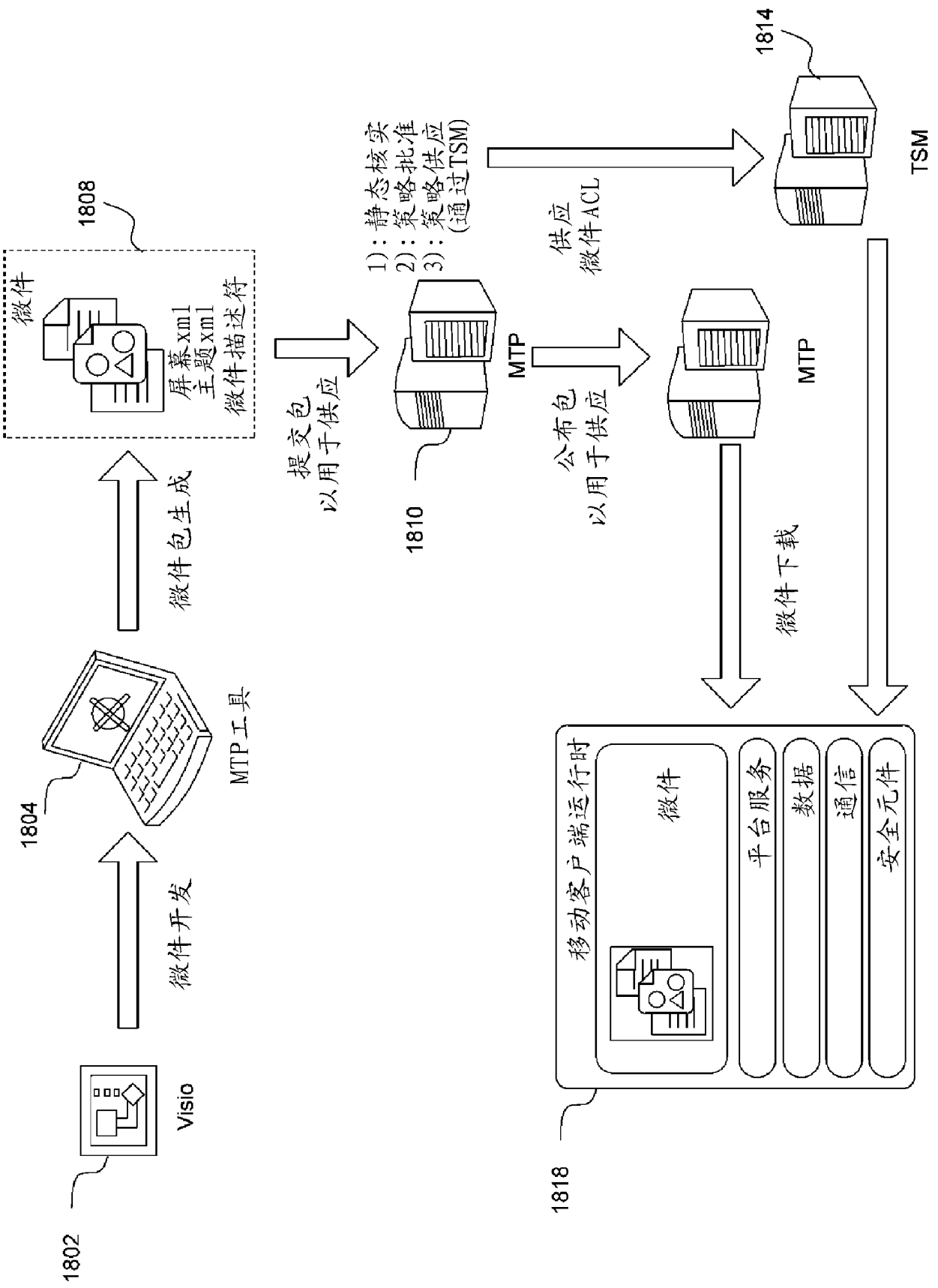

1802
Visio
微件开发
1804
MTP工具
微件包生成
1808
微件
屏幕xml
主题xml
微件描述符
提交包
以用于供应
1810
MTP
1)：静态核实
2)：策略批准
3)：策略供应
(通过TSM)
公布包
以用于供应
供应
微件ACL
1814
TSM
MTP
微件下载
1818
移动客户端运行时
微件
平台服务
数据
通信
安全元件

图 18

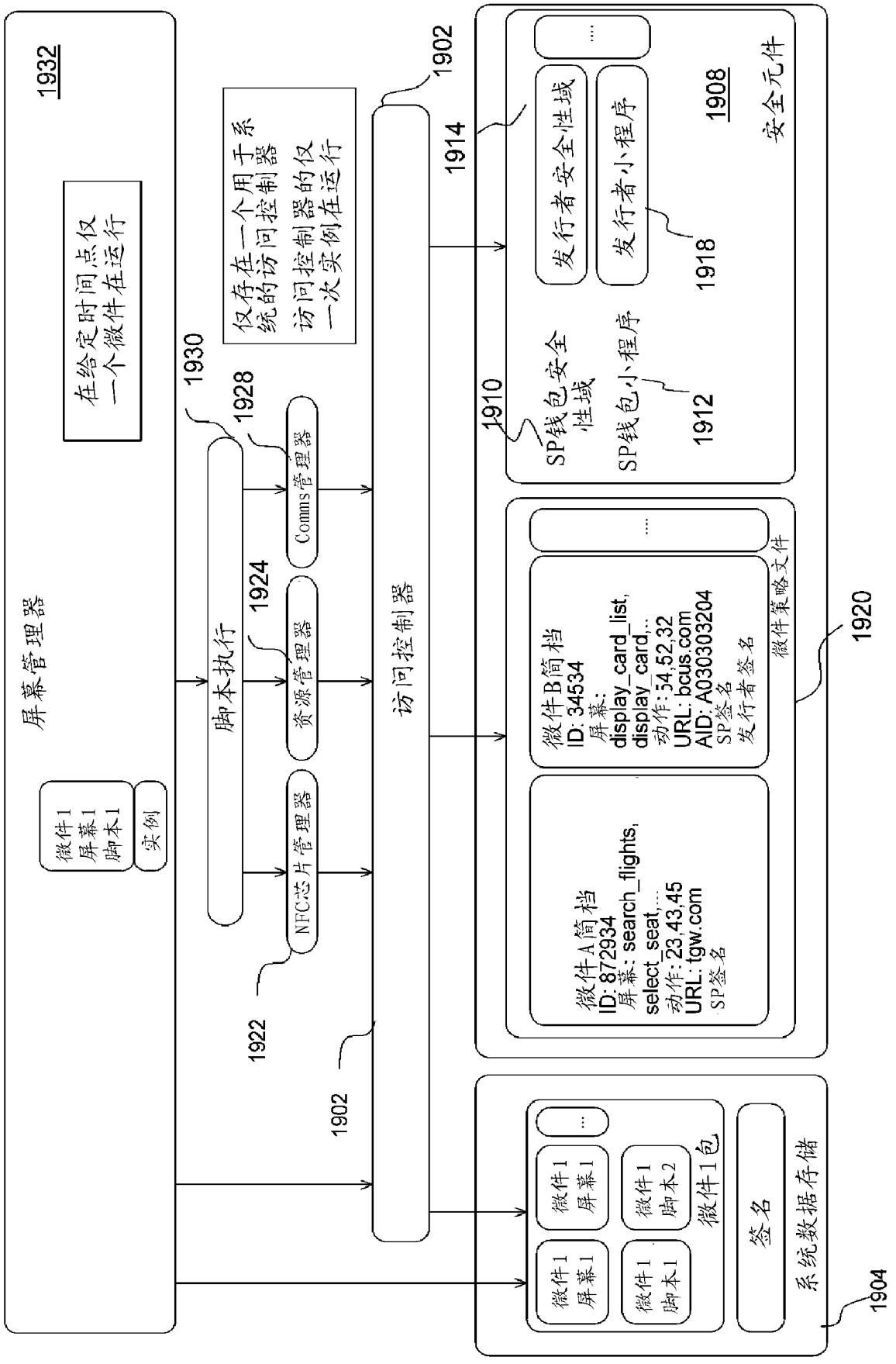

屏幕管理器
1932
在给定时间点仅一个微件在运行
微件1 屏幕1 脚本1
实例
1930
脚本执行
1928
Comms管理器
1924
资源管理器
NFC芯片管理器
1922
仅存在一个用于系统的访问控制器的仅一次实例在运行
访问控制器
1902
1902
1914
安全元件
1910
SP钱包安全性域
发行者安全性域
SP钱包小程序
发行者小程序
1912
1918
1908
微件B简档
ID: 34534
屏幕:
display_card_list,
display_card,...
动作:34,52,32
URL: bcus.com
AID: A030303204
SP签名
发行者签名
微件A简档
ID: 872934
屏幕: search_flights,
select_seat,...
动作:23,43,45
URL: tgw.com
SP签名
微件策略文件
1920
微件1 屏幕1
微件1 脚本1
微件1 屏幕1
微件1 脚本2
微件1包
签名
系统数据存储
1904

图 19

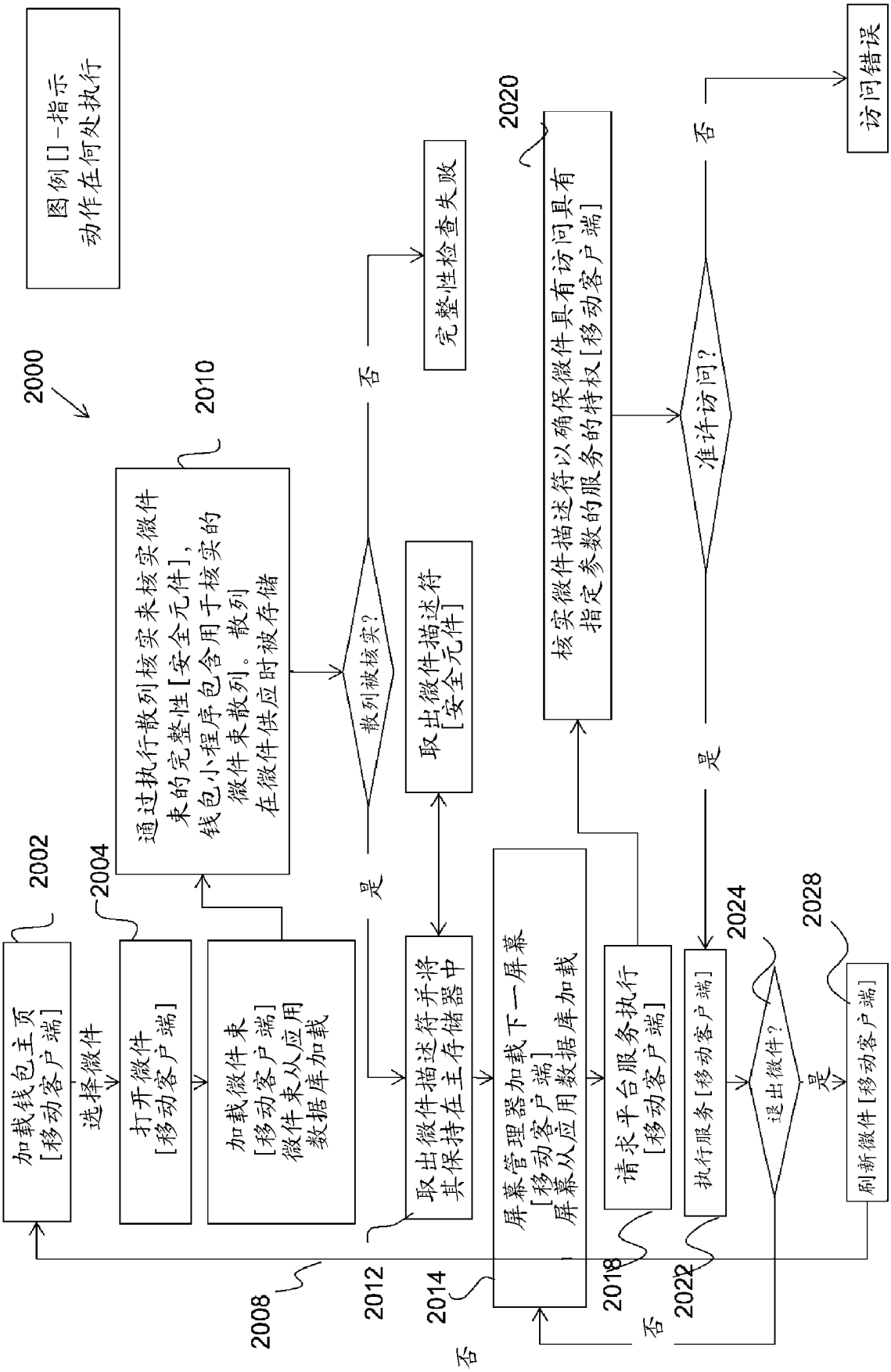

图例[]-指示
动作在何处执行
2000
加载钱包主页
[移动客户端]
2002
选择微件
打开微件
[移动客户端]
2004
加载微件来
[移动客户端]
微件来从应用
数据库加载
2008
通过执行散列核实来核实微件
来的完整性[安全元件],
钱包小程序包含用于核实的
微件来散列。散列
在微件供应时被存储
2010
散列被核实?
否
完整性检查失败
是
取出微件描述符
[安全元件]
取出微件描述符并将
其保持在主存储器中
2012
屏幕管理器加载下一屏幕
[移动客户端]
屏幕从应用数据库加载
2014
否
请求平台服务执行
[移动客户端]
2018
执行服务[移动客户端]
2022
退出微件?
否
是
2024
刷新微件[移动客户端]
2028
核实微件描述符以确保微件具有访问具有
指定参数的服务的特权[移动客户端]
2020
准许访问?
否
访问错误
是

图 20

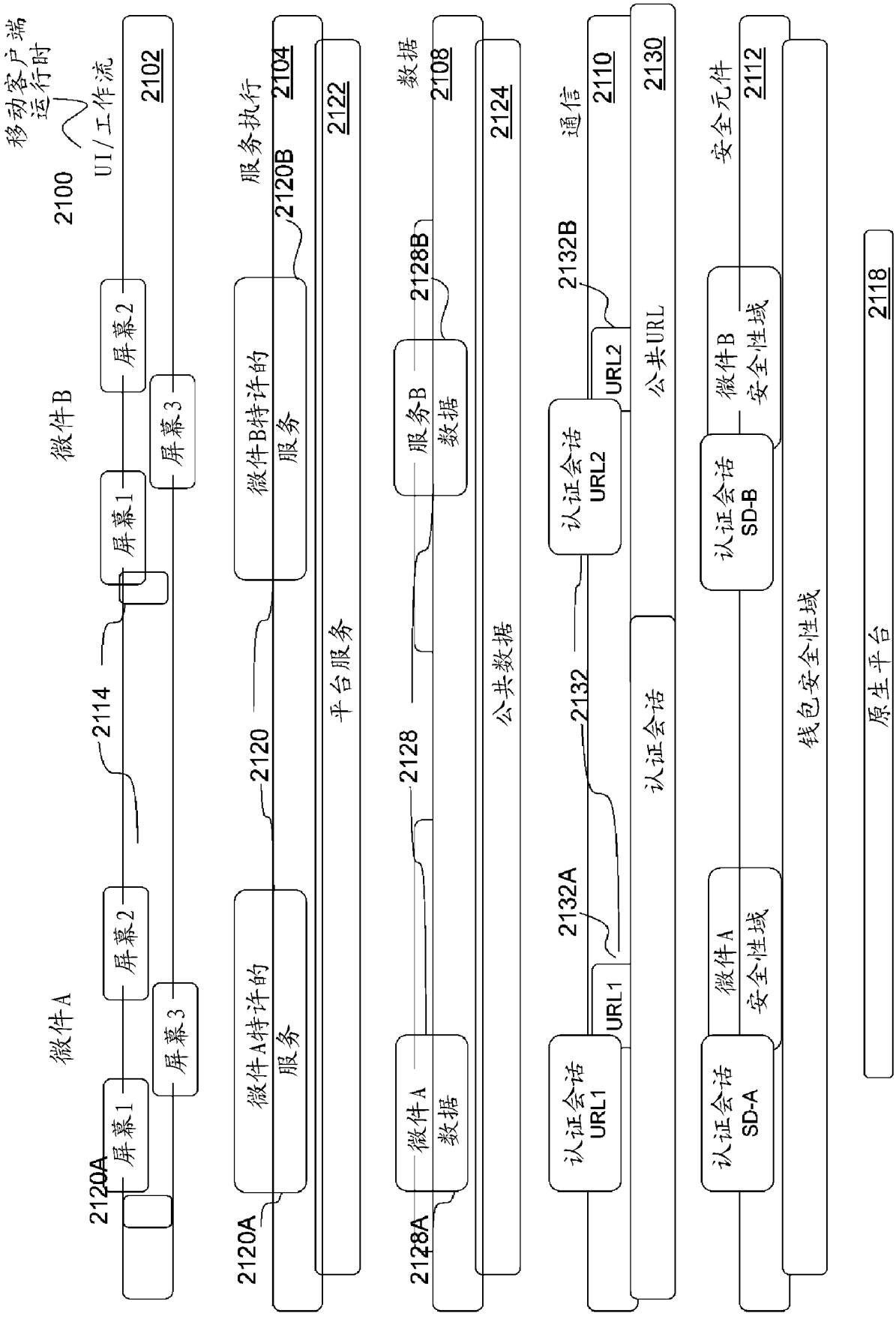

移动客户端运行时
2100
UI/工作流
2102
微件A
屏幕1
屏幕2
屏幕3
2120A
2114
微件B
屏幕1
屏幕2
屏幕3
2120
微件A特许的服务
2120A
平台服务
2122
服务执行
2104
微件B特许的服务
2120B
数据
2108
服务B数据
2128B
2128
微件A数据
2128A
公共数据
2124
通信
2110
认证会话URL1
URL1
2132A
2132
认证会话
认证会话URL2
URL2
2132B
公共URL
2130
安全元件
2112
认证会话SD-A
微件A安全性域
认证会话SD-B
微件B安全性域
钱包安全性域
原生平台
2118

图 21

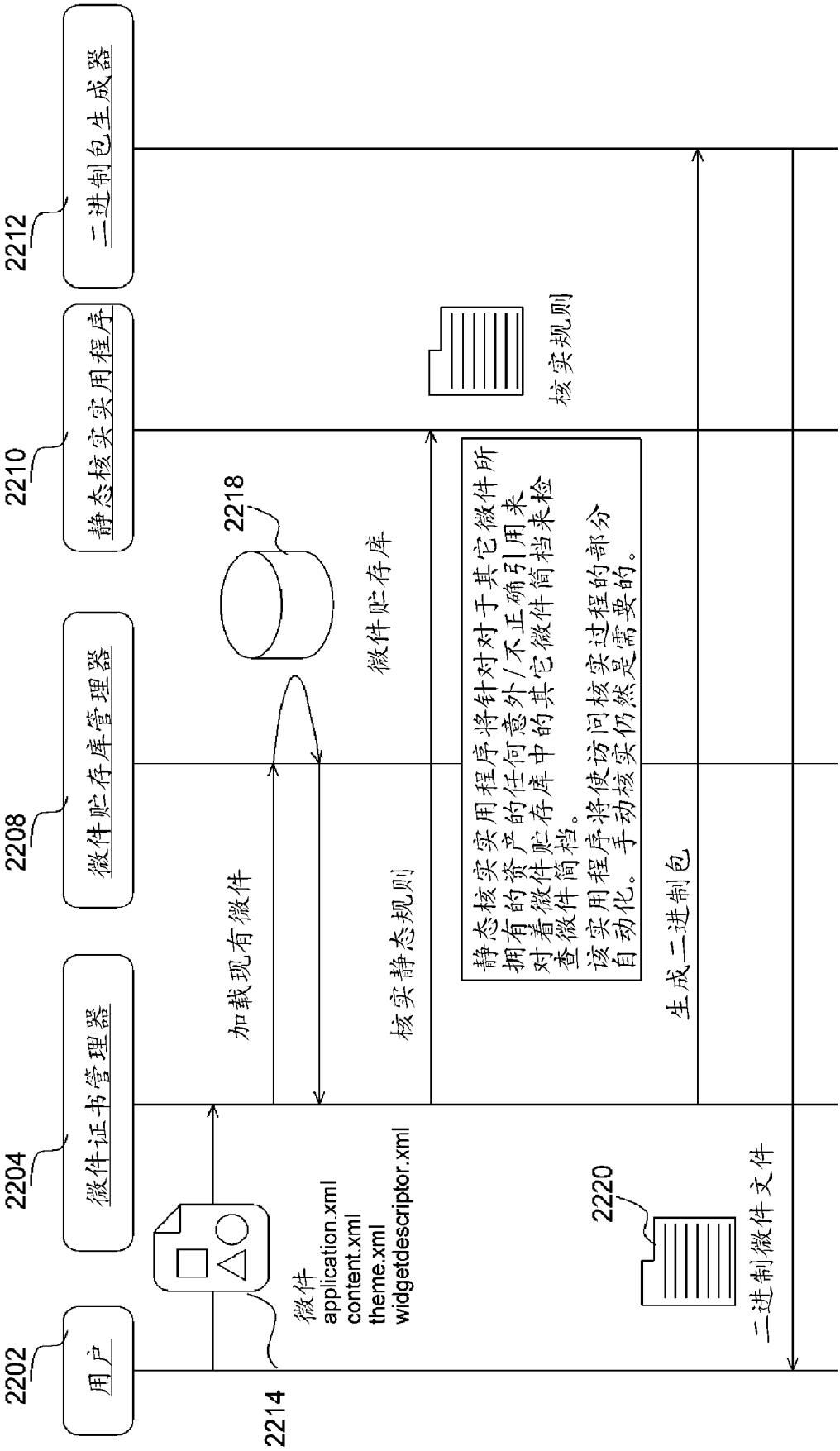

2202
用户
2204
微件证书管理器
2208
微件贮存库管理器
2210
静态核实实用程序
2212
二进制包生成器
2214
微件
application.xml
content.xml
theme.xml
widgetdescriptor.xml
加载现有微件
核实静态规则
2218
微件贮存库
静态核实实用程序将针对对于其它微件所拥有的资产的任何意外/不正确引用来对着微件贮存库中的其它微件简档来检查微件简档。
该实用程序将使访问核实过程的部分自动化。手动核实仍然是需要的。
核实规则
生成二进制包
2220
二进制微件文件

