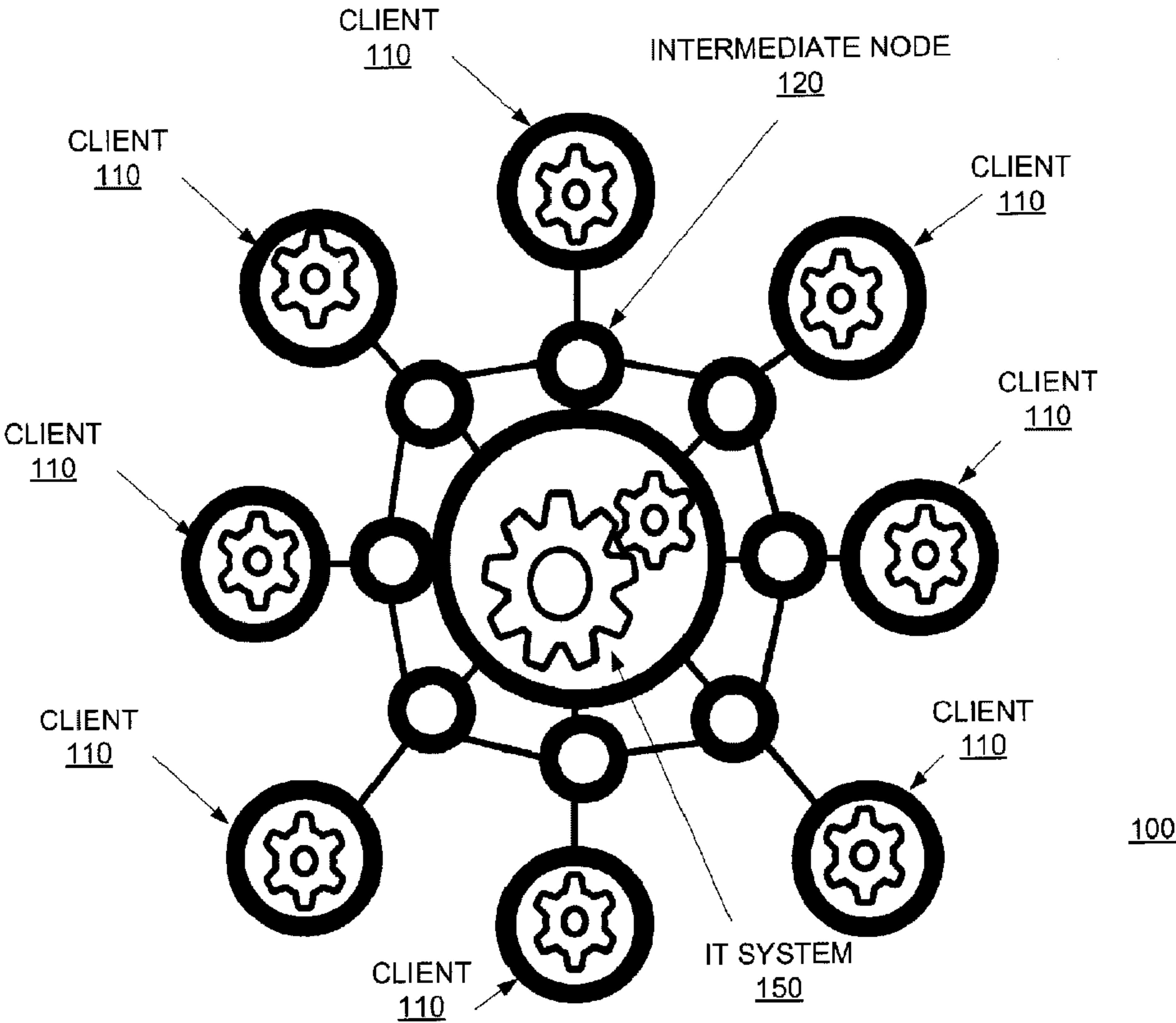




(22) Date de dépôt/Filing Date: 2015/04/07 (41) Mise à la disp. pub./Open to Public Insp.: 2015/10/07 (45) Date de délivrance/Issue Date: 2018/10/16 (30) Priorité/Priority: 2014/04/07 (US61/976,074)	(51) Cl.Int./Int.Cl. G06Q 50/10 (2012.01) (72) Inventeurs/Inventors: CHOE, MOK, CA; CHAN, PAUL MON-WAH, CA; HOSSEINIAN, ALI, CA (73) Propriétaire/Owner: THE TORONTO-DOMINION BANK, CA (74) Agent: GOWLING WLG (CANADA) LLP
--	---

(54) **Titre :** SYSTEMES ET METHODES DE PRESENTATION D'ACTIFS DE TECHNOLOGIE D'INFORMATION DANS UN ENVIRONNEMENT OUVERT

(54) **Title:** SYSTEMS AND METHODS FOR PROVIDING INFORMATION-TECHNOLOGY ASSETS IN AN OPEN ENVIRONMENT



(57) **Abrégé/Abstract:**
The disclosed embodiments include methods and systems for providing information-technology (IT) assets in an open environment. In certain aspects, an IT system may provide an online portal that presents an IT self-serve store of IT assets that a

(57) Abrégé(suite)/Abstract(continued):

user may select for access or use. The IT system may, for example, determine that the selected IT assets comply with one or more compliance and permissioning policies, and based on the determination, may assemble the selected IT assets to generate or "spin-up" instances of applications executable by various hardware platforms. The IT system may also perform processes that perform automated operations relating to certain IT assets, such as testing, modeling, recommendations, etc.

ABSTRACT OF THE DISCLOSURE

The disclosed embodiments include methods and systems for providing information-technology (IT) assets in an open environment. In certain aspects, an IT system may provide an online portal that presents an IT self-serve store of IT assets that a user may select for access or use. The IT system may, for example, determine that the selected IT assets comply with one or more compliance and permissioning policies, and based on the determination, may assemble the selected IT assets to generate or “spin-up” instances of applications executable by various hardware platforms. The IT system may also perform processes that perform automated operations relating to certain IT assets, such as testing, modeling, recommendations, etc.

SYSTEMS AND METHODS FOR PROVIDING INFORMATION-TECHNOLOGY ASSETS IN AN OPEN ENVIRONMENT

BACKGROUND

Technical Field

[001] The present disclosure generally relates to systems and methods for providing services, and more particularly to systems and methods for providing access to information technology assets in an open access environment.

Background

[002] The financial services industry continues to grow as networking and client-server technologies advance. Financial service providers, such as banks, continue to leverage the capabilities of the Internet, mobile computing, and the like, to provide services to its customers. To provide such services, financial service providers may develop hardware and software platforms that function as the backbone of such offerings. Conventional approaches include developing customized information technology (IT) services, databases, data structures, scripts, and models geared toward delivering the functionalities intended to support services for end-customers.

[003] But these approaches had drawbacks. For example, IT approaches that relied on monolithic and vertically integrated mainframe systems became too complex and customized. Distributing operational management across many nodes increased costs, resulted in frequent business outages. And, where project delivery took precedence over IT, software patches overwhelmed the integrity of the core systems, resulting in even more complex IT offerings that camouflaged the core engines that were introduced. Consequently, many conventional IT approaches result in too many

one-off tools and functionalities that are not portable across other platforms, services, or business units of an enterprise. New tools, infrastructures, and/or software may have to be developed and implemented each time a new service is envisioned. Further, access to IT resources that support such services may be limited and were incapable of being shared internally and externally to a financial service provider's enterprise. And because financial service providers may lack the resources to keep up with ever-changing technologies and capabilities, they may be forced to rely on outside sources, such as vendors, to develop, maintain, and/or host the IT backbone that support service offerings.

SUMMARY

[004] Consistent with a disclosed embodiment, an apparatus includes a storage device storing instructions and at least one processor configured to execute the instructions to receive, from a device of a user, a request to generate an instance of an application. In some aspects, the request may include information identifying a plurality of first information-technology assets available to the user. In response to the received request, the at least one processor may be further configured to determine whether the first information-technology assets are consistent with at least one permissioning or compliance policy. When the first information technology assets are determined to be consistent with the at least one permissioning or compliance policy, the at least one processor may be further configured to establish the application instance that corresponds to the first information-technology assets. The at least one processor may be further configured to generate an electronic command to transmit information associated with the established instance to the user device.

[005] Consistent with another disclosed embodiment, a computer-implemented method may receive, using at least one processor, a request to generate an instance of an application from a device of a user. In some aspects, the request may include information identifying a plurality of first information-technology assets accessible by the user. In response to the received request, the method may determine, using the at least one processor, whether the first information-technology assets are consistent with at least one of a permissioning or compliance policy associated with the user. When the first information technology assets are determined to be consistent with the at least one permissioning or compliance policy, the method may include establishing, using the at least one processor, the application instance that corresponds to the first information-technology assets. The method may also generate, using the at least one processor, an electronic command to transmit information associated with the established instance to the user device.

[006] Other embodiments of the present disclosure relate to a tangible computer-readable medium with stored instructions that, when executed by a processor, perform a method. The method may receive a request to generate an instance of an application from a device of a user. In some aspects, the request may include information identifying a plurality of first information-technology assets accessible by the user. In response to the received request, the method may determine whether the first information-technology assets are consistent with at least one of a permissioning or compliance policy associated with the user. When the first information technology assets are determined to be consistent with the at least one permissioning or compliance policy, the method may include establishing the application instance that

corresponds to the first information-technology assets. The method may also generate an electronic command to transmit information associated with the established instance to the user device.

[007] It is to be understood that both the foregoing general description and the following detailed description are exemplary and explanatory only and are not restrictive of the claims.

BRIEF DESCRIPTION OF THE DRAWINGS

[008] FIG. 1 depicts an exemplary computing environment consistent with disclosed embodiments.

[009] FIG. 2 depicts another exemplary computing environment consistent with the disclosed embodiments.

[010] FIG. 3 depicts another exemplary computing environment consistent with the disclosed embodiments.

[011] FIG. 4 depicts another exemplary computing environment consistent with the disclosed embodiments.

[012] FIG. 5 is an exemplary block diagram of certain aspects regarding the disclosed embodiments.

[013] FIG. 6 is a flowchart of an exemplary IT asset process consistent with disclosed embodiments.

[014] FIG. 7 is a flowchart of an exemplary app development IT asset process consistent with disclosed embodiments.

[015] FIG. 8 depicts an exemplary interface for an IT self-serve store consistent with disclosed embodiments.

[016] FIG. 9 is an exemplary block diagram of certain aspects regarding the disclosed embodiments.

[017] FIG. 10A and 10B are flowcharts of exemplary IT asset processes consistent with disclosed embodiments.

[018] FIG. 10C is a flowchart of an exemplary process for certifying, registering, and consuming IT assets consistent with disclosed embodiments.

[019] FIG. 11 is an exemplary block diagram of certain aspects regarding the disclosed embodiments.

[020] FIG. 12 is a flowchart of an exemplary IT asset execution process consistent with disclosed embodiments.

[021] FIGS. 13 and 14 illustrate exemplary layered services, consistent with disclosed embodiments.

[022] FIG. 15 is a flowchart of an exemplary process for generating instances of requested applications, consistent with disclosed embodiments

DESCRIPTION OF THE EMBODIMENTS

[023] Reference will now be made in detail to embodiments of the present disclosure, examples of which are illustrated in the accompanying drawings. Wherever possible, the same reference numbers will be used throughout the drawings to refer to the same or like parts.

[024] FIG. 1 illustrates an exemplary computing environment 100 consistent with certain disclosed embodiments. In one aspect, computing environment 100 may include one or more clients 110, one or more intermediate nodes 120, an IT system

150, and a communications network interconnecting one or more of the components of environment 100.

[025] In one embodiment, IT system 150 may be one or more computer systems configured to process and store information, and execute software instructions to perform one or more processes consistent with the disclosed embodiments. In certain embodiments, IT system 150 may include one or more servers and memories. Additional details of exemplary system components of an exemplary IT system 150 is described below in connection with FIG. 2.

[026] A client 110 may be one or more client devices. In certain embodiments, client 110 may be associated with one or more users. In one example, a user may use client 110 to provide input that may be used to perform one or more processes consistent with the disclosed embodiments. In certain aspects, client 110 can include, but is not limited to, a personal computer, a laptop computer, a tablet computer, a notebook computer, a hand-held computer, a personal digital assistant, a portable navigation device, a mobile phone, a wearable device, an embedded device, a smart phone, and any additional or alternate computing device. Client 110 may be implemented with one or more processors or computer-based systems. As shown in FIG. 1, system environment 100 (and other environments consistent with certain embodiments) may include multiple clients 110 that may be each associated with one or more users.

[027] An intermediate node 120 may be a computing device that links clients 110 to IT system 150. Aspects of the disclosed embodiments may be configured without intermediate node(s) 120. For example, intermediate nodes may be routing

devices that execute software to intelligently route requests and responses to/from a client 110 and IT system 150. In other aspects, intermediate nodes 120 may perform other operations or functionalities.

[028] System environment 100 shows the components interconnected by a network of communication links (not labeled). In one embodiment, the network may be a cloud-based network. For instance, the network may be based on a wired or wireless Metropolitan Area Network (MAN) connecting multiple wireless LANs to form a wide area network ("WAN"). For example, consistent with embodiments of the present disclosure, the network may include the Internet and any publicly accessible network or networks interconnected via one or more communication protocols, including, but not limited to, hypertext transfer protocol (HTTP) and transmission control protocol/internet protocol (TCP/IP).

[029] FIG. 2 shows an exemplary system environment 200 (that may be similar to environment 100) showing exemplary system components of IT system 150. For instance, IT system 150 may be configured with one or more servers 252 and one or more memories 254. Server 252 may be, for example, a computing system that processes, among other things, IT asset requests consistent with certain embodiments. Server 252 may also execute software (stored in memory 254 or elsewhere) that provides IT services or access to IT services that may be provided by remote or different computer systems.

[030] In one embodiment, server 252 may include separate computing components that provide operations individually or collectively, although the configuration of server 252 is not limited to such configurations. In one example, server

252 may be incorporated into a single computer, a single server, or any additional or alternate computing device apparent to one or skill in the art. In other embodiments, server 252 may include distributed computing devices. Further, in one embodiment, software programs, such as a software application (e.g., a web service) may be executed by one or more processors included in server 252. Server 252 may be configured to execute software instructions to perform one or more processes consistent with the disclosed embodiments. In one embodiment, client devices 210, 220 may exchange information with system 150.

[031] IT system 150 may include one or more web servers or similar computing components that provide online portals (either through the Internet or other forms of networks) that are accessible by users internal and/or external to IT system 150 (or an entity associated with IT system 150). For example, IT system 150 may perform processes that generate and provide interfaces with content relating to open IT asset operations consistent with the disclosed embodiments.

[032] Memory 254 may be one or more data storages configured to store information and software instructions consistent with the disclosed embodiments. In one aspect, memory 254 may be a data repository that may include actual or virtual customer data, account data, and transaction data. In one aspect, customer data may include one or more data records uniquely identifying one or more customers of a business entity associated with system 150. Alternatively, customer data may include virtual data (e.g., data that has similar format, fields, configurations, etc. of actual customer data, but with random or otherwise “fake” data values and customer identifications and/or accounts. For example, customer data may include personal

information associated with a user or a virtual user (e.g., a name, home address, or date of birth), demographic information (e.g., educational level, income level), government-issued identifiers (e.g., driver's license numbers or Social Security numbers), employment information (e.g., employer name or address), and/or contact information (e.g., e-mail addresses, home numbers, work numbers, or mobile numbers). Other types of customer information may be stored and used by the disclosed embodiments.

[033] In certain aspects, account data may include account identification information identifying one or more actual or virtual accounts of customers of a financial service provider (e.g., a business entity associated with system 150). In one embodiment, account identification information may include financial service account information. For example, such service account information may include a checking account, a savings account, a revolving credit line, an account linked to a credit or debit card, a brokerage account, and any additional or alternate account provided or supported by the issuing bank. In other embodiments, account data may include information identifying actual or virtual investment portfolios held by one or more customers of the financial institution.

[034] Transaction data may include information identifying one or more actual or virtual transactions involving one or more actual or virtual customers or accounts of a financial service provider. In one embodiment, such transactions may include, but are not limited to, purchase transactions (e.g., purchases of goods and/or services from electronic or physical retailers), financial service transactions (e.g., fund transfers), bill payment transactions (e.g., electronic bill payment transactions), financial instrument or

security transactions (e.g., purchases of securities), deposits or withdrawals of funds, or applications for credit from the financial institution or other entity.

[035] In exemplary system environment 200 (which may be similar to environment 100), client devices 210 and 220 (that may correspond to a client 110) may be operated by a respective user (e.g., user 211, 221). Client devices 210 and 220 may connect to IT system 150 over a network (e.g., network 225) that may be similar to the network disclosed above in connection with FIG. 1.

[036] Clients (including client 110, client devices 210, 220 or other clients disclosed herein), IT system 150, or other computer systems disclosed herein may include computing components known to those skilled in the art that may be configured, programmed, or the like in specific manners that allow processes or operations to be performed consistent with disclosed embodiments. Such components may include one or more processors, memory devices (e.g., tangible and non-transitory computer-readable medium(s) having stored therein computer programs, sets of instructions, code, or data to be executed by processor(s)).

[037] In certain embodiments, the term "computer-readable medium" may refer to devices including, but not limited to, tangible and non-transitory memory device mediums, which may provide computer programs and/or sets of instructions executable by one or more processors. Such computer programs and sets of instructions can be stored within one or more computer-readable media. Additionally or alternatively, computer programs and sets of instructions may also be received via a communications interface and stored on the one or more computer-readable media. Such computer programs and instructions, when executed by a processor, may enable the processor to

perform one or more processes consistent with the disclosed embodiments. Examples of program instructions include, for example, machine code, such as code produced by a compiler, and files containing a high-level code that can be executed by the processor using an interpreter.

[038] The disclosed embodiments may provide computer-implemented systems and methods for providing IT assets. An IT asset, in certain embodiments, may be a service, software, process, information, etc. that may be generated, maintained, linked, or otherwise provided by IT system 150 that may be used for IT services. For example, an IT asset may be a software patch that may be provided for one or more financial service provider services provided by a financial service provider. An IT asset may also include data, such as transaction data that is associated with financial service transactions. Transaction data may be associated with actual customer transactions of actual customers of a financial service provider. In other aspects, transaction data may reflect test transaction data that has parameters and attributes similar to actual transactions, but are not affiliated with an actual customer of a financial service provider. In other aspects, an IT asset may reflect packages of IT services, that may include software, workflows, regulatory information, and the like. An IT asset may also reflect any of these exemplary information or processes individually. As another example, an IT asset may also be a model, which may be software that, when executed, provides modeling processes for certain use cases, applications, or environments. An IT asset may also reflect an API that may be preconfigured by IT system 150 (or preconfigured and programmed by a user and stored and made available by IT system 150). Alternatively, an API IT asset may be software that may be

used by a user or system (e.g., client 110) to develop, build, test, and run an API to provide access to another IT asset, or a service offered by a system, such as IT system 150, client 110, or another system. Other examples of IT assets may be described herein and the above list is not limiting or exhaustive of the type of IT assets that may be provided, used, generated, maintained, and implemented by the disclosed embodiments.

[039] In certain embodiments, a client (e.g., 110, 210, 220) may request access to IT system 150 to request one or more IT assets, or to perform one or more IT services (e.g., tests, modeling, etc.). In certain aspects, access to IT assets provided by IT system may be through one or more APIs. FIG. 3 shows an exemplary system environment 300 including IT system 150, network 225 interconnected through one or more APIs 312, 314, 316 (although the disclosed embodiments may be implemented with one or any number of APIs). In certain aspects, such APIs may provide primary access in and external to a business entity associated with IT system 150. IT system 150 may store relevant data and software that may drive decision making for building services (e.g., financial service provider services). IT system 150 may be configured to provide on-demand access to IT assets through one or more APIs, which may be linked individually, or collectively, to one or more IT assets.

[040] FIG. 4 shows another exemplary system environment 400 consistent with disclosed embodiments. System environment 400 may include client devices 410, 420, IT system 150, interconnected by network 425. In certain embodiments, client devices 410 and 420 may be similar in configuration, operation, and functionality as a client 110 described above in connection with FIG. 1. Client device 410 may be

associated with a user 411 and client device 420 may be associated with a user 421. IT system 150 in FIG. 4 may be similar in configuration, operation, and functionality as IT system 150 of FIG. 1. In certain exemplary embodiments, although not required, IT system 150 may be associated with a business entity 455. In certain embodiments, business entity 455 may be any type of business entity, such as a financial service provider. For example, business entity 455 may be a commercial bank, an investment bank, a provider of a payment instrument or financial service accounts, etc. In some embodiments, a financial service account may be a check, savings, credit, debit, and/or a reward or loyalty account. In some aspects, a payment instrument may include, but is not limited to, a personal or corporate credit card, a debit card, or check instruments.

[041] Business entity 455 may be an entity that provides accounts that may be used for purchasing goods and/or services, which may, in certain embodiments, be used for purchases associated with one or more merchants. Business entity 455 may be an entity that provides IT services consistent with certain disclosed embodiments. In one aspect, business entity 455 may be a financial service provider that provides financial services to customers, and may provide IT services to one or more business units associated with business entity 455 or external and not affiliated with business entity 455. For example, client device 410 may be a client that is operated by a user 411 who is associated with business entity 415. In one embodiment, business entity 415 may be the same business entity as entity 455, or may be a different entity. Entity 415 may be a business unit or other type of department, organization, etc. associated with business entity 455 (e.g., a card services division, a payment services division, etc.). Entity 425 may be, in this example, a third party business entity in relation to

entity 455. For example, entity 425 may be a business that provides mobile applications (apps) for financial service providers or their customers. Client devices 410 and/or 420 need not be associated with a business entity. For example, user 421 may be a user that seeks to access and use IT asserts provided by IT system 150, but has no affiliation with a business entity.

