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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
10 January 2013 (10.01.2013)

WIPO | PCT

(10) International Publication Number
WO 2013/006725 A2

(51) International Patent Classification: Not classified

(21) International Application Number: PCT/US2012/045601

(22) International Filing Date: 5 July 2012 (05.07.2012)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
61/504,348 5 July 2011 (05.07.2011) US

(71) Applicant (*for all designated States except US)*: **VISA INTERNATIONAL SERVICE ASSOCIATION** [US/US]; P.O. Box 8999, MS M1-11F, San Francisco, CA 94128-8999 (US).

(72) Inventor; and
(75) Inventor/Applicant (*for US only)*: **KALGI, Avinash** [US/US]; 12813 NE 106th PL, Kirkland, WA 98033 (US).

(74) Agent: **HANCHUK, Walter**; Chadbourne & Parke LLP, 30 Rockefeller Plaza, New York, NY 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, 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:

— *without international search report and to be republished upon receipt of that report (Rule 48.2(g))*

(54) Title: ELECTRONIC WALLET CHECKOUT PLATFORM APPARATUSES, METHODS AND SYSTEMS

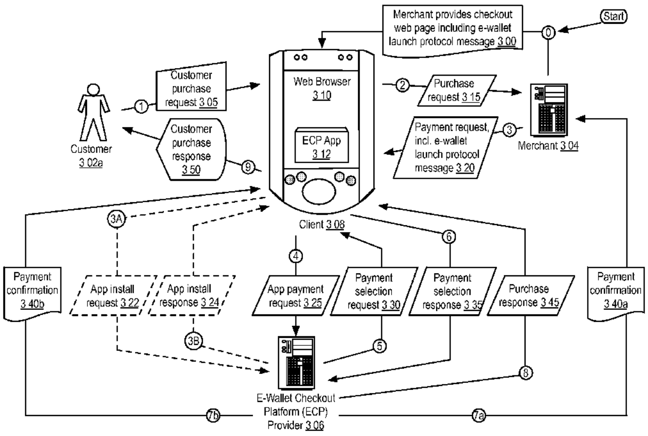

Start
0
Merchant provides checkout web page including e-wallet launch protocol message 3 00
Web Browser 3 10
ECP App 3 12
Customer 3 02a
1
Customer purchase request 3 05
Customer purchase response 3 50
9
2
Purchase request 3 15
Merchant 3 04
3
Payment request, incl. e-wallet launch protocol message 3 20
Client 3 08
3A
4
6
Payment confirmation 3 40b
App install request 3 22
App install response 3 24
App payment request 3 25
Payment selection request 3 30
Payment selection response 3 35
Purchase response 3 45
Payment confirmation 3 40a
3B
5
8
E-Wallet Checkout Platform (ECP) Provider 3 06
7b
7a

FIGURE 3

Example: E-Wallet Checkout Data Flow

(57) **Abstract**: The ELECTRONIC WALLET CHECKOUT PLATFORM APPARATUSES, METHODS AND SYSTEMS (EWCP) transform customer purchase requests triggering electronic wallet applications via EWCP components into electronic purchase confirmation and receipts. In one implementation, the EWCP receives a merchant payment request, and determines a payment protocol handler associated with the merchant payment request. The EWCP instantiates a wallet application via the payment protocol handler. The EWCP obtains a payment method selection via the wallet application, wherein the selected payment method is one of a credit card, a debit card, a gift card selected from an electronic wallet, and sends a transaction execution request for a transaction associated with the merchant payment request. Also, the EWCP receives a purchase response to the transaction execution request, and outputs purchase response information derived from the received purchase response.

ELECTRONIC WALLET CHECKOUT PLATFORM APPARATUSES, METHODS AND SYSTEMS

[0001] This application for letters patent discloses and describes various novel innovations and inventive aspects of ELECTRONIC WALLET CHECKOUT PLATFORM technology (hereinafter “disclosure”) and contains material that is subject to copyright, mask work, and/or other intellectual property protection. The respective owners of such intellectual property have no objection to the facsimile reproduction of the disclosure by anyone as it appears in published Patent Office file/records, but otherwise reserve all rights.

PRIORITY CLAIM

[0002] This application claims priority under 35 USC § 119 to United States provisional patent application serial no. 61/504,348 filed July 5, 2011, entitled “ELECTRONIC WALLET CHECKOUT PLATFORM APPARATUSES, METHODS AND SYSTEMS,” attorney docket no. 11US01|20270-177PV. The entire contents of the aforementioned application are expressly incorporated by reference herein.

FIELD

[0003] The present innovations generally address apparatuses, methods, and systems for electronic purchase transactions, and more particularly, include ELECTRONIC WALLET CHECKOUT PLATFORM APPARATUSES, METHODS AND SYSTEMS (“EWCP”).

BACKGROUND

[0004] Credit cards, debit cards, and gift cards have supplemented cash in facilitating purchase transactions. Such forms of payment are used extensively in online purchase transactions, where direct exchange of cash is usually not practical.

BRIEF DESCRIPTION OF THE DRAWINGS

[0005] The accompanying appendices, drawings, figures, images, etc. illustrate various example, non-limiting, inventive aspects, embodiments, and features ("e.g.," or "example(s)") in accordance with the present disclosure:

[0006] FIGURE 1 shows a block diagram of an exemplary e-wallet checkout platform usage scenario in one embodiment of the EWCP;

[0007] FIGURE 2 shows a screen shot diagram illustrating a mobile EWCP application in one embodiment of the EWCP;

[0008] FIGURE 3 shows a data flow diagram for e-wallet checkout in one embodiment of the EWCP;

[0009] FIGURE 4 shows a logic flow diagram of an E-Wallet Checkout (EWC) component in one embodiment of the EWCP;

[0010] FIGURE 5 shows a logic flow diagram of an E-Wallet Checkout Payment Acquisition (EWCPA) component in one embodiment of the EWCP;

[0011] FIGURE 6 shows a logic flow diagram of an E-Wallet Checkout Purchase Transaction (EWCPT) component in one embodiment of the EWCP;

[0012] FIGURE 7 shows a datagraph diagram illustrating example aspects of transforming a user checkout request input via a User Purchase Checkout ("UPC") component into a checkout data display output;

[0013] FIGURE 8 shows a logic flow diagram illustrating example aspects of transforming a user checkout request input via a User Purchase Checkout ("UPC") component into a checkout data display;

[0014] FIGURES 9A-B show datagraph diagrams illustrating example aspects of transforming a user virtual wallet access input via a Purchase Transaction Authorization ("PTA") component into a purchase transaction receipt notification;

[0015] FIGURES 10A-B show logic flow diagrams illustrating example aspects of transforming a user virtual wallet access input via a Purchase Transaction Authorization ("PTA") component into a purchase transaction receipt notification;

[0016] FIGURES 11A-B show datagraph diagrams illustrating example aspects of transforming a merchant transaction batch data query via a Purchase Transaction Clearance ("PTC") component into an updated payment ledger record;

[0017] FIGURES 12A-B show logic flow diagrams illustrating example aspects of transforming a merchant transaction batch data query via a Purchase Transaction Clearance ("PTC") component into an updated payment ledger record;

[0018] FIGURE 13 shows a user interface diagram illustrating an overview of example features of virtual wallet applications in some embodiments of the EWCP;

[0019] FIGURES 14A-K show user interface and logic flow diagrams illustrating example features of virtual wallet applications in a shopping mode, in some embodiments of the EWCP;

[0020] FIGURES 15A-F show user interface diagrams illustrating example features of virtual wallet applications in a payment mode, in some embodiments of the EWCP;

[0021] FIGURE 16 shows a user interface diagram illustrating example features of virtual wallet applications, in a history mode, in some embodiments of the EWCP;

[0022] FIGURES 17A-E show user interface diagrams illustrating example features of virtual wallet applications in a snap mode, in some embodiments of the EWCP;

[0023] FIGURE 18 shows a user interface diagram illustrating example features of virtual wallet applications, in an offers mode, in some embodiments of the EWCP;

[0024] FIGURES 19A-B show user interface diagrams illustrating example features of virtual wallet applications, in a security and privacy mode, in some embodiments of the EWCP; and

[0025] FIGURE 20 shows a block diagram illustrating example aspects of a EWCP controller.

[0026] The leading number of each reference number within the drawings indicates the figure in which that reference number is introduced and/or detailed. As such, a detailed discussion of reference number 101 would be found and/or introduced in FIGURE 1. Reference number 201 is introduced in FIGURE 2, etc.

DETAILED DESCRIPTION

ELECTRONIC WALLET CHECKOUT PLATFORM (EWCP)

[0027] The ELECTRONIC WALLET CHECKOUT PLATFORM APPARATUSES, METHODS AND SYSTEMS (hereinafter "EWCP") transform customer purchase requests triggering electronic wallet applications, via EWCP components, into electronic purchase confirmation and receipts. A consumer who wants to purchase a product and/or a service from a merchant may go to the merchant's website to engage in a purchase transaction. For example, this may involve using the merchant's website to provide personal information, such as credit card billing address and credit card information, to the merchant to facilitate the purchase transaction. Such online shopping experience may be fraught with security risks and usability problems. The customer's data may be exposed or misused by the merchant during the checkout and/or payment flow. If the merchant stores the customer's personal information, this information may be also vulnerable to security breaches that may happen long after the customer engaged in a purchase transaction. In addition, different checkout and payment implementations provide inconsistent user experience across merchant websites, which may result in customer confusion and unpleasant shopping experience. The EWCP externalizes the checkout and/or payment flow from the web based, merchant driven model. Using the EWCP, the customer may engage in a purchase transaction via a secure platform that may provide a consistent user experience across merchant websites.

[0028] FIGURE 1 shows a block diagram of an exemplary e-wallet checkout platform usage scenario in one embodiment of the EWCP. In FIGURE 1, a customer 102a may wish to purchase a product and/or service 110 from a merchant 104 via a web browser (e.g., at the merchant's website). For example, the customer may be shopping online for an item at the merchant's website, via consumer device 102b. In another example, the customer may be shopping at the merchant's retail location and may purchase an item using a mobile device to scan the item's product identifier, as described in more detail in U.S. provisional patent application serial no. 61/467,890 filed March 25, 2011, entitled "In-Person One-Tap Purchasing Apparatuses, Methods and Systems," attorney docket no. P-42011PRV|20270-126PV, and in U.S. provisional patent application serial no. 61/467,969 filed March 25, 2011, entitled "In-Person One-Tap Purchasing Apparatuses, Methods and Systems," attorney docket no. P-42162PRV|20270-126PV2 (the entire contents of the aforementioned applications are herein expressly incorporated by reference).

[0029] The merchant may provide an indication 115 that the customer may use an e-wallet app (e.g., provided by the EWCP as described with reference to FIGURES 13-19B) to pay for an item. For example, such an indication may take the form of using an EWCP-supported protocol at the checkout webpage. The EWCP may detect that the merchant uses an EWCP-supported protocol (e.g., via a mobile EWCP application described in more detail in FIGURE 2), and may prompt the customer to use an EWCP electronic wallet (hereinafter "E-Wallet" or "wallet application") to facilitate payment. Using the E-Wallet may facilitate payment for the item without revealing the customer's personal information to the merchant. The customer may be asked to provide authentication and/or authorization information to use the E-Wallet. The customer may also provide a selection of a payment method (e.g., a specific credit card) 120. In some embodiments, the customer device providing the E-Wallet may be a separate device from the customer device using which the customer initiated shopping with the merchant.