图 22

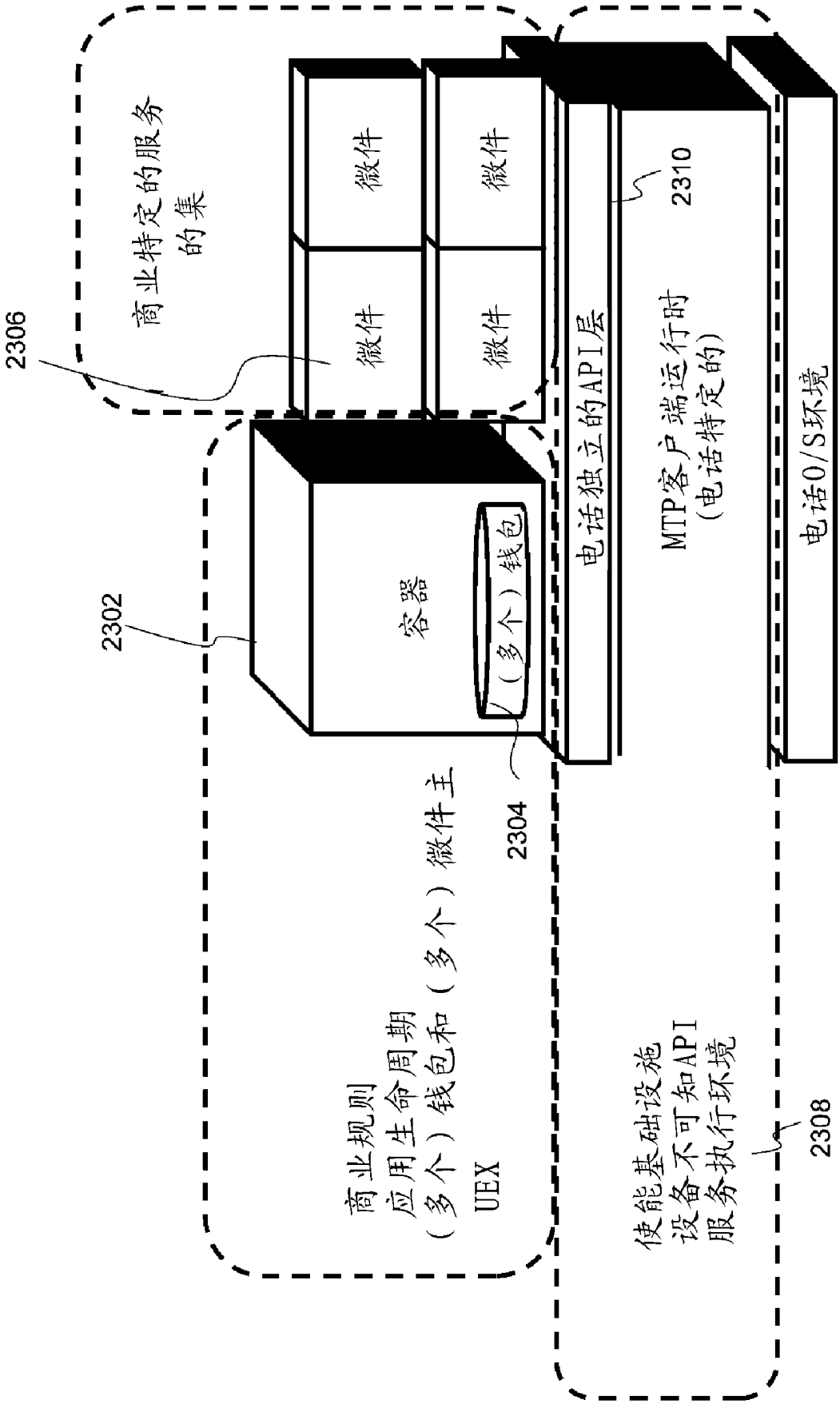

2306
商业特定的服务的集
微件
微件
微件
微件
2310
电话独立的API层
MTP客户端运行时（电话特定的）
电话O/S环境
2302
容器
（多个）钱包
2304
商业规则
应用生命周期
（多个）钱包和（多个）微件主
UEX
使能基础设施
设备不可知API
服务执行环境
2308

图 23

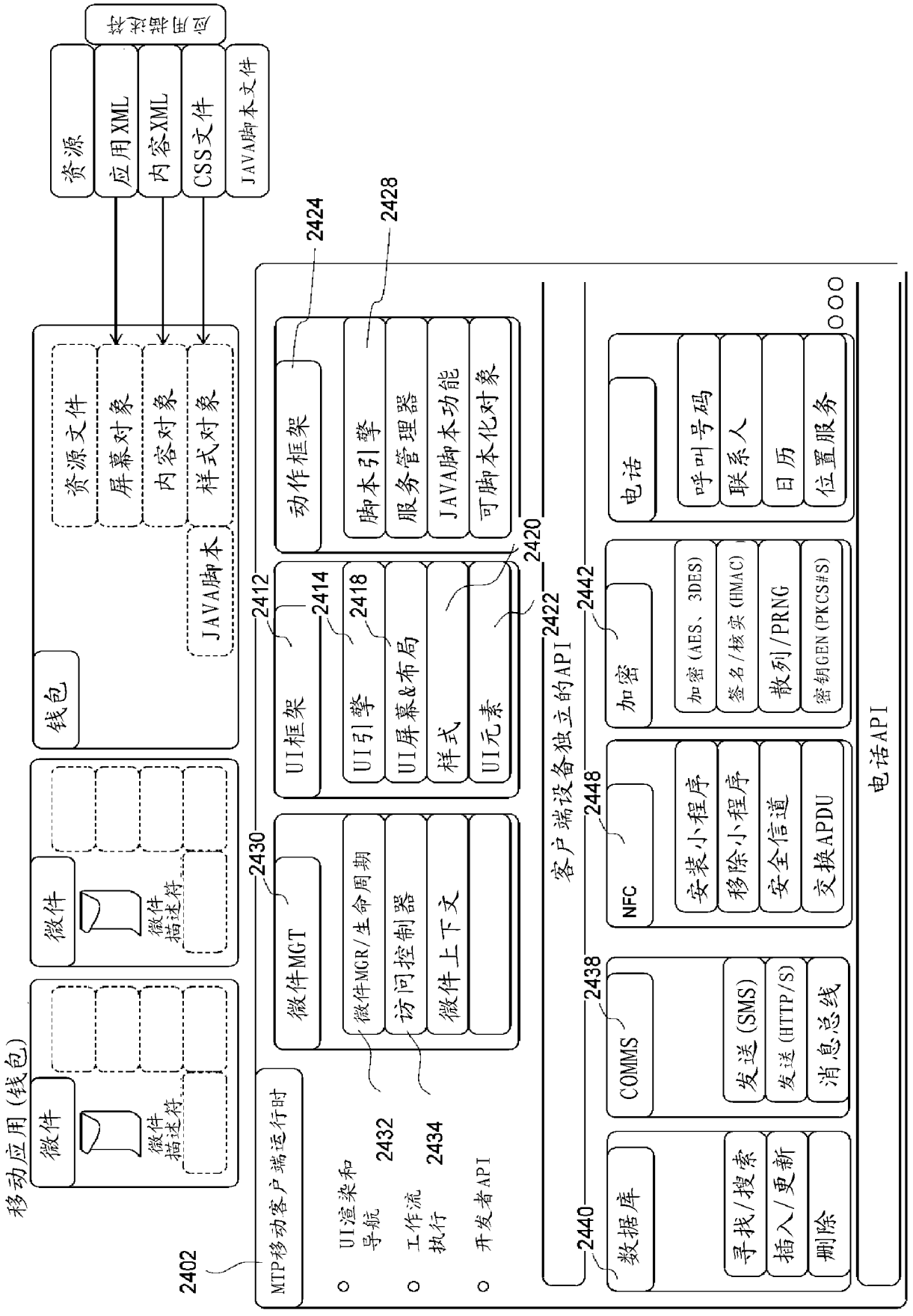

移动应用(钱包)
微件
微件描述符
微件
微件描述符
钱包
资源文件
屏幕对象
内容对象
样式对象
JAVA脚本
资源
应用XML
内容XML
CSS文件
JAVA脚本文件
其他资源
2402
MTP移动客户端运行时
UI渲染和导航 2432
工作流执行 2434
开发者API
2430
微件MGT
微件MGR/生命周期
访问控制器
微件上下文
2412
UI框架
2414
UI引擎
2418
UI屏幕&布局
样式
UI元素
2420
2424
动作框架
2428
脚本引擎
服务管理器
JAVA脚本功能
可脚本化对象
客户端设备独立的API 2422
2440
数据库
寻找/搜索
插入/更新
删除
2438
COMMS
发送(SMS)
发送(HTTP/S)
消息总线
2448
NFC
安装小程序
移除小程序
安全信道
交换APDU
2442
加密
加密(AES、3DES)
签名/核实(HMAC)
散列/PRNG
密钥GEN(PKCS#S)
电话
呼叫号码
联系人
日历
位置服务
电话API

图 24

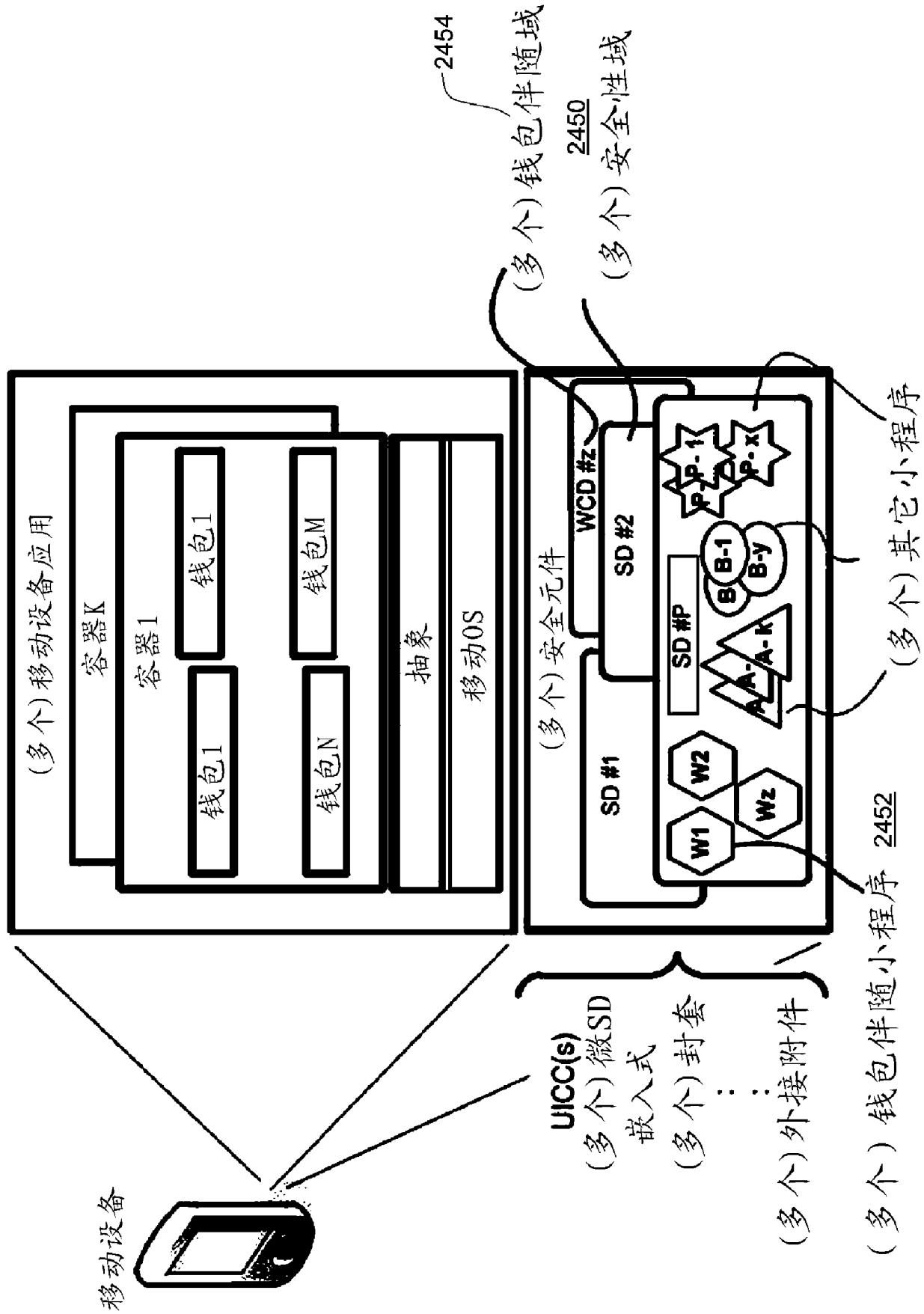

移动设备
(多个)移动设备应用
容器K
容器1
钱包1
钱包1
钱包M
钱包N
抽象
移动OS
2454
(多个)钱包伴随域
2450
(多个)安全性域
(多个)安全元件
WCD #z
SD #2
SD #1
SD #P
W1
W2
Wz
A-1
A-k
B-1
B-y
P-1
P-x
(多个)其它小程序
UICC(s)
(多个)微SD
嵌入式
(多个)封套
(多个)外接附件
(多个)钱包伴随小程序 2452

图 24A

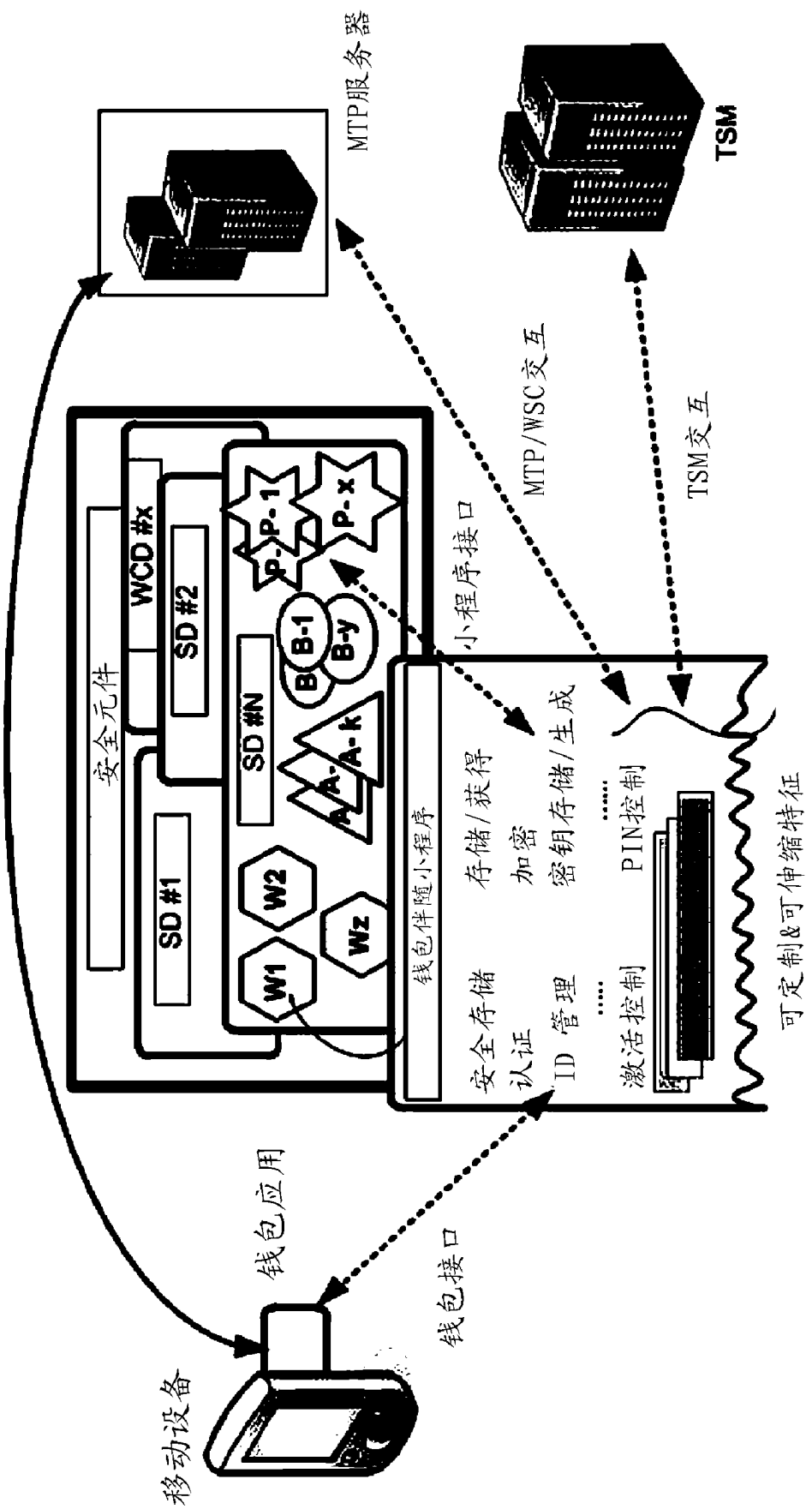

MTP服务器
TSM
安全元件
WCD #x
SD #2
SD #1
SD #N
P-1
P-x
B-1
B-y
A-k
W1
W2
Wz
钱包伴随小程序
小程序接口
MTP/WSC交互
TSM交互
安全存储
认证
ID 管理
.....
激活控制
存储/获得
加密
密钥存储/生成
.....
PIN控制
.....
可定制&可伸缩特征
移动设备
钱包应用
钱包接口

图 24B

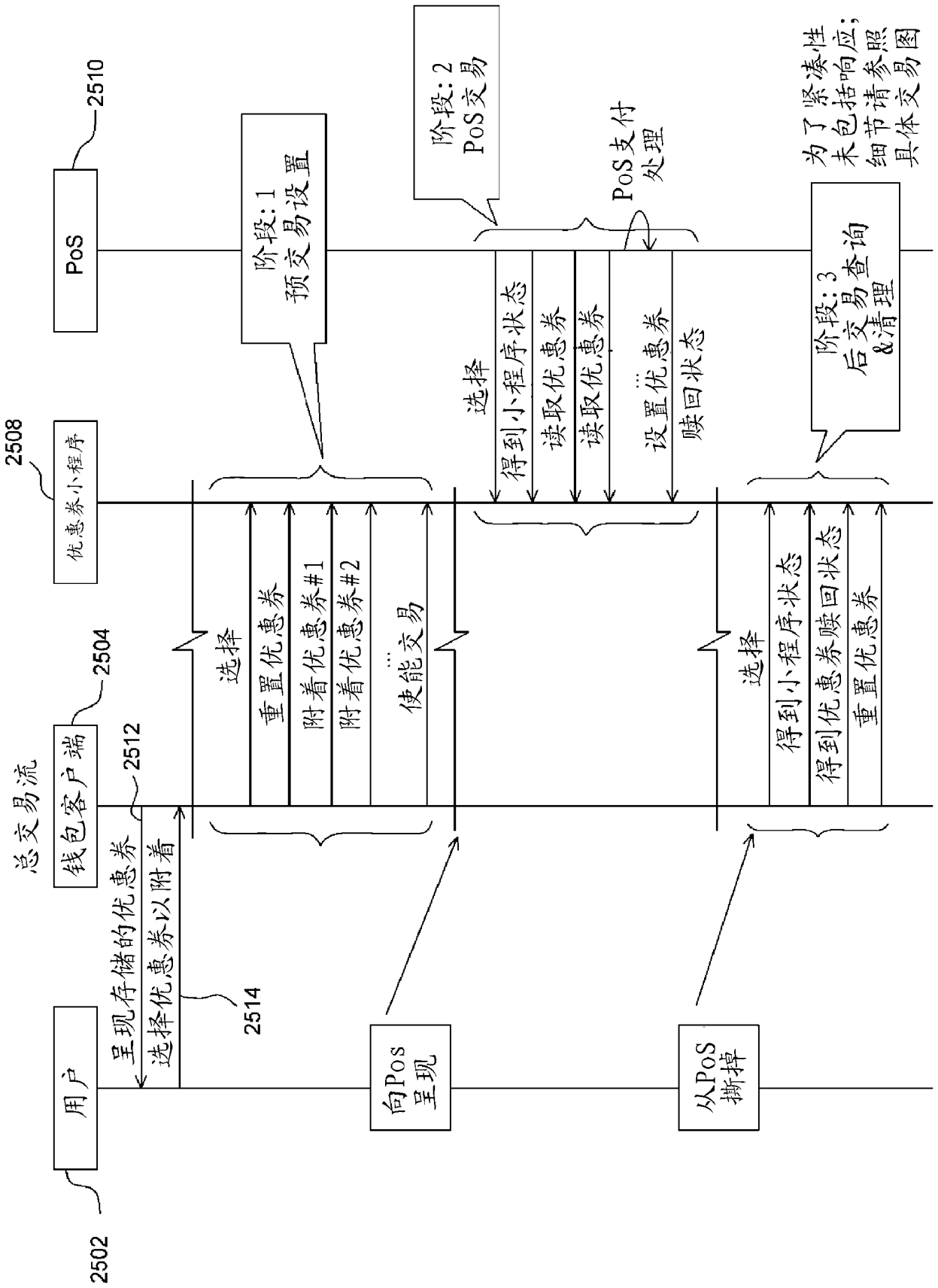

总交易流
用户
2502
钱包客户端
2504
2512
优惠券小程序
2508
PoS
2510
呈现存储的优惠券
选择优惠券以附着
2514
选择
重置优惠券
附着优惠券#1
附着优惠券#2
...
使能交易
阶段:1
预交易设置
向Pos
呈现
阶段:2
PoS交易
选择
得到小程序状态
读取优惠券
读取优惠券
...
设置优惠券
赎回状态
PoS支付
处理
从PoS
撕掉
选择
得到小程序状态
得到优惠券赎回状态
重置优惠券
阶段:3
后交易查询
&清理
为了紧凑性
未包括响应;
细节请参照
具体交易图