[042] In certain embodiments, IT system 150 may be configured to provide IT assets in the form of apps that may be configured as core building blocks for larger applications that, when executed, provide IT services used to provide other types of services, such as financial provider services. For example, IT system 150 may generate and store apps that provide granular capabilities that act as the core building block, which may be assembled into larger apps that provide services. Such services may provide legendary customer and user experiences (e.g., financial service related services) that are not limited by temporal limitations (e.g., services and assets available at any time), location limitations (e.g., services and assets available from anywhere via the cloud), or device limitations (e.g., services and assets available for different types of client devices). IT system 150 may also provide APIs that may facilitate a standard method of access one or more core engines (e.g., software processes configured to provide core functionalities for certain services) and hide complexities of connecting to the engines. Moreover, in certain aspects, IT system 150 may be configured to provide on-demand infrastructures that allow IT assets, services, operations, and the like to be mobilized, shared, and ported to other systems and services, in a time-efficient manner. The disclosed embodiments enable a business entity associated with IT system 150 to provide IT as a service. In certain aspects, IT system 150 may be configured to control

access to IT assets through a number of APIs that may, for example, be configured to limit the variations in their functionalities. Moreover, IT system 150 may, through processes and configurations consistent with disclosed embodiments, provide a controlled environment for users (e.g., internal or external to a business entity associated with IT system 150). In one example, IT system 150 may generate, maintain, and provide container-based architectures and certifications that control consumption of IT assets and how clients may use them. For example, IT system 150 may provide common container apps, SDKs, and the like, that democratize consumption. IT system 150 may also be configured to provide a self-serve portal that provides access, through cloud-based computing, to IT assets. Such a portal may offer access to IT assets through a menu-based (or other formatted access) interface(s).

FIG. 5 shows an exemplary block diagram of exemplary features of certain disclosed embodiments consistent with one or more of these features.

[043] In certain aspects, certain embodiments may provide processes for requesting, receiving, accessing, using, testing, etc. of IT assets. Users internal or external to a business entity associated with IT system 150 may wish to search for, identify, and gain access to such assets to assist in developing a service for a business, such as a financial service for customers. IT system 150 may be configured to store preprogrammed and configured IT scripts, programs, and the like that may, when executed by a processor, automatically perform IT processes that would typically take conventional approaches a long time to configure for certain services. This, coupled with the on-demand accessibility of such automated IT processes, enable users (including less technologically sophisticated users) to consumer the processes in real-

time. For example, a business stakeholder (e.g., a CIO, lead developer, etc.) for a financial service provider may wish to develop a regulatory project associated with a financial service provider project.

[044] Through certain disclosed embodiments, the user may access, through for example the self-service IT asset store consistent with disclosed embodiments, one or more IT assets to being modeling or executing models to develop, test, and confirm the project. For instance, the business stakeholder may request certain regulation guidelines that may be stored as IT assets by IT system 150 and request proof of concept functions for certain regulation-related concepts. IT system 150 may be configured to automatically determine, based on identified regulation project inquiry responses (e.g., provided by the stakeholder to IT system 150 through an interactive menu-driven interface) regulation guidelines, data or variables, and models that may be needed to authenticate and/or prove the project.

[045] In one example, IT system 150 may provide one or more preconfigured models that may be related to certain regulation project concepts, and automatically identify and/or provide the models to the stakeholder through the portal. In certain embodiments, IT system 150 may run the models using data identified by the stakeholder or by system 150, and provide results of the models. Alternatively, IT system 150 may provide access to IT assets that enable the stakeholder to generate, develop, build and run a model that may be customized for the desired regulatory project, based on core engine capabilities that may be reusable by other systems or users.

[046] As another example, a user associated with a client external to a business entity associated with IT system 150 (e.g., user 421, client device 420, entity 425), may wish to develop, test, prove, and/or perform other tasks for a new payment app for a financial service provider service. In this example, the user may request access to one or more IT assets that will facilitate those needs through the online portal provided by IT system 150. For instance, the user may request IT assets in the form of transaction data (virtual or real) to test the operation of the payment app under development. IT system 150 may perform processes that authenticate and determine authorization to such information by the user (e.g., credential testing, associate partnership agreements, etc.). IT system 150 may provide the data to the user (e.g., via client device 420, APIs, network 425, etc.). Alternatively, or in addition, IT system 150 may provide IT assets in the form of app test processes that allow the user to not only develop the app and configure a test script for the app, but also run tests on the app using data provided by IT system 150. In certain aspects, IT system 150 may provide access to knowledge engines and models that user can leverage to refine the design of the app to maximize the effectiveness of the app. In other embodiments, IT system 150 may offer IT assets that automate production of the app. This may include, for example, access to references (links, etc.) to other IT services that the user may access to complete production of the app.

[047] IT system 150 may be configured to execute processes that monitor the access and use of IT assets. It may perform metering processes that determine the number, types, and the like of IT assets requested and accessed by users, and determine, if applicable, appropriate fees for such access and use. IT system 150 may

be configured to determine costs for access to one or more IT assets, and provide interfaces, reports, and automated payment services, for charging the fees and, in some embodiments, collecting payment for such fees. For example, IT system 150 may execute software that generates shopping cart information when a client request access to one or more IT assets, and upon payment, provides the IT assets (or access thereto). Other forms of metering access to IT assets consistent with the disclosed embodiments may be implemented and provided.

[048] FIG. 6 is a flowchart of an exemplary IT asset process consistent with disclosed embodiments. In one example, IT system 150 may receive an IT asset request (Step 610). The request may come from a client (e.g., client 110, 210, 220, 410, 420). In certain embodiments, the request may be received during an online session between the client and IT system 150's online portal (e.g., self-serve IT asset store). A user operating the client may select on an interface provided via the portal one or more IT assets. Based on the request, IT system 150 may execute processes that automatically identify the requested IT asset(s) (step 620). In one example, the identified IT asset may be directly related to the IT asset identified in the IT asset request. In other embodiments, IT system 150 may determine one or more IT assets that best relate to the request received by the client. In certain embodiments, IT system 150 may also identify any IT sub-assets of the identified IT asset (step 630). An IT sub-asset may be an IT asset that is related to, required by, or suggested for operation with, another IT asset. For example, an IT sub-asset may be a particular script or macro software program that is required to allow another program identified as the primary IT

asset identified by IT system 620. Other examples of an IT sub-asset may include an API, model, test script, html programs, or the like.

[049] In certain aspects, IT system 150 may perform processes that generate an IT asset checkout cart (step 640). For example, IT system 150 may track the IT asset(s) and/or sub-assets that have been identified, and determine a fee, cost, or other form of metering value, for the identified assets. For example, IT system 150 may be configured to assign monetary fee values for IT assets and sub-assets that are requested by a user who is external to the business entity associated with IT system 150. In certain aspects, IT system 150 may determine not to charge a monetary fee for the same identified assets and sub-assets when requested by a user (and client) associated with the business entity related to IT system 150. The disclosed embodiments are not limited to any particular manner of metering access to IT assets and one of ordinary skill in the art would appreciate that the disclosed embodiments may implement other ways of charging fees or tracking access and use to IT assets.

[050] In another embodiment, IT system 150 may be configured to generate and provide an IT asset canvas (step 650). In certain aspects, an IT asset canvas may be a representation of IT assets and their relationship to other IT assets based on the type of use for those assets. For example, IT system 150 may determine based on the identified IT assets and/or sub-assets, and from input from the requesting user regarding a scope of use for the requested assets (e.g., regulatory project, app development, etc.), a relationship canvas that links the IT assets to functionalities, use, and capabilities, to each other and other assets not requested by the user. IT system 150 may be configured to generate and provide an interface that provides a graphically

representation of the canvas showing the relationships between the requested and unrequested assets. In certain aspects, IT system 150 may execute software that generates recommendations for the unrequested assets with explanations for their need, requirement, and/or suggested need or use. IT system 150 may generate content reflecting the recommendations (and/or canvas) that is provided for display in an interface to the client of the user.

[051] IT system 150 may also provide access to, or provide, the identified IT assets and/or sub-assets (step 660). In certain embodiments, IT system 150 may electronically transmit the identified IT assets and/or sub-assets to the client requesting the IT asset(s). In other embodiments, IT system 150 may provide a link that when selected by the user via the client, provides access to the IT asset(s). In other embodiments, IT system 150 may provide access to IT assets in the form of executable programs (e.g., test scripts, production processes, development or design applications, etc.) that allow the user via the client to configure and run IT assets through IT system 150 remotely over the interconnecting network between IT system 150 and the client.

[052] For example, FIG. 7 shows a flowchart of an exemplary app development IT asset process consistent with disclosed embodiments. In one aspect, steps 710-730 may be similar to process steps 610-630 of FIG. 6, but based on the example of an app development project. In this exemplary embodiment, IT system 150 may also provide test data (e.g., step 740) that may be used by the user (via the client) to test functionality etc. of the app under development, which may be based on the identified IT assets (e.g., identified via steps 710-730). For example, IT system 150 may provide actual or virtual data that may be automatically or manually configured into

an IT asset for running a test associated with the app under development. In addition, consistent with the examples mentioned above, IT system 150 may be configured to provide and run a test environment (e.g., step 750) for performing the test of the app based on the test data. In further embodiments, IT system 150 may also provide an app production service that may provide services, or access to services, that enable the requesting user to refine and complete production and launch of the developed app (step 760).

[053] FIG. 8 shows an exemplary interface of an exemplary self-service IT asset store consistent with disclosed embodiments. In one embodiment, IT system 150 may provide an interface through an online portal that includes one or more options for selecting one or more IT assets. For example, as shown in FIG. 8, the interface may include different categories (e.g., infrastructure store, data store, container store, API and APP store, Management store, Automation store, and Policy store) of IT assets that may be selected by a user via a client (e.g., client 110 display the interface through a browser or similar software executing on the client). Each category may include one or more IT assets (and sub-assets) that relate to IT services, operations, functionalities, etc. associated with the category. For instance, IT system 150 may provide a container store category that includes one or more software containers that a user may select to use for a particular use case. IT system 150 may provide a Policy store that includes data and features regarding policies, such as payment card industry (PCI) policies. The exemplary API and app store category may include different APIs that a user may select to use for certain use cases. For instance, an application developer may select one or more APIs from the API and app store to use in developing an app. Also in that

category may be one or more apps that may be used as core applications that may be built upon by the developer.

[054] The Automation store category may include IT assets that provide automated processes that may be used by a user remotely through the network connecting the client with IT system 150. For example, IT system 150 may provide automated design IT assets, automated testing IT assets, and “DevOps,” which may be a production development IT asset that provides features that allow a user to develop and move to production an app under development. IT system 150 may also provide a management store category that includes IT assets relating to managing a business, such as workflows regarding approval, billing, inventory management, and the like. IT system 150 may also provide an infrastructure store that may include IT assets relating to infrastructure assets, such as memory, server functionalities, etc. For instance, IT system 150 may provide access to operating system software, patches, and the like that a user may access, load, and/or use (via the client) for certain service-related reasons.

[055] IT system 150 may also provide a data store category that includes IT assets that provide data, information, models, reports, reporting software, etc. that may be used by a user (via the client). For example, a user may request and access or obtain data (virtual or real) relating to financial service transactions for testing, proving, and/or designing a transaction payment app. IT system 150 may provide a data catalogue that may include a user-friendly catalog of different use-case data that a user may select based on categories (e.g., data for purchase transactions, data segmented based on customer demographics (virtual or real), data segmented based on transaction types, transaction size (amount of purchase, transfer, etc.)).

[056] In some aspects, an application developer may access the self-serve IT store provided by IT system 150 and obtain apps, APIs, widgets, and other IT assets that may be assembled by an application developer (e.g., user 211) to develop, test, benchmark, and provision mobile apps. The mobile apps may, for example, be capable of providing a particular customer and/or user experience when executed by communications devices implementing various operating systems (e.g., iOS™, Android™, BlackBerry OS™, etc.) and various web browsers (e.g., Internet Explorer™, Safari™, Chrome™, etc.).

[057] In certain embodiments, and prior to accessing the self-serve IT store, the application developer (e.g., user 211) may provide one or more authentication credentials (e.g., a user name, password, biometric credential, etc.) as input to a web page or other graphical user interface (GUI) associated with IT system 150 and presented by a device (e.g., client device 210). Client device 210 may, in some aspects, be configured to transmit the received authentication data to IT system 150. IT system 150 may access profile data associated with user 211, and based on one or more compliance and permissioning policies associated with user 211 and/or the self-serve IT store, IT system 150 may selectively grant user 211 permission to access to the self-server IT store, to access different of IT assets within the self-serve IT store (e.g., the infrastructure store, the data store, the container store, the API and APP store, Management store, Automation store, and Policy store), and additionally or alternatively, to access specific types of IT assets or specific IT assets within the individual categories.

[058] In certain instances, based on the accessed profile data and one or more of the compliance and permissioning policies, IT system 150 may determine that user 211 lacks the ability to access the self-serve IT store. For example, user 211 may be a new hire within an organization or other business entity, and may lack status within the business entity sufficient to access the self-serve IT asset store. In other example, user 211 may be associated with a prior violation of terms and conditions associated with the self-serve IT asset store, and IT system 150 may suspend user 211's ability to access the self-serve IT asset store for a certain time period. In some aspects, upon determination that user 211 is unable to access the self-serve IT asset store, IT system 150 may transmit a notification to device 210 that indicates the lack of access, which device 210 may render for presentation to user 211.

[059] In other instances, IT system 150 may allow user 211 to access only certain categories of IT assets available within the self-serve IT asset store. By way of example, based on the accessed profile data and one or more of the compliance and permissioning policies, IT system 150 may determine that user 211 has permission to access certain categories of IT assets (e.g., the infrastructure store, the data store, the container store, and/or the API and APP store), but may be unable to access other categories of IT assets (e.g., the Management store, the Automation store, and/or the Policy store). In certain aspects, IT system 150 may be configured to transmit data to device 210 that indicates user 211's ability to access the self-serve IT asset store and identifies the categories of IT assets available to user 211. Device 210 may be configured to render and present the interface corresponding to the self-serve IT asset

store, and may populate the self-serve IT asset store with only those categories of IT assets available to user 211.

[060] Further, in some aspects, IT system 150 may grant user 211 permission to access only specific assets or types of assets within a particular category of IT assets available within the self-serve IT asset store. For example, and based on the accessed profile data and one or more of the compliance and permissioning policies, IT system 150 may grant user 211 permission to access the Data Store within the self-serve IT asset store, but may block user 211's access of any customer data and/or IT assets related to analytics or reporting within the Data Store (e.g., "Model Analytics" and/or "Reports"). In some aspects, IT system 150 may be configured to transmit data to device 210 indicative of the specific assets and/or asset types available to user 211 within a corresponding category of IT assets. Upon receipt of the transmitted data (and additionally or alternatively, upon detection of an attempt by user 211 to access the corresponding category of IT assets), device 210 may present the available assets and/or asset types within the interface of the self-serve IT asset store. In other aspects, however, device 210 may be configured to present to user 211 all assets and asset types associated with a specific category of IT assets within the self-serve IT asset store, and to render and present an error message when user 211 attempt to access a prohibited asset or asset type.

[061] In the embodiments described above, IT system 150 may be configured to determines grant user 211 permission to access to the self-server IT store, to access different of IT assets within the self-serve IT store, and additionally or alternatively, specific types of IT assets or specific IT assets within the individual categories. In other

aspects, however, device 210 may be configured to store user profile data and the one or more compliance and permissioning policies, and may be configured to grant permission to user 211 to access any combination of the self-serve IT asset store, categories of IT assets, and/or specific assets or asset types independently of IT system 150 and/or in conjunction with IT system 150.

[062] In an embodiment, the apps, APIs, widgets, and other IT assets may provide a corresponding granular functionality when executed and rendered by a client device (e.g., client device 210). Thus, in some instances, the application developer may identify subsets of these IT assets for re-use in mobile apps that provide different customer and user experiences. In some aspects, to facilitate the application developer's ability to reliably and robustly re-use IT assets over time and across multiple customer experiences, IT system 150 may be configured to perform certification processes that govern a quality and resilience of apps, widgets, and other IT assets proposed for inclusion in the self-service IT asset store. Further, in other aspects, IT system 150 may perform processes that register certified IT assets (e.g., apps, APIs, widgets, etc.) for inclusion within the self-service IT store and establish an availability of these registered IT assets for consumption by application developers and other users.

[063] In some embodiments, IT system 150 may certify a candidate IT asset (e.g., a candidate app) for inclusion within the self-serve IT asset store based on a performance of the app when subjected to a predetermined set of test cases. For example, the predetermined set of test cases may be associated with input and/or configuration data that mimics an operational state of a client device during execution of the candidate app, and further, may be associated with predetermined outcomes.

Furthermore, IT system 150 may establish the predetermined test cases, and the corresponding input and/or configuration data, based on a performance of previously certified IT assets on various device platforms, and/or based on peer input identifying bugs or other instances of asset failure.

[064] In certain aspects, IT system 150 may perform processes that execute the candidate app using the input and/or configuration data associated with each of the predetermined set of test cases, and that determine whether to certify the candidate app based on a comparison of the outcomes obtained from the executed apps and corresponding ones of the predetermined outcomes. By way of example, IT system 150 may certify the candidate app for inclusion in the self-serve IT asset store when at least a threshold portion of the obtained outcomes correspond to the predetermined outcomes. IT system 150 may establish the threshold portion as 100% of the predetermined outcomes, 90% of the predetermined outcomes, or any additional or alternate portion of the predetermined outcomes sufficient to establish a quality, stability, and resilience of IT assets despite their evolution over time.

[065] In further aspects, IT system 150 may certify the candidate app for inclusion within the self-serve IT asset store based on a peer review of the performance of candidate app on one or more device and browser platforms. Additionally or alternatively, IT system 150 may certify the candidate app based on review of the quality of the uncompiled code associated with the candidate app. For instance, the certification of the candidate app for inclusion in the self-serve IT asset store may require not only robust and reliable performance across a variety of device and browser platforms, but may require that the candidate app's underlying code comply with at least

a minimum standard of quality deemed appropriate by peer review. In other aspects, IT system 150 may certify the candidate app based on the quality, scope, and/or coverage of the predetermined set of test cases. By way of example, IT system 150 may ensure predetermined test cases simulate range of expected operating conditions across a variety of device and browser platforms.

[066] In some aspects, IT system 150 may establish and provide to users (e.g., “peer” users) standardized tools and services that facilitate a consistent peer review of candidate apps, APIs, widgets, and other IT assets. For example, IT system 150 may provide, to the peer users, fully documented software development kits (e.g., SDKs), published sets of test data, standardized and consistent environments to perform peer review and testing, and/or a published and functional test “harness” that facilitates testing in the provided environments.

[067] As described above, IT system 150 may perform processes that certify IT assets for registration in the self-server IT asset store and consumption by application developers. In some aspects, the certified IT assets may include executable apps and widgets. In other aspects, however, the certified IT assets may also include APIs that facilitate a consistent exchange of data between apps executed by a client device, between containers stored on the client device, and between various devices and systems connected across network 225. In certain embodiments, an API manager of IT system 150 (e.g., a lightweight server in communication with IT system 150 or a software-based service provided by IT system 150) may be configured to perform the exemplary processes outlined above to provide robust and reliable APIs for inclusion within the self-serve IT asset store and consumption by application developers.

[068] In some embodiments, IT system 150 may provide APIs within that self-server IT asset store that facilitate a functionality of containers and corresponding container-based apps, widgets, and other IT assets (e.g., container APIs). By way of example, container APIs consistent with the disclosed embodiments may provide interfaces that facilitate and simplify interactions with between containers, between container-based apps and other apps or containers, and further, between containers or container-based apps and device-specific capabilities (e.g., an on-board camera, a global positioning system (GPS), etc.). APIs consistent with the disclosed embodiments also include channel APIs that provides interfaces enabling apps to consume capabilities and services provided by the business entity or financial institution associated with IT system 150 and/or other systems accessible to a client device (e.g., client device 210) across network 225. Further, in additional embodiments, IT system 150 may provide product and customer system APIs that enable apps, widgets, and other IT assets to access computer systems and servers associated with products, services, and customers of the business entity or financial institution associated with IT system 150, other business entities, and/or other financial institutions.

[069] The disclosed embodiments also enable a business entity associated with IT system 150 to provide information and application development tools as “services” that enable application developers to rapidly and robustly develop and deploy mobile applications and other IT assets for execution on various computing platforms (e.g., iOS devices, Android device, etc.). For instance, and as described above, IT system 150 may be configured to control access to IT assets through a number of APIs that may, for example, be configured to limit the variations in their functionalities.

Moreover, IT system 150 may, through processes and configurations consistent with disclosed embodiments, provide a controlled environment for users (e.g., internal or external to a business entity associated with IT system 150). In other aspects, IT system 150 may generate, maintain, and provide frameworks, architectures, and certifications that control consumption and client usage of mobile applications (e.g., “apps”), widgets, and other IT assets. For example, IT system 150 may provide common container apps, software development kits (SDKs), and other information and application development tools that democratize consumption of IT assets. IT system 150 may also be configured to provide a self-serve portal that provides access, through cloud-based computing, to IT assets (e.g., a self-service IT asset store). Such a portal may offer access to IT assets through a menu based (or other formatted access) interface(s).

[070] In certain aspects, frameworks and architectures consistent with the disclosed embodiments may incorporate containers that provide run-time environments for the mobile apps and other IT assets, which interact with systems of various business entities (e.g., IT system 150), devices (e.g., client devices 210 and 220), and other containers using consistent application programming interfaces (APIs). FIG. 9 is a block diagram illustrating features of an exemplary framework 900 that facilitates the development, consumption, and deployment of mobile apps and other IT assets, in accordance with disclosed embodiments. In some aspects, the elements of framework 900 may be implemented by one or more the client devices (e.g., client devices 110, 210, and 220), intermediate nodes (e.g., intermediate node 120), and IT systems (e.g.,

IT system 150) interconnected within the computing environments (e.g., environments 100 and 200) of Figs. 1 and 2.

