

(51) International Patent Classification:
G06Q 20/00 (2012.01)(21) International Application Number:
PCT/US2011/065305(22) International Filing Date:
15 December 2011 (15.12.2011)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:

61/423,588	15 December 2010 (15.12.2010)	US
61/431,818	11 January 2011 (11.01.2011)	US
61/432,031	12 January 2011 (12.01.2011)	US
61/432,583	13 January 2011 (13.01.2011)	US
61/466,927	23 March 2011 (23.03.2011)	US
61/467,302	24 March 2011 (24.03.2011)	US

(71) Applicant (for all designated States except US): VISA INTERNATIONAL SERVICE ASSOCIATION [US/US]; P.O. Box 8999, San Francisco, California 94128 (US).

(71) Applicant (for US only): HUA, Julian [US/US]; 338 Spear Street, Unit 9e, San Francisco, California 94105 (US).

(72) Inventor; and

(75) Inventor/Applicant (for US only): KATZIN, Edward [US/US]; 1126 Masonic Avenue, San Francisco, California 94117 (US).

(74) Agent: HANCHUK, Walter; Chadbourne & Parke LLP, 30 Rockefeller Plaza, New York, New York 10112 (US).

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

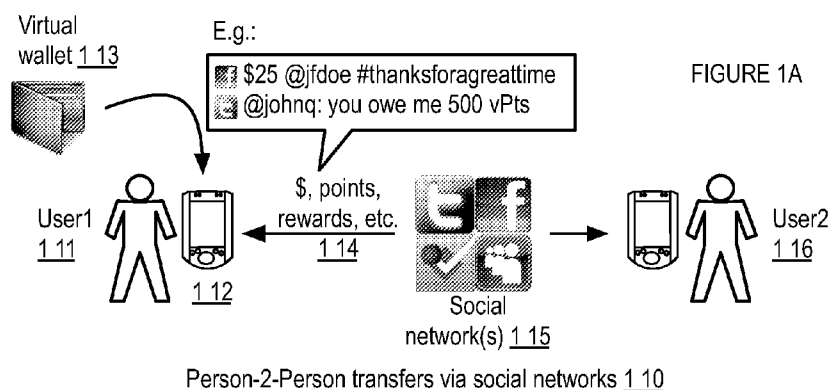
(84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU,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

Published:

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

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

(54) Title: SOCIAL MEDIA PAYMENT PLATFORM APPARATUSES, METHODS AND SYSTEMS



(57) Abstract: The SOCIAL MEDIA PAYMENT PLATFORM APPARATUSES, METHODS AND SYSTEMS ("SocialPay") transform message posts to social networks via SocialPay components into payment transaction receipts social merchant-consumer bridging offers. In one implementation, the SocialPay obtains a user social pay initiation trigger, and obtains user social data from a social networking site. The SocialPay identifies a social pay command within the user social data. In one implementation, the SocialPay, in identifying the social pay command within the user social data, parses the user social data, and extracts a social pay command string within the user social data. The SocialPay determines a payor identifier, a payee identifier, and a payment amount using the social pay command string. Based on the identified social pay command, the SocialPay initiates a funds payment transaction.

SOCIAL MEDIA PAYMENT PLATFORM APPARATUSES, METHODS AND SYSTEMS

1
2
3 **[0001]** This patent for letters patent disclosure document describes inventive
4 aspects that include various novel innovations (hereinafter “disclosure”) and contains
5 material that is subject to copyright, mask work, and/or other intellectual property
6 protection. The respective owners of such intellectual property have no objection to the
7 facsimile reproduction of the disclosure by anyone as it appears in published Patent
8 Office file/records, but otherwise reserve all rights.

PRIORITY CLAIM

9
10 **[0002]** This application claims priority under 35 U.S.C. § 119 to: United States
11 provisional patent application serial no. 61/423,588 filed December 15, 2010, entitled
12 “APPARATUSES, METHODS AND SYSTEMS FOR SECURE OFFERS, COMMERCE
13 AND SERVICES ON SOCIAL NETWORKS,” attorney docket no. P-41928PRV|20270-
14 087PV1; United States provisional patent application serial no. 61/431,818 filed January
15 11, 2011, entitled “APPARATUSES, METHODS AND SYSTEMS FOR A SOCIAL MEDIA
16 PAYMENT PLATFORM,” attorney docket no. P-41929PRV|20270-087PV2; United
17 States provisional patent application serial no. 61/432,031 filed January 12, 2011,
18 entitled “APPARATUSES, METHODS AND SYSTEMS FOR A SOCIAL MEDIA
19 PAYMENT PLATFORM,” attorney docket no. P-41930PRV|20270-087PV3; United
20 States provisional patent application serial no. 61/432,583 filed January 13, 2011,
21 entitled “APPARATUSES, METHODS AND SYSTEMS FOR A SOCIAL MEDIA

1 PAYMENT PLATFORM,” attorney docket no. P-41979PRV|20270-087PV4; United
2 States provisional patent application serial no. 61/466,927 filed March 23, 2011, entitled
3 “APPARATUSES, METHODS AND SYSTEMS FOR A SOCIAL MEDIA PAYMENT
4 PLATFORM,” attorney docket no. P-42156PRV|20270-087PV5; and United States
5 provisional patent application serial no. 61/467,302 filed March 24, 2011, entitled
6 “APPARATUSES, METHODS AND SYSTEMS FOR A SOCIAL MEDIA PAYMENT
7 PLATFORM,” attorney docket no. P-42161PRV|20270-087PV6. The entire contents of
8 the aforementioned applications are expressly incorporated by reference herein.

9 FIELD

10 **[0003]** The present innovations generally address apparatuses, methods, and
11 systems for e-commerce, and more particularly, include SOCIAL MEDIA PAYMENT
12 PLATFORM APPARATUSES, METHODS AND SYSTEMS (“SocialPay”).

13 BACKGROUND

14 **[0004]** Consumer transactions typically require a customer to select a product
15 from a store shelf or website, and then to check out at a checkout counter or webpage.
16 Product information is typically selected from a webpage catalog or entered into a point-
17 of-sale terminal device, or the information is automatically entered by scanning an item
18 barcode with an integrated barcode scanner, and the customer is usually provided with a
19 number of payment options, such as cash, check, credit card or debit card. Once
20 payment is made and approved, the point-of-sale terminal memorializes the transaction

1 in the merchant's computer system, and a receipt is generated indicating the satisfactory
2 consummation of the transaction.

3 BRIEF DESCRIPTION OF THE DRAWINGS

4 **[0005]** The accompanying appendices and/or drawings illustrate various non-
5 limiting, example, inventive aspects in accordance with the present disclosure:

6 **[0006]** FIGURES 1A-B show block diagrams illustrating example aspects of
7 payment transactions via social networks in some embodiments of the SocialPay;

8 **[0007]** FIGURE 2 shows a data flow diagram illustrating an example social pay
9 enrollment procedure in some embodiments of the SocialPay;

10 **[0008]** FIGURE 3 shows a logic flow diagram illustrating example aspects of
11 social pay enrollment in some embodiments of the SocialPay, e.g., a Social Pay
12 Enrollment (“SPE”) component 300;

13 **[0009]** FIGURES 4A-C show data flow diagrams illustrating an example social
14 payment triggering procedure in some embodiments of the SocialPay;

15 **[0010]** FIGURES 5A-C show logic flow diagrams illustrating example aspects of
16 social payment triggering in some embodiments of the SocialPay, e.g., a Social Payment
17 Triggering (“SPT”) component 500;

18 **[0011]** FIGURES 6A-B show logic flow diagrams illustrating example aspects of
19 implementing wallet security and settings in some embodiments of the SocialPay, e.g., a
20 Something (“WSS”) component 600;

1 **[0012]** FIGURE 7 shows a data flow diagram illustrating an example social
2 merchant consumer bridging procedure in some embodiments of the SocialPay;

3 **[0013]** FIGURE 8 shows a logic flow diagram illustrating example aspects of
4 social merchant consumer bridging in some embodiments of the SocialPay, e.g., a Social
5 Merchant Consumer Bridging (“SMCB”) component 800;

6 **[0014]** FIGURE 9 shows a logic flow diagram illustrating example aspects of
7 transaction data aggregation in some embodiments of the SocialPay, e.g., a Transaction
8 Data Aggregation (“TDA”) component 900;

9 **[0015]** FIGURE 10 shows a logic flow diagram illustrating example aspects of
10 transaction-based offer generation in some embodiments of the SocialPay, e.g., a
11 Transaction-Based Offer Generation (“TBOG”) component 1000;

12 **[0016]** FIGURE 11 shows a data flow diagram illustrating an example user
13 purchase checkout procedure in some embodiments of the SocialPay;

14 **[0017]** FIGURE 12 shows a logic flow diagram illustrating example aspects of a
15 user purchase checkout in some embodiments of the SocialPay, e.g., a User Purchase
16 Checkout (“UPC”) component 1200;

17 **[0018]** FIGURES 13A-B show data flow diagrams illustrating an example
18 purchase transaction authorization procedure in some embodiments of the SocialPay;

19 **[0019]** FIGURES 14A-B show logic flow diagrams illustrating example aspects of
20 purchase transaction authorization in some embodiments of the SocialPay, e.g., a
21 Purchase Transaction Authorization (“PTA”) component 1400;

1 **[0020]** FIGURES 15A-B show data flow diagrams illustrating an example
2 purchase transaction clearance procedure in some embodiments of the SocialPay;

3 **[0021]** FIGURES 16A-B show logic flow diagrams illustrating example aspects of
4 purchase transaction clearance in some embodiments of the SocialPay, e.g., a Purchase
5 Transaction Clearance (“PTC”) component 1600;

6 **[0022]** FIGURE 17 shows a user interface diagram illustrating an overview of
7 example features of virtual wallet applications in some embodiments of the SocialPay;

8 **[0023]** FIGURES 18A-G show user interface diagrams illustrating example
9 features of virtual wallet applications in a shopping mode, in some embodiments of the
10 SocialPay;

11 **[0024]** FIGURES 19A-F show user interface diagrams illustrating example
12 features of virtual wallet applications in a payment mode, in some embodiments of the
13 SocialPay;

14 **[0025]** FIGURE 20 shows a user interface diagram illustrating example features
15 of virtual wallet applications, in a history mode, in some embodiments of the SocialPay;

16 **[0026]** FIGURES 21A-E show user interface diagrams illustrating example
17 features of virtual wallet applications in a snap mode, in some embodiments of the
18 SocialPay;

19 **[0027]** FIGURE 22 shows a user interface diagram illustrating example features
20 of virtual wallet applications, in an offers mode, in some embodiments of the SocialPay;

1 **[0028]** FIGURES 23A-B show user interface diagrams illustrating example
2 features of virtual wallet applications, in a security and privacy mode, in some
3 embodiments of the SocialPay; and

4 **[0029]** FIGURE 24 shows a block diagram illustrating embodiments of a
5 SocialPay controller.

6 **[0030]** The leading number of each reference number within the drawings
7 indicates the FIGURE in which that reference number is introduced and/or detailed. As
8 such, a detailed discussion of reference number 101 would be found and/or introduced
9 in FIGURE 1. Reference number 201 is introduced in FIGURE 2, etc.

DETAILED DESCRIPTION

SOCIAL MEDIA PAYMENT PLATFORM (SocialPay)

[0031] The SOCIAL MEDIA PAYMENT PLATFORM APPARATUSES, METHODS AND SYSTEMS (hereinafter “SocialPay”) transform message posts to social networks, via SocialPay components, into payment transaction receipts social merchant-consumer bridging offers. FIGURES 1A-B show block diagrams illustrating example aspects of payment transactions via social networks in some embodiments of the SocialPay. In some embodiments, the SocialPay may facilitate per-2-person transfers of money via social networks. For example, a user (user1 111) may wish to provide funds (dollars, rewards, points, miles, etc. 114) to another user (user2 116). The user may utilize a virtual wallet to provide a source of funds. In some embodiments, the user may utilize a device 112 (such as a smartphone, mobile device, laptop computer, desktop computer, and/or the like) to send a social post message via the social network 115. In some embodiments, the social post message may include information on an amount of funds to be transferred and an identity of another user to whom the funds should be transferred. The SocialPay may intercept the message before it is sent to the social networking service, or it may obtain the message from the social networking service. Using the social post message, the SocialPay may resolve the identities of a payor and payee in the transaction. The SocialPay may identify accounts of the payor and payee to/from which funds need be credited or debited, and an amount of credit/debit to apply to each of the accounts. The SocialPay may, on the basis of resolving this information, execute a transaction to transfer funds from the payor to the payee. For

1 example, the SocialPay may allow a payor, by sending a tweet on Twitter™ such as “\$25
2 @jfdoe #ackpls” to transfer funds to a payee (user ID jfdoe), and request an
3 acknowledgement from SocialPay of receipt of funds. In another example, the SocialPay
4 may allow a potential payee to request funds from another user by sending a tweet on
5 Twitter™ such as “@johnq, you owe me 50000 Visa rewards points #id1234”; the
6 SocialPay may automatically provide an alert within a virtual wallet application of the
7 user with user ID johnq to provide the funds to the potential payee user. The user johnq
8 may respond by sending a tweet in response, referencing the id (#id1234), such as
9 “50000 vpts @jfdoe #id1234”; the SocialPay may transfer the funds and recognize
10 transaction request #id1234 as being fulfilled. In some embodiments, the SocialPay
11 may generate transaction/request ID numbers for the users to prevent coinciding
12 transaction/request ID numbers for different transaction/requests.

13 **[0032]** In some embodiments, the SocialPay may utilize one or more social
14 networking services (e.g., Facebook®, Twitter™, MySpace™, etc.). In some
15 embodiments, the SocialPay may allow users across different social networks to transact
16 with each other. For example, a user may make a request for payment on one social
17 network. As an example, a Twitter™ user may tweet “@johnq@facebook.com, you owe
18 me 500 vpts #ID7890”). The SocialPay may provide an alert to the user with ID
19 johnq@facebook.com either via the other social networking or via the user’s virtual
20 wallet. In response, the payee may social post to Facebook® a message “@jfdoe: here’s
21 your 500 vpts #ID7890”, and the SocialPay may facilitate the payment transaction and
22 provide a receipt/acknowledgment to the two users on their respective social networks
23 or virtual wallets.

1 **[0033]** In some embodiments, the SocialPay may facilitate transfers of funds to
2 more than one payee by a payor via a single social post message. In some embodiments,
3 the SocialPay may facilitate use of more than one source of funds of a payee to fund
4 payment of funds to one or more payors via a single post message. For example, the
5 SocialPay may utilize default settings or customized rules, stored within a virtual wallet
6 of a payor, to determine which funding sources to utilize to fund a payment transaction
7 to one or more payees via a social post message.

8 **[0034]** In some implementations, the SocialPay may facilitate merchants to make
9 offers of products and/or services to consumers via social networks 120. For example, a
10 merchant 126 may sign up to participate in the SocialPay. The SocialPay may aggregate
11 transactions of a user, and determine any products or services that may relevant for
12 offering to the user. The SocialPay may determine whether any participating merchants
13 are available to provide the products or services for the users. If so, the SocialPay may
14 provide social post messages via a social network 125 on behalf of the merchants (or,
15 alternatively, inform the merchants who may then send social post messages to the
16 users) providing the offers 124a to the user 121. An example of an offer to the followers
17 of a merchant on may be “@amazon offers the new Kindle™ at only \$149.99 – click here
18 to buy.” In such an example, the offer posted on the social networking site may have a
19 link embedded (e.g., “here”) that users can click to make the purchase (which may be
20 automatically performed with one-click if they are currently logged into their virtual
21 wallet accounts 123). Another example of a merchant offer may be “@amazon offers the
22 new Kindle™ at only \$149.99 – reply with #offerID123456 to buy.” In such an example,
23 the hash tag value serves as an identifier of the offer, which the users can reference
24 when making their purchase via their social post messages (e.g., “buy from @amazon

1 #offerID123456”). In some embodiments, merchants may provide two or more offers
2 via a single social post message. In some embodiments, users may reference two or
3 more offers in the same social post message.

4 **[0035]** In some implementations, users and/or merchants may utilize alternate
5 messaging modes. For example, a user may be able to utilize electronic mail, SMS
6 messages, phone calls, etc., to communicate with the SocialPay and the social networks.
7 For example, a merchant may provide a social post message offer such as ““@amazon
8 offers the new Kindle™ at only \$149.99 – text #offerID123456 to buy”. When a user
9 utilize a mobile phone to send a text message to redeem the offer, the SocialPay may
10 utilize a user profile of the user store on the social networking service to identify an
11 identifying attribute of the user’s mobile phone (e.g., a phone number), using which the
12 SocialPay may correlate the text message to a particular user. Thus, the SocialPay may
13 be able to process a transaction with the merchant on behalf of the user, using user
14 information from the user’s virtual wallet. In some embodiments where a social
15 network is incapable of handling a particular mode of communication, the SocialPay
16 may serve as an intermediary translator to convert the message to a form that can be
17 utilized by the social network.

18 **[0036]** FIGURE 2 shows a data flow diagram illustrating an example social pay
19 enrollment procedure in some embodiments of the SocialPay. In some embodiments, a
20 user, e.g., 201, may desire to enroll in SocialPay. The user may communicate with a
21 SocialPay server, e.g., 203a, via a client such as, but not limited to: a personal computer,
22 mobile device, television, point-of-sale terminal, kiosk, ATM, and/or the like (e.g., 202).
23 For example, the user may provide user input, e.g., social pay enrollment input 211, into

1 the client indicating the user's desire to enroll in social network authenticated purchase
2 payment. In various implementations, the user input may include, but not be limited to:
3 a single tap (e.g., a one-tap mobile app purchasing embodiment) of a touchscreen
4 interface, keyboard entry, card swipe, activating a RFID/NFC enabled hardware device
5 (e.g., electronic card having multiple accounts, smartphone, tablet, etc.) within the user
6 device, mouse clicks, depressing buttons on a joystick/game console, voice commands,
7 single/multi-touch gestures on a touch-sensitive interface, touching user interface
8 elements on a touch-sensitive display, and/or the like.

9 **[0037]** In some implementations, using the user's input, the client may generate a
10 social pay enrollment request, e.g., 212, and provide the enrollment request to the
11 SocialPay server 203a. For example, the client may provide a (Secure) Hypertext
12 Transfer Protocol ("HTTP(S)") POST message including data formatted according to the
13 eXtensible Markup Language ("XML"). Below is an example HTTP(S) POST message
14 including an XML-formatted enrollment request for the social pay server:

```
15 POST /enroll.php HTTP/1.1
16 Host: www.socialpay.com
17 Content-Type: Application/XML
18 Content-Length: 484
19 <?XML version = "1.0" encoding = "UTF-8"?>
20 <enrollment_request>
21   <request_ID>4NFU4RG94</request_ID>
22   <timestamp>2011-02-22 15:22:43</timestamp>
23   <user_ID>john.q.public@facebook.com</user_ID>
24   <wallet_account_ID>7865493028712345</wallet_account_ID>
25   <client_details>
26     <client_IP>192.168.23.126</client_IP>
27     <client_type>smartphone</client_type>
28     <client_model>HTC Hero</client_model>
29     <OS>Android 2.2</OS>
30     <app_installed_flag>true</app_installed_flag>
31   </client_details>
32 </enrollment_request>
33
34
```

35 **[0038]** In some embodiments, the social pay server may obtain the enrollment
36 request from the client, and extract the user's payment detail (e.g., XML data) from the

1 enrollment request. For example, the social pay server may utilize a parser such as the
 2 example parsers described below in the discussion with reference to FIGURE 24. In
 3 some implementations, the social pay server may query, e.g., 213, a social pay database,
 4 e.g., 203b, to obtain a social network request template, e.g., 214, to process the
 5 enrollment request. The social network request template may include instructions,
 6 data, login URL, login API call template and/or the like for facilitating social network
 7 authentication. For example, the database may be a relational database responsive to
 8 Structured Query Language ("SQL") commands. The merchant server may execute a
 9 hypertext preprocessor ("PHP") script including SQL commands to query the database
 10 for product data. An example PHP/SQL command listing, illustrating substantive
 11 aspects of querying the database, e.g., 214-215, is provided below:

```

12  <?PHP
13  header('Content-Type: text/plain');
14  mysql_connect("254.93.179.112",$DBserver,$password); // access database server
15  mysql_select_db("SOCIALPAY.SQL"); // select database table to search
16  //create query
17  $query = "SELECT template FROM EnrollTable WHERE network LIKE '%" $socialnet";
18  $result = mysql_query($query); // perform the search query
19  mysql_close("SOCIALAUTH.SQL"); // close database access
20  ?>
21
22

```

23 **[0039]** In some implementations, the social pay server may redirect the client to a
 24 social network server, e.g., 204a, by providing a HTTP(S) REDIRECT 300 message,
 25 similar to the example below:

```

26  HTTP/1.1 300 Multiple Choices
27  Location:
28      https://www.facebook.com/dialog/oauth?client_id=snpa_app_ID&redirect_uri=
29      www.paynetwork.com/enroll.php
30  <html>
31      <head><title>300 Multiple Choices</title></head>
32      <body><h1>Multiple Choices</h1></body>
33  </html>
34
35

```

1 **[0040]** In some implementations, the social pay server may provide information
 2 extracted from the social pay enrollment request to the social network server as part of a
 3 user authentication/ social pay app enroll request, e.g., 215. For example, the social pay
 4 server may provide a HTTP(S) POST message to the social network server, similar to the
 5 example below:

```

6 POST /authenticate_enroll.php HTTP/1.1
7 Host: www.socialnet.com
8 Content-Type: Application/XML
9 Content-Length: 484
10 <?XML version = "1.0" encoding = "UTF-8"?>
11 <enrollment_request>
12   <request_ID>4NFU4RG94</request_ID>
13   <timestamp>2011-02-22 15:22:43</timestamp>
14   <user_ID>john.q.public@facebook.com</user_ID>
15   <wallet_account_ID>7865493028712345</wallet_account_ID>
16   <client_details>
17     <client_IP>192.168.23.126</client_IP>
18     <client_type>smartphone</client_type>
19     <client_model>HTC Hero</client_model>
20     <OS>Android 2.2</OS>
21     <app_installed_flag>true</app_installed_flag>
22   </client_details>
23 </enrollment_request>
24
25
```

26 **[0041]** In some implementations, the social network server may provide a social
 27 network login request, e.g., 216, to the client. For example, the social network server
 28 may provide a HTML input form to the client. The client may display, e.g., 217, the login
 29 form for the user. In some implementations, the user may provide login input into the
 30 client, e.g., 218, and the client may generate a social network login response, e.g., 219,
 31 for the social network server. In some implementations, the social network server may
 32 authenticate the login credentials of the user, and upon doing so, update the profile of
 33 the user to indicate the user's enrollment in the social pay system. For example, in a
 34 social networking service such as Facebook®, the social network server may provide
 35 permission to a social pay third-party developer app to access the user's information
 36 stored within the social network. In some embodiments, such enrollment may allow a

1 virtual wallet application installed on a user device of to access the user's social profile
2 information stored within the social network. Upon authentication, the social network
3 server may generate an updated data record for the user, e.g., 220, and provide an
4 enrollment notification, e.g., 221, to the social pay server. For example, the social
5 network server may provide a HTTP(S) POST message similar to the example below:

```
6 POST /enrollnotification.php HTTP/1.1
7 Host: www.socialpay.com
8 Content-Type: Application/XML
9 Content-Length: 1306
10 <?XML version = "1.0" encoding = "UTF-8"?>
11 <enroll_notification>
12     <request_ID>4NFU4RG94</request_ID>
13     <timestamp>2011-02-22 15:22:43</timestamp>
14     <result>enrolled</result>
15 </enroll_notification>
16
17
```

18 **[0042]** Upon receiving notification of enrollment from the social network server,
19 the social pay server may generate, e.g., 222, a user enrollment data record, and store
20 the enrollment data record in a social pay database, e.g., 223, to complete enrollment.
21 In some implementations, the enrollment data record may include the information from
22 the enrollment notification 221.

23 **[0043]** FIGURE 3 shows a logic flow diagram illustrating example aspects of
24 social pay enrollment in some embodiments of the SocialPay, e.g., a Social Pay
25 Enrollment ("SPE") component 300. In some embodiments, a user may desire to enroll
26 in SocialPay. The user may provide user input, e.g., social pay enrollment input 301,
27 into the client indicating the user's desire to enroll in social network authenticated
28 purchase payment. In some implementations, using the user's input, the client may
29 generate a social pay enrollment request, e.g., 302, and provide the enrollment request
30 to the social pay server. In some embodiments, the social pay server may obtain the
31 enrollment request from the client, and extract the user's payment detail (e.g., XML

1 data) from the enrollment request. For example, the social pay server may utilize a
2 parser such as the example parsers described below in the discussion with reference to
3 FIGURE 24. In some implementations, the social pay server may query, e.g., 303, a
4 social pay database to obtain a social network request template to process the
5 enrollment request. The social network request template may include instructions,
6 data, login URL, login API call template and/or the like for facilitating social network
7 authentication. In some implementations, the social pay server may redirect the client to
8 a social network server. In some implementations, the social pay server may provide
9 information extracted from the social pay enrollment request to the social network
10 server as part of a user authentication/ social pay app enroll request, e.g., 305. In some
11 implementations, the social network server may provide a social network login request,
12 e.g., 306, to the client. For example, the social network server may provide a HTML
13 input form to the client. The client may display, e.g., 307, the login form for the user. In
14 some implementations, the user may provide login input into the client, e.g., 308, and
15 the client may generate a social network login response, e.g., 309, for the social network
16 server. In some implementations, the social network server may authenticate the login
17 credentials of the user, and upon doing so, update the profile of the user to indicate the
18 user's enrollment in the social pay system. For example, in a social networking service
19 such as Facebook®, the social network server may provide permission to a social pay
20 third-party developer app to access the user's information stored within the social
21 network. In some embodiments, such enrollment may allow a virtual wallet application
22 installed on a user device of to access the user's social profile information stored within
23 the social network. Upon authentication, the social network server may generate an
24 updated data record for the user, e.g., 310-311, and provide an enrollment notification,

1 e.g., 312 to the social pay server. Upon receiving notification of enrollment from the
2 social network server, the social pay server may generate, e.g., 313, a user enrollment
3 data record, and store the enrollment data record in a social pay database, e.g., 314, to
4 complete enrollment. In some implementations, the enrollment data record may
5 include the information from the enrollment notification.

6 **[0044]** FIGURES 4A-C show data flow diagrams illustrating an example social
7 payment triggering procedure in some embodiments of the SocialPay. With reference to
8 FIGURE 4A, in some embodiments, a user, e.g., user1 401a, may desire to provide or
9 request funds from another (e.g., a user, a participating merchant, etc.). The user may
10 communicate with a social network server, e.g., 403a, via a client (client1 402a) such as,
11 but not limited to: a personal computer, mobile device, television, point-of-sale
12 terminal, kiosk, ATM, and/or the like. For example, the user may provide social
13 payment input 411, into the client indicating the user's desire to provide or request
14 funds from another. In various embodiments, the user input may include, but not be
15 limited to: a single tap (e.g., a one-tap mobile app purchasing embodiment) of a
16 touchscreen interface, keyboard entry, card swipe, activating a RFID/NFC enabled
17 hardware device (e.g., electronic card having multiple accounts, smartphone, tablet,
18 etc.) within the user device, mouse clicks, depressing buttons on a joystick/game
19 console, voice commands, single/multi-touch gestures on a touch-sensitive interface,
20 touching user interface elements on a touch-sensitive display, and/or the like. In
21 response, the client may provide a social message post request 412 to the social network
22 server. In some implementations, a virtual wallet application executing on the client
23 may provide the user with an easy-to-use interface to generate and send the social
24 message post request. In alternate implementations, the user may utilize other

1 applications to provide the social message post request. For example, the client may
 2 provide a social message post request to the social network server as a HTTP(S) POST
 3 message including XML-formatted data. An example listing of a social message post
 4 request 412, substantially in the form of a HTTP(S) POST message including XML-
 5 formatted data, is provided below:

```

6  POST /socialpost.php HTTP/1.1
7  Host: www.socialnetwork.com
8  Content-Type: Application/XML
9  Content-Length: 310
10 <?XML version = "1.0" encoding = "UTF-8"?>
11 <message_post_request>
12   <request_ID>value</request_ID>
13   <timestamp>2011-02-02 03:04:05</timestamp>
14   <sender_id>jfdoe@facebook.com</sender_id>
15   <receiver_id>johnqp@facebook.com</receiver_id>
16   <message>$25 @johnqp #thanksforagreattimelastnite</message>
17 </message_post_request>
18
19

```

20 **[0045]** The user may have signed up for numerous wallets. The message 412 may
 21 be sent be sent from the user 402a to a second user via the social network 404a. In this
 22 example, user1 401a sent \$25 to johnq with a message "#thanksforagreattime" 412b.
 23 SocialPay may later append various messages and/or send additional various messages
 24 that will appear to the target user to have been sent by user1 401a. As an example, here
 25 the SocialPay determined (determination and parsing as described further below, e.g.,
 26 Figures 6, 9 and 10 et al.) that user2 does not have a wallet into which they may redeem
 27 payment. As such SocialPay upon parsing and determination may append a message to
 28 allow the receiving user to sign up for a wallet and thus obtain the payment from user1;
 29 in this example, social pay appended "signup at visa.com/wallet to redeem this
 30 payment." It should be noted that various wallets may be employed and/or offered; for
 31 example, a social network may itself offer a wallet and as such another message of
 32 "signup at twitter.com/wallet to redeem this payment" may be appended. In another

1 embodiment, social pay itself may host an e-wallet and as such the following message
2 may be appended "signup at socialpay.com/wallet to redeem this payment." In one
3 example, the SocialPay server may use login credentials provided by a user to
4 automatically simultaneously and permanently be logged in reading every social
5 message/post entered by the user from other client programs and in addition received
6 messages that are sent to the user by other users. As such, SocialPay may parse all
7 transactions send by the user and/or received messages that were directed to the user
8 and determine which messages are directed to make person to person payments. In
9 another embodiment, this type of interception parsing may be employed at the social
10 network servers instead of at the SocialPay servers. In yet another embodiment, both
11 the SocialPay server and the social network server may do this type of interception
12 parsing, the details of which are described further below (e.g., Figures 6, 9 and 10 et al.).
13 It should be noted when this type of interception parsing is ongoing, which will be all
14 the time unless a user specifically requests the cessation of such interception parsing,
15 when the SocialPay server and/or other servers intercept messages and parse them and
16 determine, e.g., they are triggers for payments, those servers may go on to process the
17 parsed message triggering payment and other activities. For example, if the target user
18 does not have an e-wallet account, upon look up and determination by the server, then
19 the server may send a message in addition to the social message POST request 412,
20 where the additional message will provide details for where the target user may sign up
21 and create an e-wallet account and redeem payment provided to them by another
22 user/system. If SocialPay, instead, determines that the target user is already enrolled in
23 an e-wallet, it may initiate and then facilitate the transfer of payment from the first user
24 to the target user's account without further messaging or interaction (e.g., it may also

1 require the target user to accept such payments, in which case it can send a second
 2 message to the target user asking them to reply to social pay saying yes to effectuate
 3 payment before such funds are delivered to the target user's e-wallet account).

4 **[0046]** In another embodiment, a Social Pay post message may be for an item. In
 5 such a sense, it may become a social gift message. For example, the message may be
 6 substantially in the form of a HTTP(S) POST message including XML-formatted data, is
 7 provided below:

```

8 POST /socialpost.php HTTP/1.1
9 Host: www.socialnetwork.com
10 Content-Type: Application/XML
11 Content-Length: 310
12 <?XML version = "1.0" encoding = "UTF-8"?>
13 <message_post_request>
14     <message_link_label>iPad</message_link_label >
15
16     <message_link_label_address>http:store.apple.com/?itemquery?ipad_32GB
17 _WiFi_white</message_link_label_address>
18     <request_ID>value</request_ID>
19     <store_login>jfdoe@mac.com</store_login>
20     <store_pass>abc123</store_pass>
21     <store_wallet_account>Apple Store ID123</store_wallet_account>
22 <timestamp>2011-02-02 03:04:05</timestamp>
23 <sender_id>jfdoe@facebook.com</sender_id>
24 <receiver_id>johnqp@facebook.com</receiver_id>
25 <message>iPad @johnqp #christmasgift</message>
26 </message_post_request>

```

27 **[0047]** In such an example, the user may post a link to an item (e.g., drag and
 28 drop a link for a product into their social messaging application which will translate
 29 and/or include both the link label (e.g., iPad) and the address for the item (e.g.,
 30 http:store.apple.com/?itemquery?ipad_32GB_WiFi_white) identifying the product
 31 skew at the merchant. Social Pay may then see if the user's wallet has an account with
 32 that merchant and provide login credentials to affect a purchase through the merchant
 33 and identify shipping addresses from the target user. In another embodiment, the
 34 gifting user may be prompted for login information, which may then be passed along to
 35 Social Pay to affect the purchase.

1 **[0048]** In some embodiments, the social network server 404a may query its social
 2 network database for a social graph of the user, e.g., 413. For example, the social
 3 network server may issue PHP/SQL commands to query a database table (such as
 4 FIGURE 24, Social Graph 2419p) for social graph data associated with the user. An
 5 example user social graph query 413, substantially in the form of PHP/SQL commands,
 6 is provided below:

```

7  <?PHP
8  header('Content-Type: text/plain');
9  mysql_connect("254.93.179.112",$DBserver,$password); // access database server
10 mysql_select_db("SocialPay_DB.SQL"); // select database table to search
11 //create query
12 $query = "SELECT friend_name friend_type friend_weight message_params_list
13          messaging_restrictions FROM SocialGraphTable WHERE user LIKE '%" $user_id";
14 $result = mysql_query($query); // perform the search query
15 mysql_close("SocialPay_DB.SQL"); // close database access
16 ?>
17
18

```

19 **[0049]** In some embodiments, the social network database may provide the
 20 requested social graph data in response, e.g., 414. Using the social graph data, the social
 21 network server may generate message(s) as appropriate for the user and/or members of
 22 the user's social graph, e.g., 415, and store the messages 416 for the user and/or social
 23 graph members.

24 **[0050]** With reference to FIGURE 4B, in some embodiments, such posting of
 25 social messages may trigger SocialPay actions. For example, a social pay server 403a
 26 may be triggered to scan the social data for pay commands. In embodiments where
 27 every social post message originates from the virtual wallet application of a user, the
 28 SocialPay may optionally obtain the pay commands from the virtual wallet applications,
 29 and skip scanning the social networks for pay commands associated with the user. In
 30 embodiments where a user is allowed to issue pay commands from any device (even
 31 those not linked to the user's virtual wallet), the SocialPay may periodically, or even

1 continuously scan the social networks for pay commands, e.g., 421. In embodiments
 2 where the SocialPay scans the social networks, the social pay server may query a social
 3 pay database for a profile of the user. For example, the social pay server may request a
 4 user ID and password for the social networks that the user provided to the social pay
 5 server during the enrollment phase (see, e.g., FIGURES 2-3). For example, the
 6 SocialPay server may issue PHP/SQL commands to query a database table (such as
 7 FIGURE 24, Users 2419a) for user profile data. An example user profile data query 422,
 8 substantially in the form of PHP/SQL commands, is provided below:

```

9  <?PHP
10 header('Content-Type: text/plain');
11 mysql_connect("254.93.179.112",$DBserver,$password); // access database server
12 mysql_select_db("SocialPay_DB.SQL"); // select database table to search
13 //create query
14 $query = "SELECT network_id network_name network_api user_login user_pass FROM
15     UsersTable WHERE userid LIKE '%" $user_id";
16 $result = mysql_query($query); // perform the search query
17 mysql_close("SocialPay_DB.SQL"); // close database access
18 ?>
19
20
```

21 **[0051]** In response, the social pay database may provide the requested
 22 information, e.g., 423. In some embodiments, the social pay server may provide a user
 23 social data request 424 to the social network server. An example listing of commands to
 24 issue a user social data request 424, substantially in the form of PHP commands, is
 25 provided below:

```

26 <?PHP
27 header('Content-Type: text/plain');
28
29 // Obtain user ID(s) of friends of the logged-in user
30 $friends =
31     json_decode(file_get_contents('https://graph.facebook.com/me/friends?access
32     token='.$cookie['oauth_access_token']), true);
33 $friend_ids = array_keys($friends);
34
35 // Obtain message feed associated with the profile of the logged-in user
36 $feed =
37     json_decode(file_get_contents('https://graph.facebook.com/me/feed?access_tok
38     en='.$cookie['oauth_access_token']), true);
39
40 // Obtain messages by the user's friends

```

```
1 $result = mysql_query('SELECT * FROM content WHERE uid IN ('
2 .implode($friend_ids, ',') . ')');
3 $friend_content = array();
4 while ($row = mysql_fetch_assoc($result))
5 $friend_content [] $row;
6 ?>
```

9 **[0052]** The user may have signed up for numerous wallets. The message 412, 424
10 may be sent be sent from the user 402a to a second user via the social network 404a. In
11 this example, user1 401a sent \$25 to johnq with a message "#thanksforagreattime"
12 412b. SocialPay may later append various messages and/or send additional various
13 messages, which will appear to the target user to have been sent by user1 401a. As an
14 example, here the SocialPay determined (determination and parsing as described
15 further below, e.g., Figures 6, 9 and 10 et al.) that user2 does not have a wallet into
16 which they may redeem payment. As such SocialPay upon parsing and determination
17 may append a message to allow the receiving user to sign up for a wallet and thus obtain
18 the payment from user1; in this example, SocialPay appended "signup at
19 visa.com/wallet to redeem this payment." It should be noted that various wallets may be
20 employed and/or offered; for example, a social network may itself offer a wallet and as
21 such another message of "signup at twitter.com/wallet to redeem this payment" may be
22 appended. In another embodiment, SocialPay itself may host an e-wallet and as such the
23 following message may be appended "signup at socialpay.com/wallet to redeem this
24 payment." In one example, the SocialPay server may use login credentials provided by a
25 user to automatically simultaneously and permanently be logged in reading every social
26 message/post entered by the user from other client programs and in addition received
27 messages that are sent to the user by other users. As such, SocialPay may parse all
28 transactions send by the user and/or received messages that were directed to the user
29 and determine which messages are directed to make person to person/entity payments.

1 In another embodiment, this type of interception parsing may be employed at the social
2 network servers instead of at the SocialPay servers. In yet another embodiment, both
3 the SocialPay server and the social network server may do this type of interception
4 parsing, the details of which are described further below (e.g., Figures 6, 9 and 10 et al.).
5 It should be noted when this type of interception parsing is ongoing, which will be all
6 the time unless a user specifically requests the cessation of such interception parsing,
7 when the SocialPay server and/or other servers intercept messages and parse them and
8 determine, e.g., they are triggers for payments, those servers may go on to process the
9 parsed message triggering payment and other activities. For example, if the target user
10 does not have an e-wallet account, upon look up and determination by the server, then
11 the server may send a message in addition to the social message POST request 412, 424,
12 where the additional message will provide details for where the target user may sign up
13 and create an e-wallet account and redeem payment provided to them by another
14 user/system. If SocialPay, instead, determines that the target user is already enrolled in
15 an e-wallet, it may initiate and then facilitate the transfer of payment from the first user
16 to the target user's account without further messaging or interaction (e.g., it may also
17 require the target user to accept such payments, in which case it can send a second
18 message to the target user asking them to reply to SocialPay saying yes to effectuate
19 payment before such funds are delivered to the target user's e-wallet account).

20 **[0053]** In some embodiments, the social network server may query, e.g., 426, it
21 social network database 404b for social data results falling within the scope of the
22 request. In response to the query, the database may provide social data, e.g., 427. The
23 social network server may return the social data obtained from the databases, e.g., 428,

1 to the social pay server. An example listing of user social data 428, substantially in the
2 form of JavaScript Object Notation (JSON)-formatted data, is provided below:

```
3
4  [ "data": [
5      {      "name": "Tabatha Orloff",
6              "id": "483722"},
7      {      "name": "Darren Kinnaman",
8              "id": "86S743"},
9      {      "name": "Sharron Jutras",
10             "id": "091274"}
11  ] }
```

14 **[0054]** In some embodiments, the social pay server may query the social pay
15 database for social pay rules, e.g., 429. For example, the social pay server may issue
16 PHP/SQL commands to query a database table (such as FIGURE 24, Social Pay Rules
17 2419q) for the social pay rules 430. An example pay rules query 429, substantially in
18 the form of PHP/SQL commands, is provided below:

```
19  <?PHP
20  header('Content-Type: text/plain');
21  mysql_connect("254.93.179.112",$DBserver,$password); // access database server
22  mysql_select_db("SocialPay_DB.SQL"); // select database table to search
23  //create query
24  $query = "SELECT rule_id rule_type rule_description rule_priority rule_source
25           FROM SocialPayRulesTable WHERE rule_type LIKE pay_rules";
26  $result = mysql_query($query); // perform the search query
27  mysql_close("SocialPay_DB.SQL"); // close database access
28  ?>
29
30
```

31 **[0055]** In some embodiments, the social pay server may process the user social
32 data using the social pay rules to identify pay commands, pay requests, merchant offers,
33 and/or like content of the user social data. In some embodiments, rules may be provided
34 by the SocialPay to ensure the privacy and security of the user's social data and virtual
35 wallet. As another example, the rules may include procedures to detect fraudulent
36 transaction attempts, and request user verification before proceeding, or cancel the
37 transaction request entirely. In some embodiments, the social pay server may utilize a

1 wallet security and settings component, such as the example WSS 600 component
2 described further below in the discussion with reference to FIGURES 6A-B.

3 **[0056]** With reference to FIGURE 4C, in some embodiments, the social pay server
4 may optionally determine that, based on processing of the rules, user verification is
5 needed to process a transaction indicated in a pay command. For example, if the rules
6 processing indicated that there is a probability of the pay command being an attempt at
7 a fraudulent transaction attempt, the social pay server may determine that the user
8 must be contacted for payment verification before the transaction can be processed. In
9 such scenarios, the social pay server may provide a pay command verification request
10 433 to the client, which the client may display, e.g., 434, to the user. For example, the
11 social pay server may provide a pay command verification request to the client 402a as a
12 HTTP(S) POST message including XML-formatted data. An example listing of a pay
13 command verification request 433, substantially in the form of a HTTP(S) POST
14 message including XML-formatted data, is provided below:

```
15 POST /verifyrequest.php HTTP/1.1
16 Host: www.client.com
17 Content-Type: Application/XML
18 Content-Length: 256
19 <?XML version = "1.0" encoding = "UTF-8"?>
20 <verify_request>
21   <transaction_ID>AE1234</transaction_ID>
22   <timestamp>2011-02-02 03:04:05</timestamp>
23   <amount>50000 vpts</amount>
24   <message_string>5000000 vpts @jfdoe #thx</message_string>
25 </verify_request>
26
27
```

28 **[0057]** In some embodiments, the user may provide a verification input 435 into
29 the client, which may provide a pay command verification response to the social pay
30 server. The social pay server may determine whether the payor verified payment,
31 whether payee information available is sufficient to process the transaction, and/or the

1 like. In scenarios where sufficient payee information is unavailable, the social pay server
2 may optionally provide a social post message 438 to a social networking service
3 associated with the potential payee requesting the payee to enroll in social pay service
4 (e.g., using the SPE 300 component described above in the discussion with reference to
5 FIGURES 2-3), which the social network server may post 439 for the payee. If all the
6 requirements are met for processing the transaction, the social pay server may generate
7 a unique transaction trigger associated with the triggering social post message, e.g., 437,
8 and store a transaction trigger ID, triggering social post message, etc., for recordkeeping
9 or analytics purposes, e.g., 440. The social pay server may provide the transaction
10 trigger to trigger a purchase transaction 441, e.g., via a purchase transaction
11 authorization such as the example PTA 1400 component described below in the
12 discussion with reference to FIGURES 13-14.

13 **[0058]** FIGURES 5A-C show logic flow diagrams illustrating example aspects of
14 social payment triggering in some embodiments of the SocialPay, e.g., a Social Payment
15 Triggering (“SPT”) component 500. With reference to FIGURE 5A, in some
16 embodiments, a user may desire to provide or request funds from another (e.g., a user, a
17 participating merchant, etc.). The user may communicate with a social network server
18 via a client. For example, the user may provide social payment input 501, into the client
19 indicating the user’s desire to provide or request funds from another. In response, the
20 client may generate and provide a social message post request 502 to the social network
21 server. In some implementations, a virtual wallet application executing on the client
22 may provide the user with an easy-to-use interface to generate and send the social
23 message post request. In alternate implementations, the user may utilize other
24 applications to provide the social message post request. In some embodiments, the

1 social network server may query its social network database for a social graph of the
2 user, e.g., 503. In response, the social network database may provide the requested
3 social graph data, e.g., 504. Using the social graph data, the social network server may
4 generate message(s) as appropriate for the user and/or members of the user's social
5 graph, e.g., 505, and store the messages 506 for the user and/or social graph members.

6 **[0059]** With reference to FIGURE 5B, in some embodiments, such posting of
7 social messages may trigger SocialPay actions. For example, a social pay server may be
8 triggered to scan the social data for pay commands, e.g., 507. In embodiments where
9 every social post message originates from the virtual wallet application of a user, the
10 SocialPay may optionally obtain the pay commands from the virtual wallet applications,
11 and skip scanning the social networks for pay commands associated with the user. In
12 embodiments where a user is allowed to issue pay commands from any device (even
13 those not linked to the user's virtual wallet), the SocialPay may periodically, or even
14 continuously scan the social networks for pay commands. In embodiments where the
15 SocialPay scans the social networks, the social pay server may query a social pay
16 database for a profile of the user, 508. For example, the social pay server may request a
17 user ID and password for the social networks that the user provided to the social pay
18 server during the enrollment phase (see, e.g., FIGURES 2-3). In response, the social pay
19 database may provide the requested information, e.g., 509. In some embodiments, the
20 social pay server may generate provide a user social data request 510 to the social
21 network server.

22 **[0060]** In some embodiments, the social network server may extract a user ID
23 from the user social data request, e.g., 511. The social network server may query, e.g.,

1 512, it social network database to determine whether the user is enrolled in SocialPay
2 with the social network (e.g., “did the user allow the SocialPay Facebook® app to access
3 user data?”). In response, the social network database may provide user enrollment
4 data relating to SocialPay. The social network server may determine whether the user is
5 enrolled, and thus whether the social pay server is authorized to access the user social
6 data, 514. If the social network server determines that the social pay server is not
7 authorized, 515, option “No,” it may generate a service denial message, 516, and provide
8 the message to the social pay server. If the social network server determines that the
9 social pay server is authorized to access the user social data, 515, option “Yes,” the social
10 network server may generate a user social data query 517, and provide it to the social
11 network database. In response, the social network database may provide the user social
12 data requested, 518. The social network server may provide the user social data 519 to
13 the social pay server.

14 **[0061]** In some embodiments, the social pay server may query the social pay
15 database for social pay rules, e.g., 520-521. In some embodiments, the social pay server
16 may process the user social data using the social pay rules to identify pay commands,
17 pay requests, merchant offers, and/or like content of the user social data, 522. In some
18 embodiments, rules may be provided by the SocialPay to ensure the privacy and security
19 of the user’s social data and virtual wallet. As another example, the rules may include
20 procedures to detect fraudulent transaction attempts, and request user verification
21 before proceeding, or cancel the transaction request entirely. In some embodiments, the
22 social pay server may utilize a wallet security and settings component, such as the
23 example WSS 600 component described further below in the discussion with reference
24 to FIGURES 6A-B.