图 25

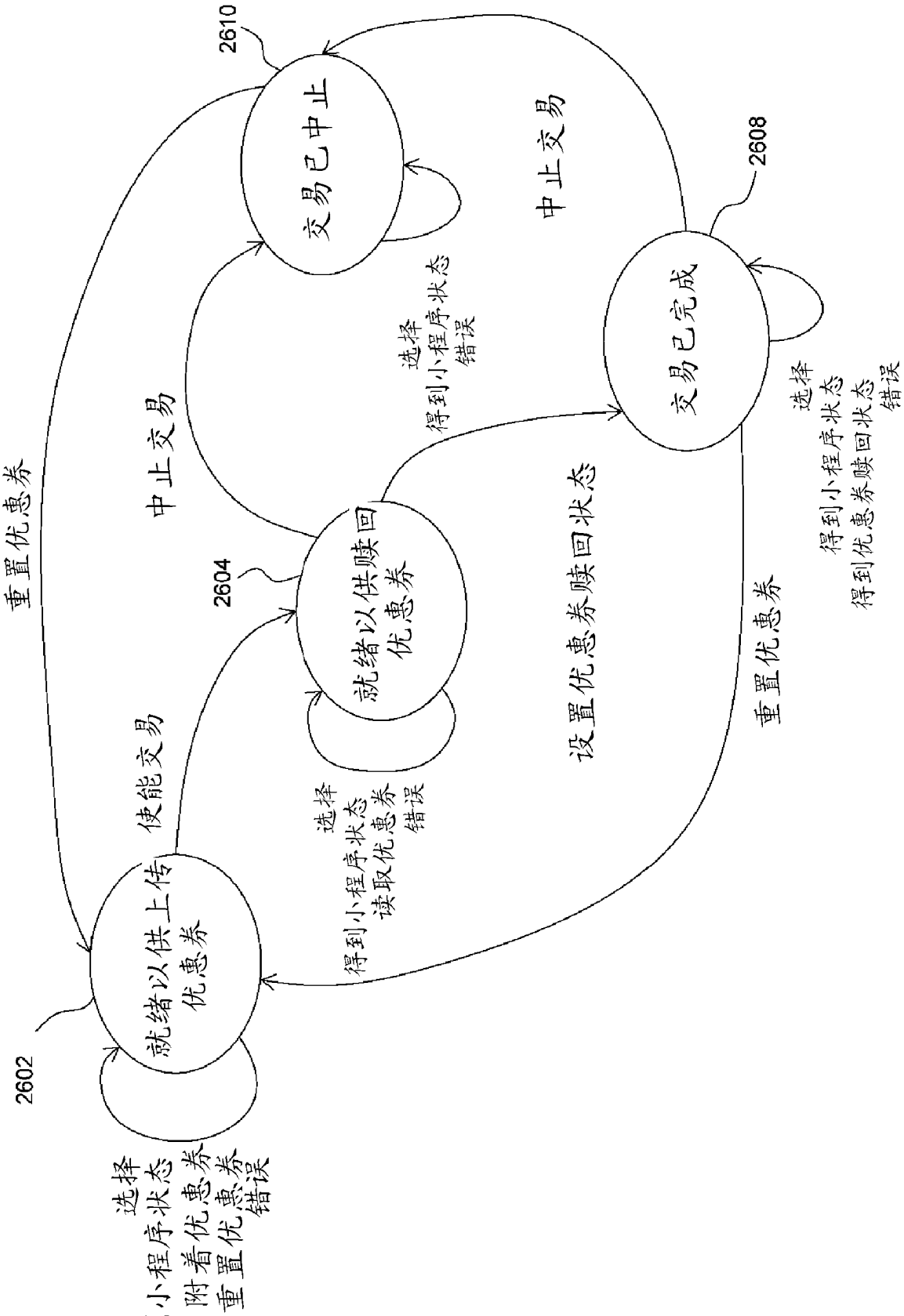

2602
就绪以供上传优惠券
选择
得到小程序状态
附着优惠券
重置优惠券
错误
使能交易
2604
就绪以供赎回优惠券
选择
得到小程序状态
读取优惠券
错误
2608
交易已完成
选择
得到小程序状态
得到优惠券赎回状态
错误
2610
交易已中止
选择
得到小程序状态
错误
重置优惠券
中止交易
设置优惠券赎回状态
中止交易
重置优惠券

图 26

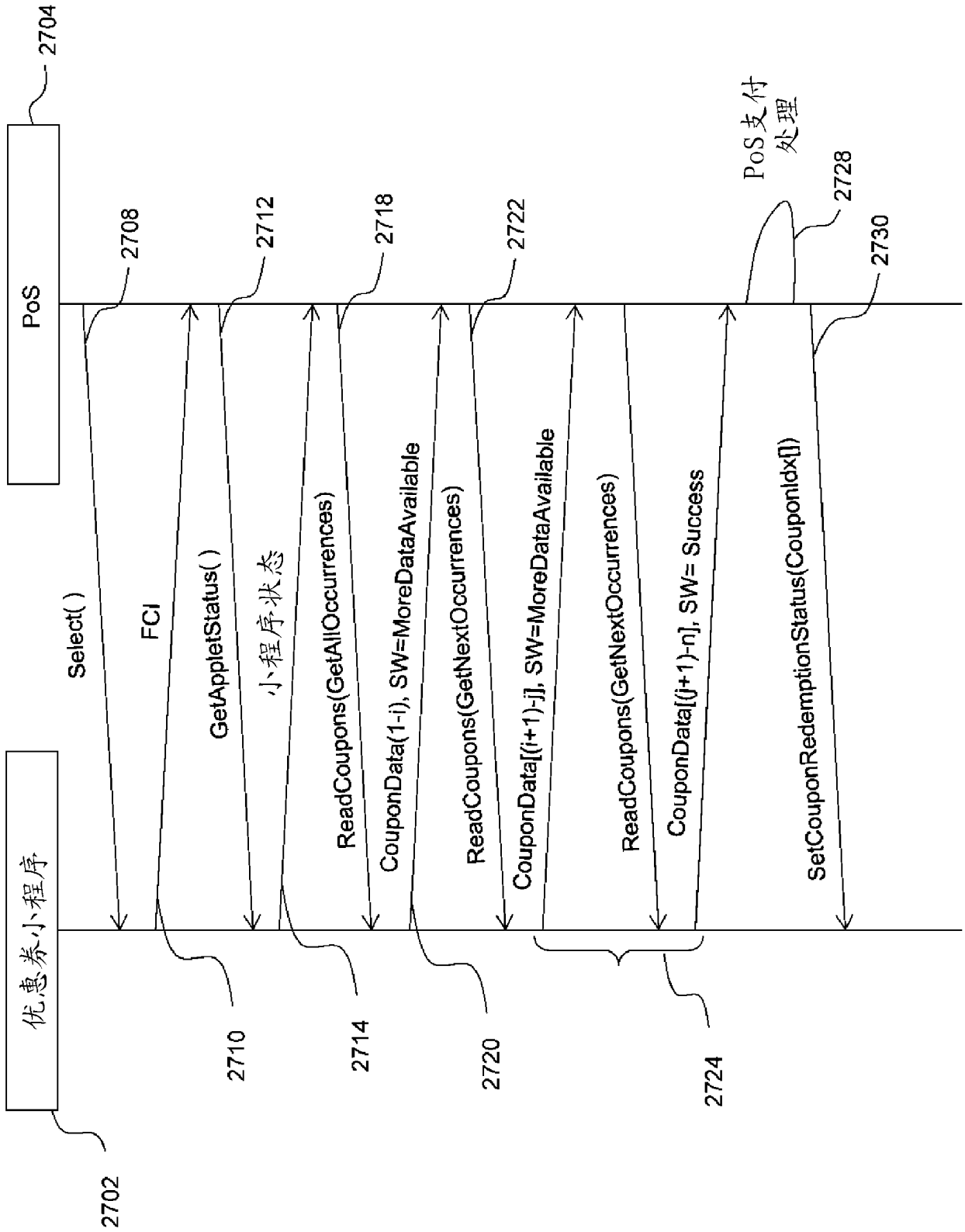

PoS
2704
优惠券小程序
2702
2708
2712
2718
2722
PoS支付处理
2728
2730
Select( )
FCI
GetAppletStatus( )
小程序状态
ReadCoupons(GetAllOccurrences)
CouponData(1-i), SW=MoreDataAvailable
ReadCoupons(GetNextOccurrences)
CouponData[(i+1)-j], SW=MoreDataAvailable
ReadCoupons(GetNextOccurrences)
CouponData[(j+1)-n], SW= Success
SetCouponRedemptionStatus(CouponIdx[])
2710
2714
2720
2724

图 27

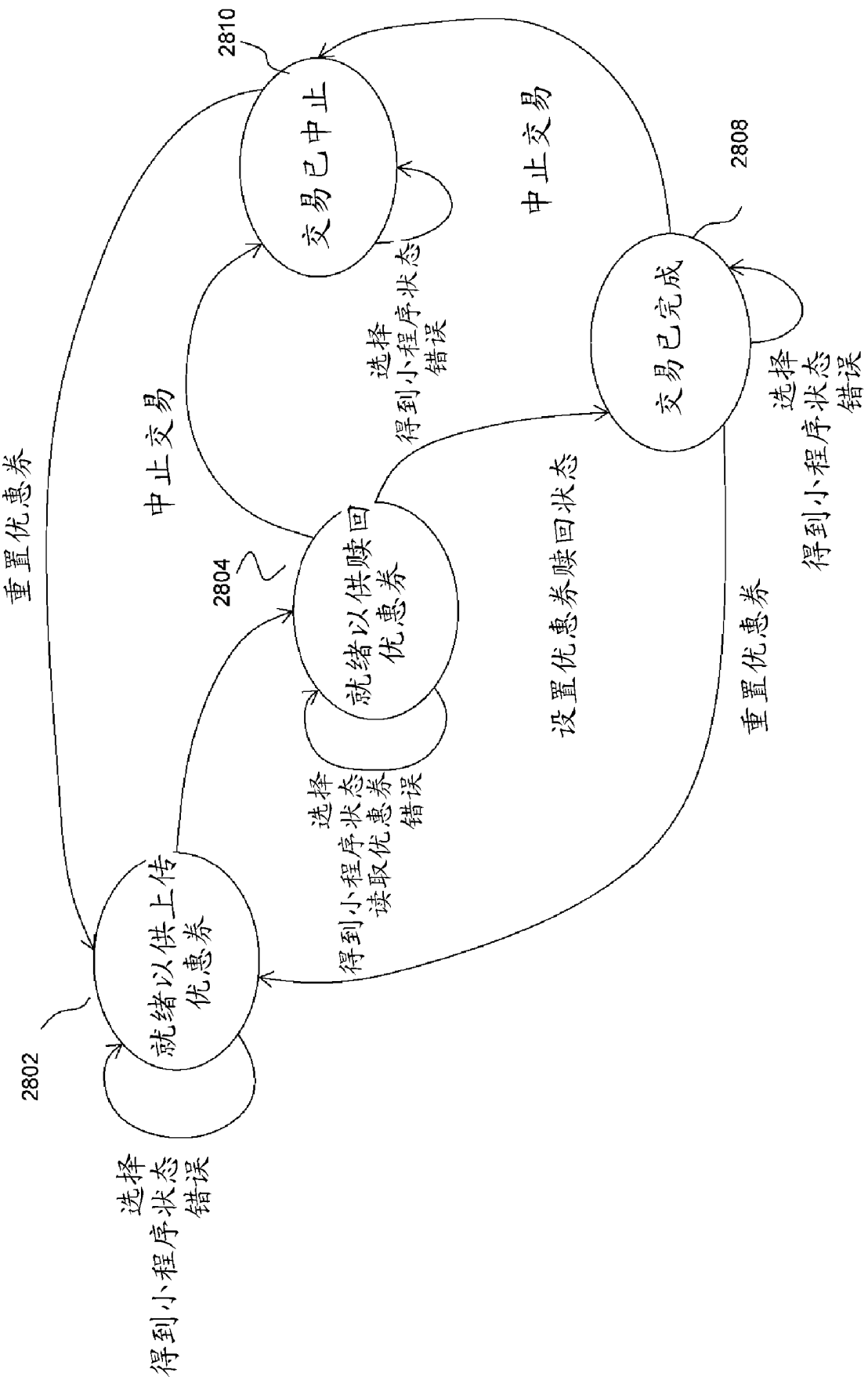

2802
就绪以供上传优惠券
选择得到小程序状态错误
重置优惠券
2804
就绪以供赎回优惠券
选择得到小程序状态读取优惠券错误
中止交易
2810
交易已中止
选择得到小程序状态错误
中止交易
设置优惠券赎回状态
2808
交易已完成
选择得到小程序状态错误
重置优惠券

图 28

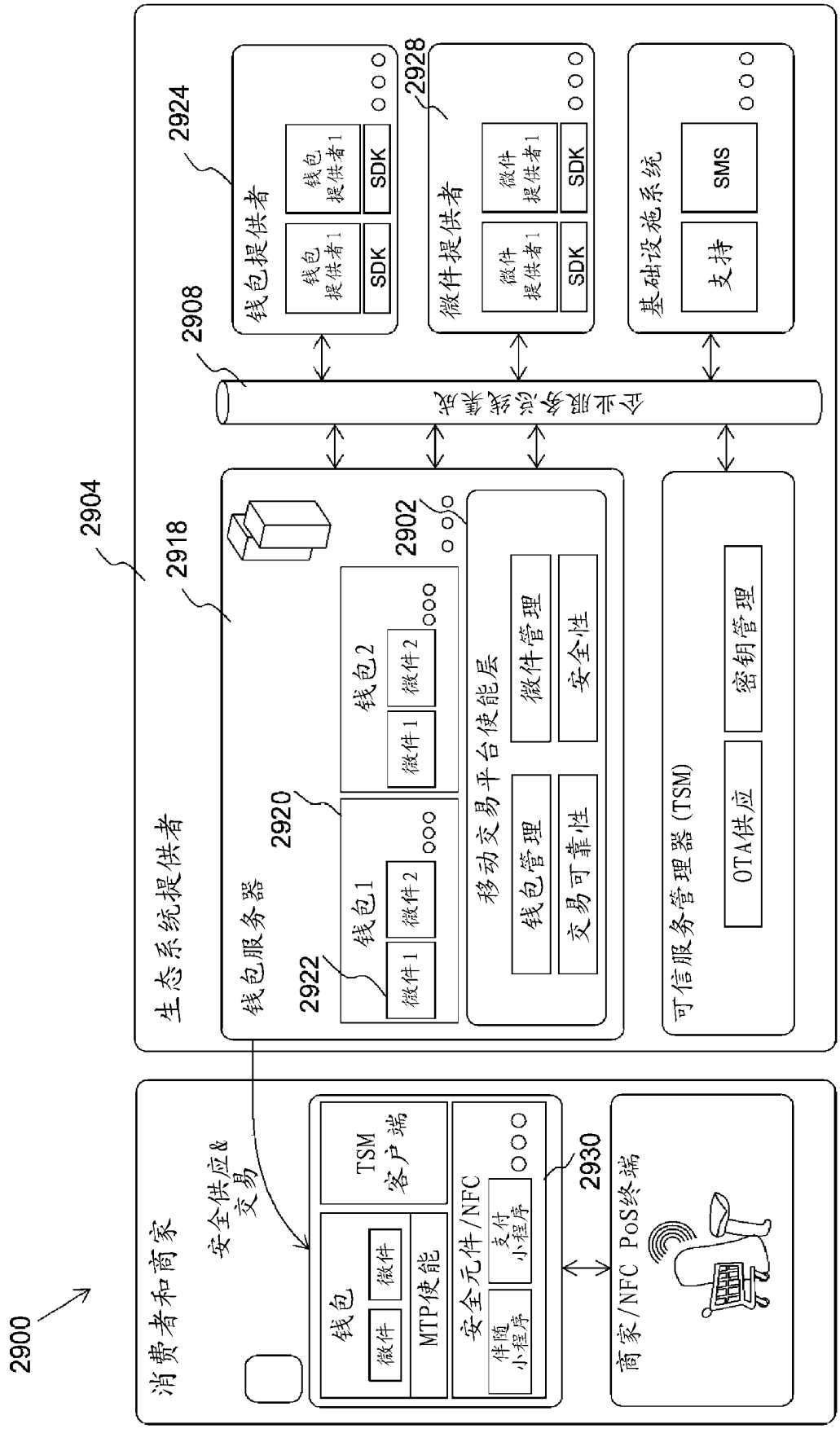

2900
消费者和商家
安全供应&
交易
钱包
微件
微件
MTP使能
TSM
客户端
安全元件/NFC
伴随
小程序
支付
小程序
2930
商家/NFC PoS终端
2904
生态系统提供者
2918
钱包服务器
2920
2922
钱包1
微件1
微件2
2902
钱包2
微件1
微件2
移动交易平台使能层
钱包管理
交易可靠性
微件管理
安全性
可信服务管理器(TSM)
OTA供应
密钥管理
2908
安全渠道平台
2924
钱包提供者
钱包
提供者1
SDK
钱包
提供者1
SDK
2928
微件提供者
微件
提供者1
SDK
微件
提供者1
SDK
基础设施系统
支持
SMS

图 29

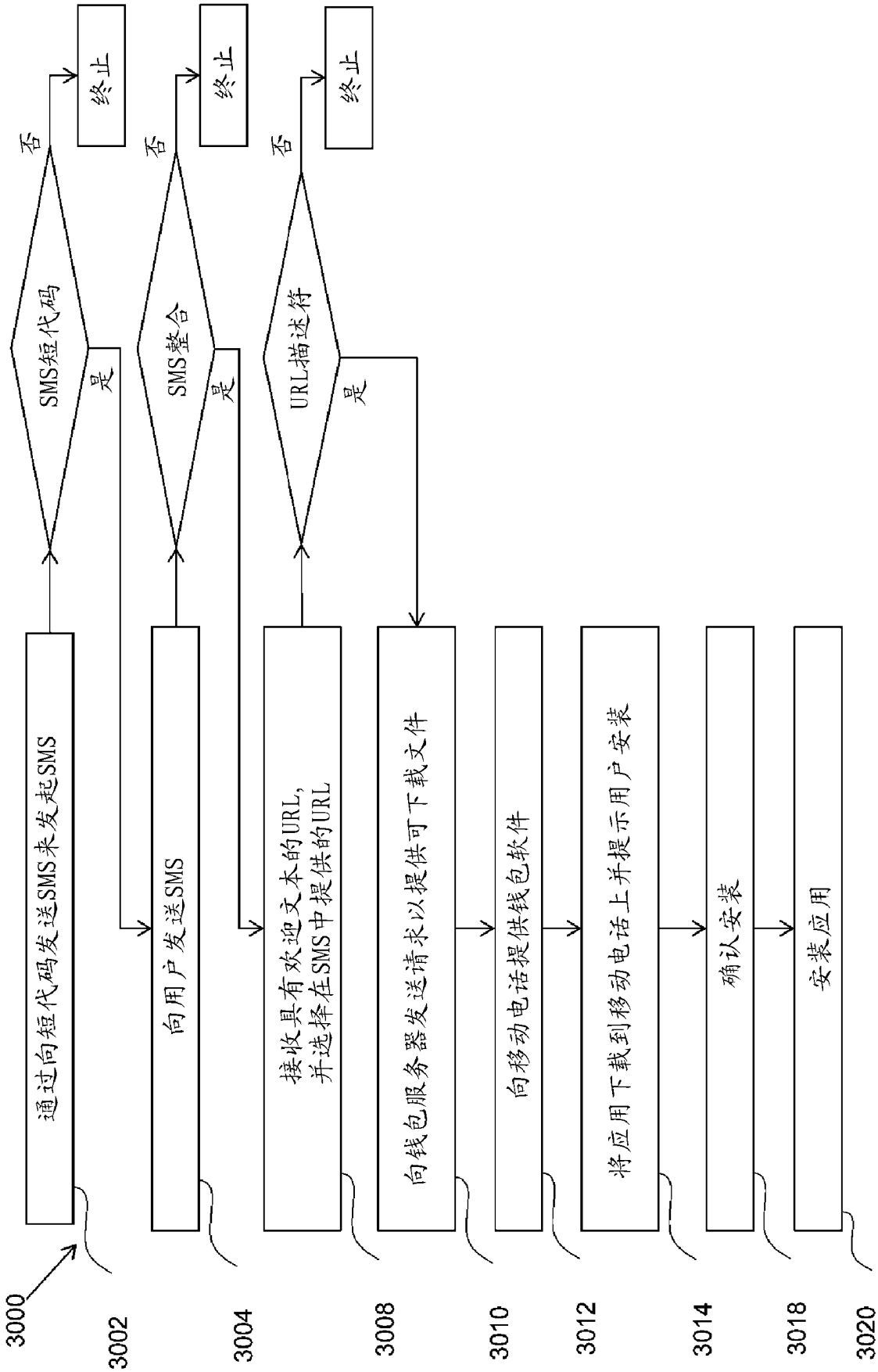

3000
通过向短代码发送SMS来发起SMS
3002
SMS短代码
否
终止
是
向用户发送SMS
3004
SMS整合
否
终止
是
接收具有欢迎文本的URL，
并选择在SMS中提供的URL
3008
URL描述符
否
终止
是
向钱包服务器发送请求以提供可下载文件
3010
向移动电话提供钱包软件
3012
将应用下载到移动电话上并提示用户安装
3014
确认安装
3018
安装应用
3020