[071] In FIG. 9, framework 900 may include one or more application containers stored on a client device (e.g., client device 210) that manage an interaction of a consumer with client device 210. For instance, an application container 902 stored on client device 210 may include instructions, scripts, and other code (e.g., HTML5, CSS and/or JavaScript) that, when executed by a processor, manage a consumer's interaction with client device 210 through an assisted channel 901A and/or an unassisted channel 901B.

[072] By way of example, assisted channels consistent with the disclosed embodiments may include, but are not limited to, a retail location associated with a business entity, a branch of a financial institution, a phone channel or call center associated with the business entity or financial institution, and/or a direct sales force associated with the business entity or the financial institution. In other instances, unassisted channels consistent with the disclosed embodiments may include, but are not limited to, a mobile communications channel, a wireless communications channel (e.g., WiFi or another wireless LAN), and channel facilitated by a user's interaction with an automated teller machine (ATM).

[073] In certain aspects, a container may operate on a particular client device (e.g., client device 210), and may be specific to client device 210, and additionally or alternatively, to applications executed by client device 210 (e.g., various web browsers or operating systems). For instance, containers consistent with the disclosed embodiments may be specific to mobile communications devices executing iOS™,

Android™, and/or a BlackBerry OS™. In other aspects, containers consistent with the disclosed embodiments may be specific to devices executing an Internet Explorer™, a Chrome™ browser, a Firefox™ browser, and/or a Safari™ browser.

[074] Furthermore, containers consistent with the disclosed embodiments (e.g., application container 902 of FIG. 9) may provide a run-time environment and a set of common services for apps, APIs, and other IT assets. For instance, a container may be associated with a container specification that establishes instructions, scripts, and other executable code that facilitate an execution of an app or other IT asset by a supported client device (e.g., client device 210) or browser executing on client device 210. In some aspects, the container specification may be established wholly or partially by a business entity or financial institution associated with IT system 150.

[075] In certain aspects, apps and other IT assets that adhere to the container specification may access a consistent set of common capabilities on the supported device. By way of example, common capabilities consistent with the disclosed embodiments may include, but are not limited to, processes that allow apps to register and login into a corresponding container, to access local cache as well as session information and attributes, and to access device specific capabilities (for example, GPS, accelerometer, and camera).

[076] Referring back to FIG. 9, container-based framework 900 also includes a consumer API 904, an integration system 906, a provider API 908, provider integration processes 910, provider systems 912 (e.g., associated with a financial institution and providing information associated with customers and corresponding deposits, lending, and payment), and app store 914. In some aspects, client device 210 may store and

execute consumer API 904 to receive, process, and format for integration requests for information and/or services received from one or more executed apps. For instance, consumer API 904 may include programming interfaces that support message formulation (e.g., by providing access to JSON or XML interfaces), that support message composure and generation (e.g., REST and/or SOAP messages), and that send and/or receive messages (e.g., an HTTP API). In other aspects, browser-compatible code and scripts (e.g., Javascript) may establish the consumer API, which client device 210 may implement in conjunction with a web browser (e.g., as a plug-in, etc.).

[077] In certain aspects, integration system 906 may receive requests for information and/or services from client device 210 through consumer API 904, and may be configured to perform session management, authentication, and messaging functions associated with the received requests and/or destinations of the received requests. For instance, integration system 906 may correspond to an intermediate node (e.g., intermediate node 120) that connects client device 210 to IT system 150. Further, in some aspects, integration system 906 may perform load-balancing operations that route the received requests to appropriate computer systems and servers associated with a destination. Integration system 906 may also be configured to perform processes that address cross-cutting concerns, and further, that perform messaging and queuing services that guarantee the received request are delivered to the appropriate destination computer systems or servers. In additional aspects, integration system 906 may modify the received requests to conform to one or more

canonical data models (e.g., associated with common data requests for customer or transaction information).

[078] Integration system 906 may, for example, route the received requests to a corresponding destination system or server (e.g., one or provider systems 912) through corresponding provider API 908. In some aspects, provider API 908 may receive requests for data or services from integration system 906, and may simplify and normalize access to provider systems 912, which may service the requests and provide the requested data or services to client device 210. As described above, provider API 908 may include programming interfaces that support message formulation (e.g., by providing access to JSON or XML interfaces), that support message composure and generation (e.g., REST and/or SOAP messages), and that send and/or receive messages (e.g., an HTTP API). In further aspects, a lightweight application server (not depicted in FIG. 9) associated with the corresponding one of provider systems 912 may execute instructions and scripts (e.g., Java code) to establish provider API 908 at the corresponding one of the provider systems 912.

[079] In additional embodiments, the lightweight application server and/or the corresponding one of the provider systems 912 may execute instructions to perform provider integration processes 910 in response to the request processed by provider API 908. In an embodiment, provider integration processes 910 may access a portion of the requested data stored by the corresponding one of provider systems 912 or legacy systems associated with provider systems 912, and may return the accessed data to client device 210 as a response to the request. For instance, upon obtaining the requested data portions from provider systems 912 and/or the associated legacy

systems, the lightweight application server and/or the corresponding one of the provider systems 912 may be configured to transmit the requested data portion across network 225 to client device 210 via provider API 908 and integration systems 906. Consumer API 904 may receive the transmitted data portion, and process and/or format the received data portion for presentation to user 211 and/or user by one or more additional apps executed by client device 210.

[080] FIG. 10A illustrates an exemplary process 1000 for generating, processing, and routing requests for data or services within container-based framework 900, in accordance with disclosed embodiments. For example, a customer (e.g., user 211) may access a web page associated with a financial institution via a client device (e.g., client device 210), and may navigate to a corresponding portion of the web page to request a current account balance. In some instances, a mobile app compatible with client device 210 and an executed web browser (e.g., an app associated with container and device 902) may issue a request to obtain account balance information for the customer from a system associated with the financial institution (e.g., in step 1002).

[081] In step 1004, a consumer API executed by client device 210 (e.g., consumer API 904) may process and route the issued request across network 225 to an appropriate provider system capable of servicing the request (e.g., to a corresponding one of provider systems 912 or to a server associated with the corresponding one of provider systems 912, which perform provider integration services 910). In some instances, consumer API 904 may route the issued request to an integration system (e.g., integration system 906) for customer authentication and other messaging and session management services.

[082] Upon receipt of the issued request by the corresponding provider system 912 (or the associated server), corresponding provider system 912 (or the associated server) may perform provider integration services 910 and execute the request to identify and obtain at least a portion of the information associated with the issued request (e.g., in step 1006). In some instances, the obtained data portion may be obtained from a data repository associated with the corresponding provider system 912, and additionally or alternatively, from a legacy system, server, or data repository associated with corresponding provider system 912. The disclosed embodiments may be further configured to provide the requested information (e.g., the account balance information) to client device 210 for presentation to user 211, and additionally or alternatively, for use by another executed app (e.g., in step 1008). For instance, the provider integration services performed by corresponding provider system 912 or the associated server may transmit the requested information to consumer API 902 of client device 210, or alternatively, transmit the requested information to integration system 906, which may perform one or more session management or messaging services prior to transmitting the requested information to consumer API 902. Exemplary method 1000 is then complete.

[083] FIG. 10B illustrates an exemplary process 1020 for generating, processing, and routing requests for data or services within container-based framework 900, in accordance with disclosed embodiments. For example, an employee of a financial institution may access web page interface associated the financial institution through a tablet computer (e.g., client device 210), and in response to a conversation with a customer, submit a balance inquiry on behalf of the customer. In some

instances, a mobile app compatible with client device 210 and an executed web browser (e.g., an app associated with container 902 and device 210) may issue a request to obtain account balance information on behalf of the customer from a system associated with the financial institution (e.g., in step 1022).

[084] In step 1024, a consumer API executed by client device 210 (e.g., consumer API 904) may process and route the issued request across network 225 to integration system 906. Integration system 906 may, in some instances, determine whether the a system associated with the financial institution (e.g., a corresponding one of provider systems 912) or IT system 150 permits the employee to access the customer's account information and obtain a current account balance on behalf of the customer (e.g., in step 1026). In some aspects, integration system 906 may determine whether the employee is permitted to access the customer's account data based on profile data associated with the customer and/or the employee, which may identify one or more customer-specified access permissions. In other aspects, integration system 906 may determine whether the employee is permitted to access the customer's account data based on default permissions associated with the customer or the employee.

[085] If integration system 906 were to determine that the employee lacks permission to access the customer's account data (e.g., step 1028; NO), then exemplary method 1020 is complete. In some aspects, integration system 906 may provide an error message to consumer API 904 for presentation on client device 210.

[086] If, however, integration system were to determine that the employee is permitted to access the customer's account data (e.g., step 1028; YES), integration

system 906 may route the issued request to appropriate provider system capable of servicing the request in step 1030. For instance, the appropriate provider system may include a corresponding one of provider systems 912 or to a server associated with the corresponding one of provider systems 912, which performs provider integration services 910.

[087] Upon receipt of the issued request by the corresponding provider system 912 (or the associated server), corresponding provider system 912 (or the associated server) may perform provider integration services 910 and execute the request to identify and obtain at least a portion of the information associated with the issued request (e.g., in step 1030). In some instances, the obtained data portion may be obtained from a data repository associated with the corresponding provider system 912, and additionally or alternatively, from a legacy system, server, or data repository associated with corresponding provider system 912. The disclosed embodiments may be further configured to provide the requested information (e.g., the account balance information) to client device 210 for presentation to user 211, and additionally or alternatively, for use by another executed app (e.g., in step 1032). For instance, the provider integration services performed by corresponding provider system 912 or the associated server may transmit the requested information to consumer API 902 of client device 210, or alternatively, transmit the requested information to integration system 906, which may perform one or more session management or messaging services prior to transmitting the requested information to consumer API 902. Exemplary method 1020 is then complete.

[088] Referring back to FIG. 9, container-based framework 900 also includes app store 914 that, in some aspects, may correspond to a self-serve IT asset store provided through a web page or other digital portal associated with IT system 150. As described above, an application developer (e.g., user 211) may access the self-serve IT asset store (e.g., using a web browser or other graphical user interface on client device 210) to obtain apps, APIs, widgets, and other IT assets to develop and/or test apps for execution across various device platforms (e.g., iOS™, Android™, BlackBerry OS™, etc.) and/or browser platforms (e.g., Internet Explorer™, Safari™, Chrome™, etc.). In additional aspects, IT system 150 may also be configured to perform processes that register certified IT assets (e.g., apps, APIs, widgets, and other IT assets) for inclusion within the self-service IT store and establish an availability of these registered IT assets for consumption by application developers and other users. FIG. 10C illustrates an exemplary process 1040 for certifying and registering IT assets for consumption through a self-serve IT asset store, in accordance with disclosed embodiments.

[089] In FIG. 10C, IT system 150 may receive information identifying a candidate IT asset (e.g., an app, API, widget, etc.) for registration in the self-server IT asset store and subsequent consumption by application developers (e.g., in step 1042). In some aspects, the candidate IT asset may be submitted by a user (e.g., user 221 via client device 220) through a digital portal or web page provided by IT system 150 for submission of candidate IT assets. In other instances, the information identifying the candidate IT asset may be received from system associated with an external app store (e.g., iTunes™, Google Play™, etc.) through a corresponding API.

[090] IT system 150 may perform and/or apply one or more certification processes to the candidate IT asset to determine whether the candidate IT asset is suitable for inclusion in the self-serve IT asset store, and further, adheres to a quality, reliability, and resilience of other IT assets provided by the self-serve IT asset store (e.g., in step 1044). For example, as described above, IT system 150 may certify a candidate IT asset (e.g., a candidate app or API) for inclusion within the self-serve IT asset store based on a performance of the candidate IT asset when subjected to a predetermined set of test cases. In other aspects, IT system 150 may certify the candidate IT asset for inclusion within the self-serve IT asset store based on a peer review of the performance of candidate IT asset on one or more device and browser platforms, and additionally or alternatively, based on review of the quality and robustness of the uncompiled code associated with the candidate app. Furthermore, IT system 150 may certify the candidate IT asset based on an analysis not only on the performance of the candidate app using the predetermined set of test cases, but also on the quality, scope, and/or coverage of the predetermined set of test cases.

[091] IT system 150 may determine whether the candidate IT asset meets the requirements for quality, reliability, and resilience established by IT system 150, and thus, is certified for inclusion within the self-serve IT asset store (e.g., in step 1046). If IT system 150 were to determine the candidate IT app is not certified (e.g., step 1046; NO), IT system 150 may determine that the candidate IT asset should not be included in the self-serve IT asset store, and process 1040 may pass back to step 1042. In some aspects, IT system 150 may await submission of an additional candidate IT asset.

[092] If, however, IT system 150 certifies the candidate IT asset (e.g., step 1046; YES), IT system 150 may register the candidate IT asset in the self-serve IT asset store and maintain information identifying the IT asset (e.g., an asset name, version, specification, and/or manifest information) within a corresponding repository (e.g., in step 1048). In certain aspects, the repository may be searchable by name, specification information version, and any additional or alternate information identifying the registered IT asset and capable of being indexed by IT system 150. IT system 150 may be configured to deploy the registered IT asset within the self-serve IT asset store for consumption by application developers (e.g., in step 1050).

[093] In some aspects, IT system 150 may grant an application developer access to the registered IT asset, and additionally or alternatively, one or more apps, widgets, APIs, or other IT assets registered for consumption in the self-serve IT asset store (e.g., in step 1052). For instance, the application developer (e.g., user 211 of client device 210) may access the self-serve IT asset store through a corresponding web page or digital portal, and submit a request to obtain one or more available IT assets. In certain aspects, IT system 150 may receive the request, determine whether the application developer is associated with permissions appropriate to access the requested IT assets, and if IT system 150 determines the user is associated with appropriate permissions, facilitate the application developer's access to the requested IT assets. For instance, IT system 150 may deliver the requested IT assets to client device 210 using any of the communications protocols outlined above, and additionally or alternatively, may establish a temporal window during which user 211 may access and download the requested IT assets to client device 210.

[094] IT system 150 may, in an embodiment, monitor a digital activity of user 211 to determine not only when user 211 downloads the requested IT assets, but also to determine when and under what conditions client device 210 executes and renders the requested IT assets, either alone or assembled into another app (e.g., step 1054). For instance, in step 1054, IT system 150 may monitor user 211's digital activities to generate performance metrics, usage metrics, and other performance- and/or usage-related data suitable for incorporation in reports, invoices, etc.

[095] Further, in some instances, IT system 150 may generate reports and/or invoices based on the obtained performance and usage metrics and data (e.g., in step 1056). For instance, IT system 150 may generate a report that identifies a number of IT assets (e.g., apps, APIs, widgets, etc.) consumed by user 211 during a predetermined time period (e.g., monthly, daily, weekly, etc.). Further, in some aspects, IT system 150 may charge user 211 a predetermined fee for each consumed IT asset and/or for each IT asset executed by client device 210. Further, in other aspects, IT system 150 may assess user 211 a monthly, daily, or hourly fee for accessing the self-serve IT asset store. In some embodiments, IT system 150 may provide the generated reports and/or invoices to client device 210 across network 225 using any of the communications protocols outlined above. Exemplary process 1040 is then complete.

[096] In the embodiments described above, container-based frameworks and architectures may incorporate containers that provide run-time environments for the mobile apps and other IT assets, which interact with systems of various business entities (e.g., a business entity or financial institution associated with IT system 150), devices (e.g., client devices 210 and 220), and other containers using consistent

application programming interfaces (APIs). FIG. 11 is a block diagram illustrating components of an exemplary application container 1100, in accordance with disclosed embodiments. As described above, application container 1100 may include instructions, scripts, and other executable code that, upon execution by client device 210, manage an interaction between client device 210 and one or more consumers (e.g., through an assisted or unassisted channel).

[097] In some aspects, application container 1100 may be specific to a particular device operating system and/or browser, and in some aspects may include a device-specific shell and a browser shell. The device-specific shell may, for example, include instructions, scripts, and other executable code that facilitates an execution of one or more apps (e.g., apps 1110) on a particular device (e.g., communications devices executing an iOS™, Android™, and/or BlackBerry OS™). Further in some instances, the browser shell may include instructions, scripts, and other executable code that facilitate an execution of apps 1110 in conjunction with a corresponding web browser (e.g., an Internet Explorer™, Chrome™, Firefox™, and/or Safari™ browser).

[098] In some aspects, application container 1100 may also include a model-view-controller (MVC) framework and event manager 1102. MVC framework and event manager 1102 may, for example, be executed by client device 210 to generate an “event bus” that manages interactions between applications (e.g., apps 1110), APIs, widgets, and other IT assets implemented by client device 210. By way of example, MVC framework and event manager 1102 may facilitate communications between portions of the apps, APIs, widgets, and other IT assets present on and executed by

client device 210, and further, may coordinate the loading and unloading of corresponding apps, widgets, and IT assets for execution by client device 210.

[0099] Application container 1100 may further include a presentation layer 1104 that establishes processes than enable client device 210 to render output for presentation to user 220. For example, the established processes may include responsive web design (RWD) techniques provide an optimal user experience across a wide range of device types and form factors. Further, in some instances, presentation layer 1006 may establish techniques that render HTML content for presentation to the user using HTML5 and CSS standards.

[0100] In further aspects (not depicted in FIG. 11), application container 1100 may also include a user context layer that establishes user-specific theming and/or presentation details for the rendered content, user-specific configuration preferences for content presentation, common user-specific menus or interface elements, and/or authentication credentials or tokens for the user. For instances, application container 1100 and/or apps 1110 may rely on security processes established by a business entity or financial institution associated with IT system 150, which may require an authentication of user 220 prior to executing a particular app, widget, or other IT asset.

[0101] In some embodiments, application container 1100 may also include a device API 1106 and a context API 1108. Device API 1106 may, for instance, enable an executing app, widget, or other IT asset to access one or more capabilities of client device 210, which include, but are not limited to, on-board positioning systems (e.g., GPS), accelerometers, an on-board camera. In other instances, content API 1008 may enable an executing app, widget, or other IT asset to access transient data residing

locally within application container 1100 on client device 220. Transient data consistent with the disclosed embodiments may, for example, include data generated by one or more currently or previously executed apps, widgets, or other IT assets.

[0102] Further, in certain embodiments, application container 1100 may include one or more apps, widgets, or IT assets (e.g., apps 1110) that provide a desired user or customer experience when executed by client device 210 and rendered for presentation on client device 210 or a browser executed by client device 210. Further, although not depicted in FIG. 11, application container 1100 may also include one or more container APIs that facilitate interrogation and integration between multiple application containers, apps, widgets, or other IT assets associated with these containers, and device-specific capabilities (e.g., an on-board positioning system, an accelerometer, or an on-board camera).

[0103] As described above, an individual app may provide a granular level of user interface functionality, and may be executed and rendered by client device 210 to provide legendary user and/or customer experience across any device, accessed at anytime and anywhere. In other aspects, an application developer may assembler together a plurality of individual application that, upon execution and rendering, may provide a “macro” set of user interface functionalities. In some embodiments, the assemblage of individual apps, widgets, and/or IT assets into a “macro” app may provide a composite user interface functionality (e.g., resulting from the combination of the individual user interface functionalities) useful to customers and employees of the business entity associated with IT system 150.

[0104] In certain aspects, apps 1110 may be configured to operate in a resilient fashion independent of other apps, widgets, and/or ITS assets. To achieve this capability, an application developer may design or assemble an app in an “end-to-end” manner than addresses not only a front-end user experience, but also access and consumption of data associated with IT system 150 and computing systems associated with other business entities and financial institutions accessible across network 225.

[0105] In some embodiments, each of apps 1110 may be associated with a manifest that establishes a basic set of common identifying and configuration data, as well as application-specific metadata rendered by client device 210 for presentation to user 220 and/or available within a corresponding IT asset store. Further, in additional aspects, the container-based framework implemented by IT system 150 may require that each of apps 1110 establish a corresponding manifest prior to certification and inclusion within the self-serve IT asset store, as described above. Further, in some aspects, a manifest established for an assemblage of individual apps may provide a “contract” between the individual apps that defines parameters that facilitate the execution of the individual apps in corresponding containers and enable inter-application communication between the individual apps.

[0106] The manifest for a corresponding one of apps 1110 (and additionally or alternatively, for an assemblage of apps) may include identifying and configuration data associated with the app or apps. Manifest data consistent with the disclosed embodiments may include, but is not limited to: (i) a human-readable text string describing the corresponding app or apps; (ii) a version of the corresponding app or apps; (iii) a description of dependencies linking the corresponding app or apps to other

systems, apps, widgets, and IT assets; (iv) a description of events emitted by the corresponding app or apps; (v) a description of events consumed or processed by the corresponding app or apps; or (vi) a URL or other information that provides access to content of the corresponding app or apps.

[0107] In some aspects, IT system 150 may publish and make available to application developers the contracts established by application manifests (e.g., within the IT asset store). For example, apps 1110 may interact with other apps, widgets, and/or IT assets to provide a holistic and seamless user experience in accordance with the published contracts. Further, in other aspects, the published contracts may establish one or more communications parameters by which apps 1110 may communicate with client device 220 or a browser executed by client device 220 (e.g., through application container 1000). For instance, the published contracts may establish a “thin” façade that abstracts lower-level device capability such that changes in device capabilities are isolated from corresponding apps 1110.