[0030] An EWCP provider 106 may verify authentication and/or authorization and/or payment information. The EWCP may use information regarding the merchant selling the item and/or regarding the item (e.g., obtained via parameter values in a universal resource identifier (URI) link used to access the E-Wallet), to facilitate completion of the purchase transaction. For example, the EWCP may determine the price and/or quantity of each item being purchased, calculate the total amount due from the customer, collect the total amount due and provide appropriate payment (e.g., total amount due minus fees) to the merchant. The EWCP provider may also provide a confirmation to the merchant that payment information was obtained 125 (e.g., via an API call, via an email, and/or the like). In one embodiment, the EWCP provider may provide a receipt (e.g., a confirmation page, a confirmation email, and/or the like) to the customer. For example, this may be a receipt customized for the merchant (e.g., containing merchant branding), a generic default receipt, and/or the like. In another embodiment, the merchant may provide a receipt to the customer 130. For example, the merchant may generate the receipt based on the confirmation information received from the EWCP provider.

[0031] In some implementations, the wallet application may allow the user to shop within the inventories of merchants participating in the EWCP. For example, the inventories of the merchants may be provided within the wallet application for the user to make purchases. In some implementations, the wallet application may provide a virtual storefront for the user within the graphical user interface of the virtual wallet application. Thus, the user may be virtually injected into a store of the merchant participating in the EWCP's wallet application.

[0032] In some implementations, the wallet application may utilize the location coordinates of the user device (e.g., via GPS, IP address, cellular tower triangulation, etc.) to identify merchants that are in the vicinity of the user's current location. In some implementations, the wallet application may utilize such information to provide information to the user on the inventories of the merchants in the locality, and or may inject the merchant store virtually into the user's wallet application.

[0033] In some implementations, the wallet application may provide a shopping assistant. For example, a user may walk into a physical store of a merchant. The user may require assistance in the shopping experience. In some implementations, the wallet application may allow the user to turn on the shop assistant, and a store executive in the merchant store may be able to assist the user via another device. In some embodiments, a user may enter into a store (e.g., a physical brick-and-mortar store, virtual online store via a computing device, etc.) to engage in a shopping experience. The user may have a user device. The user device may have executing thereon a wallet application, including features such as those as described herein. Upon entering the store, the user device may communicate with a store management server. For example, the user device may communicate geographical location coordinates, user login information and/or like check-in information to check in automatically into the store. In some embodiments, the EWCP may inject the user into a virtual wallet store upon check in. For example, the wallet app executing on the user device may provide features as described below to augment the user's in-store shopping experience. In some embodiments, the store management server may inform a customer service representative ("CSR") of the user's arrival into the store. For example, the CSR may have a CSR device, and an app ("CSR app") may be executing thereon. For example, the app may include features such as described below in the discussion herein. The CSR app may inform the CSR of the user's entry, including providing information about the user's profile, such as the user's identity, user's prior and recent purchases, the user's spending patterns at the current and/or other merchants, and/or the like. In some embodiments, the store management server may have access to the user's prior purchasing behavior, the user's real-time in-store behavior (e.g., which items' barcode did the user scan using the user device, how many times did the user scan the barcodes, did the user engage in comparison shopping by scanning barcodes of similar types of items, and/or the like), the user's spending patterns (e.g., resolved across time, merchants, stores, geographical locations, etc.), and/or like user profile information. The store management system may utilize this information to provide offers/coupons, recommendations and/or the like to the CSR and/or the user, via the CSR device and/or user device, respectively. In some embodiments, the CSR may assist the user in the shopping experience. For example, the CSR may convey offers, coupons, recommendations, price comparisons, and/or the like, and may perform actions on behalf of the user, such as adding/removing items to the user's physical/virtual cart, applying/removing coupons to the user's purchases, searching for offers, recommendations, providing store maps, or store 3D immersion views, and/or the like. In some embodiments, when the user is ready to checkout, the EWCP may provide a checkout notification to the user's device and/or CSR device. The user may checkout using the user's wallet application executing on the user device, or may utilize a communication mechanism (e.g., near field communication, card swipe, QR code scan, etc.) to provide payment information to the CSR device. Using the payment information, the EWCP may initiate the purchase transaction(s) for the user, and provide an electronic receipt to the user device and/or CSR device. Using the electronic receipt, the user may exit the store with proof of purchase payment.

[0034] FIGURE 2 shows a screen shot diagram illustrating a mobile EWCP application in one embodiment of the EWCP. As illustrated in screen 201 of FIGURE 2, a customer may visit a merchant's website using a mobile device (e.g., a cell phone, a PDA, a tablet, and/or the like). The customer may wish to purchase two 204 Micro SD cards 202 at a price 206 of $3.45 each. The customer may click the "Buy" button 208a to purchase these items. The mobile EWCP application may detect that the merchant's website uses an EWCP-supported protocol (see FIGURE 5 for additional details regarding detecting merchant support for an EWCP-supported protocol), and may prompt the customer to use an E-Wallet to facilitate payment. In one embodiment, the E-Wallet may be a part of the mobile EWCP application. In another embodiment, the E-Wallet may be a different mobile EWCP application. In yet another embodiment, the E-Wallet may be a website associated with the EWCP provider, and the customer may be redirected to this website.

[0035] In one embodiment, the "Buy" button 208a may facilitate a single click purchase. The "More" button 208b may be used to configure the default payment method to be used upon clicking the "Buy" button 208a (e.g., the customer may select the default payment method). For example, clicking the "More" button 208b may bring up an E-Wallet popup 208c (e.g., a Web 2.0 popup, an HTML popup , a Javascript popup, a Java applet popup, and/or the like) that facilitates selection of the default payment method. In one embodiment, the selections may be set as default for future preferences/transactions and may be synched to other applications and/or devices as default settings for E-Wallet transactions (e.g., as in FIGURES 13-19B). See FIGURES 13-19B for additional details regarding exemplary E-Wallet popup features and/or embodiments. In one embodiment, the E-Wallet may challenge the user with a password login/dialog box prior to displaying the information shown in the E-Wallet popup 208c and/or prior to allowing a single click purchase. In one embodiment, subsequent selections of the "Buy" or "Authorize" button(s) may use these default settings. Upon clicking the "Buy" button 208a, the default payment method (e.g., a specific credit card) may be used instead of prompting the customer to select a wallet and/or a payment method to be used.

[0036] In another embodiment, the "Buy" button 208a may trigger the launch of an E-Wallet application and the application may be instantiated. As illustrated in screen 211, upon authentication and/or authorization to use the E-Wallet, the customer may be prompted to select an available wallet. In one embodiment, the customer may select a wallet stored on the mobile device 212 (e.g., stored on the mobile device's memory card, such as on a secure micro SD card). In another embodiment, the customer may select an online wallet 214 (e.g., stored in a database of the EWCP provider and which may be accessed via a Java applet, HTML5 application, Javascript application, and/or the like).

[0037] As illustrated in screen 221, the customer may be presented with a choice of payment methods 222a-222d (e.g., credit cards, debit cards, gift cards, and/or the like) available in the wallet selected by the customer. In one embodiment, the available payment methods in the wallet may be presented to the customer. In another embodiment, the merchant may specify criteria (e.g., via parameter values in a URI link used to access the E-Wallet, via a configuration file stored on the EWCP provider's server, and/or the like) for payment methods that are acceptable (e.g., credit cards and the merchant's gift cards) and only the payment methods that satisfy the specified criteria may be presented to the customer. In some embodiments, merchants may also specify (e.g., via parameter values in a URI link used to access the E-Wallet, via a configuration file stored on the EWCP provider's server, and/or the like) rewards, discounts, bonus items, and/or the like associated with using various payment methods. For example, a merchant may specify that using the merchant's gift card gives an additional 2% discount off the price of the order. The customer may also add additional payment methods if desired. For example, the customer may add a new credit card and select it as the payment method. In some embodiments, a merchant may specify (e.g., via parameter values in a URI link used to access the E-Wallet, via a configuration file stored on the EWCP provider's server, and/or the like) that a customer who does not have a preferred payment method (e.g., the merchant's credit card) should be prompted to sign up for and/or add the preferred payment method to the wallet. The merchant may also specify rewards, discounts, bonus items, and/or the like associated with signing up for and/or adding the preferred payment method to the wallet. Upon selecting a payment method (e.g., 222a), the customer may use the "Complete the purchase..." button 224 to submit payment information.

[0038] The EWCP provider may analyze payment information provided by the customer and/or information regarding the merchant and/or the item (e.g., obtained via parameter values in a URI link used to access the E-Wallet, via a configuration file stored on the EWCP provider's server, and/or the like) to facilitate the purchase transaction. For example, the EWCP provider may verify payment information, determine the amount to charge the customer, verify that the merchant is authorized to receive payment, and/or the like. As illustrated in screen 231, the EWCP provider may provide a receipt 232 to the customer confirming the purchase transaction. The receipt may include item name 234, quantity purchased 236, item price 238, transaction authorization and/or verification code 240, payment information 242, and/or the like. The EWCP provider may also provide a payment confirmation to the merchant to confirm that the payment has been made.

[0039] FIGURE 3 shows a data flow diagram for e-wallet checkout in one embodiment of the EWCP. In FIGURE 3, a merchant may have initially provided to a web browser executing on client 308, for a customer 302a, a checkout page (e.g., a HTML web page), including an e-wallet launch protocol message, e.g., 300. The customer 302a may send, using the client 308, a customer purchase request 305 to client 308. For example, the customer may send a request to purchase an item from a merchant 304. In one implementation, the user may use a keyboard, a mouse, a touch screen, and/or the like to input the customer purchase request via a merchant's website. In another implementation, the customer may use a camera of the client (e.g., of a cell phone, a PDA, a tablet, and/or the like) to scan the item's product identifier. The client may provide a purchase request 315 to the merchant 304 indicating to the merchant that the customer wants to purchase the item. For example, the client may use a web browser

310 to submit the purchase request to the merchant's website and the purchase request may be in XML format substantially in the following form:

```
<XML>
   <PurchaseRequest>
         <ItemName>SD Card</ItemName>
         <ItemID>ID123I</ItemID>
         <ItemPrice>$3.45</ItemPrice>
         <ItemQuantity>2</ItemQuantity>
         <UserID>ID123U</UserID>
         <PromotionalCodes>Code123C</PromotionalCodes>
         <RewardCodes>None</RewardCodes>
   </PurchaseRequest>
</XML>
```

[0040] The merchant may respond with a payment request 320. The payment request may prompt the customer to provide payment information to pay for the item. In some embodiments, the payment request 320 may include an e-wallet launch protocol message. For example, the payment request may be a webpage that uses a protocol string (e.g., a string starting with "ewalletcheckout://") to indicate that the merchant uses an EWCP-supported protocol. In one embodiment, the protocol string may be detected by an EWCP application 312 (see FIGURE 5 for additional details regarding detecting merchant support for an EWCP-supported protocol), and the E-Wallet may be activated in response. In another embodiment, the webpage may detect (e.g., via Javascript) whether an EWCP application 312 is installed on the client and use the protocol string if the EWCP application is installed (and use a non EWCP payment method otherwise). In some embodiments, if the EWCP application is not installed on the client, the merchant's webpage may prompt the customer to install the EWCP application on the client. If the customer chooses to install the EWCP application, an application installation request 322 may be sent by the client to obtain the EWCP application from the EWCP provider 306 via an application installation response 324. For example, the merchant may provide a link to a webpage of the EWCP provider 306, and the customer may follow that link to obtain an installation file for the EWCP application. In another example, the merchant may provide a link to an Apple App Store and/or Android Market distributed application that the customer may follow to obtain the EWCP application.