图 30

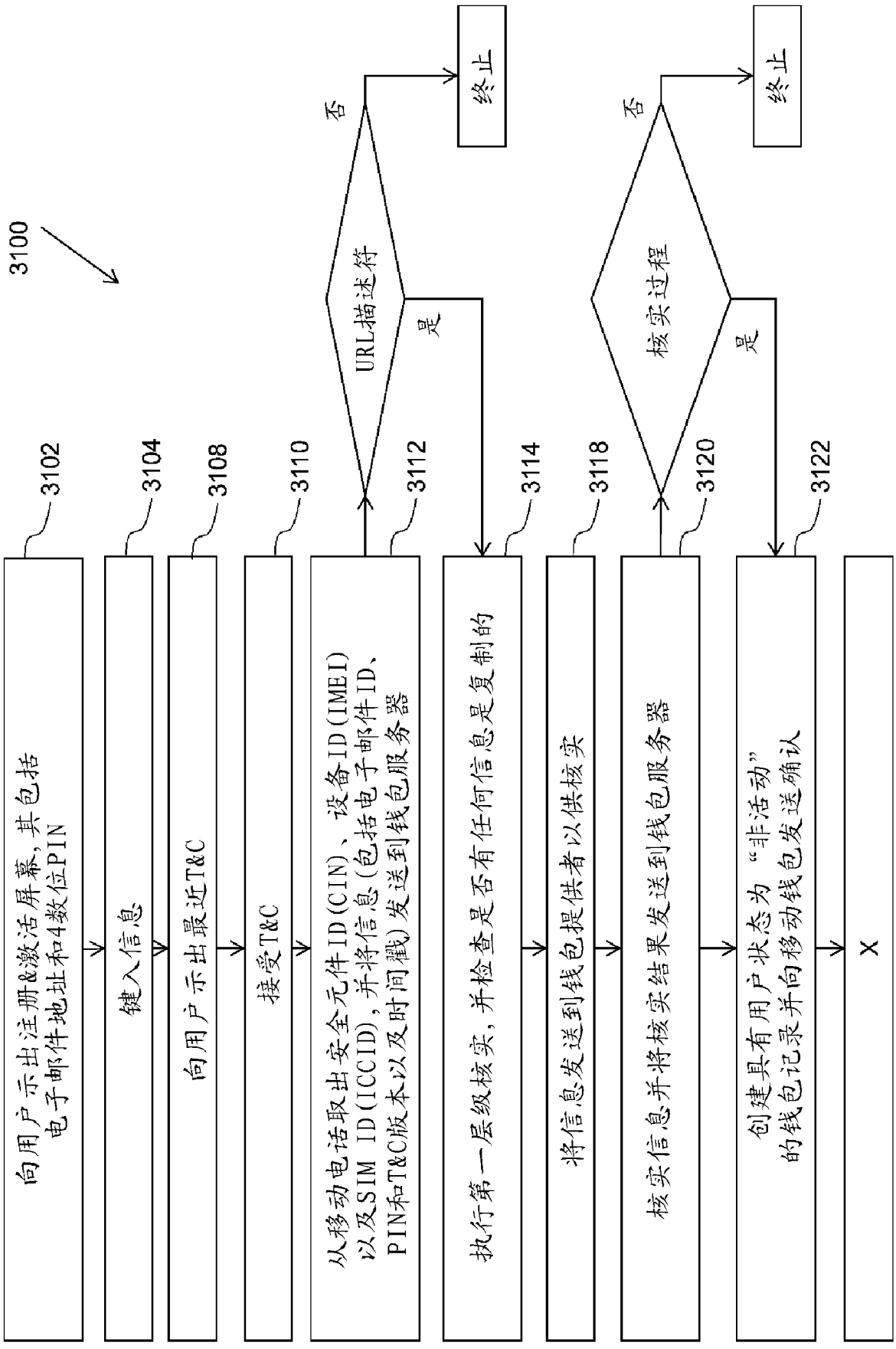

3100
向用户示出注册&激活屏幕，其包括
电子邮件地址和4数位PIN
3102
键入信息
3104
向用户示出最近T&C
3108
接受T&C
3110
从移动电话取出安全元件ID(CIN)、设备ID(IMEI)
以及SIM ID(ICCID)，并将信息(包括电子邮件ID、
PIN和T&C版本以及时间戳)发送到钱包服务器
3112
URL描述符
否
终止
是
执行第一层级核实，并检查是否有任何信息是复制的
3114
将信息发送到钱包提供者以供核实
3118
核实信息并将核实结果发送到钱包服务器
3120
核实过程
否
终止
是
创建具有用户状态为“非活动”
的钱包记录并向移动钱包发送确认
3122
X

图 31A

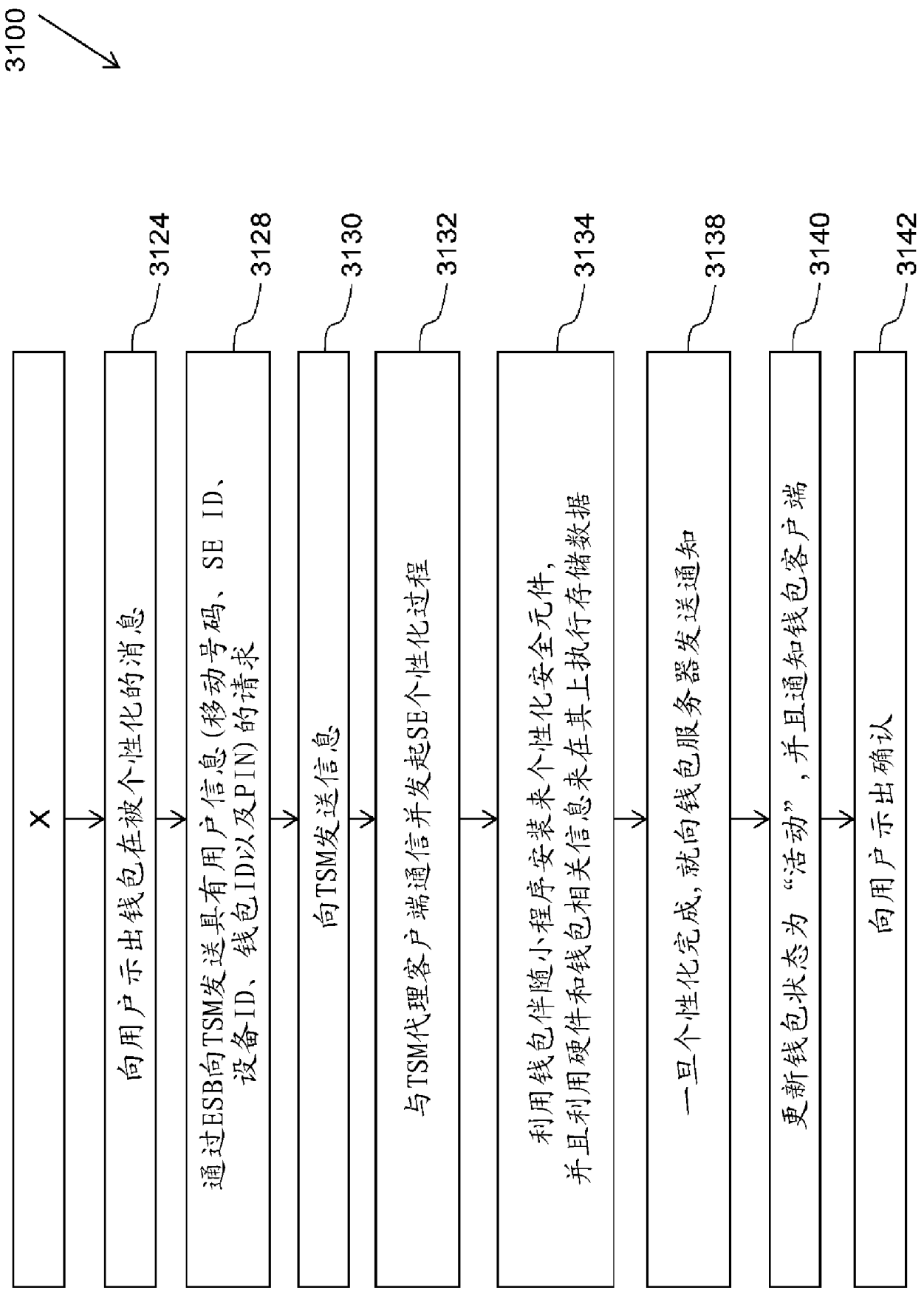

3100
X
向用户示出钱包在被个性化的消息
3124
通过ESB向TSM发送具有用户信息(移动号码、SE ID、设备ID、钱包ID以及PIN)的请求
3128
向TSM发送信息
3130
与TSM代理客户端通信并发起SE个性化过程
3132
利用钱包伴随小程序安装来个性化安全元件，并且利用硬件和钱包相关信息来在其上执行存储数据
3134
一旦个性化完成，就向钱包服务器发送通知
3138
更新钱包状态为“活动”，并且通知钱包客户端
3140
向用户示出确认
3142

图 31B

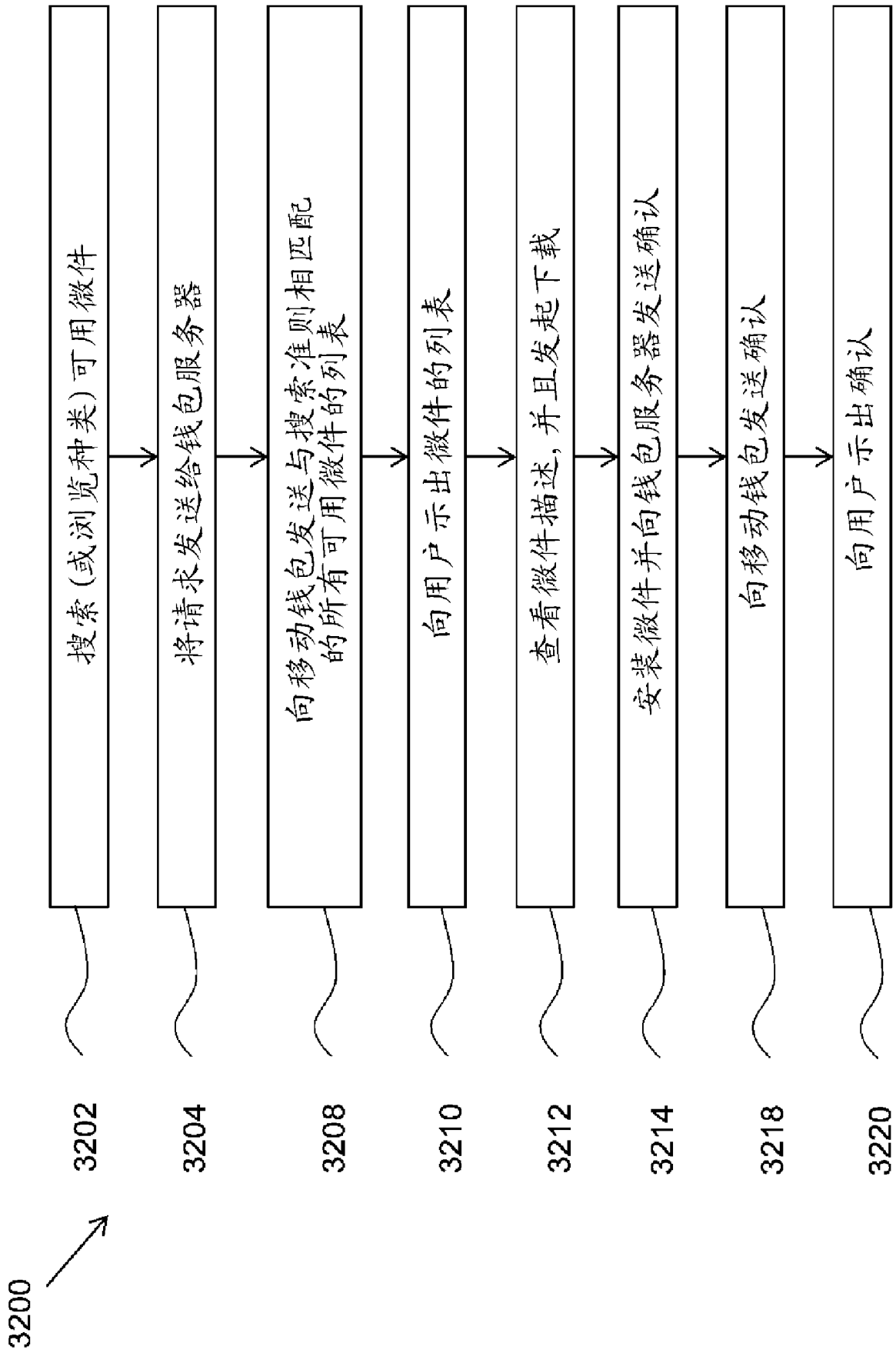

3200
搜索（或浏览种类）可用微件
3202
将请求发送给钱包服务器
3204
向移动钱包发送与搜索准则相匹配的所有可用微件的列表
3208
向用户示出微件的列表
3210
查看微件描述，并且发起下载
3212
安装微件并向钱包服务器发送确认
3214
向移动钱包发送确认
3218
向用户示出确认
3220

图 32

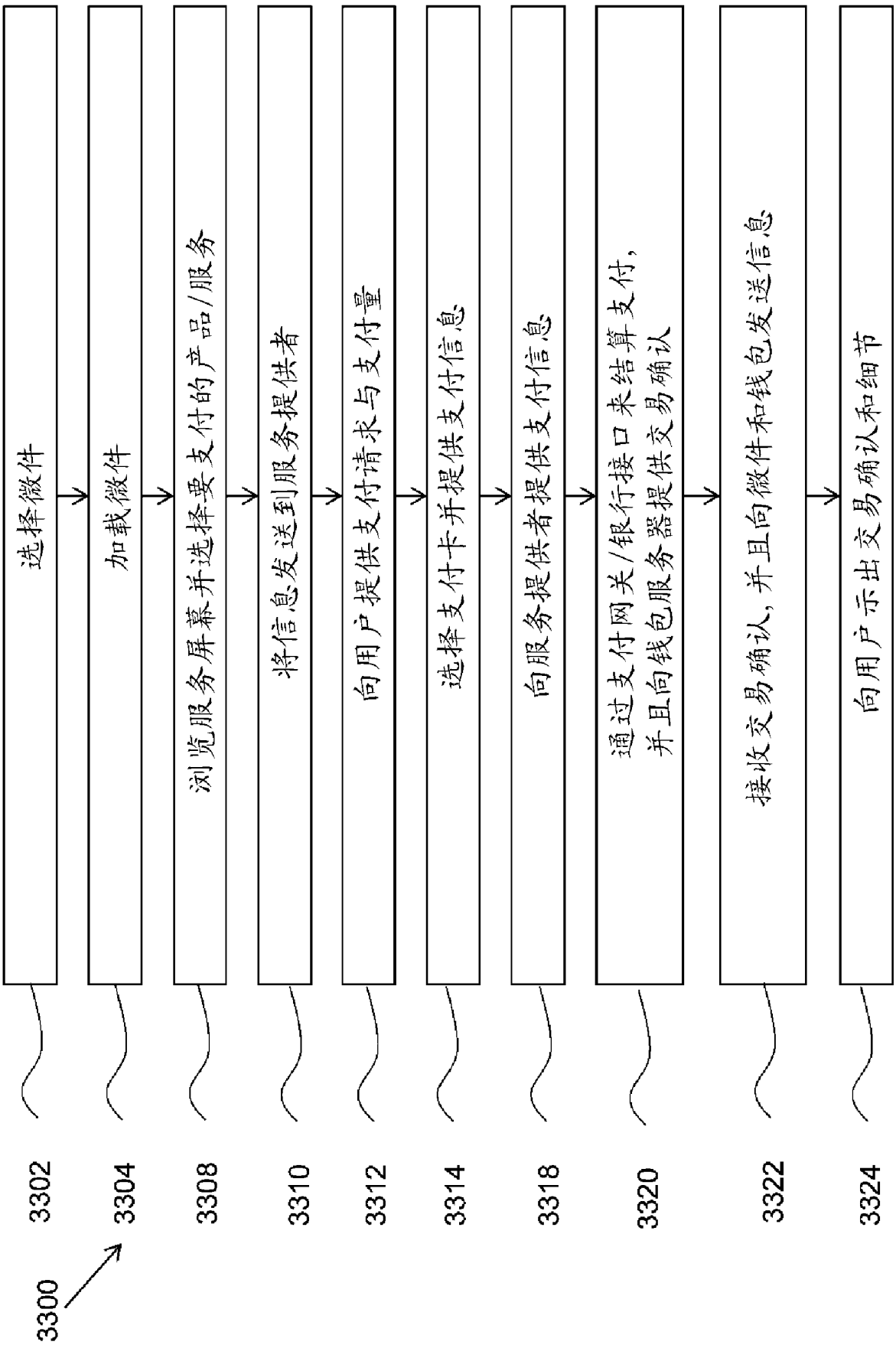

选择微件
加载微件
浏览服务屏幕并选择要支付的产品/服务
将信息发送到服务提供者
向用户提供支付请求与支付量
选择支付卡并提供支付信息
向服务提供者提供支付信息
通过支付网关/银行接口来结算支付，
并且向钱包服务器提供交易确认
接收交易确认，并且向微件和钱包发送信息
向用户示出交易确认和细节
3300
3302
3304
3308
3310
3312
3314
3318
3320
3322
3324

图 33

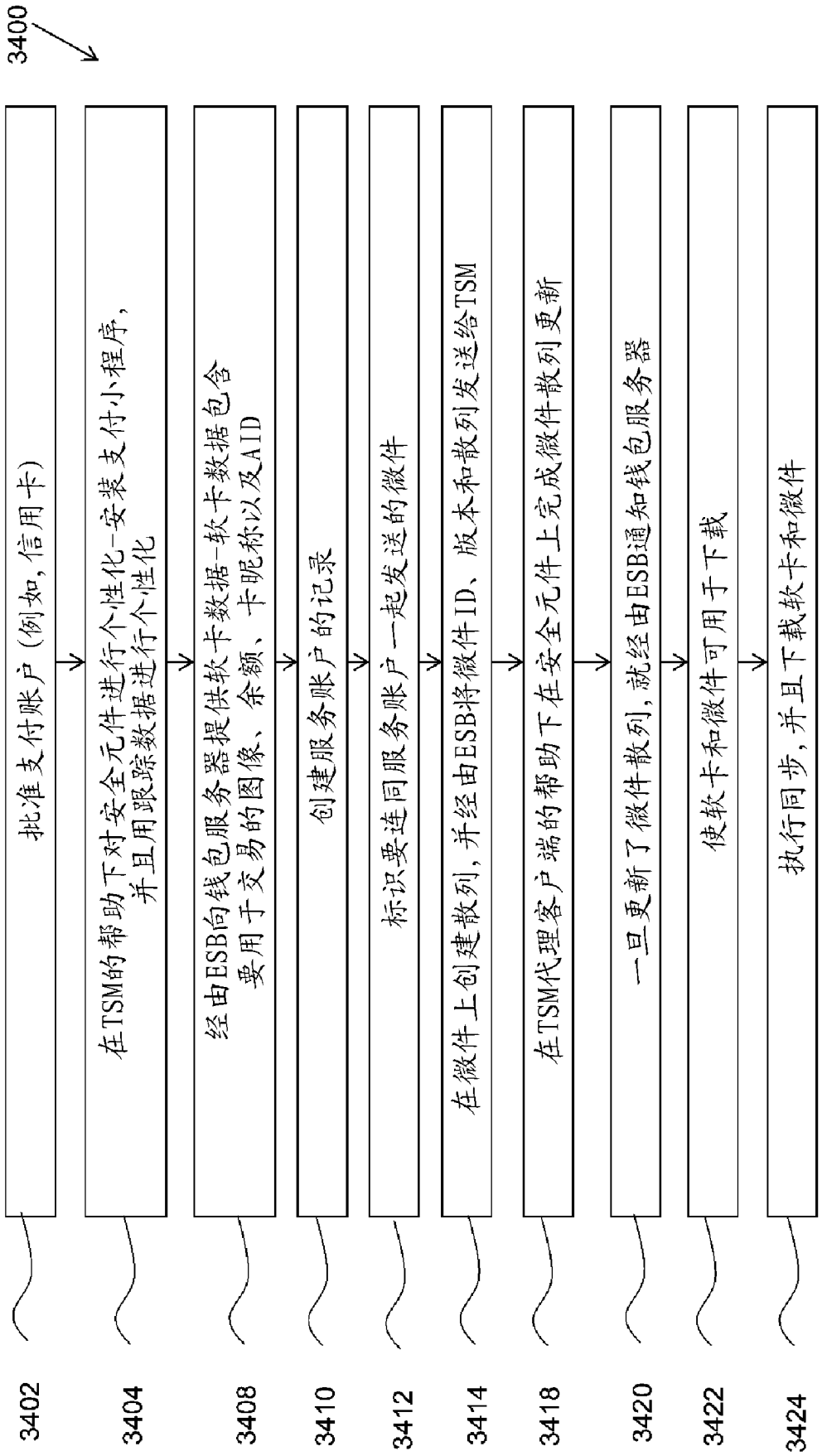

3400
批准支付账户（例如，信用卡）
在TSM的帮助下对安全元件进行个性化-安装支付小程序，并且用跟踪数据进行个性化
经由ESB向钱包服务器提供软卡数据-软卡数据包含要用于交易的图像、余额、卡昵称以及AID
创建服务账户的记录
标识要连同服务账户一起发送的微件
在微件上创建散列，并经由ESB将微件ID、版本和散列发送给TSM
在TSM代理客户端的帮助下在安全元件上完成微件散列更新
一旦更新了微件散列，就经由ESB通知钱包服务器
使软卡和微件可用于下载
执行同步，并且下载软卡和微件
3402
3404
3408
3410
3412
3414
3418
3420
3422
3424