[0108] In certain embodiments, apps 1110 may include one or more business apps that provide a business domain functionality associated with the business entity or financial institution associated with IT system 150. Business apps consistent with the disclosed embodiments may include, but are not limited to: customer apps that provide an ability to search, view, and update customer information; product apps that provide an ability to search, view, and update products; account apps that provide an ability to search, view, and update accounts; transaction apps that provide an ability to search and view transactions; and payment apps that provide an ability to select payment types and execute payments. Further, in some instances, the business apps may include

apps specific to a line-of-business (LOB) or business group of the business entity or financial institution, including but not limited to credit card apps, lending product apps, deposit product apps, insurance apps, collections apps, sales apps, and fraud or compliance apps.

[0109] In other embodiments, apps 1110 may include one or more container apps that provide a common a common set of capabilities for various communications devices and run-time environments (e.g., mobile communications devices executing iOS, Android, or BlackBerry operating systems). For example, container apps consistent with the disclosed embodiments may include, but are not limited to login containers apps, local cache apps, session management apps, navigation and banding apps, and utility apps.

[0110] For example, a local cache app may, upon execution and rendering, provide a method of managing and interrogating application- and/or container-specific information stored on the client device 210 and/or in a local cache associated with a browser executed by client device 210. Further, for example, a session management app may provide a method of managing and interrogating application- and/or container-specific information stored by systems or data repositories accessible to client device 210 across network 225. A branding app may, for example, provide a method of consistently rendering user interface elements (e.g., logos and marketing material) related to a brand of the business entity or financial institution associated with IT system 150, and a navigation app may provide processes that consistently renders user interface elements related to navigating within apps (e.g., navigation panes and/or user menu). In certain aspects, the navigation app may be dynamically generated and may

reflect one or more capabilities made available by the currently executed apps. In further aspects, a utility app, when executed and rendered by client device 210, may provide base capabilities common to all executing apps, including, but not limited to, date and other localization, logging/debugging, and notification capabilities.

[0111] In some embodiments, a login container app may, upon execution by client device 210, receive from a user (e.g., an employee or customer of a business entity associated with IT system 150) credentials that confirm that the user is known to the business entity and can use apps 101. Upon successful authentication (e.g., by client device 210, by IT system 150, or by an associated authentication system), IT system 150 may allow the employee or customer to access one or more of apps 1110. In further response to a successful authentication, IT system 150 may be configured to provide, to application container 1100, a profile associated with the employee or the customer that includes employee- or customer-specific preferences, customizations, themes, and/or navigation preferences.

[0112] Further, in some aspects, IT system 150 may confer no further authority upon the employee or customer in response to a successful authentication, and any additional requests for execute capability of the business entity may require authentication in accordance with policies established by the business entity. In other aspects, however, the container login app may represent a “single-sign-on” app than allows the employee and/or customer to switch freely between apps 1110 in response to a successful authentication. In additional aspects, the login container app may represent an “assisted-login” app that enables an employee of the business entity or financial institution to provide authentication credential of a customer, and that, in

response to a successful authentication, requests and maintains role and entitlement information for that customer. The role and entitlement information may, for instance, enhance the customer experience and simplify the ability to execute requests on behalf of a customer.

[0113] In an embodiment, and in response to a successful login by a user (e.g., a customer or an employee of a financial institution), a corresponding client device (e.g., client device 210 or 220) may be configured to obtain information identifying a container (e.g., application container 1100), receive and process a manifest containing a user profile, render base branding and navigation apps, and further, render one or more start-up apps associated with the user. FIG. 12 illustrates an exemplary process 1200 performed by a client device upon successful authentication of a user.

[0114] For instance, a user (e.g., user 211) may, via client device 210, initiate a user login container app (e.g., a ULCA) the renders display fields enabling user 211 to enter authentication credentials into client device 210 (e.g., in step 1202). In one aspects, the ULCA may be locally stored by client device 210, and additionally or alternatively, may be accessed remotely by entering a corresponding uniform resource identifier (URI) into client device 210. Further, the authentication credentials may include, but are not limited to a user name, a password, a personal identification number (PIN), and any additional or alternate credential identifying user 211 and appropriate to a business entity or financial institution associated with IT system 150.

[0115] In some instances, user 211 may provide the authentication credentials as input into the rendered fields (e.g., in step 1204), and initiated ULCA may invoke a corresponding consumer API (e.g., a User Login Container API) to generate and

transmit an authentication request across network 225 to IT system 150 or a system or server associated with an additional business entity or financial institution (e.g., in step 1206). In some aspects, the client device 210 execute instructions and/or scripts to implement the User Login Container API, which may package the inputted credentials to form the authentication request. Client device 210 may, for example, perform processes that transmit the generated request to an authorization and authentication system associated with the additional business entity, the additional financial institution, and/or IT system 150 (e.g., integration system 906 of FIG. 9).

[0116] In some embodiments, the authorization and authentication system may process the received authentication request, extract the credentials, and validate the credentials against user profile data and/or additional authentication information associated with user 211. The authorization and authentication system may, for example, transmit a response indicative of a successful or unsuccessful authentication to client device 210 through the User Login Container API.

[0117] Client device 210 may receive the response, and after processing by the User Login Container API, determine whether the authorization and authentication system successfully validated the user 211's credentials (e.g., in step 1208). If client device 210 determines that user 211's credentials are invalid (e.g., step 1208; NO), client device 210 may further execute the ULCA to present information identifying additional authentication options available to user 211 (e.g., in step 1210). For instance, client device 210 may present, to user 211, a first dialog indicating that the credentials are invalid and requesting re-entry of the appropriate credentials, a second dialog box prompting user 211 to enter an e-mail address at which user 211 will receive

a new password or information identifying a user name, and/or a third dialog box indicating that user 211 should attempt to login again after a short delay (e.g., in five seconds). In other aspects, client device 210 may disable the ULCA after a subsequent (e.g., a third) unsuccessful login attempt. Exemplary method 1200 is then complete.

[0118] If, however, client device 210 were to determine that the authorization and authentication system successfully validated the user 211's credentials (e.g., step 1208; YES), then client device 210 may retrieve and verify a manifest associated with the ULCA (e.g., in step 1212). By way of example, client device 210 may obtain the manifest from IT system 150. Furthermore, the obtained manifest may include, but is not limited to, user profile data, information identifying branding and navigation apps associated with the ULCA, user preferences, look-and-feel themes, and information identifying one or more start-up apps associated with user 210. In some aspects, client device 210 may execute and render the navigation, branding, and start-up apps to provide a legendary customer experience to user 211.

[0119] In certain aspects, client device 210 may further execute the ULCA to render the branding and navigation apps included within the manifest (e.g., in step 1214). Further, in step 1214, client device 210 may further execute the ULCA to apply themes found in the manifest, and apply any other user preferences and/or container configurations found in the manifest. Client device 210 may further execute the ULCA to initialize and render the one or more startup apps found in the manifest to provide the legendary customer experience (e.g., in step 1216). For example, by rendering the startup apps, client device 210 may present to user 211 information identifying a

financial services account of user 211 (e.g., an account balance, recent transactions, scheduled funds transfers, etc.). Exemplary method 1200 is then complete.

[0120] In the exemplary embodiments described above, a system associated with a business entity (e.g., IT system 150) may provide IT assets as “services” that enable an application developer to rapidly and robustly develop and deploy instances of applications for execution on various computing platforms (e.g., iOS devices, Android device, etc.). For example, as described above, the application developer may access a web page or graphical user interface (GUI) that identifies and presents one or more available services (e.g., the exemplary interface of the self-service IT asset store of FIG. 8). The “services” available to the application developer and presented by the web page or GUI may include, but are not limited to, infrastructure-based assets (e.g., available operating systems, processors, memory, cloud storage, etc.), platform-based assets (e.g., containers available for Android™ devices, iOS™ devices, etc.), data and database services (e.g., data sets available for quality control, testing, visualization, etc.), API-based services (e.g., available APIs that standardize access to core engines), application-based services (e.g., available business and/or container “apps” and portions of “apps”), reporting and analytics-based services, and services that permissioning and compliance with various regulations and policies (e.g., PCI policies, SOX policies, compliance with audit, logging, and reporting policies, compliance with data access policies and permissions, etc.).

[0121] In some aspects, the application developer may select a subset of the available assets, and as described above, IT system 150 may be configured to generate and present to the application developer (e.g., through the web page or GUI) an IT

asset canvas that represents the selected services and their relationship to other IT assets (e.g., required input data, etc.). Further, IT system 150 may also be configured to provide access to, or to provide, the IT assets associated with the selected services (e.g., executable, container-based applications linked to corresponding cloud-based data storage) to a device associated with the application developer (e.g., to client device 210 across network 225 of FIG. 2). For example, IT system 150 may charge the developer a predetermined fee to obtain the IT assets, and upon initiation and successful completion of a purchase transaction (e.g., through a point-of-sale (POS) system in communication with or provided by IT system 150), IT system 150 may provide the application developer with links to the IT assets, and additionally or alternatively, may transmit copies of the IT assets to the application developer's device.

[0122] In certain embodiments, the IT assets selected by the application developer and provided by IT system 150 may be configured as core building blocks for larger applications that, when executed by corresponding mobile devices, servers, and/or computer systems, facilitate the provision of services by various business entities, such as financial institutions. For example, the selected IT assets may include executable applications (e.g., mobile apps, widgets, etc.) that provide granular computational and data access capabilities. The application developer may, in some instances, assemble and link the selected IT assets to appropriate data inputs and outputs (e.g., through corresponding APIs) to generate or "spin-up" an instance of an application that provides one or more services to customers and/or employees of the business entities.

[0123] Further, in some embodiments, the application developer may spin-up multiple instances of a single application for purposes of initial testing and debugging, for purposes of development, for purposes of quality control, and/or to facilitate visualization using one or more virtual machines provided by IT system 150 (e.g., through the web page and/or GUI). In certain aspects, the application developer may link the generated instances to different types of input data, and further, the generated instances may be associated with various capabilities to modify accessed data. For example, the application developer may link an instance generated for purposes of initial testing to “exemplary” input data scrubbed of any business or customer details. Additionally or alternatively, the application developer may link an instance generated for purposes of visualization and/or quality control to actual customer and business data to provide an accurate representation of the operation and functionality of the application.

[0124] Further, in other aspects, the data linked to the various generated instances may be subject to different permissioning and compliance policies. For example, as the exemplary input data linked to the instance generated for initial testing may not include actual customer or business data, the exemplary input data may be accessible to individuals related to the business entity or the application developer with little or no restriction. In additional aspects, the actual customer and/or business data linked to the visualization or quality control may be subject to strict permissioning policies that limit access to specific individuals identified by the business entity. In some instances, the strict permissioning policies may curtail an ability of the application developer and/or the business unit to delegate visualization or quality control to third-

parties. Furthermore, the actual customer and/or business data may be subject to strict auditing and compliance policies that the business entity may not apply to the exemplary input data linked to the instance generated for initial testing.

[0125] Thus, in certain aspects, the application developer must identify available IT assets that, when assembled and linked to appropriate data sources, provide the one or more services desired by the business entities, and further, are associated with data permissioning and compliance policies that are mutually compatible with each other and with the individuals or entities charged with executing the application. The need to identify and mediate conflicts between the data permissioning and compliance policies may challenge application developers and may slow the development of complex enterprises requiring generation of instances of multiple executable applications for initial testing, development, visualization, and quality-control purposes (e.g., new mobile enterprises requiring executable applications for multiple mobile and server-based platforms).

[0126] In an embodiment, IT system 150 may be configured to execute processes that, in addition to providing one or more IT assets as “services” to application developers, also provide enterprise-building services that generate or “spin-up” instances of executable applications, that link the generated instances to sources of data having appropriate permissioning and compliance policies, and that package the instantiated applications and linked data sources into modules for delivery to an application developer. By way of example, as described above, IT system 150 may be configured to receive the application developer’s selection of available IT assets (e.g., using the self-serve IT store accessed through a web page or other GUI). In certain

aspects, IT system 150 may be configured to execute processes that provide an “orchestration engine” capable of parsing the selected IT assets, and further, determining whether the selected IT assets include infrastructure-based, platform-based, and software-based assets that are compatible and sufficient to generate or “spin-up” an instance of a corresponding application. The selected IT assets may include, but are not limited to, hardware assets (e.g., processors, mobile platforms, memory, etc.), cloud-based storage assets, operating systems, middleware, and/or granular applications supported by the operating systems and middleware and executable by the hardware assets.

[0127] Based on the determination, and without additional input from the application developer, the orchestration engine may identify and add to the selected resources one or more additional resources that may be necessary to successfully instantiate the application. In additional aspects, the orchestration engine may automatically generate an instance of the application based on the selected IT assets (and additionally or alternatively, based on compatible IT assets identified by the orchestration engine), and may link one or more input and/or output data sources (e.g., as identified by the orchestration engine or within the selected IT assets) to the generated instances.

[0128] In certain aspects, the selected IT assets may include one or more data permissioning and compliance services. The orchestration engine may, in some embodiments, confirm a compatibility of the selected data sources with the selected data permissioning and compliance services, and if one or more of the selected data sources are deemed incompatible, identify at least one additional source of input and/or

output data that is available to the application developer (or an entity identified by the application developer and/or business entity), and that consistent with data requirements (e.g., formatting, size, etc.) of the generated instances, and further, that is compatible with the selected data permissioning and compliance services.

[0129] The orchestration engine may be configured to package the generated instance and information identifying the linked data into a module having a format suitable for delivery to a device associated with the application developer (e.g., client device 210 of FIG. 2). For example, the orchestration engine may package the generated instance as Javascript or other HTML code, with information identifying the linked being provided as metadata. In other instances, the generated instances and linking information may be compiled together and provided to the device as an executable file.

[0130] Further, in additional aspects, the orchestration engine may also be configured to provide, to the user within the module, additional data suitable for presentation to the application developer, a customer, or other entity during execution of the generated instance by a corresponding device. For example, generated instance may, upon execution by a device, present information identifying a current account balance of a customer through a corresponding web page or graphical user interface. In some instances, the additional data packaged into the module may include graphical content (or links to graphical content) for concurrent display with the current account balance.

[0131] Orchestration engines consistent with the disclosed embodiments may provide enterprise-building services that generate or “spin-up” instances of executable

applications, that link the generated instances to sources of data having appropriate permissioning and compliance policies, and that package the instantiated applications and linked data sources into a module for delivery to an application developer. In some aspects, the enterprise-building services provided by the orchestration engine may, automatically generate instances of applications using multiple layered services that include, but are not limited to, infrastructure-based services incorporating hardware and cloud-stack assets, platform-based services that support various operating systems and middleware, and software-based services that support executable applications, as described below in reference to FIG. 13.

[0132] FIG. 13 is a diagram illustrating exemplary services 1300 provided by IT system 150, consistent with disclosed embodiments. For example, and as described above, an orchestration engine (e.g., as provided by IT system 150) may provide enterprise-building services that generate or “spin-up” instances of executable applications, which may incorporate multiple infrastructure-based, platform-based, and software-based services provided by IT system 150 and available to an application developer (e.g., through an interface of a self-serve IT asset store).

[0133] In FIG. 13, services 1300 may include hardware assets 1302 and cloud stack assets 1304. By way of example, hardware-based assets 1302 consistent with the disclosed embodiments may include, but are not limited to, processors, servers, systems, data centers, locally-accessible storage units, and/or networking resources. Further, in some instances, cloud-stack assets 1304 may include, but are not limited to, cloud-based storage that include customer data, business data, and any additional or alternate data associated with IT system 150 and/or a business unit associated with IT

system 150. Further, and as described above, at least a portion of the data stored within the cloud-based storage may be subject to one or more data permissions and compliance policies.

[0134] Services 1300 may also include one or more virtual machines 1306. In certain aspects, virtual machines 1306, when implemented using hardware assets 1302 and cloud stack assets 1304, may provide virtualization processes for instances of applications “spun-up” by the orchestration engine. As outlined in FIG. 13, virtual machines 1306 may include, but are not limited to, operating systems 1306A, middleware 1306B, and applications 1306C. In one aspect, operating systems 1306A may include operating systems supported by one or more devices capable of executing the generated instances, such as Android™, iOS™, Windows™, Linux™, and Unix™. Further, in some aspects, applications 1306C may include any application executable by the one or more devices and supported by operating systems 1306A, e.g., web browsers, text editors, Javascripts, etc. In additional aspects, middleware 1306B may include any elements of executable code (e.g., APIs) that “glue together” or mediate between applications 1306C and operating systems 1306A, and further, between applications 1306C and data stored within cloud stack assets 1304 and/or storage devices of hardware assets 1302.

[0135] Services 1300 may also include infrastructure-based services 1312 provided by IT system 150 and available to the application developer. In certain aspects, the available infrastructure-based services 1312 may include one or more hardware-based assets 1302 and cloud-stack assets described above.

[0136] In addition, services 1300 may include platform-based services 1322 provided by IT system 150 and available to the application developer. Platform-based services 1322 may, for example, incorporate and build upon infrastructure-based services 1312, which incorporate hardware assets 1302 and cloud-stack assets 1304 as described above. Further, in some instances, platform-based services 1322 may incorporate one or more of operating systems 1306A and middleware 1306B, as outlined above. In certain aspects, platform-based services 1322 consistent with the disclosed embodiments may include platforms specified by a particular combination of selected hardware assets (e.g., mobile devices, etc.), selected cloud stack assets, a selected operating system compatible with and appropriate to the selected hardware and cloud stack assets, and selected middleware capable of interfacing with the selected operating system and cloud stack assets.

[0137] Further, in some embodiments, services 1300 may include software-based services 1332 provided by IT system 150 and available to the application developer. Software-based services 1332 may, in certain aspects, represent combinations of applications 1306C (e.g., of virtual machine assets 1306) and corresponding ones of platform-based services 1322 (e.g., as specified by combinations of selected hardware assets, selected cloud stack assets, and corresponding selections of operation systems and middleware, as described above).

[0138] In some aspects, software-based services 1332 consistent with the disclosed embodiments may include unique combinations of an application (e.g., one of applications 1306C) and a platform capable of executing the application (e.g., one of platform-based services 1322). Further, as noted above, and in response to the

selections of the application developer (e.g., using the interface associated with the self-serve IT store), IT system 150 may generate multiple software-based services. In certain aspects, the orchestration engine (e.g., as provided by IT system 150) may assemble combinations of the software-based services to generate instances of executable applications, link the generated instances to sources of data having appropriate permissioning and compliance policies, and that package the instantiated applications and linked data sources into modules for delivery to an application developer.

[0139] FIG. 14 is a diagram illustrating exemplary services 1400 provided by IT system 150, in accordance with disclosed embodiments. As described above, an orchestration engine (e.g., as provided by IT system 150) may provide enterprise-building services that generate or “spin-up” instances of executable applications that incorporate multiple layered services provided by IT system 150 and available to an application developer (e.g., through an interface of a self-serve IT asset store). Further, in certain aspects, the orchestration engine may be configured to assemble mutually compatible combination of the layered services to generate instances of executable applications, link the generated instances to sources of data having appropriate permissioning and compliance policies, and package the instantiated applications and linked data sources into modules for delivery to an application developer.

[0140] For example, IT server 142 may provide one or more infrastructure-based services 1402 and platform-based services to an application developer (e.g., through a web page or GUI associated with a self-server IT asset store). As described above, infrastructure-based services 1402 may include, but are not limited to, hardware assets,

cloud stack asserts, network resources, and one or more virtual machines (and corresponding operating systems, middleware, and applications) available to the application developer and provided by IT system 150. Further, and as noted above, platform-based services 1404 may couple and mediate between the virtual machines and one or more of the hardware assets, cloud stack assets, and network resources of infrastructure-based services 1402. Services 1400 also include, but are not limited to, database services 1406, data services 1408, reporting services 1410, and analytics-based services 1412.

[0141] By way of example, database services 1406 or data services 1408 may facilitate access to one or more searchable databases of public information maintained by a governmental or private entity, and further, to one or more searchable databases of confidential business and/or customer data maintained by IT system 150 and/or the business entity associated with IT system 150 (e.g., a financial institution). Further, in certain aspects, reporting services 1410 and analytics-based services 1412, when provided to the application developer, may provide data indicative of a usage of one or more generated applications, a usage of IT system 150 by application developers and/or other customers, and indication related any additional or alternate activities tracked and/or monitored by IT server 150.

[0142] Further, services 1400 may also include API-based services 1414, compliance- and permissioning-based services 1416, and app-part-based services 1418. By way of example, and as described above, API-based services 1414 may include one or more application programming interfaces (API) that are available to the application developer (e.g., using the web page or GUI associated with the self-sever IT

asset store). APIs consistent with the disclosed embodiments may facilitate a standard method of access one or more core engines (e.g., software processes configured to provide core functionalities for certain services) and hide complexities of connecting to the engines.

[0143] Additionally, in certain aspects, compliance- and permissioning-based services 1416 may include one or more data, application, and/or service compliance and permissioning policies available and selectable by the application developer (e.g., through the web page or GUI of the self-sever IT asset store). For example, compliance and permissioning policies consistent with the disclosed embodiments may include one or more payment card industry (PCI) data-security policies, Sarbanes-Oxley (SOX) policies, perimeter policies, auditing compliance policies, and/or logging and reporting policies. Further, in other aspects, one or more of the compliance and permissioning policies may establish level of access to stored data for the application developer, for individuals associated with the application developer, and/or for individuals representing or associated the business entity (e.g., the financial institution).

[0144] In addition, services 1400 may also include app-part-based services 1418. In certain aspects, one or more of app-part-based services 1418 may include portions of web pages, graphical user interfaces, and other interfaces available to the application developers and selectable by the application developer for presentation during execution of a generated instance of the application. For example, app-part-based services 1418 may include, but is not limited to, a “left nav” portion of a web page that include status graphical content or links to other graphical content that a device (e.g.,

client device 210 of FIG. 2) may access and present upon executing the instantiated application.