1 **[0062]** With reference to FIGURE 5C, in some embodiments, the social pay server
2 may optionally determine that, based on processing of the rules, user verification is
3 needed to process a transaction indicated in a pay command, 523, option “Yes.” For
4 example, if the rules processing indicated that there is a probability of the pay command
5 being an attempt at a fraudulent transaction attempt, the social pay server may
6 determine that the user must be contacted for payment verification before the
7 transaction can be processed. In such scenarios, the social pay server may provide a pay
8 command verification request 525 to the client, which the client may display, e.g., 526,
9 to the user. In some embodiments, the user may provide a verification input 527 into the
10 client, which may provide a pay command verification response to the social pay server,
11 528. The social pay server may determine whether the payor verified payment, whether
12 payee information available is sufficient to process the transaction, and/or the like, 529.
13 In scenarios where sufficient payee information is unavailable or the payor needs to be
14 contacted for payment verification, 530, option “No,” the social pay server may
15 optionally provide a social post message 531 to a social networking service associated
16 with the potential payee/payor requesting the payee to enroll in social pay service (e.g.,
17 using the SPE 300 component described above in the discussion with reference to
18 FIGURES 2-3) or provide verification, which the social network server may post 532-
19 533 for the payee. If all the requirements are met for processing the transaction, 530,
20 option “Yes,” the social pay server may generate a unique transaction trigger associated
21 with the triggering social post message, e.g., 534, and may optionally store a transaction
22 trigger ID, triggering social post message, etc., for recordkeeping or analytics purposes.
23 The social pay server may provide the transaction trigger to trigger a purchase

1 transaction, e.g., via a purchase transaction authorization such as the example PTA 1400
2 component described below in the discussion with reference to FIGURES 13-14.

3 **[0063]** FIGURES 6A-B show logic flow diagrams illustrating example aspects of
4 implementing wallet security and settings in some embodiments of the SocialPay, e.g., a
5 Something (“WSS”) component 600. In some embodiments, the social pay server may
6 process the user social data using the social pay rules to identify pay commands, pay
7 requests, merchant offers, and/or like content of the user social data. In some
8 embodiments, rules may be provided by the SocialPay to ensure the privacy and security
9 of the user’s social data and virtual wallet. As another example, the rules may include
10 procedures to detect fraudulent transaction attempts, and request user verification
11 before proceeding, or cancel the transaction request entirely.

12 **[0064]** Accordingly, with reference to FIGURE 6A, in some embodiments, the
13 SocialPay may obtain a trigger to process a user’s social data (e.g., from FIGURE 5B,
14 element 531), 601. The SocialPay may obtain user and/or user social graph member
15 social data, as well as pay command rules and templates (e.g., for identifying standard
16 pay commands), 602. The SocialPay may parse the obtained user social data in
17 preparation for rules processing, 603. For example, the SocialPay may utilize parsers
18 such as the example parsers described below in the discussion with reference to
19 FIGURE 24. The SocialPay may select a pay command rule/template for processing.
20 The SocialPay may search through the parsed user social data, e.g., in a sequential
21 manner, for the selected pay command, 612, and determine whether the pay command
22 is present in the user social data, 613. It should be noted that in an alternative
23 embodiment such parsing and processing may occur continuously and in real time

1 through interception parsing where SocialPay is logged into a user social account (e.g.,
2 with a user's provided login credentials) and as such receiving every post made by the
3 user and other clients and receiving every message directed to the user and parsing such
4 messages in real time as they occur (e.g., see paragraphs [0045] and [0052] above for
5 further detail). If the pay command is identified, 614, option "Yes," the SocialPay may
6 place the identified pay command string, an identification of the rule/template, the
7 actual listing of the rule/template, and/or the like in a queue for further processing, 615.
8 The SocialPay may perform such a procedure until the entirety of the user's social data
9 has been searched through (see 616). In some embodiments, the SocialPay may
10 perform the above procedure for all available rules/templates, to identify all the pay
11 command strings included in the user social data (see 617).

12 **[0065]** In some embodiments, the SocialPay may process each pay command
13 identified from the user social data, 720. For example, the SocialPay may select a pay
14 command string from the queue and its associated template/identification rule, 621.
15 Using the rule/template and pay command string, the SocialPay may determine whether
16 the string represents a request for payment, or an order to pay, 623. If the pay
17 command string represents a request for payment (e.g., "hey @jfdoe, you owe me 25
18 bucks #cashflowblues"), 624, option "Yes," the SocialPay may determine whether the
19 user for whom the WSS component is executing is the requested payor, or the payee,
20 625. If the user has been requested to pay, 626, option "Yes," the SocialPay may add a
21 payment reminder to the user wallet account, 627. Otherwise, the SocialPay may
22 generate a user pay request record including the pay command details, 628, and store
23 the pay request record in the user's wallet account for recordkeeping purposes or future
24 analytics processing, 629.

1 **[0066]** With reference to FIGURE 6B, in some embodiments, the SocialPay may
2 extract an identification of a payor and payee in the transaction, 631. The SocialPay may
3 query a database for payee account data for payment processing, 632. If the payee data
4 available is insufficient, 633, option “Yes,” the SocialPay may generate a social post
5 message to the payee’s social network account 634, requesting that the payee either
6 enroll in the SocialPay (if not already), or provide additional information so that the
7 SocialPay may process the transaction. The SocialPay may provide 635 the social post
8 message to the social networking service associated with the payee. If sufficient payee
9 information is available, 633, option “No,” the SocialPay may query the payor’s wallet
10 account for security rules associated with utilizing the virtual wallet account, 636. The
11 SocialPay may select a wallet security rule, 637, and process the security rule using the
12 pay command string as input data, 638. Based on the processing, the SocialPay may
13 determine whether the pay command passes the security rule, or instead poses a
14 security risk to the user wallet. If the security rule is not passed, 640, option “No,” the
15 SocialPay may determine whether verification from the user can salvage the pay
16 command string, 641. If the SocialPay determines that the risk is too great, the
17 SocialPay may directly terminate the transaction and remove the pay command string
18 from the processing queue. Otherwise (641, option “Yes”), the SocialPay may generate a
19 pay command verification request for the user, 642, and provide the pay command
20 verification request as an output of the component, 643. If all security rules are passed
21 for the pay command string, 644, option “No,” the SocialPay may generate a transaction
22 trigger with a trigger ID (such as a card authorization request), and provide the
23 transaction trigger for payment processing.

1 **[0067]** FIGURE 7 shows a data flow diagram illustrating an example social
2 merchant consumer bridging procedure in some embodiments of the SocialPay. In
3 some implementations, a social pay server 703a may be triggered to provide services
4 that bridge consumers and merchants over social networks. For example, the social pay
5 server may identify a consumer is need of offers for products or services, and may
6 identify merchants participating in SocialPay that can provide the needed products or
7 services. The social pay server may generate offers on behalf of the participating
8 merchants, and provide the offers to consumers via social networks. In some
9 embodiments, the social pay server may periodically initiate merchant-consumer
10 bridging services for a user. In alternate embodiments, the social pay server may
11 initiate merchant-consumer bridging upon notification of a consumer engaging in a
12 transaction (e.g., a consumer may request checkout for a purchase via the user's virtual
13 wallet; for illustration, see the example User Purchase Checkout (UPC) component 1200
14 described further below in the discussion with reference to FIGURES 11-12), or when a
15 authorization is requested for a purchase transaction (see the example Purchase
16 Transaction Authorization (PTA) component 1400 described further below in the
17 discussion with reference to FIGURES 13-14). Upon obtaining a trigger to perform
18 merchant-consumer bridging, the social pay server may invoke a transaction data
19 aggregation component, e.g., the TDA component 900 described further below in the
20 discussion with reference to FIGURE 9. The social pay server may query a social pay
21 database 703b for offer generation rules, e.g., 713. For example, the social pay server
22 may utilize PHP/SQL commands similar to the other examples described herein. In
23 response, the database may provide the requested offer generation rules, e.g., 714. Using
24 the aggregated transaction data and the offer generation rules, the social pay server may

1 generate merchant(s) offer social post messages for posting to profiles of the user on
2 social networks, e.g., 715. For example, the social pay server may invoke a transaction-
3 based offer generation component, such as the example TBOG 1000 component
4 described further below in the discussion with reference to FIGURE 10. The social pay
5 server may provide the generated social post messages 716 to a social network server
6 704a. The social network server may store the social post messages 717 to a social
7 network database 704b for distribution to the user (e.g., when the user logs onto the
8 social networking service provided by the social network server).

9 **[0068]** FIGURE 8 shows a logic flow diagram illustrating example aspects of
10 social merchant consumer bridging in some embodiments of the SocialPay, e.g., a Social
11 Merchant Consumer Bridging (“SMCB”) component 800. In some implementations, a
12 social pay server may be triggered to provide services that bridge consumers and
13 merchants over social networks, e.g., 801. Upon obtaining a trigger to perform
14 merchant-consumer bridging, the social pay server may invoke a transaction data
15 aggregation component such as the TDA component 900 described further below in the
16 discussion with reference to FIGURE 9, e.g., 802. The social pay server may query a
17 social pay database for offer generation rules, e.g., 803. For example, the social pay
18 server may utilize PHP/SQL commands similar to the other examples described herein.
19 In response, the database may provide the requested offer generation rules, e.g., 804.
20 Using the aggregated transaction data and the offer generation rules, the social pay
21 server may generate merchant(s) offer social post messages for posting to profiles of the
22 user on social networks, e.g., 805. For example, the social pay server may invoke a
23 transaction-based offer generation component, such as the example TBOG 1000
24 component described further below in the discussion with reference to FIGURE 10. The

1 social pay server may provide the generated social post messages to a social network
2 server. The social network server may store the social post messages to a social network
3 database for distribution to the user (e.g., when the user logs onto the social networking
4 service provided by the social network server). In some embodiments, the social
5 network server may generate, using social graph data of the user, social post messages
6 for the user and/or members of the user's social graph, e.g., 806, and store the social
7 post message in a social network database for posting to their profiles, e.g., 807.

8 **[0069]** FIGURE 9 shows a logic flow diagram illustrating example aspects of
9 transaction data aggregation in some embodiments of the SocialPay, e.g., a Transaction
10 Data Aggregation ("TDA") component 900. In some embodiments, a social pay server
11 may obtain a trigger to aggregate transaction data, e.g., 901. For example, the server
12 may be configured to initiate transaction data aggregation on a regular, periodic, or
13 continuous basis. As another example, the server may be configured to initiate
14 transaction data aggregation in real-time on-demand. The social pay server may
15 determine a scope of data aggregation desired to perform the transaction analytics, e.g.,
16 902. For example, the scope of data aggregation may be pre-determined. As another
17 example, the scope of data aggregation may be determined based on a received request
18 for analytics, in real-time. The social pay server may initiate data aggregation based on
19 the determined scope. The social pay server may generate a query for addresses of
20 servers storing transaction data within the determined scope, e.g., 903. The social pay
21 server may query a database for addresses of other servers that may have stored
22 transaction data within the determined scope of the data aggregation. The database
23 may provide, e.g., 904, a list of server addresses in response to the social pay server's
24 query. Based on the list of server addresses, the social pay server may generate

1 transaction data requests, e.g., 905. The social pay server may issue the generated
2 transaction data requests to the other servers. The other servers may obtain and parse
3 the transaction data requests, e.g., 906. Based on parsing the data requests, the other
4 servers may generate transaction data queries, e.g., 907, and provide the transaction
5 data queries to their transaction databases. In response to the transaction data queries,
6 the transaction databases may provide transaction data, e.g., 908, to the other servers.
7 The other servers may return, e.g., 909, the transaction data obtained from the
8 transactions databases to the social pay server making the transaction data requests.
9 The social pay server may generate aggregated transaction data records from the
10 transaction data received from the other servers, e.g., 910, and store the aggregated
11 transaction data in a database, e.g., 911.

12 **[0070]** FIGURE 10 shows a logic flow diagram illustrating example aspects of
13 transaction-based offer generation in some embodiments of the SocialPay, e.g., a
14 Transaction-Based Offer Generation (“TBOG”) component 1000. In some
15 embodiments, a server may generate one or more offers to provide for a SocialPay user
16 on behalf of a SocialPay participating merchant, based on analyzing aggregated
17 transaction data records of the user or like users (e.g., by demographic group, location,
18 members of the user’s social graph on a social networking service, common interests,
19 etc.). The server may obtain transactions from a database that are unanalyzed, e.g.,
20 1001, and obtain rules for generating offers, e.g., 1002. For example, the database may
21 store offer generation analytics rules, such as the exemplary illustrative XML-encoded
22 analytics rule provided below:

```
23 <rule>  
24   <id>ABCDE44_45</id>  
25   <name>Demog_search_23</name>
```

```
1      <inputs>product_type</inputs>
2      <operations>
3          <1>IF (product_type == 'gaming') $search_key += 'console game online
4      videogame'</2>
5          <2> result = SEARCH ($offers_db, $search_key)
6      </operations>
7      <outputs>result</outputs>
8  </rule>
```

11 **[0071]** The server may select an unanalyzed data record for processing, e.g., 1003.

12 The server may also select an analytics rule for processing the unanalyzed data record,

13 e.g., 1004. The server may parse the analytics rule, and determine the desired inputs for

14 the rule, e.g., 1005. Based on parsing the analytics rule, the server may parse the data

15 record template, e.g., 1006, and extract the values for the fields required as inputs to the

16 analytics rule. For example, to process the rule in the example above, the server may

17 extract the value of the field 'product_type' from the transaction data record. The server

18 may parse the analytics rule, and extract the operations to be performed on the inputs

19 provided for the rule processing, e.g., 1007. Upon determining the operations to be

20 performed, the server may perform the rule-specified operations on the inputs provided

21 for the analytics rule, e.g., 1008. In some embodiments, the rule may provide threshold

22 values. For example, the rule may specify restrictions, such as, but not limited to: that

23 the number of products in the transaction, total value of the transaction, average luxury

24 rating of the products sold in the transaction, etc. may need to cross a threshold in order

25 for the label(s) associated with the rule to be applied to the transaction data record. The

26 server may parse the analytics rule to extract any threshold values required for the rule

27 to apply, e.g., 1009. The server may compare the computed values with the rule

28 thresholds, e.g., 1010. If the rule threshold(s) is crossed, e.g., 1011, option "Yes," the

29 server may generate offers for the user according to the rule and add the generated

30 offers to a data record, e.g., 1012. For example, for the example rule above, the server

1 may perform a search using the additional keywords, and add the returned results to the
2 data record. In some embodiments, the server may apply an analytics rule to an
3 individual product within the transaction, and/or to the transaction as a whole. In some
4 embodiments, the server may process the transaction data record using each rule (see,
5 e.g., 1013). Once all offer analytics rules have been processed for the transaction record,
6 e.g., 1013, option “No,” the server may generate and provide a social post message using
7 offers included in the data record, e.g., 1014. The server may perform such processing
8 for each transaction data record until all transaction data records have been processed
9 (see, e.g., 1015).

10 **[0072]** FIGURE 11 shows a data flow diagram illustrating an example user
11 purchase checkout procedure in some embodiments of the SocialPay. In some
12 embodiments, a user, e.g., 1101a, may desire to purchase a product, service, offering,
13 and/or the like (“product”), from a merchant via a merchant online site or in the
14 merchant’s store. The user may communicate with a merchant/acquirer (“merchant”)
15 server, e.g., 1103a, via a client such as, but not limited to: a personal computer, mobile
16 device, television, point-of-sale terminal, kiosk, ATM, and/or the like (e.g., 1102). For
17 example, the user may provide user input, e.g., checkout input 1111, into the client
18 indicating the user’s desire to purchase the product. In various embodiments, the user
19 input may include, but not be limited to: a single tap (e.g., a one-tap mobile app
20 purchasing embodiment) of a touchscreen interface, keyboard entry, card swipe,
21 activating a RFID/NFC enabled hardware device (e.g., electronic card having multiple
22 accounts, smartphone, tablet, etc.) within the user device, mouse clicks, depressing
23 buttons on a joystick/game console, voice commands, single/multi-touch gestures on a
24 touch-sensitive interface, touching user interface elements on a touch-sensitive display,

1 and/or the like. As an example, a user in a merchant store may scan a product barcode
 2 of the product via a barcode scanner at a point-of-sale terminal. As another example,
 3 the user may select a product from a webpage catalog on the merchant's website, and
 4 add the product to a virtual shopping cart on the merchant's website. The user may
 5 then indicate the user's desire to checkout the items in the (virtual) shopping cart. For
 6 example, the user may activate a user interface element provided by the client to
 7 indicate the user's desire to complete the user purchase checkout. The client may
 8 generate a checkout request, e.g., 1112, and provide the checkout request, e.g., 1113, to
 9 the merchant server. For example, the client may provide a (Secure) Hypertext Transfer
 10 Protocol ("HTTP(S)") POST message including the product details for the merchant
 11 server in the form of data formatted according to the eXtensible Markup Language
 12 ("XML"). An example listing of a checkout request 1112, substantially in the form of a
 13 HTTP(S) POST message including XML-formatted data, is provided below:

```

14 POST /checkoutrequest.php HTTP/1.1
15 Host: www.merchant.com
16 Content-Type: Application/XML
17 Content-Length: 667
18 <?XML version = "1.0" encoding = "UTF-8"?>
19 <checkout_request>
20   <checkout_ID>4NFU4RG94</checkout_ID>
21   <timestamp>2011-02-22 15:22:43</timestamp>
22   <purchase_detail>
23     <num_products>5</num_products>
24     <product_ID>AE95049324</product_ID>
25     <product_ID>MD09808755</product_ID>
26     <product_ID>OC12345764</product_ID>
27     <product_ID>KE76549043</product_ID>
28     <product_ID>SP27674509</product_ID>
29   </purchase_detail>
30   <!--optional parameters-->
31   <user_ID>john.q.public@gmail.com</user_ID>
32   <PoS_client_detail>
33     <client_IP>192.168.23.126</client_IP>
34     <client_type>smartphone</client_type>
35     <client_model>HTC Hero</client_model>
36     <OS>Android 2.2</OS>
37     <app_installed_flag>true</app_installed_flag>
38   </PoS_client_detail>
39 </checkout_request>
40
41
```

1 **[0073]** In some embodiments, the merchant server may obtain the checkout
 2 request from the client, and extract the checkout detail (e.g., XML data) from the
 3 checkout request. For example, the merchant server may utilize a parser such as the
 4 example parsers described below in the discussion with reference to FIGURE 24. Based
 5 on parsing the checkout request 1112, the merchant server may extract product data
 6 (e.g., product identifiers), as well as available PoS client data, from the checkout request.
 7 In some embodiments, using the product data, the merchant server may query, e.g.,
 8 1114, a merchant/acquirer (“merchant”) database, e.g., 1103b, to obtain product data,
 9 e.g., 1115, such as product information, product pricing, sales tax, offers, discounts,
 10 rewards, and/or other information to process the purchase transaction and/or provide
 11 value-added services for the user. For example, the merchant database may be a
 12 relational database responsive to Structured Query Language (“SQL”) commands. The
 13 merchant server may execute a hypertext preprocessor (“PHP”) script including SQL
 14 commands to query a database table (such as FIGURE 24, Products 2419l) for product
 15 data. An example product data query 1114, substantially in the form of PHP/SQL
 16 commands, is provided below:

```

17 <?PHP
18 header('Content-Type: text/plain');
19 mysql_connect("254.93.179.112",$DBserver,$password); // access database server
20 mysql_select_db("SocialPay_DB.SQL"); // select database table to search
21 //create query
22 $query = "SELECT product_title product_attributes_list product_price
23         tax_info_list related_products_list offers_list discounts_list rewards_list
24         merchants_list merchant_availability_list FROM ProductsTable WHERE
25         product_ID LIKE '%" $prodID";
26 $result = mysql_query($query); // perform the search query
27 mysql_close("SocialPay_DB.SQL"); // close database access
28 ?>
29
30
```

31 **[0074]** In some embodiments, in response to obtaining the product data, the
 32 merchant server may generate, e.g., 1116, checkout data to provide for the PoS client. In

1 some embodiments, such checkout data, e.g., 1117, may be embodied, in part, in a
2 HyperText Markup Language (“HTML”) page including data for display, such as
3 product detail, product pricing, total pricing, tax information, shipping information,
4 offers, discounts, rewards, value-added service information, etc., and input fields to
5 provide payment information to process the purchase transaction, such as account
6 holder name, account number, billing address, shipping address, tip amount, etc. In
7 some embodiments, the checkout data may be embodied, in part, in a Quick Response
8 (“QR”) code image that the PoS client can display, so that the user may capture the QR
9 code using a user’s device to obtain merchant and/or product data for generating a
10 purchase transaction processing request. In some embodiments, a user alert
11 mechanism may be built into the checkout data. For example, the merchant server may
12 embed a URL specific to the transaction into the checkout data. In some embodiments,
13 the alerts URL may further be embedded into optional level 3 data in card authorization
14 requests, such as those discussed further below with reference to FIGURES 13-14. The
15 URL may point to a webpage, data file, executable script, etc., stored on the merchant's
16 server dedicated to the transaction that is the subject of the card authorization request.
17 For example, the object pointed to by the URL may include details on the purchase
18 transaction, e.g., products being purchased, purchase cost, time expiry, status of order
19 processing, and/or the like. Thus, the merchant server may provide to the payment
20 network the details of the transaction by passing the URL of the webpage to the
21 payment network. In some embodiments, the payment network may provide
22 notifications to the user, such as a payment receipt, transaction authorization
23 confirmation message, shipping notification and/or the like. In such messages, the
24 payment network may provide the URL to the user device. The user may navigate to the

1 URL on the user's device to obtain alerts regarding the user's purchase, as well as other
 2 information such as offers, coupons, related products, rewards notifications, and/or the
 3 like. An example listing of a checkout data 1117, substantially in the form of XML-
 4 formatted data, is provided below:

```

5  <?XML version = "1.0" encoding = "UTF-8"?>
6  <checkout_data>
7      <session_ID>4NFU4RG94</session_ID>
8      <timestamp>2011-02-22 15:22:43</timestamp>
9      <expiry_lapse>00:00:30</expiry_lapse>
10     <transaction_cost>$34.78</transaction_cost>
11     <alerts_URL>www.merchant.com/shopcars.php?sessionID=4NFU4RG94</alerts_URL>
12     <!--optional data-->
13     <user_ID>john.q.public@gmail.com</user_ID>
14     <client_details>
15         <client_IP>192.168.23.126</client_IP>
16         <client_type>smartphone</client_type>
17         <client_model>HTC Hero</client_model>
18         <OS>Android 2.2</OS>
19         <app_installed_flag>true</app_installed_flag>
20     </client_details>
21     <purchase_details>
22         <num_products>1</num_products>
23         <product>
24             <product_type>book</product_type>
25             <product_params>
26                 <product_title>XML for dummies</product_title>
27                 <ISBN>938-2-14-168710-0</ISBN>
28                 <edition>2nd ed.</edition>
29                 <cover>hardbound</cover>
30                 <seller>bestbuybooks</seller>
31             </product_params>
32             <quantity>1</quantity>
33         </product>
34     </purchase_details>
35     <offers_details>
36         <num_offers>1</num_offers>
37         <product>
38             <product_type>book</product_type>
39             <product_params>
40                 <product_title>Here's more XML</product_title>
41                 <ISBN>922-7-14-165720-1</ISBN>
42                 <edition>1nd ed.</edition>
43                 <cover>hardbound</cover>
44                 <seller>digibooks</seller>
45             </product_params>
46             <quantity>1</quantity>
47         </product>
48     </offers_details>
49     <secure_element>www.merchant.com/securedyn/0394733/123.png</secure_element>
50     <merchant_params>
51         <merchant_id>3FBCR4INC</merchant_id>
52         <merchant_name>Books & Things, Inc.</merchant_name>
53         <merchant_auth_key>1NNF484MCP59CHB27365</merchant_auth_key>
54     </merchant_params>
55 </checkout_data>
56

```

1 **[0075]** Upon obtaining the checkout data, e.g., 1117, the PoS client may render
2 and display, e.g., 1118, the checkout data for the user.

3 **[0076]** FIGURE 12 shows a logic flow diagram illustrating example aspects of a
4 user purchase checkout in some embodiments of the SocialPay, e.g., a User Purchase
5 Checkout (“UPC”) component 1200. In some embodiments, a user may desire to
6 purchase a product, service, offering, and/or the like (“product”), from a merchant via a
7 merchant online site or in the merchant’s store. The user may communicate with a
8 merchant/acquirer (“merchant”) server via a PoS client. For example, the user may
9 provide user input, e.g., 1201, into the client indicating the user’s desire to purchase the
10 product. The client may generate a checkout request, e.g., 1202, and provide the
11 checkout request to the merchant server. In some embodiments, the merchant server
12 may obtain the checkout request from the client, and extract the checkout detail (e.g.,
13 XML data) from the checkout request. For example, the merchant server may utilize a
14 parser such as the example parsers described below in the discussion with reference to
15 FIGURE 24. Based on parsing the checkout request, the merchant server may extract
16 product data (e.g., product identifiers), as well as available PoS client data, from the
17 checkout request. In some embodiments, using the product data, the merchant server
18 may query, e.g., 1203, a merchant/acquirer (“merchant”) database to obtain product
19 data, e.g., 1204, such as product information, product pricing, sales tax, offers,
20 discounts, rewards, and/or other information to process the purchase transaction
21 and/or provide value-added services for the user. In some embodiments, in response to
22 obtaining the product data, the merchant server may generate, e.g., 1205, checkout data
23 to provide, e.g., 1206, for the PoS client. Upon obtaining the checkout data, the PoS
24 client may render and display, e.g., 1207, the checkout data for the user.

1 **[0077]** FIGURES 13A-B show data flow diagrams illustrating an example
2 purchase transaction authorization procedure in some embodiments of the SocialPay.
3 With reference to FIGURE 13A, in some embodiments, a user, e.g., 1301a, may wish to
4 utilize a virtual wallet account to purchase a product, service, offering, and/or the like
5 (“product”), from a merchant via a merchant online site or in the merchant’s store. The
6 user may utilize a physical card, or a user wallet device, e.g., 1301b, to access the user’s
7 virtual wallet account. For example, the user wallet device may be a personal/laptop
8 computer, cellular telephone, smartphone, tablet, eBook reader, netbook, gaming
9 console, and/or the like. The user may provide a wallet access input, e.g., 1311 into the
10 user wallet device. In various embodiments, the user input may include, but not be
11 limited to: a single tap (e.g., a one-tap mobile app purchasing embodiment) of a
12 touchscreen interface, keyboard entry, card swipe, activating a RFID/NFC enabled
13 hardware device (e.g., electronic card having multiple accounts, smartphone, tablet,
14 etc.) within the user device, mouse clicks, depressing buttons on a joystick/game
15 console, voice commands, single/multi-touch gestures on a touch-sensitive interface,
16 touching user interface elements on a touch-sensitive display, and/or the like. In some
17 embodiments, the user wallet device may authenticate the user based on the user’s
18 wallet access input, and provide virtual wallet features for the user.

19 **[0078]** In some embodiments, upon authenticating the user for access to virtual
20 wallet features, the user wallet device may provide a transaction authorization input,
21 e.g., 1314, to a point-of-sale (“PoS”) client, e.g., 1302. For example, the user wallet device
22 may communicate with the PoS client via Bluetooth, Wi-Fi, cellular communication, one- or
23 two-way near-field communication (“NFC”), and/or the like. In embodiments where the user
24 utilizes a plastic card instead of the user wallet device, the user may swipe the plastic card at

1 the PoS client to transfer information from the plastic card into the PoS client. For example,
 2 the PoS client may obtain, as transaction authorization input 1314, track 1 data from the
 3 user's plastic card (e.g., credit card, debit card, prepaid card, charge card, etc.), such as
 4 the example track 1 data provided below:

```
5 %B123456789012345^PUBLIC/J.Q.^99011200000000000000**901*****?*
6 (wherein '123456789012345' is the card number of 'J.Q. Public' and has a CVV
7 number of 901. '990112' is a service code, and *** represents decimal digits
8 which change randomly each time the card is used.)
9
10
```

11 **[0079]** In embodiments where the user utilizes a user wallet device, the user
 12 wallet device may provide payment information to the PoS client, formatted according
 13 to a data formatting protocol appropriate to the communication mechanism employed
 14 in the communication between the user wallet device and the PoS client. An example
 15 listing of transaction authorization input 1314, substantially in the form of XML-
 16 formatted data, is provided below:

```
17 <?XML version = "1.0" encoding = "UTF-8"?>
18 <transaction_authorization_input>
19   <payment_data>
20     <account>
21       <charge_priority>1</charge_priority>
22       <charge_ratio>40%</charge_ratio>
23       <account_number>123456789012345</account_number>
24       <account_name>John Q. Public</account_name>
25       <bill_add>987 Green St #456, Chicago, IL 94652</bill_add>
26       <ship_add>987 Green St #456, Chicago, IL 94652</ship_add>
27       <CVV>123</CVV>
28     </account>
29     <account>
30       <charge_priority>1</charge_priority>
31       <charge_ratio>60%</charge_ratio>
32       <account_number>234567890123456</account_number>
33       <account_name>John Q. Public</account_name>
34       <bill_add>987 Green St #456, Chicago, IL 94652</bill_add>
35       <ship_add>987 Green St #456, Chicago, IL 94652</ship_add>
36       <CVV>173</CVV>
37     </account>
38     <account>
39       <charge_priority>2</charge_priority>
40       <charge_ratio>100%</charge_ratio>
41       <account_number>345678901234567</account_number>
42       <account_name>John Q. Public</account_name>
43       <bill_add>987 Green St #456, Chicago, IL 94652</bill_add>
44       <ship_add>987 Green St #456, Chicago, IL 94652</ship_add>
45       <CVV>695</CVV>
```

```

1         </account>
2     </payment_data>
3     <!--optional data-->
4     <timestamp>2011-02-22 15:22:43</timestamp>
5     <expiry_lapse>00:00:30</expiry_lapse>
6     <secure_key>0445329070598623487956543322</secure_key>
7     <alerts_track_flag>TRUE</alerts_track_flag>
8     <wallet_device_details>
9         <device_IP>192.168.23.126</client_IP>
10        <device_type>smartphone</client_type>
11        <device_model>HTC Hero</client_model>
12        <OS>Android 2.2</OS>
13        <wallet_app_installed_flag>true</wallet_app_installed_flag>
14    </wallet_device_details>
15 </transaction_authorization_input>
16
17

```

[0080] In some embodiments, the PoS client may generate a card authorization request, e.g., 1315, using the obtained transaction authorization input from the user wallet device, and/or product/checkout data (see, e.g., FIGURE 11, 1115-1117). An example listing of a card authorization request 1315, substantially in the form of a HTTP(S) POST message including XML-formatted data, is provided below:

```

23 POST /authorizationrequests.php HTTP/1.1
24 Host: www.acquirer.com
25 Content-Type: Application/XML
26 Content-Length: 1306
27 <?XML version = "1.0" encoding = "UTF-8"?>
28 <card_authorization_request>
29     <session_ID>4NFU4RG94</order_ID>
30     <timestamp>2011-02-22 15:22:43</timestamp>
31     <expiry>00:00:30</expiry>
32     <alerts_URL>www.merchant.com/shopcarts.php?sessionID=AE4356</alerts_URL>
33     <!--optional data-->
34     <user_ID>john.q.public@gmail.com</user_ID>
35     <PoS_details>
36         <PoS_IP>192.168.23.126</client_IP>
37         <PoS_type>smartphone</client_type>
38         <PoS_model>HTC Hero</client_model>
39         <OS>Android 2.2</OS>
40         <app_installed_flag>true</app_installed_flag>
41     </PoS_details>
42     <purchase_details>
43         <num_products>1</num_products>
44         <product>
45             <product_type>book</product_type>
46             <product_params>
47                 <product_title>XML for dummies</product_title>
48                 <ISBN>938-2-14-168710-0</ISBN>
49                 <edition>2nd ed.</edition>
50                 <cover>hardbound</cover>
51                 <seller>bestbuybooks</seller>
52             </product_params>
53             <quantity>1</quantity>
54         </product>

```

```

1      </purchase_details>
2      <merchant_params>
3          <merchant_id>3FBCR4INC</merchant_id>
4          <merchant_name>Books & Things, Inc.</merchant_name>
5          <merchant_auth_key>1NNF484MCP59CHB27365</merchant_auth_key>
6      </merchant_params>
7      <account_params>
8          <account_name>John Q. Public</account_name>
9          <account_type>credit</account_type>
10         <account_num>123456789012345</account_num>
11         <billing_address>123 Green St., Norman, OK 98765</billing_address>
12         <phone>123-456-7809</phone>
13         <sign>/jpg</sign>
14         <confirm_type>email</confirm_type>
15         <contact_info>john.q.public@gmail.com</contact_info>
16     </account_params>
17     <shipping_info>
18         <shipping_address>same as billing</shipping_address>
19         <ship_type>expedited</ship_type>
20         <ship_carrier>FedEx</ship_carrier>
21         <ship_account>123-45-678</ship_account>
22         <tracking_flag>true</tracking_flag>
23         <sign_flag>false</sign_flag>
24     </shipping_info>
25 </card_authorization_request>
26
27

```

[0081] In some embodiments, the card authorization request generated by the user device may include a minimum of information to process the purchase transaction. For example, this may improve the efficiency of communicating the purchase transaction request, and may also advantageously improve the privacy protections provided to the user and/or merchant. For example, in some embodiments, the card authorization request may include at least a session ID for the user's shopping session with the merchant. The session ID may be utilized by any component and/or entity having the appropriate access authority to access a secure site on the merchant server to obtain alerts, reminders, and/or other data about the transaction(s) within that shopping session between the user and the merchant. In some embodiments, the PoS client may provide the generated card authorization request to the merchant server, e.g., 1316. The merchant server may forward the card authorization request to a pay gateway server, e.g., 1304a, for routing the card authorization request to the appropriate payment network for payment processing. For example, the pay gateway server may be able to

1 select from payment networks, such as Visa, Mastercard, American Express, Paypal,
 2 etc., to process various types of transactions including, but not limited to: credit card,
 3 debit card, prepaid card, B2B and/or like transactions. In some embodiments, the
 4 merchant server may query a database, e.g., merchant/acquirer database 1303b, for a
 5 network address of the payment gateway server, for example by using a portion of a user
 6 payment card number, or a user ID (such as an email address) as a keyword for the database
 7 query. For example, the merchant server may issue PHP/SQL commands to query a
 8 database table (such as FIGURE 24, Pay Gateways 2419h) for a URL of the pay gateway
 9 server. An example payment gateway address query 1317, substantially in the form of
 10 PHP/SQL commands, is provided below:

```

11  <?PHP
12  header('Content-Type: text/plain');
13  mysql_connect("254.93.179.112",$DBserver,$password); // access database server
14  mysql_select_db("SocialPay_DB.SQL"); // select database table to search
15  //create query
16  $query = "SELECT paygate_id paygate_address paygate_URL paygate_name FROM
17          PayGatewayTable WHERE card_num LIKE '%" $cardnum";
18  $result = mysql_query($query); // perform the search query
19  mysql_close("SocialPay_DB.SQL"); // close database access
20  ?>
21
22

```

23 **[0082]** In response, the merchant/acquirer database may provide the requested
 24 payment gateway address, e.g., 1318. The merchant server may forward the card
 25 authorization request to the pay gateway server using the provided address, e.g., 1319. In
 26 some embodiments, upon receiving the card authorization request from the merchant server,
 27 the pay gateway server may invoke a component to provide one or more services
 28 associated with purchase transaction authorization. For example, the pay gateway
 29 server may invoke components for fraud prevention, loyalty and/or rewards, and/or
 30 other services for which the user-merchant combination is authorized. The pay gateway
 31 server may forward the card authorization request to a social pay server, e.g., 1305a, for

1 payment processing. For example, the pay gateway server may be able to select from
 2 payment networks, such as Visa, Mastercard, American Express, Paypal, etc., to process
 3 various types of transactions including, but not limited to: credit card, debit card,
 4 prepaid card, B2B and/or like transactions. In some embodiments, the pay gateway
 5 server may query a database, e.g., pay gateway database 1304b, for a network address of
 6 the payment network server, for example by using a portion of a user payment card number,
 7 or a user ID (such as an email address) as a keyword for the database query. For example,
 8 the pay gateway server may issue PHP/SQL commands to query a database table (such
 9 as FIGURE 24, Pay Gateways 2419h) for a URL of the social pay server. An example
 10 payment network address query 1321, substantially in the form of PHP/SQL commands,
 11 is provided below:

```

12  <?PHP
13  header('Content-Type: text/plain');
14  mysql_connect("254.93.179.112", $DBserver, $password); // access database server
15  mysql_select_db("SocialPay_DB.SQL"); // select database table to search
16  //create query
17  $query = "SELECT payNET_id payNET_address payNET_URL payNET_name FROM
18  PayGatewayTable WHERE card_num LIKE '%" $cardnum";
19  $result = mysql_query($query); // perform the search query
20  mysql_close("SocialPay_DB.SQL"); // close database access
21  ?>
22
23

```

24 **[0083]** In response, the payment gateway database may provide the requested
 25 payment network address, e.g., 1322. The pay gateway server may forward the card
 26 authorization request to the social pay server using the provided address, e.g., 1323.

27 **[0084]** With reference to FIGURE 13B, in some embodiments, the social pay server
 28 may process the transaction so as to transfer funds for the purchase into an account
 29 stored on an acquirer of the merchant. For example, the acquirer may be a financial
 30 institution maintaining an account of the merchant. For example, the proceeds of

1 transactions processed by the merchant may be deposited into an account maintained
2 by at a server of the acquirer.

3 **[0085]** In some embodiments, the social pay server may generate a query, e.g.,
4 1324, for issuer server(s) corresponding to the user-selected payment options. For
5 example, the user's account may be linked to one or more issuer financial institutions
6 ("issuers"), such as banking institutions, which issued the account(s) for the user. For
7 example, such accounts may include, but not be limited to: credit card, debit card,
8 prepaid card, checking, savings, money market, certificates of deposit, stored (cash)
9 value accounts and/or the like. Issuer server(s), e.g., 1306a, of the issuer(s) may
10 maintain details of the user's account(s). In some embodiments, a database, e.g., social
11 pay database 1305b, may store details of the issuer server(s) associated with the
12 issuer(s). In some embodiments, the social pay server may query a database, e.g., social
13 pay database 1305b, for a network address of the issuer(s) server(s), for example by using a
14 portion of a user payment card number, or a user ID (such as an email address) as a keyword
15 for the database query. For example, the merchant server may issue PHP/SQL
16 commands to query a database table (such as FIGURE 24, Issuers 2419f) for network
17 address(es) of the issuer(s) server(s). An example issuer server address(es) query 1324,
18 substantially in the form of PHP/SQL commands, is provided below:

```
19 <?PHP
20 header('Content-Type: text/plain');
21 mysql_connect("254.93.179.112", $DBserver, $password); // access database server
22 mysql_select_db("SocialPay_DB.SQL"); // select database table to search
23 //create query
24 $query = "SELECT issuer_id issuer_address issuer_URL issuer_name FROM
25     IssuersTable WHERE card_num LIKE '%" $cardnum";
26 $result = mysql_query($query); // perform the search query
27 mysql_close("SocialPay_DB.SQL"); // close database access
28 ?>
29
30
```

1 **[0086]** In response to obtaining the issuer server query, e.g., 1324, the social pay
 2 database may provide, e.g., 1325, the requested issuer server data to the social pay
 3 server. In some embodiments, the social pay server may utilize the issuer server data to
 4 generate funds authorization request(s), e.g., 1326, for each of the issuer server(s)
 5 selected based on the pre-defined payment settings associated with the user's virtual
 6 wallet, and/or the user's payment options input, and provide the funds authorization
 7 request(s) to the issuer server(s). In some embodiments, the funds authorization
 8 request(s) may include details such as, but not limited to: the costs to the user involved
 9 in the transaction, card account details of the user, user billing and/or shipping
 10 information, and/or the like. An example listing of a funds authorization request 1326,
 11 substantially in the form of a HTTP(S) POST message including XML-formatted data, is
 12 provided below:

```

13 POST /fundsauthorizationrequest.php HTTP/1.1
14 Host: www.issuer.com
15 Content-Type: Application/XML
16 Content-Length: 624
17 <?XML version = "1.0" encoding = "UTF-8"?>
18 <funds_authorization_request>
19   <query_ID>VNEI39FK</query_ID>
20   <timestamp>2011-02-22 15:22:44</timestamp>
21   <transaction_cost>$22.61</transaction_cost>
22   <account_params>
23     <account_type>checking</account_type>
24     <account_num>1234567890123456</account_num>
25   </account_params>
26   <!--optional parameters-->
27   <purchase_summary>
28     <num_products>1</num_products>
29     <product>
30       <product_summary>Book - XML for dummies</product_summary>
31       <product_quantity>1</product_quantity>
32     </product>
33   </purchase_summary>
34   <merchant_params>
35     <merchant_id>3FBCR4INC</merchant_id>
36     <merchant_name>Books & Things, Inc.</merchant_name>
37     <merchant_auth_key>1NNF484MCP59CHB27365</merchant_auth_key>
38   </merchant_params>
39 </funds_authorization_request>
40
41
```

1 **[0087]** In some embodiments, an issuer server may parse the authorization
 2 request(s), and based on the request details may query a database, e.g., user profile
 3 database 1306b, for data associated with an account linked to the user. For example, the
 4 merchant server may issue PHP/SQL commands to query a database table (such as
 5 FIGURE 24, Accounts 2419d) for user account(s) data. An example user account(s)
 6 query 1327, substantially in the form of PHP/SQL commands, is provided below:

```

7  <?PHP
8  header('Content-Type: text/plain');
9  mysql_connect("254.93.179.112",$DBserver,$password); // access database server
10 mysql_select_db("SocialPay_DB.SQL"); // select database table to search
11 //create query
12 $query = "SELECT issuer user_id user_name user_balance account_type FROM
13     AccountsTable WHERE account_num LIKE '%" $accountnum";
14 $result = mysql_query($query); // perform the search query
15 mysql_close("SocialPay_DB.SQL"); // close database access
16 ?>
17
18

```

19 **[0088]** In some embodiments, on obtaining the user account(s) data, e.g., 1328,
 20 the issuer server may determine whether the user can pay for the transaction using
 21 funds available in the account, 1329. For example, the issuer server may determine
 22 whether the user has a sufficient balance remaining in the account, sufficient credit
 23 associated with the account, and/or the like. Based on the determination, the issuer
 24 server(s) may provide a funds authorization response, e.g., 1330, to the social pay server.
 25 For example, the issuer server(s) may provide a HTTP(S) POST message similar to the
 26 examples above. In some embodiments, if at least one issuer server determines that the
 27 user cannot pay for the transaction using the funds available in the account, the social
 28 pay server may request payment options again from the user (e.g., by providing an
 29 authorization fail message to the user device and requesting the user device to provide
 30 new payment options), and re-attempt authorization for the purchase transaction. In
 31 some embodiments, if the number of failed authorization attempts exceeds a threshold,

1 the social pay server may abort the authorization process, and provide an “authorization
2 fail” message to the merchant server, user device and/or client.

3 **[0089]** In some embodiments, the social pay server may obtain the funds
4 authorization response including a notification of successful authorization, and parse
5 the message to extract authorization details. Upon determining that the user possesses
6 sufficient funds for the transaction, e.g., 1331, the social pay server may invoke a
7 component to provide value-add services for the user.

8 **[0090]** In some embodiments, the social pay server may generate a transaction
9 data record from the authorization request and/or authorization response, and store the
10 details of the transaction and authorization relating to the transaction in a transactions
11 database. For example, the social pay server may issue PHP/SQL commands to store
12 the data to a database table (such as FIGURE 24, Transactions 2419i). An example
13 transaction store command, substantially in the form of PHP/SQL commands, is
14 provided below:

```
15 <?PHP
16 header('Content-Type: text/plain');
17 mysql_connect("254.92.185.103",$DBserver,$password); // access database server
18 mysql_select("SocialPay_DB.SQL"); // select database to append
19 mysql_query("INSERT INTO TransactionsTable (PurchasesTable (timestamp,
20     purchase_summary_list, num_products, product_summary, product_quantity,
21     transaction_cost, account_params_list, account_name, account_type,
22     account_num, billing_addres, zipcode, phone, sign, merchant_params_list,
23     merchant_id, merchant_name, merchant_auth_key)
24 VALUES (time(), $purchase_summary_list, $num_products, $product_summary,
25     $product_quantity, $transaction_cost, $account_params_list, $account_name,
26     $account_type, $account_num, $billing_addres, $zipcode, $phone, $sign,
27     $merchant_params_list, $merchant_id, $merchant_name, $merchant_auth_key)");
28 // add data to table in database
29 mysql_close("SocialPay_DB.SQL"); // close connection to database
30 ?>
```

33 **[0091]** In some embodiments, the social pay server may forward a transaction
34 authorization response, e.g., 1332, to the user wallet device, PoS client, and/or merchant

1 server. The merchant may obtain the transaction authorization response, and
 2 determine from it that the user possesses sufficient funds in the card account to conduct
 3 the transaction. The merchant server may add a record of the transaction for the user to
 4 a batch of transaction data relating to authorized transactions. For example, the
 5 merchant may append the XML data pertaining to the user transaction to an XML data
 6 file comprising XML data for transactions that have been authorized for various users,
 7 e.g., 1333, and store the XML data file, e.g., 1334, in a database, e.g., merchant database
 8 404. For example, a batch XML data file may be structured similar to the example XML
 9 data structure template provided below:

```

10  <?XML version = "1.0" encoding = "UTF-8"?>
11  <merchant_data>
12    <merchant_id>3FBCR4INC</merchant_id>
13    <merchant_name>Books & Things, Inc.</merchant_name>
14    <merchant_auth_key>1NNF484MCP59CHB27365</merchant_auth_key>
15    <account_number>123456789</account_number>
16  </merchant_data>
17  <transaction_data>
18    <transaction 1>
19      ...
20    </transaction 1>
21    <transaction 2>
22      ...
23    </transaction 2>
24    .
25    .
26    .
27    <transaction n>
28      ...
29    </transaction n>
30  </transaction_data>
  
```

33 **[0092]** In some embodiments, the server may also generate a purchase receipt,
 34 e.g., 1333, and provide the purchase receipt to the client, e.g., 1335. The client may
 35 render and display, e.g., 1336, the purchase receipt for the user. In some embodiments,
 36 the user's wallet device may also provide a notification of successful authorization to the
 37 user. For example, the PoS client/user device may render a webpage, electronic
 38 message, text / SMS message, buffer a voicemail, emit a ring tone, and/or play an audio

1 message, etc., and provide output including, but not limited to: sounds, music, audio,
2 video, images, tactile feedback, vibration alerts (e.g., on vibration-capable client devices
3 such as a smartphone etc.), and/or the like.

4 **[0093]** FIGURES 14A-B show logic flow diagrams illustrating example aspects of
5 purchase transaction authorization in some embodiments of the SocialPay, e.g., a
6 Purchase Transaction Authorization (“PTA”) component 1400. With reference to
7 FIGURE 14A, in some embodiments, a user may wish to utilize a virtual wallet account
8 to purchase a product, service, offering, and/or the like (“product”), from a merchant via
9 a merchant online site or in the merchant’s store. The user may utilize a physical card,
10 or a user wallet device to access the user’s virtual wallet account. For example, the user
11 wallet device may be a personal/laptop computer, cellular telephone, smartphone, tablet,
12 eBook reader, netbook, gaming console, and/or the like. The user may provide a wallet
13 access input, e.g., 1401, into the user wallet device. In various embodiments, the user
14 input may include, but not be limited to: a single tap (e.g., a one-tap mobile app
15 purchasing embodiment) of a touchscreen interface, keyboard entry, card swipe,
16 activating a RFID/NFC enabled hardware device (e.g., electronic card having multiple
17 accounts, smartphone, tablet, etc.) within the user device, mouse clicks, depressing
18 buttons on a joystick/game console, voice commands, single/multi-touch gestures on a
19 touch-sensitive interface, touching user interface elements on a touch-sensitive display,
20 and/or the like. In some embodiments, the user wallet device may authenticate the user
21 based on the user’s wallet access input, and provide virtual wallet features for the user,
22 e.g., 1402-1403.