[0041] The client may provide an application payment request 325 to the EWCP provider 306. The application payment request may be used to provide information regarding the merchant, the item, the customer, and/or the like to the EWCP provider to facilitate payment to the merchant. For example, the application payment request may identify the unique ID of the merchant that should receive payment, the unique ID and/or description of the item being purchased, the quantity being purchased, the purchase price, the customer's EWCP login information, client information, and/or the like, and may be in XML format substantially in the following form:

```
<XML>
   <ApplicationPaymentRequest>
         <MerchantID>ID456M</Merchant>
         <ItemName>SD Card</ItemName>
         <ItemID>ID123I</ItemID>
         <ItemPrice>$3.45</ItemPrice>
         <ItemQuantity>2</ItemQuantity>
         <CustomerID>ID789C</CustomerID>
         <CustomerPassword>******</CustomerPassword>
         <PromotionalCodes>Code123C</PromotionalCodes>
         <RewardCodes>None</RewardCodes>
   </ApplicationPaymentRequest>
</XML>
```

[0042] The EWCP provider may provide a payment selection request 330 to the client 308. In alternative embodiments, the merchant 304 may provide the payment selection request 330 to the client 308. The payment selection request may be used to determine the wallet and/or the payment method that the customer may wish to use to pay for the item. For example, the payment selection request may include lists of wallets and/or payment methods (e.g., credit cards, debit cards, gift cards, and/or the like) that the customer may use, and may be in XML format substantially in the following form:

```
<XML>
   <PaymentSelectionRequest>
         <Wallet>
               <WalletID>ID234W</WalletID>
               <PaymentMethods>
                     <PaymentMethod>Credit Card 1</PaymentMethod>
                     <PaymentMethod>Credit Card 2</PaymentMethod>
                     <PaymentMethod>Debit Card 1</PaymentMethod>
               </PaymentMethods>
         </Wallet>
         <Wallet>
               <WalletID>ID345W</WalletID>
               <PaymentMethods>
                     <PaymentMethod>Credit Card 3</PaymentMethod>
                     <PaymentMethod>Credit Card 4</PaymentMethod>
                     <PaymentMethod>Gift Card 1</PaymentMethod>
               </PaymentMethods>
         </Wallet>
   </PaymentSelectionRequest>
</XML>
```

[0043] In another embodiment, the payment selection request may be a store injection package. For example, the store injection package may use payment request 320 to preselect items provided through the store injection package. In a further example, the merchant may have provided the store injection package along with the payment request 320 to client 308, or may have provided it to the e-wallet checkout platform provider 306 for transmission to the client 308 with the payment selection request 330. The customer may select the desired wallet and/or payment method, and the client may provide a payment selection response 335 to the EWCP provider. For example, the payment selection response may include a payment method selection, a payment method security code, and/or the like, and may be in XML format substantially in the following form:

```
<XML>
   <PaymentSelectionResponse>
         <PaymentMethod>Credit Card 1</PaymentMethod>
         <PaymentMethodSecurityCode>432</PaymentMethodSecurityCode>
   </PaymentSelectionResponse>
</XML>
```

[0044] Upon verifying that the customer's payment information is acceptable, the EWCP provider may provide a payment confirmation 340a to the merchant, and/or 340b to the client. For example, the payment confirmation may include a transaction ID, the transaction date and/or time, transaction status, a transaction authorization and/or verification code, details regarding the purchase, and/or the like, and may be in XML format substantially in the following form:

```
<XML>
   <PaymentConfirmation>
         <TransactionID>ID567T</TransactionID>
         <TransactionDate>June 12, 2011</TransactionDate>
         <TransactionTime>9:00:00pm</TransactionTime>
         <TransactionStatus>OK</TransactionStatus>
         <AuthorizationCode>ANET10</AuthorizationCode>
         <ItemName>SD Card</ItemName>
         <ItemID>ID123I</ItemID>
         <ItemPrice>$3.45</ItemPrice>
         <ItemQuantity>2</ItemQuantity>
         <CustomerID>ID789C</CustomerID>
   </PaymentConfirmation>
```

```
</XML>
```

[0045] The EWCP provider may also provide a purchase response 345 to the client. The purchase response may facilitate providing the customer with a receipt (e.g., it may include a transaction authorization code that confirms the transaction and may be included as part of the receipt), may redirect the customer to a webpage (e.g., a merchant provided webpage to which the customer should be redirected upon successful payment), and/or the like, and may be in XML format substantially in the following form:

```
<XML>
   <PurchaseResponse>
         <TransactionAuthorizationCode>ANET10</TransactionAuthorizationCode>
         <RedirectPage>www.merchant.com/redirect_success.html<RedirectPage>
   </PurchaseResponse>
</XML>
```

[0046] The client 308 may output a customer purchase response 350 to the customer 302. The client may output the response using a display, speakers, a printer, and/or the like. For example, the client may display the receipt and/or the redirect webpage to the customer.

[0047] FIGURE 4 shows a logic flow diagram of an E-Wallet Checkout (EWC) component in one embodiment of the EWCP. In FIGURE 4, a user (e.g., a customer) may indicate that the user wants to purchase at item from a merchant at 405. The user may access a merchant site supporting the e-wallet checkout platform provided by the EWCP, e.g., 406. The user may then initiate a checkout procedure at the merchant site, e.g., 407. For example, the user may click a "Buy" button on the merchant's website, take a picture of a barcode identifying the item while shopping in a store, and/or the like. The merchant may request payment for the item from the buyer at 410. For example, the merchant's website may provide a webpage that facilitates payment (e.g., by instructing the user's mobile device to launch an EWCP application upon recognizing an EWCP-supported protocol). In another example, an EWCP application (e.g., used by the user to take a picture of the barcode) may provide a payment page upon recognizing an EWCP-supported barcode.

[0048] In one embodiment, the EWCP may already be supported and trusted by the client. In another embodiment, EWCP support may be added to the client upon the user's authorization. A determination may be made at 415 whether the EWCP is supported by the user's client. For example, the merchant's payment request webpage may execute code that queries the client regarding EWCP support, and the code may be written in JavaScript substantially in the following form:

```
//The result of the IF statement indicates whether the device supports EWCP
if (navigator.mimeTypes && navigator.mimeTypes["application/EWCP"])
```

If the client does not support EWCP, the merchant's website may prompt the user to add EWCP support 420. For example, the merchant's website may present a webpage to the user. In various embodiments, the webpage may be a webpage with a link that the user may use to add EWCP support, a redirect to a webpage of the EWCP provider, a webpage with a Java applet that provides EWCP support, a Web 2.0 and/or Javascript widget loaded/embedded into the currently viewed webpage, and/or the like. In one embodiment, the user may add EWCP support by installing a browser extension, a plug-in, an add-on, an applet, and/or the like. Such an extension may be used to trigger, launch and instantiate an E-Wallet application. In another embodiment, the user may add EWCP support by installing a mobile EWCP application. Thus, regardless of the medium, location, and/or format the user may enjoy full access to the EWCP and/or E-Wallet.

[0049] A determination may be made at 425 whether the user added EWCP support. If the user does not add EWCP support 430, a non-EWCP payment scheme may be used by the merchant if available. If the user has and/or adds EWCP support, the merchant may obtain payment via EWCP at 435. For example, the user may provide payment information to the EWCP provider authorized by the merchant to collect payment. See FIGURE 5 for additional details regarding obtaining payment via EWCP.

[0050] A determination may be made at 440 whether the payment was successful (e.g., whether the user's credit card was charged successfully). For example, such a determination may be based on a payment confirmation from the EWCP provider (e.g., transaction details may be verified and the TransactionStatus field of the PaymentConfirmation data structure may be examined to determine whether the value of the field is "OK"). If the payment was unsuccessful, the purchase may be denied at 445. For example, the user may be directed to a webpage that informs the user that the payment was unsuccessful and/or provides the user an opportunity to correct payment information and/or choose a different payment method. If the payment was successful, the purchase transaction may be completed at 450. For example, the user may be provided with a receipt and/or redirected to a merchant specified webpage. See FIGURE 6 for additional details regarding completing the purchase transaction.

[0051] FIGURE 5 shows a logic flow diagram of an E-Wallet Checkout Payment Acquisition (EWCPA) component in one embodiment of the EWCP. In FIGURE 5, a request to obtain payment via EWCP may be received at 505. For example, an EWCP browser extension may detect that a merchant's webpage is using an EWCP-supported protocol (e.g., by determining existence of a protocol string starting with "ewalletcheckout://", "specialcheckout://", "specialwallet://", and/or the like) to request payment from a user. The appropriate protocol handler may be determined at 510. Such a determination may be made by examining the protocol string. In some embodiments, the protocol handler may parsed from the request. For example, the EWCP may utilize various parsers to parse the protocol handler from the request, including those discussed below in the description with reference to FIGURE 20. The EWCP may look for string matches (e.g., using the Perl m// operator) to identify the protocol handlers from the parsed request. In one embodiment, different protocol handlers may be available for different customer types. For example, "ewalletcheckout://" may be handled by a protocol handler available to consumers, while "specialcheckout://" may be handled by a protocol handler available to selected corporate customers. In another embodiment, different protocol handlers may be available for different device types, operating systems, configurations, and/or the like. For example, "ewalletcheckout://" may be handled by a mobile EWCP application, while "specialcheckout://" may be handled by a desktop EWCP application. In some embodiments, the client may support a plurality of protocol handlers, and the protocol handler that is used may be determined based on a ranking of the protocol handlers. In one implementation, this ranking may be based on a configuration file (e.g., which lists the protocol handlers in order of preference) stored on the client, and the client may choose the appropriate protocol handler. In another implementation, the merchant's website may detect (e.g., via Javascript) that the client supports multiple protocol handlers and provide a protocol string based on a ranking specified by the merchant. If a supported protocol handler is found, and the protocol handler is one that is handled by an EWCP application, the client may utilize the protocol handler to instantiate a EWCP application (e.g., an e-wallet application) and pass any parameters to it. For example, the client may first authenticate the user, e.g., 511. For example, the client may challenge the user in a variety of ways. As non-limiting illustrative example, the user may be required to enter login information, answer a challenge question, submit a voice/biometric signature, provide a digital certificate, enter into a video call for verification purposes, and/or the like. If the user is not verified, e.g., 512, option "No," the client may use an alternative payment scheme if available (see FIGURE 4, 440). If the user is verified, e.g., 512, option "Yes," the client may obtain security credentials (e.g., a hash code, a secure key, etc.), for example, retrieved from the client's memory, to launch the wallet application, and may instantiate the wallet application, e.g., 513.

[0052] In some embodiments, the client may determine whether a store injection package is available to inject into the wallet application, e.g., 514a. For example, the merchant and/or e-wallet checkout platform provider may have provided a store injection package to the client, via which the user may engage in additional shopping actions. The determination of whether a store injection package is available to injection into the wallet application may include a determination of whether the user is authenticated or authorized to receive store injection features via the wallet application. If the client determines that a store injection package is available and should be injected into the wallet application, e.g., 514a, option "Yes," the client may provide the store injection package to the wallet application executing on the client, and the user may engage in shopping actions via the store injected into the wallet application, e.g., 514b (for example, see FIGURES 14H-K). For example, items selected by the user while shopping via the store injection into the wallet application may be added on the items selected by the user during the web-browser-based shopping session with the merchant.