[0145] By way of example, a business unit of a financial institution may provide its customers with decisions on loans and other lines of credit, and may require a new mobile platform to facilitate a real-time delivery of these decisions to mobile devices, smart phones, and other wearable devices (e.g., smart watches, etc.) registered to the customers. For example, to generate the required mobile platform, an application developer may create, debug, and deploy mobile applications executable by a variety of potential customer devices (e.g., mobile devices executing Android™ operating systems and/or iOS™). The created mobile application may, for instance, exchange authentication and/or encryption credentials and other data across a communication network using one or more communication protocols (e.g., packetized communications across network 225), and further, may obtain, store, and/or modify business and customer data accessible to the servers and systems of the financial institution (e.g., stored in one or more cloud-based databases or data repositories accessible to the servers and systems).

[0146] In some aspects, the application developer, through a corresponding device (e.g., client device 210 of FIG. 2), may access a web page or other GUI associated with a self-serve IT asset store populated with services available to the application developer (e.g., services 1400 of FIG. 14). The application developer may, for example, select one or more of the available services within the self-serve IT asset store (e.g., one or more of services 1400), and IT system 150 may be configured to present the selected services to the application developer for review (e.g., within a

corresponding IT asset canvas). Further, as described above, the application developer may request access to the selected services (e.g., by clicking on or otherwise selecting a “check out” region of the self-service IT asset store), and an orchestration engine associated with IT system 150 may be configured to assemble one or more mutually compatible combinations of the selected services to generate or “spin-up” instances of the required mobile applications, link the generated instances to sources of data having appropriate permissioning and compliance policies, and package the instantiated applications and linked data sources into modules for delivery to the application developer, and described below in reference to FIG. 15.

[0147] FIG. 15 is a flow chart of an exemplary method 1500 for generating instances of requested applications that incorporate selected information-technology assets, in accordance with the disclosed embodiments. In an embodiment, an orchestration engine associated with and executed by an IT system (e.g., IT system 150 of FIG. 1) may identify one or more services selected by an application developer (e.g., through a web page or GUI associated with a self-serve IT asset store). As described above, the selected services may correspond to information-technology assets accessible to the application developer, and the orchestration engine may be configured compile, link, and/or assemble one or more mutually compatible combinations of the selected services (e.g., infrastructural assets, platform assets, data sources, application programming interfaces, and/or application parts, “apps,” and widgets) to generate or “spin-up” instances of mobile applications. The orchestration engine may, for example, link the generated instances to sources of data having appropriate permissioning and

compliance policies, and may package the instantiated applications and linked data sources into modules for delivery to the application developer.

[0148] In one aspect, the orchestration engine may receive a request from the application developer to access one or more selected services (e.g., in step 1502). As noted above, the application developer may select the available services (e.g., one or more of services 1400) from those presented within a self-serve IT asset store, and IT system 150 may be configured to present the selected services to the application developer for review (e.g., within a corresponding IT asset canvas). In certain aspects, the selected services may correspond to one or more information-technology assets accessible by the application developer, which may include, but are not limited to, infrastructural assets, platform assets, data or database assets, analytical services, reporting services, application programming interfaces (APIs), and application parts. Further, as described above, the application developer may request access to the selected services (e.g., by clicking on or otherwise selecting a “check out” region of the self-service IT asset store).

[0149] In response to the received request, the orchestration engine may determine whether the application developer provided or is capable of providing funds to IT system 150 that are sufficient to obtain access to the selected services (e.g., in step 1504). If the orchestration engine were to determine that the application developer did not provide or is incapable of providing the sufficient funds (e.g., step 1504; NO), the orchestration engine may provide a notification (e.g., within the web page or interface associated with the self-serve IT asset store) that alerts the application

developer to the deficiency (e.g., in step 1506). Exemplary method 1500 is then complete in step 1508.

[0150] If, however, the orchestration engine were to determine that the application developer provides or is capable of providing the sufficient funds (e.g., step 1504; YES), the orchestration engine may determine whether the selected services include at least a minimum set of services required to generate or “spin-up” an instance of the mobile application (e.g., in step 1510). For example, the minimum set of services may include, but is not limited to, at least one selected hardware service, at least one selected network resource, at least one selected cloud stack service, selected virtual machine assets, and corresponding APIs that facilitate communications between the selected assets, resources, and services. The minimum set of services may, for example, include services having predetermined classes of information-technology assets (e.g., “asset classes) required to spin-up the instance and enable the instance to be fully functional when executed by a specified combination of infrastructural assets (e.g., as included within the selected services).

[0151] In some instances, IT system 150 and/or the business entity may establish the minimum set of services required to spin-up the instance. In other instances, however, the minimum set of services may vary in accordance with characteristics of the set of services (e.g., the minimum services necessary to support a mobile app for a financial institution’s loan business may differ from the minimum services necessary to support a mobile banking application).

[0152] If the orchestration engine were to determine the application developer did not select the minimum number of services (e.g., in step 1510; NO), the orchestration

engine may identify one or more additional services that, in conjunction with the selected services, form at least the minimum number of services required to spin up the mobile application instance (e.g., in step 1512). In certain aspects, the orchestration engine may determine that the services selected by the application developer fail to include information-technology assets associated with at least one of the predetermined asset classes (e.g., which may be required to spin-up the instance), and the one or more additional services identified in step 1512 may correspond to information-technology assets associated with the at least one predetermined asset classes.

[0153] By way of example, selected services may include hardware assets (e.g., hardware assets 1302 of FIG. 13) and cloud connectivity (e.g., cloud stack assets 1304 of FIG. 13), but may not include the necessary operating system and middleware. In some aspects, the orchestration engine may identify an operating system (e.g., operating systems 1306A of FIG. 13) and corresponding middleware (e.g., middleware 1306B of FIG. 13) that are appropriate for the selected hardware assets and cloud connectivity. In step 1512, the orchestration engine may add the identified operating system and middleware to the selected services.

[0154] In other instances, the selected services may include the necessary hardware assets, cloud connectivity, operating system, and middleware, but may fail to include a selection of an application supported by a corresponding virtual machine. In one aspect, the orchestration engine may select a web server, a web farm, and/or clustering infrastructure (e.g., one of applications 1306C of FIG. 13) appropriate to the previously selected hardware assets, cloud connectivity, operating system, and

middleware. In step 1512, the orchestration engine may add the identified application (or applications) to the selected services.

[0155] In step 1514, the orchestration engine may obtain information (e.g., metadata) associated with the selected services, and may determine whether the selected services are functionally compatible and capable of exchanging data and other programmatic commands through the corresponding selected APIs. Further, in step 1514, the orchestration engine may further determine whether the selected services are consistent with at least one compliance or permissioning policy specified by the application developer (e.g., one or more of services 1416 of FIG. 14) and/or associated with the orchestration engine.

[0156] In certain instances, if the orchestration engine were to identify an incompatible or inconsistent one of the selected services (e.g., step 1514; NO), the orchestration engine may identify a replacement service that is functionally compatible with the other selected services (e.g., in step 1516). The orchestration engine may, for example, automatically add the replacement service to the other services selected by the application developer in step 1516.

[0157] In some embodiments, the orchestration engine may be configured to may assemble, compile, and/or link combinations of the services to generate or “spin-up” one or more instances of mobile application requested by the application developer (e.g., in step 1518). In one instance, the orchestration engine may be configured to spin-up a single instance of the requested mobile application based on the selected services. In other aspects, however, the orchestration engine may be configured to

spin-up any arbitrary number of instances of the requested mobile application (e.g., instances 1460 of FIG. 14).

[0158] By way of example, the orchestration engine may spin-up instances of the requested mobile application appropriate for initial testing and development, for virtualization (e.g., using applications, middleware, and operating systems associated with a virtual machine), and further, for quality assurance purposes. In certain aspects, the application developer may associate one or more database services (e.g., service 1406 of FIG. 14) and/or data services (e.g., services 1408 of FIG. 14) with corresponding ones of the generated instances, and further, the generated instances may be associated with various capabilities to modify accessed data. Further, as described above, the application developer may delegate one or more of these testing, development, virtualization, and quality-assurance processes to individuals having various relationships with the financial institution and/or the developer, and thus, having various capabilities to access the underlying business and customer data required by the applications.

[0159] Further, as noted above, the generated instances may be associated with one or more selected data permissioning and compliance services (e.g., one or more of services 1416 of FIG. 14). In some embodiments, the orchestration server may access information associated with the selected data permissioning and compliance services associated with the generated instances (e.g., metadata provided by IT system 150), and may determine, for each of the generated instances, whether the selected services (including the selected data and database services) are consistent with the selected data permissioning and compliance services (e.g., in step 1520). Further, in step 1520,

the orchestration server may also determine whether data access permissions associated with the individuals to whom the application developer delegated execution of the generated instances are consistent with the selected data permissioning and compliance services.

[0160] If it is determined that no inconsistencies exist (e.g., step 1520; NO), the orchestration engine may establish links between the generated instances and corresponding ones of the selected data and/or database services (e.g., in step 1522). The orchestration engine may be further configured to package the generated instances and information identifying the linked data and/or database services into a module having a format suitable for delivery to a device (e.g., client device 210 of FIG. 2) associated with the application developer (e.g., in step 1524). By way of example, the orchestration engine may package the generated instance as Javascript or other HTML code, with information identifying the linked being provided as metadata. In other instances, the generated instances and linking information may be compiled together and provided to the device as an executable file.

[0161] In some aspects, the orchestration engine may be configured to provide the one or more packaged modules to the application developer's device using any of the communications protocols outlined above (e.g., in step 1524). Additionally or alternatively, the orchestration engine may be configured to provide information and/or links to the application developer's device that enable the application developer to download or otherwise obtain the one or more packaged modules from a locally accessible data repository. In certain aspects, and using the packaged modules, the application developer may generate, debug, and deploy the mobile applications

necessary to provide the mobile platform requested by the financial institution.

Exemplary method 1500 is then complete in step 1508.

[0162] If, however, it is determined that inconsistencies exist between the selected services (and/or the delegated individuals) and the selected data permissioning and compliance services (e.g., step 1520; YES), exemplary processes may pass back to step 1506. In some aspects, the orchestration engine may provide a notification (e.g., within the web page or interface associated with the self-serve IT asset store) that alerts the application developer to the inconsistency, and exemplary method 1500 is then complete in step 1508.

[0163] Referring back to step 1510, if the orchestration engine were to determine that the application developer selected at least the minimum number of services (e.g., step 1510; YES), the orchestration engine may obtain information (e.g., metadata) associated with the selected services, and determine whether the selected services are mutually compatible and capable of exchanging data and other programmatic commands through the corresponding selected APIs, as described above in reference to step 1514. Further, if it is determined that the selected services are mutually compatible (e.g., step 1514; YES), the orchestration engine may be configured to assemble combinations of the selected services to generate or “spin-up” one or more instances of mobile application requested by the application developer, as described above in step 1518.

[0164] Using the disclosed embodiments, an orchestration engine provided by IT system 150 may provide enterprise-building services that generate or “spin-up” instances of executable applications, that link the generated instances to sources of

data having appropriate permissioning and compliance policies, and that package the instantiated applications and linked data sources into a module for delivery to an application developer. In certain instances, enterprise-building services consistent with the disclosed embodiments may enable an application developed to rapidly generate, debug, and deploy mobile applications supporting new and complex enterprise platforms.

[0165] Further, in the embodiments described above. IT system 150 may execute one or more software processes to provide the functionality of the orchestration system. In other embodiments, an additional computer system in communication with IT system 150 may be configured to execute the software processes that provide the orchestration engine functionality. Further, in additional embodiments, the orchestration engine may be provided as specialized hardware in communication with IT system 150 and devices of the application developer across a corresponding communications network.

Other embodiments will be apparent to those skilled in the art from consideration of the specification and practice of the embodiments disclosed herein. The scope of the claims should not be limited by the embodiments set forth in the examples, but should be given the broadest interpretation consistent with the description as a whole.

WHAT IS CLAIMED IS:

1. An apparatus, comprising:

a communications module;

a storage device storing instructions; and

at least one processor coupled to the communications module and the storage device, and configured to execute the instructions to:

receive, from a device through the communications module,

a request to generate an instance of an application,

the request comprising information identifying first

information-technology assets, the first

information-technology assets comprising a first

executable application part;

in response to the received request, determine that the first

information-technology assets are consistent with a

permissioning or compliance policy;

obtain first data associated with the first information-

technology assets in response to the determination,

the obtained first data characterizing functionalities of

the first information-technology assets;

detect, based on the obtained first data, an incompatibility

between the functionality of the first executable

application part and a functionality of a first

programmatic interface associated with a subset of

the first information-technology assets;

in response to the detected incompatibility, identify a second

executable application part having a functionality that

is compatible with the functionality of the first

programmatic interface, the second executable

application part being consistent with the
permissioning or compliance policy;

generate the application instance based on the subset of the
first information-technology assets, the first
programmatic interface, and the second executable
application part; and

transmit, through the communications module, information
associated with the generated application instance to
the device, the generated application instance being
executable by the device.

2. The apparatus of claim 1, wherein:

the first information-technology assets comprise an infrastructural asset, a
platform asset, a source of data, a reporting service, an analytics
service, an additional programmatic interface, or an additional
executable application part; and

the infrastructural assets comprise a hardware asset, a cloud-stack asset,
a network resource, or an operating system and corresponding
middleware.

3. The apparatus of claim 2, wherein the first information-technology assets specify
a combination of the infrastructural and platform assets, the generated
application instance being executable by the combination of infrastructural and
platform assets.

4. The apparatus of claim 3, wherein:

the first information-technology assets specify combinations of
infrastructural and platform assets; and

the at least one processor is further configured to generate application instances executable by corresponding ones of the combinations of infrastructural and platform assets.

5. The apparatus of any one of claims 1 - 4, wherein:

the at least one processor is further configured to link together the second executable application part and the first programmatic interface, the linked second executable application part and the first programmatic interface establishing the instance of the application.

6. The apparatus of any one of claims 1 - 5, wherein:

the first information-technology assets comprise a source of data; and

the at least one processor is further configured to:

determine whether the data source complies with the permissioning or compliance policy;

when the data source is determined to comply with the permissioning or compliance policy, associate the application instance with the data source; and

generate an electronic command to transmit information identifying the application instance and the associated data source to the device.

7. The apparatus of any one of claims 1 - 6, wherein the first information-technology assets specify the permissioning or compliance policy, the permissioning or compliance policy establishing (i) a first restriction on an ability of the device to access the first information-technology assets or (ii) a second restriction on an ability of a corresponding one of the first information-technology assets to exchange data with another one of the first information-technology assets.

8. The apparatus of any one of claims 1 - 7, wherein the at least one processor is further configured to:
- identify asset classes associated with the first information-technology assets;
 - determine whether the identified asset classes include predetermined asset classes; and
 - generate the application instance when the identified asset classes are determined to include the predetermined asset classes.
9. The apparatus of claim 8, wherein the at least one processor is further configured to:
- when the identified asset classes are determined not to include at least one of the predetermined asset classes, identify a second information-technology asset associated with the at least one predetermined asset class;
 - obtain second data characterizing a functionality of the second information-technology asset;
 - detect the incompatibility between the functionalities of the first executable application part and the first programmatic interface based on the obtained first and second data, the functionality of the first programmatic interface being compatible with functionalities of a subset of the first and second information-technology assets; and
 - assemble the subset of the first and information-technology assets, the second application part, and the first programmatic interface, to generate the application instance.

10. The apparatus of any one of claims 1 - 9, wherein the at least one processor is further configured to:
- identify a second information-technology asset when at least one of the first information technology assets is determined to be inconsistent with the permissioning or compliance policy, the second information-technology asset being consistent with the permissioning or compliance policy;
 - obtain second data characterizing a functionality of the second information-technology asset;
 - detect the incompatibility between the functionalities of the first executable application part and the first programmatic interface based on the obtained first and second data, the functionality of the first programmatic interface being compatible with the functionalities of a subset of the first and second information-technology assets; and
 - generate the application instance based on the subset of the first and second information technology assets, the second application part, and the first programmatic interface.
11. The apparatus of any one of claims 1 - 10, wherein the at least one processor is further configured to:
- provide, to the device, information identifying candidate information-technology assets, the information instructing the device to present at least a portion of the candidate information-technology services through an interface; and
 - receive, from the device, a selection of a corresponding one of the candidate information-technology assets; and
 - establish the selection as a portion of the first information-technology assets.

12. The apparatus of any one of claims 1 - 11, wherein the at least one processor is further configured to establish a container-based framework for the instance, the container-based framework being consistent with of the subset of the first information-technology assets, the first programmatic interface, or the second executable application part.
13. The apparatus of any one of claims 1 - 12, wherein the at least one processor is further configured to determine whether the first information-technology assets are consistent with a plurality of permissioning or compliance policies, at least one of the permissioning or compliance policies being specified by a user of the device.
14. A computer-implemented method, comprising:
 - receiving, using at least one processor, and from a device, a request to generate an instance of an application, the request comprising information identifying first information-technology assets, and the first information-technology assets comprising a first executable application part;
 - in response to the received request, determining, using the at least one processor, that the first information-technology assets are consistent with a permissioning or compliance policy;
 - obtaining, using the at least one processor, first data associated with the first information-technology assets in response to the determination, the obtained data characterizing functionalities of the first information-technology assets;
 - based on the obtained first data, detecting, using the at least one processor, an incompatibility between the functionality of the first executable application part and a functionality of a first

programmatic interface associated with a subset of the first information-technology assets;

in response to the detected incompatibility, identifying, using the at least one processor, a second executable application part having a functionality that is compatible with the functionality of the first programmatic interface, the second executable application part being consistent with the permissioning or compliance policy;

generating, using the at least one processor, the application instance based on the subset of the first information-technology assets, the first programmatic interface, and the second executable application part; and

transmitting, using the at least one processor, information associated with the generated application instance to the device, the generated application instance being executable by the device.

15. The computer-implemented method of claim 14, wherein:

the first information-technology assets comprise an infrastructural asset, a platform asset, a source of data, a reporting service, an analytics service, an additional programmatic interface, or an additional executable application part; and

the infrastructural assets comprise a hardware asset, a cloud-stack asset, a network resource, or an operating system and corresponding middleware.

16. The computer-implemented method of claim 15, wherein:

the first information-technology assets specify combinations of the infrastructural and platform assets; and

the method further comprises generating application instances executable by corresponding ones of the combinations of infrastructural and platform assets.

17. The computer-implemented method of claim 15, wherein the method further comprises linking together the second executable application part and the first programmatic interface, the linked second executable application part and the first programmatic interface establishing the instance of the application.
18. The computer-implemented method of any one of claims 14 - 17, wherein the first information-technology assets comprise a source of data, the method further comprising:
 - determining whether the data source complies with the permissioning or compliance policy;
 - when the data source is determined to comply with the permissioning or compliance policy, associating the instance with the data source;
 - and
 - generating an electronic command to transmit information identifying the instance and the associated data source to the device.
19. The computer-implemented method of any one of claims 14 - 18, wherein the first information-technology assets specify the permissioning or compliance policy, the permissioning or compliance policy establishing (i) a first restriction on an ability of the device to access the first information-technology assets or (ii) a second restriction on an ability of a corresponding one of the first information-technology assets to exchange data with another one of the first information-technology assets.

20. The computer-implemented method of claim 14, further comprising:
- identifying asset classes associated with the first information-technology assets;
 - determining whether the identified asset classes include predetermined asset classes; and
 - generate the instance of the application when the identified asset classes are determined to include the predetermined asset classes, and when the first information-technology assets are consistent with the permissioning or compliance policy.
21. The computer-implemented method of claim 20, further comprising:
- when the identified asset classes are determined not to include at least one of the predetermined asset classes, identifying a second third information-technology service associated with the at least one predetermined asset class;
 - obtaining second data characterizing a functionality of the second information-technology asset;
 - detecting the incompatibility between the functionalities of the first executable application part and the first programmatic interface based on the obtained first and second data, the functionality of the first programmatic interface being compatible with the functionalities of a subset of the first and second information-technology assets; and
 - assembling the subset of the first and information-technology assets, the second application part, and the first programmatic interface, to generate the application instance.

22. The computer-implemented method of any one of claims 14 - 21, further comprising:
- identifying a third second information-technology asset when at least one of the first information technology assets is determined to be inconsistent with the permissioning or compliance policy, the second information-technology asset being consistent with the permissioning or compliance policy;
 - obtaining second data characterizing a functionality of the second information-technology asset;
 - detecting the incompatibility between the functionalities of the first executable application part and the first programmatic interface based on the obtained first and second data, the functionality of the first programmatic interface being compatible with the functionalities of a subset of the first and second information-technology assets
 - generating the application instance based on the subset of the first and second information technology assets, the second application part, and the first programmatic interface.
23. The computer-implemented method of any one of claims 14 - 22, further comprising:
- providing, to the device, information identifying candidate information-technology assets, the information instructing the user device to present of the candidate information-technology services to the user within an interface; and
 - receiving, from the device, a selection of at least one of the candidate information-technology assets; and
 - establishing the user selection as a portion of the first information-technology assets.

24. A tangible, non-transitory computer-readable medium storing instructions that, when executed by at least one processor, perform a method comprising:
- receiving, from a device, a request to generate an instance of an application, the request comprising information identifying first information-technology assets, and the first information-technology assets comprising a first executable application part;
 - in response to the received request, determining that the first information-technology assets are consistent with a permissioning or compliance policy;
 - obtaining first data associated with the first information-technology assets in response to the determination, the obtained first data characterizing functionalities of the first information-technology assets;
 - based on the obtained first data, detecting an incompatibility between the functionality of the first executable application part and a first programmatic interface associated with a subset of the first information-technology assets;
 - in response to the detected incompatibility, identifying a second executable application part having a functionality that is compatible with the functionality of the first programmatic interface, the second executable application part being consistent with the permissioning or compliance policy;
 - generating the application instance based on the subset of the first information-technology assets, the first programmatic interface, and the second executable application part; and
 - transmitting information associated with the generated application instance to the device, the generated application instance being executable by the device.