1 **[0094]** In some embodiments, upon authenticating the user for access to virtual
2 wallet features, the user wallet device may provide a transaction authorization input,
3 e.g., 1404, to a point-of-sale (“PoS”) client. For example, the user wallet device may
4 communicate with the PoS client via Bluetooth, Wi-Fi, cellular communication, one- or two-
5 way near-field communication (“NFC”), and/or the like. In embodiments where the user
6 utilizes a plastic card instead of the user wallet device, the user may swipe the plastic card at
7 the PoS client to transfer information from the plastic card into the PoS client. In
8 embodiments where the user utilizes a user wallet device, the user wallet device may
9 provide payment information to the PoS client, formatted according to a data formatting
10 protocol appropriate to the communication mechanism employed in the communication
11 between the user wallet device and the PoS client.

12 **[0095]** In some embodiments, the PoS client may obtain the transaction
13 authorization input, and parse the input to extract payment information from the
14 transaction authorization input, e.g., 1405. For example, the PoS client may utilize a
15 parser, such as the example parsers provided below in the discussion with reference to
16 FIGURE 24. The PoS client may generate a card authorization request, e.g., 1406, using
17 the obtained transaction authorization input from the user wallet device, and/or
18 product/checkout data (see, e.g., FIGURE 11, 1115-1117).

19 **[0096]** In some embodiments, the PoS client may provide the generated card
20 authorization request to the merchant server. The merchant server may forward the
21 card authorization request to a pay gateway server, for routing the card authorization
22 request to the appropriate payment network for payment processing. For example, the
23 pay gateway server may be able to select from payment networks, such as Visa,

1 Mastercard, American Express, Paypal, etc., to process various types of transactions
2 including, but not limited to: credit card, debit card, prepaid card, B2B and/or like
3 transactions. In some embodiments, the merchant server may query a database, e.g.,
4 1408, for a network address of the payment gateway server, for example by using a portion of
5 a user payment card number, or a user ID (such as an email address) as a keyword for the
6 database query. In response, the merchant/acquirer database may provide the requested
7 payment gateway address, e.g., 1410. The merchant server may forward the card
8 authorization request to the pay gateway server using the provided address. In some
9 embodiments, upon receiving the card authorization request from the merchant server, the
10 pay gateway server may invoke a component to provide one or more service associated
11 with purchase transaction authorization, e.g., 1411. For example, the pay gateway server
12 may invoke components for fraud prevention (see e.g., VerifyChat, FIGURE 3E), loyalty
13 and/or rewards, and/or other services for which the user-merchant combination is
14 authorized.

15 **[0097]** The pay gateway server may forward the card authorization request to a
16 social pay server for payment processing, e.g., 1414. For example, the pay gateway
17 server may be able to select from payment networks, such as Visa, Mastercard,
18 American Express, Paypal, etc., to process various types of transactions including, but
19 not limited to: credit card, debit card, prepaid card, B2B and/or like transactions. In
20 some embodiments, the pay gateway server may query a database, e.g., 1412, for a
21 network address of the payment network server, for example by using a portion of a user
22 payment card number, or a user ID (such as an email address) as a keyword for the database
23 query. In response, the payment gateway database may provide the requested payment

1 network address, e.g., 1413. The pay gateway server may forward the card authorization
2 request to the social pay server using the provided address, e.g., 1414.

3 **[0098]** With reference to FIGURE 14B, in some embodiments, the social pay server
4 may process the transaction so as to transfer funds for the purchase into an account
5 stored on an acquirer of the merchant. For example, the acquirer may be a financial
6 institution maintaining an account of the merchant. For example, the proceeds of
7 transactions processed by the merchant may be deposited into an account maintained
8 by at a server of the acquirer. In some embodiments, the social pay server may generate
9 a query, e.g., 1415, for issuer server(s) corresponding to the user-selected payment
10 options. For example, the user's account may be linked to one or more issuer financial
11 institutions ("issuers"), such as banking institutions, which issued the account(s) for the
12 user. For example, such accounts may include, but not be limited to: credit card, debit
13 card, prepaid card, checking, savings, money market, certificates of deposit, stored
14 (cash) value accounts and/or the like. Issuer server(s) of the issuer(s) may maintain
15 details of the user's account(s). In some embodiments, a database, e.g., a social pay
16 database, may store details of the issuer server(s) associated with the issuer(s). In some
17 embodiments, the social pay server may query a database, e.g., 1415, for a network
18 address of the issuer(s) server(s), for example by using a portion of a user payment card
19 number, or a user ID (such as an email address) as a keyword for the database query.

20 **[0099]** In response to obtaining the issuer server query, the social pay database
21 may provide, e.g., 1416, the requested issuer server data to the social pay server. In
22 some embodiments, the social pay server may utilize the issuer server data to generate
23 funds authorization request(s), e.g., 1417, for each of the issuer server(s) selected based

1 on the pre-defined payment settings associated with the user's virtual wallet, and/or the
2 user's payment options input, and provide the funds authorization request(s) to the
3 issuer server(s). In some embodiments, the funds authorization request(s) may include
4 details such as, but not limited to: the costs to the user involved in the transaction, card
5 account details of the user, user billing and/or shipping information, and/or the like. In
6 some embodiments, an issuer server may parse the authorization request(s), e.g., 1418,
7 and based on the request details may query a database, e.g., 1419, for data associated
8 with an account linked to the user.

9 **[00100]** In some embodiments, on obtaining the user account(s) data, e.g., 1420,
10 the issuer server may determine whether the user can pay for the transaction using
11 funds available in the account, e.g., 1421. For example, the issuer server may determine
12 whether the user has a sufficient balance remaining in the account, sufficient credit
13 associated with the account, and/or the like. Based on the determination, the issuer
14 server(s) may provide a funds authorization response, e.g., 1422, to the social pay server.
15 In some embodiments, if at least one issuer server determines that the user cannot pay
16 for the transaction using the funds available in the account, the social pay server may
17 request payment options again from the user (e.g., by providing an authorization fail
18 message to the user device and requesting the user device to provide new payment
19 options), and re-attempt authorization for the purchase transaction. In some
20 embodiments, if the number of failed authorization attempts exceeds a threshold, the
21 social pay server may abort the authorization process, and provide an "authorization
22 fail" message to the merchant server, user device and/or client.

1 **[00101]** In some embodiments, the social pay server may obtain the funds
2 authorization response including a notification of successful authorization, and parse
3 the message to extract authorization details. Upon determining that the user possesses
4 sufficient funds for the transaction, e.g., 1423, the social pay server may invoke a
5 component to provide value-add services for the user, e.g., 1423.

6 **[00102]** In some embodiments, the social pay server may forward a transaction
7 authorization response to the user wallet device, PoS client, and/or merchant server.
8 The merchant may parse, e.g., 1424, the transaction authorization response, and
9 determine from it that the user possesses sufficient funds in the card account to conduct
10 the transaction, e.g., 1425, option “Yes.” The merchant server may add a record of the
11 transaction for the user to a batch of transaction data relating to authorized
12 transactions. For example, the merchant may append the XML data pertaining to the
13 user transaction to an XML data file comprising XML data for transactions that have
14 been authorized for various users, e.g., 1426, and store the XML data file, e.g., 1427, in a
15 database. In some embodiments, the server may also generate a purchase receipt, e.g.,
16 1428, and provide the purchase receipt to the client. The client may render and display,
17 e.g., 1429, the purchase receipt for the user. In some embodiments, the user’s wallet
18 device may also provide a notification of successful authorization to the user. For
19 example, the PoS client/user device may render a webpage, electronic message, text /
20 SMS message, buffer a voicemail, emit a ring tone, and/or play an audio message, etc.,
21 and provide output including, but not limited to: sounds, music, audio, video, images,
22 tactile feedback, vibration alerts (e.g., on vibration-capable client devices such as a
23 smartphone etc.), and/or the like.

1 **[00103]** FIGURES 15A-B show data flow diagrams illustrating an example
2 purchase transaction clearance procedure in some embodiments of the SocialPay. With
3 reference to FIGURE 15A, in some embodiments, a merchant server, e.g., 1503a, may
4 initiate clearance of a batch of authorized transactions. For example, the merchant
5 server may generate a batch data request, e.g., 1511, and provide the request, to a
6 merchant database, e.g., 1503b. For example, the merchant server may utilize PHP/SQL
7 commands similar to the examples provided above to query a relational database. In
8 response to the batch data request, the database may provide the requested batch data,
9 e.g., 1512. The server may generate a batch clearance request, e.g., 1513, using the batch
10 data obtained from the database, and provide, e.g., 1514, the batch clearance request to
11 an acquirer server, e.g., 1507a. For example, the merchant server may provide a
12 HTTP(S) POST message including XML-formatted batch data in the message body for
13 the acquirer server. The acquirer server may generate, e.g., 1515, a batch payment
14 request using the obtained batch clearance request, and provide, e.g., 1518, the batch
15 payment request to the social pay server, e.g., 1505a. The social pay server may parse
16 the batch payment request, and extract the transaction data for each transaction stored
17 in the batch payment request, e.g., 1519. The social pay server may store the transaction
18 data, e.g., 1520, for each transaction in a database, e.g., social pay database 1505b. In
19 some embodiments, the social pay server may invoke a component to provide value-add
20 analytics services based on analysis of the transactions of the merchant for whom the
21 SocialPay is clearing purchase transactions. For example, the social pay server may
22 invoke a component such as the example card transaction-based analytics component
23 discussed above with reference to FIGURE 10. Thus, in some embodiments, the social

1 pay server may provide analytics-based value-added services for the merchant and/or
2 the merchant's users.

3 **[00104]** With reference to FIGURE 15B, in some embodiments, for each extracted
4 transaction, the social pay server may query, e.g., 1523, a database, e.g., social pay
5 database 1505b, for an address of an issuer server. For example, the social pay server
6 may utilize PHP/SQL commands similar to the examples provided above. The social
7 pay server may generate an individual payment request, e.g., 1525, for each transaction
8 for which it has extracted transaction data, and provide the individual payment request,
9 e.g., 1525, to the issuer server, e.g., 1506a. For example, the social pay server may
10 provide an individual payment request to the issuer server(s) as a HTTP(S) POST
11 message including XML-formatted data. An example listing of an individual payment
12 request 1525, substantially in the form of a HTTP(S) POST message including XML-
13 formatted data, is provided below:

```

14 POST /paymentrequest.php HTTP/1.1
15 Host: www.issuer.com
16 Content-Type: Application/XML
17 Content-Length: 788
18 <?XML version = "1.0" encoding = "UTF-8"?>
19 <pay_request>
20   <request_ID>CNI4ICNW2</request_ID>
21   <timestamp>2011-02-22 17:00:01</timestamp>
22   <pay_amount>$34.78</pay_amount>
23   <account_params>
24     <account_name>John Q. Public</account_name>
25     <account_type>credit</account_type>
26     <account_num>123456789012345</account_num>
27     <billing_address>123 Green St., Norman, OK 98765</billing_address>
28     <phone>123-456-7809</phone>
29     <sign>/jq</sign>
30   </account_params>
31   <merchant_params>
32     <merchant_id>3FBCR4INC</merchant_id>
33     <merchant_name>Books & Things, Inc.</merchant_name>
34     <merchant_auth_key>1NNF484MCP59CHB27365</merchant_auth_key>
35   </merchant_params>
36   <purchase_summary>
37     <num_products>1</num_products>
38     <product>
39       <product_summary>Book - XML for dummies</product_summary>
40       <product_quantity>1</product_quantity>

```

```
1         </product>
2     </purchase_summary>
3 </pay_request>
4
5
```

6 **[00105]** In some embodiments, the issuer server may generate a payment
7 command, e.g., 1527. For example, the issuer server may issue a command to deduct
8 funds from the user's account (or add a charge to the user's credit card account). The
9 issuer server may issue a payment command, e.g., 1527, to a database storing the user's
10 account information, e.g., user profile database 1506b. The issuer server may provide an
11 individual payment confirmation, e.g., 1528, to the social pay server, which may
12 forward, e.g., 1529, the funds transfer message to the acquirer server. An example
13 listing of an individual payment confirmation 1528, substantially in the form of a
14 HTTP(S) POST message including XML-formatted data, is provided below:

```
15 POST /clearance.php HTTP/1.1
16 Host: www.acquirer.com
17 Content-Type: Application/XML
18 Content-Length: 206
19 <?XML version = "1.0" encoding = "UTF-8"?>
20 <deposit_ack>
21     <request_ID>CNI4ICNW2</request_ID>
22     <clear_flag>true</clear_flag>
23     <timestamp>2011-02-22 17:00:02</timestamp>
24     <deposit_amount>$34.78</deposit_amount>
25 </deposit_ack>
26
27
```

28 **[00106]** In some embodiments, the acquirer server may parse the individual
29 payment confirmation, and correlate the transaction (e.g., using the request_ID field in
30 the example above) to the merchant. The acquirer server may then transfer the funds
31 specified in the funds transfer message to an account of the merchant. For example, the
32 acquirer server may query, e.g. 1530, an acquirer database 1507b for payment ledger
33 and/or merchant account data, e.g., 1531. The acquirer server may utilize payment
34 ledger and/or merchant account data from the acquirer database, along with the
35 individual payment confirmation, to generate updated payment ledger and/or merchant

1 account data, e.g., 1532. The acquirer server may then store, e.g., 1533, the updated
2 payment ledger and/or merchant account data to the acquire database.

3 **[00107]** FIGURES 16A-B show logic flow diagrams illustrating example aspects of
4 purchase transaction clearance in some embodiments of the SocialPay, e.g., a Purchase
5 Transaction Clearance (“PTC”) component 1600. With reference to FIGURE 16A, in
6 some embodiments, a merchant server may initiate clearance of a batch of authorized
7 transactions. For example, the merchant server may generate a batch data request, e.g.,
8 1601, and provide the request to a merchant database. In response to the batch data
9 request, the database may provide the requested batch data, e.g., 1602. The server may
10 generate a batch clearance request, e.g., 1603, using the batch data obtained from the
11 database, and provide the batch clearance request to an acquirer server. The acquirer
12 server may parse, e.g., 1604, the obtained batch clearance request, and generate, e.g.,
13 1607, a batch payment request using the obtained batch clearance request to provide,
14 the batch payment request to a social pay server. For example, the acquirer server may
15 query, e.g., 1605, an acquirer database for an address of a payment network server, and
16 utilize the obtained address, e.g., 1606, to forward the generated batch payment request
17 to the social pay server.

18 **[00108]** The social pay server may parse the batch payment request obtained from
19 the acquirer server, and extract the transaction data for each transaction stored in the
20 batch payment request, e.g., 1608. The social pay server may store the transaction data,
21 e.g., 1609, for each transaction in a social pay database. In some embodiments, the
22 social pay server may invoke a component, e.g., 1610, to provide analytics based on the
23 transactions of the merchant for whom purchase transaction are being cleared. For

1 example, the social pay server may invoke a component such as the example card
2 transaction-based analytics component discussed above with reference to FIGURE 10.

3 **[00109]** With reference to FIGURE 16B, in some embodiments, for each extracted
4 transaction, the social pay server may query, e.g., 1611, a social pay database for an
5 address of an issuer server. The social pay server may generate an individual payment
6 request, e.g., 1613, for each transaction for which it has extracted transaction data, and
7 provide the individual payment request to the issuer server. In some embodiments, the
8 issuer server may parse the individual payment request, e.g., 1614, and generate a
9 payment command, e.g., 1615, based on the parsed individual payment request. For
10 example, the issuer server may issue a command to deduct funds from the user's
11 account (or add a charge to the user's credit card account). The issuer server may issue
12 a payment command, e.g., 1615, to a database storing the user's account information,
13 e.g., a user profile database. The issuer server may provide an individual payment
14 confirmation, e.g., 1617, to the social pay server, which may forward, e.g., 1618, the
15 individual payment confirmation to the acquirer server.

16 **[00110]** In some embodiments, the acquirer server may parse the individual
17 payment confirmation, and correlate the transaction (e.g., using the request_ID field in
18 the example above) to the merchant. The acquirer server may then transfer the funds
19 specified in the funds transfer message to an account of the merchant. For example, the
20 acquirer server may query, e.g. 1619, an acquirer database for payment ledger and/or
21 merchant account data, e.g., 1620. The acquirer server may utilize payment ledger
22 and/or merchant account data from the acquirer database, along with the individual
23 payment confirmation, to generate updated payment ledger and/or merchant account

1 data, e.g., 1621. The acquirer server may then store, e.g., 1622, the updated payment
2 ledger and/or merchant account data to the acquire database.

3 **[00111]** FIGURE 17 shows a user interface diagram illustrating an overview of
4 example features of virtual wallet applications in some embodiments of the SocialPay.
5 FIGURE 17 shows an illustration of various exemplary features of a virtual wallet mobile
6 application 1700. Some of the features displayed include a wallet 1701, social integration
7 via TWITTER, FACEBOOK, etc., offers and loyalty 1703, snap mobile purchase 1704,
8 alerts 1705 and security, setting and analytics 1796. These features are explored in
9 further detail below.

10 **[00112]** FIGURES 18A-G show user interface diagrams illustrating example
11 features of virtual wallet applications in a shopping mode, in some embodiments of the
12 SocialPay. With reference to FIGURE 18A, some embodiments of the virtual wallet
13 mobile app facilitate and greatly enhance the shopping experience of consumers. A
14 variety of shopping modes, as shown in FIGURE 18A, may be available for a consumer
15 to peruse. In one implementation, for example, a user may launch the shopping mode by
16 selecting the shop icon 1810 at the bottom of the user interface. A user may type in an
17 item in the search field 1812 to search and/or add an item to a cart 1811. A user may also
18 use a voice activated shopping mode by saying the name or description of an item to be
19 searched and/or added to the cart into a microphone 1813. In a further implementation,
20 a user may also select other shopping options 1814 such as current items 1815, bills
21 1816, address book 1817, merchants 1818 and local proximity 1819.

22 **[00113]** In one embodiment, for example, a user may select the option current
23 items 1815, as shown in the left most user interface of FIGURE 18A. When the current

1 items 1815 option is selected, the middle user interface may be displayed. As shown, the
2 middle user interface may provide a current list of items 1815a-h in a user's shopping
3 cart 1811. A user may select an item, for example item 1815a, to view product description
4 1815j of the selected item and/or other items from the same merchant. The price and
5 total payable information may also be displayed, along with a QR code 1815k that
6 captures the information necessary to effect a snap mobile purchase transaction.

7 **[00114]** With reference to FIGURE 18B, in another embodiment, a user may select
8 the bills 1816 option. Upon selecting the bills 1816 option, the user interface may display
9 a list of bills and/or receipts 1816a-h from one or more merchants. Next to each of the
10 bills, additional information such as date of visit, whether items from multiple stores are
11 present, last bill payment date, auto-payment, number of items, and/or the like may be
12 displayed. In one example, the wallet shop bill 1816a dated January 20, 2911 may be
13 selected. The wallet shop bill selection may display a user interface that provides a
14 variety of information regarding the selected bill. For example, the user interface may
15 display a list of items 1816k purchased, <<1816i>>, a total number of items and the
16 corresponding value. For example, 7 items worth \$102.54 were in the selected wallet
17 shop bill. A user may now select any of the items and select buy again to add purchase
18 the items. The user may also refresh offers 1816j to clear any invalid offers from last
19 time and/or search for new offers that may be applicable for the current purchase. As
20 shown in FIGURE 18B, a user may select two items for repeat purchase. Upon addition,
21 a message 1816l may be displayed to confirm the addition of the two items, which makes
22 the total number of items in the cart 14.

1 **[00115]** With reference to FIGURE 18C, in yet another embodiment, a user may
2 select the address book option 1817 to view the address book 1817a which includes a list
3 of contacts 1817b and make any money transfers or payments. In one embodiment, the
4 address book may identify each contact using their names and available and/or
5 preferred modes of payment. For example, a contact Amanda G. may be paid via social
6 pay (e.g., via FACEBOOK) as indicated by the icon 1817c. In another example, money
7 may be transferred to Brian S. via QR code as indicated by the QR code icon 1817d. In
8 yet another example, Charles B. may accept payment via near field communication
9 1817e, Bluetooth 1817f and email 1817g. Payment may also be made via USB 1817h (e.g.,
10 by physically connecting two mobile devices) as well as other social channels such as
11 TWITTER.

12 **[00116]** In one implementation, a user may select Joe P. for payment. Joe P., as
13 shown in the user interface, has an email icon 1817g next to his name indicating that Joe
14 P. accepts payment via email. When his name is selected, the user interface may display
15 his contact information such as email, phone, etc. If a user wishes to make a payment to
16 Joe P. by a method other than email, the user may add another transfer mode 1817j to
17 his contact information and make a payment transfer. With reference to FIGURE 18D,
18 the user may be provided with a screen 1817k where the user can enter an amount to
19 send Joe, as well as add other text to provide Joe with context for the payment
20 transaction 1817l. The user can choose modes (e.g., SMS, email, social networking) via
21 which Joe may be contacted via graphical user interface elements, 1817m. As the user
22 types, the text entered may be provided for review within a GUI element 1817n. When
23 the user has completed entering in the necessary information, the user can press the
24 send button 1817o to send the social message to Joe. If Joe also has a virtual wallet

1 application, Joe may be able to review 1817p social pay message within the app, or
2 directly at the website of the social network (e.g., for Twitter™, Facebook®, etc.).
3 Messages may be aggregated from the various social networks and other sources (e.g.,
4 SMS, email). The method of redemption appropriate for each messaging mode may be
5 indicated along with the social pay message. In the illustration in FIGURE 18D, the
6 SMS 1817q Joe received indicates that Joe can redeem the \$5 obtained via SMS by
7 replying to the SMS and entering the hash tag value '#1234'. In the same illustration,
8 Joe has also received a message 1817r via Facebook®, which includes a URL link that
9 Joe can activate to initiate redemption of the \$25 payment.

10 **[00117]** With reference to FIGURE 18E, in some other embodiments, a user may
11 select merchants 1818 from the list of options in the shopping mode to view a select list
12 of merchants 1818a-e. In one implementation, the merchants in the list may be affiliated
13 to the wallet, or have affinity relationship with the wallet. In another implementation,
14 the merchants may include a list of merchants meeting a user-defined or other criteria.
15 For example, the list may be one that is curated by the user, merchants where the user
16 most frequently shops or spends more than an x amount of sum or shopped for three
17 consecutive months, and/or the like. In one implementation, the user may further select
18 one of the merchants, Amazon 1818a for example. The user may then navigate through
19 the merchant's listings to find items of interest such as 1818f-j. Directly through the
20 wallet and without visiting the merchant site from a separate page, the user may make a
21 selection of an item 1818j from the catalog of Amazon 1818a. As shown in the right most
22 user interface of FIGURE 18D, the selected item may then be added to cart. The
23 message 1818k indicates that the selected item has been added to the cart, and updated
24 number of items in the cart is now 13.

1 **[00118]** With reference to FIGURE 18F, in one embodiment, there may be a local
2 proximity option 1819 which may be selected by a user to view a list of merchants that
3 are geographically in close proximity to the user. For example, the list of merchants
4 1819a-e may be the merchants that are located close to the user. In one implementation,
5 the mobile application may further identify when the is in a store based on the user's
6 location. For example, position icon 1819d may be displayed next to a store (e.g.,
7 Walgreens) when the user is in close proximity to the store. In one implementation, the
8 mobile application may refresh its location periodically in case the user moved away
9 from the store (e.g., Walgreens). In a further implementation, the user may navigate the
10 offerings of the selected Walgreens store through the mobile application. For example,
11 the user may navigate, using the mobile application, to items 1819f-j available on aisle 5
12 of Walgreens. In one implementation, the user may select corn 1819i from his or her
13 mobile application to add to cart 1819k.

14 **[00119]** With reference to FIGURE 18G, in another embodiment, the local
15 proximity option 1819 may include a store map and a real time map features among
16 others. For example, upon selecting the Walgreens store, the user may launch an aisle
17 map 1819l which displays a map 1819m showing the organization of the store and the
18 position of the user (indicated by a yellow circle). In one implementation, the user may
19 easily configure the map to add one or more other users (e.g., user's kids) to share each
20 other's location within the store. In another implementation, the user may have the
21 option to launch a "store view" similar to street views in maps. The store view 1819n
22 may display images/video of the user's surrounding. For example, if the user is about to
23 enter aisle 5, the store view map may show the view of aisle 5. Further the user may

1 manipulate the orientation of the map using the navigation tool 18190 to move the store
2 view forwards, backwards, right, left as well clockwise and counterclockwise rotation

3 **[00120]** FIGURES 19A-F show user interface diagrams illustrating example
4 features of virtual wallet applications in a payment mode, in some embodiments of the
5 SocialPay. With reference to FIGURE 19A, in one embodiment, the wallet mobile
6 application may provide a user with a number of options for paying for a transaction via
7 the wallet mode 1910. In one implementation, an example user interface 1911 for making
8 a payment is shown. The user interface may clearly identify the amount 1912 and the
9 currency 1913 for the transaction. The amount may be the amount payable and the
10 currency may include real currencies such as dollars and euros, as well as virtual
11 currencies such as reward points. The amount of the transaction 1914 may also be
12 prominently displayed on the user interface. The user may select the funds tab 1916 to
13 select one or more forms of payment 1917, which may include various credit, debit, gift,
14 rewards and/or prepaid cards. The user may also have the option of paying, wholly or in
15 part, with reward points. For example, the graphical indicator 1918 on the user interface
16 shows the number of points available, the graphical indicator 1919 shows the number of
17 points to be used towards the amount due 234.56 and the equivalent 1920 of the
18 number of points in a selected currency (USD, for example).

19 **[00121]** In one implementation, the user may combine funds from multiple
20 sources to pay for the transaction. The amount 1915 displayed on the user interface may
21 provide an indication of the amount of total funds covered so far by the selected forms of
22 payment (e.g., Discover card and rewards points). The user may choose another form of
23 payment or adjust the amount to be debited from one or more forms of payment until

1 the amount 1915 matches the amount payable 1914. Once the amounts to be debited
2 from one or more forms of payment are finalized by the user, payment authorization
3 may begin.

4 **[00122]** In one implementation, the user may select a secure authorization of the
5 transaction by selecting the cloak button 1922 to effectively cloak or anonymize some
6 (e.g., pre-configured) or all identifying information such that when the user selects pay
7 button 1921, the transaction authorization is conducted in a secure and anonymous
8 manner. In another implementation, the user may select the pay button 1921 which may
9 use standard authorization techniques for transaction processing. In yet another
10 implementation, when the user selects the social button 1923, a message regarding the
11 transaction may be communicated to one of more social networks (set up by the user)
12 which may post or announce the purchase transaction in a social forum such as a wall
13 post or a tweet. In one implementation, the user may select a social payment processing
14 option 1923. The indicator 1924 may show the authorizing and sending social share data
15 in progress.

16 **[00123]** In another implementation, a restricted payment mode 1925 may be
17 activated for certain purchase activities such as prescription purchases. The mode may
18 be activated in accordance with rules defined by issuers, insurers, merchants, payment
19 processor and/or other entities to facilitate processing of specialized goods and services.
20 In this mode, the user may scroll down the list of forms of payments 1926 under the
21 funds tab to select specialized accounts such as a flexible spending account (FSA) 1927,
22 health savings account (HAS), and/or the like and amounts to be debited to the selected

1 accounts. In one implementation, such restricted payment mode 1925 processing may
2 disable social sharing of purchase information.

3 **[00124]** In one embodiment, the wallet mobile application may facilitate importing
4 of funds via the import funds user interface 1928. For example, a user who is
5 unemployed may obtain unemployment benefit fund 1929 via the wallet mobile
6 application. In one implementation, the entity providing the funds may also configure
7 rules for using the fund as shown by the processing indicator message 1930. The wallet
8 may read and apply the rules prior, and may reject any purchases with the
9 unemployment funds that fail to meet the criteria set by the rules. Example criteria may
10 include, for example, merchant category code (MCC), time of transaction, location of
11 transaction, and/or the like. As an example, a transaction with a grocery merchant
12 having MCC 5411 may be approved, while a transaction with a bar merchant having an
13 MCC 5813 may be refused.

14 **[00125]** With reference to FIGURE 19B, in one embodiment, the wallet mobile
15 application may facilitate dynamic payment optimization based on factors such as user
16 location, preferences and currency value preferences among others. For example, when
17 a user is in the United States, the country indicator 1931 may display a flag of the United
18 States and may set the currency 1933 to the United States. In a further implementation,
19 the wallet mobile application may automatically rearrange the order in which the forms
20 of payments 1935 are listed to reflect the popularity or acceptability of various forms of
21 payment. In one implementation, the arrangement may reflect the user's preference,
22 which may not be changed by the wallet mobile application.

1 **[00126]** Similarly, when a German user operates a wallet in Germany, the mobile
2 wallet application user interface may be dynamically updated to reflect the country of
3 operation 1932 and the currency 1934. In a further implementation, the wallet
4 application may rearrange the order in which different forms of payment 1936 are listed
5 based on their acceptance level in that country. Of course, the order of these forms of
6 payments may be modified by the user to suit his or her own preferences.

7 **[00127]** With reference to FIGURE 19C, in one embodiment, the payee tab 1937 in
8 the wallet mobile application user interface may facilitate user selection of one or more
9 payees receiving the funds selected in the funds tab. In one implementation, the user
10 interface may show a list of all payees 1938 with whom the user has previously
11 transacted or available to transact. The user may then select one or more payees. The
12 payees 1938 may include larger merchants such as Amazon.com Inc., and individuals
13 such as Jane P. Doe. Next to each payee name, a list of accepted payment modes for the
14 payee may be displayed. In one implementation, the user may select the payee Jane P.
15 Doe 1939 for receiving payment. Upon selection, the user interface may display
16 additional identifying information relating to the payee.

17 **[00128]** With reference to FIGURE 19D, in one embodiment, the mode tab 1940
18 may facilitate selection of a payment mode accepted by the payee. A number of payment
19 modes may be available for selection. Example modes include, blue tooth 1941, wireless
20 1942, snap mobile by user-obtained QR code 1943, secure chip 1944, TWITTER 1945,
21 near-field communication (NFC) 1946, cellular 1947, snap mobile by user-provided QR
22 code 1948, USB 1949 and FACEBOOK 1950, among others. In one implementation, only

1 the payment modes that are accepted by the payee may be selectable by the user. Other
2 non-accepted payment modes may be disabled.

3 **[00129]** With reference to FIGURE 19E, in one embodiment, the offers tab 1951
4 may provide real-time offers that are relevant to items in a user's cart for selection by
5 the user. The user may select one or more offers from the list of applicable offers 1952
6 for redemption. In one implementation, some offers may be combined, while others
7 may not. When the user selects an offer that may not be combined with another offer,
8 the unselected offers may be disabled. In a further implementation, offers that are
9 recommended by the wallet application's recommendation engine may be identified by
10 an indicator such as the one shown by 1953. In a further implementation, the user may
11 read the details of the offer by expanding the offer row as shown by 1954 in the user
12 interface.

13 **[00130]** With reference to FIGURE 19F, in one embodiment, the social tab 1955
14 may facilitate integration of the wallet application with social channels 1956. In one
15 implementation, a user may select one or more social channels 1956 and may sign in to
16 the selected social channel from the wallet application by providing to the wallet
17 application the social channel user name and password 1957 and signing in 1958. The
18 user may then use the social button 1959 to send or receive money through the
19 integrated social channels. In a further implementation, the user may send social share
20 data such as purchase information or links through integrated social channels. In
21 another embodiment, the user supplied login credentials may allow SocialPay to engage
22 in interception parsing (e.g., as described in paragraphs [0045], [0052] and [0064]).

1 **[00131]** FIGURE 20 shows a user interface diagram illustrating example features
2 of virtual wallet applications, in a history mode, in some embodiments of the SocialPay.
3 In one embodiment, a user may select the history mode 2010 to view a history of prior
4 purchases and perform various actions on those prior purchases. For example, a user
5 may enter a merchant identifying information such as name, product, MCC, and/or the
6 like in the search bar 2011. In another implementation, the user may use voice activated
7 search feature by clicking on the microphone icon 2014. The wallet application may
8 query the storage areas in the mobile device or elsewhere (e.g., one or more databases
9 and/or tables remote from the mobile device) for transactions matching the search
10 keywords. The user interface may then display the results of the query such as
11 transaction 2015. The user interface may also identify the date 2012 of the transaction,
12 the merchants and items 2013 relating to the transaction, a barcode of the receipt
13 confirming that a transaction was made, the amount of the transaction and any other
14 relevant information.

15 **[00132]** In one implementation, the user may select a transaction, for example
16 transaction 2015, to view the details of the transaction. For example, the user may view
17 the details of the items associated with the transaction and the amounts 2016 of each
18 item. In a further implementation, the user may select the show option 2017 to view
19 actions 2018 that the user may take in regards to the transaction or the items in the
20 transaction. For example, the user may add a photo to the transaction (e.g., a picture of
21 the user and the iPad the user bought). In a further implementation, if the user
22 previously shared the purchase via social channels, a post including the photo may be
23 generated and sent to the social channels for publishing. In one implementation, any
24 sharing may be optional, and the user, who did not share the purchase via social

1 channels, may still share the photo through one or more social channels of his or her
2 choice directly from the history mode of the wallet application. In another
3 implementation, the user may add the transaction to a group such as company expense,
4 home expense, travel expense or other categories set up by the user. Such grouping may
5 facilitate year-end accounting of expenses, submission of work expense reports,
6 submission for value added tax (VAT) refunds, personal expenses, and/or the like. In yet
7 another implementation, the user may buy one or more items purchased in the
8 transaction. The user may then execute a transaction without going to the merchant
9 catalog or site to find the items. In a further implementation, the user may also cart one
10 or more items in the transaction for later purchase.

11 **[00133]** The history mode, in another embodiment, may offer facilities for
12 obtaining and displaying ratings 2019 of the items in the transaction. The source of the
13 ratings may be the user, the user's friends (e.g., from social channels, contacts, etc.),
14 reviews aggregated from the web, and/or the like. The user interface in some
15 implementations may also allow the user to post messages to other users of social
16 channels (e.g., TWITTER or FACEBOOK). For example, the display area 2020 shows
17 FACEBOOK message exchanges between two users. In one implementation, a user may
18 share a link via a message 2021. Selection of such a message having embedded link to a
19 product may allow the user to view a description of the product and/or purchase the
20 product directly from the history mode.

21 **[00134]** In one embodiment, the history mode may also include facilities for
22 exporting receipts. The export receipts pop up 2022 may provide a number of options
23 for exporting the receipts of transactions in the history. For example, a user may use one

1 or more of the options 2025 which include save (to local mobile memory, to server, to a
2 cloud account, and/or the like), print to a printer, fax, email, and/or the like. The user
3 may utilize his or her address book 2023 to look up email or fax number for exporting.
4 The user may also specify format options 2024 for exporting receipts. Example format
5 options may include, without limitation, text files (.doc, .txt, .rtf, iif, etc.), spreadsheet
6 (.csv, .xls, etc.), image files (.jpg, .tff, .png, etc.), portable document format (.pdf),
7 postscript (.ps), and/or the like. The user may then click or tap the export button 2027
8 to initiate export of receipts.

9 **[00135]** FIGURES 21A-E show user interface diagrams illustrating example
10 features of virtual wallet applications in a snap mode, in some embodiments of the
11 SocialPay. With reference to FIGURE 21A, in one embodiment, a user may select the
12 snap mode 2110 to access its snap features. The snap mode may handle any machine-
13 readable representation of data. Examples of such data may include linear and 2D bar
14 codes such as UPC code and QR codes. These codes may be found on receipts, product
15 packaging, and/or the like. The snap mode may also process and handle pictures of
16 receipts, products, offers, credit cards or other payment devices, and/or the like. An
17 example user interface in snap mode is shown in FIGURE 21A. A user may use his or her
18 mobile phone to take a picture of a QR code 2115 and/or a barcode 2114. In one
19 implementation, the bar 1213 and snap frame 2115 may assist the user in snapping codes
20 properly. For example, the snap frame 2115, as shown, does not capture the entirety of
21 the code 2116. As such, the code captured in this view may not be resolvable as
22 information in the code may be incomplete. This is indicated by the message on the bar
23 2113 that indicates that the snap mode is still seeking the code. When the code 2116 is
24 completely framed by the snap frame 2115, the bar message may be updated to, for

1 example, “snap found.” Upon finding the code, in one implementation, the user may
2 initiate code capture using the mobile device camera. In another implementation, the
3 snap mode may automatically snap the code using the mobile device camera.

4 **[00136]** With reference to FIGURE 21B, in one embodiment, the snap mode may
5 facilitate payment reallocation post transaction. For example, a user may buy grocery
6 and prescription items from a retailer Acme Supermarket. The user may, inadvertently
7 or for ease of checkout for example, use his or her Visa card to pay for both grocery and
8 prescription items. However, the user may have an FSA account that could be used to
9 pay for prescription items, and which would provide the user tax benefits. In such a
10 situation, the user may use the snap mode to initiate transaction reallocation.

11 **[00137]** As shown, the user may enter a search term (e.g., bills) in the search bar
12 2121. The user may then identify in the tab 2122 the receipt 2123 the user wants to
13 reallocate. Alternatively, the user may directly snap a picture of a barcode on a receipt,
14 and the snap mode may generate and display a receipt 2123 using information from the
15 barcode. The user may now reallocate 2125. In some implementations, the user may also
16 dispute the transaction 2124 or archive the receipt 2126.

17 **[00138]** In one implementation, when the reallocate button 2125 is selected, the
18 wallet application may perform optical character recognition (OCR) of the receipt. Each
19 of the items in the receipt may then be examined to identify one or more items which
20 could be charged to which payment device or account for tax or other benefits such as
21 cash back, reward points, etc. In this example, there is a tax benefit if the prescription
22 medication charged to the user’s Visa card is charged to the user’s FSA. The wallet
23 application may then perform the reallocation as the back end. The reallocation process

1 may include the wallet contacting the payment processor to credit the amount of the
2 prescription medication to the Visa card and debit the same amount to the user's FSA
3 account. In an alternate implementation, the payment processor (e.g., Visa or
4 MasterCard) may obtain and OCR the receipt, identify items and payment accounts for
5 reallocation and perform the reallocation. In one implementation, the wallet application
6 may request the user to confirm reallocation of charges for the selected items to another
7 payment account. The receipt 2127 may be generated after the completion of the
8 reallocation process. As discussed, the receipt shows that some charges have been
9 moved from the Visa account to the FSA.

10 **[00139]** With reference to FIGURE 21C, in one embodiment, the snap mode may
11 facilitate payment via pay code such as barcodes or QR codes. For example, a user may
12 snap a QR code of a transaction that is not yet complete. The QR code may be displayed
13 at a merchant POS terminal, a web site, or a web application and may be encoded with
14 information identifying items for purchase, merchant details and other relevant
15 information. When the user snaps such as a QR code, the snap mode may decode the
16 information in the QR code and may use the decoded information to generate a receipt
17 2132. Once the QR code is identified, the navigation bar 2131 may indicate that the pay
18 code is identified. The user may now have an option to add to cart 2133, pay with a
19 default payment account 2134 or pay with wallet 2135.

20 **[00140]** In one implementation, the user may decide to pay with default 2134. The
21 wallet application may then use the user's default method of payment, in this example
22 the wallet, to complete the purchase transaction. Upon completion of the transaction, a
23 receipt may be automatically generated for proof of purchase. The user interface may

1 also be updated to provide other options for handling a completed transaction. Example
2 options include social 2137 to share purchase information with others, reallocate 2138
3 as discussed with regard to FIGURE 21B, and archive 2139 to store the receipt.

4 **[00141]** With reference to FIGURE 21D, in one embodiment, the snap mode may
5 also facilitate offer identification, application and storage for future use. For example, in
6 one implementation, a user may snap an offer code 2141 (e.g., a bar code, a QR code,
7 and/or the like). The wallet application may then generate an offer text 2142 from the
8 information encoded in the offer code. The user may perform a number of actions on the
9 offer code. For example, the user use the find button 2143 to find all merchants who
10 accept the offer code, merchants in the proximity who accept the offer code, products
11 from merchants that qualify for the offer code, and/or the like. The user may also apply
12 the offer code to items that are currently in the cart using the add to cart button 2144.
13 Furthermore, the user may also save the offer for future use by selecting the save button
14 2145.

15 **[00142]** In one implementation, after the offer or coupon 2146 is applied, the user
16 may have the option to find qualifying merchants and/or products using find, the user
17 may go to the wallet using 2148, and the user may also save the offer or coupon 2146 for
18 later use.

19 **[00143]** With reference to FIGURE 21E, in one embodiment, the snap mode may
20 also offer facilities for adding a funding source to the wallet application. In one
21 implementation, a pay card such as a credit card, debit card, pre-paid card, smart card
22 and other pay accounts may have an associated code such as a bar code or QR code.
23 Such a code may have encoded therein pay card information including, but not limited

1 to, name, address, pay card type, pay card account details, balance amount, spending
2 limit, rewards balance, and/or the like. In one implementation, the code may be found
3 on a face of the physical pay card. In another implementation, the code may be obtained
4 by accessing an associated online account or another secure location. In yet another
5 implementation, the code may be printed on a letter accompanying the pay card. A user,
6 in one implementation, may snap a picture of the code. The wallet application may
7 identify the pay card 2151 and may display the textual information 2152 encoded in the
8 pay card. The user may then perform verification of the information 2152 by selecting
9 the verify button 2153. In one implementation, the verification may include contacting
10 the issuer of the pay card for confirmation of the decoded information 2152 and any
11 other relevant information. In one implementation, the user may add the pay card to the
12 wallet by selecting the add to wallet button 2154. The instruction to add the pay card to
13 the wallet may cause the pay card to appear as one of the forms of payment under the
14 funds tab 1916 discussed in FIGURE 19A. The user may also cancel importing of the pay
15 card as a funding source by selecting the cancel button 2155. When the pay card has
16 been added to the wallet, the user interface may be updated to indicate that the
17 importing is complete via the notification display 2156. The user may then access the
18 wallet 2157 to begin using the added pay card as a funding source.

19 **[00144]** FIGURE 22 shows a user interface diagram illustrating example features
20 of virtual wallet applications, in an offers mode, in some embodiments of the SocialPay.
21 In some implementations, the SocialPay may allow a user to search for offers for
22 products and/or services from within the virtual wallet mobile application. For
23 example, the user may enter text into a graphical user interface (“GUI”) element 2211, or
24 issue voice commands by activating GUI element 2212 and speaking commands into the

1 device. In some implementations, the SocialPay may provide offers based on the user's
2 prior behavior, demographics, current location, current cart selection or purchase items,
3 and/or the like. For example, if a user is in a brick-and-mortar store, or an online
4 shopping website, and leaves the (virtual) store, then the merchant associated with the
5 store may desire to provide a sweetener deal to entice the consumer back into the
6 (virtual) store. The merchant may provide such an offer 2213. For example, the offer
7 may provide a discount, and may include an expiry time. In some implementations,
8 other users may provide gifts (e.g., 2214) to the user, which the user may redeem. In
9 some implementations, the offers section may include alerts as to payment of funds
10 outstanding to other users (e.g., 2215). In some implementations, the offers section may
11 include alerts as to requesting receipt of funds from other users (e.g., 2216). For
12 example, such a feature may identify funds receivable from other applications (e.g.,
13 mail, calendar, tasks, notes, reminder programs, alarm, etc.), or by a manual entry by
14 the user into the virtual wallet application. In some implementations, the offers section
15 may provide offers from participating merchants in the SocialPay, e.g., 2217-2219, 2220.
16 These offers may sometimes be assembled using a combination of participating
17 merchants, e.g., 2217. In some implementations, the SocialPay itself may provide offers
18 for users contingent on the user utilizing particular payment forms from within the
19 virtual wallet application, e.g., 2220.

20 **[00145]** FIGURES 23A-B show user interface diagrams illustrating example
21 features of virtual wallet applications, in a security and privacy mode, in some
22 embodiments of the SocialPay. With reference to FIGURE 23A, in some
23 implementations, the user may be able to view and/or modify the user profile and/or
24 settings of the user, e.g., by activating a user interface element. For example, the user

1 may be able to view/modify a user name (e.g., 2311a-b), account number (e.g., 2312a-
2 b), user security access code (e.g., 2313-b), user pin (e.g., 2314-b), user address (e.g.,
3 2315-b), social security number associated with the user (e.g., 2316-b), current device
4 GPS location (e.g., 2317-b), user account of the merchant in whose store the user
5 currently is (e.g., 2318-b), the user's rewards accounts (e.g., 2319-b), and/or the like. In
6 some implementations, the user may be able to select which of the data fields and their
7 associated values should be transmitted to facilitate the purchase transaction, thus
8 providing enhanced data security for the user. For example, in the example illustration
9 in FIGURE 23A, the user has selected the name 2311a, account number 2312a, security
10 code 2313a, merchant account ID 2318a and rewards account ID 2319a as the fields to
11 be sent as part of the notification to process the purchase transaction. In some
12 implementations, the user may toggle the fields and/or data values that are sent as part
13 of the notification to process the purchase transactions. In some implementations, the
14 app may provide multiple screens of data fields and/or associated values stored for the
15 user to select as part of the purchase order transmission. In some implementations, the
16 app may provide the SocialPay with the GPS location of the user. Based on the GPS
17 location of the user, the SocialPay may determine the context of the user (e.g., whether
18 the user is in a store, doctor's office, hospital, postal service office, etc.). Based on the
19 context, the user app may present the appropriate fields to the user, from which the user
20 may select fields and/or field values to send as part of the purchase order transmission.

21 **[00146]** For example, a user may go to doctor's office and desire to pay the co-pay
22 for doctor's appointment. In addition to basic transactional information such as
23 account number and name, the app may provide the user the ability to select to transfer
24 medical records, health information, which may be provided to the medical provider,

1 insurance company, as well as the transaction processor to reconcile payments between
2 the parties. In some implementations, the records may be sent in a Health Insurance
3 Portability and Accountability Act (HIPAA)-compliant data format and encrypted, and
4 only the recipients who are authorized to view such records may have appropriate
5 decryption keys to decrypt and view the private user information.

6 **[00147]** With reference to FIGURE 23B, in some implementations, the app
7 executing on the user's device may provide a "VerifyChat" feature for fraud prevention.
8 For example, the SocialPay may detect an unusual and/or suspicious transaction. The
9 SocialPay may utilize the VerifyChat feature to communicate with the user, and verify
10 the authenticity of the originator of the purchase transaction. In various
11 implementations, the SocialPay may send electronic mail message, text (SMS)
12 messages, Facebook® messages, Twitter™ tweets, text chat, voice chat, video chat (e.g.,
13 Apple FaceTime), and/or the like to communicate with the user. For example, the
14 SocialPay may initiate a video challenge for the user, e.g., 2321. For example, the user
15 may need to present him/her-self via a video chat, e.g., 2322. In some implementations,
16 a customer service representative, e.g., agent 2324, may manually determine the
17 authenticity of the user using the video of the user. In some implementations, the
18 SocialPay may utilize face, biometric and/or like recognition (e.g., using pattern
19 classification techniques) to determine the identity of the user. In some
20 implementations, the app may provide reference marker (e.g., cross-hairs, target box,
21 etc.), e.g., 2323, so that the user may the video to facilitate the SocialPay's automated
22 recognition of the user. In some implementations, the user may not have initiated the
23 transaction, e.g., the transaction is fraudulent. In such implementations, the user may

1 cancel the challenge. The SocialPay may then cancel the transaction, and/or initiate
2 fraud investigation procedures on behalf of the user.