[0053] Purchase information may be obtained via the protocol handler at 515. Such information may include the unique ID of the merchant that should receive payment, the unique ID and/or description of the item being purchased, the quantity being purchased, the purchase price, and/or the like. For example, the purchase information may be obtained by parsing (e.g., using the Perl m// operator) the protocol string that may be substantially in the following form:

```
ewalletcheckout://<globallyUniqueId>/<productId>/<productDetail>

```

The purchase information may be provided to the EWCP provider via the protocol handler to facilitate a purchase transaction. It is to be understaood that numerous languages, forms, implementations, and expressions are contemplated (e.g., JavaScript™, Adobe® Flash, HTML5, downloadable compiled plug-ins, etc.); however, in one non-limiting example implementation, a Java program substantially in the following form may be used to externalize the checkout and/or payment flow:

```

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.webkit.WebView;
import android.webkit.WebViewClient;

/**
 * An example implementation of approach to externalize the checkout or payment
 * flow from the web based, merchant site driven checkout and
 * payment flow.
*/
public class Example extends Activity {

```

```
//this is the payment protocol that we intercept i.e. ewalletcheckout://
//Although, what is shown here is a single protocol, this could be a
//plurality of protocols.
//ex: ewalletcheckout://, specialcheckout://, specialwallet:// etc.
private static final String EWALLET_CHECKOUT = "ewalletcheckout://A?";

/**
 * May be called when the activity is first created.
 * One approach may be as follows:
 * 1. Use the web view (browser) of the OS (android in this
 *    example)
 * 2. Create an intercept mechanism where we get to handle the request for
 *    a resource
 * 3. When we detect that the protocol is one of the ones we can handle,
 *    launch the wallet. In the case shown below, we launch our wallet in
 *    response to ewalletcheckout://
 * 4. In addition to the protocol, the specified URI may contain items of
 *    interest (e.g., regarding a transaction), for example:
 *    a. Globally unique identification of the merchant making the sale.
 *    b. Identification of the product being sold
 *    ex: ewalletcheckout://<globallyUniqueId>/<productId>/<productDetail>
 * 5. When intercepted, the requested URI may be handled to send the
 *    payment request to an entry point that may handle the payment or
 *    switch it out based on the Globally unique ID of the merchant, to
 *    the processing gateway.
 */
@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);

    //create the view where we can display the browser.
    WebView wv = new WebView(this);
    wv.getSettings().setJavaScriptEnabled(true);

    //create the OS specific browser.
    wv.setWebViewClient(

            //create the intercept mechanism.
            new WebViewClient(){

                //this may be called when we request a URI.
                @Override
```

```
                    public void onLoadResource(WebView view, String url) {
                        if(url.startsWith(EWALLET_CHECKOUT))
                        {
                            //stop the browser from continuing with the request
                            view.stopLoading();

                            //start handling the URI
                            String modUrl = url.replace(EWALLET_CHECKOUT, "");

                            //Launch the handler. In this case this is the
                            //wallet. Although the decision may be made here
                            //to launch the specific handler in one
                            //implementation, this may be abstracted
                            //out further in other
                            //implementations to make the handler
                            //registration and invocation more dynamic
                            //and update friendly and to handle more protocols.
                            Intent intent = new Intent(
                                    Example.this,Payment.class);

                            //pass the URI to the handler.
                            intent.putExtra("mydata",modUrl);

                            //start the handler.
                            startActivity(intent);

                            //In some implementations, we may block here so
                            //that we can deal with the return
                            //from the handler. This may be a receipt
                            //(text, bar code, and/or the like)
                            //that may be displayed and/or analyzed.
                        }
                    }
        });

        //set the view for display.
        setContentView(wv);

        //first invocation to load the site.
        wv.loadData("<a href='http://.../android/first.html?"+
                System.currentTimeMillis()
                +"'>Use Shop Keeper.</a>","text/html","utf-8");
        }
    }
```

[0054] The user's authentication information (e.g., login information for the EWCP) may be obtained. For example, the protocol handler may request authentication information to determine whether to allow the user access to the E-Wallet. A determination may be made at 517 regarding whether the authentication is Near Field Communication (NFC) based. For example, if the user taps the client with an NFC capable credit card, the authentication may be NFC based, otherwise, if the user enters a password, the authentication may be non-NFC based. If the authentication is non-NFC based, the user's non-NFC based authentication information may be obtained at 520. For example, a password associated with the E-Wallet entered by the user may be obtained. If the authentication is NFC based, the user's NFC based authentication information may be obtained at 522. For example, if the user tapped the client with an NFC capable credit card and provided a pin associated with that credit card (e.g., a pin that decrypts a certificate associated with the client that decrypts data from a tag associated with the credit card), information transmitted via NFC and the pin may be obtained. In one implementation, a program substantially in the following form may be used to facilitate NFC based authentication:

```
############ START ISSUER ###############
//at the issuer side before providing the tag
Tag newTag;
String cardTrack1 = "<track1>";
String cardTrack2 = "<track2>";
String pin = generateRandomNCharPin();

cardTrack1 = encrypt(pin //key used to encrypt
                    ,cardTrack1);
cardTrack2 = encrypt(pin //key used to encrypt
                 ,cardTrack2);

//form the data to be pushed on to the tag and write it
String finalData = cardTrack1 + MARKER+ cardTrack2;
newTag.write(finalData);

//issuer may then provide the PIN via an electronic PIN retrieval system, via
   mail, and/or
//the like to the user

########### END ISSUER #################


########### START CARD USER ############
//Gets the tag and PIN and uses the wallet/application
//to activate the tag for the card (e.g., by re-encrypting the tag using a
//client certificate encrypted by a PIN.
Tag recdTag;
String tagData = recdTag.read();

//get the (temporary) PIN provided to the user.
String mailedPin = promptUserForPIN();

//split the data based on the marker
Byte[] cardTrack1; //obtained from tagData above
Byte[] cardTrack2; //obtained from tagData above

//decrypt the data using the temporary PIN
cardTrack1 = decrypt(mailedPin, cardTrack1);
cardTrack2 = decrypt(mailedPin, cardTrack2);

//get the PIN for device use.
String pin = promptUserForPIN():

//decrypt the certificate to use.
Certificate cert = getDecryptedCert(pin);

//encrypt the data using the public key.
cardTrack1 = cert.encryptWithPublicKey(
                  cardTrack1);
cardTrack2 = cert.encryptWithPublicKey(
                  cardTrack2);
```

```

//form the data to be pushed on to the
//tag and write it.
Byte [] finalData =
       cardTrack1
       + MARKER
       + cardTrack2;

//push the data on to the tag.
recdTag.write(finalData);
//stick the tag on the card.
################ END CARD USER ##########

```

[0055] A determination may be made at 525 whether the authentication information provided by the user is valid. If the user does not provide valid authentication information, access to the EWCP may be denied at 530. For example, the user may be informed that login information was incorrect, and/or may be provided opportunity to re-enter correct login information. If the user provides valid authentication information, access to the EWCP may be granted. The authentication information may be provided to the EWCP provider via the protocol handler to facilitate the purchase transaction. For example, the login information provided by the user may be sent via SSL to an authentication server of the EWCP provider, which may verify the information.

[0056] A determination may be made at 532 whether a default payment method has been specified by the user. For example, the user may have selected a Visa credit card as the default payment method, for all client devices, for this client, and/or the like. In one implementation, this determination may be made by examining a default payment method field of a User Accounts table and/or Client Accounts table via a SQL statement substantially in the following form:

```
SELECT DefaultPaymentMethod
FROM UserAccounts
WHERE UserAccounts.CustomerID = 'Customer's ID' AND
   ClientAccounts.ClientID = 'Client device ID'
```

In another implementation, this determination may be by examining a default payment method setting stored on the client device (e.g., in a settings file of the E-Wallet). In yet another embodiment, if the user uses an NFC capable credit card to authenticate, that credit card may be used as the default payment method.

[0057] If the default payment method is specified, the EWCP may select the default payment method at 534. Otherwise, a selection of a wallet may be obtained at 535. For example, the user may be prompted to select an available wallet via the E-Wallet. In one embodiment, the user may select a wallet stored on the user's client (e.g., stored on a secure micro SD card of a mobile device). In another embodiment, the user may select an online wallet (e.g., stored in a database of the EWCP provider). A selection of a payment method associated with the selected wallet may be obtained at 540. For example, the payment method may be a credit card, debit card, gift card, and/or the like.

[0058] A determination may be made at 545 whether any rewards, discounts, bonus items, and/or the like are associated with using the selected payment method. For example, a merchant may specify that using the merchant's gift card gives the customer 100 merchant reward points with every order. If there is a promotion associated with the selected payment method, adjusted price (e.g., discounted price), rewards points, and/or the like may be calculated in accordance with promotion rules.

[0059] The selected payment method may be used to obtain payment for the item being purchased at 555. In one embodiment, the user may not have to enter additional information to use the selected payment method. In another embodiment, the user may have to enter a payment method security code (e.g., a three digit security code on the back of a credit card) to use the selected payment method. The EWCP provider may obtain the selected payment method and verify that the user may pay for the item using the selected payment method. For example, the EWCP provider may verify the payment security code, that the amount does not exceed the user's credit card limit, that the transaction is not suspicious, and/or the like.

[0060] FIGURE 6 shows a logic flow diagram of an E-Wallet Checkout Purchase Transaction (EWCPT) component in one embodiment of the EWCP. In FIGURE 6, a request to complete a purchase transaction may be received at 605. For example, such a request may be received upon verifying that a user successfully paid for an item (e.g., via a callback function). The EWCP provider may provide the merchant with a payment confirmation at 610. The payment confirmation may indicate that the user successfully paid for the item and may include a transaction ID, the transaction date and/or time, transaction status, a transaction authorization and/or verification code, details regarding the purchase, and/or the like. In one embodiment, the payment confirmation may be provided to the merchant upon verifying the payment method. For example, upon verifying that a user successfully paid for an item, the EWCP provider may send an email to the merchant that includes the payment confirmation. In another example, the EWCP provider may access the merchant's webpage with parameters that correspond to the data in the payment confirmation (e.g., http://www.merchant.com/payment_confirm.html?TransactionID=ID567T&AuthorizationCode=ANET10). In another embodiment, a plurality of payment confirmations may be provided to the merchant at specified times (e.g., daily, weekly, monthly, and/or the like). For example, the EWCP provider may store (e.g., in the Transactions table 2019f) payment confirmations for the merchant during a week via a SQL statement substantially in the following form:

```
INSERT INTO ECP_Transactions (TransactionID, AuthorizationCode, Date)
VALUES ('ID567T', 'ANET10', 'Date of the transaction')
```

The EWCP provider may retrieve this data from the database at the end of the week via a SQL statement substantially in the following form:

```
SELECT TransactionID, AuthorizationCode
FROM ECP_Transactions
WHERE Date='Dates for the week'
```

and provide the retrieved data to the merchant by sending an email, by accessing the merchant's webpage, and/or the like. In some embodiments, a history of purchase transactions may be stored (e.g., in the Transactions table 2019f) and/or made available to various applications (e.g., EWCP applications, third party applications, and/or the like). For example, an application may access the transaction history and analyze data (e.g., determine the amount spent per month and/or the types of products purchased by the user) for a variety of purposes (e.g., to determine which protocol handlers the user may be allowed to use).

[0061] A determination may be made at 615 whether a receipt should be generated. In one embodiment, the EWCP provider may generate the receipt and provide the receipt to the client (e.g., an image may be generated and sent to the client). In another embodiment, the client may obtain applicable information (e.g., transaction authorization code) and generate the receipt via the EWCP application. If the receipt is generated, the receipt may be provided to the user (e.g., via the EWCP application) at

620. A determination may be made at 622 whether to continue the user experience in-wallet (e.g., whether to continue the user's shopping experience in a store injected into the wallet), or to redirect the user back to an online website of the merchant that the user was shopping at via the user's web browser, e.g., 622. For example, the client may prompt the user to provide a one-time/default preference selection on whether to continue within the wallet application or redirect to the client's web browser. If the determination is made to continue in-wallet, the client may (re)load/refresh the store injection package via the wallet application, and the user may continue the in-wallet shopping experience, e.g., 623. A determination may be made at 625 whether a redirect page was specified by the merchant (e.g., by checking a "RedirectPage" field associated with the merchant's database record). For example, the merchant may wish to redirect the user to a "Thank you" webpage upon completion of the purchase transaction. If the merchant specifies a redirect page, the EWCP application may redirect the user's browser to the redirect page at 630. Otherwise, the user may be redirected to a redirect page specified by the EWCP provider 635 (e.g., a generic "Purchase Complete" webpage).