图 34


3500
发起挂起钱包活动
更新钱包状态为“TERMINATION_PENDING”（终止_待决）
向ESB发送请求
从钱包服务器取出服务账户列表
向发行者发送请求以终止服务账户
在后端终止账户，并且还在TSM的帮助下从SE移除支付小程序
向TSM发送请求以移除钱包伴随小程序
向TSM代理客户端发送消息
TSM代理客户端从SE移除钱包伴随小程序
一旦完成更新，TSM就向ESB发送通知
钱包服务器接收更新，并且将钱包状态改变为“已终止”
3502
3504
3508
3510
3512
3514
3518
3520
3522
3524
3528

图 35

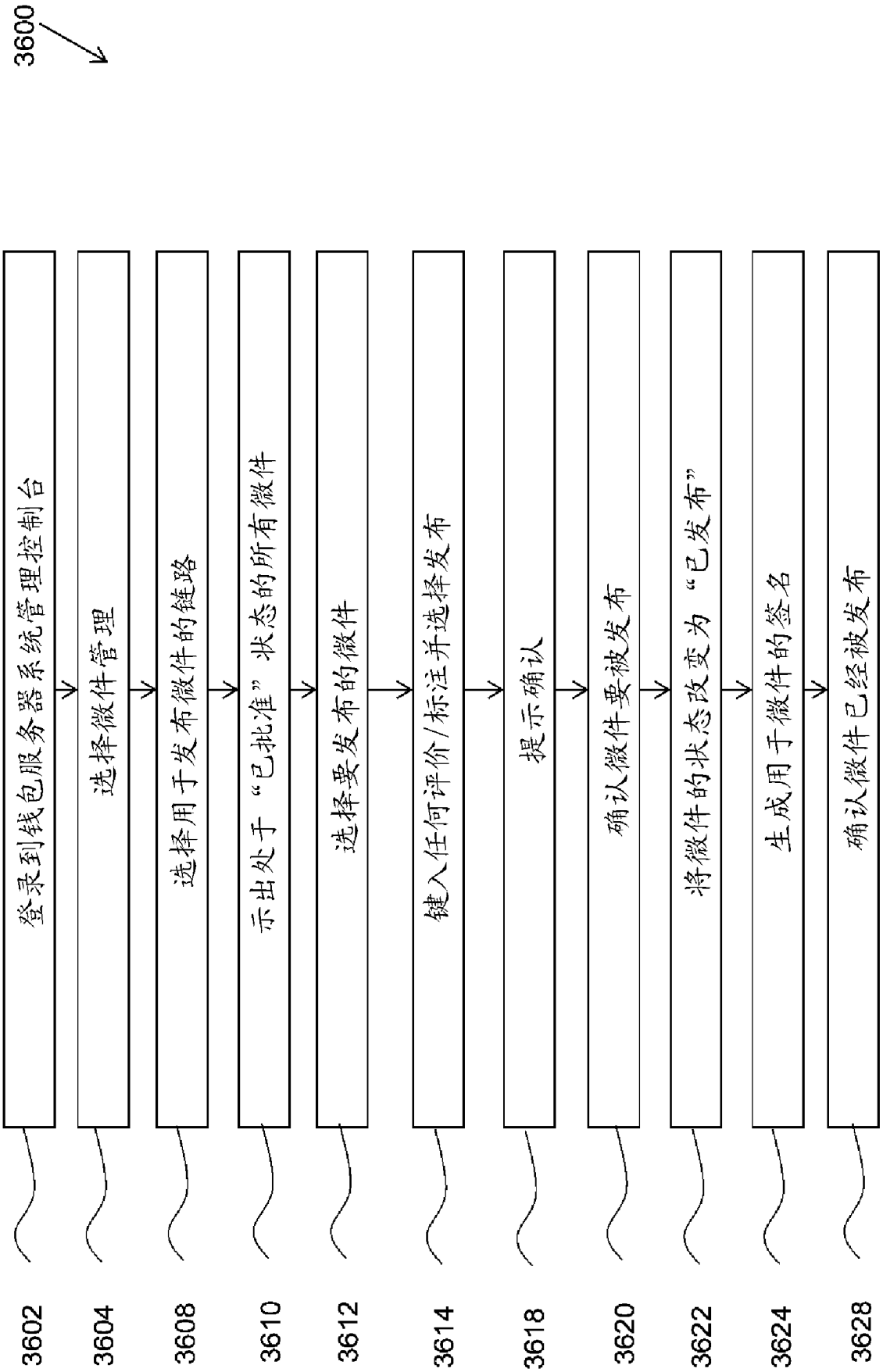

3600
登录到钱包服务器系统管理控制台
选择微件管理
选择用于发布微件的链路
示出处于“已批准”状态的所有微件
选择要发布的微件
键入任何评价/标注并选择发布
提示确认
确认微件要被发布
将微件的状态改变为“已发布”
生成用于微件的签名
确认微件已经被发布
3602
3604
3608
3610
3612
3614
3618
3620
3622
3624
3628

图 36

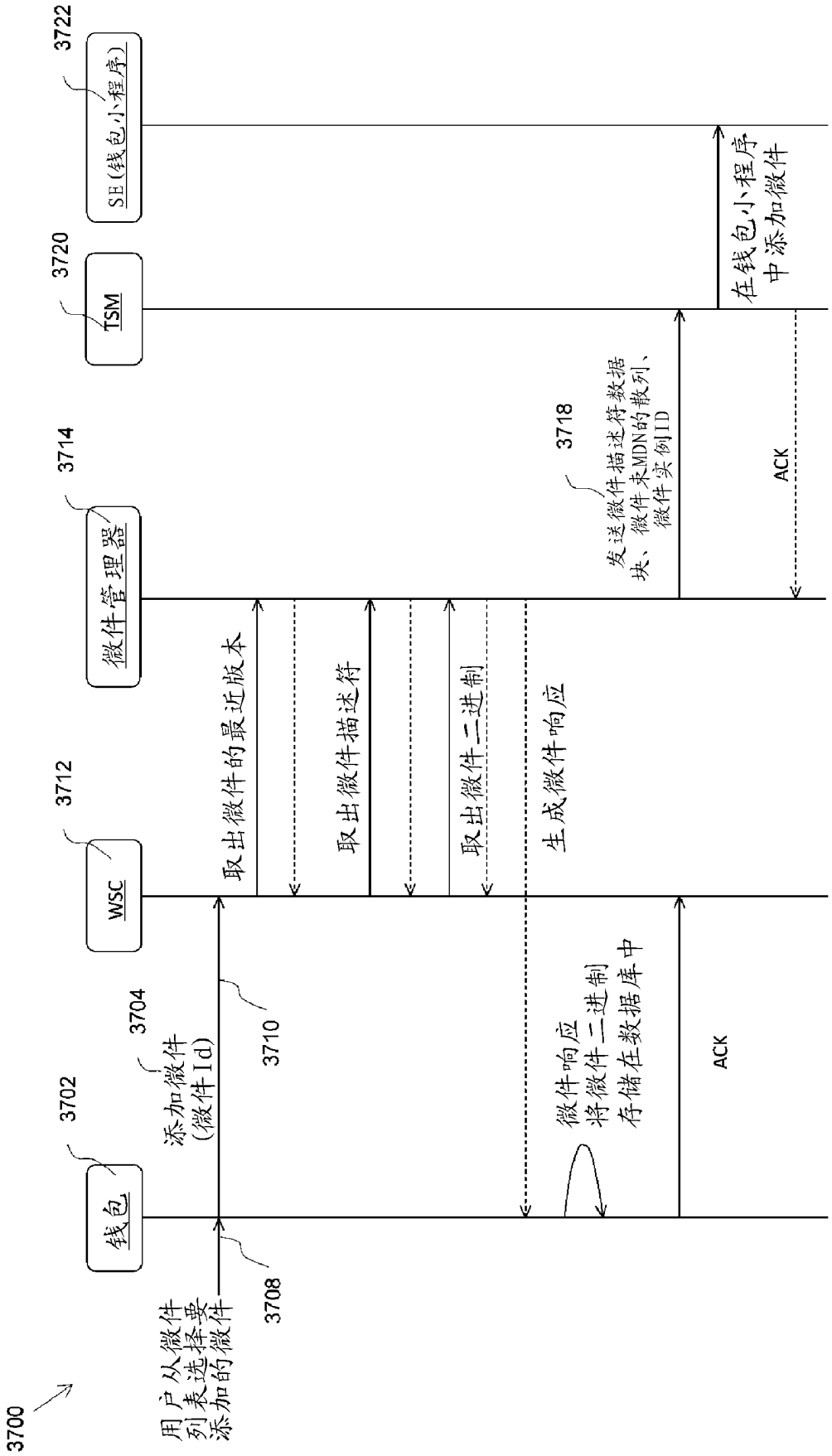

3700
用户从微件列表选择要添加的微件
3708
钱包
3702
添加微件(微件Id)
3704
3710
WSC
3712
取出微件的最近版本
取出微件描述符
取出微件二进制
生成微件响应
微件管理器
3714
发送微件描述符数据块、微件来MDN的散列、微件实例IID
3718
TSM
3720
SE(钱包小程序)
3722
在钱包小程序中添加微件
ACK
微件响应
将微件二进制存储在数据库中
ACK

图 37

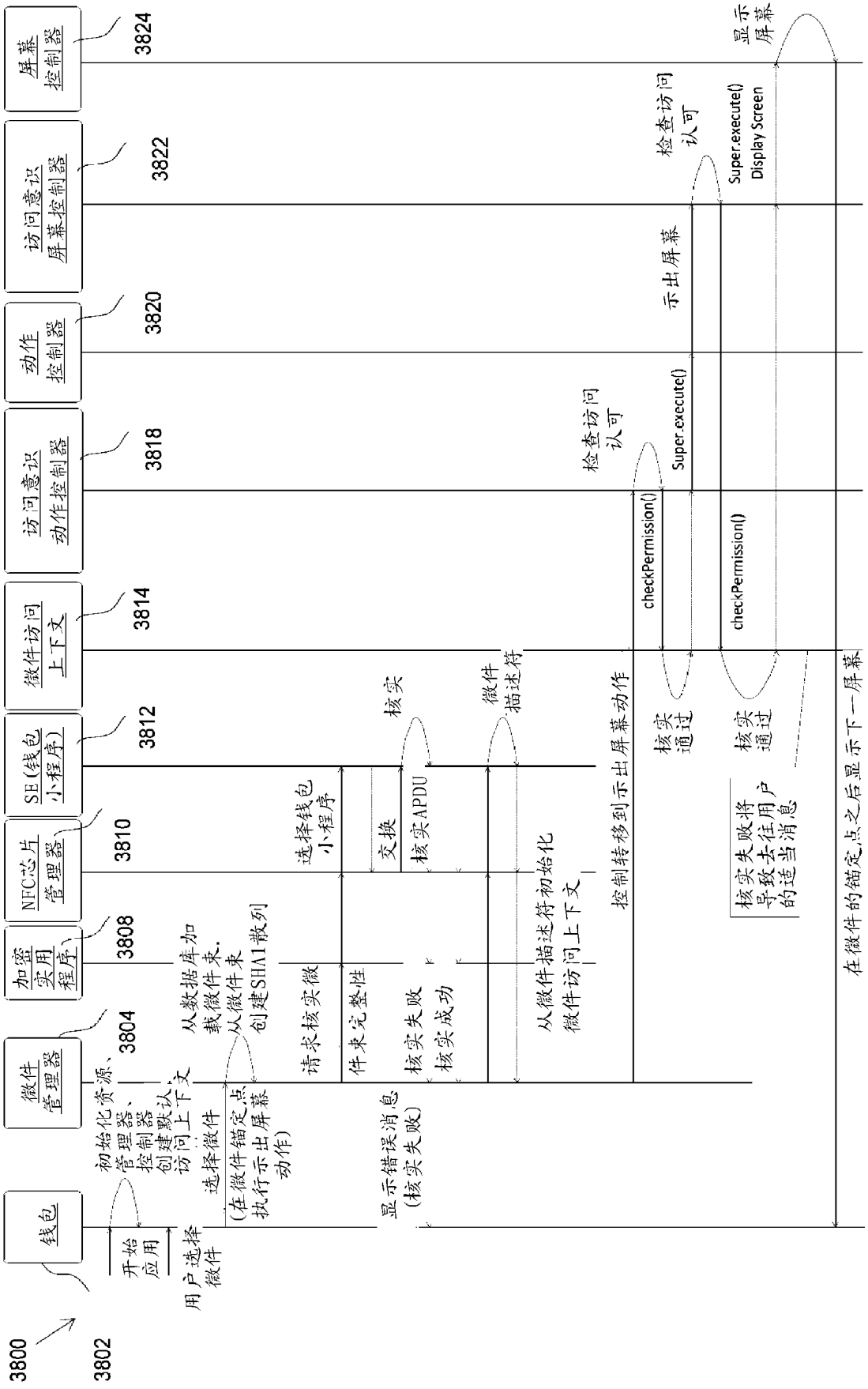

3800
3802
钱包
微件管理器
3804
加密实用程序
3808
NFC芯片管理器
3810
SE(钱包小程序)
3812
微件访问上下文
3814
访问意识动作控制器
3818
动作控制器
3820
访问意识屏幕控制器
3822
屏幕控制器
3824
开始应用
用户选择微件
初始化资源、管理器、控制器创建默认访问上下文
选择微件(在微件锚定点执行示出屏幕动作)
从数据库加载微件束.
从微件束创建SHA1散列
请求核实微件束完整性
选择钱包小程序
交换
核实APDU
核实失败
核实成功
微件描述符
从微件描述符初始化微件访问上下文
控制转移到示出屏幕动作
显示错误消息(核实失败)
检查访问认可
checkPermission()
Super.execute()
示出屏幕
核实通过
检查访问认可
checkPermission()
Super.execute()
Display Screen
显示屏幕
核实通过
核实失败将导致去往用户的适当消息
在微件的锚定点之后显示下一屏幕

图 38

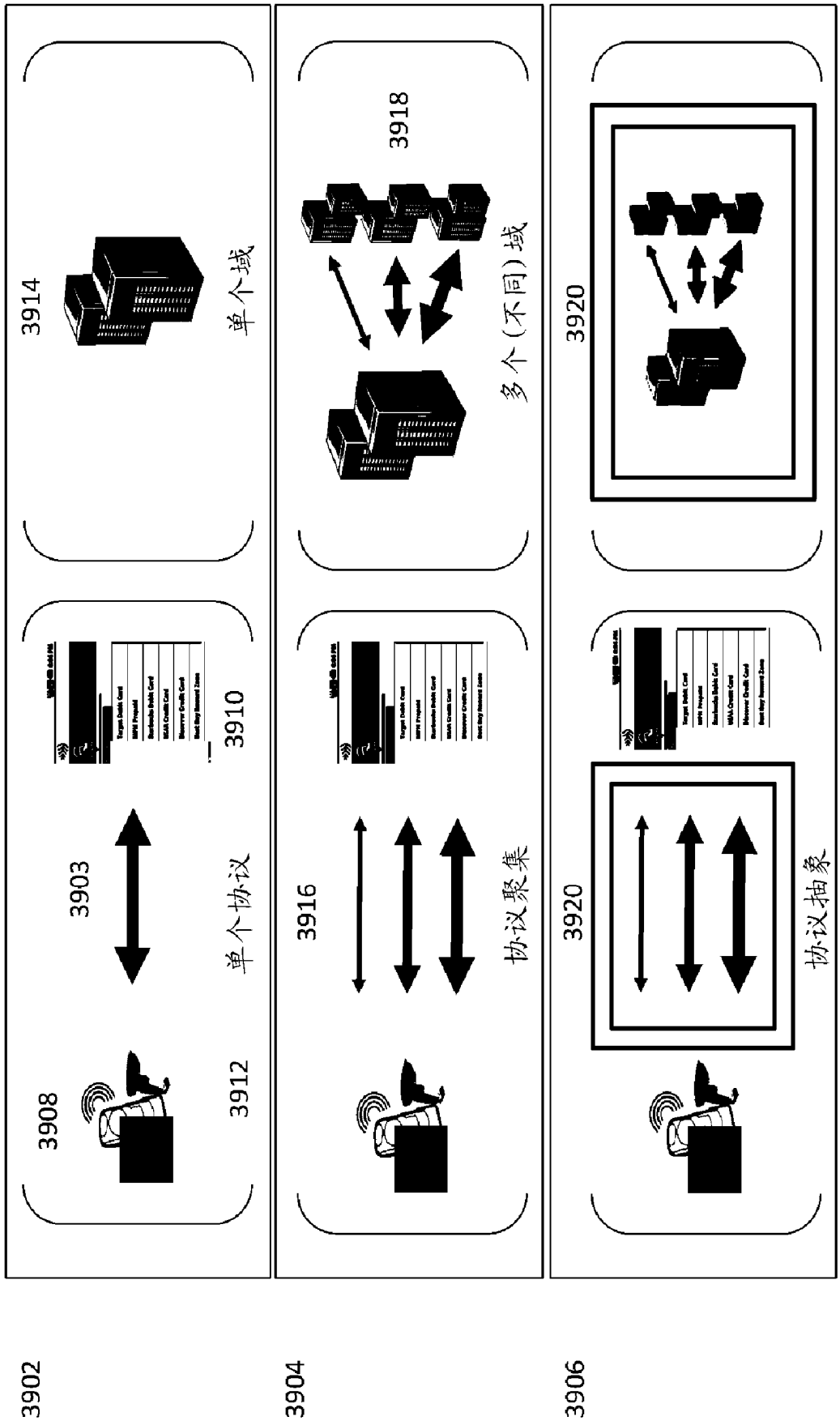

3902
3904
3906
3908
3912
3903
单个协议
3910
3914
单个域
3916
协议聚集
3918
多个(不同)域
3920
协议抽象