3 **[00148]** In some implementations, the SocialPay may utilize a text challenge
4 procedure to verify the authenticity of the user, e.g., 2325. For example, the SocialPay
5 may communicate with the user via text chat, SMS messages, electronic mail,
6 Facebook® messages, Twitter™ tweets, and/or the like. The SocialPay may pose a
7 challenge question, e.g., 2326, for the user. The app may provide a user input interface
8 element(s) (e.g., virtual keyboard 2328) to answer the challenge question posed by the
9 SocialPay. In some implementations, the challenge question may be randomly selected
10 by the SocialPay automatically; in some implementations, a customer service
11 representative may manually communicate with the user. In some implementations,
12 the user may not have initiated the transaction, e.g., the transaction is fraudulent. In
13 such implementations, the user may cancel the text challenge. The SocialPay may
14 cancel the transaction, and/or initiate fraud investigation on behalf of the user.

15 SocialPay Controller

16 **[00149]** FIGURE 24 shows a block diagram illustrating embodiments of a
17 SocialPay controller 2401. In this embodiment, the SocialPay controller 2401 may serve
18 to aggregate, process, store, search, serve, identify, instruct, generate, match, and/or
19 facilitate interactions with a computer through various technologies, and/or other
20 related data.

21 **[00150]** Typically, users, e.g., 2433a, which may be people and/or other systems,
22 may engage information technology systems (e.g., computers) to facilitate information

1 processing. In turn, computers employ processors to process information; such
2 processors 2403 may be referred to as central processing units (CPU). One form of
3 processor is referred to as a microprocessor. CPUs use communicative circuits to pass
4 binary encoded signals acting as instructions to enable various operations. These
5 instructions may be operational and/or data instructions containing and/or referencing
6 other instructions and data in various processor accessible and operable areas of
7 memory 2429 (e.g., registers, cache memory, random access memory, etc.). Such
8 communicative instructions may be stored and/or transmitted in batches (e.g., batches
9 of instructions) as programs and/or data components to facilitate desired operations.
10 These stored instruction codes, e.g., programs, may engage the CPU circuit components
11 and other motherboard and/or system components to perform desired operations. One
12 type of program is a computer operating system, which, may be executed by CPU on a
13 computer; the operating system enables and facilitates users to access and operate
14 computer information technology and resources. Some resources that may be employed
15 in information technology systems include: input and output mechanisms through
16 which data may pass into and out of a computer; memory storage into which data may
17 be saved; and processors by which information may be processed. These information
18 technology systems may be used to collect data for later retrieval, analysis, and
19 manipulation, which may be facilitated through a database program. These information
20 technology systems provide interfaces that allow users to access and operate various
21 system components.

22 **[00151]** In one embodiment, the SocialPay controller 2401 may be connected to
23 and/or communicate with entities such as, but not limited to: one or more users from
24 user input devices 2411; peripheral devices 2412; an optional cryptographic processor

1 device 2428; and/or a communications network 2413. For example, the SocialPay
2 controller 2401 may be connected to and/or communicate with users, e.g., 2433a,
3 operating client device(s), e.g., 2433b, including, but not limited to, personal
4 computer(s), server(s) and/or various mobile device(s) including, but not limited to,
5 cellular telephone(s), smartphone(s) (e.g., iPhone®, Blackberry®, Android OS-based
6 phones etc.), tablet computer(s) (e.g., Apple iPad™, HP Slate™, Motorola Xoom™, etc.),
7 eBook reader(s) (e.g., Amazon Kindle™, Barnes and Noble's Nook™ eReader, etc.),
8 laptop computer(s), notebook(s), netbook(s), gaming console(s) (e.g., XBOX Live™,
9 Nintendo® DS, Sony PlayStation® Portable, etc.), portable scanner(s), and/or the like.

10 **[00152]** Networks are commonly thought to comprise the interconnection and
11 interoperation of clients, servers, and intermediary nodes in a graph topology. It should
12 be noted that the term “server” as used throughout this application refers generally to a
13 computer, other device, program, or combination thereof that processes and responds to
14 the requests of remote users across a communications network. Servers serve their
15 information to requesting “clients.” The term “client” as used herein refers generally to a
16 computer, program, other device, user and/or combination thereof that is capable of
17 processing and making requests and obtaining and processing any responses from
18 servers across a communications network. A computer, other device, program, or
19 combination thereof that facilitates, processes information and requests, and/or
20 furthers the passage of information from a source user to a destination user is
21 commonly referred to as a “node.” Networks are generally thought to facilitate the
22 transfer of information from source points to destinations. A node specifically tasked
23 with furthering the passage of information from a source to a destination is commonly
24 called a “router.” There are many forms of networks such as Local Area Networks

1 (LANs), Pico networks, Wide Area Networks (WANs), Wireless Networks (WLANs), etc.
2 For example, the Internet is generally accepted as being an interconnection of a
3 multitude of networks whereby remote clients and servers may access and interoperate
4 with one another.

5 **[00153]** The SocialPay controller 2401 may be based on computer systems that
6 may comprise, but are not limited to, components such as: a computer systemization
7 2402 connected to memory 2429.

8 **Computer Systemization**

9 **[00154]** A computer systemization 2402 may comprise a clock 2430, central
10 processing unit ("CPU(s)" and/or "processor(s)" (these terms are used interchangeable
11 throughout the disclosure unless noted to the contrary)) 2403, a memory 2429 (e.g., a
12 read only memory (ROM) 2406, a random access memory (RAM) 2405, etc.), and/or an
13 interface bus 2407, and most frequently, although not necessarily, are all interconnected
14 and/or communicating through a system bus 2404 on one or more (mother)board(s)
15 2402 having conductive and/or otherwise transportive circuit pathways through which
16 instructions (e.g., binary encoded signals) may travel to effectuate communications,
17 operations, storage, etc. The computer systemization may be connected to a power
18 source 2486; e.g., optionally the power source may be internal. Optionally, a
19 cryptographic processor 2426 and/or transceivers (e.g., ICs) 2474 may be connected to
20 the system bus. In another embodiment, the cryptographic processor and/or
21 transceivers may be connected as either internal and/or external peripheral devices
22 2412 via the interface bus I/O. In turn, the transceivers may be connected to antenna(s)
23 2475, thereby effectuating wireless transmission and reception of various

1 communication and/or sensor protocols; for example the antenna(s) may connect to: a
2 Texas Instruments WiLink WL1283 transceiver chip (e.g., providing 802.11n, Bluetooth
3 3.0, FM, global positioning system (GPS) (thereby allowing SocialPay controller to
4 determine its location)); Broadcom BCM4329FKUBG transceiver chip (e.g., providing
5 802.11n, Bluetooth 2.1 + EDR, FM, etc.); a Broadcom BCM4750IUB8 receiver chip (e.g.,
6 GPS); an Infineon Technologies X-Gold 618-PMB9800 (e.g., providing 2G/3G
7 HSDPA/HSUPA communications); and/or the like. The system clock typically has a
8 crystal oscillator and generates a base signal through the computer systemization's
9 circuit pathways. The clock is typically coupled to the system bus and various clock
10 multipliers that will increase or decrease the base operating frequency for other
11 components interconnected in the computer systemization. The clock and various
12 components in a computer systemization drive signals embodying information
13 throughout the system. Such transmission and reception of instructions embodying
14 information throughout a computer systemization may be commonly referred to as
15 communications. These communicative instructions may further be transmitted,
16 received, and cause of return and/or reply communications beyond the instant
17 computer systemization to: communications networks, input devices, other computer
18 systemizations, peripheral devices, and/or the like. It should be understood that in
19 alternative embodiments, any of the above components may be connected directly to
20 one another, connected to the CPU, and/or organized in numerous variations employed
21 as exemplified by various computer systems.

22 **[00155]** The CPU comprises at least one high-speed data processor adequate to
23 execute program components for executing user and/or system-generated requests.
24 Often, the processors themselves will incorporate various specialized processing units,

1 such as, but not limited to: integrated system (bus) controllers, memory management
2 control units, floating point units, and even specialized processing sub-units like
3 graphics processing units, digital signal processing units, and/or the like. Additionally,
4 processors may include internal fast access addressable memory, and be capable of
5 mapping and addressing memory 2429 beyond the processor itself; internal memory
6 may include, but is not limited to: fast registers, various levels of cache memory (e.g.,
7 level 1, 2, 3, etc.), RAM, etc. The processor may access this memory through the use of a
8 memory address space that is accessible via instruction address, which the processor
9 can construct and decode allowing it to access a circuit path to a specific memory
10 address space having a memory state. The CPU may be a microprocessor such as:
11 AMD's Athlon, Duron and/or Opteron; ARM's application, embedded and secure
12 processors; IBM and/or Motorola's DragonBall and PowerPC; IBM's and Sony's Cell
13 processor; Intel's Celeron, Core (2) Duo, Itanium, Pentium, Xeon, and/or XScale;
14 and/or the like processor(s). The CPU interacts with memory through instruction
15 passing through conductive and/or transportive conduits (e.g., (printed) electronic
16 and/or optic circuits) to execute stored instructions (i.e., program code) according to
17 conventional data processing techniques. Such instruction passing facilitates
18 communication within the SocialPay controller and beyond through various interfaces.
19 Should processing requirements dictate a greater amount speed and/or capacity,
20 distributed processors (e.g., Distributed SocialPay), mainframe, multi-core, parallel,
21 and/or super-computer architectures may similarly be employed. Alternatively, should
22 deployment requirements dictate greater portability, smaller Personal Digital Assistants
23 (PDAs) may be employed.

1 **[00156]** Depending on the particular implementation, features of the SocialPay
2 may be achieved by implementing a microcontroller such as CAST's R8051XC2
3 microcontroller; Intel's MCS 51 (i.e., 8051 microcontroller); and/or the like. Also, to
4 implement certain features of the SocialPay, some feature implementations may rely on
5 embedded components, such as: Application-Specific Integrated Circuit ("ASIC"),
6 Digital Signal Processing ("DSP"), Field Programmable Gate Array ("FPGA"), and/or the
7 like embedded technology. For example, any of the SocialPay component collection
8 (distributed or otherwise) and/or features may be implemented via the microprocessor
9 and/or via embedded components; e.g., via ASIC, coprocessor, DSP, FPGA, and/or the
10 like. Alternately, some implementations of the SocialPay may be implemented with
11 embedded components that are configured and used to achieve a variety of features or
12 signal processing.

13 **[00157]** Depending on the particular implementation, the embedded components
14 may include software solutions, hardware solutions, and/or some combination of both
15 hardware/software solutions. For example, SocialPay features discussed herein may be
16 achieved through implementing FPGAs, which are a semiconductor devices containing
17 programmable logic components called "logic blocks", and programmable
18 interconnects, such as the high performance FPGA Virtex series and/or the low cost
19 Spartan series manufactured by Xilinx. Logic blocks and interconnects can be
20 programmed by the customer or designer, after the FPGA is manufactured, to
21 implement any of the SocialPay features. A hierarchy of programmable interconnects
22 allow logic blocks to be interconnected as needed by the SocialPay system
23 designer/administrator, somewhat like a one-chip programmable breadboard. An
24 FPGA's logic blocks can be programmed to perform the operation of basic logic gates

1 such as AND, and XOR, or more complex combinational operators such as decoders or
2 simple mathematical operations. In most FPGAs, the logic blocks also include memory
3 elements, which may be circuit flip-flops or more complete blocks of memory. In some
4 circumstances, the SocialPay may be developed on regular FPGAs and then migrated
5 into a fixed version that more resembles ASIC implementations. Alternate or
6 coordinating implementations may migrate SocialPay controller features to a final ASIC
7 instead of or in addition to FPGAs. Depending on the implementation all of the
8 aforementioned embedded components and microprocessors may be considered the
9 “CPU” and/or “processor” for the SocialPay.

10 **Power Source**

11 **[00158]** The power source 2486 may be of any standard form for powering small
12 electronic circuit board devices such as the following power cells: alkaline, lithium
13 hydride, lithium ion, lithium polymer, nickel cadmium, solar cells, and/or the like.
14 Other types of AC or DC power sources may be used as well. In the case of solar cells, in
15 one embodiment, the case provides an aperture through which the solar cell may
16 capture photonic energy. The power cell 2486 is connected to at least one of the
17 interconnected subsequent components of the SocialPay thereby providing an electric
18 current to all subsequent components. In one example, the power source 2486 is
19 connected to the system bus component 2404. In an alternative embodiment, an outside
20 power source 2486 is provided through a connection across the I/O 2408 interface. For
21 example, a USB and/or IEEE 1394 connection carries both data and power across the
22 connection and is therefore a suitable source of power.

Interface Adapters

1
2 **[00159]** Interface bus(es) 2407 may accept, connect, and/or communicate to a
3 number of interface adapters, conventionally although not necessarily in the form of
4 adapter cards, such as but not limited to: input output interfaces (I/O) 2408, storage
5 interfaces 2409, network interfaces 2410, and/or the like. Optionally, cryptographic
6 processor interfaces 2427 similarly may be connected to the interface bus. The interface
7 bus provides for the communications of interface adapters with one another as well as
8 with other components of the computer systemization. Interface adapters are adapted
9 for a compatible interface bus. Interface adapters conventionally connect to the
10 interface bus via a slot architecture. Conventional slot architectures may be employed,
11 such as, but not limited to: Accelerated Graphics Port (AGP), Card Bus, (Extended)
12 Industry Standard Architecture ((E)ISA), Micro Channel Architecture (MCA), NuBus,
13 Peripheral Component Interconnect (Extended) (PCI(X)), PCI Express, Personal
14 Computer Memory Card International Association (PCMCIA), and/or the like.

15 **[00160]** Storage interfaces 2409 may accept, communicate, and/or connect to a
16 number of storage devices such as, but not limited to: storage devices 2414, removable
17 disc devices, and/or the like. Storage interfaces may employ connection protocols such
18 as, but not limited to: (Ultra) (Serial) Advanced Technology Attachment (Packet
19 Interface) ((Ultra) (Serial) ATA(PI)), (Enhanced) Integrated Drive Electronics ((E)IDE),
20 Institute of Electrical and Electronics Engineers (IEEE) 1394, fiber channel, Small
21 Computer Systems Interface (SCSI), Universal Serial Bus (USB), and/or the like.

22 **[00161]** Network interfaces 2410 may accept, communicate, and/or connect to a
23 communications network 2413. Through a communications network 2413, the SocialPay

1 controller is accessible through remote clients 2433b (e.g., computers with web
2 browsers) by users 2433a. Network interfaces may employ connection protocols such as,
3 but not limited to: direct connect, Ethernet (thick, thin, twisted pair 10/100/1000 Base
4 T, and/or the like), Token Ring, wireless connection such as IEEE 802.11a-x, and/or the
5 like. Should processing requirements dictate a greater amount speed and/or capacity,
6 distributed network controllers (e.g., Distributed SocialPay), architectures may similarly
7 be employed to pool, load balance, and/or otherwise increase the communicative
8 bandwidth required by the SocialPay controller. A communications network may be any
9 one and/or the combination of the following: a direct interconnection; the Internet; a
10 Local Area Network (LAN); a Metropolitan Area Network (MAN); an Operating
11 Missions as Nodes on the Internet (OMNI); a secured custom connection; a Wide Area
12 Network (WAN); a wireless network (e.g., employing protocols such as, but not limited
13 to a Wireless Application Protocol (WAP), I-mode, and/or the like); and/or the like. A
14 network interface may be regarded as a specialized form of an input output interface.
15 Further, multiple network interfaces 2410 may be used to engage with various
16 communications network types 2413. For example, multiple network interfaces may be
17 employed to allow for the communication over broadcast, multicast, and/or unicast
18 networks.

19 **[00162]** Input Output interfaces (I/O) 2408 may accept, communicate, and/or
20 connect to user input devices 2411, peripheral devices 2412, cryptographic processor
21 devices 2428, and/or the like. I/O may employ connection protocols such as, but not
22 limited to: audio: analog, digital, monaural, RCA, stereo, and/or the like; data: Apple
23 Desktop Bus (ADB), IEEE 1394a-b, serial, universal serial bus (USB); infrared; joystick;
24 keyboard; midi; optical; PC AT; PS/2; parallel; radio; video interface: Apple Desktop

1 Connector (ADC), BNC, coaxial, component, composite, digital, Digital Visual Interface
2 (DVI), high-definition multimedia interface (HDMI), RCA, RF antennae, S-Video, VGA,
3 and/or the like; wireless transceivers: 802.11a/b/g/n/x; Bluetooth; cellular (e.g., code
4 division multiple access (CDMA), high speed packet access (HSPA(+)), high-speed
5 downlink packet access (HSDPA), global system for mobile communications (GSM),
6 long term evolution (LTE), WiMax, etc.); and/or the like. One typical output device may
7 include a video display, which typically comprises a Cathode Ray Tube (CRT) or Liquid
8 Crystal Display (LCD) based monitor with an interface (e.g., DVI circuitry and cable)
9 that accepts signals from a video interface, may be used. The video interface composites
10 information generated by a computer systemization and generates video signals based
11 on the composited information in a video memory frame. Another output device is a
12 television set, which accepts signals from a video interface. Typically, the video interface
13 provides the composited video information through a video connection interface that
14 accepts a video display interface (e.g., an RCA composite video connector accepting an
15 RCA composite video cable; a DVI connector accepting a DVI display cable, etc.).

16 **[00163]** User input devices 2411 often are a type of peripheral device 2412 (see
17 below) and may include: card readers, dongles, finger print readers, gloves, graphics
18 tablets, joysticks, keyboards, microphones, mouse (mice), remote controls, retina
19 readers, touch screens (e.g., capacitive, resistive, etc.), trackballs, trackpads, sensors
20 (e.g., accelerometers, ambient light, GPS, gyroscopes, proximity, etc.), styluses, and/or
21 the like.

22 **[00164]** Peripheral devices 2412 may be connected and/or communicate to I/O
23 and/or other facilities of the like such as network interfaces, storage interfaces, directly

1 to the interface bus, system bus, the CPU, and/or the like. Peripheral devices may be
2 external, internal and/or part of the SocialPay controller. Peripheral devices may
3 include: antenna, audio devices (e.g., line-in, line-out, microphone input, speakers, etc.),
4 cameras (e.g., still, video, webcam, etc.), dongles (e.g., for copy protection, ensuring
5 secure transactions with a digital signature, and/or the like), external processors (for
6 added capabilities; e.g., crypto devices 2428), force-feedback devices (e.g., vibrating
7 motors), network interfaces, printers, scanners, storage devices, transceivers (e.g.,
8 cellular, GPS, etc.), video devices (e.g., goggles, monitors, etc.), video sources, visors,
9 and/or the like. Peripheral devices often include types of input devices (e.g., cameras).

10 **[00165]** It should be noted that although user input devices and peripheral devices
11 may be employed, the SocialPay controller may be embodied as an embedded,
12 dedicated, and/or monitor-less (i.e., headless) device, wherein access would be provided
13 over a network interface connection.

14 **[00166]** Cryptographic units such as, but not limited to, microcontrollers,
15 processors 2426, interfaces 2427, and/or devices 2428 may be attached, and/or
16 communicate with the SocialPay controller. A MC68HC16 microcontroller,
17 manufactured by Motorola Inc., may be used for and/or within cryptographic units. The
18 MC68HC16 microcontroller utilizes a 16-bit multiply-and-accumulate instruction in the
19 16 MHz configuration and requires less than one second to perform a 512-bit RSA
20 private key operation. Cryptographic units support the authentication of
21 communications from interacting agents, as well as allowing for anonymous
22 transactions. Cryptographic units may also be configured as part of the CPU. Equivalent
23 microcontrollers and/or processors may also be used. Other commercially available

1 specialized cryptographic processors include: the Broadcom's CryptoNetX and other
2 Security Processors; nCipher's nShield, SafeNet's Luna PCI (e.g., 7100) series;
3 Semaphore Communications' 40 MHz Roadrunner 184; Sun's Cryptographic
4 Accelerators (e.g., Accelerator 6000 PCIe Board, Accelerator 500 Daughtercard); Via
5 Nano Processor (e.g., L2100, L2200, U2400) line, which is capable of performing 500+
6 MB/s of cryptographic instructions; VLSI Technology's 33 MHz 6868; and/or the like.

7 **Memory**

8 **[00167]** Generally, any mechanization and/or embodiment allowing a processor to
9 affect the storage and/or retrieval of information is regarded as memory 2429. However,
10 memory is a fungible technology and resource, thus, any number of memory
11 embodiments may be employed in lieu of or in concert with one another. It is to be
12 understood that the SocialPay controller and/or a computer systemization may employ
13 various forms of memory 2429. For example, a computer systemization may be
14 configured wherein the operation of on-chip CPU memory (e.g., registers), RAM, ROM,
15 and any other storage devices are provided by a paper punch tape or paper punch card
16 mechanism; however, such an embodiment would result in an extremely slow rate of
17 operation. In a typical configuration, memory 2429 will include ROM 2406, RAM 2405,
18 and a storage device 2414. A storage device 2414 may be any conventional computer
19 system storage. Storage devices may include a drum; a (fixed and/or removable)
20 magnetic disk drive; a magneto-optical drive; an optical drive (i.e., Blu-ray, CD
21 ROM/RAM/Recordable (R)/ReWritable (RW), DVD R/RW, HD DVD R/RW etc.); an
22 array of devices (e.g., Redundant Array of Independent Disks (RAID)); solid state
23 memory devices (USB memory, solid state drives (SSD), etc.); other processor-readable

1 storage mediums; and/or other devices of the like. Thus, a computer systemization
2 generally requires and makes use of memory.

3 **Component Collection**

4 **[00168]** The memory 2429 may contain a collection of program and/or database
5 components and/or data such as, but not limited to: operating system component(s)
6 2415 (operating system); information server component(s) 2416 (information server);
7 user interface component(s) 2417 (user interface); Web browser component(s) 2418
8 (Web browser); database(s) 2419; mail server component(s) 2421; mail client
9 component(s) 2422; cryptographic server component(s) 2420 (cryptographic server);
10 the SocialPay component(s) 2435; and/or the like (i.e., collectively a component
11 collection). These components may be stored and accessed from the storage devices
12 and/or from storage devices accessible through an interface bus. Although non-
13 conventional program components such as those in the component collection, typically,
14 are stored in a local storage device 2414, they may also be loaded and/or stored in
15 memory such as: peripheral devices, RAM, remote storage facilities through a
16 communications network, ROM, various forms of memory, and/or the like.

17 **Operating System**

18 **[00169]** The operating system component 2415 is an executable program
19 component facilitating the operation of the SocialPay controller. Typically, the operating
20 system facilitates access of I/O, network interfaces, peripheral devices, storage devices,
21 and/or the like. The operating system may be a highly fault tolerant, scalable, and
22 secure system such as: Apple Macintosh OS X (Server); AT&T Plan 9; Be OS; Unix and
23 Unix-like system distributions (such as AT&T's UNIX; Berkley Software Distribution

1 (BSD) variations such as FreeBSD, NetBSD, OpenBSD, and/or the like; Linux
2 distributions such as Red Hat, Ubuntu, and/or the like); and/or the like operating
3 systems. However, more limited and/or less secure operating systems also may be
4 employed such as Apple Macintosh OS, IBM OS/2, Microsoft DOS, Microsoft Windows
5 2000/2003/3.1/95/98/CE/Millennium/NT/Vista/XP (Server), Palm OS, and/or the like.
6 An operating system may communicate to and/or with other components in a
7 component collection, including itself, and/or the like. Most frequently, the operating
8 system communicates with other program components, user interfaces, and/or the like.
9 For example, the operating system may contain, communicate, generate, obtain, and/or
10 provide program component, system, user, and/or data communications, requests,
11 and/or responses. The operating system, once executed by the CPU, may enable the
12 interaction with communications networks, data, I/O, peripheral devices, program
13 components, memory, user input devices, and/or the like. The operating system may
14 provide communications protocols that allow the SocialPay controller to communicate
15 with other entities through a communications network 2413. Various communication
16 protocols may be used by the SocialPay controller as a subcarrier transport mechanism
17 for interaction, such as, but not limited to: multicast, TCP/IP, UDP, unicast, and/or the
18 like.

19 Information Server

20 **[00170]** An information server component 2416 is a stored program component
21 that is executed by a CPU. The information server may be a conventional Internet
22 information server such as, but not limited to Apache Software Foundation's Apache,
23 Microsoft's Internet Information Server, and/or the like. The information server may

1 allow for the execution of program components through facilities such as Active Server
2 Page (ASP), ActiveX, (ANSI) (Objective-) C (++), C# and/or .NET, Common Gateway
3 Interface (CGI) scripts, dynamic (D) hypertext markup language (HTML), FLASH, Java,
4 JavaScript, Practical Extraction Report Language (PERL), Hypertext Pre-Processor
5 (PHP), pipes, Python, wireless application protocol (WAP), WebObjects, and/or the like.
6 The information server may support secure communications protocols such as, but not
7 limited to, File Transfer Protocol (FTP); HyperText Transfer Protocol (HTTP); Secure
8 Hypertext Transfer Protocol (HTTPS), Secure Socket Layer (SSL), messaging protocols
9 (e.g., America Online (AOL) Instant Messenger (AIM), Application Exchange (APEX),
10 ICQ, Internet Relay Chat (IRC), Microsoft Network (MSN) Messenger Service, Presence
11 and Instant Messaging Protocol (PRIM), Internet Engineering Task Force's (IETF's)
12 Session Initiation Protocol (SIP), SIP for Instant Messaging and Presence Leveraging
13 Extensions (SIMPLE), open XML-based Extensible Messaging and Presence Protocol
14 (XMPP) (i.e., Jabber or Open Mobile Alliance's (OMA's) Instant Messaging and
15 Presence Service (IMPS)), Yahoo! Instant Messenger Service, and/or the like. The
16 information server provides results in the form of Web pages to Web browsers, and
17 allows for the manipulated generation of the Web pages through interaction with other
18 program components. After a Domain Name System (DNS) resolution portion of an
19 HTTP request is resolved to a particular information server, the information server
20 resolves requests for information at specified locations on the SocialPay controller based
21 on the remainder of the HTTP request. For example, a request such as
22 <http://123.124.125.126/myInformation.html> might have the IP portion of the request
23 "123.124.125.126" resolved by a DNS server to an information server at that IP address;
24 that information server might in turn further parse the http request for the

1 “/myInformation.html” portion of the request and resolve it to a location in memory
2 containing the information “myInformation.html.” Additionally, other information
3 serving protocols may be employed across various ports, e.g., FTP communications
4 across port 21, and/or the like. An information server may communicate to and/or with
5 other components in a component collection, including itself, and/or facilities of the
6 like. Most frequently, the information server communicates with the SocialPay database
7 2419, operating systems, other program components, user interfaces, Web browsers,
8 and/or the like.

9 **[00171]** Access to the SocialPay database may be achieved through a number of
10 database bridge mechanisms such as through scripting languages as enumerated below
11 (e.g., CGI) and through inter-application communication channels as enumerated below
12 (e.g., CORBA, WebObjects, etc.). Any data requests through a Web browser are parsed
13 through the bridge mechanism into appropriate grammars as required by the SocialPay.
14 In one embodiment, the information server would provide a Web form accessible by a
15 Web browser. Entries made into supplied fields in the Web form are tagged as having
16 been entered into the particular fields, and parsed as such. The entered terms are then
17 passed along with the field tags, which act to instruct the parser to generate queries
18 directed to appropriate tables and/or fields. In one embodiment, the parser may
19 generate queries in standard SQL by instantiating a search string with the proper
20 join/select commands based on the tagged text entries, wherein the resulting command
21 is provided over the bridge mechanism to the SocialPay as a query. Upon generating
22 query results from the query, the results are passed over the bridge mechanism, and
23 may be parsed for formatting and generation of a new results Web page by the bridge

1 mechanism. Such a new results Web page is then provided to the information server,
2 which may supply it to the requesting Web browser.

3 **[00172]** Also, an information server may contain, communicate, generate, obtain,
4 and/or provide program component, system, user, and/or data communications,
5 requests, and/or responses.

6 **User Interface**

7 **[00173]** Computer interfaces in some respects are similar to automobile operation
8 interfaces. Automobile operation interface elements such as steering wheels, gearshifts,
9 and speedometers facilitate the access, operation, and display of automobile resources,
10 and status. Computer interaction interface elements such as check boxes, cursors,
11 menus, scrollers, and windows (collectively and commonly referred to as widgets)
12 similarly facilitate the access, capabilities, operation, and display of data and computer
13 hardware and operating system resources, and status. Operation interfaces are
14 commonly called user interfaces. Graphical user interfaces (GUIs) such as the Apple
15 Macintosh Operating System's Aqua, IBM's OS/2, Microsoft's Windows
16 2000/2003/3.1/95/98/CE/Millennium/NT/XP/Vista/7 (i.e., Aero), Unix's X-Windows
17 (e.g., which may include additional Unix graphic interface libraries and layers such as K
18 Desktop Environment (KDE), mythTV and GNU Network Object Model Environment
19 (GNOME)), web interface libraries (e.g., ActiveX, AJAX, (D)HTML, FLASH, Java,
20 JavaScript, etc. interface libraries such as, but not limited to, Dojo, jQuery(UI),
21 MooTools, Prototype, script.aculo.us, SWFObject, Yahoo! User Interface, any of which
22 may be used and) provide a baseline and means of accessing and displaying information
23 graphically to users.

1 **[00174]** A user interface component 2417 is a stored program component that is
2 executed by a CPU. The user interface may be a conventional graphic user interface as
3 provided by, with, and/or atop operating systems and/or operating environments such
4 as already discussed. The user interface may allow for the display, execution,
5 interaction, manipulation, and/or operation of program components and/or system
6 facilities through textual and/or graphical facilities. The user interface provides a facility
7 through which users may affect, interact, and/or operate a computer system. A user
8 interface may communicate to and/or with other components in a component
9 collection, including itself, and/or facilities of the like. Most frequently, the user
10 interface communicates with operating systems, other program components, and/or the
11 like. The user interface may contain, communicate, generate, obtain, and/or provide
12 program component, system, user, and/or data communications, requests, and/or
13 responses.

14 **Web Browser**

15 **[00175]** A Web browser component 2418 is a stored program component that is
16 executed by a CPU. The Web browser may be a conventional hypertext viewing
17 application such as Microsoft Internet Explorer or Netscape Navigator. Secure Web
18 browsing may be supplied with 128bit (or greater) encryption by way of HTTPS, SSL,
19 and/or the like. Web browsers allowing for the execution of program components
20 through facilities such as ActiveX, AJAX, (D)HTML, FLASH, Java, JavaScript, web
21 browser plug-in APIs (e.g., FireFox, Safari Plug-in, and/or the like APIs), and/or the
22 like. Web browsers and like information access tools may be integrated into PDAs,
23 cellular telephones, and/or other mobile devices. A Web browser may communicate to

1 and/or with other components in a component collection, including itself, and/or
2 facilities of the like. Most frequently, the Web browser communicates with information
3 servers, operating systems, integrated program components (e.g., plug-ins), and/or the
4 like; e.g., it may contain, communicate, generate, obtain, and/or provide program
5 component, system, user, and/or data communications, requests, and/or responses.
6 Also, in place of a Web browser and information server, a combined application may be
7 developed to perform similar operations of both. The combined application would
8 similarly affect the obtaining and the provision of information to users, user agents,
9 and/or the like from the SocialPay enabled nodes. The combined application may be
10 nugatory on systems employing standard Web browsers.

11 Mail Server

12 **[00176]** A mail server component 2421 is a stored program component that is
13 executed by a CPU 2403. The mail server may be a conventional Internet mail server
14 such as, but not limited to sendmail, Microsoft Exchange, and/or the like. The mail
15 server may allow for the execution of program components through facilities such as
16 ASP, ActiveX, (ANSI) (Objective-) C (++), C# and/or .NET, CGI scripts, Java,
17 JavaScript, PERL, PHP, pipes, Python, WebObjects, and/or the like. The mail server
18 may support communications protocols such as, but not limited to: Internet message
19 access protocol (IMAP), Messaging Application Programming Interface
20 (MAPI)/Microsoft Exchange, post office protocol (POP3), simple mail transfer protocol
21 (SMTP), and/or the like. The mail server can route, forward, and process incoming and
22 outgoing mail messages that have been sent, relayed and/or otherwise traversing
23 through and/or to the SocialPay.

1 **[00177]** Access to the SocialPay mail may be achieved through a number of APIs
2 offered by the individual Web server components and/or the operating system.

3 **[00178]** Also, a mail server may contain, communicate, generate, obtain, and/or
4 provide program component, system, user, and/or data communications, requests,
5 information, and/or responses.

6 **Mail Client**

7 **[00179]** A mail client component 2422 is a stored program component that is
8 executed by a CPU 2403. The mail client may be a conventional mail viewing application
9 such as Apple Mail, Microsoft Entourage, Microsoft Outlook, Microsoft Outlook
10 Express, Mozilla, Thunderbird, and/or the like. Mail clients may support a number of
11 transfer protocols, such as: IMAP, Microsoft Exchange, POP3, SMTP, and/or the like. A
12 mail client may communicate to and/or with other components in a component
13 collection, including itself, and/or facilities of the like. Most frequently, the mail client
14 communicates with mail servers, operating systems, other mail clients, and/or the like;
15 e.g., it may contain, communicate, generate, obtain, and/or provide program
16 component, system, user, and/or data communications, requests, information, and/or
17 responses. Generally, the mail client provides a facility to compose and transmit
18 electronic mail messages.

19 **Cryptographic Server**

20 **[00180]** A cryptographic server component 2420 is a stored program component
21 that is executed by a CPU 2403, cryptographic processor 2426, cryptographic processor
22 interface 2427, cryptographic processor device 2428, and/or the like. Cryptographic

1 processor interfaces will allow for expedition of encryption and/or decryption requests
2 by the cryptographic component; however, the cryptographic component, alternatively,
3 may run on a conventional CPU. The cryptographic component allows for the
4 encryption and/or decryption of provided data. The cryptographic component allows for
5 both symmetric and asymmetric (e.g., Pretty Good Protection (PGP)) encryption and/or
6 decryption. The cryptographic component may employ cryptographic techniques such
7 as, but not limited to: digital certificates (e.g., X.509 authentication framework), digital
8 signatures, dual signatures, enveloping, password access protection, public key
9 management, and/or the like. The cryptographic component will facilitate numerous
10 (encryption and/or decryption) security protocols such as, but not limited to: checksum,
11 Data Encryption Standard (DES), Elliptical Curve Encryption (ECC), International Data
12 Encryption Algorithm (IDEA), Message Digest 5 (MD5, which is a one way hash
13 operation), passwords, Rivest Cipher (RC5), Rijndael, RSA (which is an Internet
14 encryption and authentication system that uses an algorithm developed in 1977 by Ron
15 Rivest, Adi Shamir, and Leonard Adleman), Secure Hash Algorithm (SHA), Secure
16 Socket Layer (SSL), Secure Hypertext Transfer Protocol (HTTPS), and/or the like.
17 Employing such encryption security protocols, the SocialPay may encrypt all incoming
18 and/or outgoing communications and may serve as node within a virtual private
19 network (VPN) with a wider communications network. The cryptographic component
20 facilitates the process of “security authorization” whereby access to a resource is
21 inhibited by a security protocol wherein the cryptographic component effects authorized
22 access to the secured resource. In addition, the cryptographic component may provide
23 unique identifiers of content, e.g., employing and MD5 hash to obtain a unique
24 signature for an digital audio file. A cryptographic component may communicate to

1 and/or with other components in a component collection, including itself, and/or
2 facilities of the like. The cryptographic component supports encryption schemes
3 allowing for the secure transmission of information across a communications network
4 to enable the SocialPay component to engage in secure transactions if so desired. The
5 cryptographic component facilitates the secure accessing of resources on the SocialPay
6 and facilitates the access of secured resources on remote systems; i.e., it may act as a
7 client and/or server of secured resources. Most frequently, the cryptographic
8 component communicates with information servers, operating systems, other program
9 components, and/or the like. The cryptographic component may contain, communicate,
10 generate, obtain, and/or provide program component, system, user, and/or data
11 communications, requests, and/or responses.

12 **The SocialPay Database**

13 **[00181]** The SocialPay database component 2419 may be embodied in a database
14 and its stored data. The database is a stored program component, which is executed by
15 the CPU; the stored program component portion configuring the CPU to process the
16 stored data. The database may be a conventional, fault tolerant, relational, scalable,
17 secure database such as Oracle or Sybase. Relational databases are an extension of a flat
18 file. Relational databases consist of a series of related tables. The tables are
19 interconnected via a key field. Use of the key field allows the combination of the tables
20 by indexing against the key field; i.e., the key fields act as dimensional pivot points for
21 combining information from various tables. Relationships generally identify links
22 maintained between tables by matching primary keys. Primary keys represent fields that

1 uniquely identify the rows of a table in a relational database. More precisely, they
2 uniquely identify rows of a table on the “one” side of a one-to-many relationship.

3 **[00182]** Alternatively, the SocialPay database may be implemented using various
4 standard data-structures, such as an array, hash, (linked) list, struct, structured text file
5 (e.g., XML), table, and/or the like. Such data-structures may be stored in memory
6 and/or in (structured) files. In another alternative, an object-oriented database may be
7 used, such as Frontier, ObjectStore, Poet, Zope, and/or the like. Object databases can
8 include a number of object collections that are grouped and/or linked together by
9 common attributes; they may be related to other object collections by some common
10 attributes. Object-oriented databases perform similarly to relational databases with the
11 exception that objects are not just pieces of data but may have other types of capabilities
12 encapsulated within a given object. If the SocialPay database is implemented as a data-
13 structure, the use of the SocialPay database 2419 may be integrated into another
14 component such as the SocialPay component 2435. Also, the database may be
15 implemented as a mix of data structures, objects, and relational structures. Databases
16 may be consolidated and/or distributed in countless variations through standard data
17 processing techniques. Portions of databases, e.g., tables, may be exported and/or
18 imported and thus decentralized and/or integrated.

19 **[00183]** In one embodiment, the database component 2419 includes several tables
20 2419a-r. A Users table 2419a may include fields such as, but not limited to: user_id, ssn,
21 dob, first_name, last_name, age, state, address_firstline, address_secondline, zipcode,
22 devices_list, contact_info, contact_type, alt_contact_info, alt_contact_type, and/or the
23 like. The Users table may support and/or track multiple entity accounts on a SocialPay.

1 A Devices table 2419b may include fields such as, but not limited to: device_ID,
2 device_name, device_IP, device_MAC, device_type, device_model, device_version,
3 device_OS, device_apps_list, device_securekey, wallet_app_installed_flag, and/or the
4 like. An Apps table 2419c may include fields such as, but not limited to: app_ID,
5 app_name, app_type, app_dependencies, and/or the like. An Accounts table 2419d
6 may include fields such as, but not limited to: account_number, account_security_code,
7 account_name, issuer_acquirer_flag, issuer_name, acquirer_name, account_address,
8 routing_number, access_API_call, linked_wallets_list, and/or the like. A Merchants
9 table 2419e may include fields such as, but not limited to: merchant_id,
10 merchant_name, merchant_address, ip_address, mac_address, auth_key, port_num,
11 security_settings_list, and/or the like. An Issuers table 2419f may include fields such
12 as, but not limited to: issuer_id, issuer_name, issuer_address, ip_address,
13 mac_address, auth_key, port_num, security_settings_list, and/or the like. An
14 Acquirers table 2419g may include fields such as, but not limited to: account_firstname,
15 account_lastname, account_type, account_num, account_balance_list, billingaddress_
16 line1, billingaddress_line2, billing_zipcode, billing_state, shipping_preferences,
17 shippingaddress_line1, shippingaddress_line2, shipping_zipcode, shipping_state,
18 and/or the like. A Pay Gateways table 2419h may include fields such as, but not limited
19 to: gateway_id, gateway_ip, gateway_mac, gateway_secure_key, gateway_services_list,
20 gateway_applications_list, and/or the like. A Transactions table 2419i may include
21 fields such as, but not limited to: order_id, user_id, timestamp, transaction_cost,
22 purchase_details_list, num_products, products_list, product_type,
23 product_params_list, product_title, product_summary, quantity, user_id, client_id,
24 client_ip, client_type, client_model, operating_system, os_version, app_installed_flag,

1 user_id, account_firstname, account_lastname, account_type, account_num,
2 account_priority_account_ratio, billingaddress_line1, billingaddress_line2,
3 billing_zipcode, billing_state, shipping_preferences, shippingaddress_line1,
4 shippingaddress_line2, shipping_zipcode, shipping_state, merchant_id,
5 merchant_name, merchant_auth_key, and/or the like. A Batches table 2419j may
6 include fields such as, but not limited to: batch_id, transaction_id_list, timestamp_list,
7 cleared_flag_list, clearance_trigger_settings, and/or the like. A Ledgers table 2419k
8 may include fields such as, but not limited to: request_id, timestamp, deposit_amount,
9 batch_id, transaction_id, clear_flag, deposit_account, transaction_summary, payor_
10 name, payor_account, and/or the like. A Products table 2419l may include fields such
11 as, but not limited to: product_ID, product_title, product_attributes_list,
12 product_price, tax_info_list, related_products_list, offers_list, discounts_list,
13 rewards_list, merchants_list, merchant_availability_list, and/or the like. An Offers
14 table 2419m may include fields such as, but not limited to: offer_ID, offer_title,
15 offer_attributes_list, offer_price, offer_expiry, related_products_list, discounts_list,
16 rewards_list, merchants_list, merchant_availability_list, and/or the like. A Behavior
17 Data table 2419n may include fields such as, but not limited to: user_id,
18 activity_timestamp, activity_type, activity_cost, activity_location, activity_duration,
19 activity_genre, and/or the like. An Analytics table 2419o may include fields such as, but
20 not limited to: rule_id, rule_type, rule_inputs, rule_process-Steps_list, rule_outputs,
21 and/or the like. A Social Graph table 2419p may include fields such as, but not limited
22 to: user_id, socialnetwork_id, friend_id_list, friend_type, friend_weight, and/or the
23 like. A Social Pay Rules table 2419q may include fields such as, but not limited to:
24 user_id, rule_id, rule_type, rule_inputs, rule_processes, rule_outputs, rule_priority,

1 and/or the like. A Social Offer Rules table 2419r may include fields such as, but not
2 limited to: offer_id, offer_rule_id, offer_rule_type, offer_rule_inputs,
3 offer_rule_processes, offer_rule_outputs, offer_rule_priority, and/or the like.

4 **[00184]** In one embodiment, the SocialPay database may interact with other
5 database systems. For example, employing a distributed database system, queries and
6 data access by search SocialPay component may treat the combination of the SocialPay
7 database, an integrated data security layer database as a single database entity.

8 **[00185]** In one embodiment, user programs may contain various user interface
9 primitives, which may serve to update the SocialPay. Also, various accounts may require
10 custom database tables depending upon the environments and the types of clients the
11 SocialPay may need to serve. It should be noted that any unique fields may be
12 designated as a key field throughout. In an alternative embodiment, these tables have
13 been decentralized into their own databases and their respective database controllers
14 (i.e., individual database controllers for each of the above tables). Employing standard
15 data processing techniques, one may further distribute the databases over several
16 computer systemizations and/or storage devices. Similarly, configurations of the
17 decentralized database controllers may be varied by consolidating and/or distributing
18 the various database components 2419a-r. The SocialPay may be configured to keep
19 track of various settings, inputs, and parameters via database controllers.

20 **[00186]** The SocialPay database may communicate to and/or with other
21 components in a component collection, including itself, and/or facilities of the like.
22 Most frequently, the SocialPay database communicates with the SocialPay component,

1 other program components, and/or the like. The database may contain, retain, and
2 provide information regarding other nodes and data.

3 The SocialPays

4 **[00187]** The SocialPay component 2435 is a stored program component that is
5 executed by a CPU. In one embodiment, the SocialPay component incorporates any
6 and/or all combinations of the aspects of the SocialPay discussed in the previous
7 FIGURES. As such, the SocialPay affects accessing, obtaining and the provision of
8 information, services, transactions, and/or the like across various communications
9 networks.

10 **[00188]** The SocialPay component may transform message posts to social networks
11 via SocialPay components into payment transaction receipts social merchant-consumer
12 bridging offers, and/or the like and use of the SocialPay. In one embodiment, the
13 SocialPay component 2435 takes inputs (see in the FIGURES, e.g., 211, 218, 214, 411,
14 414, 421, 423, 430, 435, 711, 714, 1111, 1115, 1311, 1314, 1318, 1322, 1325, 1328, 1512,
15 1516, 1524, 1525, and/or the like) etc., and transforms the inputs via various SocialPay
16 components (e.g., UPC 2441, PTA 2442, PTC 2443, SPE 2444, SPT 2445, WSS 2446,
17 SMCB 2447, TDA 2448, TBOG 2449, and/or the like), into outputs (see in the
18 FIGURES, e.g., 221, 412, 416, 428, 438-440, 717, 1113, 1117, 1316, 1319, 1323, 1332, 1334,
19 1335, 1514, 1518, 1520, 1533, and/or the like).

[00189] The SocialPay component enabling access of information between nodes
may be developed by employing standard development tools and languages such as, but
not limited to: Apache components, Assembly, ActiveX, binary executables, (ANSI)
(Objective-) C (++), C# and/or .NET, database adapters, CGI scripts, Java, JavaScript,

1 mapping tools, procedural and object oriented development tools, PERL, PHP, Python,
2 shell scripts, SQL commands, web application server extensions, web development
3 environments and libraries (e.g., Microsoft's ActiveX; Adobe AIR, FLEX & FLASH;
4 AJAX; (D)HTML; Dojo, Java; JavaScript; jQuery(UI); MooTools; Prototype;
5 script.aculo.us; Simple Object Access Protocol (SOAP); SWFObject; Yahoo! User
6 Interface; and/or the like), WebObjects, and/or the like. In one embodiment, the
7 SocialPay server employs a cryptographic server to encrypt and decrypt
8 communications. The SocialPay component may communicate to and/or with other
9 components in a component collection, including itself, and/or facilities of the like.
10 Most frequently, the SocialPay component communicates with the SocialPay database,
11 operating systems, other program components, and/or the like. The SocialPay may
12 contain, communicate, generate, obtain, and/or provide program component, system,
13 user, and/or data communications, requests, and/or responses.

14 **Distributed SocialPays**

15 **[00190]** The structure and/or operation of any of the SocialPay node controller
16 components may be combined, consolidated, and/or distributed in any number of ways
17 to facilitate development and/or deployment. Similarly, the component collection may
18 be combined in any number of ways to facilitate deployment and/or development. To
19 accomplish this, one may integrate the components into a common code base or in a
20 facility that can dynamically load the components on demand in an integrated fashion.

21 **[00191]** The component collection may be consolidated and/or distributed in
22 countless variations through standard data processing and/or development techniques.
23 Multiple instances of any one of the program components in the program component

1 collection may be instantiated on a single node, and/or across numerous nodes to
2 improve performance through load-balancing and/or data-processing techniques.
3 Furthermore, single instances may also be distributed across multiple controllers
4 and/or storage devices; e.g., databases. All program component instances and
5 controllers working in concert may do so through standard data processing
6 communication techniques.