[0062] FIGURE 7 shows a datagraph diagram illustrating example aspects of transforming a user checkout request input via a User Purchase Checkout ("UPC") component into a checkout data display. In some embodiments, a user, e.g., 701a, may desire to purchase a product, service, offering, and/or the like ("product"), from a merchant via a merchant online site or in the merchant's store. The user may communicate with a merchant/acquirer ("merchant") server, e.g., 703a, via a client such as, but not limited to: a personal computer, mobile device, television, point-of-sale terminal, kiosk, ATM, and/or the like (e.g., 702). For example, the user may provide user input, e.g., checkout input 711, into the client indicating the user's desire to purchase the product. In various embodiments, the user input may include, but not be limited to: a single tap (e.g., a one-tap mobile app purchasing embodiment) of a touchscreen interface, keyboard entry, card swipe, activating a RFID/NFC equipped hardware device (e.g., electronic card having multiple accounts, smartphone, tablet, etc.) within the user device, mouse clicks, depressing buttons on a joystick/game console, voice commands, single/multi-touch gestures on a touch-sensitive interface, touching user interface elements on a touch-sensitive display, and/or the like. As an example, a user in a merchant store may scan a product barcode of the product via a barcode scanner at a point-of-sale terminal. As another example, the user may select a product from a webpage catalog on the merchant's website, and add the product to a virtual shopping cart on the merchant's website. The user may then indicate the user's desire to checkout the items in the (virtual) shopping cart. For example, the user may activate a user interface element provided by the client to indicate the user's desire to complete the user purchase checkout. The client may generate a checkout request, e.g., 712, and provide the checkout request, e.g., 713, to the merchant server. For example, the client may provide a (Secure) Hypertext Transfer Protocol ("HTTP(S)") POST message including the product details for the merchant server in the form of data formatted according to the eXtensible Markup Language ("XML"). An example listing of a checkout request 712, substantially in the form of a HTTP(S) POST message including XML-formatted data, is provided below:

```
POST /checkoutrequest.php HTTP/1.1
Host: www.merchant.com
Content-Type: Application/XML
Content-Length: 667
```

```
<?XML version = “1.0” encoding = “UTF-8”?>
<checkout_request>
   <session_ID>4NFU4RG94</session_ID>
<!--optional parameters-->
   <timestamp>2011-02-22 15:22:41</timestamp>
   <user_ID>john.q.public@gmail.com</user_ID>
   <device_fingerprint>
         <device_IP>192.168.23.126</device_IP>
         <device_MAC>0123.4567.89ab</device_MAC>
         <device_serial>312456768798765432</device_serial>
         <device_ECID>00000AEBCDF12345</device_ECID>
         <device_identifier>jqp_air</device_identifier>
         <device_UDID>21343e34-14f4-8jn4-7yfe-124578632134</device_UDID>
         <device_browser>firefox 2.2</device_browser>
         <device_type>smartphone</device_type>
         <device_model>HTC Hero</device_model>
         <OS>Android 2.2</OS>
         <wallet_app_installed_flag>true</wallet_app_installed_flag>
   </device_fingerprint>
</checkout_request>
```

[0063] In some embodiments, the merchant server may obtain the checkout request from the client, and extract the checkout detail (e.g., XML data) from the checkout request. For example, the merchant server may utilize a parser such as the example parsers described below in the discussion with reference to FIGURE 20. Based on parsing the checkout request 712, the merchant server may extract product data (e.g., product identifiers), as well as available PoS client data, from the checkout request. In some embodiments, using the product data, the merchant server may query, e.g., 714, a merchant/acquirer (“merchant”) database, e.g., 703b, to obtain product data, e.g., 715, such as product information, product pricing, sales tax, offers, discounts, rewards, and/or other information to process the purchase transaction and/or provide value-added services for the user. For example, the merchant database may be a relational database responsive to Structured Query Language (“SQL”) commands. The merchant server may execute a hypertext preprocessor (“PHP”) script including SQL commands to query a database table (such as FIGURE 20, Products 2019l) for product data. An example product data query 714, substantially in the form of PHP/SQL commands, is provided below:

```
<?PHP
header('Content-Type: text/plain');
mysql_connect(“254.93.179.112”,$DBserver,$password); // access database server
mysql_select_db(“EWCP_DB.SQL”); // select database table to search
//create query
$query = “SELECT product_title product_attributes_list product_price
   tax_info_list related_products_list offers_list discounts_list rewards_list
   merchants_list merchant_availability_list FROM ProductsTable WHERE
   product_ID LIKE '%' $prodID”;
$result = mysql_query($query); // perform the search query
mysql_close(“EWCP_DB.SQL”); // close database access
?>
```

[0064] In some embodiments, in response to obtaining the product data, the merchant server may generate, e.g., 716, checkout data to provide for the PoS client. In some embodiments, such checkout data, e.g., 717, may be embodied, in part, in a HyperText Markup Language (“HTML”) page including data for display, such as product detail, product pricing, total pricing, tax information, shipping information, offers, discounts, rewards, value-added service information, etc., and input fields to provide payment information to process the purchase transaction, such as account holder name, account number, billing address, shipping address, tip amount, etc. For example, the checkout data may be a HTML webpage that includes a protocol string (e.g., a string starting with "ewalletcheckout://") to indicate that the merchant uses an EWCP-supported protocol, and may facilitate triggering the instantiation of an e-wallet checkout application. In some embodiments, the checkout data may be embodied, in part, in a Quick Response (“QR”) code image that the PoS client can display, so that the user may capture the QR code using a user’s device to obtain merchant and/or product data for generating a purchase transaction processing request. In some embodiments, a user alert mechanism may be built into the checkout data. For example, the merchant server may embed a URL specific to the transaction into the checkout data. In some embodiments, the alerts URL may further be embedded into optional level 3 data in card authorization requests, such as those discussed further below with reference to FIGURES 9-10. The URL may point to a webpage, data file, executable script, etc., stored on the merchant's server dedicated to the transaction that is the subject of the card authorization request. For example, the object pointed to by the URL may include details on the purchase transaction, e.g., products being purchased, purchase cost, time expiry, status of order processing, and/or the like. Thus, the merchant server may provide to the payment network the details of the transaction by passing the URL of the webpage to the payment network. In some embodiments, the payment network may provide notifications to the user, such as a payment receipt, transaction authorization confirmation message, shipping notification and/or the like. In such messages, the payment network may provide the URL to the user device. The user may navigate to the URL on the user's device to obtain alerts regarding the user's purchase, as well as other information such as offers, coupons, related products, rewards notifications, and/or the like. An example listing of a checkout data 717, substantially in the form of XML-formatted data, is provided below:

```
<?XML version = “1.0” encoding = “UTF-8”?>
<checkout_data>
   <session_ID>4NFU4RG94</session_ID>
   <!--optional data-->
```

```
<timestamp>2011-02-22 15:22:43</timestamp>
<expiry_lapse>00:00:30</expiry_lapse>
<total_cost>$121.49</total_cost>
<alerts_URL>www.merchant.com/shopcarts.php?sessionID=4NFU4RG94</alerts_URL>
<user_ID>john.q.public@gmail.com</user_ID>
<user_device_fingerprint>
    <device_IP>192.168.23.126</device_IP>
    <device_MAC>0123.4567.89ab</device_MAC>
    <device_serial>312456768798765432</device_serial>
    <device_ECID>00000AEBCDF12345</device_ECID>
    <device_identifier>jqp_air</device_identifier>
    <device_UDID>21343e34-14f4-8jn4-7yfe-124578632134</device_UDID>
    <device_browser>firefox 2.2</device_browser>
    <device_type>smartphone</device_type>
    <device_model>HTC Hero</device_model>
    <OS>Android 2.2</OS>
    <wallet_app_installed_flag>true</wallet_app_installed_flag>
</user_device_fingerprint>
<purchase_detail>
    <cart>
        <product>
            <merchant_params>
                <merchant_id>54TBRELF8</merchant_id>
                <merchant_name>BARNES, Inc.</merchant_name>
                <merchant_auth_key>TMN45GER98</merchant_auth_key>
            </merchant_params>
            <product_type>book</product_type>
            <product_params>
                <product_title>XML for dummies</product_title>
                <ISBN>938-2-14-168710-0</ISBN>
                <edition>2nd ed.</edition>
                <cover>hardbound</cover>
            </product_params>
            <quantity>2</quantity>
            <unit_cost>$14.46</unit_cost>
            <coupon_id>AY34567</coupon_id>
        <social_flag>ON</social_flag>
        <social_message>Look what I bought today!</social_message>
        <social_networks>facebook twitter</social_networks>
        </product>
        <product>
            <merchant_params>
```

```
                <merchant_id>3FBCR4INC</merchant_id>
                <merchant_name>Books, Inc.</merchant_name>
                <merchant_auth_key>1N484MCP</merchant_auth_key>
            </merchant_params>
            <product_type>book</product_type>
            <product_params>
                <product_title>Sophie’s World</product_title>
                <ISBN>955-2-14-112310-0</ISBN>
                <edition>NULL</edition>
                <cover>hardbound</cover>
            </product_params>
            <quantity>1</quantity>
            <unit_cost>$34.78</unit_cost>
            <coupon_id>null</coupon_id>
        <social_flag>OFF</social_flag>
        </product>
    </cart>
    <cart>
        <product>
            <merchant_params>
                <merchant_id>RFH5IB4FT</merchant_id>
                <merchant_name>Amzn, Inc.</merchant_name>
                <merchant_auth_key>44543DSJFG</merchant_auth_key>
            </merchant_params>
            <product_type>book</product_type>
            <product_params>
                <product_title>XML - a primer</product_title>
                <ISBN>938-2-14-1436710-0</ISBN>
                <edition>2nd ed.</edition>
                <cover>hardbound</cover>
            </product_params>
            <quantity>1</quantity>
            <unit_cost>$12.93</unit_cost>
            <coupon_id>AY34567</coupon_id>
        <social_flag>ON</social_flag>
        <social_message>Look what I bought today!</social_message>
        <social_networks>facebook twitter</social_networks>
        </product>
        <product>
            <merchant_params>
                <merchant_id>3FBCR4INC</merchant_id>
                <merchant_name>BestBooks, Inc.</merchant_name>
```

```
                    <merchant_auth_key>1N484MCP</merchant_auth_key>
                </merchant_params>
                <product_type>book</product_type>
                <product_params>
                    <product_title>Sophie’s Choice</product_title>
                    <ISBN>938-2-14-168710-0</ISBN>
                    <edition>1st ed.</edition>
                </product_params>
                <quantity>1</quantity>
                <unit_cost>$44.86</unit_cost>
                <coupon_id>null</coupon_id>
            <social_flag>OFF</social_flag>
            </product>
        </cart>
    </purchase_detail>
<checkout_data>
```

[0065] Upon obtaining the checkout data, e.g., 717, the PoS client may render and display, e.g., 718, the checkout data for the user.