图 39

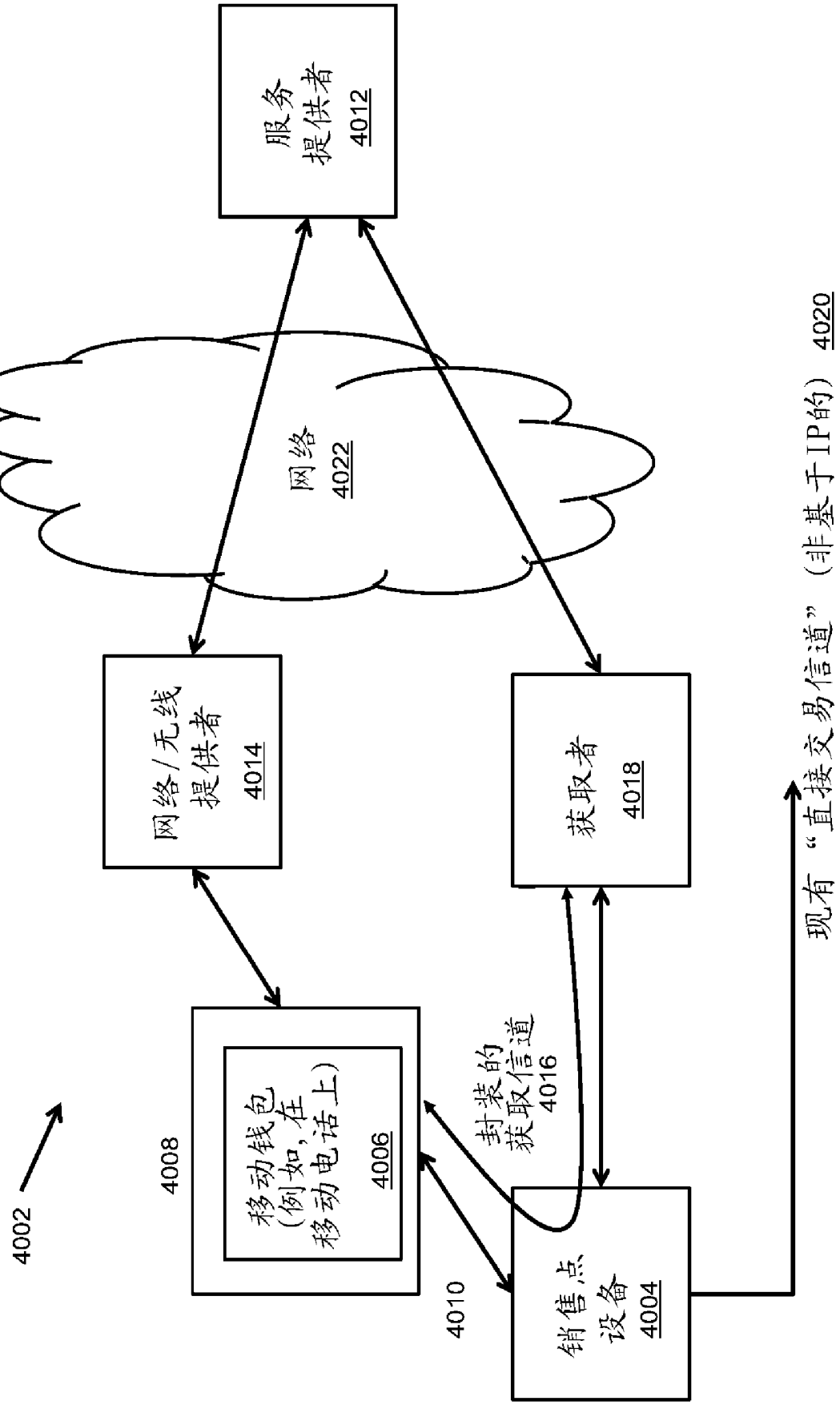

服务提供者 4012
网络 4022
网络/无线提供者 4014
获取者 4018
现有“直接交易信道”（非基于IP的） 4020
4002
4008
移动钱包（例如，在移动电话上） 4006
封装的获取信道 4016
4010
销售点设备 4004

图 40

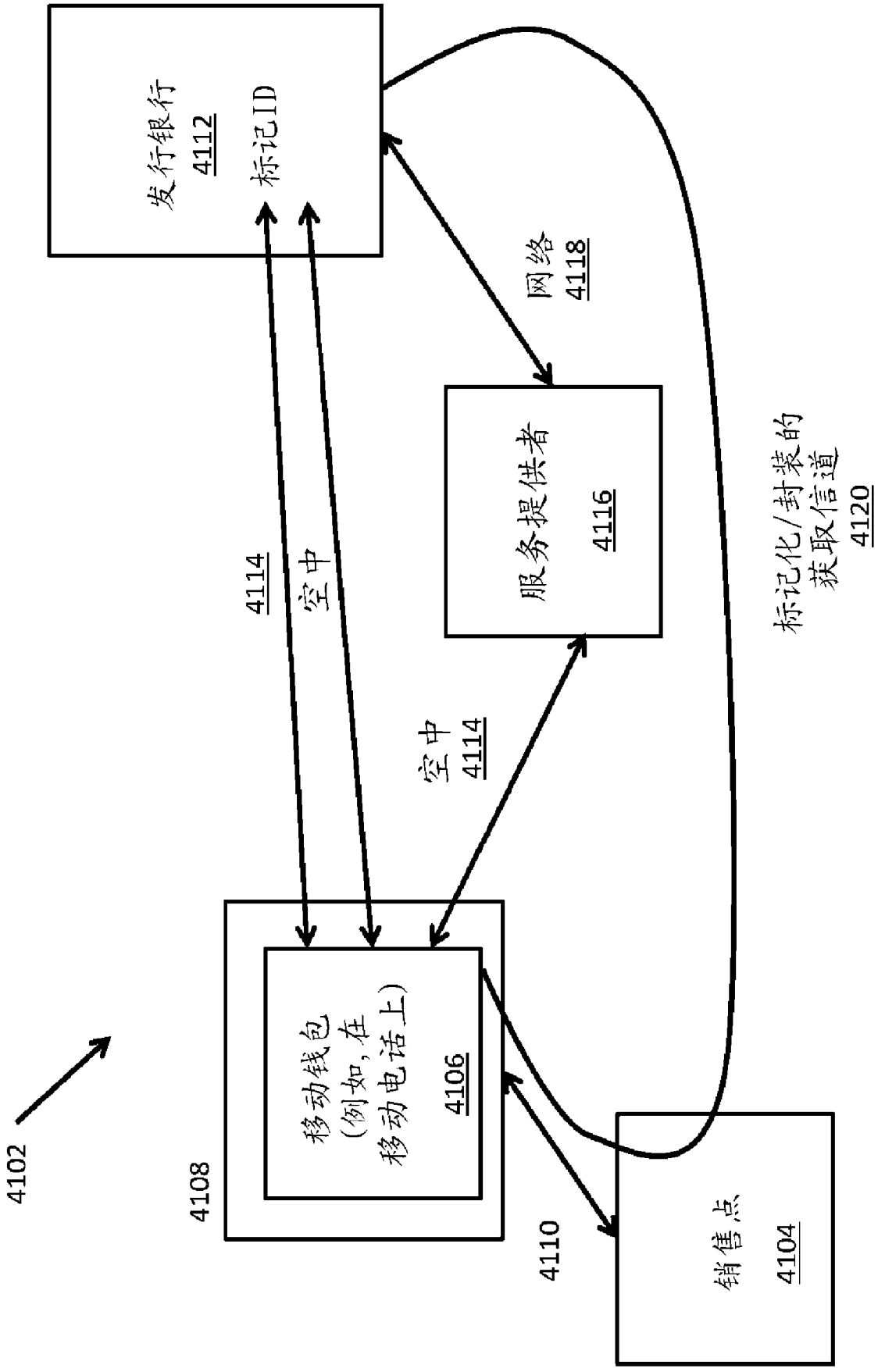

发行银行
4112
标记ID
网络
4118
服务提供者
4116
4114
空中
空中
4114
标记化/封装的
获取信道
4120
4102
4108
移动钱包
(例如,在
移动电话上)
4106
4110
销售点
4104

图 41

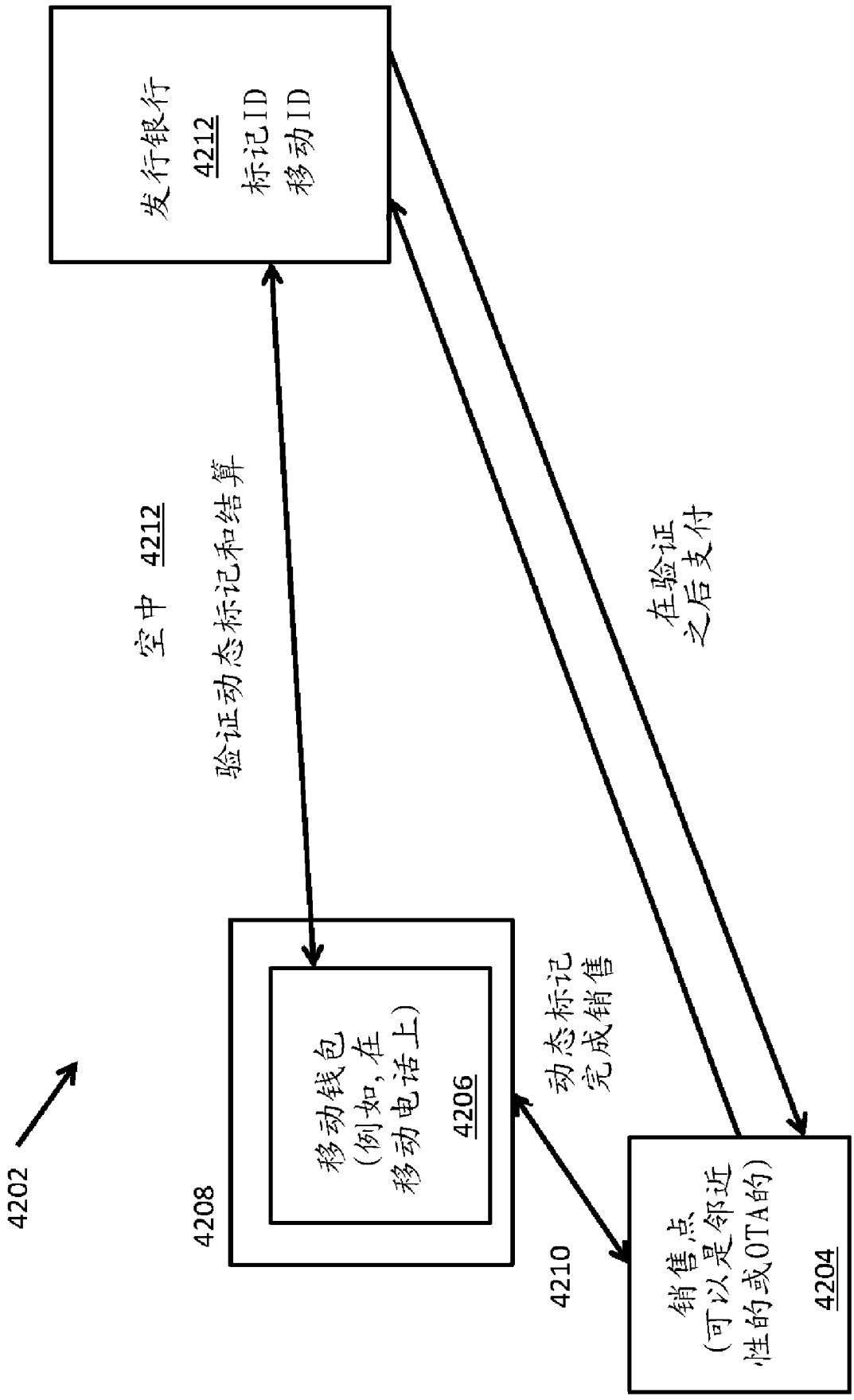

发行银行
4212
标记ID
移动ID
空中 4212
验证动态标记和结算
在验证
之后支付
4202
4208
移动钱包
(例如,在
移动电话上)
4206
动态标记
完成销售
4210
销售点
(可以是邻近
性的或OTA的)
4204

图 42

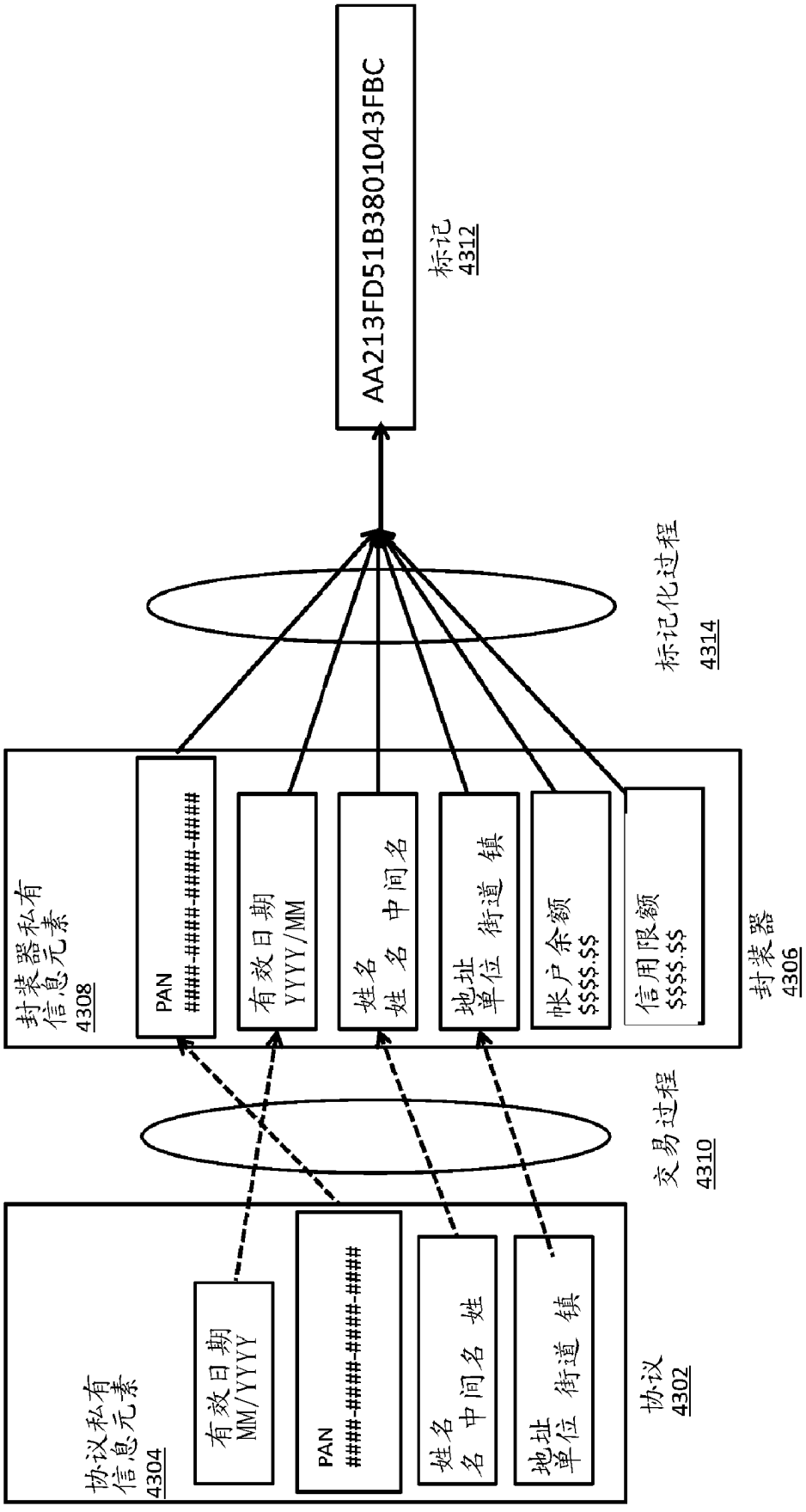

AA213FD51B3801043FBC
标记
4312
标记化过程
4314
封装器私有
信息元素
4308
PAN
#####-#####-#####-####
有效日期
YYYY/MM
姓名
姓 名 中间名
地址
单位 街道 镇
帐户余额
$$$$.$$
信用限额
$$$$.$$
封装器
4306
交易过程
4310
协议私有
信息元素
4304
有效日期
MM/YYYY
PAN
#####-#####-#####-####
姓名
名 中间名 姓
地址
单位 街道 镇
协议
4302

图 43

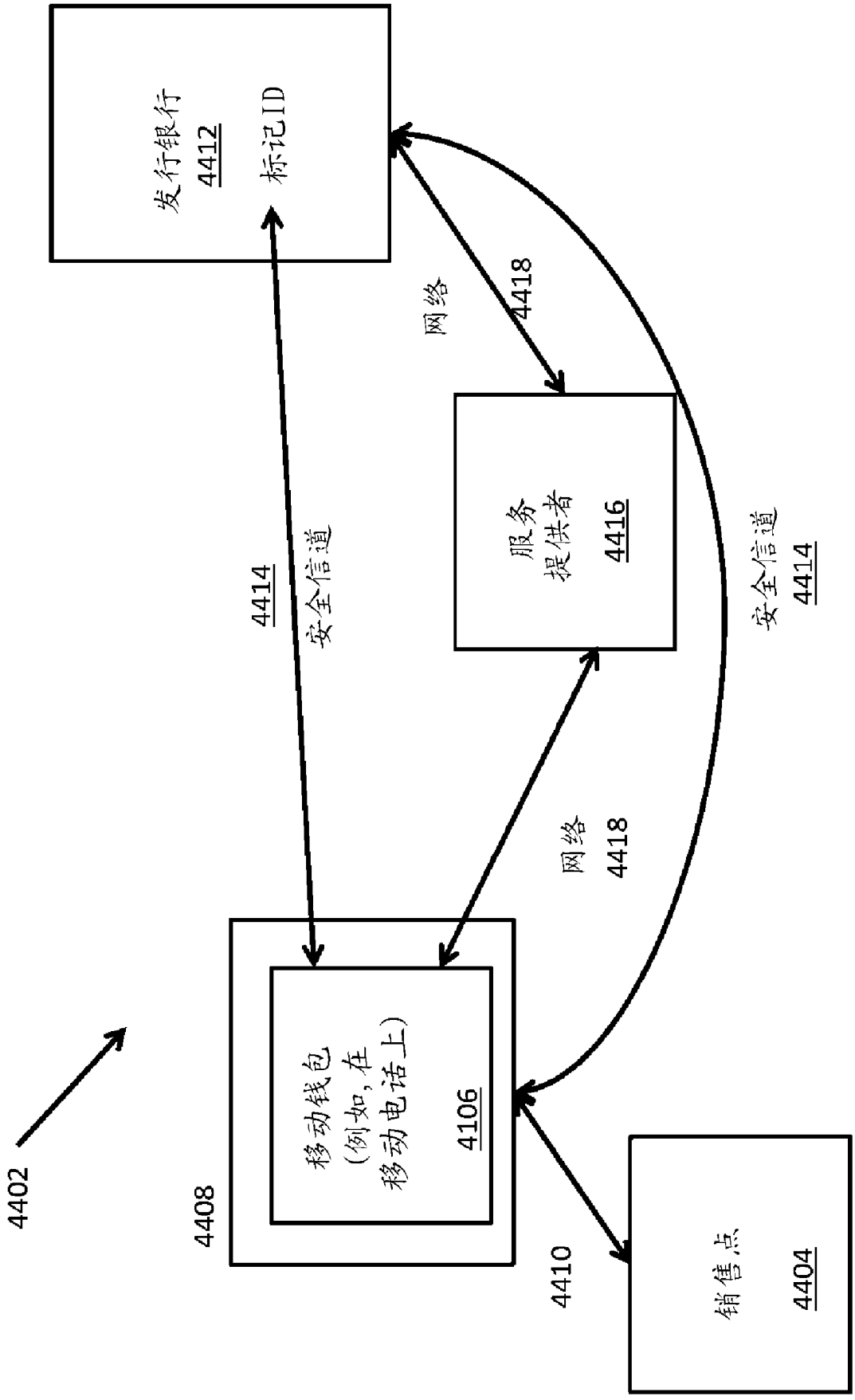

发行银行
4412
标记ID
网络
4418
4414
安全信道
服务
提供者
4416
安全信道
4414
网络
4418
移动钱包
(例如,在
移动电话上)
4106
4408
4402
4410
销售点
4404

图 44

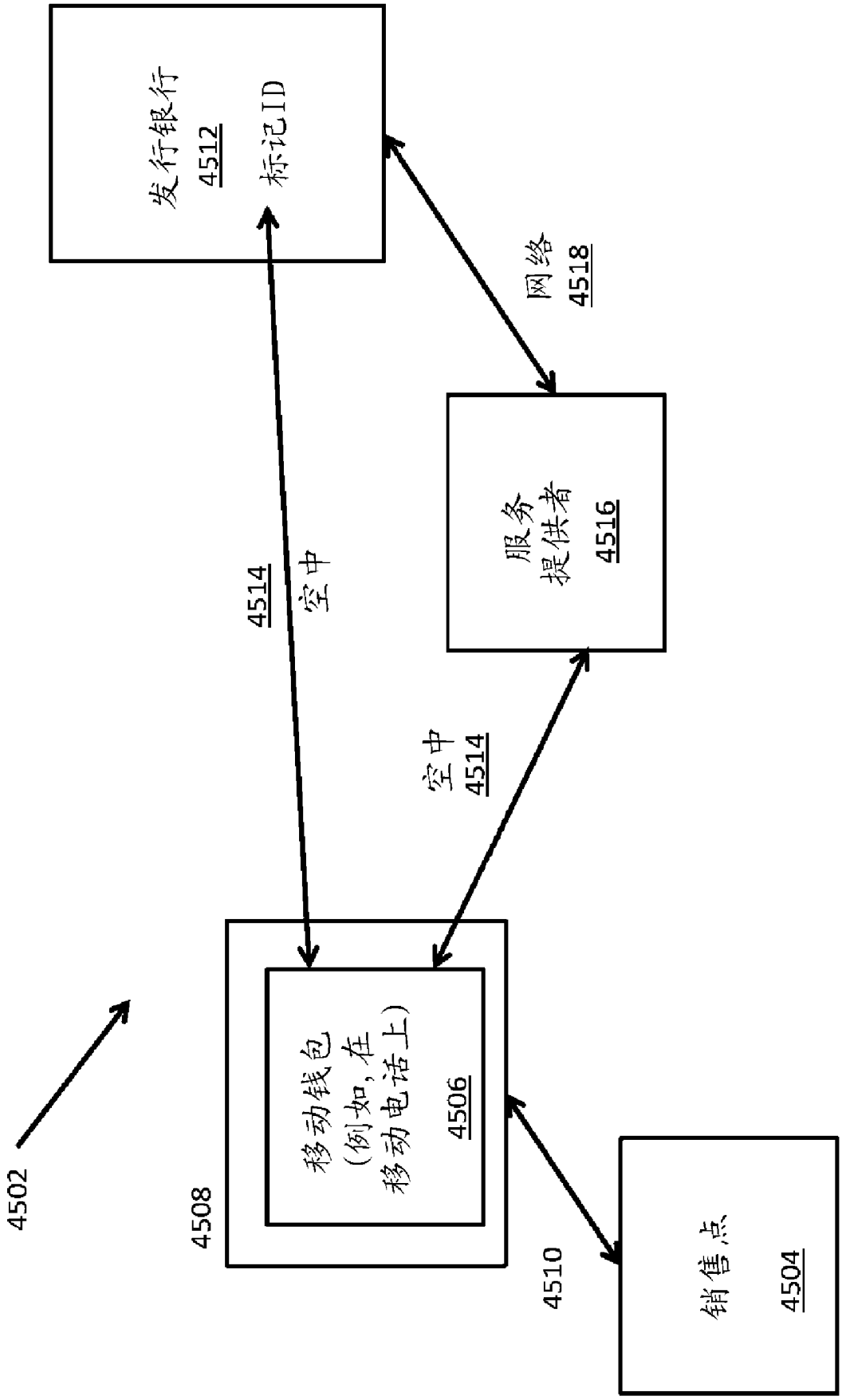

发行银行
4512
标记ID
网络
4518
4514
空中
服务
提供者
4516
空中
4514
移动钱包
(例如,在
移动电话上)
4506
4502
4508
4510
销售点
4504