7 **[00192]** The configuration of the SocialPay controller will depend on the context of
8 system deployment. Factors such as, but not limited to, the budget, capacity, location,
9 and/or use of the underlying hardware resources may affect deployment requirements
10 and configuration. Regardless of if the configuration results in more consolidated
11 and/or integrated program components, results in a more distributed series of program
12 components, and/or results in some combination between a consolidated and
13 distributed configuration, data may be communicated, obtained, and/or provided.
14 Instances of components consolidated into a common code base from the program
15 component collection may communicate, obtain, and/or provide data. This may be
16 accomplished through intra-application data processing communication techniques
17 such as, but not limited to: data referencing (e.g., pointers), internal messaging, object
18 instance variable communication, shared memory space, variable passing, and/or the
19 like.

20 **[00193]** If component collection components are discrete, separate, and/or
21 external to one another, then communicating, obtaining, and/or providing data with
22 and/or to other components may be accomplished through inter-application data
23 processing communication techniques such as, but not limited to: Application Program

1 Interfaces (API) information passage; (distributed) Component Object Model
2 ((D)COM), (Distributed) Object Linking and Embedding ((D)OLE), and/or the like),
3 Common Object Request Broker Architecture (CORBA), Jini local and remote
4 application program interfaces, JavaScript Object Notation (JSON), Remote Method
5 Invocation (RMI), SOAP, process pipes, shared files, and/or the like. Messages sent
6 between discrete component components for inter-application communication or within
7 memory spaces of a singular component for intra-application communication may be
8 facilitated through the creation and parsing of a grammar. A grammar may be
9 developed by using development tools such as lex, yacc, XML, and/or the like, which
10 allow for grammar generation and parsing capabilities, which in turn may form the basis
11 of communication messages within and between components.

12 **[00194]** For example, a grammar may be arranged to recognize the tokens of an
13 HTTP post command, e.g.:

```
14         w3c -post http://... Value1  
15  
16
```

17 **[00195]** where Value1 is discerned as being a parameter because “http://” is part of
18 the grammar syntax, and what follows is considered part of the post value. Similarly,
19 with such a grammar, a variable “Value1” may be inserted into an “http://” post
20 command and then sent. The grammar syntax itself may be presented as structured data
21 that is interpreted and/or otherwise used to generate the parsing mechanism (e.g., a
22 syntax description text file as processed by lex, yacc, etc.). Also, once the parsing
23 mechanism is generated and/or instantiated, it itself may process and/or parse
24 structured data such as, but not limited to: character (e.g., tab) delineated text, HTML,
25 structured text streams, XML, and/or the like structured data. In another embodiment,

1 inter-application data processing protocols themselves may have integrated and/or
2 readily available parsers (e.g., JSON, SOAP, and/or like parsers) that may be employed
3 to parse (e.g., communications) data. Further, the parsing grammar may be used
4 beyond message parsing, but may also be used to parse: databases, data collections, data
5 stores, structured data, and/or the like. Again, the desired configuration will depend
6 upon the context, environment, and requirements of system deployment.

7 **[00196]** For example, in some implementations, the SocialPay controller may be
8 executing a PHP script implementing a Secure Sockets Layer (“SSL”) socket server via
9 the information server, which listens to incoming communications on a server port to
10 which a client may send data, e.g., data encoded in JSON format. Upon identifying an
11 incoming communication, the PHP script may read the incoming message from the
12 client device, parse the received JSON-encoded text data to extract information from the
13 JSON-encoded text data into PHP script variables, and store the data (e.g., client
14 identifying information, etc.) and/or extracted information in a relational database
15 accessible using the Structured Query Language (“SQL”). An exemplary listing, written
16 substantially in the form of PHP/SQL commands, to accept JSON-encoded input data
17 from a client device via a SSL connection, parse the data to extract variables, and store
18 the data to a database, is provided below:

```
19 <?PHP
20 header('Content-Type: text/plain');
21
22 // set ip address and port to listen to for incoming data
23 $address = '192.168.0.100';
24 $port = 255;
25
26 // create a server-side SSL socket, listen for/accept incoming communication
27 $sock = socket_create(AF_INET, SOCK_STREAM, 0);
28 socket_bind($sock, $address, $port) or die('Could not bind to address');
29 socket_listen($sock);
30 $client = socket_accept($sock);
31
32 // read input data from client device in 1024 byte blocks until end of message
```

```

1  do {
2      $input = "";
3      $input = socket_read($client, 1024);
4      $data .= $input;
5  } while($input != "");
6
7  // parse data to extract variables
8  $obj = json_decode($data, true);
9
10 // store input data in a database
11 mysql_connect("201.408.185.132",$DBserver,$password); // access database server
12 mysql_select("CLIENT_DB.SQL"); // select database to append
13 mysql_query("INSERT INTO UserTable (transmission)
14 VALUES ($data)"); // add data to UserTable table in a CLIENT database
15 mysql_close("CLIENT_DB.SQL"); // close connection to database
16 ?>
17
18

```

19 **[00197]** Also, the following resources may be used to provide example
 20 embodiments regarding SOAP parser implementation:

```

21  http://www.xav.com/perl/site/lib/SOAP/Parser.html
22  http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/index.jsp?topic=/com.ibm
23    .IBMDI.doc/referenceguide295.htm
24
25

```

26 **[00198]** and other parser implementations:

```

27  http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/index.jsp?topic=/com.ibm
28    .IBMDI.doc/referenceguide259.htm
29
30

```

31 **[00199]** all of which are hereby expressly incorporated by reference herein.

32 **[00200]** In order to address various issues and advance the art, the entirety of this
 33 application for SOCIAL MEDIA PAYMENT PLATFORM APPARATUSES, METHODS
 34 AND SYSTEMS (including the Cover Page, Title, Headings, Field, Background,
 35 Summary, Brief Description of the Drawings, Detailed Description, Claims, Abstract,
 36 FIGURES, Appendices and/or otherwise) shows by way of illustration various
 37 embodiments in which the claimed innovations may be practiced. The advantages and
 38 features of the application are of a representative sample of embodiments only, and are
 39 not exhaustive and/or exclusive. They are presented only to assist in understanding and
 40 teach the claimed principles. It should be understood that they are not representative of

1 all claimed innovations. As such, certain aspects of the disclosure have not been
2 discussed herein. That alternate embodiments may not have been presented for a
3 specific portion of the innovations or that further undescribed alternate embodiments
4 may be available for a portion is not to be considered a disclaimer of those alternate
5 embodiments. It will be appreciated that many of those undescribed embodiments
6 incorporate the same principles of the innovations and others are equivalent. Thus, it is
7 to be understood that other embodiments may be utilized and functional, logical,
8 operational, organizational, structural and/or topological modifications may be made
9 without departing from the scope and/or spirit of the disclosure. As such, all examples
10 and/or embodiments are deemed to be non-limiting throughout this disclosure. Also, no
11 inference should be drawn regarding those embodiments discussed herein relative to
12 those not discussed herein other than it is as such for purposes of reducing space and
13 repetition. For instance, it is to be understood that the logical and/or topological
14 structure of any combination of any program components (a component collection),
15 other components and/or any present feature sets as described in the FIGUREs and/or
16 throughout are not limited to a fixed operating order and/or arrangement, but rather,
17 any disclosed order is exemplary and all equivalents, regardless of order, are
18 contemplated by the disclosure. Furthermore, it is to be understood that such features
19 are not limited to serial execution, but rather, any number of threads, processes,
20 services, servers, and/or the like that may execute asynchronously, concurrently, in
21 parallel, simultaneously, synchronously, and/or the like are contemplated by the
22 disclosure. As such, some of these features may be mutually contradictory, in that they
23 cannot be simultaneously present in a single embodiment. Similarly, some features are
24 applicable to one aspect of the innovations, and inapplicable to others. In addition, the

1 disclosure includes other innovations not presently claimed. Applicant reserves all
2 rights in those presently unclaimed innovations, including the right to claim such
3 innovations, file additional applications, continuations, continuations in part, divisions,
4 and/or the like thereof. As such, it should be understood that advantages, embodiments,
5 examples, functional, features, logical, operational, organizational, structural,
6 topological, and/or other aspects of the disclosure are not to be considered limitations
7 on the disclosure as defined by the claims or limitations on equivalents to the claims. It
8 is to be understood that, depending on the particular needs and/or characteristics of a
9 SocialPay individual and/or enterprise user, database configuration and/or relational
10 model, data type, data transmission and/or network framework, syntax structure,
11 and/or the like, various embodiments of the SocialPay may be implemented that enable
12 a great deal of flexibility and customization. For example, aspects of the SocialPay may
13 be adapted for communication platforms, resource allocation systems, and/or the like.
14 While various embodiments and discussions of the SocialPay have been directed to e-
15 commerce, however, it is to be understood that the embodiments described herein may
16 be readily configured and/or customized for a wide variety of other applications and/or
17 implementations.

CLAIMS

What is claimed is:

1. A social pay processor-implemented method, comprising:

obtaining a user social pay initiation trigger;

obtaining user social data from a social networking site;

identifying, via a processor, a social pay command within the user social data; and

initiating a funds payment transaction based on the identified social pay command.

2. The method of claim 1, wherein identifying the social pay command within the user social data includes:

parsing the user social data;

extracting a social pay command string within the user social data; and

determining a payor identifier, a payee identifier, and a payment amount using the social pay command string.

3. The method of claim 2, further comprising:

querying a database for an identifier of a funds account using the payee identifier;

determining, based on querying the database, that a payee associated with the payee identifier should be enrolled in social pay services; and

1 providing a request to enroll in social pay services.

2

3 4. The method of claim 1, further comprising:

4 analyzing the social pay command to determine whether the funds
5 payment transaction should be initiated;

6 determining that payment verification is required from a payee of the
7 funds payment transaction; and

8 generating a payment verification request using the social pay command;

9 and

10 providing the payment verification request.

11

12 5. The method of claim 1, further comprising:

13 analyzing the social pay command to determine whether the funds
14 payment transaction should be initiated;

15 determining that the social pay command is a fraudulent transaction
16 attempt; and

17 providing a notification to terminate the funds payment transaction.

18

19 6. The method of claim 1, further comprising:

20 determining that the social pay command is a request for payment; and

21 providing an indication of the request for payment for display within a
22 virtual wallet application.

23

24

1 7. The method of claim 6, further comprising:

2 obtaining permission to initiate the funds payment transaction, in
3 response to the provided indication of the request for payment for display within the
4 virtual wallet application; and

5 wherein the funds payment transaction is initiated in response to
6 obtaining the permission to initiate it.

7
8 8. A social pay processor-implemented system, comprising:

9 a processor; and

10 a memory disposed in communication with the processor and storing processor-
11 executable instructions to:

12 obtain a user social pay initiation trigger;

13 obtain user social data from a social networking site;

14 identify, via the processor, a social pay command within the user social
15 data; and

16 initiate a funds payment transaction based on the identified social pay
17 command.

18
19 9. The system of claim 8, wherein identifying the social pay command within the
20 user social data includes:

21 parsing the user social data;

22 extracting a social pay command string within the user social data; and

23 determining a payor identifier, a payee identifier, and a payment amount
24 using the social pay command string.

1
2 10. The system of claim 9, the memory further storing instructions to:

3 query a database for an identifier of a funds account using the payee
4 identifier;

5 determine, based on querying the database, that a payee associated with
6 the payee identifier should be enrolled in social pay services; and

7 provide a request to enroll in social pay services.
8

9 11. The system of claim 8, the memory further storing instructions to:

10 analyze the social pay command to determine whether the funds payment
11 transaction should be initiated;

12 determine that payment verification is required from a payee of the funds
13 payment transaction; and

14 generate a payment verification request using the social pay command;
15 and

16 provide the payment verification request.
17

18 12. The system of claim 8, the memory further storing instructions to:

19 analyze the social pay command to determine whether the funds payment
20 transaction should be initiated;

21 determine that the social pay command is a fraudulent transaction
22 attempt; and

23 provide a notification to terminate the funds payment transaction.
24

1 13. The system of claim 8, the memory further storing instructions to:

2 determine that the social pay command is a request for payment; and

3 provide an indication of the request for payment for display within a
4 virtual wallet application.

5
6 14. The system of claim 13, the memory further storing instructions to:

7 obtain permission to initiate the funds payment transaction, in response to
8 the provided indication of the request for payment for display within the virtual wallet
9 application; and

10 wherein the funds payment transaction is initiated in response to
11 obtaining the permission to initiate it.

12
13 15. A computer-readable tangible medium storing processor-executable social
14 pay instructions to:

15 obtain a user social pay initiation trigger;

16 obtain user social data from a social networking site;

17 identify, via the processor, a social pay command within the user social
18 data; and

19 initiate a funds payment transaction based on the identified social pay
20 command.

21
22 16. The medium of claim 15, wherein identifying the social pay command within
23 the user social data includes:

24 parsing the user social data;

1 extracting a social pay command string within the user social data; and
2 determining a payor identifier, a payee identifier, and a payment amount
3 using the social pay command string.

4

5 17. The medium of claim 16, further storing instructions to:
6 query a database for an identifier of a funds account using the payee
7 identifier;
8 determine, based on querying the database, that a payee associated with
9 the payee identifier should be enrolled in social pay services; and
10 provide a request to enroll in social pay services.

11

12 18. The medium of claim 15, further storing instructions to:
13 analyze the social pay command to determine whether the funds payment
14 transaction should be initiated;
15 determine that payment verification is required from a payee of the funds
16 payment transaction; and
17 generate a payment verification request using the social pay command;
18 and
19 provide the payment verification request.

20

21 19. The medium of claim 15, further storing instructions to:
22 analyze the social pay command to determine whether the funds payment
23 transaction should be initiated;

1 determine that the social pay command is a fraudulent transaction
2 attempt; and

3 provide a notification to terminate the funds payment transaction.
4

5 20. The medium of claim 15, further storing instructions to:

6 determine that the social pay command is a request for payment; and

7 provide an indication of the request for payment for display within a
8 virtual wallet application.
9

10 21. The medium of claim 20, further storing instructions to:

11 obtain permission to initiate the funds payment transaction, in response to
12 the provided indication of the request for payment for display within the virtual wallet
13 application; and

14 wherein the funds payment transaction is initiated in response to
15 obtaining the permission to initiate it.
16

17 22. A social pay processor-implemented method, comprising:

18 obtaining a users social graph system login credentials;

19 logging into the users social graph system;

20 obtaining social user messages sent by the user and to the user;

21 determining if any of the social user messages include social payment
22 message;

23 determining if the social payment message is from the user sending
24 payment to a second entity;

1 determining a source payment account from the user for payment;
2 determining a target payment account of the second entity;
3 transferring a payment specified in the social payment message from the
4 user's source payment account to the second entity's target payment account.
5 obtaining user social data from a social networking site;
6 identifying, via a processor, a social pay command within the user social
7 data; and
8 initiating a funds payment transaction based on the identified social pay
9 command.

10 23. The method of claim 22, wherein the second entity is another user.

11 24. The method of claim 22, wherein the second entity is a payment system.

12 25. The method of claim 22, wherein the second entity is a system.

13 26. The method of claim 22, wherein the second entity is an organization.

14 27. The method of claim 22, wherein the obtaining of social messages is
15 continually intercepted in real time while the user engages in messaging on another
16 messaging client.

17 28. The method of claim 22, wherein the obtaining of social messages occurs as if
18 a messaging client is logged into the user's social graph account by an additional client.

19 29. The method of claim 28, wherein the additional client is logged in in such a
20 manner that the user may simultaneously be logged into their social graph account
21 without either client interfering with another.

22 30. The method of claim 22, further, comprising:

23 appending an assistive payment message to obtained social user messages that
24 have been determined to have social payment messages.

1 31. The method of claim 28, wherein the assistive payment message is appended
2 when a user or entity receiving payment directed from the social payment message does
3 not have a payment account.

4 32. The method of claim 31, wherein the assistive payment message is appended
5 by sending a second message from the additional client that appears to have originated
6 from the user's social messaging client.

7 33. The method of claim 32, wherein the assistive payment message is includes a
8 link to sign up for a payment account which may receive the payment from the social
9 payment message.

10 34. The method of claim 28, wherein the funds payment transaction affects
11 purchase of an item purchased from a merchant and wherein the item purchased is
12 specified in the social pay command and includes an identifier of the item from the
13 merchant and account information to affect purchase of the item and shipping
14 information to send the item to a destination.

15 35. A social pay system, comprising:

16 means to obtain a users social graph system login credentials;

17 means to log into the users social graph system;

18 means to obtain social user messages sent by the user and to the user;

19 means to determine if any of the social user messages include social
20 payment message;

21 means to determine if the social payment message is from the user
22 sending payment to a second entity;

23 means to determine a source payment account from the user for payment;

24 means to determine a target payment account of the second entity;

1 means to transfer a payment specified in the social payment message from
2 the user's source payment account to the second entity's target payment account.

3 means to obtain user social data from a social networking site;

4 means to identify a social pay command within the user social data; and

5 means to initiate a funds payment transaction based on the identified
6 social pay command.

7 36. A processor-readable medium storing processor-issuable instructions to:

8 obtain a users social graph system login credentials;

9 log into the users social graph system;

10 obtain social user messages sent by the user and to the user;

11 determine if any of the social user messages include social payment
12 message;

13 determine if the social payment message is from the user sending payment
14 to a second entity;

15 determine a source payment account from the user for payment;

16 determine a target payment account of the second entity;

17 transfer a payment specified in the social payment message from the user's
18 source payment account to the second entity's target payment account.

19 obtain user social data from a social networking site;

20 identify a social pay command within the user social data; and

21 initiate a funds payment transaction based on the identified social pay
22 command.

23 37. A social pay apparatus, comprising:

24 a memory;

1 a processor disposed in communication with said memory, and configured
2 to issue a plurality of processing instructions stored in the memory, wherein the
3 processor issues instructions to:

4 obtain a users social graph system login credentials;
5 log into the users social graph system;
6 obtain social user messages sent by the user and to the user;
7 determine if any of the social user messages include social payment
8 message;

9 determine if the social payment message is from the user sending
10 payment to a second entity;

11 determine a source payment account from the user for payment;

12 determine a target payment account of the second entity;

13 transfer a payment specified in the social payment message from
14 the user's source payment account to the second entity's target payment account.

15 obtain user social data from a social networking site;

16 identify a social pay command within the user social data; and

17 initiate a funds payment transaction based on the identified social
18 pay command.

19 38. A social pay processor-implemented method, comprising:

20 obtaining a users social graph system login credentials;

21 logging into the users social graph system;

22 obtaining social user messages sent by the user and to the user;

23 determining if any of the social user messages include social payment
24 message;

- 1 determining if the social payment message is from a second entity sending
2 payment to the user;
- 3 determining a source payment account from the second entity for
4 payment;
- 5 determining a target payment account of the user;
- 6 transferring a payment specified in the social payment message from the
7 second entity's source payment account to the user's target payment account.
- 8 obtaining user social data from a social networking site;
- 9 identifying, via a processor, a social pay command within the user social
10 data; and
- 11 initiating a funds payment transaction based on the identified social pay
12 command.
- 13 39. The method of claim 38, wherein the second entity is another user.
- 14 40. The method of claim 38, wherein the second entity is a payment system.
- 15 41. The method of claim 38, wherein the second entity is a system.
- 16 42. The method of claim 38, wherein the second entity is an organization.
- 17 43. The method of claim 38, wherein the obtaining of social messages is
18 continually intercepted in real time while the user engages in messaging on another
19 messaging client.
- 20 44. The method of claim 38, wherein the obtaining of social messages occurs as if
21 a messaging client is logged into the user's social graph account by an additional client.
- 22 45. The method of claim 44, wherein the additional client is logged in in such a
23 manner that the user may simultaneously be logged into their social graph account
24 without either client interfering with another.

1 46. The method of claim 38, further, comprising:

2 appending an assistive payment message to obtained social user messages that
3 have been determined to have social payment messages.

4 47. The method of claim 44, wherein the assistive payment message is appended
5 when a user or entity receiving payment directed from the social payment message does
6 not have a payment account.

7 48. The method of claim 47, wherein the assistive payment message is appended
8 by sending a second message from the additional client that appears to have originated
9 from the user's social messaging client.

10 49. The method of claim 48, wherein the assistive payment message includes a
11 link to sign up for a payment account which may receive the payment from the social
12 payment message.

13 50 The method of claim 38, wherein the funds payment transaction affects
14 purchase of an item purchased from a merchant and wherein the item purchased is
15 specified in the social pay command and includes an identifier of the item from the
16 merchant and account information to affect purchase of the item and shipping
17 information to send the item to a destination.

18 51. A social pay system, comprising:

19 means to obtain a users social graph system login credentials;

20 means to log into the users social graph system;

21 means to obtain social user messages sent by the user and to the user;

22 means to determine if any of the social user messages include social
23 payment message;

- 1 means to determine if the social payment message is from a second entity
2 sending payment to the user;
- 3 means to determine a source payment account from the second entity for
4 payment;
- 5 means to determine a target payment account of the user;
- 6 means to transfer a payment specified in the social payment message from
7 the second entity's source payment account to the user's target payment account.
- 8 means to obtain user social data from a social networking site;
- 9 means to identify a social pay command within the user social data; and
- 10 means to initiate a funds payment transaction based on the identified
11 social pay command.
- 12 52. A processor-readable medium storing processor-issuable instructions to:
- 13 obtain a users social graph system login credentials;
- 14 log into the users social graph system;
- 15 obtain social user messages sent by the user and to the user;
- 16 determine if any of the social user messages include social payment
17 message;
- 18 determine if the social payment message is from a second entity sending
19 payment to the user;
- 20 determine a source payment account from the second entity for payment;
- 21 determine a target payment account of the user;
- 22 transfer a payment specified in the social payment message from the
23 second entity's source payment account to the user's target payment account.
- 24 obtain user social data from a social networking site;

1 identify a social pay command within the user social data; and
2 initiate a funds payment transaction based on the identified social pay
3 command.

4 53. A social pay apparatus, comprising:
5 a memory;
6 a processor disposed in communication with said memory, and configured
7 to issue a plurality of processing instructions stored in the memory, wherein the
8 processor issues instructions to:
9 obtain a users social graph system login credentials;
10 log into the users social graph system;
11 obtain social user messages sent by the user and to the user;
12 determine if any of the social user messages include social payment
13 message;
14 determine if the social payment message is from a second entity
15 sending payment to the user;
16 determine a source payment account from the second entity for
17 payment;
18 determine a target payment account of the user;
19 transfer a payment specified in the social payment message from
20 the second entity's source payment account to the user's target payment account.
21 obtain user social data from a social networking site;
22 identify a social pay command within the user social data; and
23 initiate a funds payment transaction based on the identified social
24 pay command.

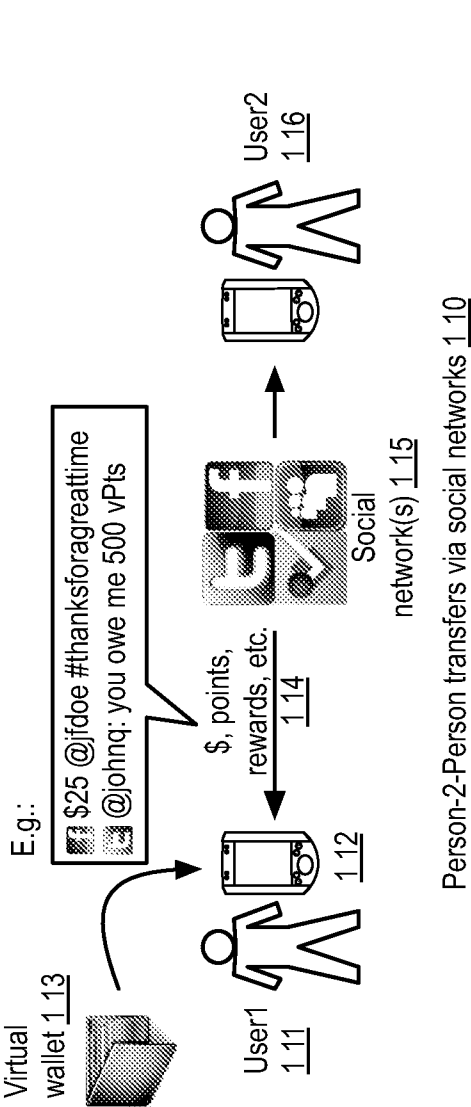


FIGURE 1A

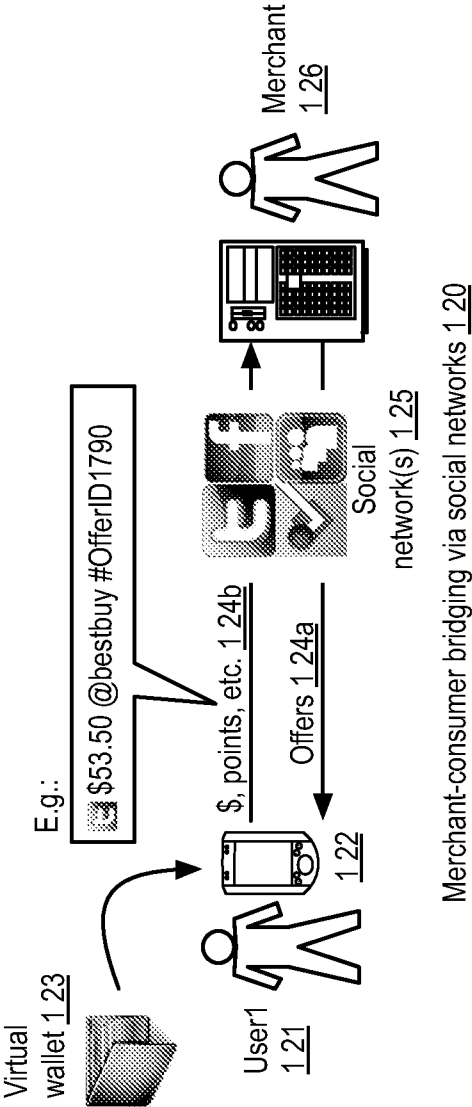


FIGURE 1B

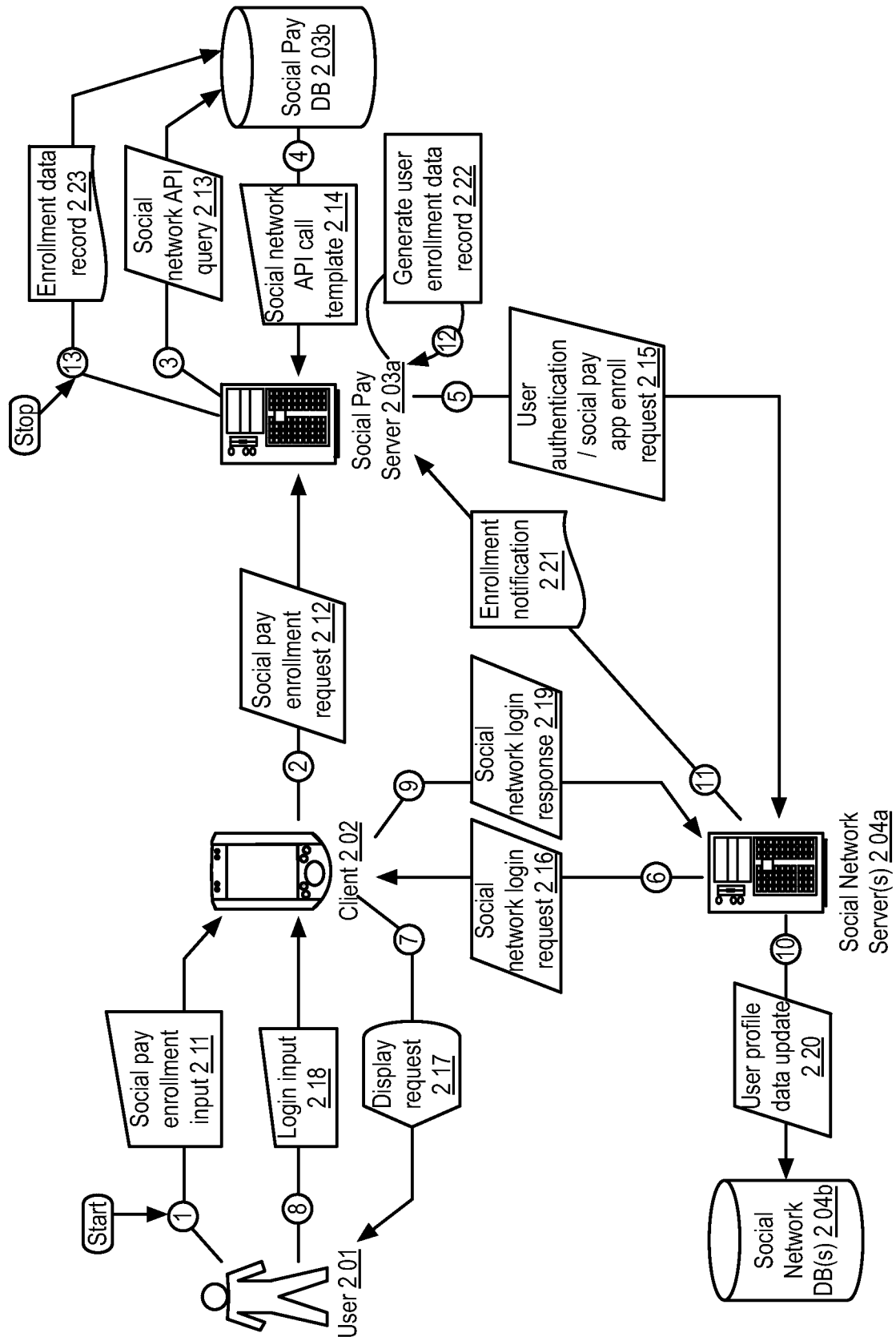


FIGURE 2

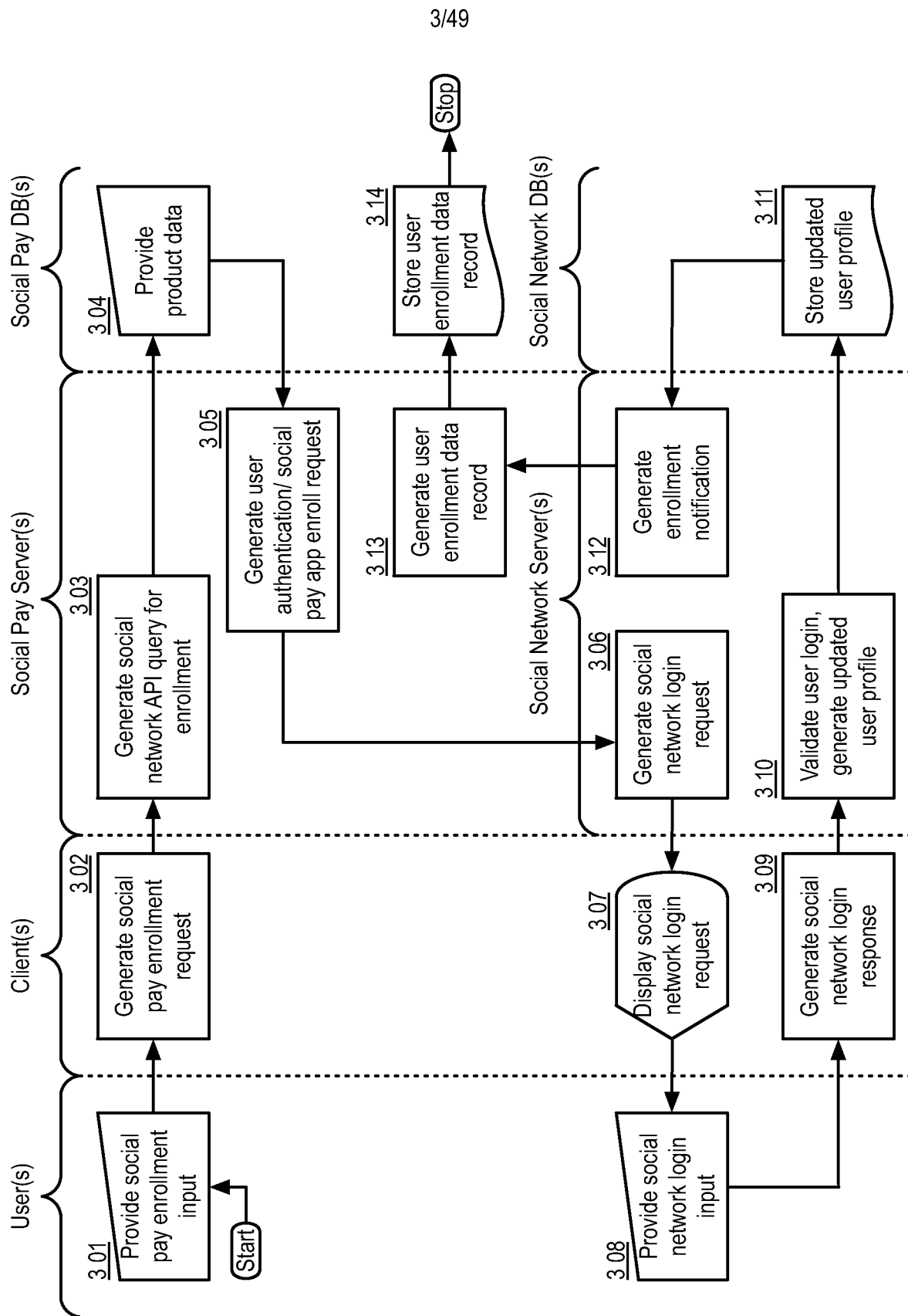


FIGURE 3 Example Logic Flow: Social Pay Enrollment ("SPE") component 300

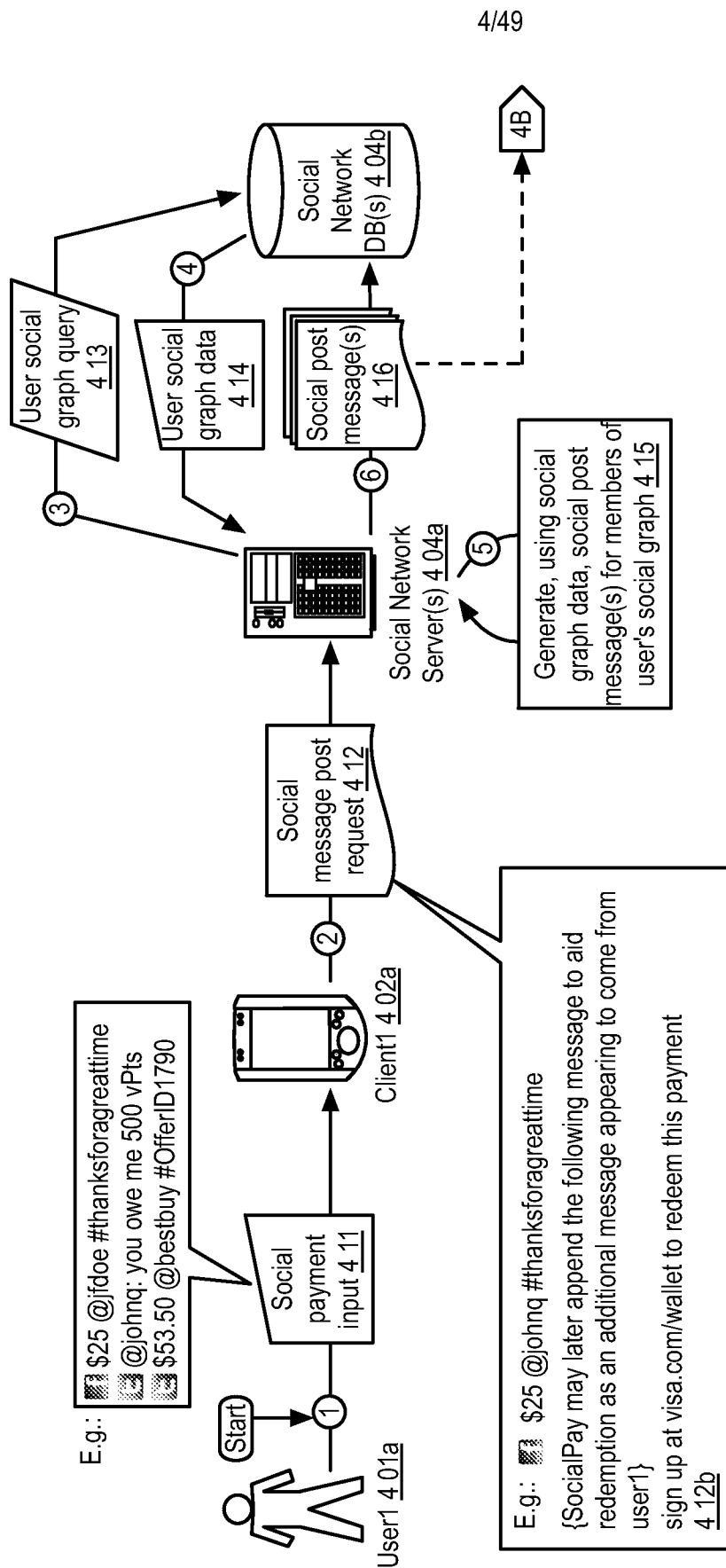
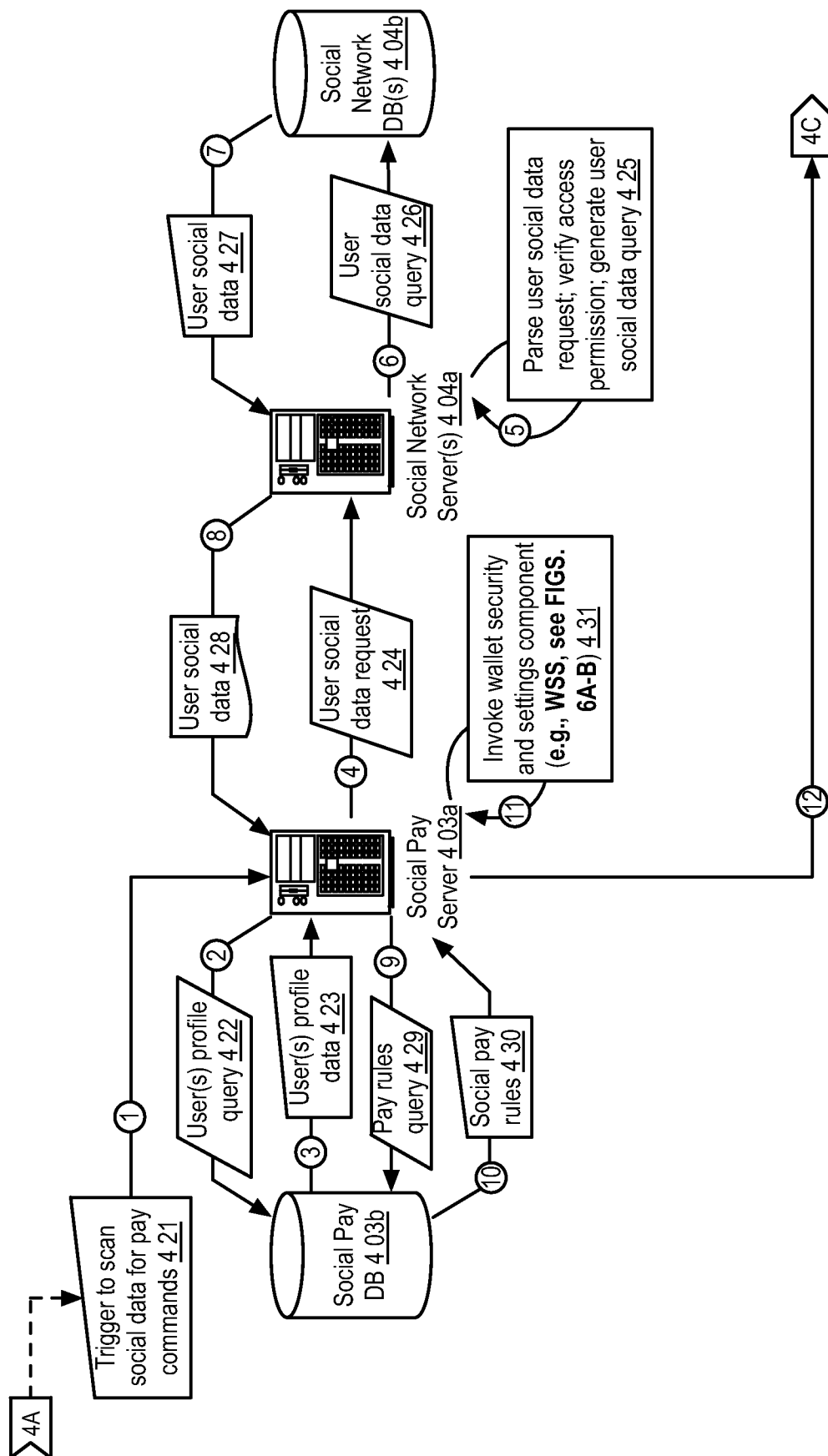


FIGURE 4A

Example Data Flow: Social Payment Triggering



Example Data Flow: Social Payment Triggering

FIGURE 4B

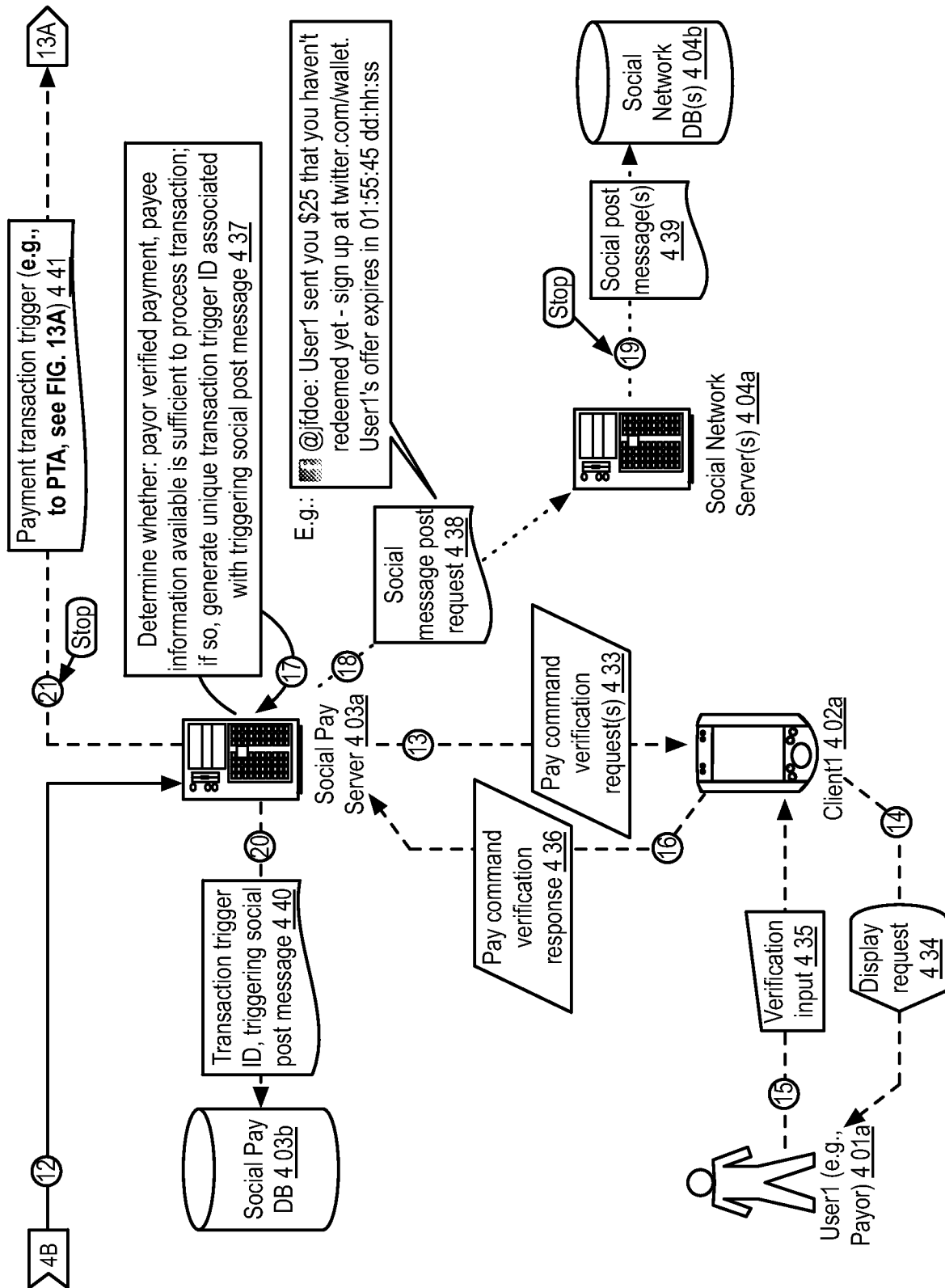


FIGURE 4C

Example Data Flow: Social Payment Triggering

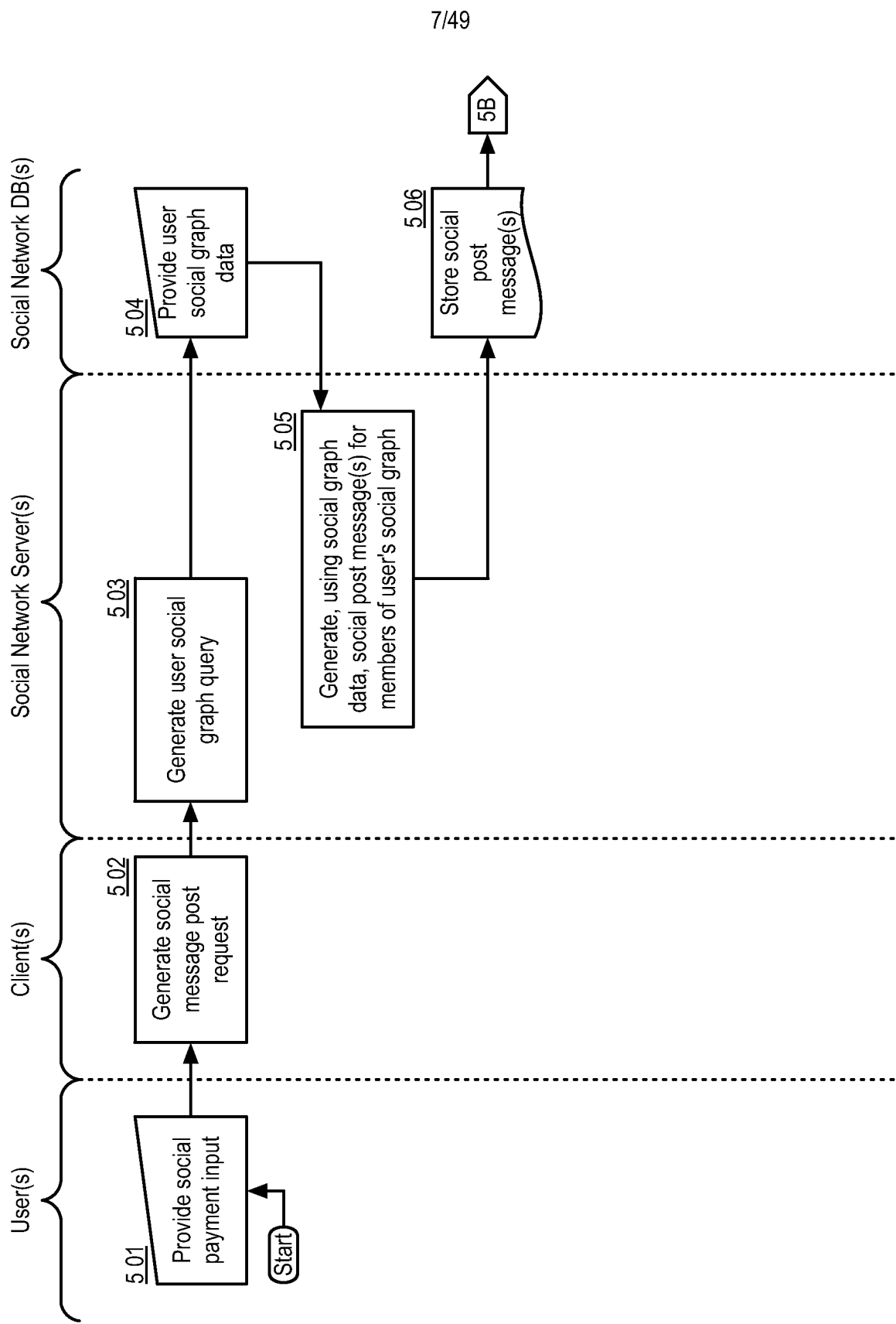
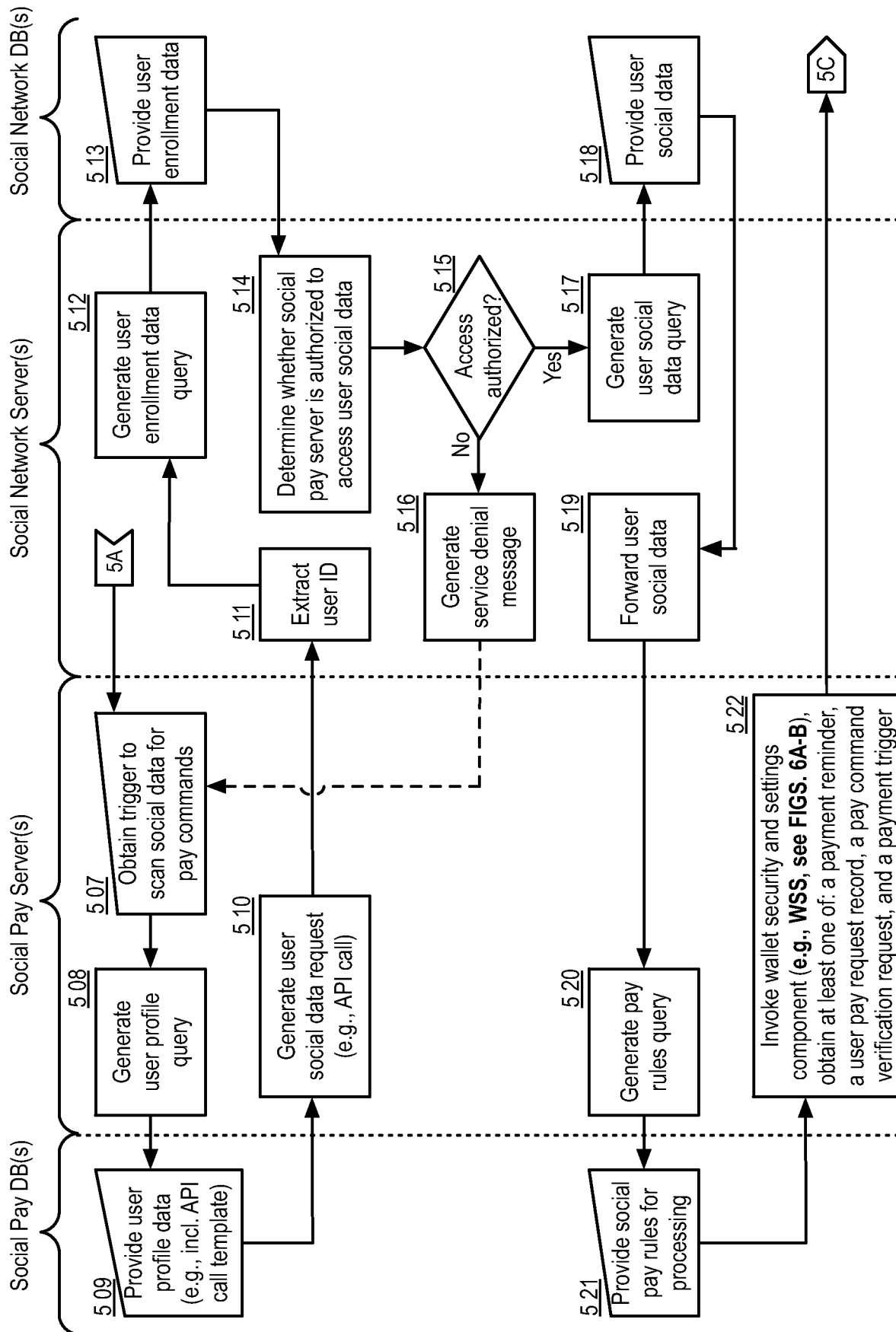