25. The apparatus of any one of claims 1 - 13, wherein the at least one processor is further configured to:
- determine, based on the obtained first data, that the first executable application part is incapable of exchanging data with the subset of the first information-technology assets through the first programmatic interface;
 - detect the incompatibility between the functionalities of the first executable application part and the first programmatic interface in response to the determination that the first executable application part is incapable of exchanging the data with the subset of the first information-technology assets through the first programmatic interface;
 - establish that the second executable application part is capable of exchanging data with the subset of the first information-technology assets through the first programmatic interface; and
 - generate the application instance in response to the established capability of the second executable application part.
26. The apparatus of claim 25, wherein:
- the first programmatic interface comprises an application programming interface; and
 - the at least one processor is further configured to:
 - determine, based on the obtained first data, that the first executable application part is incapable of exchanging programmatic commands with the application programming interface; and
 - detect the incompatibility between the functionalities of the first executable application part and the application programming interface in

response to the determination that the first executable application part is incapable of exchanging the programmatic commands within the application programming interface.

27. The apparatus of any one of claims 1 – 13, 25 and 26, wherein:

the first information-technology assets further comprise the first programmatic interface; and

the at least one processor is further configured to:

identify a second programmatic interface having a functionality that is compatible with (i) the functionality of the first executable application part and (ii) the functionalities of the subset of the first information-technology assets; and

generate the application instance based on the subset of the first information-technology assets, the first executable application part, and the second programmatic interface.