图 45

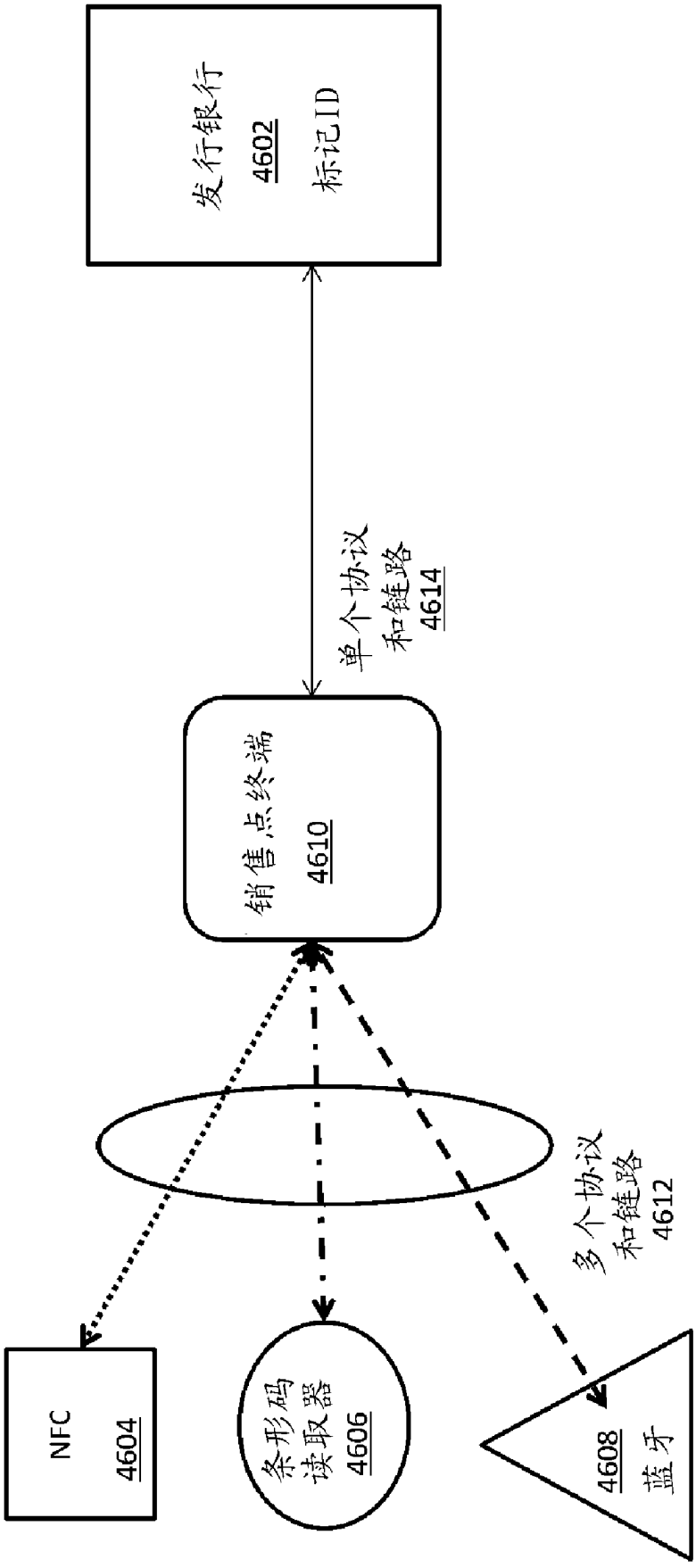

发行银行
4602
标记ID
单个协议
和链路
4614
销售点终端
4610
多个协议
和链路
4612
NFC
4604
条形码
读取器
4606
4608
蓝牙

图 46

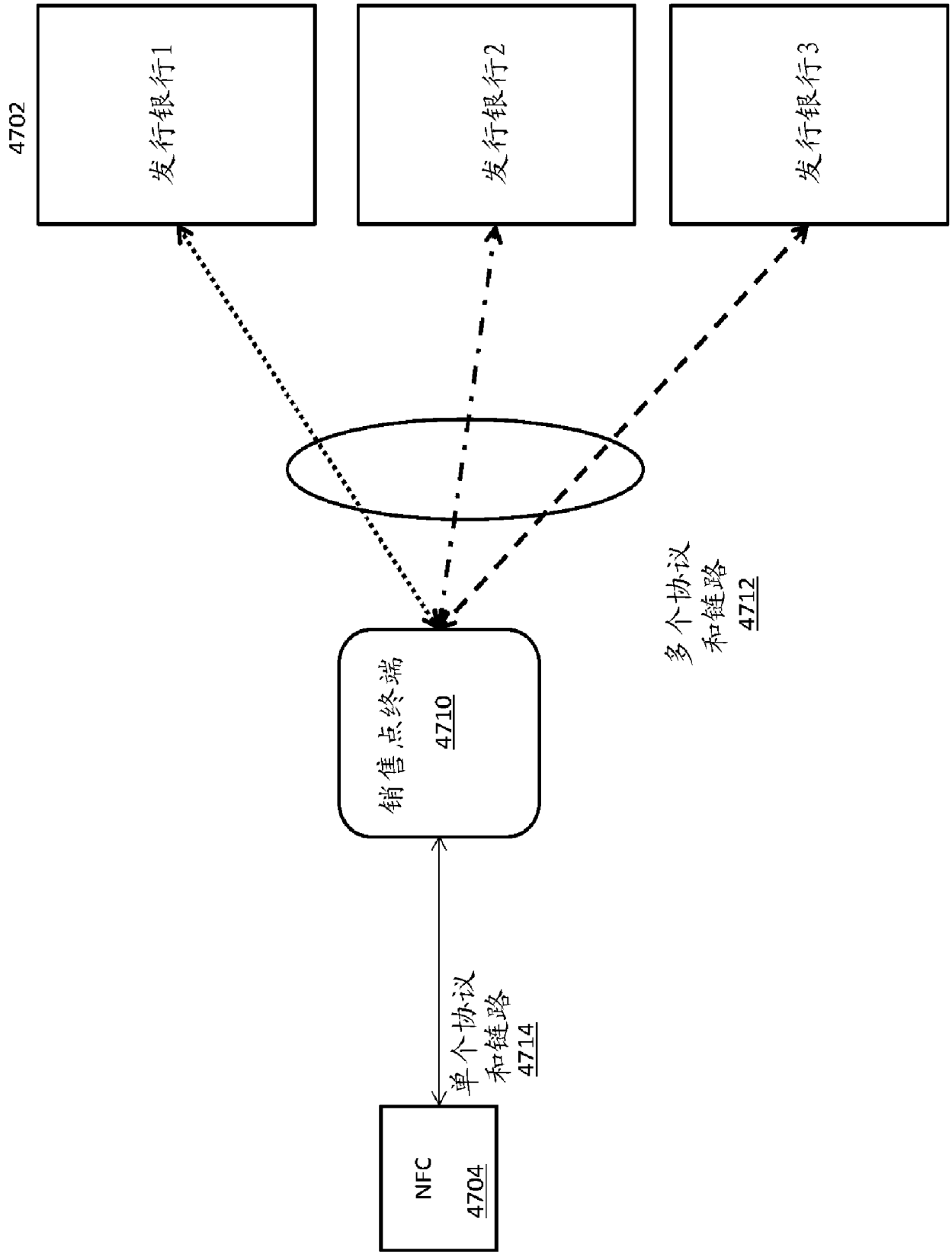

4702
发行银行1
发行银行2
发行银行3
多个协议和链路
4712
销售点终端
4710
单个协议和链路
4714
NFC
4704

图 47

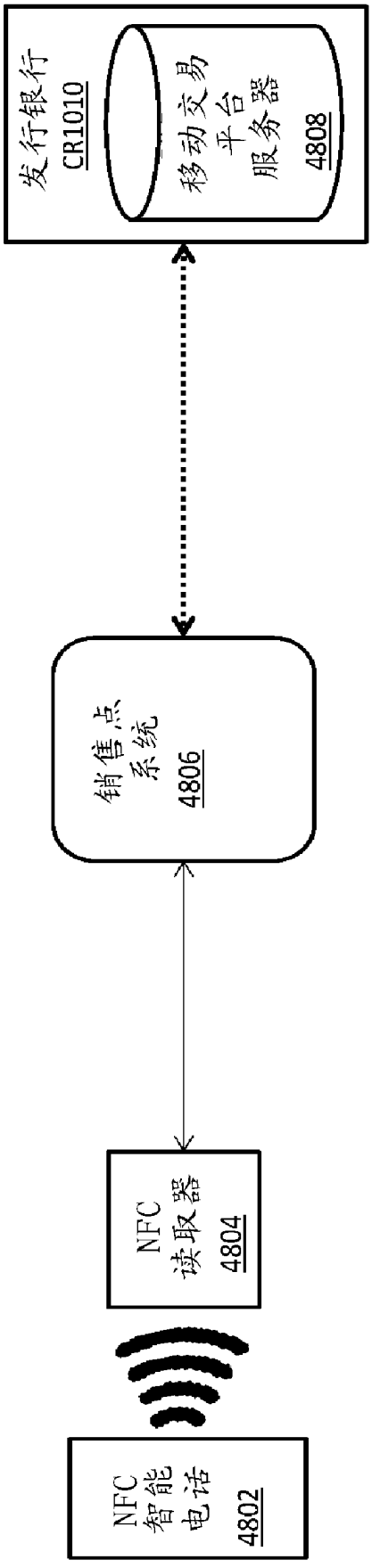

发行银行
CR1010
移动交易
平台
服务器
4808
销售点
系统
4806
NFC
读取器
4804
NFC
智能
电话
4802

图 48

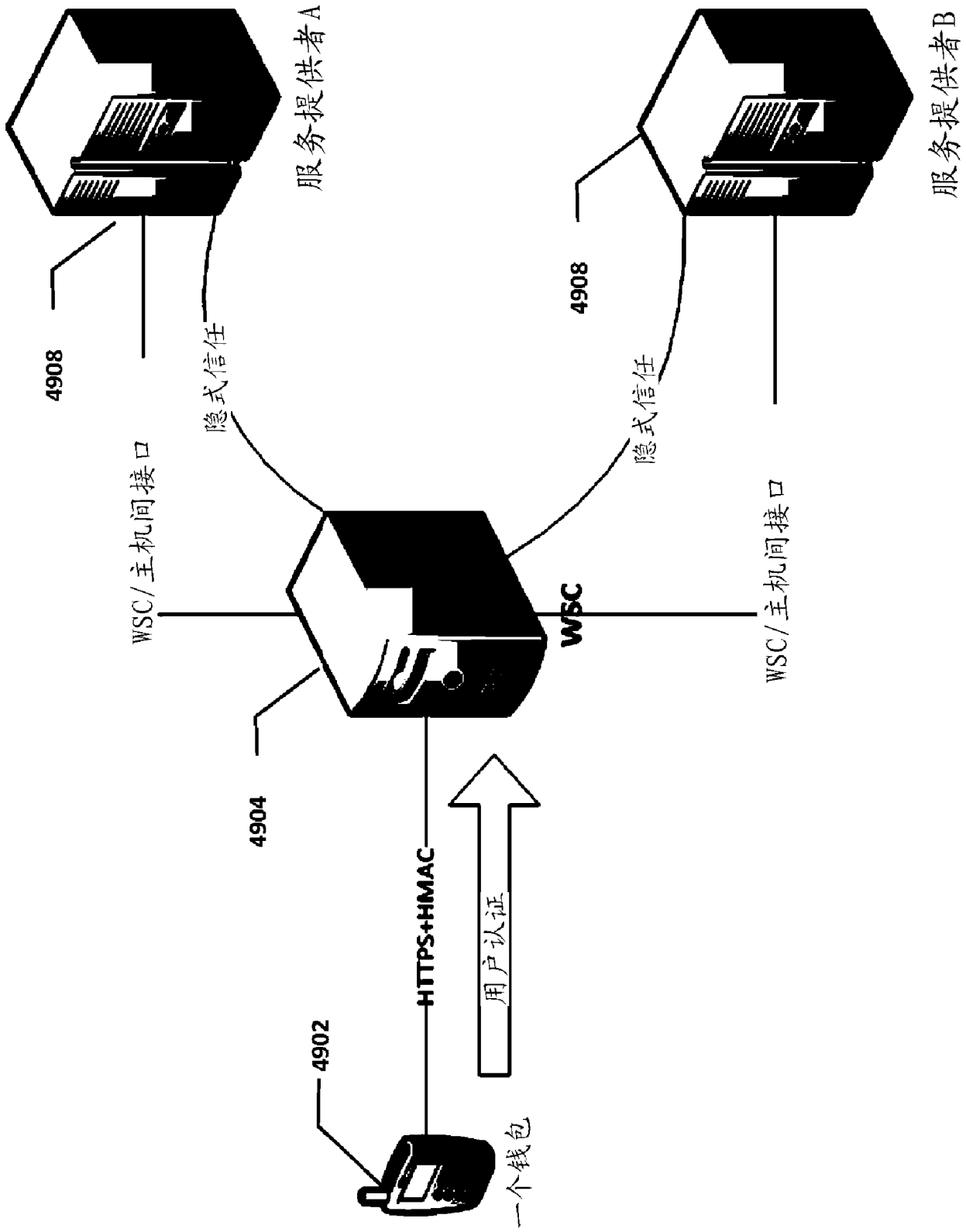

服务提供者A
服务提供者B
4908
4908
隐式信任
隐式信任
WSC/主机间接口
WSC/主机间接口
WSC
4904
HTTPS+HMAC
用户认证
4902
一个钱包

图 49

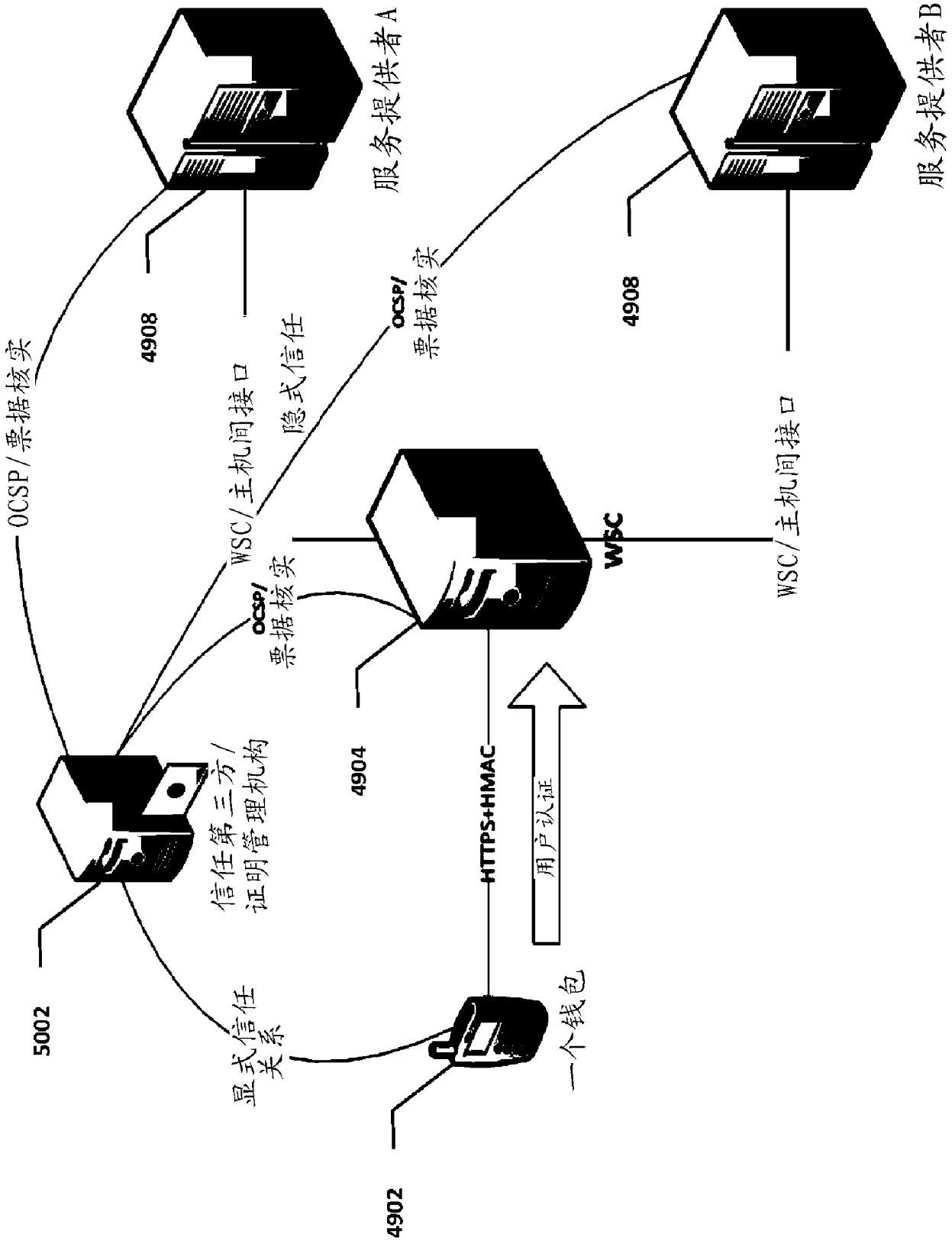

服务提供者A
服务提供者B
4908
4908
OCSP/票据核实
OCSP/票据核实
WSC/主机间接口
隐式信任
WSC/主机间接口
WSC
OCSP/票据核实
4904
信任第三方/证明管理机构
HTTPS+HMAC
用户认证
5002
显式信任关系
一个钱包
4902

图 50

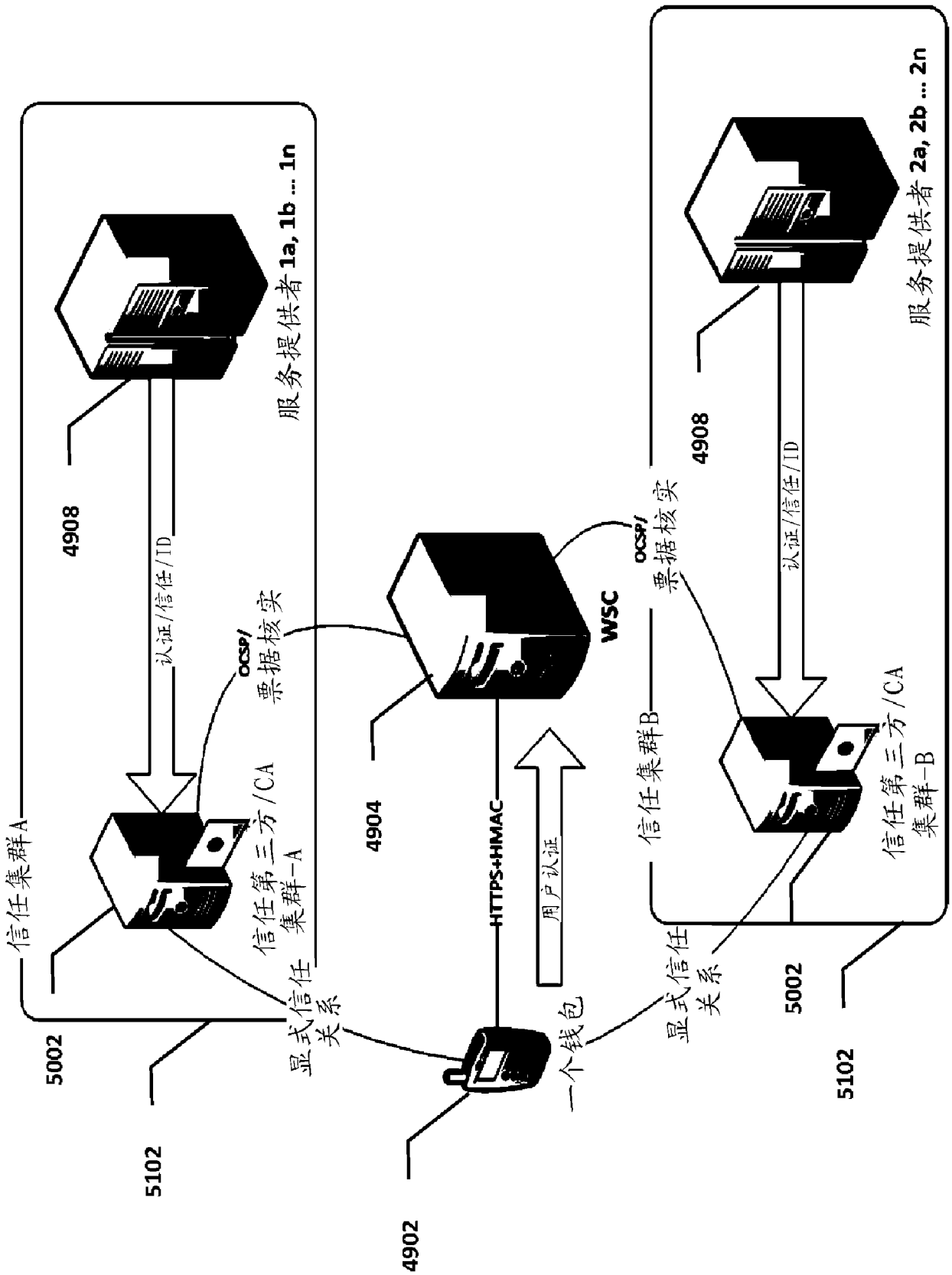

信任集群A
服务提供者 1a, 1b … 1n
4908
认证/信任/ID
OCSP/
票据核实
信任第三方/CA
集群-A
4904
WSC
HTTPS+HMAC
用户认证
一个钱包
显式信任
关系
5002
5102
4902
信任集群B
服务提供者 2a, 2b … 2n
4908
认证/信任/ID
OCSP/
票据核实
信任第三方/CA
集群-B
显式信任
关系
5002
5102