[0066] FIGURE 8 shows a logic flow diagram illustrating example aspects of transforming a user checkout request input via a User Purchase Checkout (“UPC”) component into a checkout data display. In some embodiments, a user may desire to purchase a product, service, offering, and/or the like (“product”), from a merchant via a merchant online site or in the merchant’s store. The user may communicate with a merchant/acquirer (“merchant”) server via a PoS client. For example, the user may provide user input, e.g., 801, into the client indicating the user’s desire to purchase the product. The client may generate a checkout request, e.g., 802, and provide the checkout request to the merchant server. In some embodiments, the merchant server may obtain the checkout request from the client, and extract the checkout detail (e.g., XML data) from the checkout request. For example, the merchant server may utilize a parser such as the example parsers described below in the discussion with reference to

FIGURE 20. Based on parsing the checkout request, the merchant server may extract product data (e.g., product identifiers), as well as available PoS client data, from the checkout request. In some embodiments, using the product data, the merchant server may query, e.g., 803, a merchant/acquirer ("merchant") database to obtain product data, e.g., 804, such as product information, product pricing, sales tax, offers, discounts, rewards, and/or other information to process the purchase transaction and/or provide value-added services for the user. In some embodiments, in response to obtaining the product data, the merchant server may generate, e.g., 805, checkout data to provide, e.g., 806, for the PoS client. For example, the checkout data may be a HTML webpage that includes a protocol string (e.g., a string starting with "ewalletcheckout://") to indicate that the merchant uses an EWCP-supported protocol, and may facilitate triggering the instantiation of an e-wallet checkout application Upon obtaining the checkout data, the PoS client may render and display, e.g., 807, the checkout data for the user.

[0067] FIGURES 9A-B show datagraph diagrams illustrating example aspects of transforming a user virtual wallet access input via a Purchase Transaction Authorization ("PTA") component into a purchase transaction receipt notification. With reference to FIGURE 9A, in some embodiments, a user, e.g., 901a, may wish to utilize a virtual wallet account to purchase a product, service, offering, and/or the like ("product"), from a merchant via a merchant online site or in the merchant's store. The user may utilize a physical card, or a user wallet device, e.g., 901b, to access the user's virtual wallet account. For example, the user wallet device may be a personal/laptop computer, cellular telephone, smartphone, tablet, eBook reader, netbook, gaming console, and/or the like. The user may provide a wallet access input, e.g., 911 into the user wallet device. In various embodiments, the user input may include, but not be limited to: a single tap (e.g., a one-tap mobile app purchasing embodiment) of a touchscreen interface, keyboard entry, card swipe, activating a RFID/NFC equipped hardware device (e.g., electronic card having multiple accounts, smartphone, tablet, etc.) within the user device, mouse clicks, depressing buttons on a joystick/game console, voice commands, single/multi-touch gestures on a touch-sensitive interface, touching user interface elements on a touch-sensitive display, and/or the like. In some embodiments, the user wallet device may authenticate the user based on the user's wallet access input, instantiate a wallet application (see, e.g., FIGURES 13-19B) upon authenticating the user, and provide virtual wallet features for the user via the wallet application. For example, the user wallet device may utilize the components described above in the description with respect to FIGURES 3-6 to provide e-wallet checkout services for the user via the wallet application.

[0068] In some embodiments, upon authenticating the user for access to virtual wallet features, the user wallet device may provide a transaction authorization input, e.g., 914, to a point-of-sale ("PoS") client, e.g., 902. For example, the user wallet device may communicate with the PoS client via Bluetooth, Wi-Fi, cellular communication, one- or two-way near-field communication ("NFC"), and/or the like. In embodiments where the user utilizes a plastic card instead of the user wallet device, the user may swipe the plastic card at the PoS client to transfer information from the plastic card into the PoS client. For example, the PoS client may obtain, as transaction authorization input 914, track 1 data from the user's plastic card (e.g., credit card, debit card, prepaid card, charge card, etc.), such as the example track 1 data provided below:

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

[0069] In embodiments where the user utilizes a user wallet device, the user wallet device may provide payment information to the PoS client, formatted according to a data formatting protocol appropriate to the communication mechanism employed in the communication between the user wallet device and the PoS client. An example listing of transaction authorization input 914, substantially in the form of XML-formatted data, is provided below:

```
<?XML version = “1.0” encoding = “UTF-8”?>
<transaction_authorization_input>
   <payment_data>
          <account>
                 <charge_priority>1</charge_priority>
                 <charge_ratio>40%</charge_ratio>
                 <account_type>debit</account_type>
                 <value_exchange_symbol>USD</value_exchange_symbol>
                 <account_number>123456789012345</account_number>
                 <account_name>John Q. Public</account_name>
                 <bill_add>987 Green St #456, Chicago, IL 94652</bill_add>
                 <ship_add>987 Green St #456, Chicago, IL 94652</ship_add>
                 <CVV_type>dynamic<CVV_type>
                 <CVV>http://www.paynet.com/dcvv.php?sessionID=4NFU4RG94</CVV>
                 <cloak_flag>ON</cloak_flag>
                 <alert_rules>tar1 tar4 tar12</alert_rules>
                 <mode>NFC</mode>
          </account>
          <account>
                 <charge_priority>1</charge_priority>
                 <charge_ratio>60%</charge_ratio>
                 <account_type>rewards</account_type>
                 <value_exchange_symbol>VME</value_exchange_symbol>
                 <account_number>234567890123456</account_number>
                 <account_name>John Q. Public</account_name>
                 <bill_add>987 Green St #456, Chicago, IL 94652</bill_add>
```

```
            <ship_add>987 Green St #456, Chicago, IL 94652</ship_add>
            <CVV_type>static<CVV_type>
            <CVV>173</CVV>
            <cloak_flag>ON</cloak_flag>
            <alert_rules>tar1 tar4 tar12</alert_rules>
            <mode>Bluetooth</mode>
        </account>
        <account>
            <charge_priority>2</charge_priority>
            <charge_ratio>100%</charge_ratio>
            <account_number>345678901234567</account_number>
            <account_type>credit</account_type>
            <value_exchange_symbol>USD</value_exchange_symbol>
            <account_name>John Q. Public</account_name>
            <bill_add>987 Green St #456, Chicago, IL 94652</bill_add>
            <ship_add>987 Green St #456, Chicago, IL 94652</ship_add>
            <CVV_type>static<CVV_type>
            <CVV>173</CVV>
            <cloak_flag>ON</cloak_flag>
            <alert_rules>tar1 tar4 tar12</alert_rules>
            <mode>NFC</mode>
        </account>
    </payment_data>
    <!--optional data-->
    <timestamp>2011-02-22 15:22:43</timestamp>
    <expiry_lapse>00:00:30</expiry_lapse>
    <secure_key>0445329070598623487956543322</secure_key>
    <alerts_track_flag>TRUE</alerts_track_flag>
    <device_fingerprint>
        <device_IP>192.168.23.126</device_IP>
        <device_MAC>0123.4567.89ab</device_MAC>
        <device_serial>312456768798765432</device_serial>
        <device_ECID>00000AEBCDF12345</device_ECID>
        <device_identifier>jqp_air</device_identifier>
        <device_UDID>21343e34-14f4-8jn4-7yfe-124578632134</device_UDID>
        <device_browser>firefox 2.2</device_browser>
        <device_type>smartphone</device_type>
        <device_model>HTC Hero</device_model>
        <OS>Android 2.2</OS>
        <wallet_app_installed_flag>true</wallet_app_installed_flag>
    </device_fingerprint>
</transaction_authorization_input>
```

[0070] In some embodiments, the PoS client may generate a card authorization request, e.g., 915, using the obtained transaction authorization input from the user wallet device, and/or product/checkout data (see, e.g., FIGURE 7, 715-717). An example listing of a card authorization request 915-916, substantially in the form of a HTTP(S) POST message including XML-formatted data, is provided below:

```
POST /authorizationrequests.php HTTP/1.1
Host: www.acquirer.com
Content-Type: Application/XML
Content-Length: 1306
<?XML version = “1.0” encoding = “UTF-8”?>
<card_authorization_request>
   <session_ID>4NFU4RG94</order_ID>
   <!--optional data-->
   <timestamp>2011-02-22 15:22:43</timestamp>
   <expiry>00:00:30</expiry>
   <alerts_URL>www.merchant.com/shopcarts.php?sessionID=AEBB4356</alerts_URL>
   <user_ID>john.q.public@gmail.com</user_ID>
   <device_fingerprint>
          <device_IP>192.168.23.126</device_IP>
          <device_MAC>0123.4567.89ab</device_MAC>
          <device_serial>312456768798765432</device_serial>
          <device_ECID>00000AEBCDF12345</device_ECID>
          <device_identifier>jqp_air</device_identifier>
          <device_UDID>21343e34-14f4-8jn4-7yfe-124578632134</device_UDID>
          <device_browser>firefox 2.2</device_browser>
          <device_type>smartphone</device_type>
          <device_model>HTC Hero</device_model>
          <OS>Android 2.2</OS>
          <wallet_app_installed_flag>true</wallet_app_installed_flag>
   </device_fingerprint>
   <purchase_details>
          <total_cost>$121.49</total_cost>
          <cart>
                 <product>
                        <merchant_params>
                               <merchant_id>54TBRELF8</merchant_id>
                               <merchant_name>BARNES, Inc.</merchant_name>
```

```
                    <merchant_auth_key>TMN45GER98</merchant_auth_key>
                </merchant_params>
                <product_type>book</product_type>
                <product_params>
                    <product_title>XML for dummies</product_title>
                    <ISBN>938-2-14-168710-0</ISBN>
                    <edition>2nd ed.</edition>
                    <cover>hardbound</cover>
                </product_params>
                <quantity>2</quantity>
                <unit_cost>$14.46</unit_cost>
                <coupon_id>AY34567</coupon_id>
            <social_flag>ON</social_flag>
            <social_message>Look what I bought today!</social_message>
            <social_networks>facebook twitter</social_networks>
            </product>
            <product>
                <merchant_params>
                    <merchant_id>3FBCR4INC</merchant_id>
                    <merchant_name>Books, Inc.</merchant_name>
                    <merchant_auth_key>1N484MCP</merchant_auth_key>
                </merchant_params>
                <product_type>book</product_type>
                <product_params>
                    <product_title>Sophie’s World</product_title>
                    <ISBN>955-2-14-112310-0</ISBN>
                    <edition>NULL</edition>
                    <cover>hardbound</cover>
                </product_params>
                <quantity>1</quantity>
                <unit_cost>$34.78</unit_cost>
                <coupon_id>null</coupon_id>
            <social_flag>OFF</social_flag>
            </product>
        </cart>
        <cart>
            <product>
                <merchant_params>
                    <merchant_id>RFH5IB4FT</merchant_id>
                    <merchant_name>Amzn, Inc.</merchant_name>
                    <merchant_auth_key>44543DSJFG</merchant_auth_key>
                </merchant_params>
                <product_type>book</product_type>
                <product_params>
                    <product_title>XML - a primer</product_title>
                    <ISBN>938-2-14-1436710-0</ISBN>
                    <edition>2nd ed.</edition>
                    <cover>hardbound</cover>
                </product_params>
                <quantity>1</quantity>
                <unit_cost>$12.93</unit_cost>
                <coupon_id>AY34567</coupon_id>
            <social_flag>ON</social_flag>
            <social_message>Look what I bought today!</social_message>
            <social_networks>facebook twitter</social_networks>
            </product>
            <product>
                <merchant_params>
                    <merchant_id>3FBCR4INC</merchant_id>
                    <merchant_name>BestBooks, Inc.</merchant_name>
                    <merchant_auth_key>1N484MCP</merchant_auth_key>
                </merchant_params>
                <product_type>book</product_type>
                <product_params>
                    <product_title>Sophie’s Choice</product_title>
                    <ISBN>938-2-14-168710-0</ISBN>
                    <edition>1st ed.</edition>
                </product_params>
                <quantity>1</quantity>
                <unit_cost>$44.86</unit_cost>
                <coupon_id>null</coupon_id>
            <social_flag>OFF</social_flag>
            </product>
        </cart>
    </purchase_details>
    <account_params>
        <account>
            <charge_priority>1</charge_priority>
            <charge_ratio>40%</charge_ratio>
            <account_type>debit</account_type>
            <value_exchange_symbol>USD</value_exchange_symbol>
            <account_number>123456789012345</account_number>
            <account_name>John Q. Public</account_name>
            <bill_add>987 Green St #456, Chicago, IL 94652</bill_add>
            <ship_add>987 Green St #456, Chicago, IL 94652</ship_add>
            <CVV_type>dynamic<CVV_type>
            <CVV>http://www.paynet.com/dcvv.php?sessionID=4NFU4RG94</CVV>
            <cloak_flag>ON</cloak_flag>
            <alert_rules>tar1 tar4 tar12</alert_rules>
            <mode>NFC</mode>
        </account>
        <account>
            <charge_priority>1</charge_priority>
            <charge_ratio>60%</charge_ratio>
            <account_type>rewards</account_type>
            <value_exchange_symbol>VME</value_exchange_symbol>
            <account_number>234567890123456</account_number>
            <account_name>John Q. Public</account_name>
            <bill_add>987 Green St #456, Chicago, IL 94652</bill_add>
            <ship_add>987 Green St #456, Chicago, IL 94652</ship_add>
            <CVV_type>static<CVV_type>
            <CVV>173</CVV>
            <cloak_flag>ON</cloak_flag>
            <alert_rules>tar1 tar4 tar12</alert_rules>
            <mode>Bluetooth</mode>
        </account>
        <account>
            <charge_priority>2</charge_priority>
            <charge_ratio>100%</charge_ratio>
            <account_number>345678901234567</account_number>
            <account_type>credit</account_type>
            <value_exchange_symbol>USD</value_exchange_symbol>
            <account_name>John Q. Public</account_name>
            <bill_add>987 Green St #456, Chicago, IL 94652</bill_add>
            <ship_add>987 Green St #456, Chicago, IL 94652</ship_add>
            <CVV_type>static<CVV_type>
            <CVV>173</CVV>
            <cloak_flag>ON</cloak_flag>
            <alert_rules>tar1 tar4 tar12</alert_rules>
            <mode>NFC</mode>
        </account>
    </account_params>
    <shipping_info>
        <shipping_adress>#ref-ANON-123-45-678</shipping_address>
        <ship_type>expedited</ship_type>
        <ship_carrier>FedEx</ship_carrier>
```

```
            <ship_account>ANON-123-45-678</ship_account>
            <tracking_flag>true</tracking_flag>
            <sign_flag>false</sign_flag>
    </shipping_info>
</card_authorization_request>
```