FIGURE 5A

Example Logic Flow: Social Payment Triggering ("SPT") component 500

8/49

FIGURE 5B
Example Logic Flow: Social Payment Triggering ("SPT") component 500

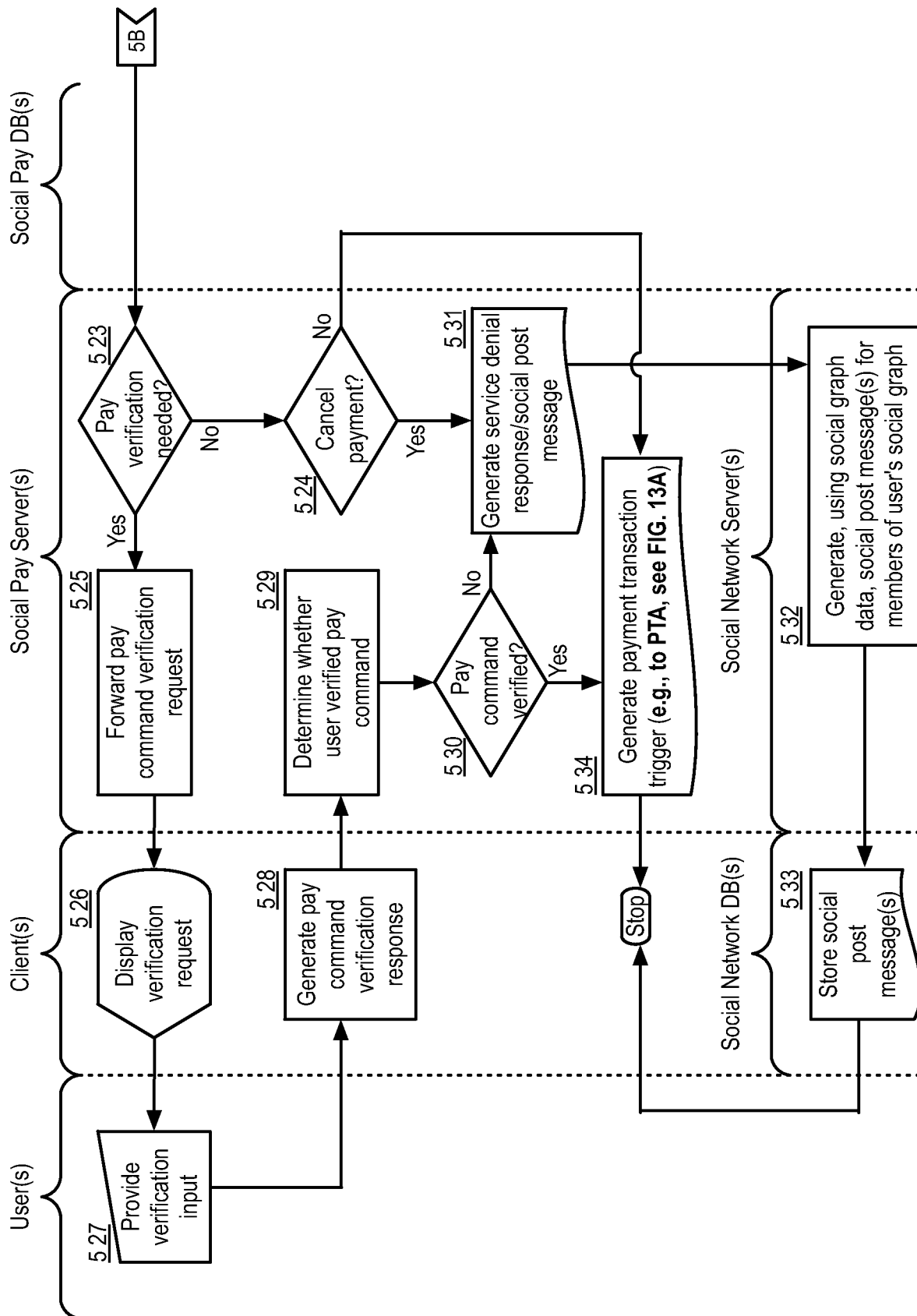


FIGURE 5C

Example Logic Flow: Social Payment Triggering ("SPT") component 500

10/49

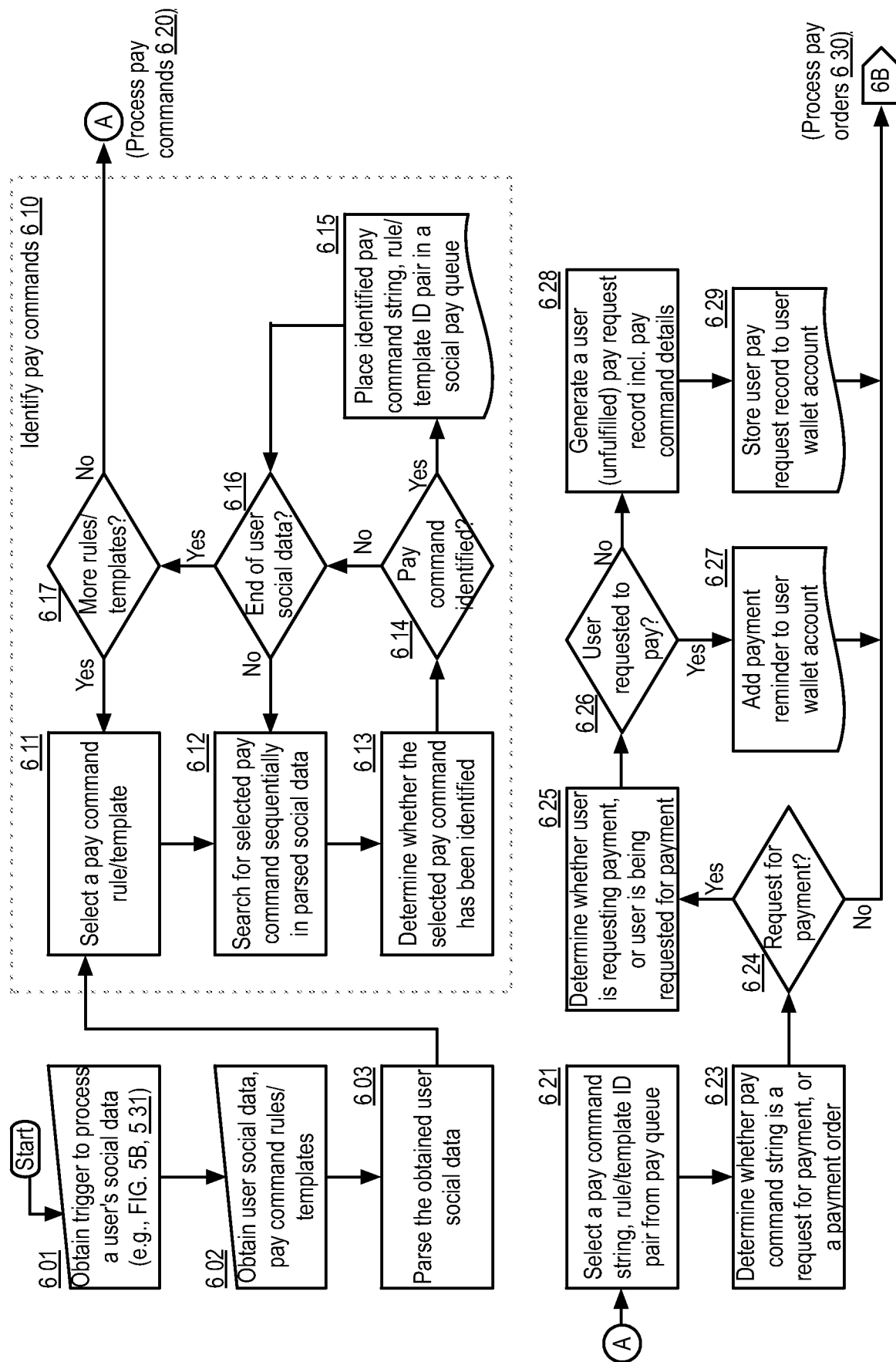


FIGURE 6A

Example Logic Flow: Wallet Security and Settings ("WSS") component 600

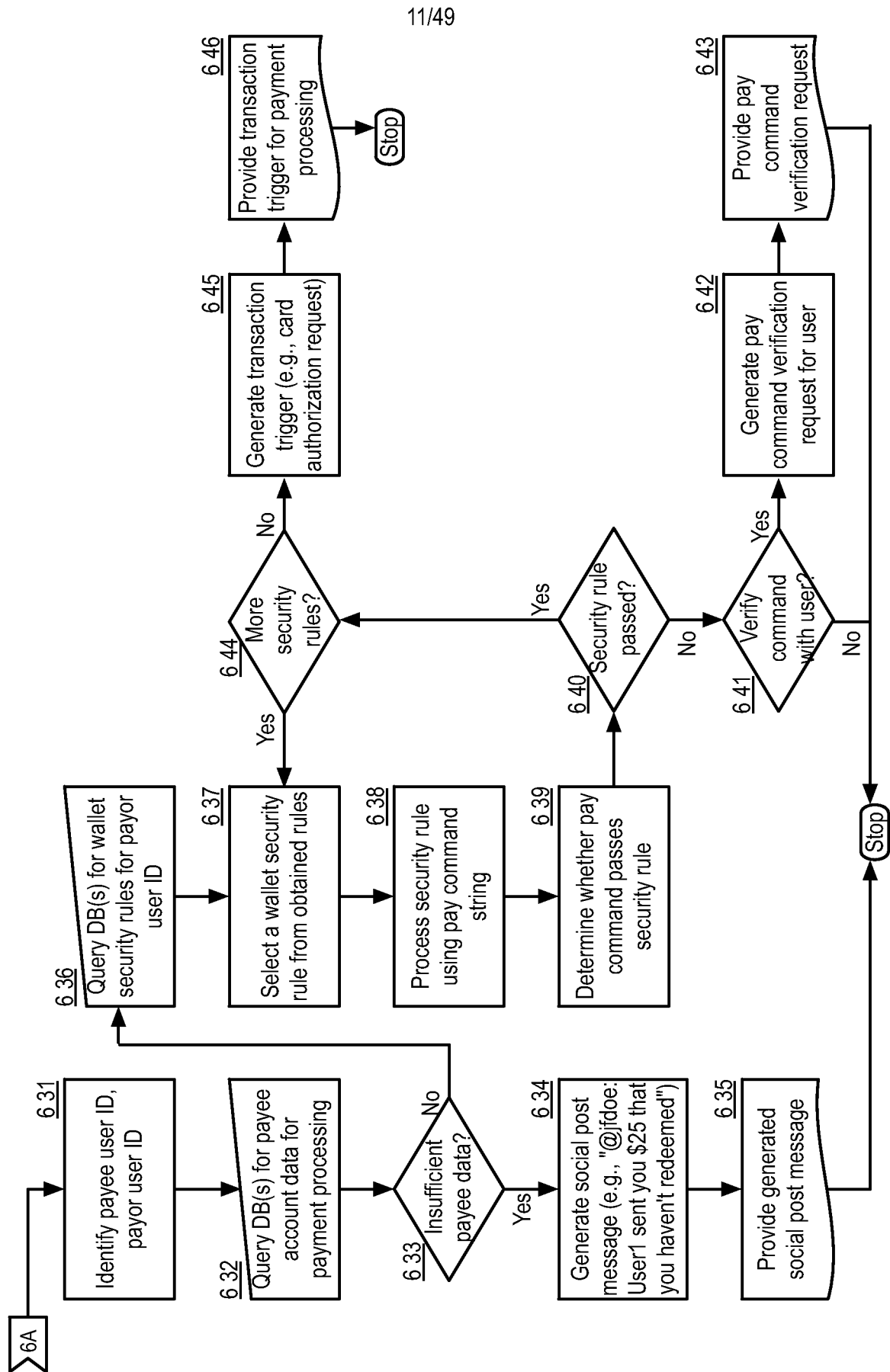


FIGURE 6B

Example Logic Flow: Wallet Security and Settings ("WSS") component 600

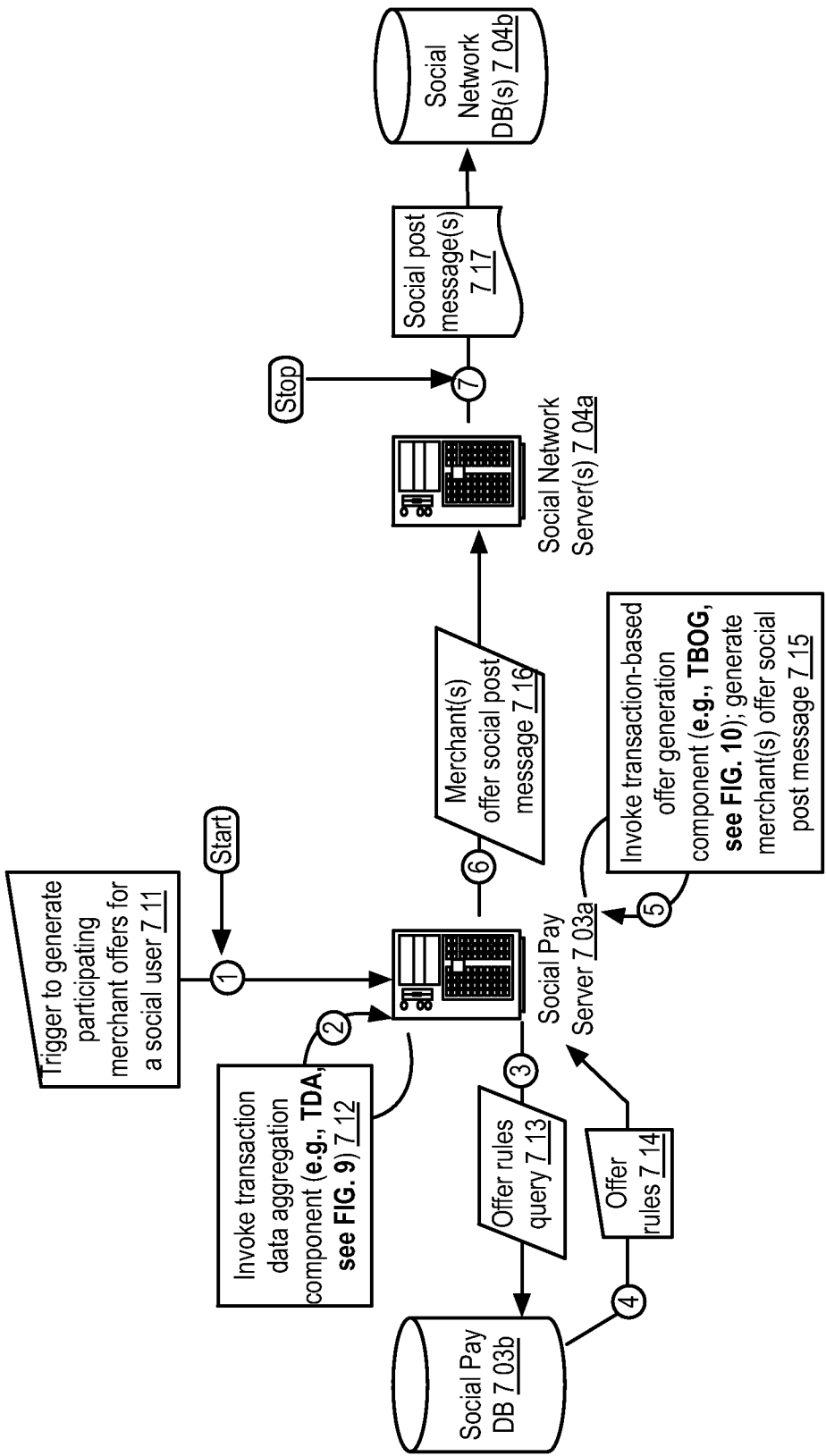


FIGURE 7

Example Data Flow: Social Merchant Consumer Bridging

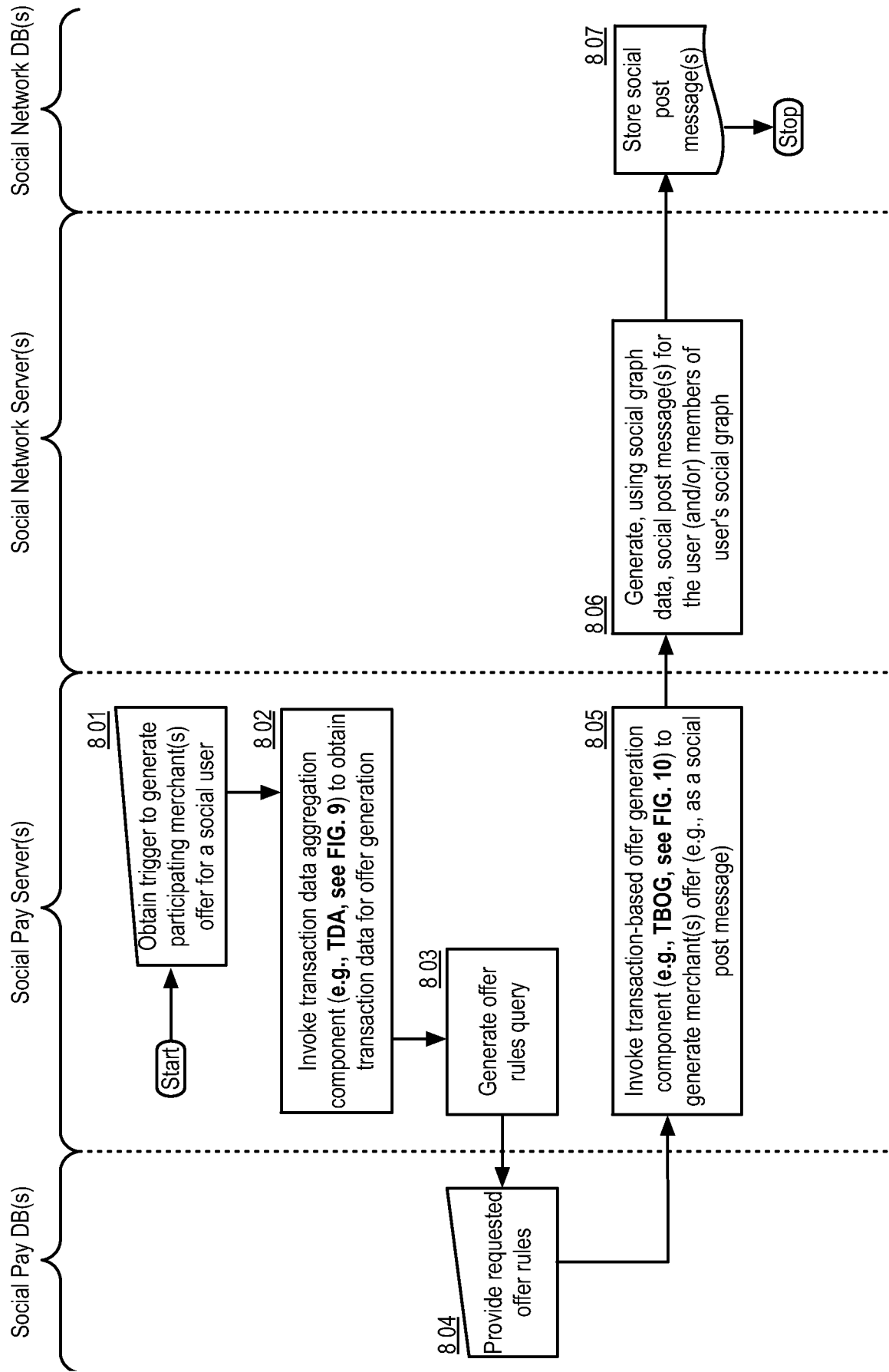


FIGURE 8 Example Logic Flow: Social Merchant Consumer Bridging ("SMCB") component 800

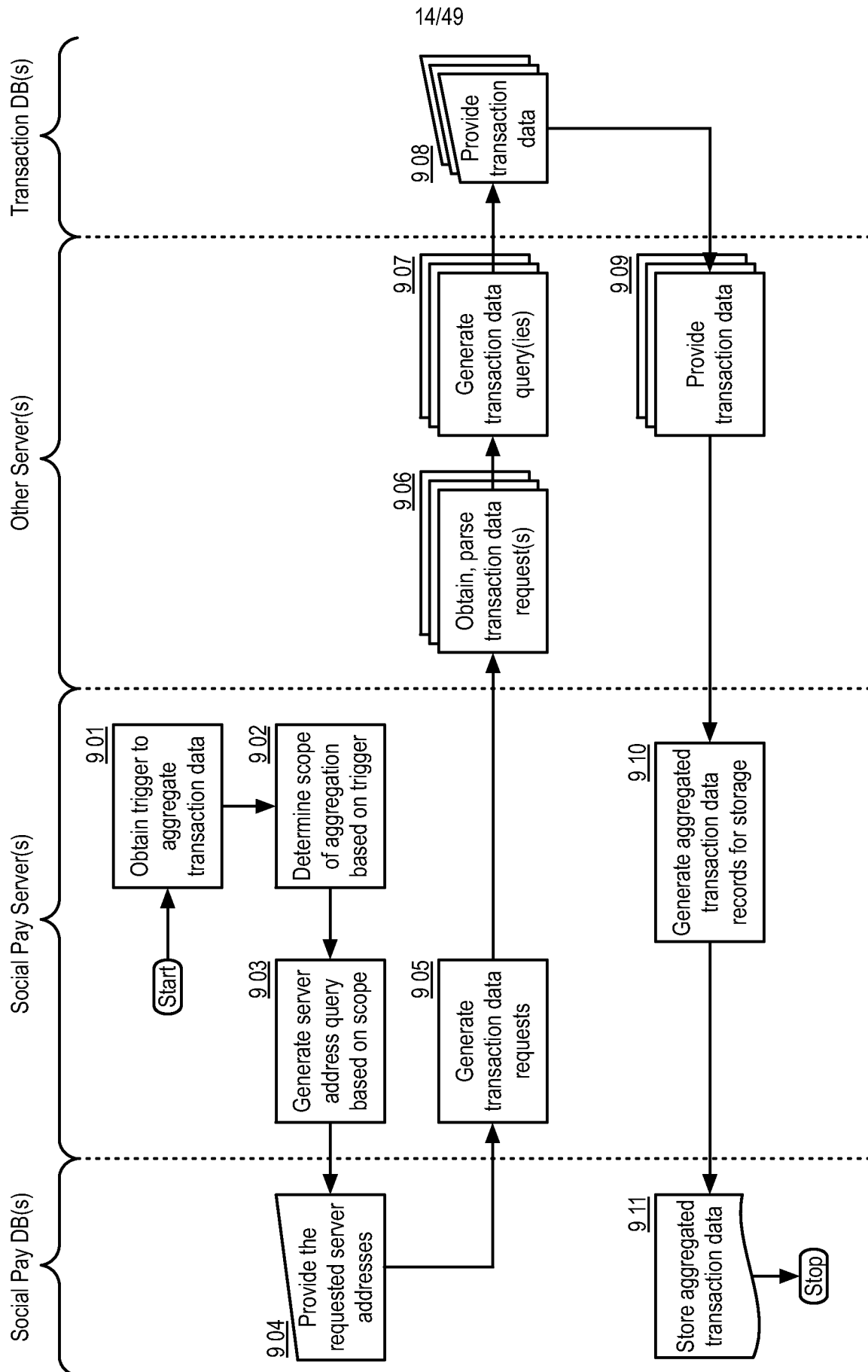


FIGURE 9 Example Logic Flow: Transaction Data Aggregation ("TDA") component 900

FIGURE 9

15/49

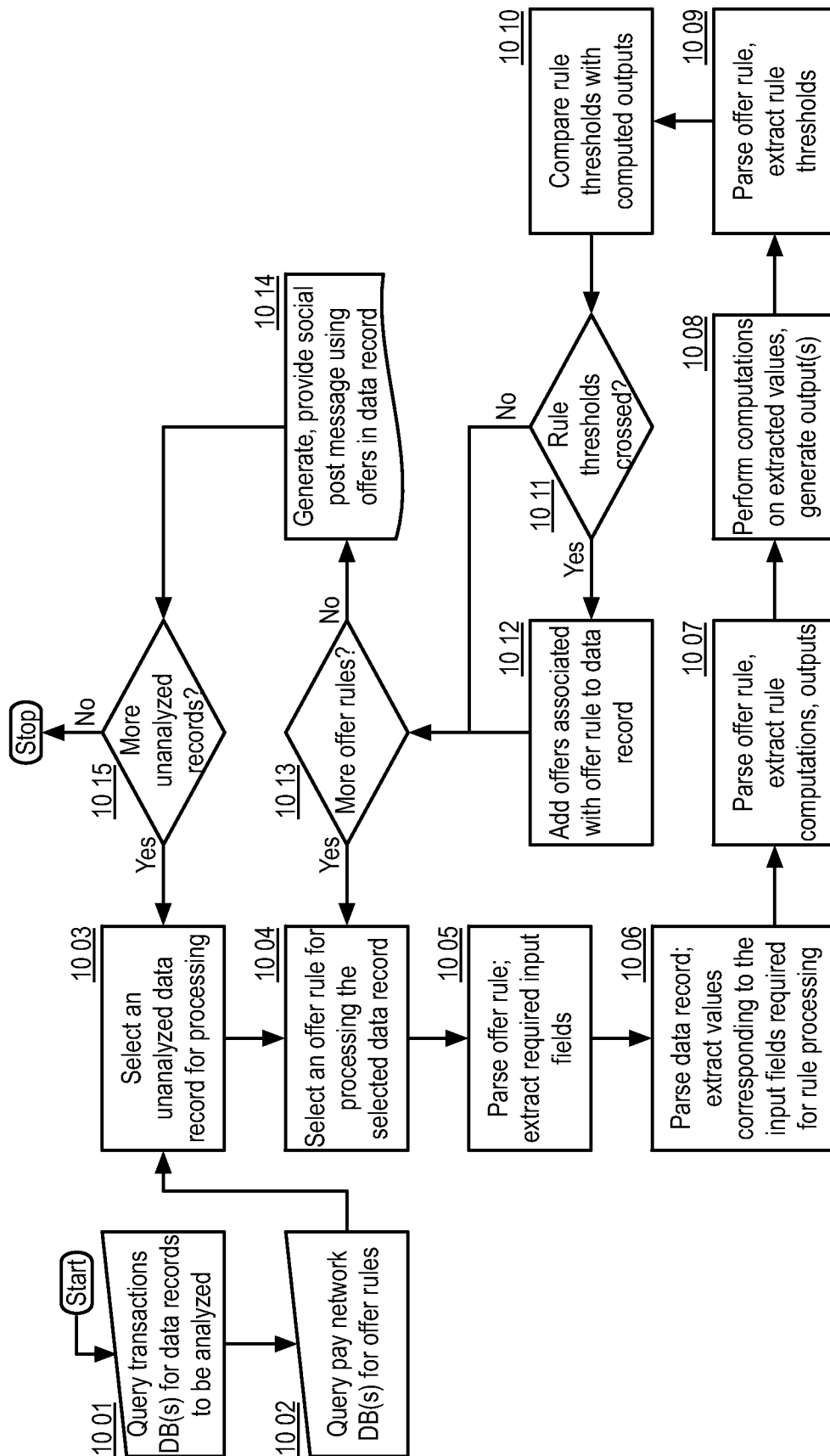
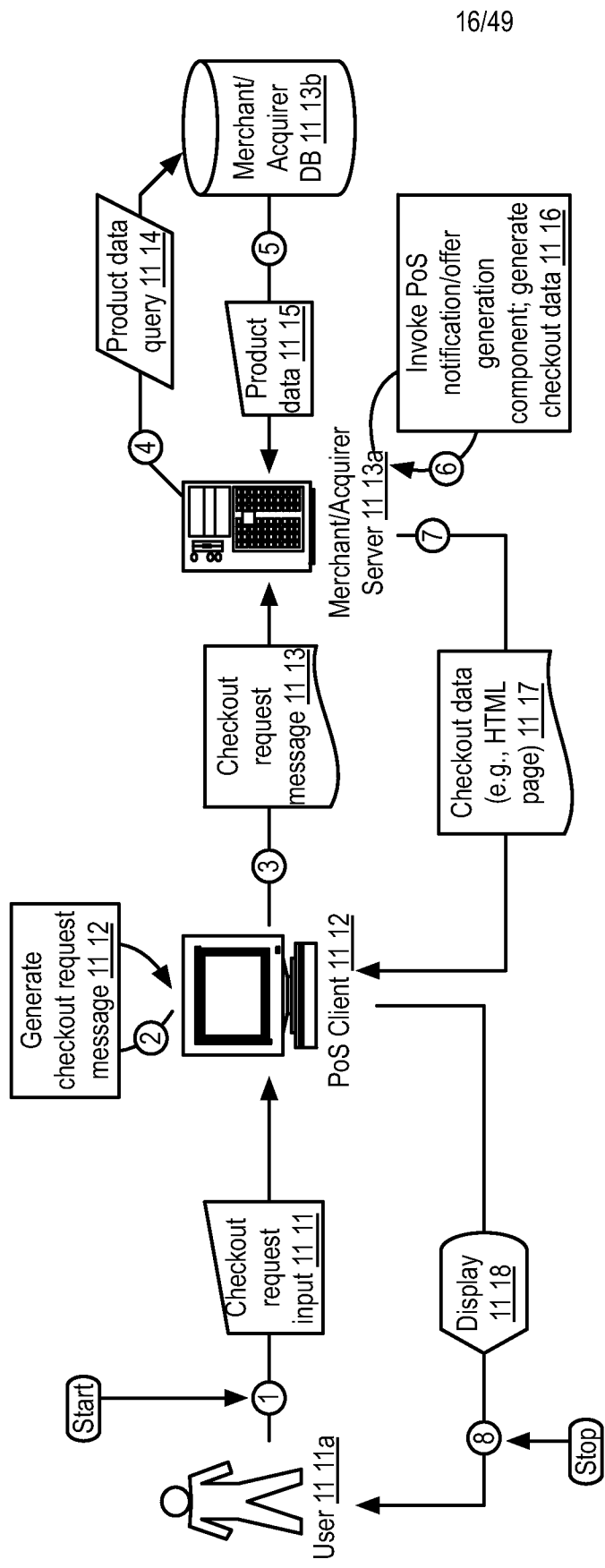


FIGURE 10

Example: Transaction-Based Offer Generation (TBOG) component 1000



Example Data Flow: User Purchase Checkout

FIGURE 11

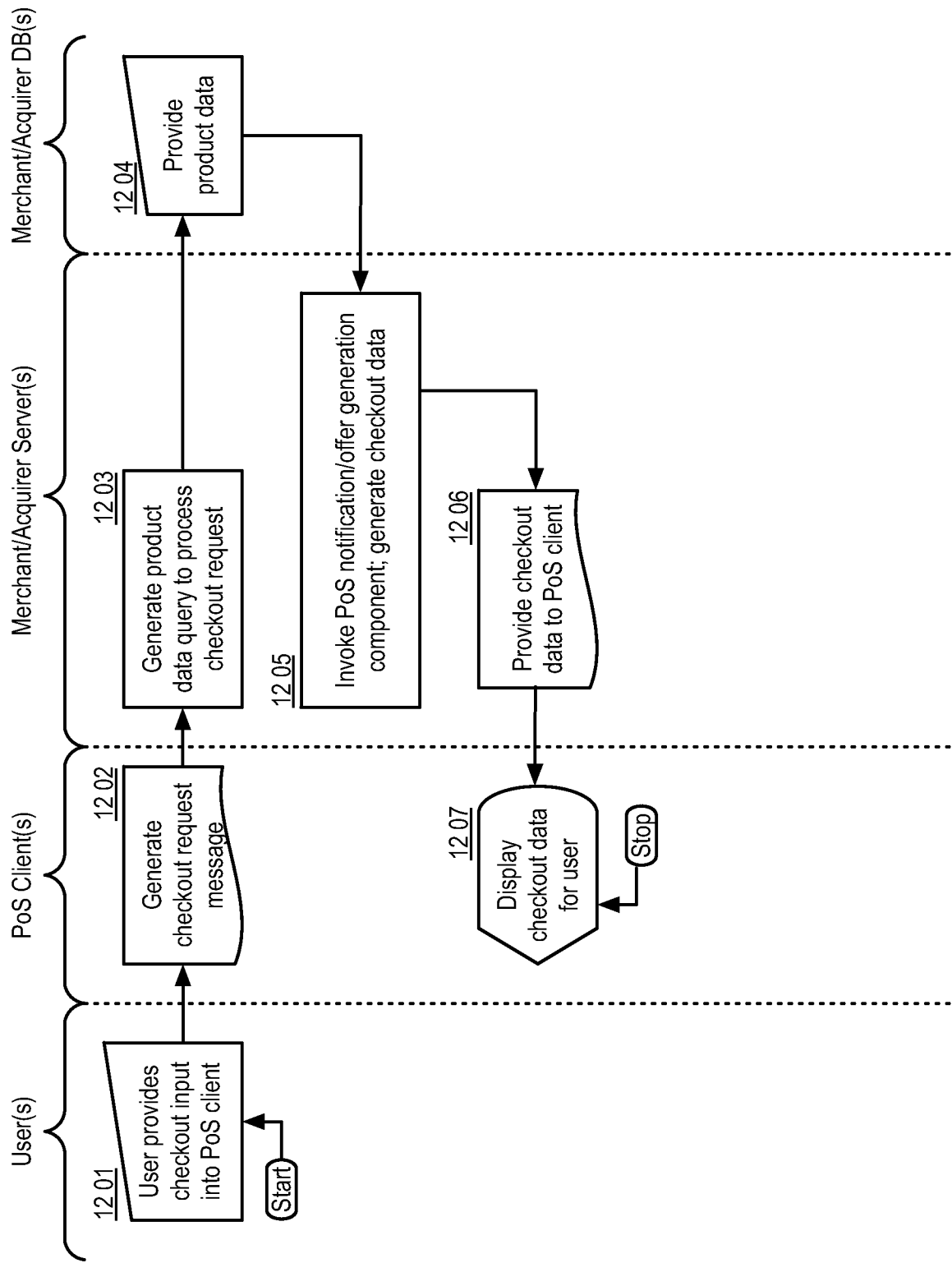


FIGURE 12 Example Logic Flow: User Purchase Checkout ("UPC") component 1200

18/49

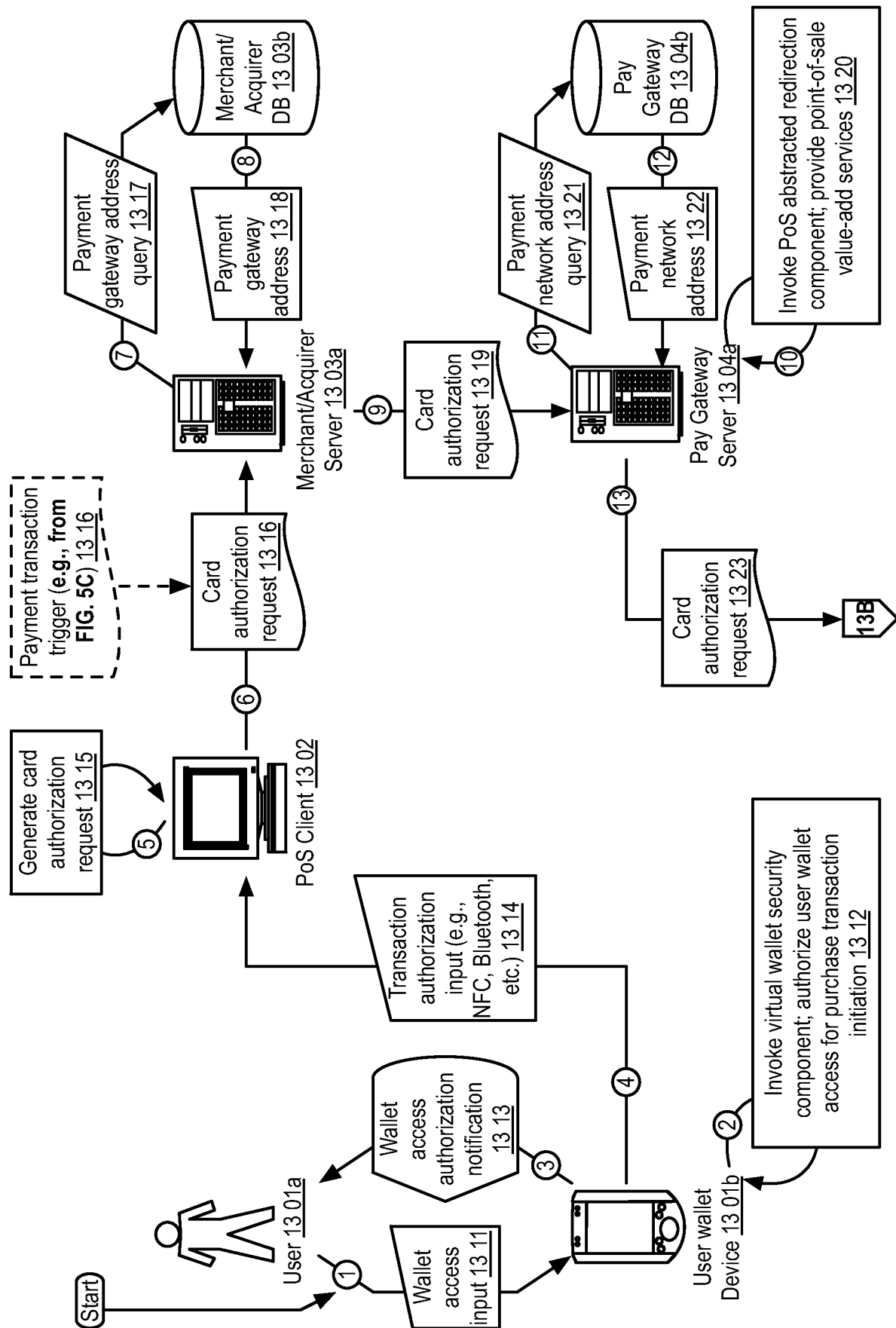
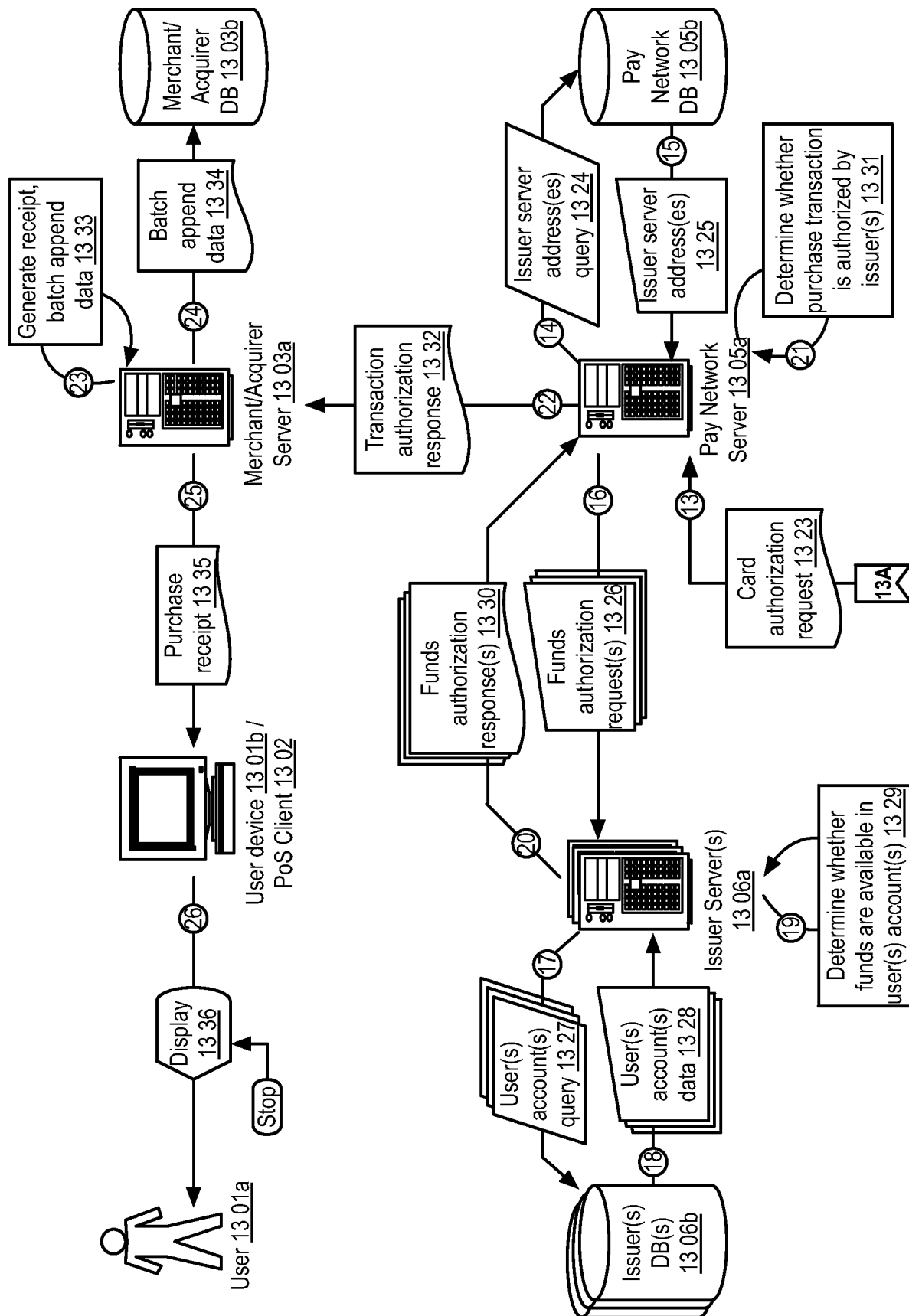