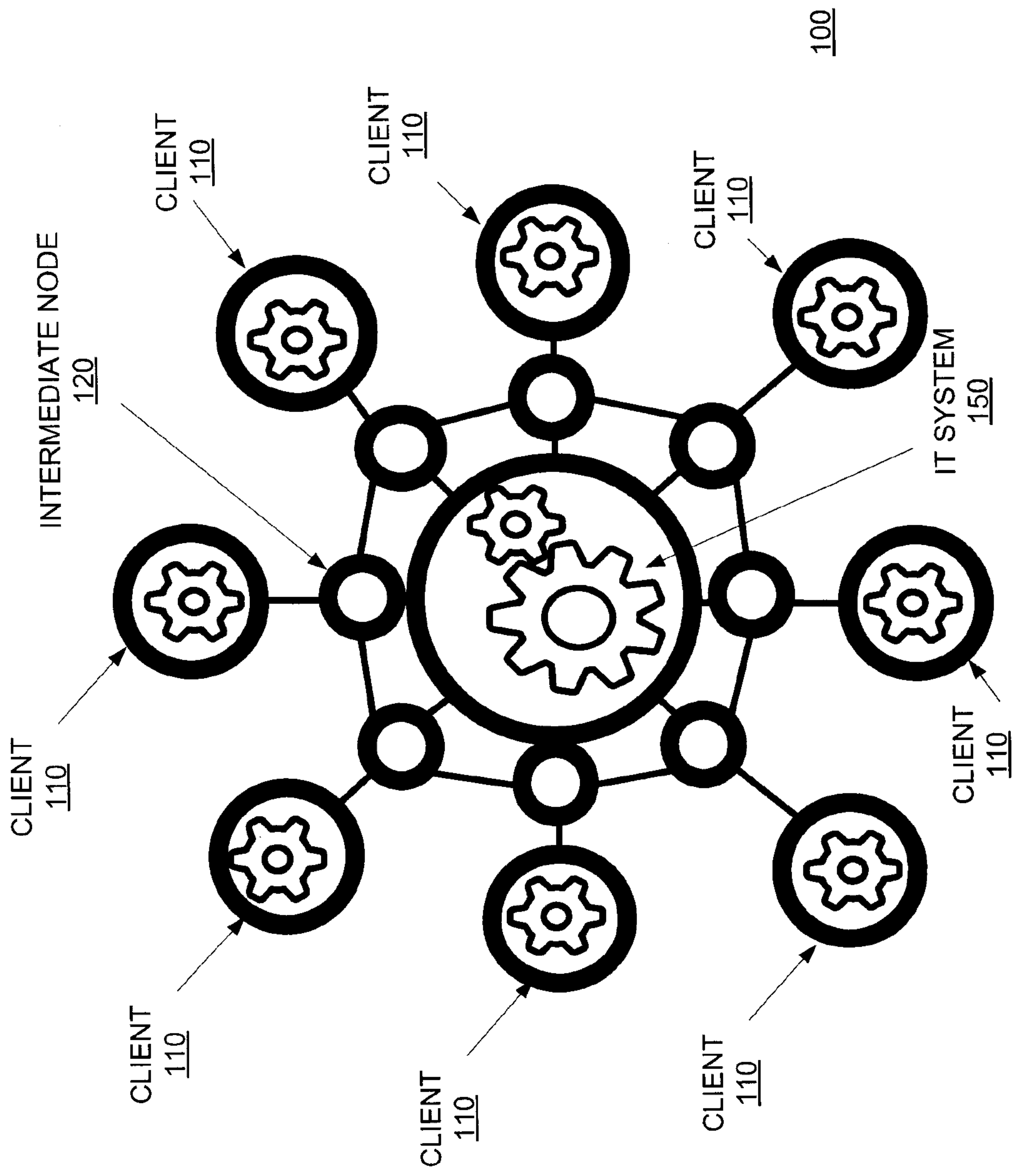


FIG. 1

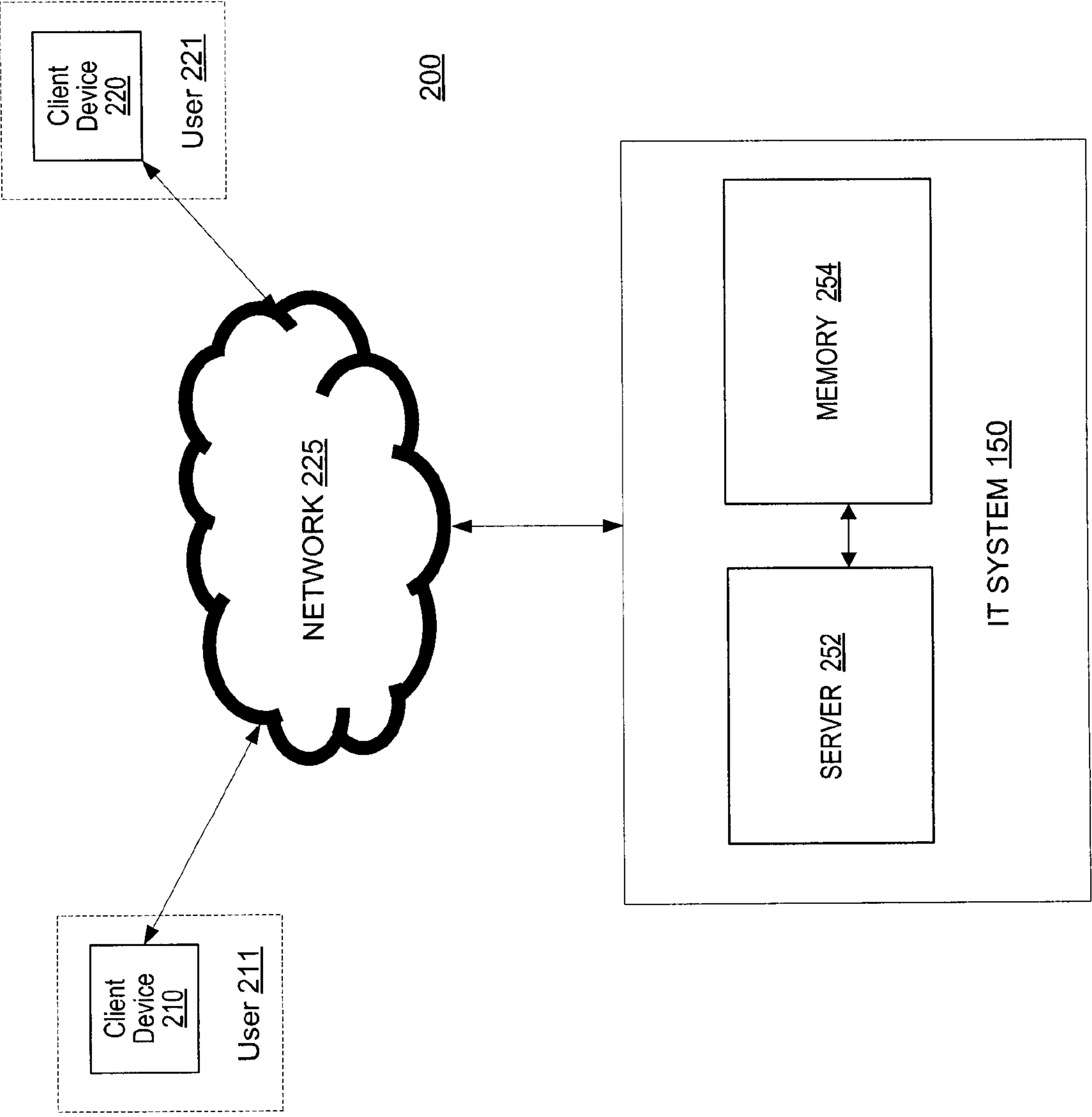


FIG. 2

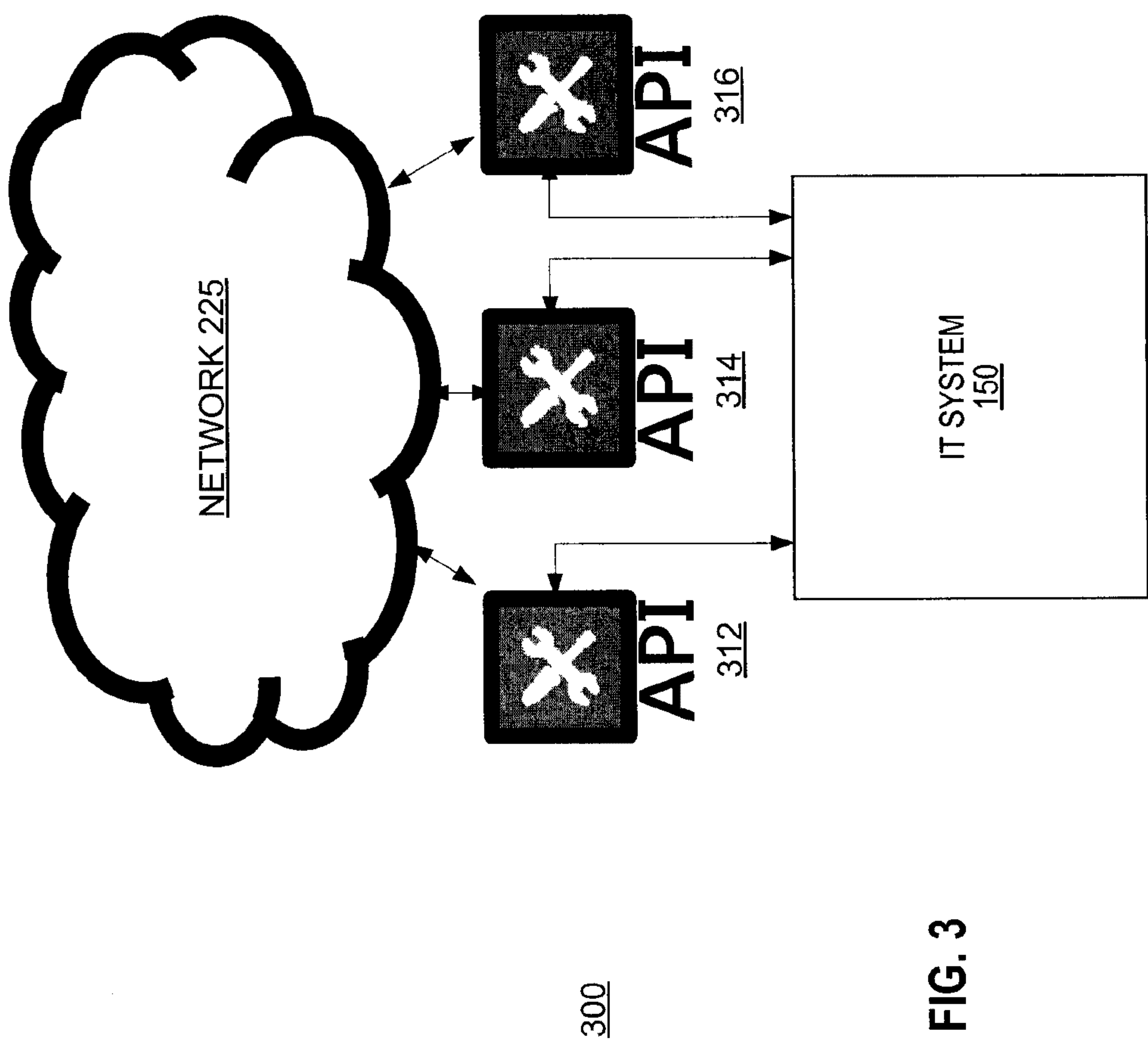


FIG. 3

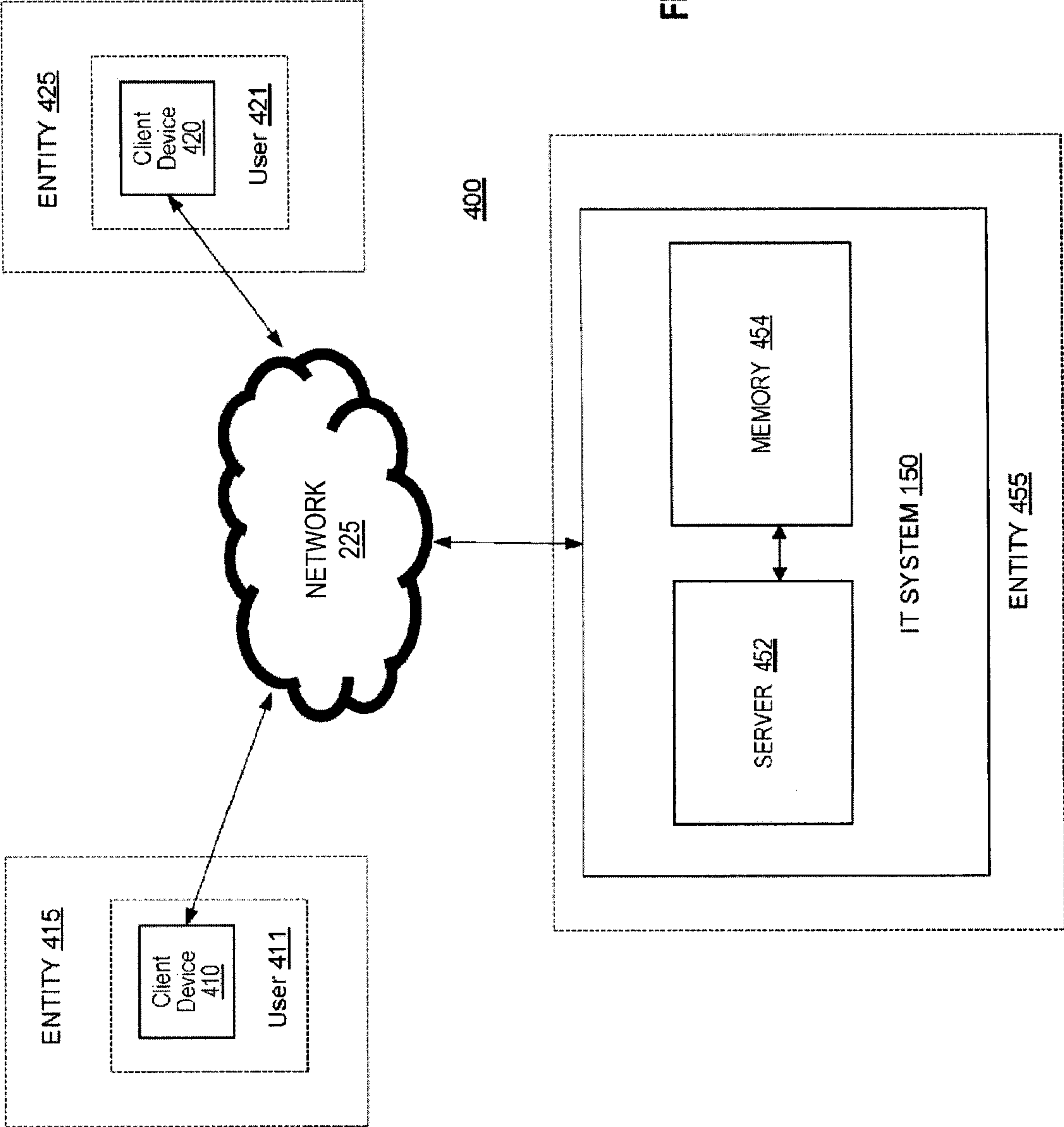


FIG. 4

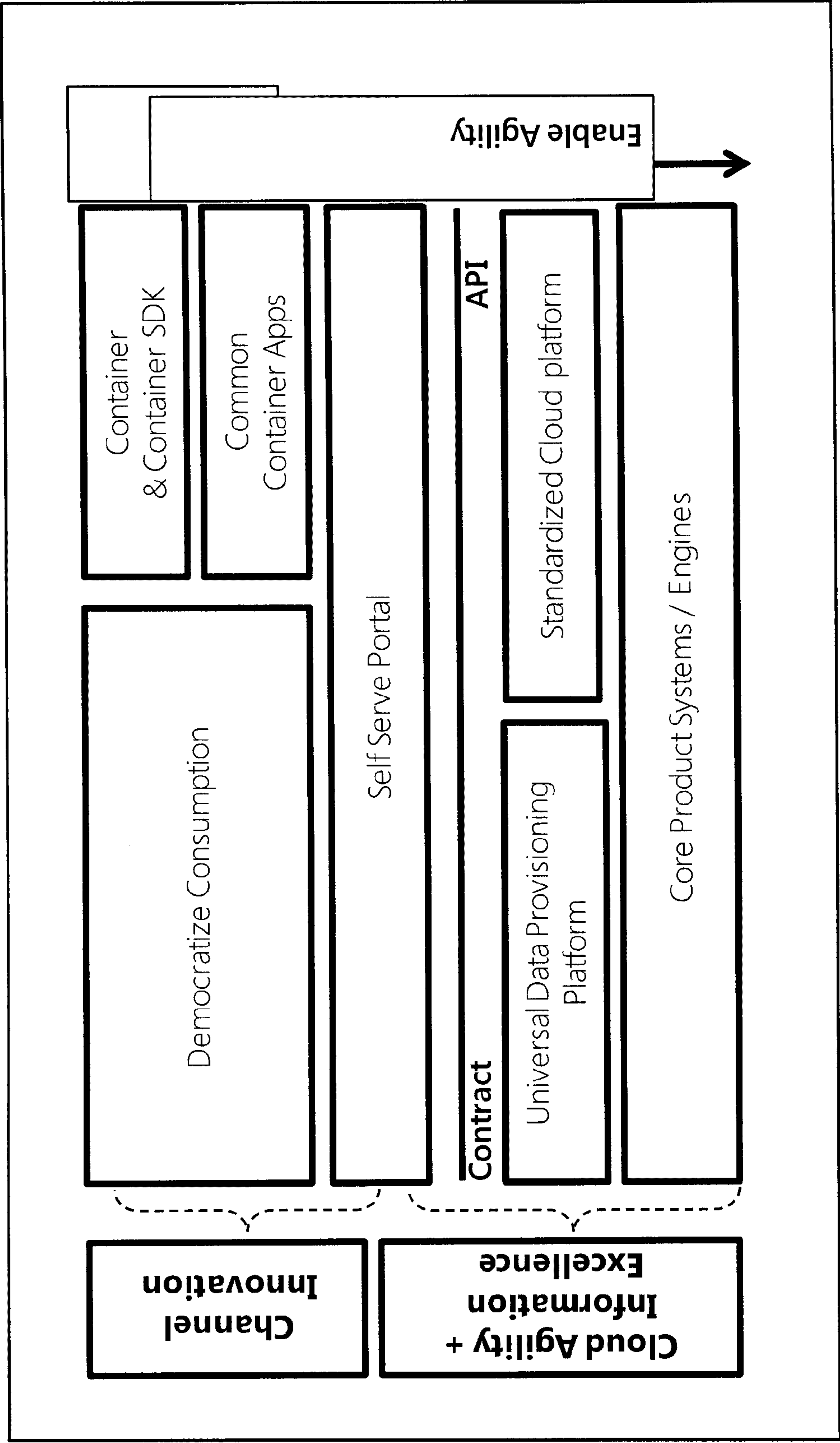


FIG. 5

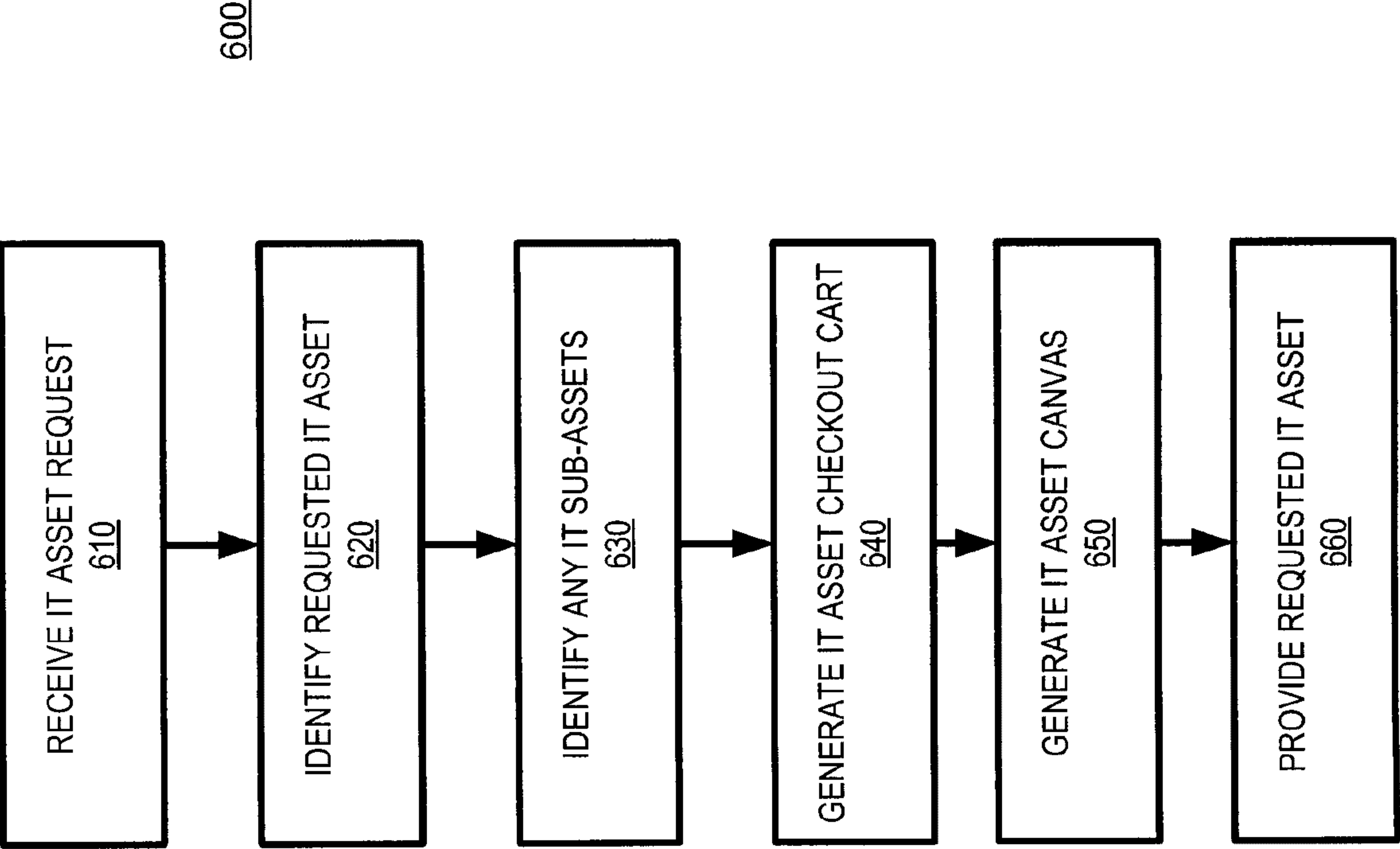


FIG. 6

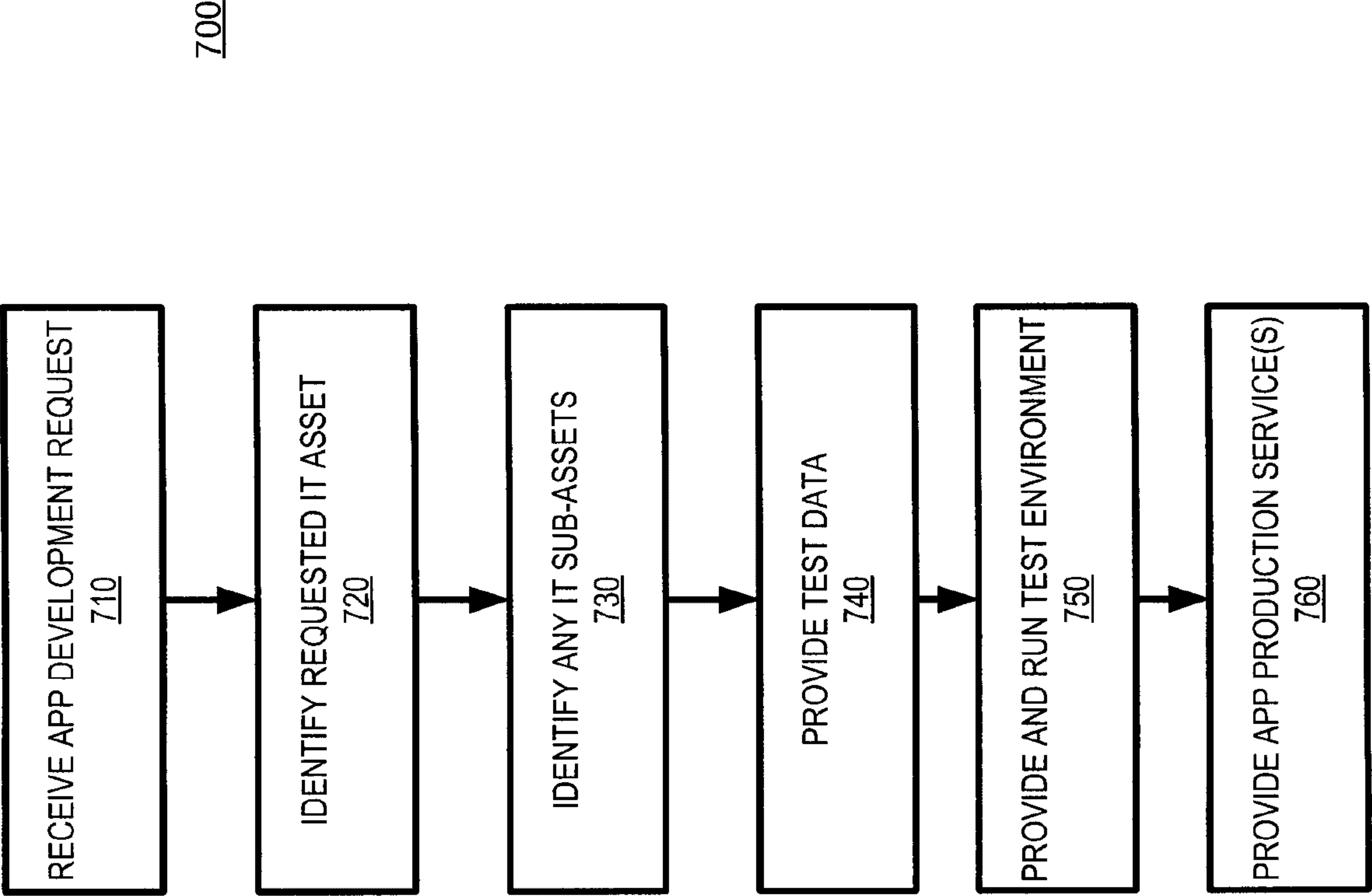


FIG. 7

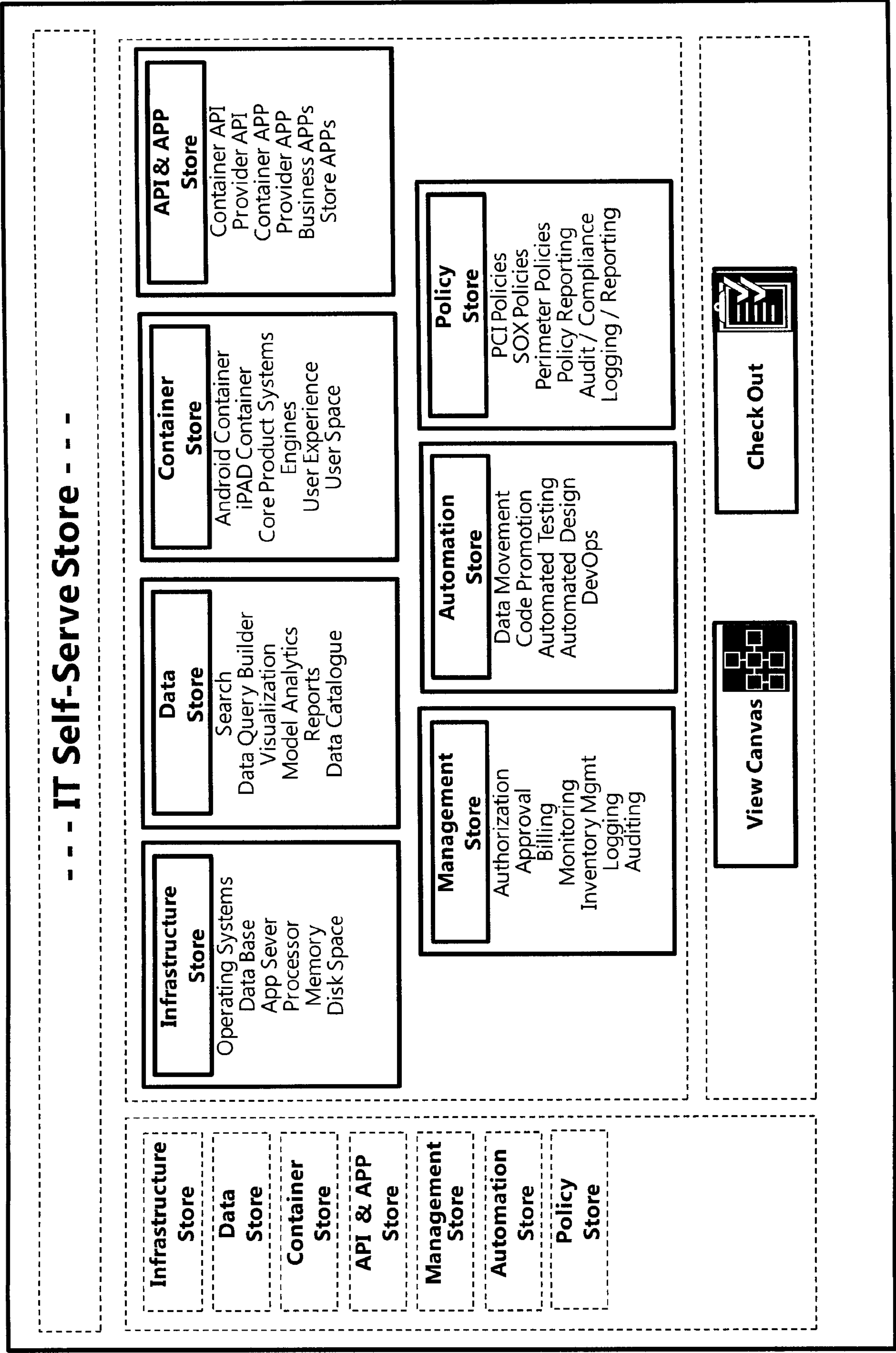


FIG. 8

900

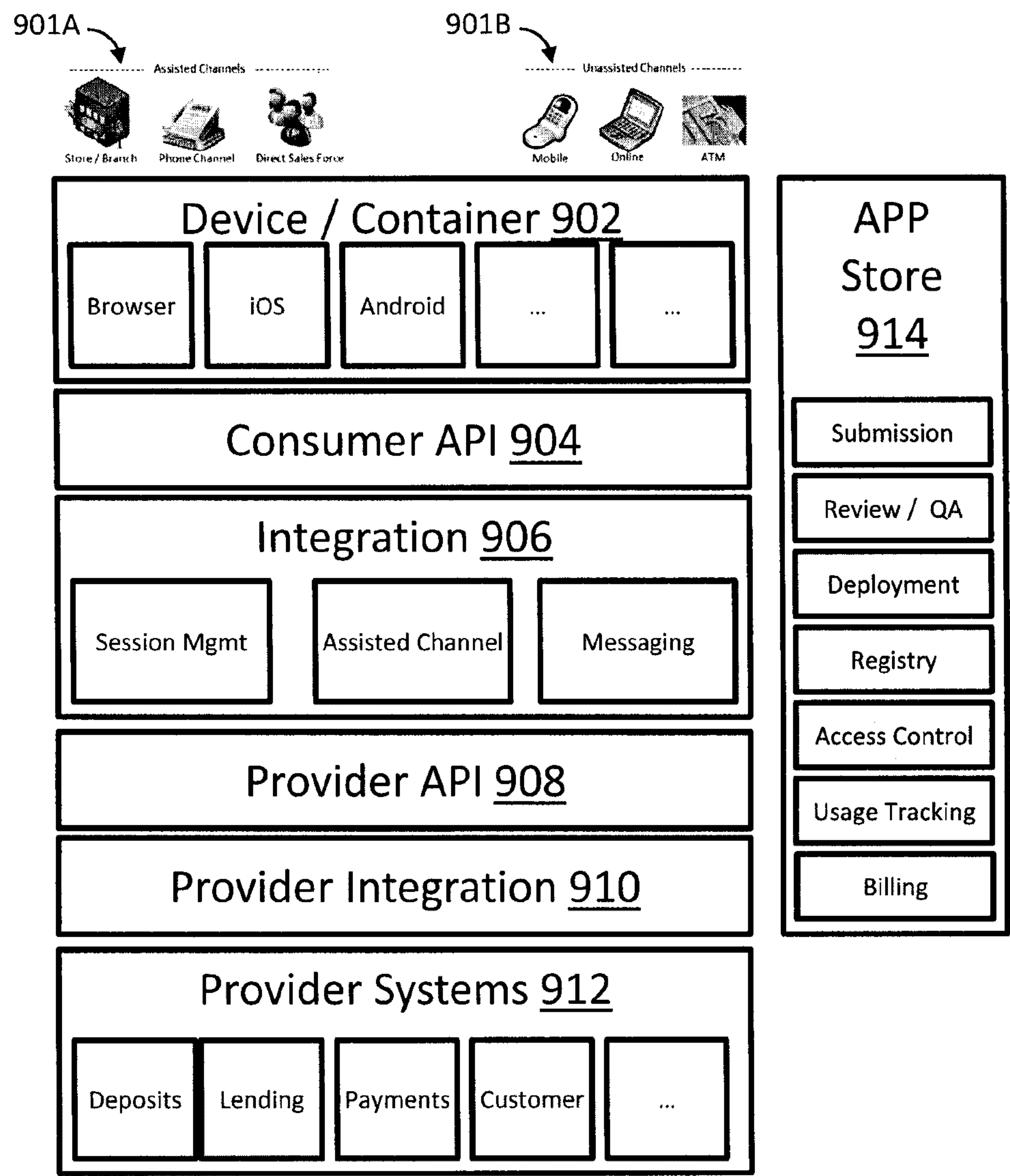