[0071] In some embodiments, the card authorization request generated by the user device may include a minimum of information required 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., 916. The merchant server may forward the card authorization request to a pay gateway server, e.g., 904a, for routing the card authorization request to the appropriate payment network for payment processing. For example, the pay gateway server may be able to select from payment networks, such as Visa, Mastercard, American Express, Paypal, etc., to process various types of transactions including, but not limited to: credit card, debit card, prepaid card, B2B and/or like transactions. In some embodiments, the merchant server may query a database, e.g., merchant/acquirer database 903b, for a network address of the payment gateway server, for example by using a portion of a user payment card number, or a user ID (such as an email address) as a keyword for the database query. For example, the merchant server may issue PHP/SQL commands to query a database table (such as FIGURE 20, Pay Gateways 2019h) for a URL of the pay gateway server. An example payment gateway address query 917, substantially in the form of PHP/SQL commands, is provided below:

```
<?PHP
header('Content-Type: text/plain');
mysql_connect("254.93.179.112",$DBserver,$password); // access database server
mysql_select_db("EWCP_DB.SQL"); // select database table to search
//create query
$query = "SELECT paygate_id paygate_address paygate_URL paygate_name FROM
    PayGatewayTable WHERE card_num LIKE '%' $cardnum";
$result = mysql_query($query); // perform the search query
mysql_close("EWCP_DB.SQL"); // close database access
?>
```

[0072] In response, the merchant/acquirer database may provide the requested payment gateway address, e.g., 918. The merchant server may forward the card authorization request to the pay gateway server using the provided address, e.g., 919. In some embodiments, upon receiving the card authorization request from the merchant server, the pay gateway server may invoke a component to provide one or more services associated with purchase transaction authorization. For example, the pay gateway server may invoke components for fraud prevention, loyalty and/or rewards, and/or other services for which the user-merchant combination is authorized. The pay gateway server may forward the card authorization request to a pay network server, e.g., 905a, for payment processing. For example, the pay gateway server may be able to select from payment networks, such as Visa, Mastercard, American Express, Paypal, etc., to process various types of transactions including, but not limited to: credit card, debit card, prepaid card, B2B and/or like transactions. In some embodiments, the pay gateway server may query a database, e.g., pay gateway database 904b, for a network address of the payment network server, for example by using a portion of a user payment card number, or a user ID (such as an email address) as a keyword for the database query. For example, the pay gateway server may issue PHP/SQL commands to query a database table (such as FIGURE 20, Pay Gateways 2019h) for a URL of the pay network server. An example payment network address query 921, substantially in the form of PHP/SQL commands, is provided below:

```
<?PHP
header('Content-Type: text/plain');
mysql_connect(“254.93.179.112”,$DBserver,$password); // access database server
mysql_select_db(“EWCP_DB.SQL”); // select database table to search
//create query
$query = “SELECT payNET_id payNET_address payNET_URL payNET_name FROM
    PayGatewayTable WHERE card_num LIKE '%' $cardnum”;
$result = mysql_query($query); // perform the search query
mysql_close(“EWCP_DB.SQL”); // close database access
?>
```

[0073] In response, the payment gateway database may provide the requested payment network address, e.g., 922. The pay gateway server may forward the card authorization request to the pay network server using the provided address, e.g., 923.

[0074] With reference to FIGURE 9B, in some embodiments, the pay network server may process the transaction so as to transfer funds for the purchase into an account stored on an acquirer of the merchant. For example, the acquirer may be a financial institution maintaining an account of the merchant. For example, the proceeds of transactions processed by the merchant may be deposited into an account maintained by at a server of the acquirer.

[0075] In some embodiments, the pay network server may generate a query, e.g., 924, for issuer server(s) corresponding to the user-selected payment options. For example, the user's account may be linked to one or more issuer financial institutions ("issuers"), such as banking institutions, which issued the account(s) for the user. For example, such accounts may include, but not be limited to: credit card, debit card, prepaid card, checking, savings, money market, certificates of deposit, stored (cash) value accounts and/or the like. Issuer server(s), e.g., 906a, of the issuer(s) may maintain details of the user's account(s). In some embodiments, a database, e.g., pay network database 905b, may store details of the issuer server(s) associated with the issuer(s). In some embodiments, the pay network server may query a database, e.g., pay network database 905b, for a network address of the issuer(s) server(s), for example by using a portion of a user payment card number, or a user ID (such as an email address) as a keyword for the database query. For example, the merchant server may issue PHP/SQL commands to query a database table (such as FIGURE 20, Issuers 2019f) for network address(es) of the issuer(s) server(s). An example issuer server address(es) query 924, substantially in the form of PHP/SQL commands, is provided below:

```
<?PHP
header('Content-Type: text/plain');
mysql_connect(“254.93.179.112”,$DBserver,$password); // access database server
mysql_select_db(“EWCP_DB.SQL”); // select database table to search
//create query
$query = “SELECT issuer_id issuer_address issuer_URL issuer_name FROM
   IssuersTable WHERE card_num LIKE '%' $cardnum”;
$result = mysql_query($query); // perform the search query
mysql_close(“EWCP_DB.SQL”); // close database access
?>
```

[0076] In response to obtaining the issuer server query, e.g., 924, the pay network database may provide, e.g., 925, the requested issuer server data to the pay network server. In some embodiments, the pay network server may utilize the issuer server data to generate funds authorization request(s), e.g., 926, for each of the issuer server(s) selected based on the pre-defined payment settings associated with the user's virtual wallet, and/or the user's payment options input, and provide the funds authorization request(s) to the issuer server(s). In some embodiments, the funds authorization request(s) may include details such as, but not limited to: the costs to the user involved in the transaction, card account details of the user, user billing and/or shipping information, and/or the like. An example listing of a funds authorization request 926, substantially in the form of a HTTP(S) POST message including XML-formatted data, is provided below:

```
POST /fundsauthorizationrequest.php HTTP/1.1
Host: www.issuer.com
Content-Type: Application/XML
Content-Length: 624
<?XML version = “1.0” encoding = “UTF-8”?>
<funds_authorization_request>
   <request_ID>VNEI39FK</request_ID>
   <timestamp>2011-02-22 15:22:44</timestamp>
   <debit_amount>$72.89</debit_amount>
   <account_params>
         <account>
               <account_type>debit</account_type>
               <value_exchange_symbol>USD</value_exchange_symbol>
               <account_number>123456789012345</account_number>
               <account_name>John Q. Public</account_name>
               <bill_add>987 Green St #456, Chicago, IL 94652</bill_add>
               <ship_add>987 Green St #456, Chicago, IL 94652</ship_add>
               <CVV>1234</CVV>
         </account>
   </account_params>
   <!--optional parameters-->
   <user_device_fingerprint>
          <device_IP>192.168.23.126</device_IP>
          <device_MAC>0123.4567.89ab</device_MAC>
          <device_serial>312456768798765432</device_serial>
          <device_ECID>00000AEBCDF12345</device_ECID>
          <device_identifier>jqp_air</device_identifier>
          <device_UDID>21343e34-14f4-8jn4-7yfe-124578632134</device_UDID>
          <device_browser>firefox 2.2</device_browser>
          <device_type>smartphone</device_type>
          <device_model>HTC Hero</device_model>
          <OS>Android 2.2</OS>
          <wallet_app_installed_flag>true</wallet_app_installed_flag>
   </user_device_fingerprint>
</funds_authorization_request>
```

[0077] In some embodiments, an issuer server may parse the authorization request(s), and based on the request details may query a database, e.g., user profile database 906b, for data associated with an account linked to the user. For example, the merchant server may issue PHP/SQL commands to query a database table (such as FIGURE 20, Accounts 2019d) for user account(s) data. An example user account(s) query 927, substantially in the form of PHP/SQL commands, is provided below:

```
<?PHP
header('Content-Type: text/plain');
mysql_connect(“254.93.179.112”,$DBserver,$password); // access database server
mysql_select_db(“EWCP_DB.SQL”); // select database table to search
//create query
$query = “SELECT issuer user_id user_name user_balance account_type FROM
   AccountsTable WHERE account_num LIKE '%' $accountnum”;
$result = mysql_query($query); // perform the search query
mysql_close(“EWCP_DB.SQL”); // close database access
?>
```

[0078] In some embodiments, on obtaining the user account(s) data, e.g., 928, the issuer server may determine whether the user can pay for the transaction using funds available in the account, 929. For example, the issuer server may determine whether the user has a sufficient balance remaining in the account, sufficient credit associated with the account, and/or the like. Based on the determination, the issuer server(s) may provide a funds authorization response, e.g., 930, to the pay network server. For example, the issuer server(s) may provide a HTTP(S) POST message similar to the examples above. In some embodiments, if at least one issuer server determines that the user cannot pay for the transaction using the funds available in the account, the pay network server may request payment options again from the user (e.g., by providing an authorization fail message to the user device and requesting the user device to provide new payment options), and re-attempt authorization for the purchase transaction. In some embodiments, if the number of failed authorization attempts exceeds a threshold, the pay network server may abort the authorization process, and provide an “authorization fail” message to the merchant server, user device and/or client.