FIGURE 13A

Example Data Flow: Purchase Transaction Authorization

19/49



Example Data Flow: Purchase Transaction Authorization

FIGURE 13B

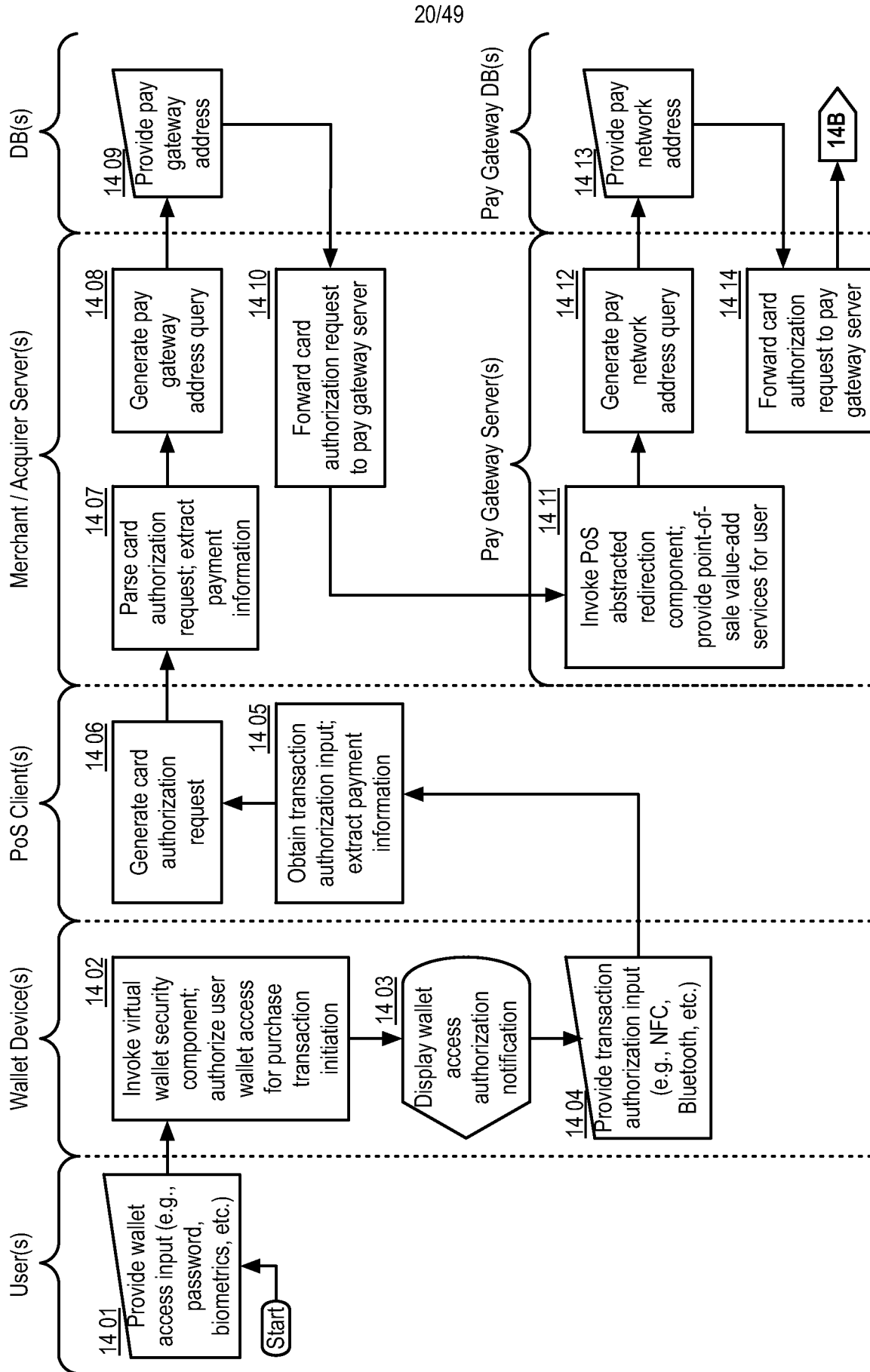
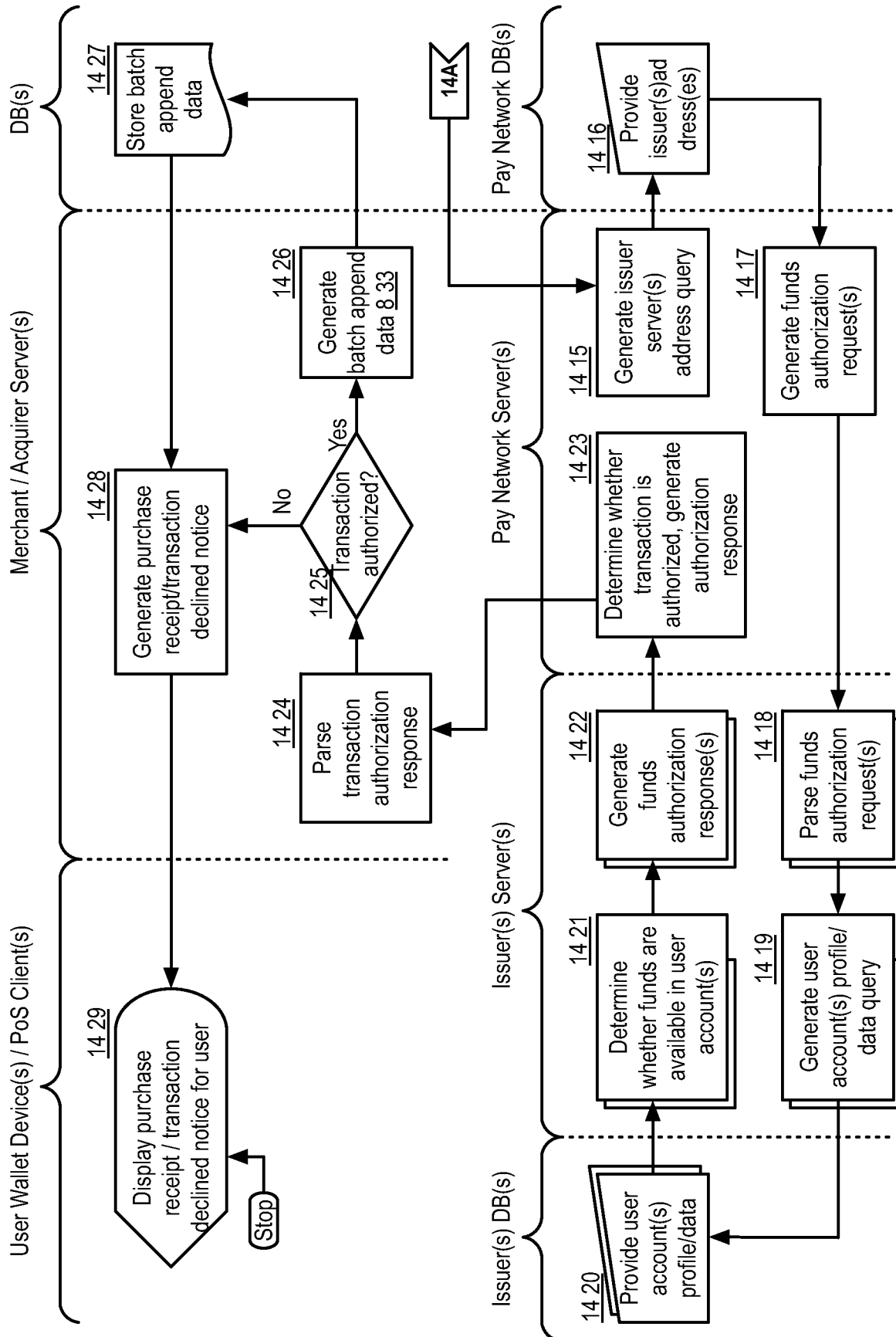


FIGURE 14A Example: Purchase Transaction Authorization ("PTA") component 1400



Example: Purchase Transaction Authorization ("PTA") component 1400

FIGURE 14B

22/49

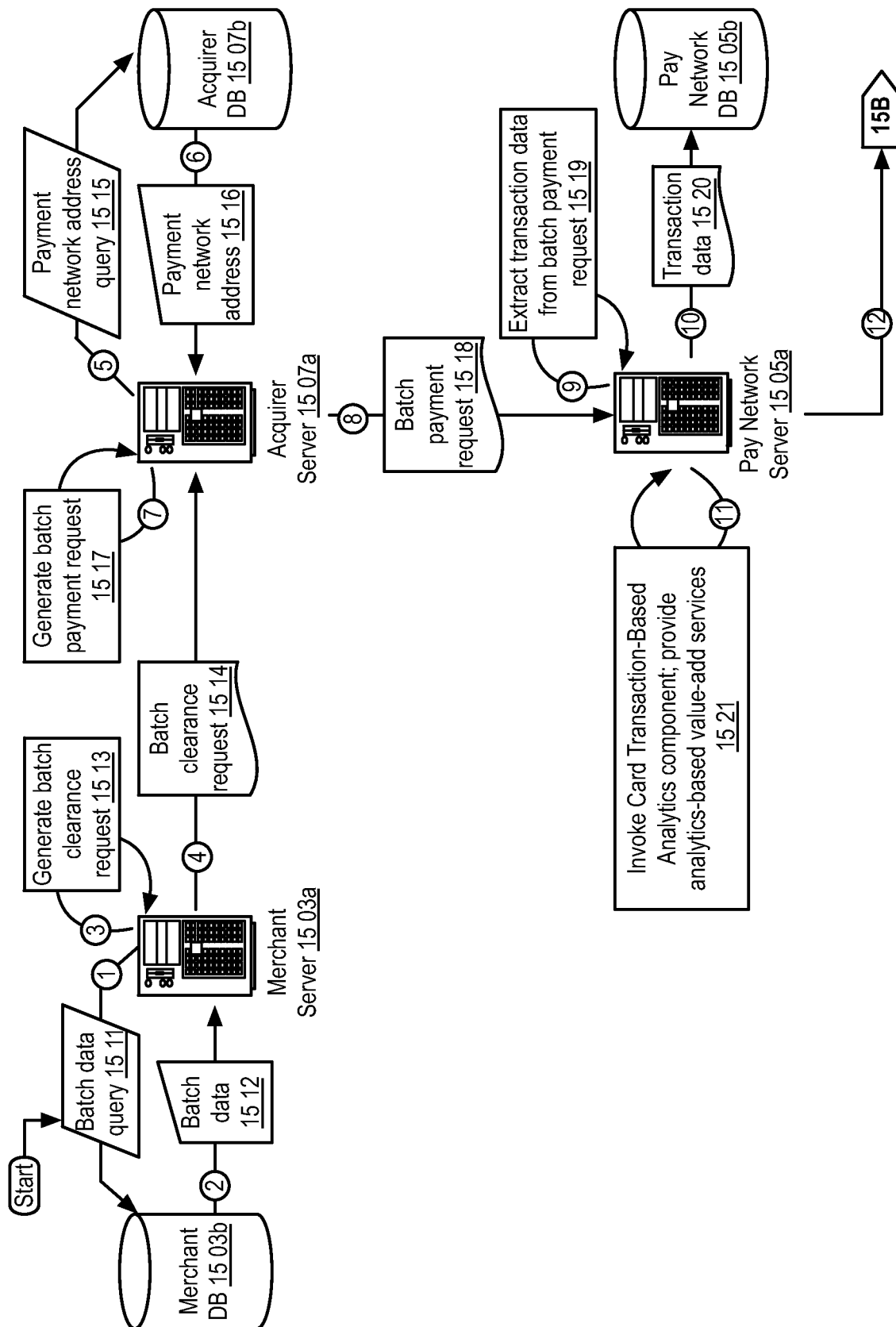


FIGURE 15A

Example Data Flow: Purchase Transaction Clearance

23/49

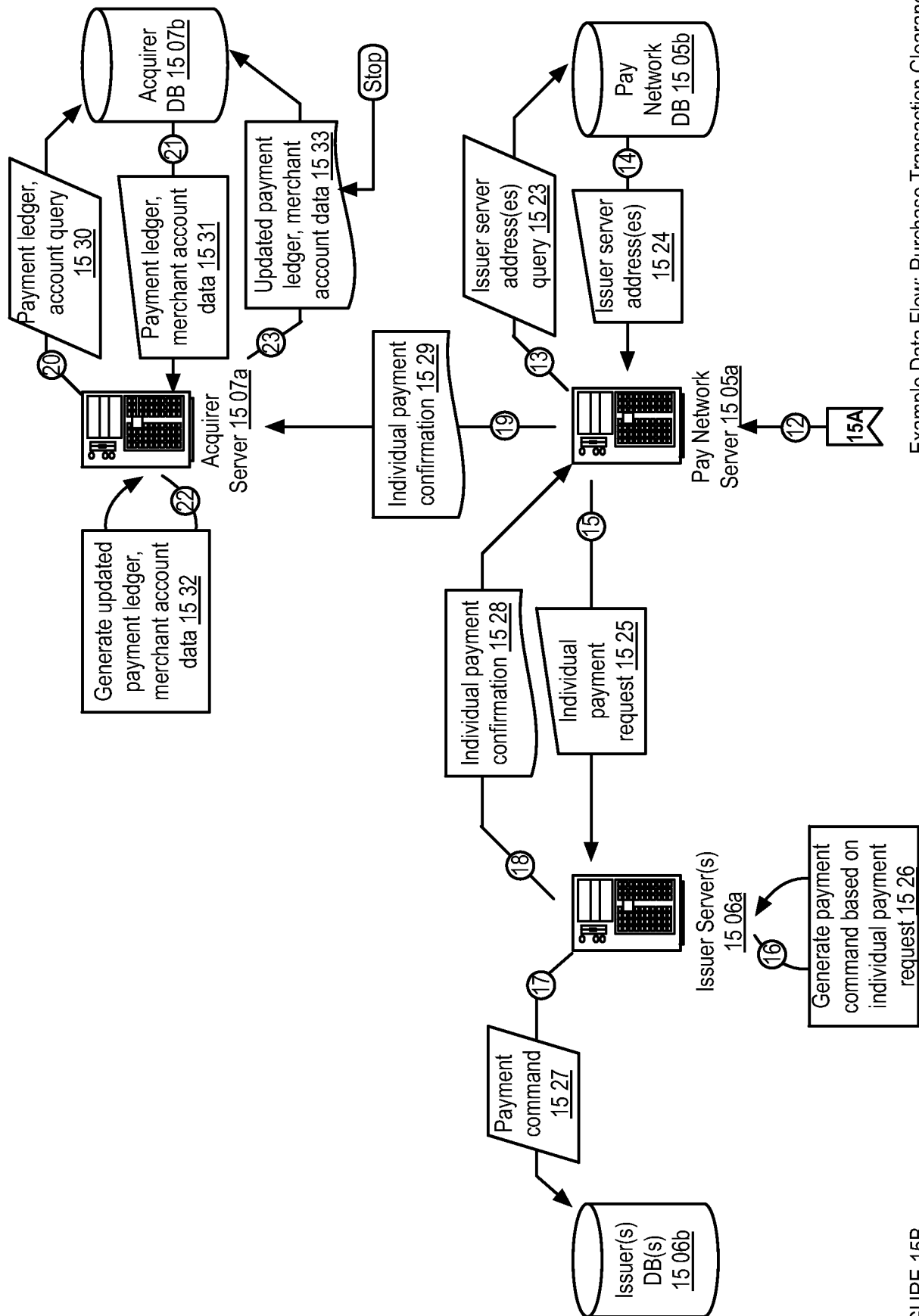


FIGURE 15B Example Data Flow: Purchase Transaction Clearance

24/49

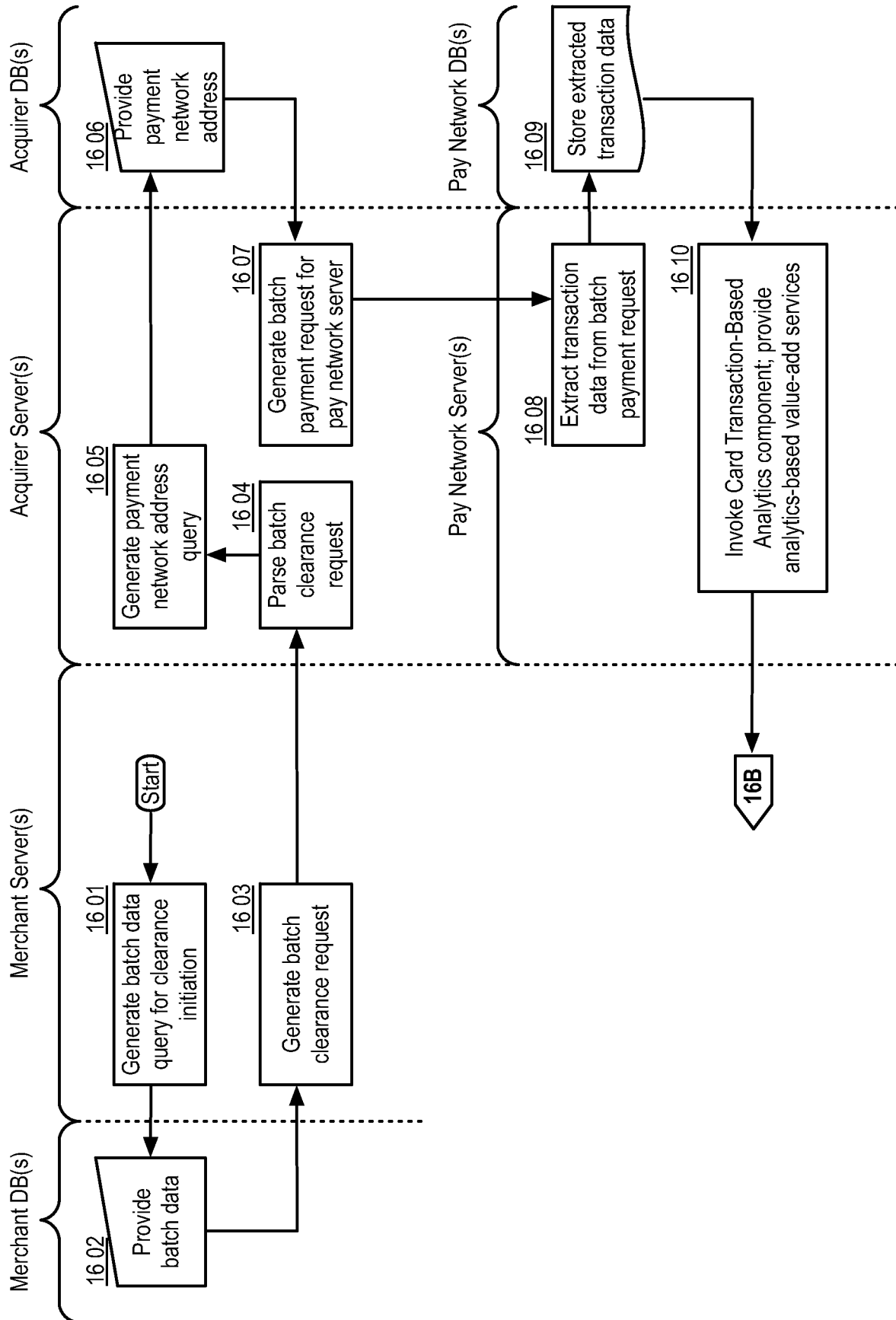


FIGURE 16A Example Logic Flow: Purchase Transaction Clearance ("PTC") component 1600

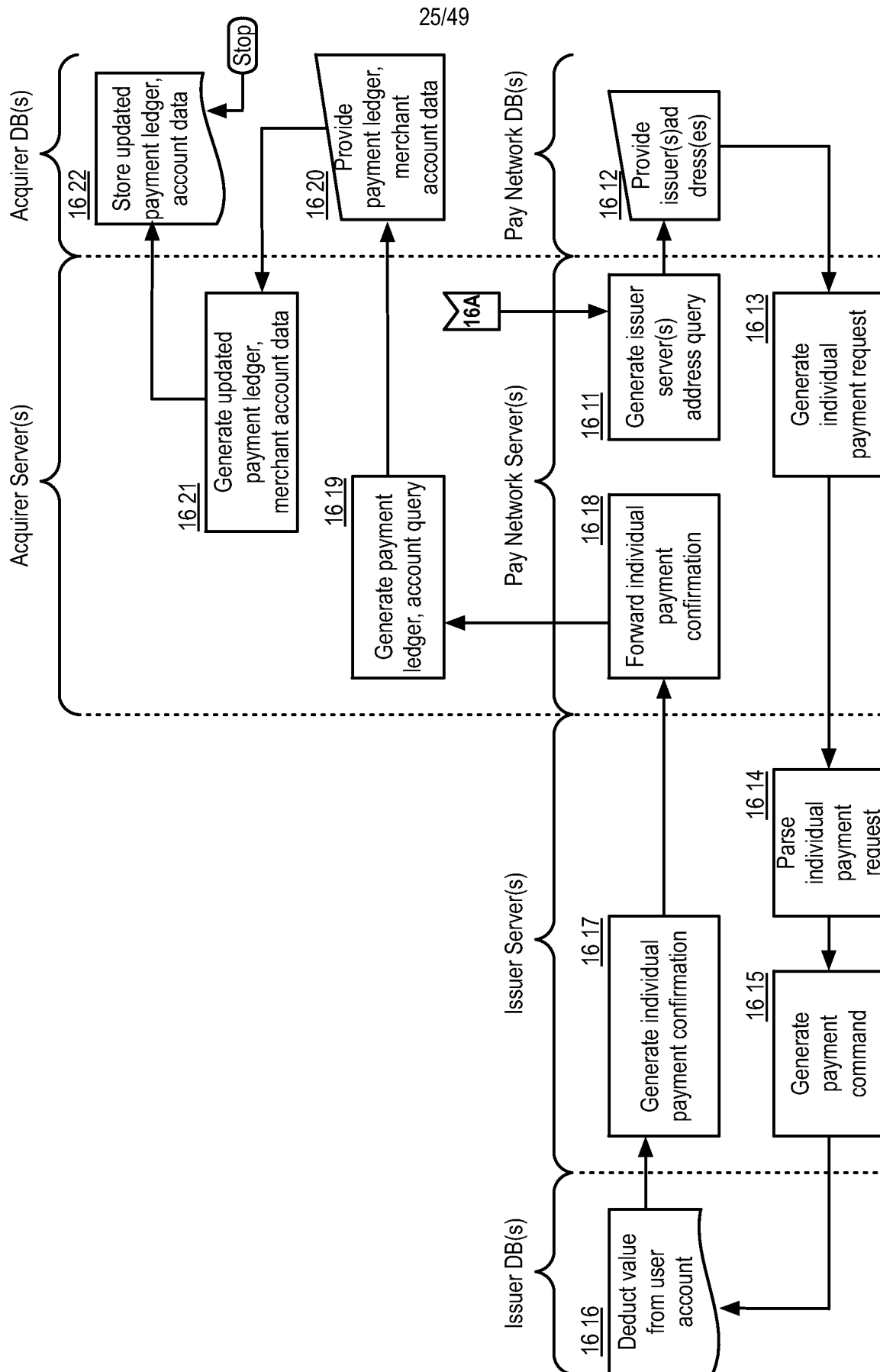


FIGURE 16B Example Logic Flow: Purchase Transaction Clearance ("PTC") component 1600

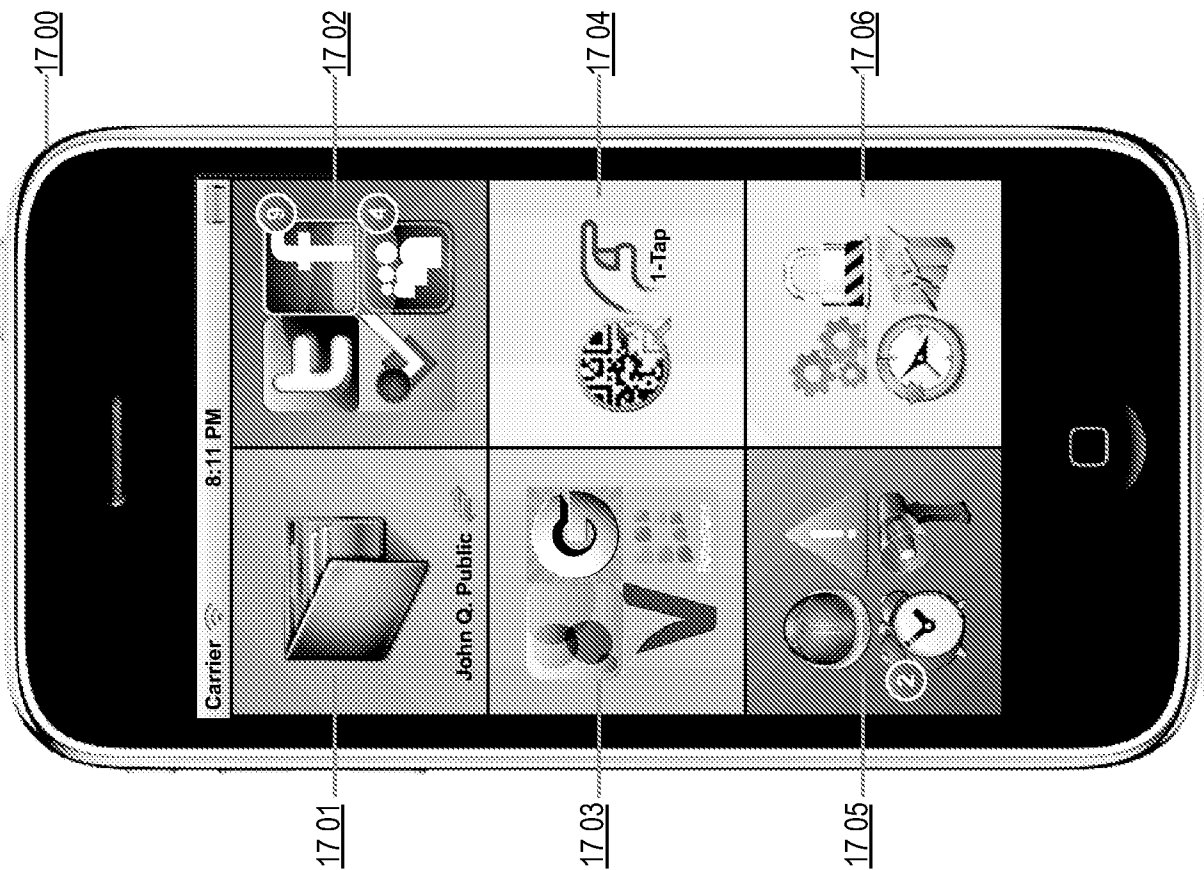


FIGURE 17

Example: Virtual Wallet Mobile App - Feature Overview

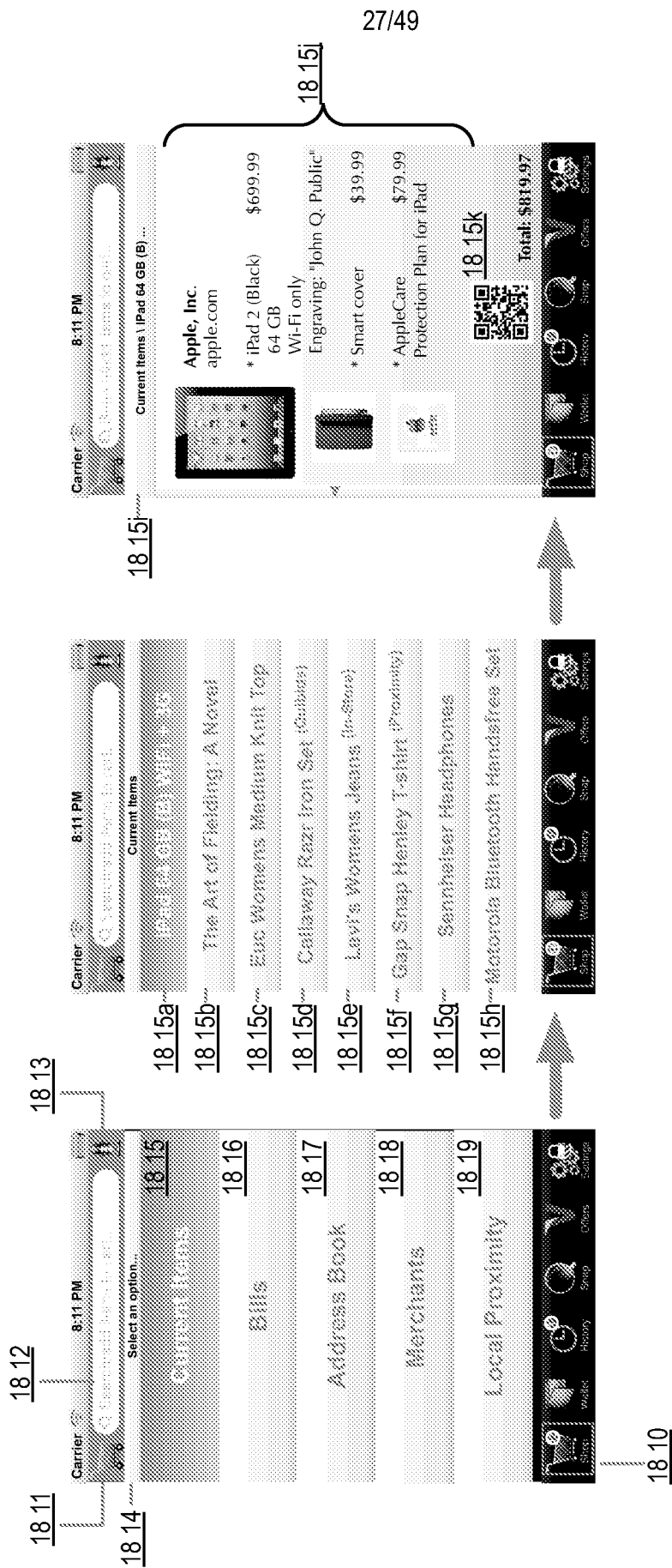
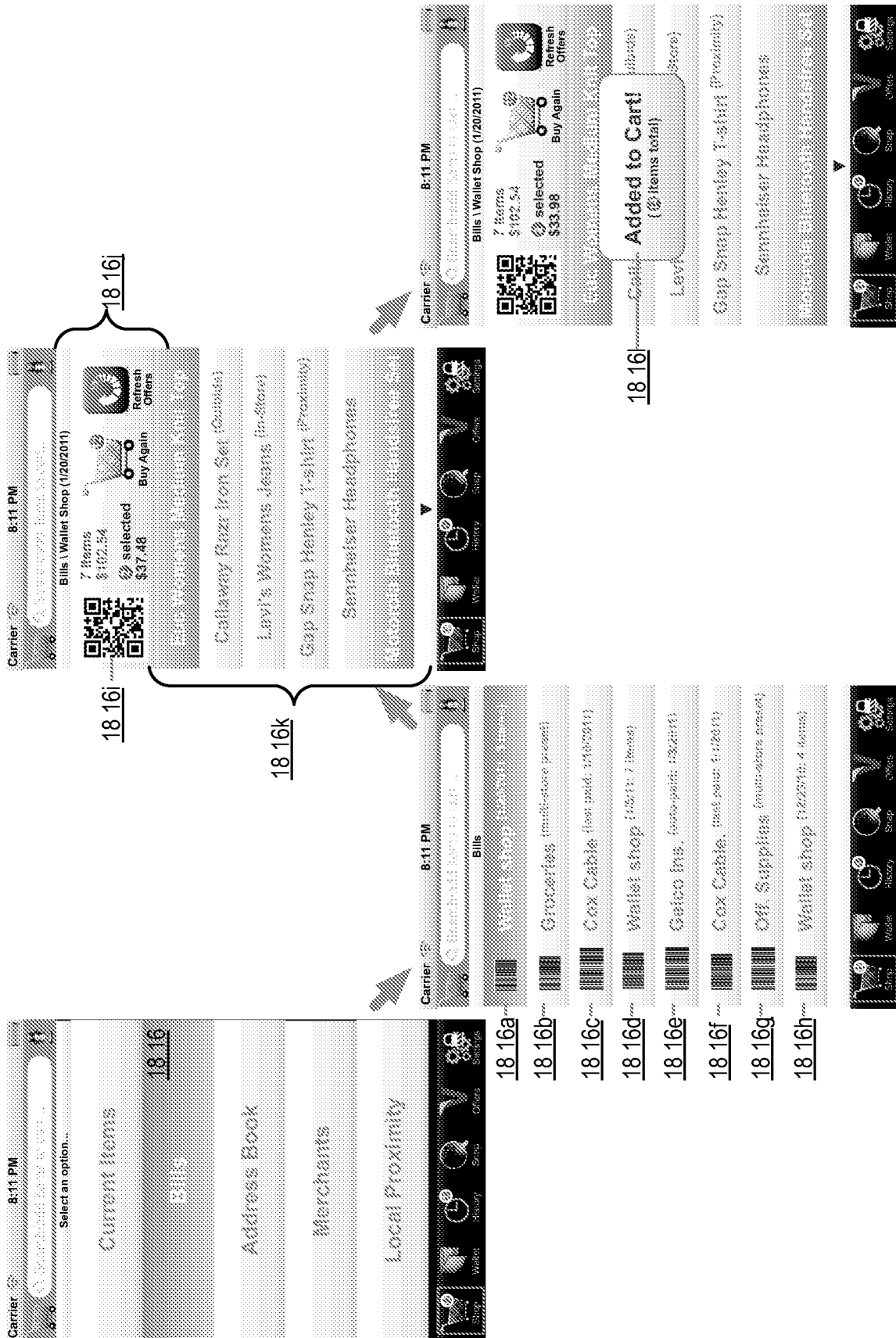


FIGURE 18A

Example: Virtual Wallet Mobile App - Shopping Mode



Example: Virtual Wallet Mobile App - Shopping Mode

FIGURE 18B

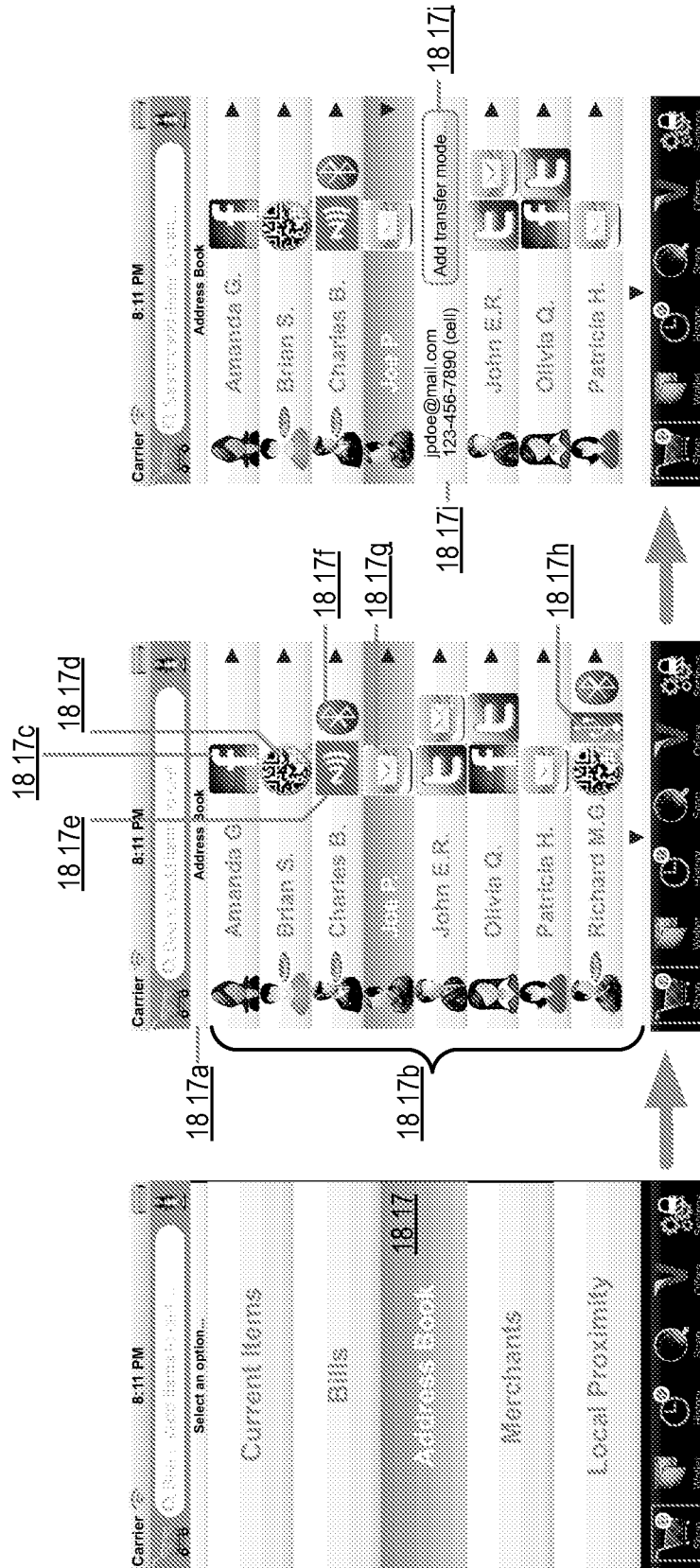


FIGURE 18C

Example: Virtual Wallet Mobile App - Shopping Mode

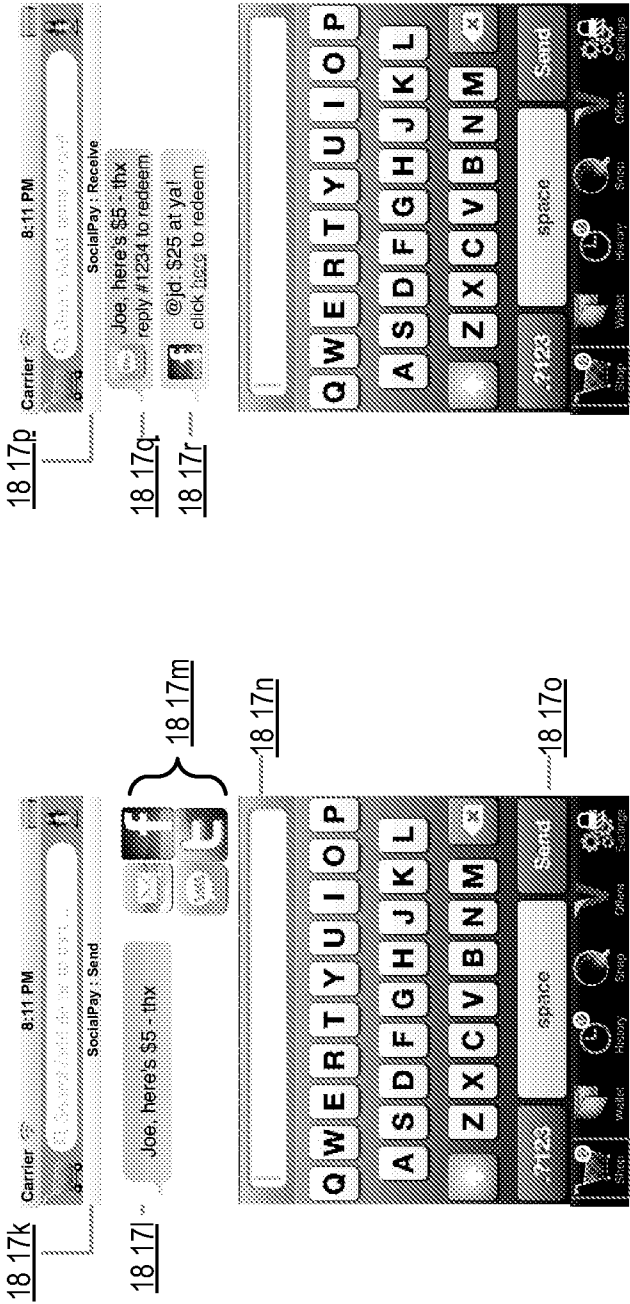


FIGURE 18D

Example: Virtual Wallet Mobile App - SocialPay Mode



FIGURE 18E

Example: Virtual Wallet Mobile App - Shopping Mode

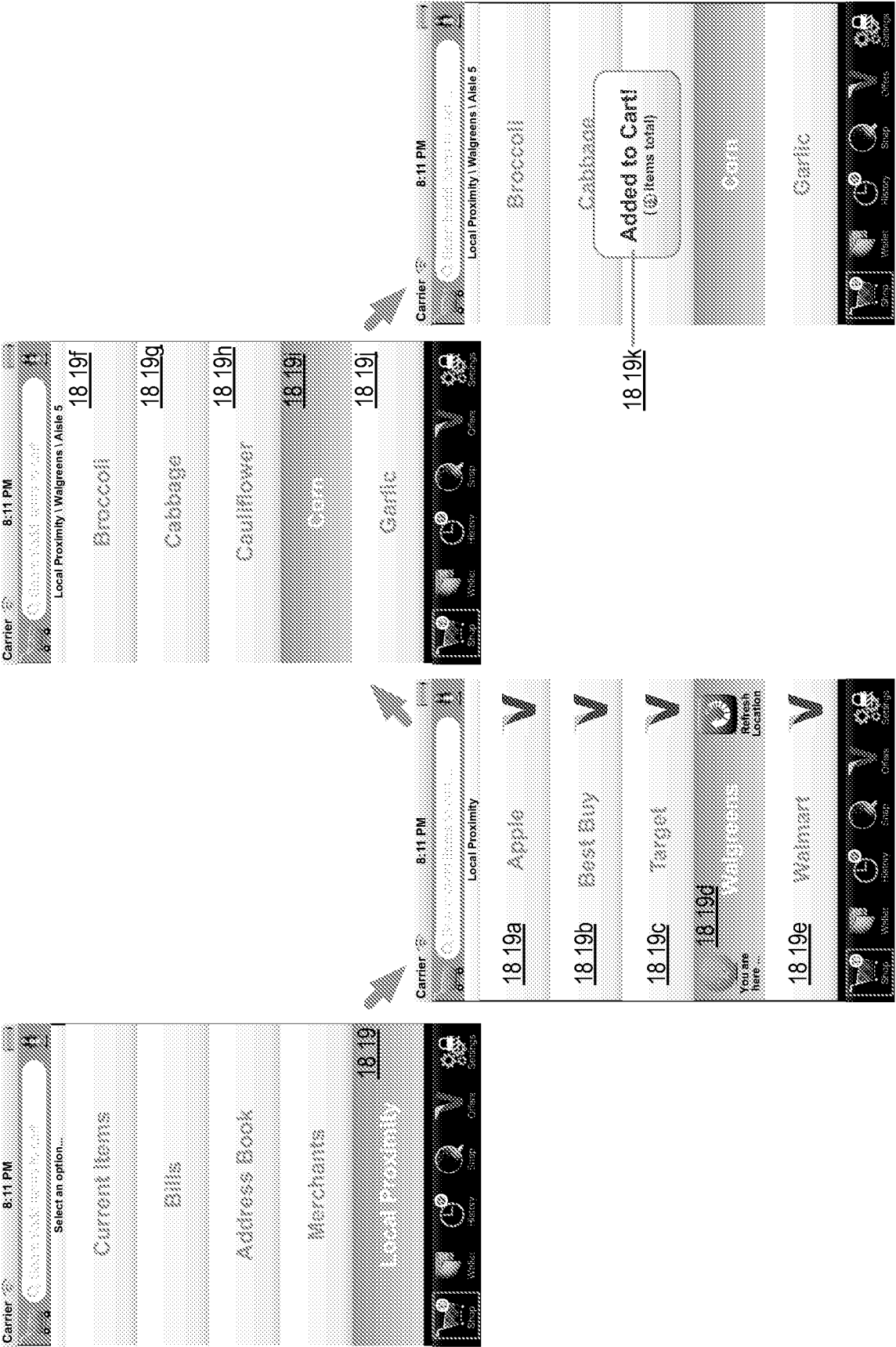
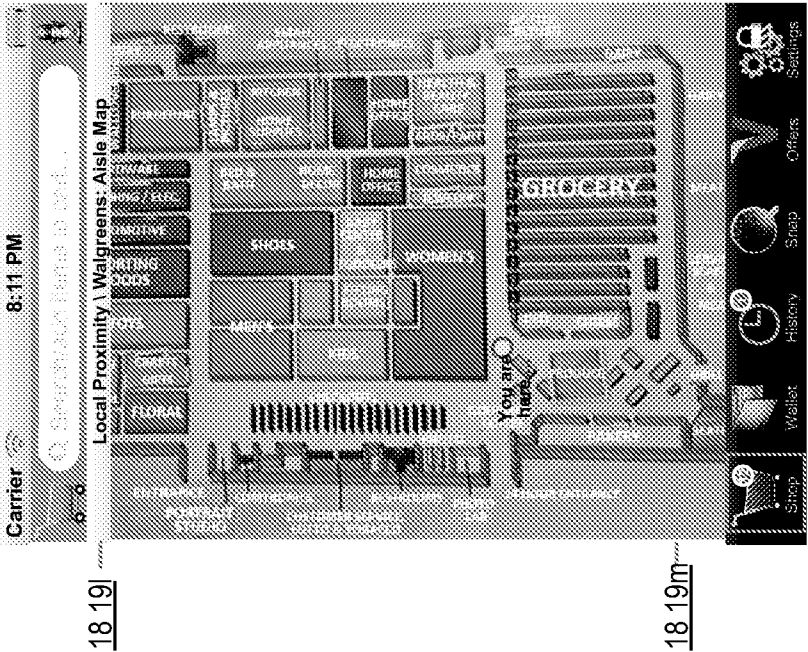
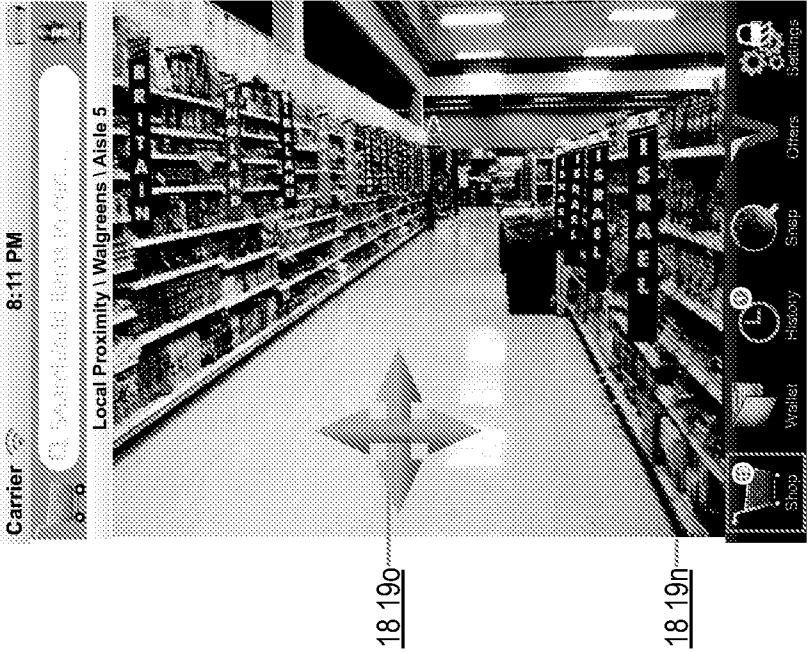