FIG. 9

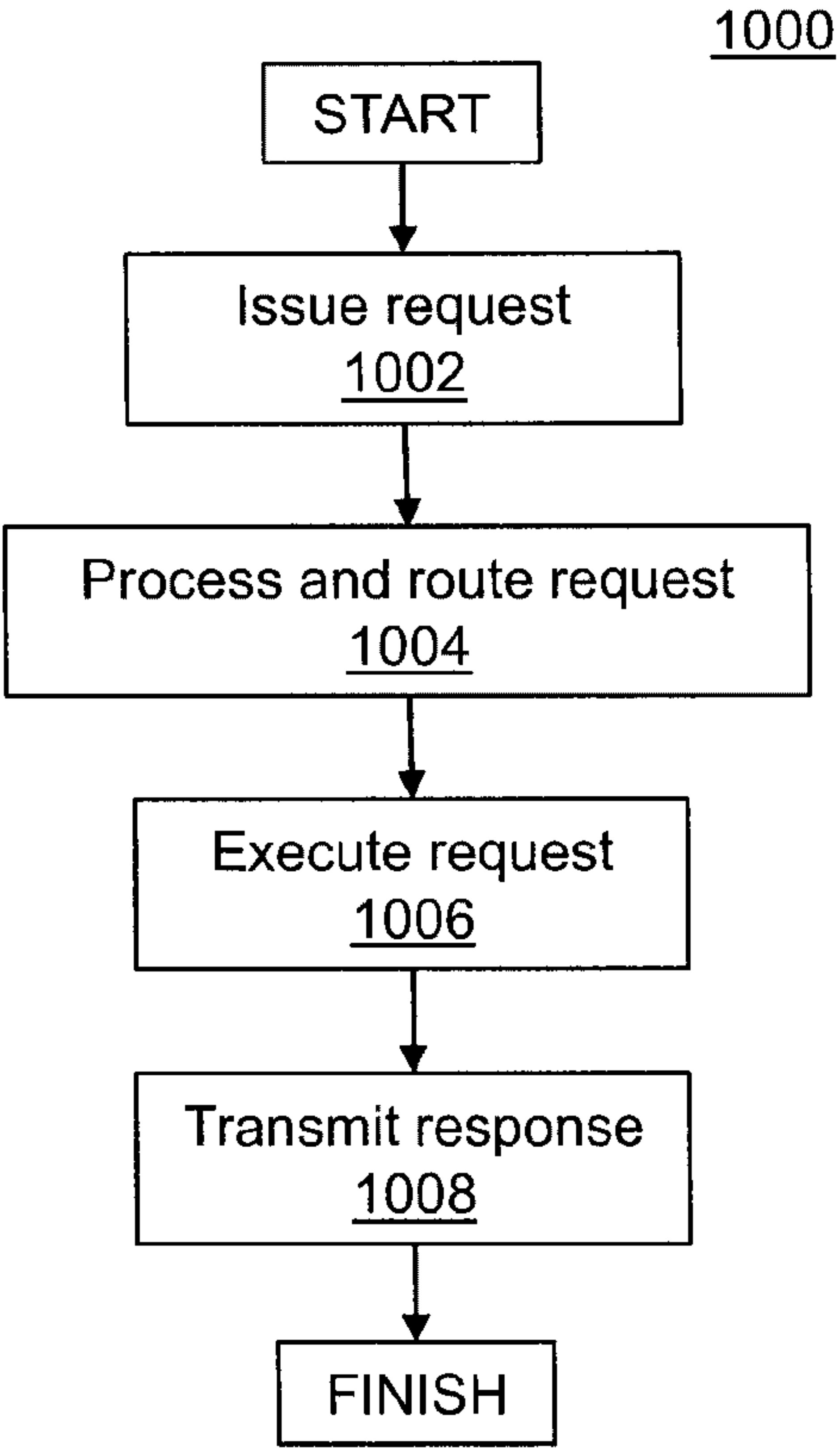


FIG. 10A

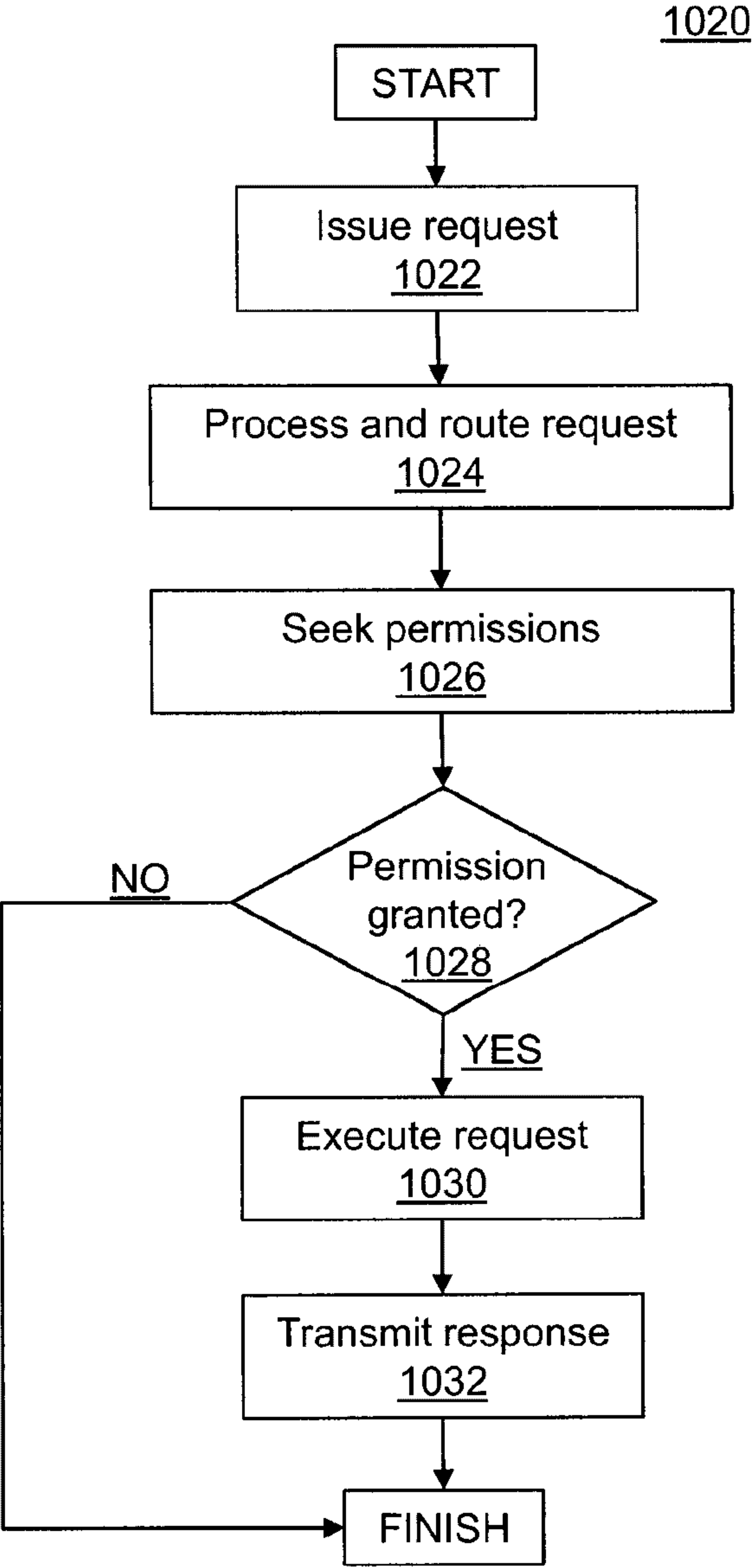


FIG. 10B

1040

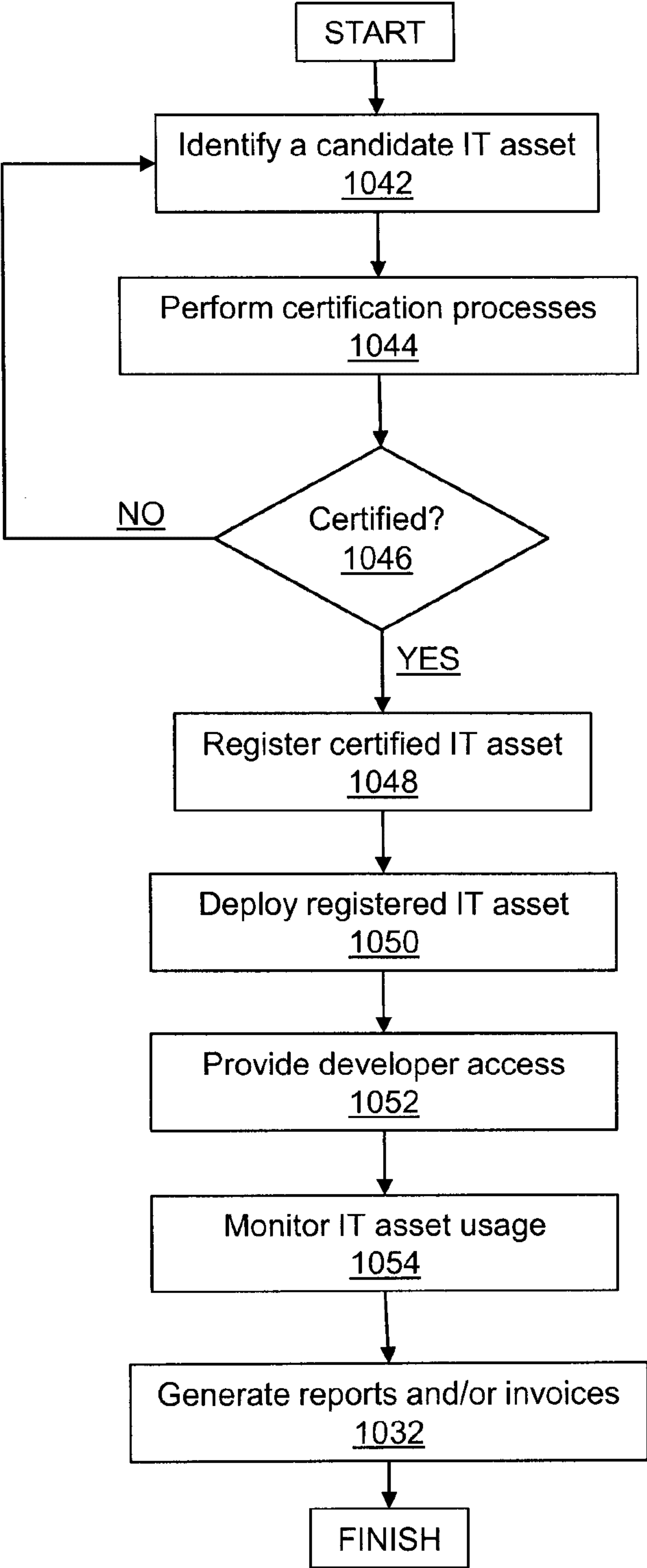


FIG. 10C

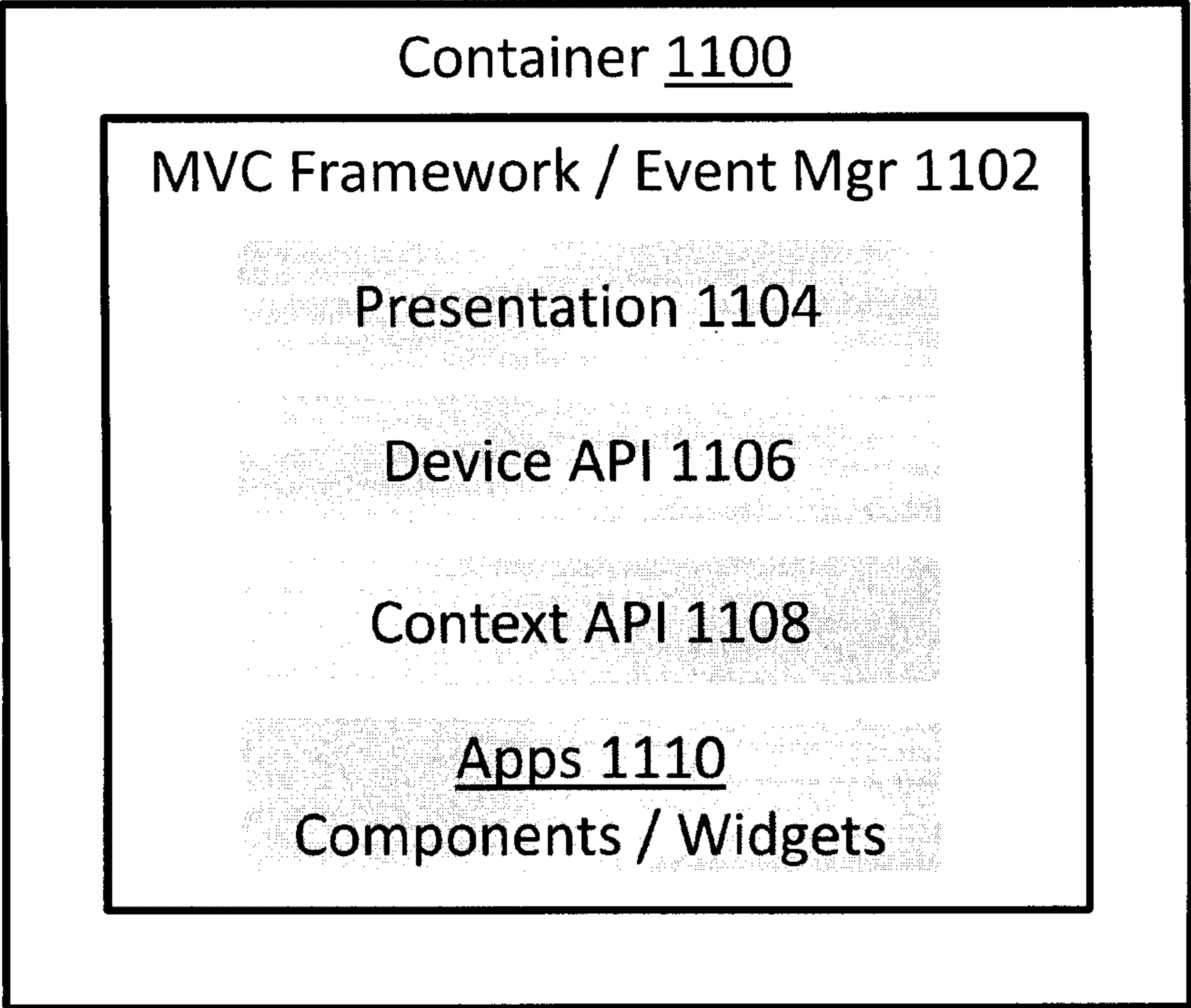


FIG. 11

1200

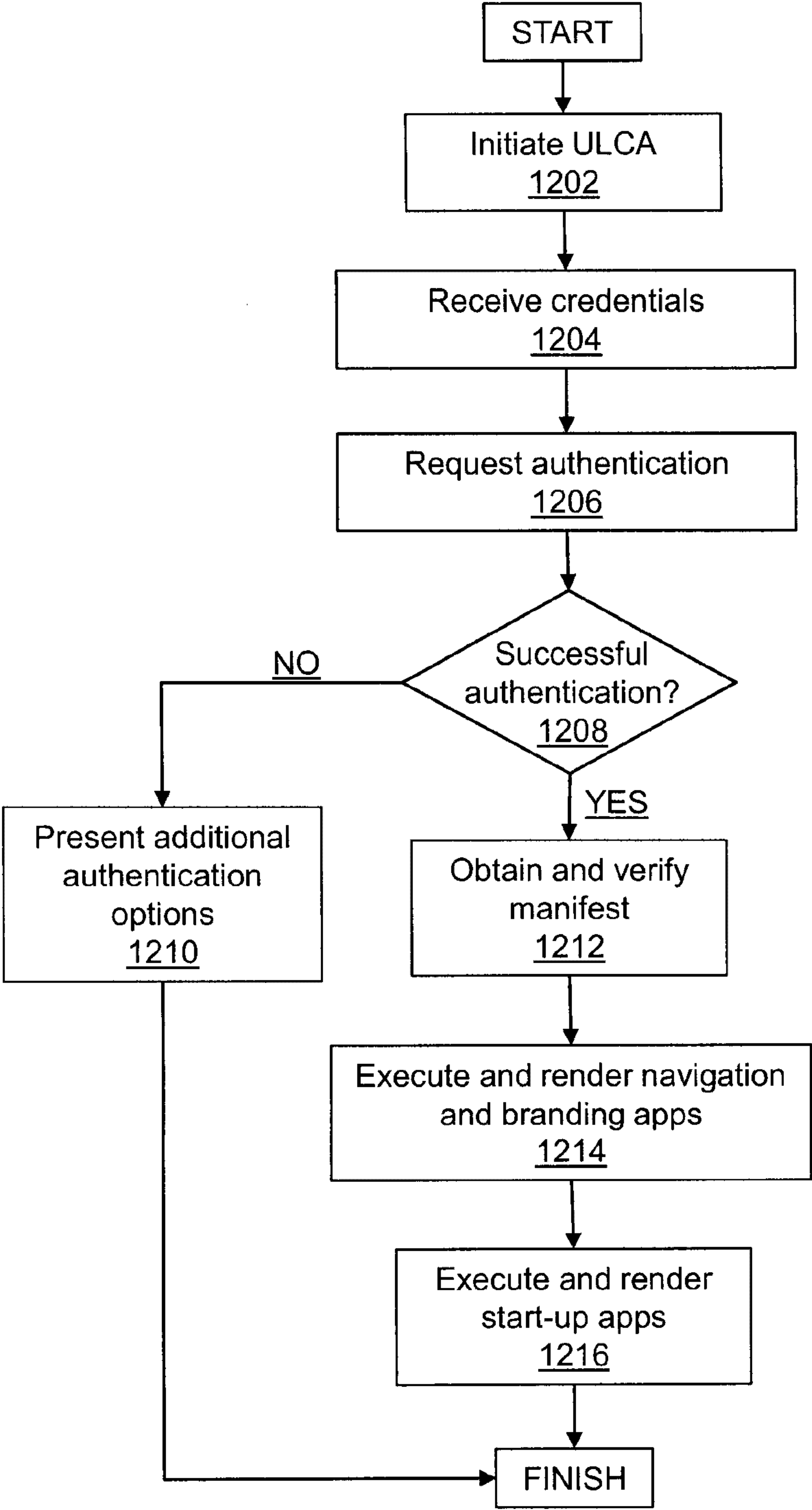


FIG. 12

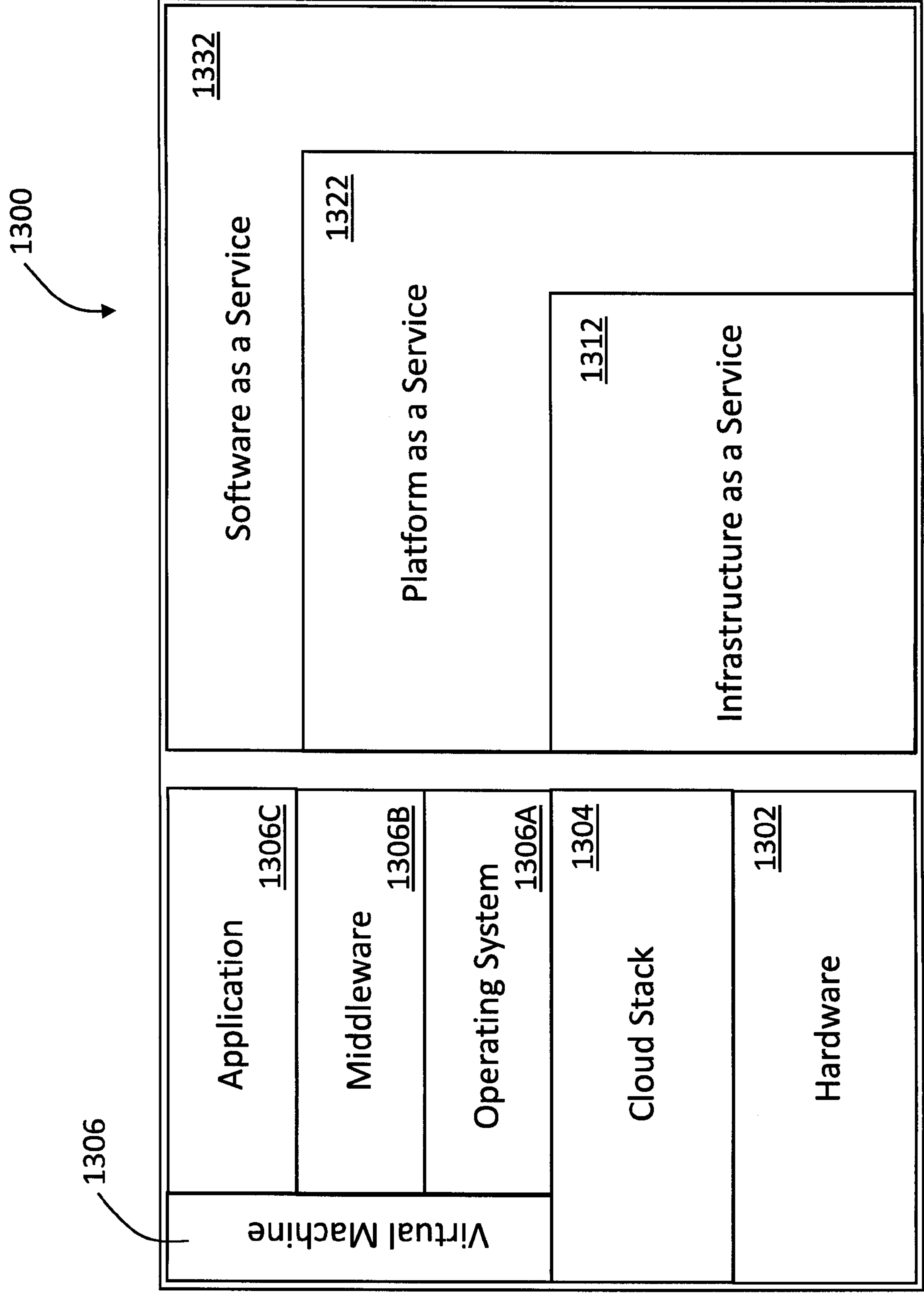


FIG. 13

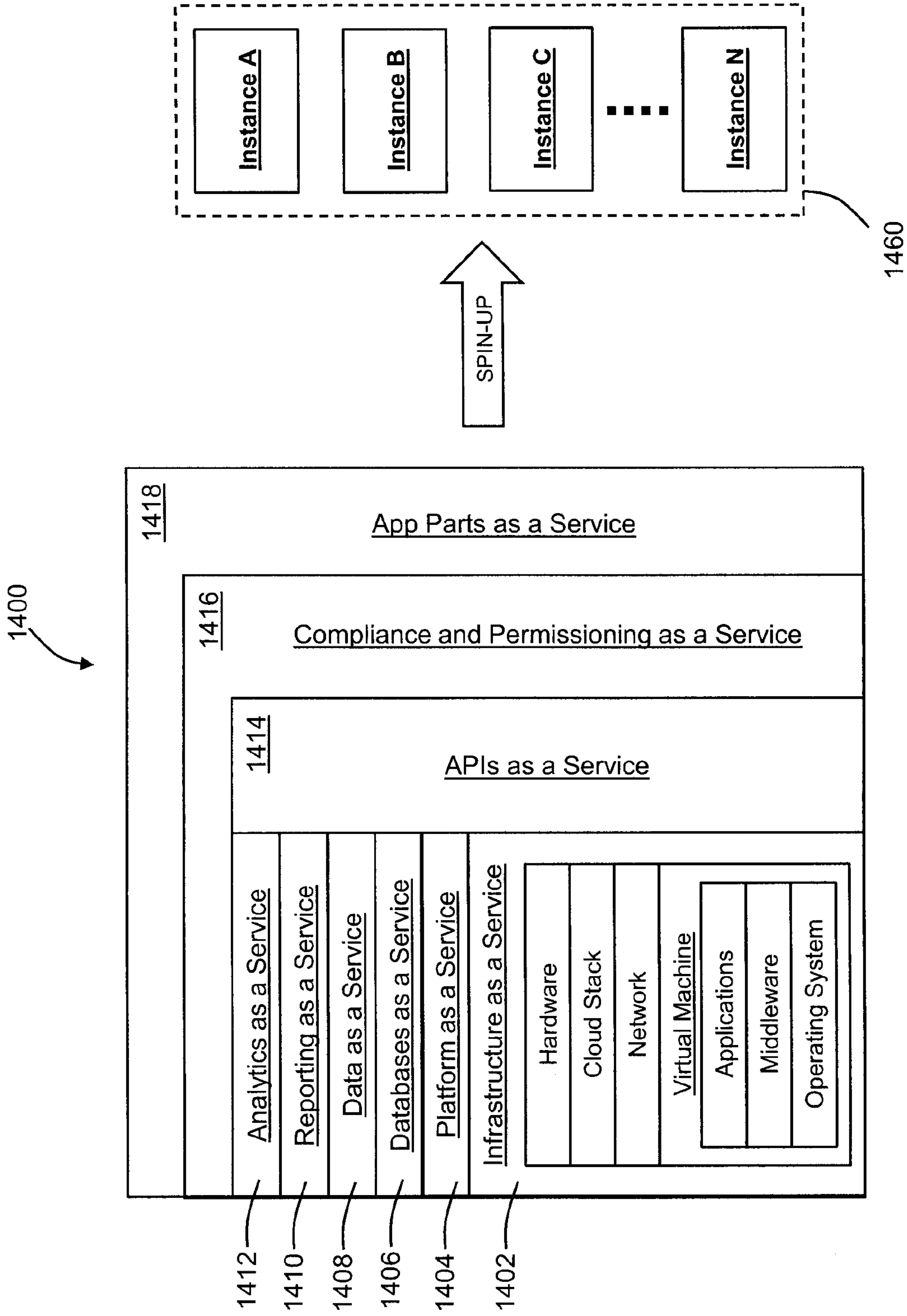


FIG. 14

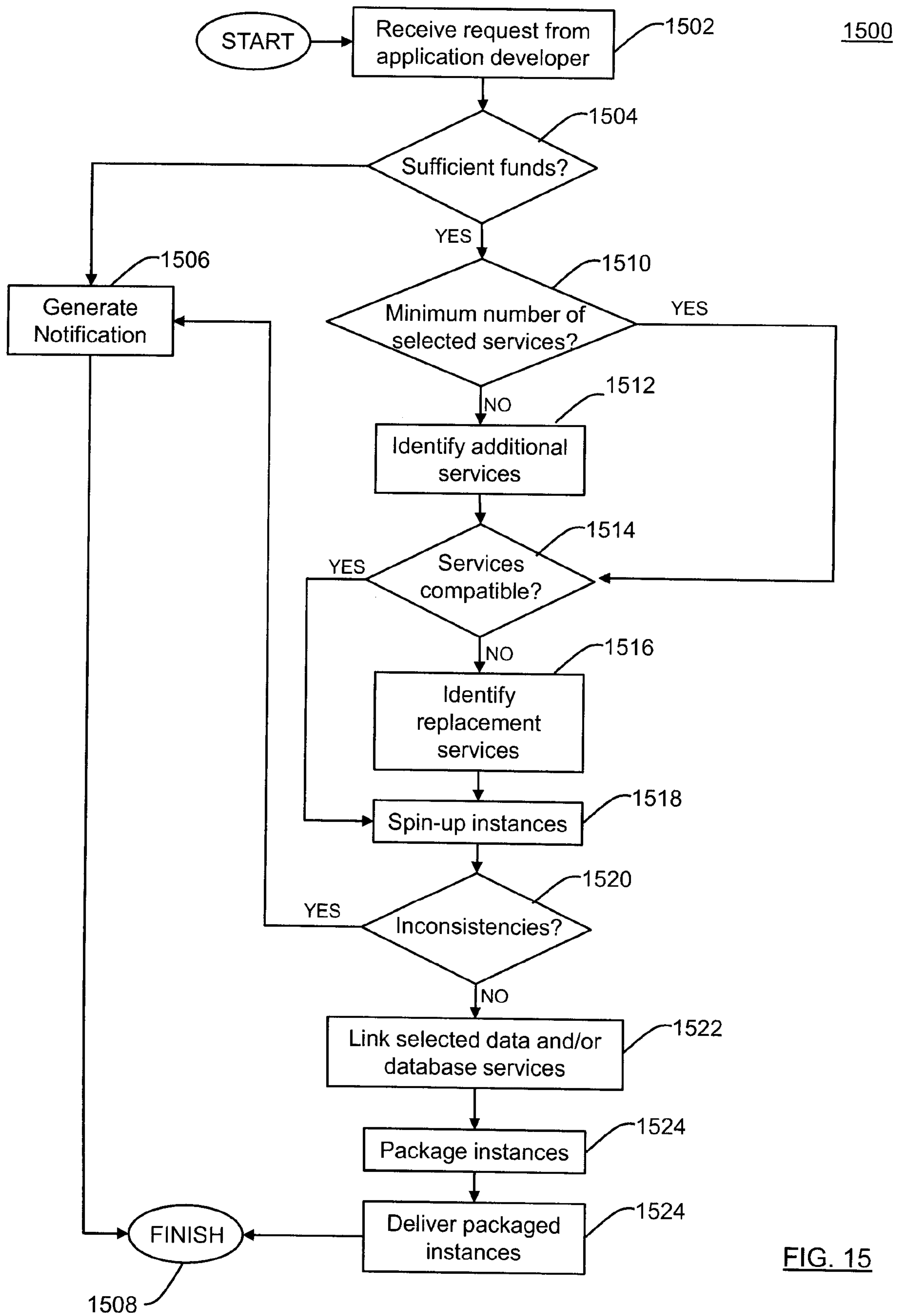


FIG. 15