图 51

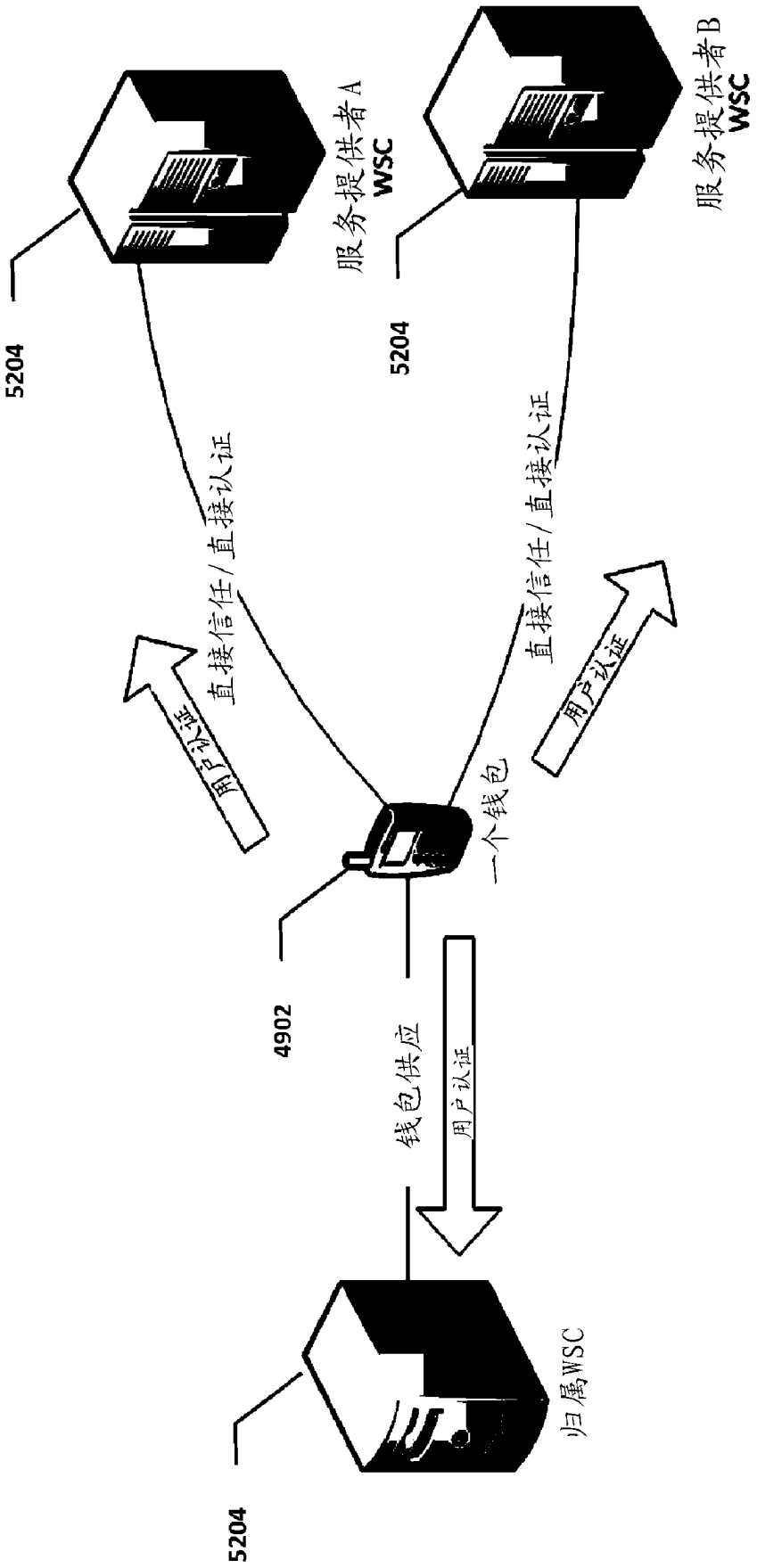

服务提供者A
WSC
服务提供者B
WSC
5204
5204
5204
4902
直接信任/直接认证
直接信任/直接认证
用户认证
用户认证
一个钱包
钱包供应
用户认证
归属WSC

图 52

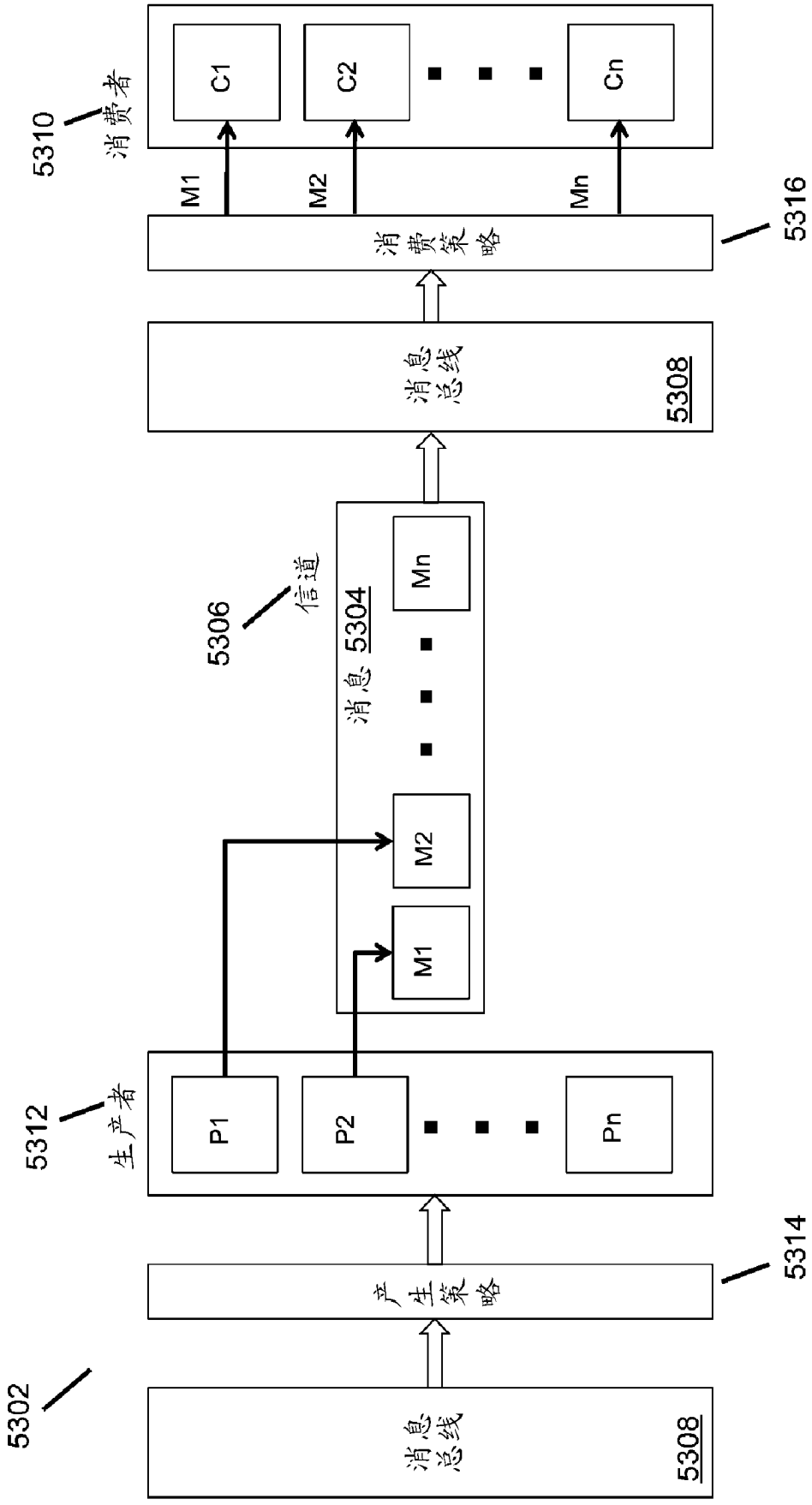

5310
消费者
C1
C2
Cn
M1
M2
Mn
5316
消费策略
消息总线
5308
5306
信道
消息 5304
Mn
M2
M1
5312
生产者
P1
P2
Pn
5314
产生策略
5302
消息总线
5308

图 53

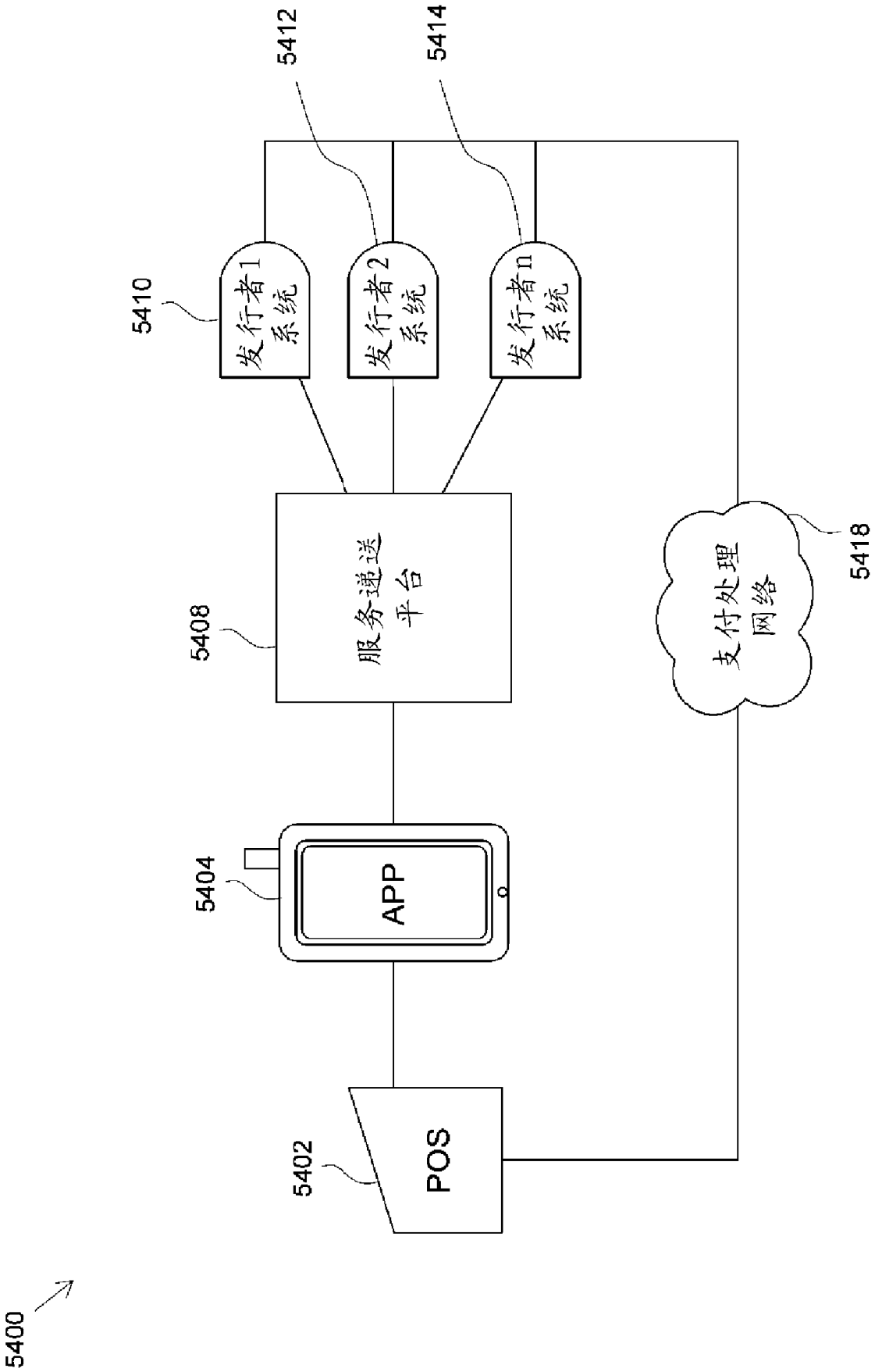

5412
5414
5410
发行者1系统
发行者2系统
发行者n系统
5408
服务递送平台
5418
支付处理网络
5404
APP
5402
POS
5400

图 54

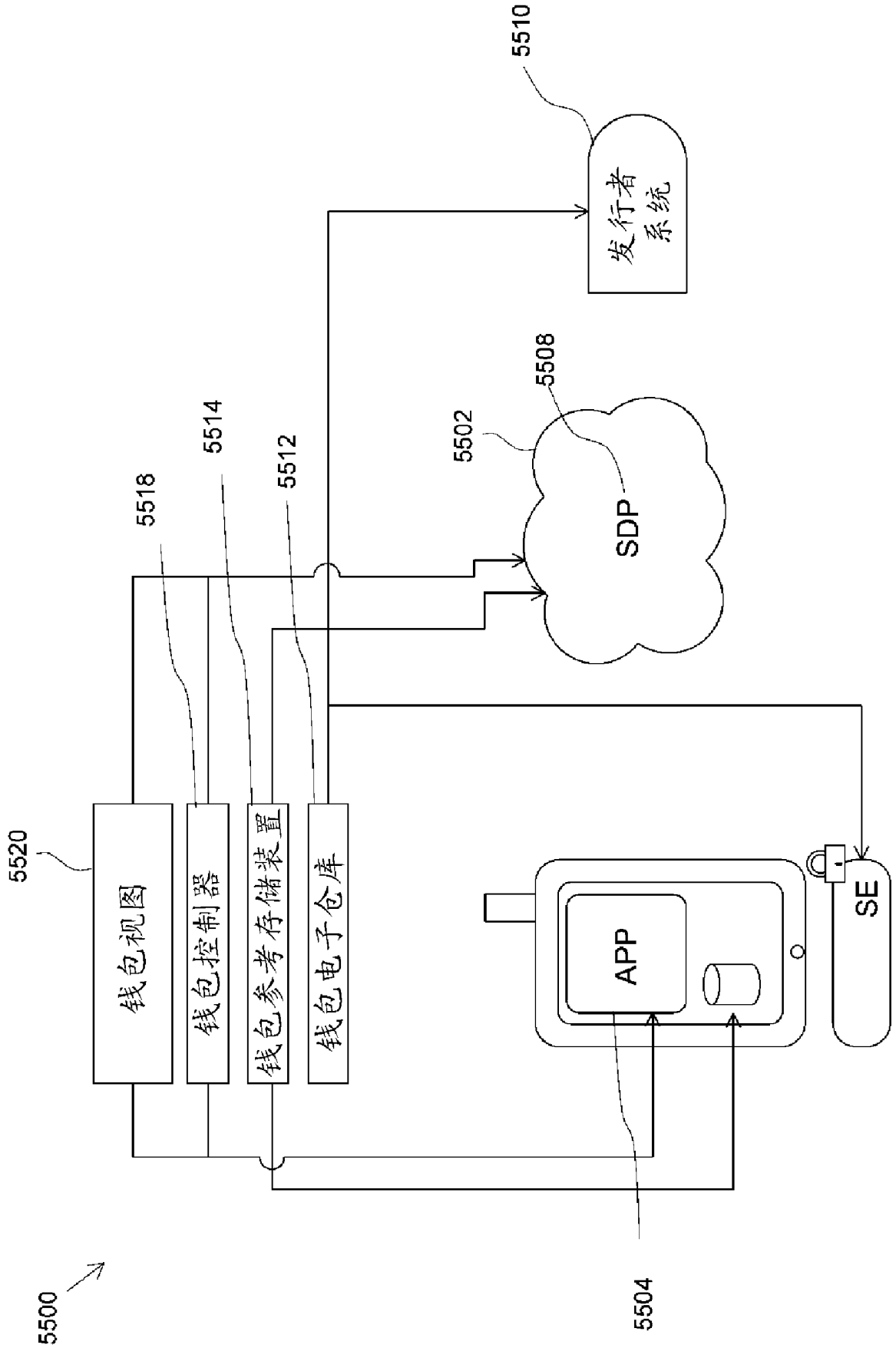

5510
发行者系统
5508
5502
SDP
5514
5512
5518
5520
钱包视图
钱包控制器
钱包参考存储装置
钱包电子仓库
APP
SE
5500
5504

图 55

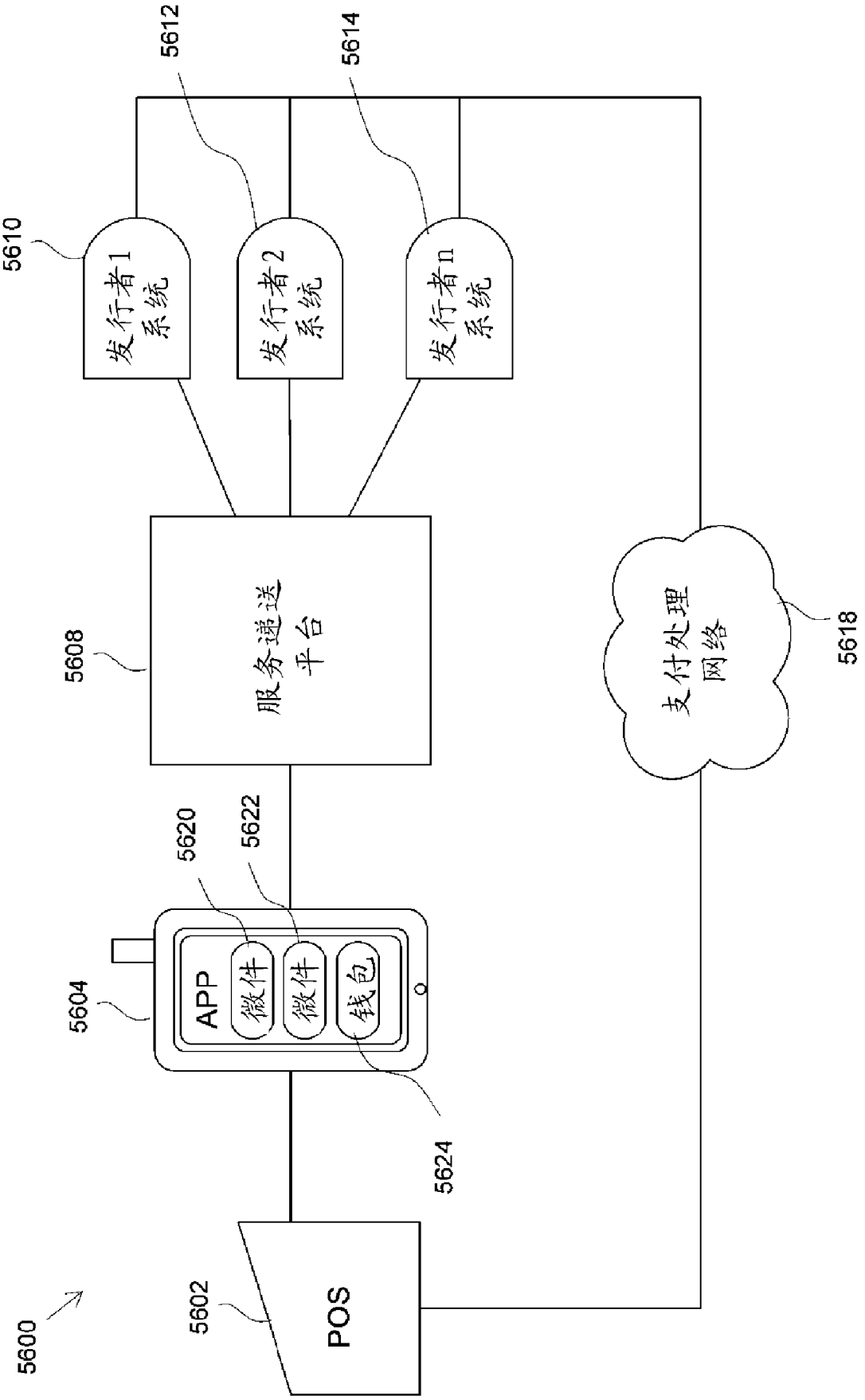

5612
5614
5610
发行者1系统
发行者2系统
发行者n系统
服务递送平台
5608
支付处理网络
5618
5620
5622
APP
微件
微件
钱包
5604
5624
POS
5602
5600

图 56