FIGURE 18F

Example: Virtual Wallet Mobile App - Shopping Mode



Example: Virtual Wallet Mobile App - Shopping Mode

FIGURE 18G

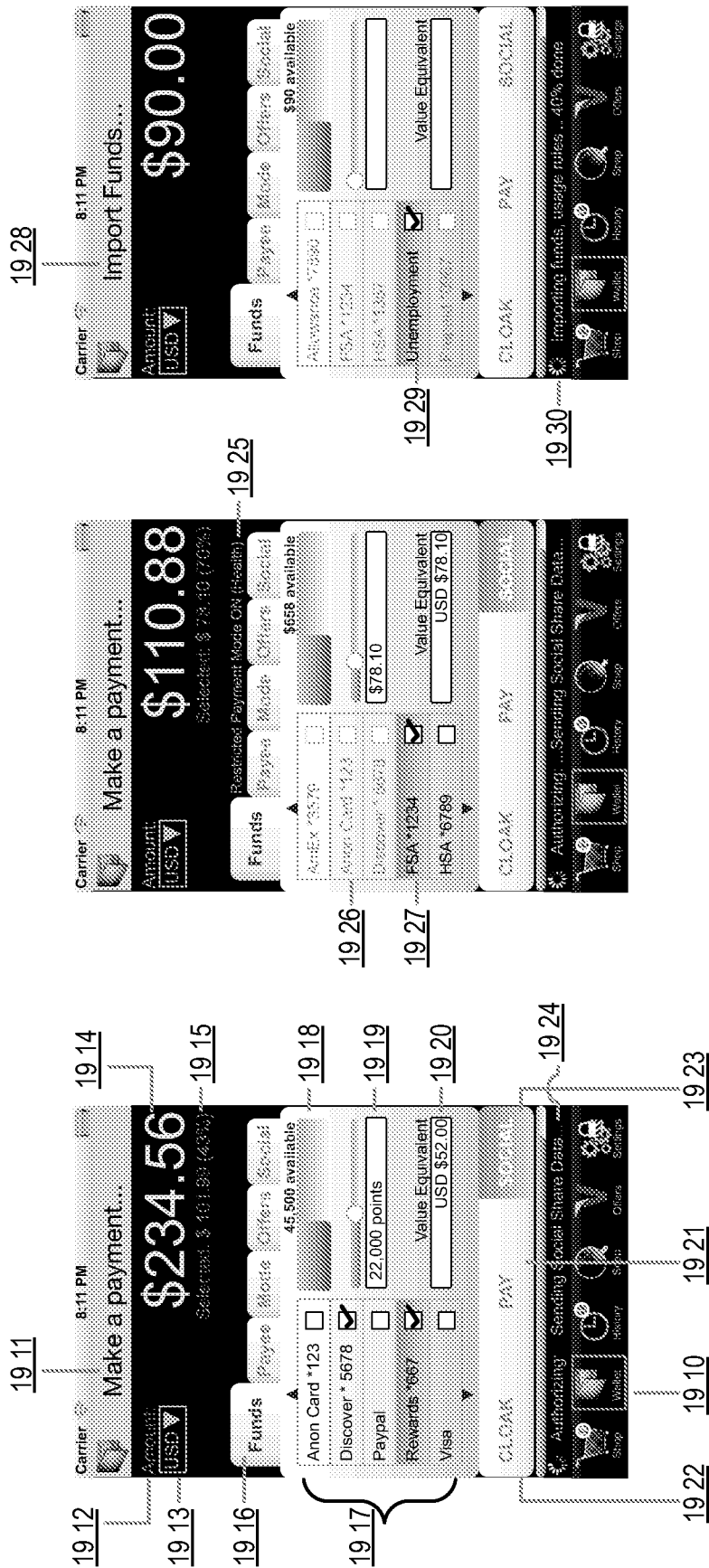


FIGURE 19A

Example: Virtual Wallet Mobile App - Payment Mode

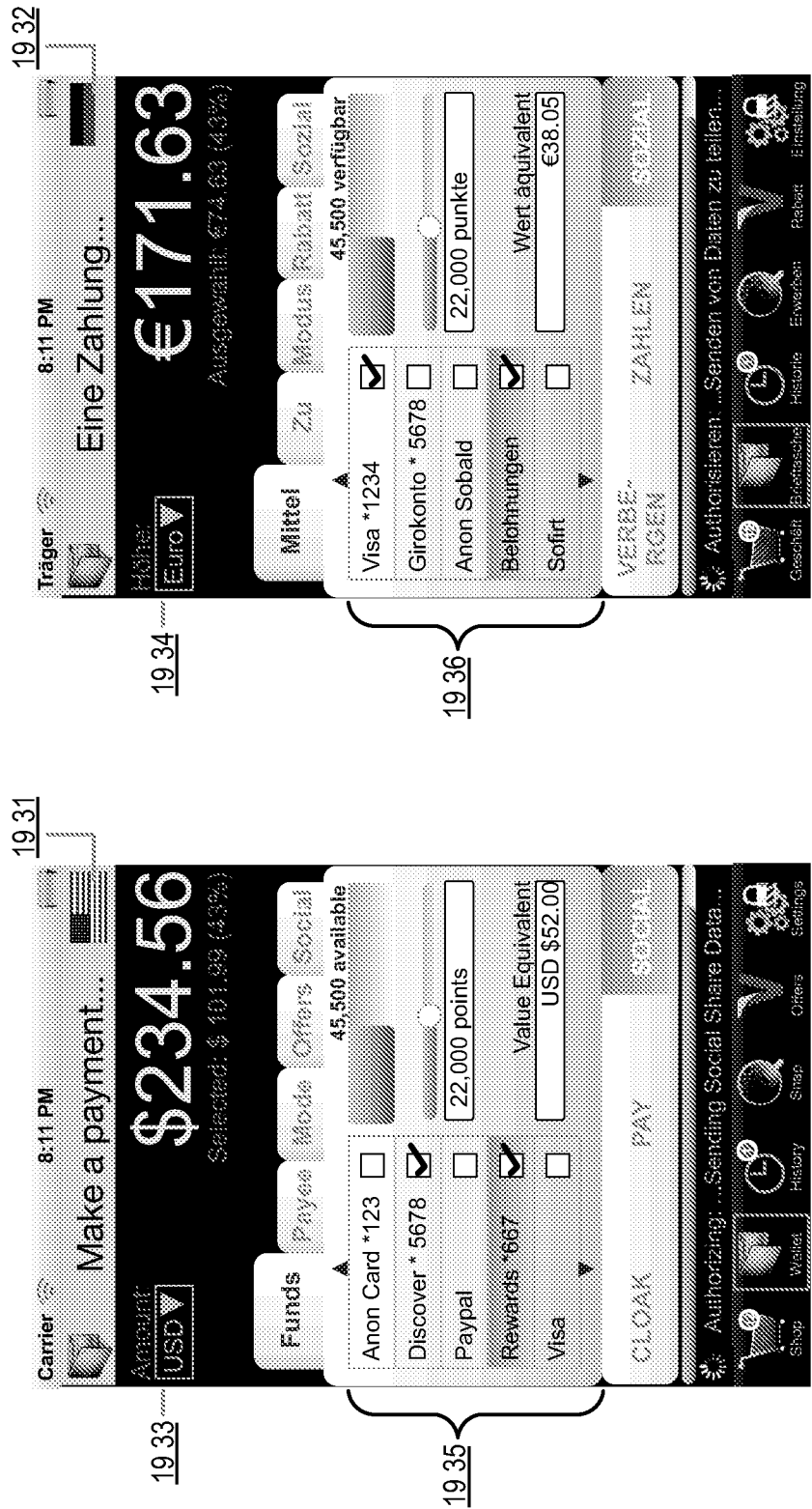
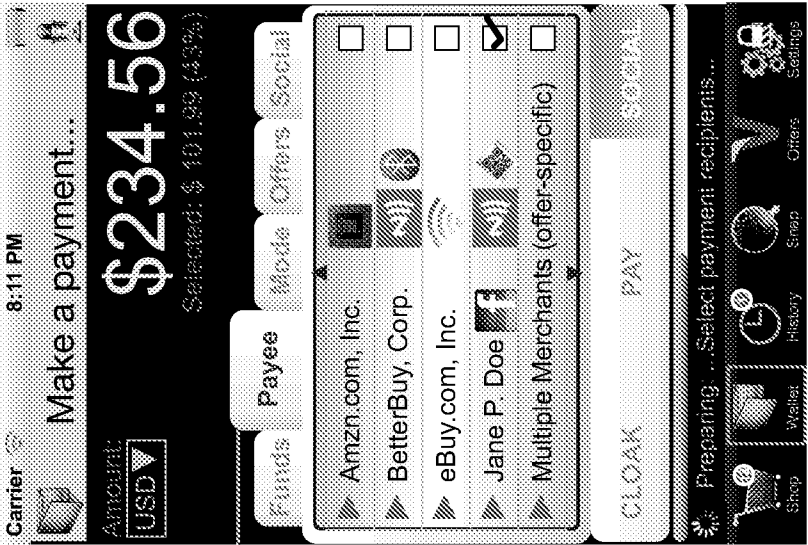


FIGURE 19B

Example: Virtual Wallet Mobile App - Dynamic Payment Optimization



19 39



19 37

19 38

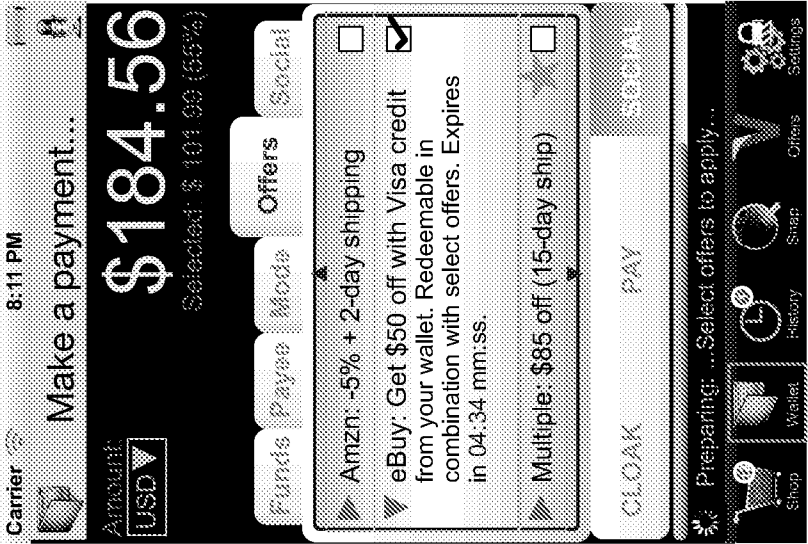
Example: Virtual Wallet Mobile App

FIGURE 19C

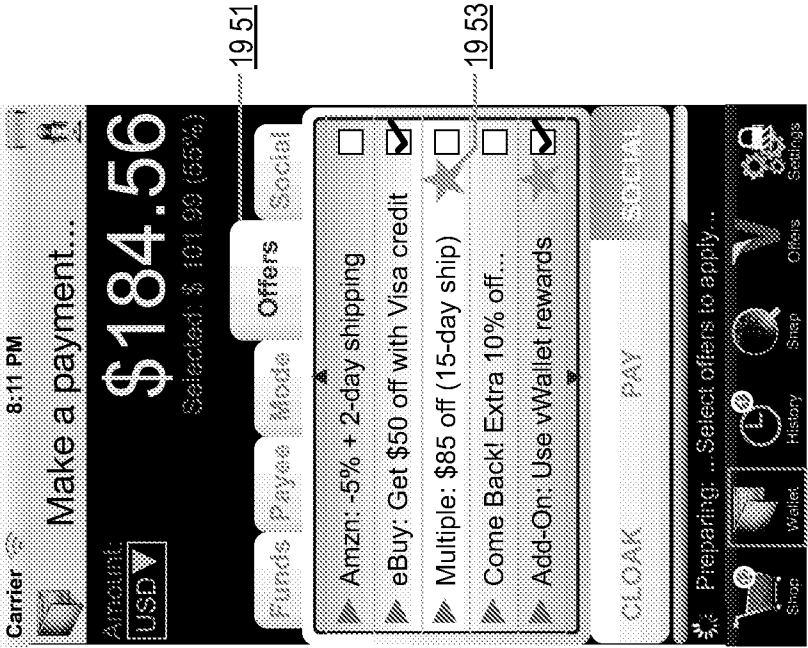


FIGURE 19D

Example: Virtual Wallet Mobile App



19 54



19 52

19 51

19 53

FIGURE 19E

Example: Virtual Wallet Mobile App

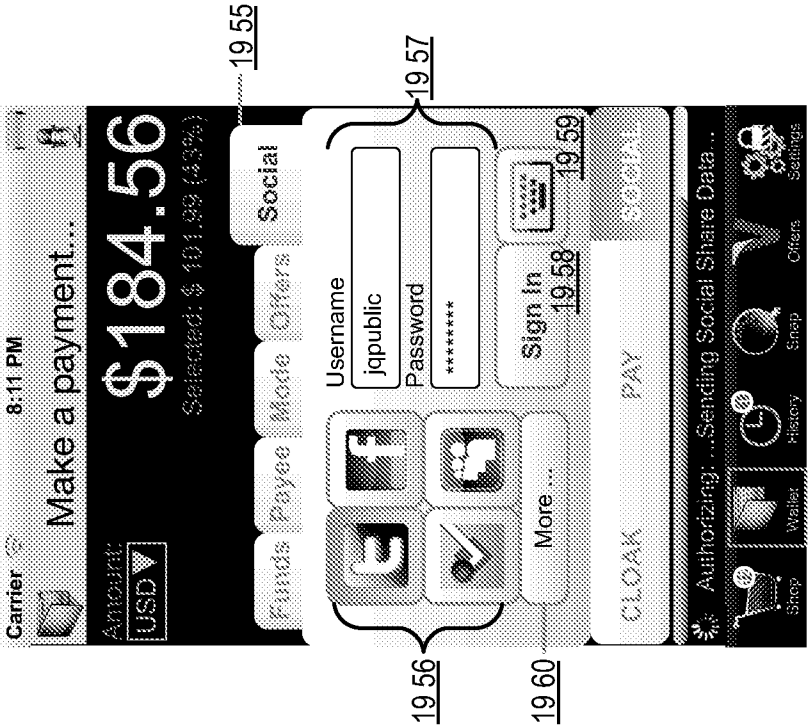


FIGURE 19F

Example: Virtual Wallet Mobile App

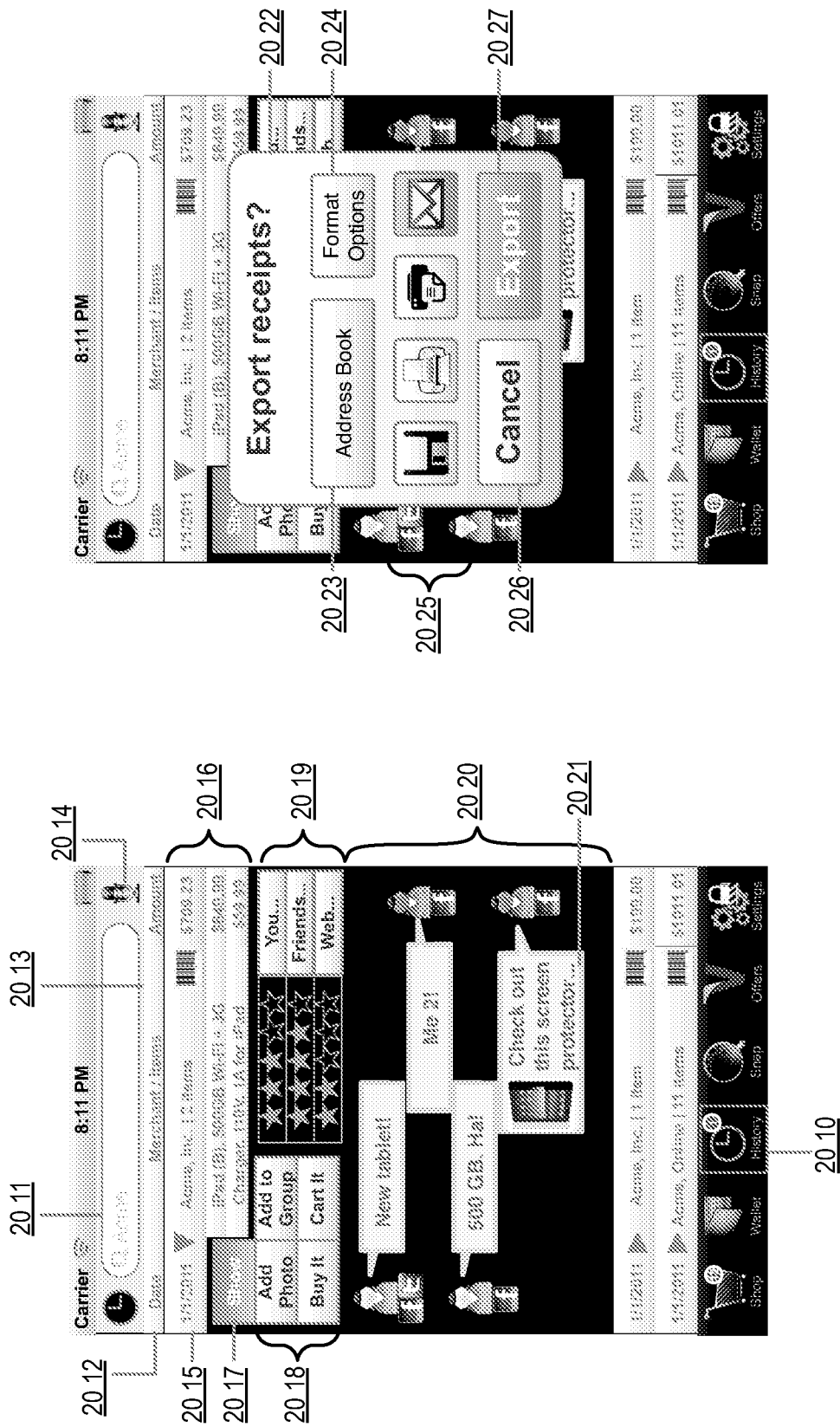


FIGURE 20 Example: Virtual Wallet Mobile App - History

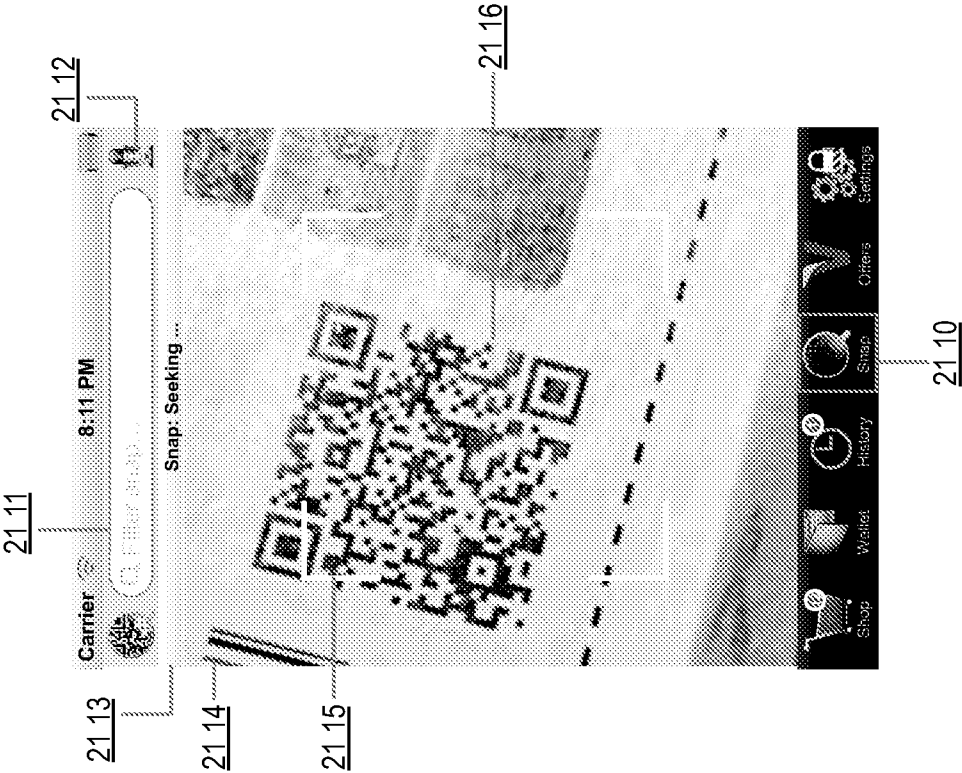


FIGURE 21A

Example: Virtual Wallet Mobile App - Snap Mode

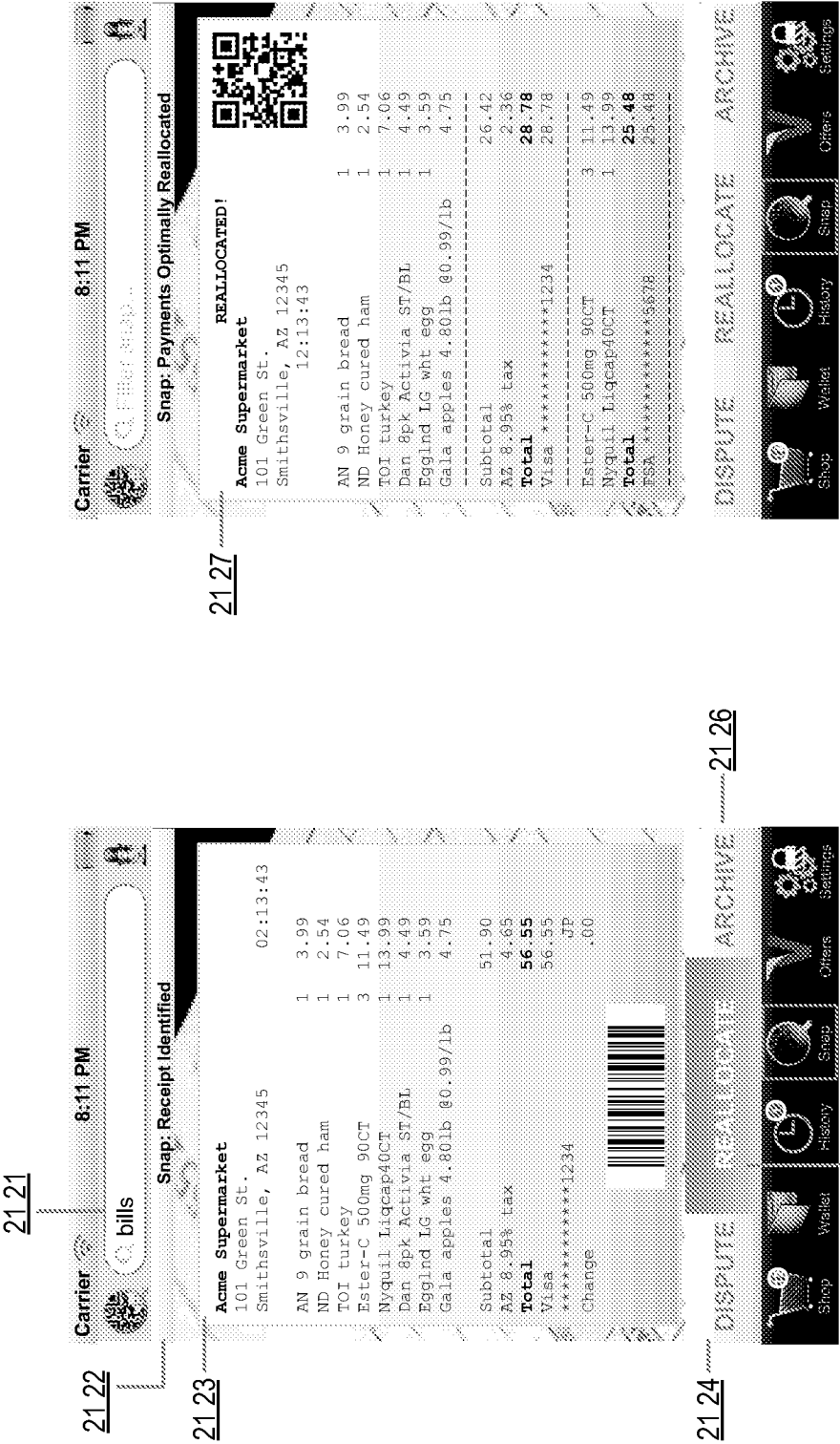
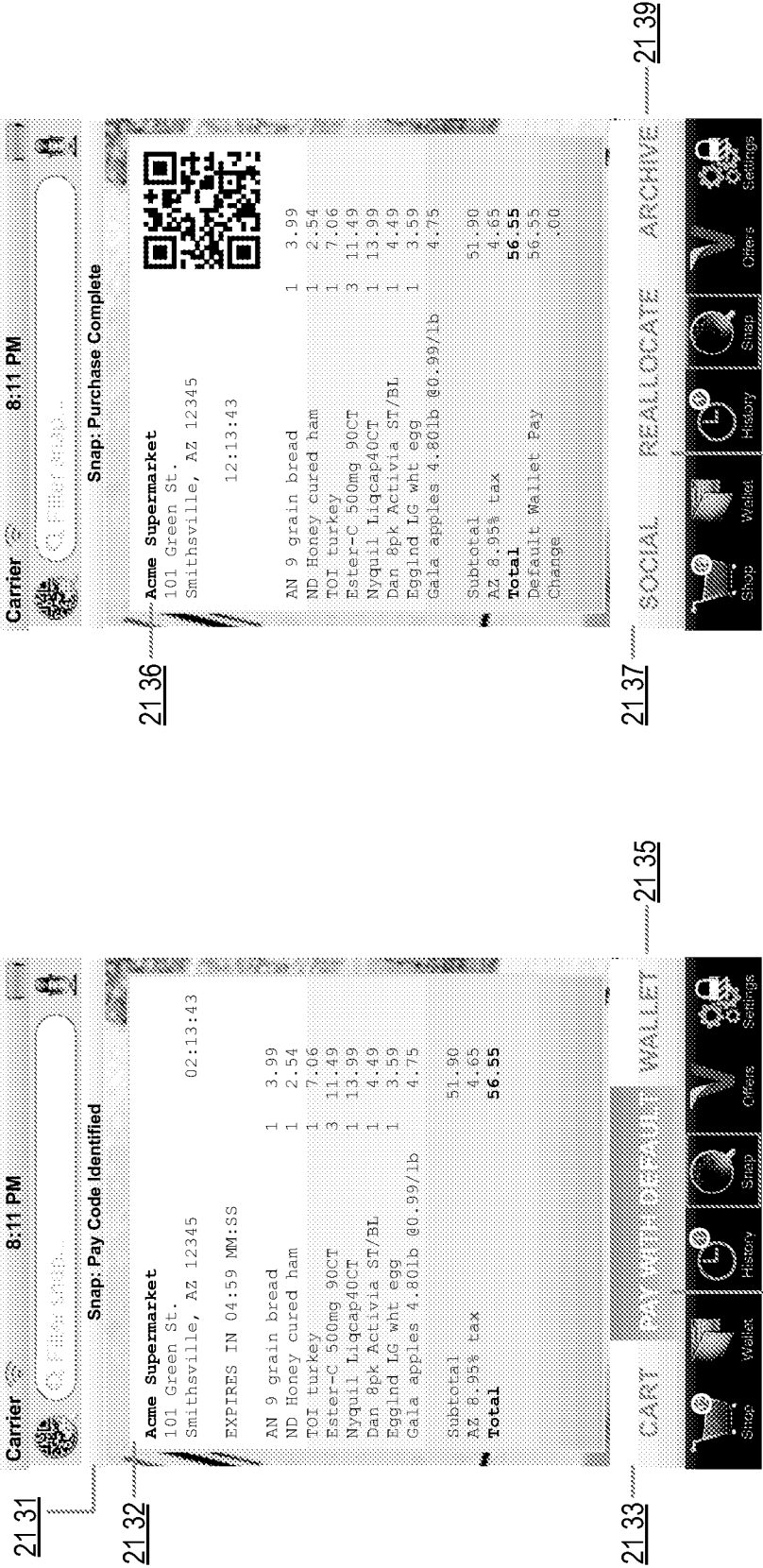


FIGURE 21B Example: Virtual Wallet Mobile App - Snap Mode



Example: Virtual Wallet Mobile App - Snap Mode

FIGURE 21C

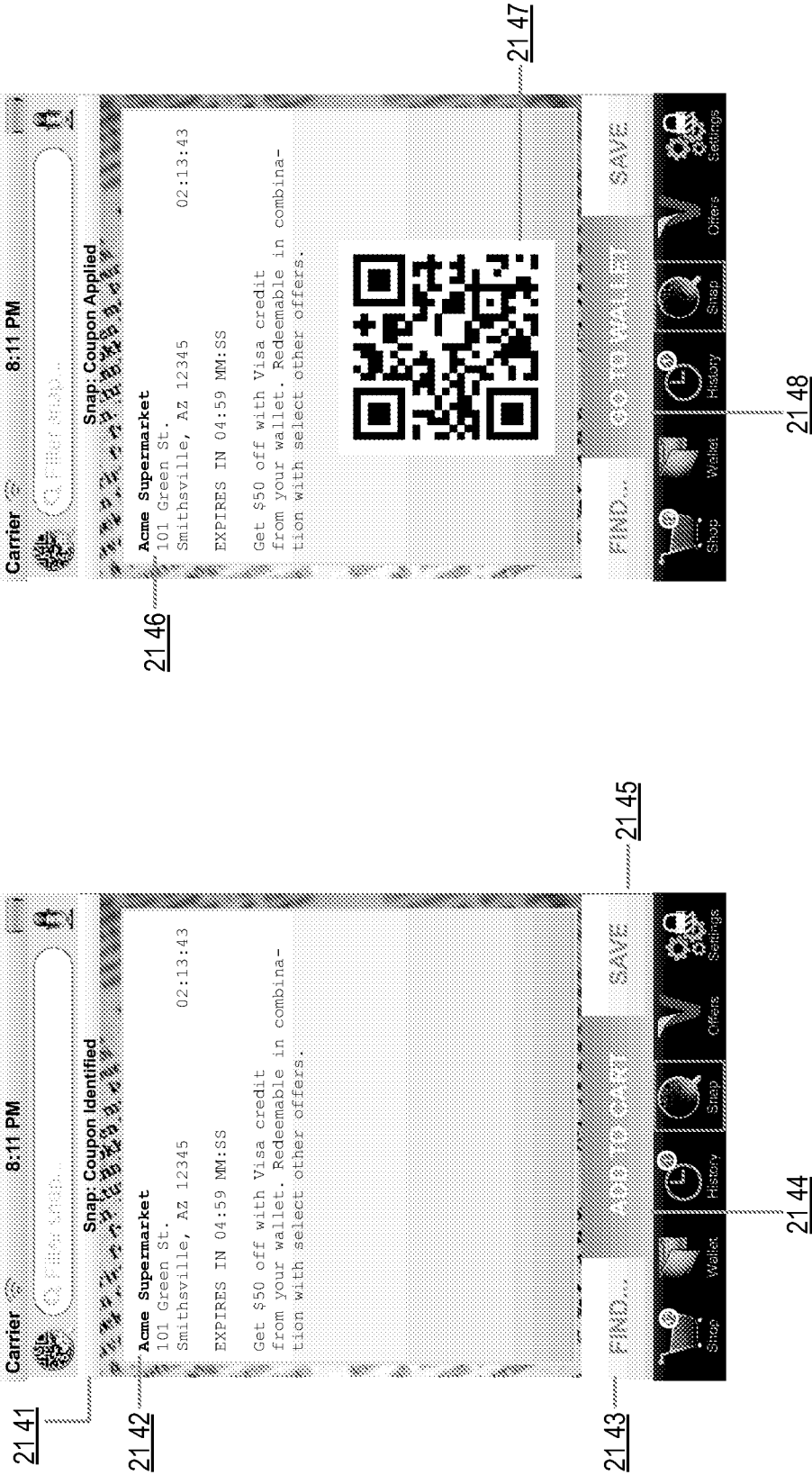


FIGURE 21D

Example: Virtual Wallet Mobile App - Snap Mode

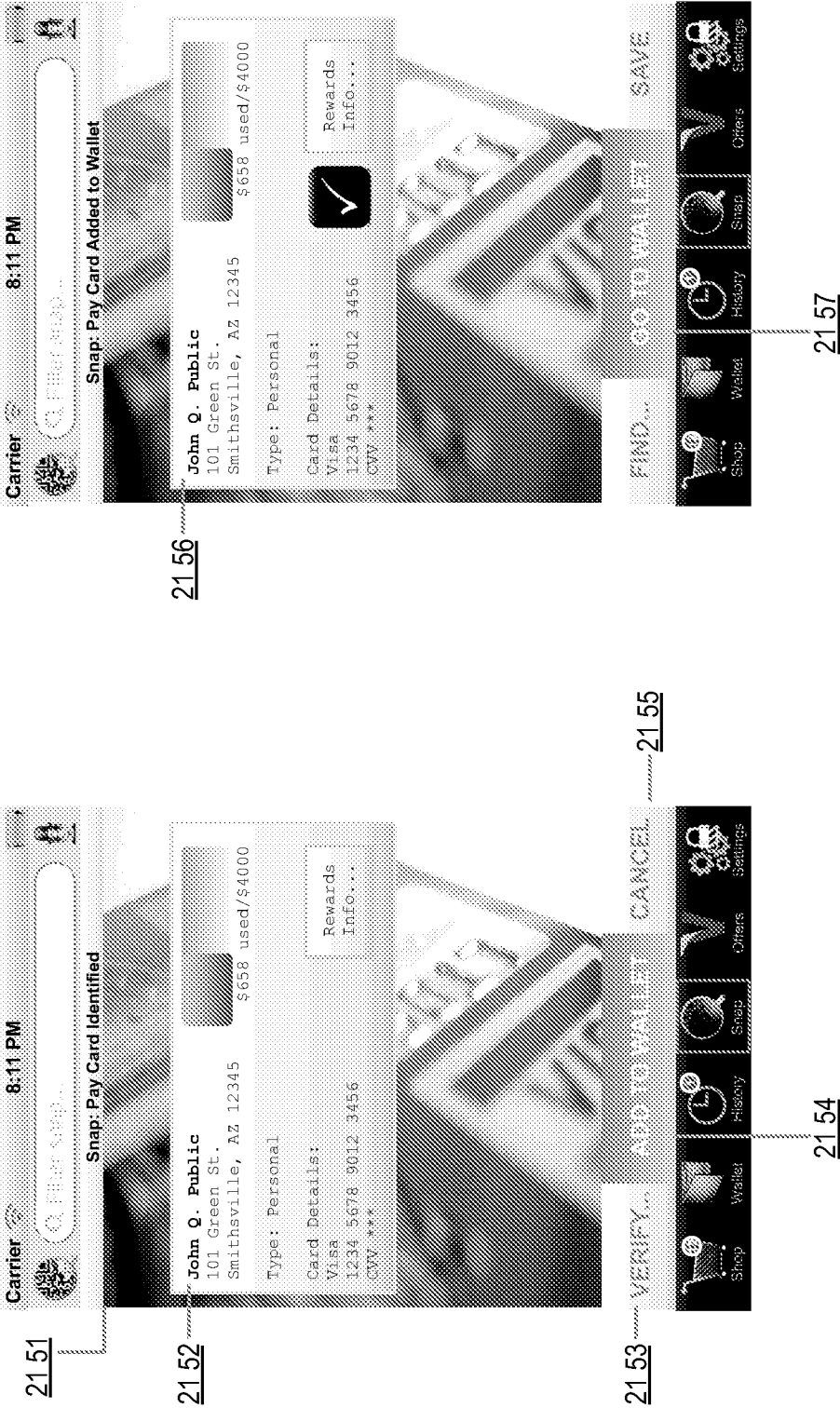


FIGURE 21E

Example: Virtual Wallet Mobile App - Snap Mode

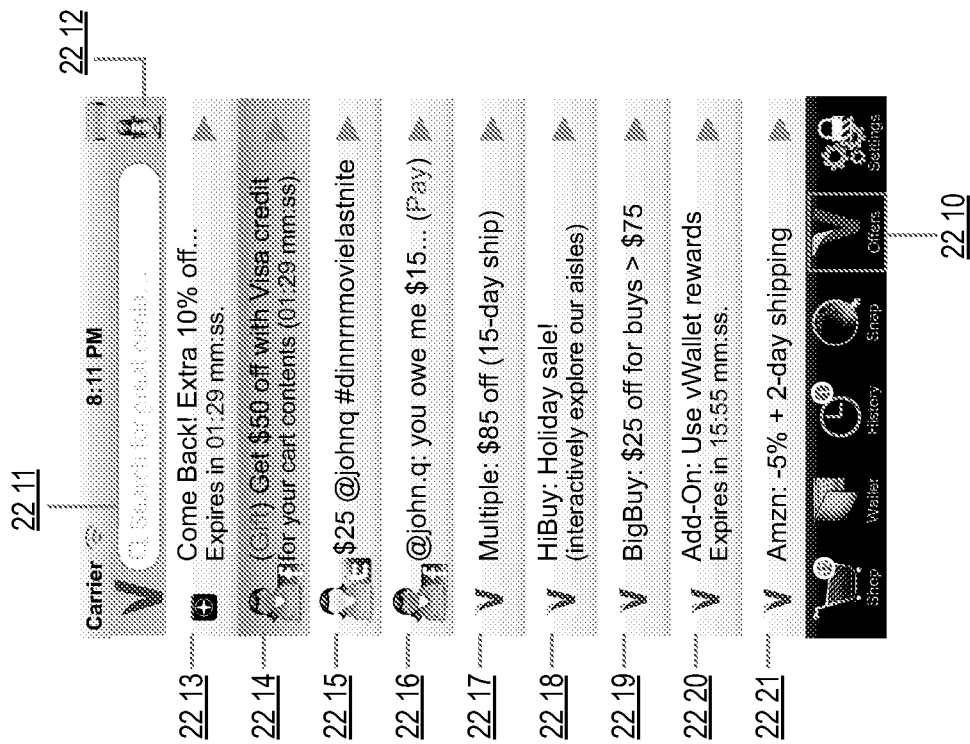


FIGURE 22

Example: Virtual Wallet Mobile App - Offers

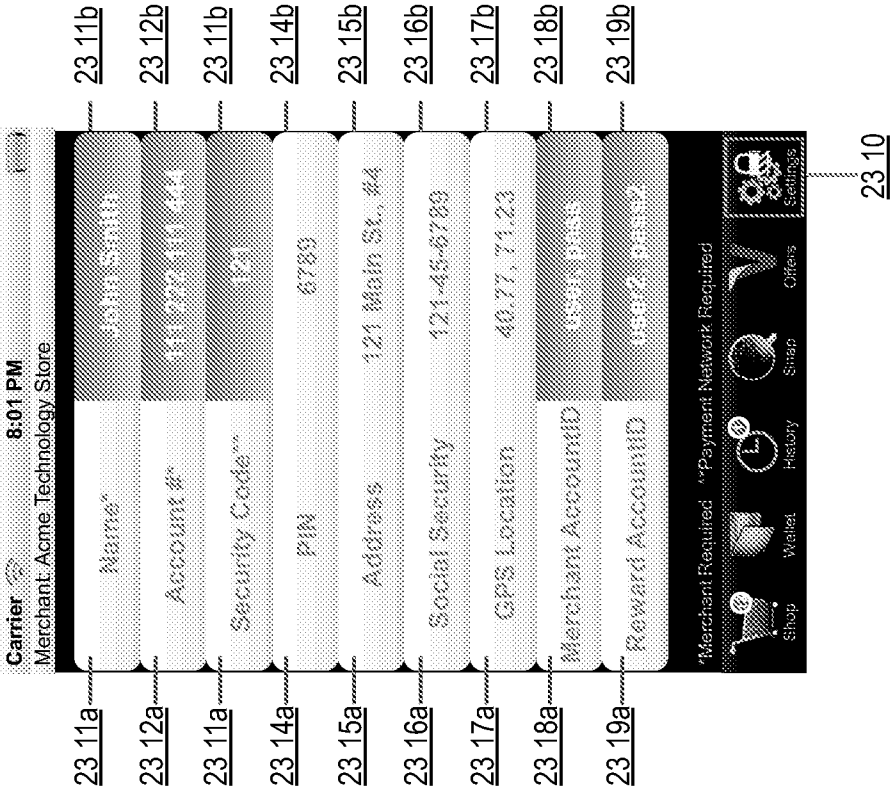


FIGURE 23A

Example: Virtual Wallet Mobile App

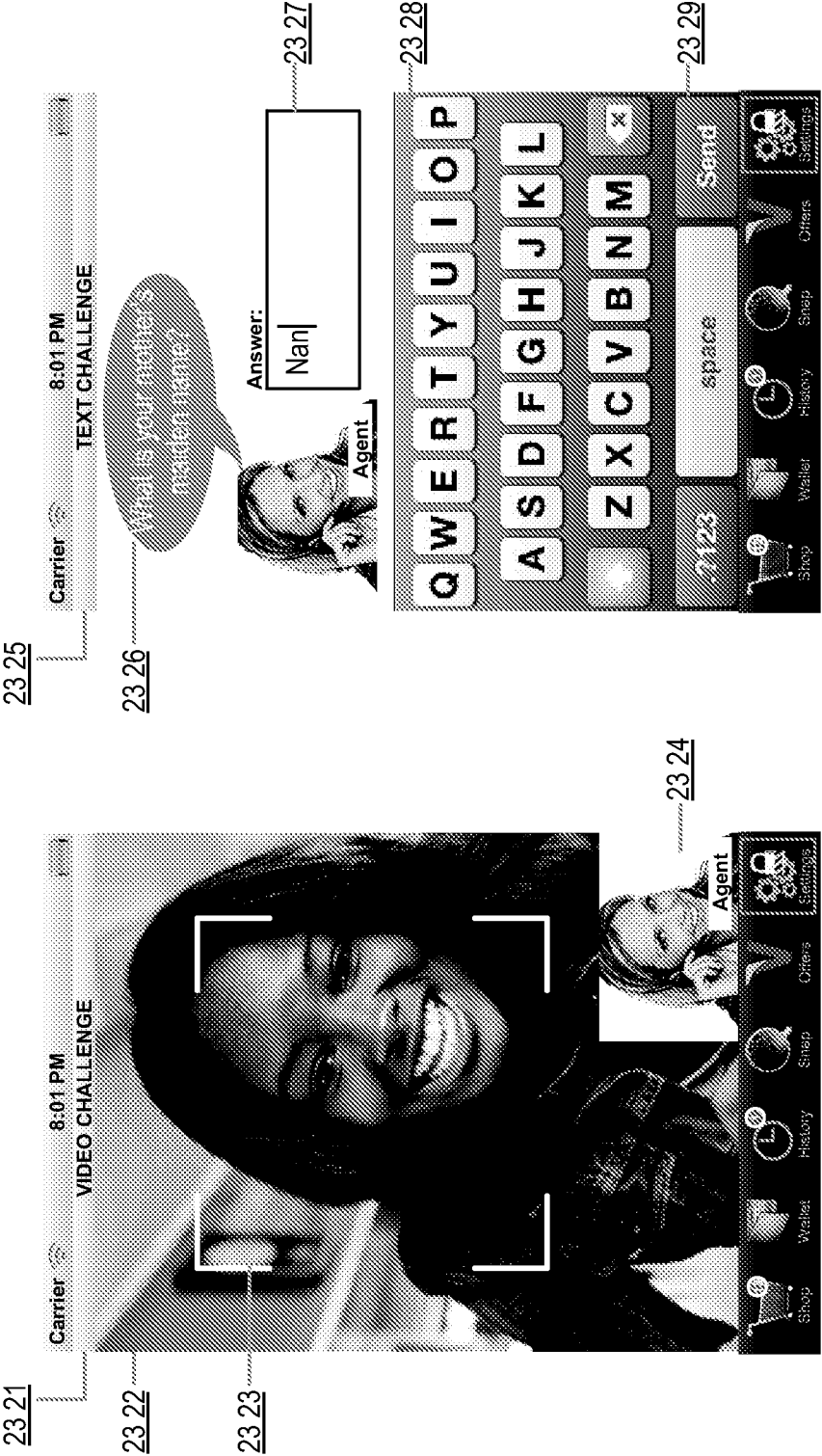
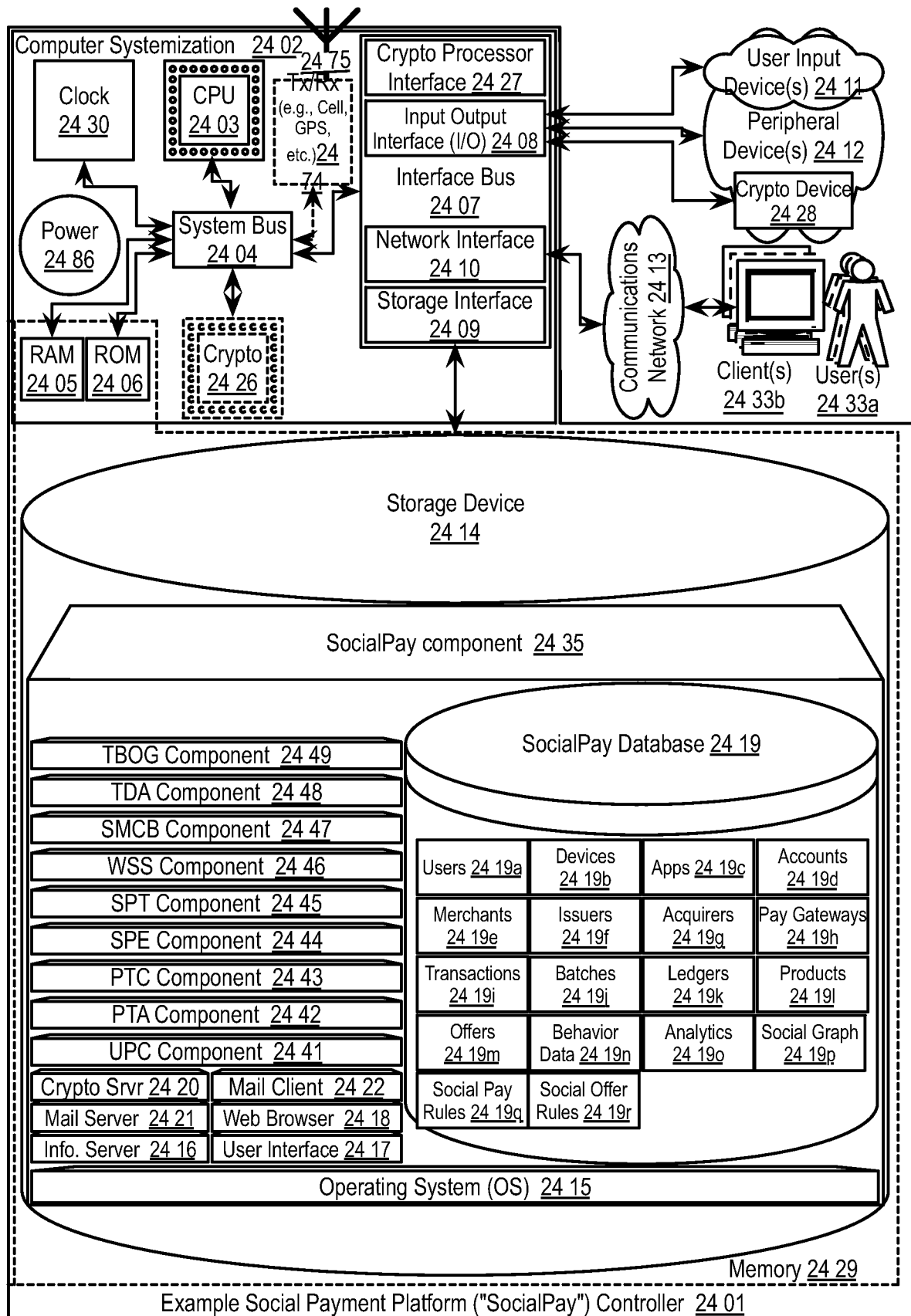


FIGURE 23B

Example: Virtual Wallet Mobile App

FIGURE 24

49/49



INTERNATIONAL SEARCH REPORT

International application No.

PCT/US2011/065305

A. CLASSIFICATION OF SUBJECT MATTER

IPC(8) - G06Q 20/00 (2012.01)

USPC - 705/39

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC(8) - G06Q 20/00, 30/00, 40/00 (2012.01)

USPC - 705/35, 39-42, 44, 319; 725/5

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

MicroPatent, Google Patents, Google Scholar

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2009/0182664 A1 (TROMBLEY) 16 July 2009 (16.07.2009) entire document	1-2, 4-9, 11-16, 18-26, 30, 35-42, 46, 51-53
--		
Y		3, 10, 17, 27-29, 31-34, 43-45, 47-50
Y	US 2009/0288012 A1 (HERTEL et al) 19 November 2009 (19.11.2009) entire document	3, 10, 17, 28-29, 31-34, 44-45, 47-50
Y	US 2010/0023386 A1 (AVISAR et al) 28 January 2010 (28.01.2010) entire document	27, 43
A	US 2010/0125803 A1 (JOHNSON) 20 May 2010 (20.05.2010) entire document	1-53
A	US 2009/0241159 A1 (CAMPAGNA et al) 24 September 2009 (24.09.2009) entire document	1-53
A	US 2008/0114737 A1 (NEELY et al) 15 May 2008 (15.05.2008) entire document	1-53
A	US 2007/0214250 A1 (AHMED et al) 13 September 2007 (13.09.2007) entire document	1-53
A	US 2010/0121707 A1 (GOELDI) 13 May 2010 (13.05.2010) entire document	1-53

☐ Further documents are listed in the continuation of Box C.


* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search

09 April 2012

Date of mailing of the international search report

16 APR 2012

Name and mailing address of the ISA/US

Mail Stop PCT, Attn: ISA/US, Commissioner for Patents

P.O. Box 1450, Alexandria, Virginia 22313-1450

Facsimile No. 571-273-3201

Authorized officer:

Blaine R. Copenheaver

PCT Helpdesk: 571-272-4300

PCT OSP: 571-272-7774