[0079] In some embodiments, the pay network server may obtain the funds authorization response including a notification of successful authorization, and parse the message to extract authorization details. Upon determining that the user possesses sufficient funds for the transaction, e.g., 931, the pay network server may invoke a component to provide value-add services for the user.

[0080] In some embodiments, the pay network server may generate a transaction data record from the authorization request and/or authorization response, and store the details of the transaction and authorization relating to the transaction in a transactions database. For example, the pay network server may issue PHP/SQL commands to store the data to a database table (such as FIGURE 20, Transactions 2019i). An example transaction store command, substantially in the form of PHP/SQL commands, is provided below:

```
<?PHP
header('Content-Type: text/plain');
mysql_connect("254.92.185.103”,$DBserver,$password); // access database server
mysql_select("EWCP_DB.SQL"); // select database to append
mysql_query(“INSERT INTO TransactionsTable (PurchasesTable (timestamp,
   purchase_summary_list, num_products, product_summary, product_quantity,
   transaction_cost, account_params_list, account_name, account_type,
   account_num, billing_addres, zipcode, phone, sign, merchant_params_list,
   merchant_id, merchant_name, merchant_auth_key)
VALUES (time(), $purchase_summary_list, $num_products, $product_summary,
   $product_quantity, $transaction_cost, $account_params_list, $account_name,
   $account_type, $account_num, $billing_addres, $zipcode, $phone, $sign,
   $merchant_params_list, $merchant_id, $merchant_name, $merchant_auth_key)”);
   // add data to table in database
mysql_close("EWCP_DB.SQL"); // close connection to database
?>
```

[0081] In some embodiments, the pay network server may forward a transaction authorization response, e.g., 932, to the user wallet device, PoS client, and/or merchant server. The merchant may obtain the transaction authorization response, and determine from it that the user possesses sufficient funds in the card account to conduct the transaction. The merchant server may add a record of the transaction for the user to a batch of transaction data relating to authorized transactions. For example, the merchant may append the XML data pertaining to the user transaction to an XML data file comprising XML data for transactions that have been authorized for various users, e.g., 933, and store the XML data file, e.g., 934, in a database, e.g., merchant database 404. For example, a batch XML data file may be structured similar to the example XML data structure template provided below:

```
<?XML version = “1.0” encoding = “UTF-8”?>
```

```
<merchant_data>
   <merchant_id>3FBCR4INC</merchant_id>
   <merchant_name>Books & Things, Inc.</merchant_name>
   <merchant_auth_key>1NNF484MCP59CHB27365</merchant_auth_key>
   <account_number>123456789</account_number>
</merchant_data>
<transaction_data>
   <transaction 1>
         ...
   </transaction 1>
   <transaction 2>
         ...
   </transaction 2>
         .
         .
         .
   <transaction n>
         ...
   </transaction n>
</transaction_data>
```

[0082] In some embodiments, the server may also generate a purchase receipt, e.g., 933, and provide the purchase receipt to the client, e.g., 935. The client may render and display, e.g., 936, the purchase receipt for the user. In some embodiments, the user's wallet device may also provide a notification of successful authorization to the user. For example, the PoS client/user device may render a webpage, electronic message, text / SMS message, buffer a voicemail, emit a ring tone, and/or play an audio message, etc., and provide output including, but not limited to: sounds, music, audio, video, images, tactile feedback, vibration alerts (e.g., on vibration-capable client devices such as a smartphone etc.), and/or the like.

[0083] FIGURES 10A-B show logic flow diagrams illustrating example aspects of transforming a user virtual wallet access input via a Purchase Transaction Authorization ("PTA") component into a purchase transaction receipt notification. With reference to

FIGURE 10A, in some embodiments, a user may wish to utilize a virtual wallet account to purchase a product, service, offering, and/or the like ("product"), from a merchant via a merchant online site or in the merchant's store. The user may utilize a physical card, or a user wallet device to access the user's virtual wallet account. For example, the user wallet device may be a personal/laptop computer, cellular telephone, smartphone, tablet, eBook reader, netbook, gaming console, and/or the like. The user may provide a wallet access input, e.g., 1001, into the user wallet device. In various embodiments, the user input may include, but not be limited to: a single tap (e.g., a one-tap mobile app purchasing embodiment) of a touchscreen interface, keyboard entry, card swipe, activating a RFID/NFC equipped hardware device (e.g., electronic card having multiple accounts, smartphone, tablet, etc.) within the user device, mouse clicks, depressing buttons on a joystick/game console, voice commands, single/multi-touch gestures on a touch-sensitive interface, touching user interface elements on a touch-sensitive display, and/or the like. In some embodiments, the user wallet device may authenticate the user based on the user's wallet access input, instantiate a wallet application (see, e.g., FIGURES 13-19B) upon authenticating the user, and provide virtual wallet features for the user via the wallet application, e.g., 1002-1003.

[0084] In some embodiments, upon authenticating the user for access to virtual wallet features, the user wallet device may provide a transaction authorization input, e.g., 1004, to a point-of-sale ("PoS") client. For example, the user wallet device may communicate with the PoS client via Bluetooth, Wi-Fi, cellular communication, one- or two-way near-field communication ("NFC"), and/or the like. In embodiments where the user utilizes a plastic card instead of the user wallet device, the user may swipe the plastic card at the PoS client to transfer information from the plastic card into the PoS client. In embodiments where the user utilizes a user wallet device, the user wallet device may provide payment information to the PoS client, formatted according to a data formatting protocol appropriate to the communication mechanism employed in the communication between the user wallet device and the PoS client.

[0085] In some embodiments, the PoS client may obtain the transaction authorization input, and parse the input to extract payment information from the transaction authorization input, e.g., 1005. For example, the PoS client may utilize a parser, such as the example parsers provided below in the discussion with reference to FIGURE 20. The PoS client may generate a card authorization request, e.g., 1006, using the obtained transaction authorization input from the user wallet device, and/or product/checkout data (see, e.g., FIGURE 7, 715-717).

[0086] In some embodiments, the PoS client may provide the generated card authorization request to the merchant server. The merchant server may forward the card authorization request to a pay gateway server, for routing the card authorization request to the appropriate payment network for payment processing. For example, the pay gateway server may be able to select from payment networks, such as Visa, Mastercard, American Express, Paypal, etc., to process various types of transactions including, but not limited to: credit card, debit card, prepaid card, B2B and/or like transactions. In some embodiments, the merchant server may query a database, e.g., 1008, for a network address of the payment gateway server, for example by using a portion of a user payment card number, or a user ID (such as an email address) as a keyword for the database query. In response, the merchant/acquirer database may provide the requested payment gateway address, e.g., 1010. The merchant server may forward the card authorization request to the pay gateway server using the provided address. In some embodiments, upon receiving the card authorization request from the merchant server, the pay gateway server may invoke a component to provide one or more service associated with purchase transaction authorization, e.g., 1011. For example, the pay gateway server may invoke components for fraud prevention (see e.g., VerifyChat, FIGURE 3E), loyalty and/or rewards, and/or other services for which the user-merchant combination is authorized.

[0087] The pay gateway server may forward the card authorization request to a pay network server for payment processing, e.g., 1014. For example, the pay gateway server may be able to select from payment networks, such as Visa, Mastercard, American Express, Paypal, etc., to process various types of transactions including, but not limited to: credit card, debit card, prepaid card, B2B and/or like transactions. In some embodiments, the pay gateway server may query a database, e.g., 1012, for a network address of the payment network server, for example by using a portion of a user payment card number, or a user ID (such as an email address) as a keyword for the database query. In response, the payment gateway database may provide the requested payment network address, e.g., 1013. The pay gateway server may forward the card authorization request to the pay network server using the provided address, e.g., 1014.

[0088] With reference to FIGURE 10B, in some embodiments, the pay network server may process the transaction so as to transfer funds for the purchase into an account stored on an acquirer of the merchant. For example, the acquirer may be a financial institution maintaining an account of the merchant. For example, the proceeds of transactions processed by the merchant may be deposited into an account maintained by at a server of the acquirer. In some embodiments, the pay network server may generate a query, e.g., 1015, for issuer server(s) corresponding to the user-selected payment options. For example, the user's account may be linked to one or more issuer financial institutions ("issuers"), such as banking institutions, which issued the account(s) for the user. For example, such accounts may include, but not be limited to: credit card, debit card, prepaid card, checking, savings, money market, certificates of deposit, stored (cash) value accounts and/or the like. Issuer server(s) of the issuer(s) may maintain details of the user's account(s). In some embodiments, a database, e.g., a pay network database, may store details of the issuer server(s) associated with the issuer(s). In some embodiments, the pay network server may query a database, e.g., 1015, for a network address of the issuer(s) server(s), for example by using a portion of a user payment card number, or a user ID (such as an email address) as a keyword for the database query.

[0089] In response to obtaining the issuer server query, the pay network database may provide, e.g., 1016, the requested issuer server data to the pay network server. In some embodiments, the pay network server may utilize the issuer server data to generate funds authorization request(s), e.g., 1017, for each of the issuer server(s) selected based on the pre-defined payment settings associated with the user's virtual wallet, and/or the user's payment options input, and provide the funds authorization request(s) to the issuer server(s). In some embodiments, the funds authorization request(s) may include details such as, but not limited to: the costs to the user involved in the transaction, card account details of the user, user billing and/or shipping information, and/or the like. In some embodiments, an issuer server may parse the authorization request(s), e.g., 1018, and based on the request details may query a database, e.g., 1019, for data associated with an account linked to the user.

[0090] In some embodiments, on obtaining the user account(s) data, e.g., 1020, the issuer server may determine whether the user can pay for the transaction using funds available in the account, e.g., 1021. For example, the issuer server may determine whether the user has a sufficient balance remaining in the account, sufficient credit associated with the account, and/or the like. Based on the determination, the issuer server(s) may provide a funds authorization response, e.g., 1022, to the pay network server. In some embodiments, if at least one issuer server determines that the user cannot pay for the transaction using the funds available in the account, the pay network server may request payment options again from the user (e.g., by providing an authorization fail message to the user device and requesting the user device to provide new payment options), and re-attempt authorization for the purchase transaction. In some embodiments, if the number of failed authorization attempts exceeds a threshold, the pay network server may abort the authorization process, and provide an "authorization fail" message to the merchant server, user device and/or client.

[0091] In some embodiments, the pay network server may obtain the funds authorization response including a notification of successful authorization, and parse the message to extract authorization details. Upon determining that the user possesses sufficient funds for the transaction, e.g., 1023, the pay network server may invoke a component to provide value-add services for the user, e.g., 1023.

[0092] In some embodiments, the pay network server may forward a transaction authorization response to the user wallet device, PoS client, and/or merchant server.

The merchant may parse, e.g., 1024, the transaction authorization response, and determine from it that the user possesses sufficient funds in the card account to conduct the transaction, e.g., 1025, option"Yes." The merchant server may add a record of the transaction for the user to a batch of transaction data relating to authorized transactions. For example, the merchant may append the XML data pertaining to the user transaction to an XML data file comprising XML data for transactions that have been authorized for various users, e.g., 1026, and store the XML data file, e.g., 1027, in a database. In some embodiments, the server may also generate a purchase receipt, e.g., 1028, and provide the purchase receipt to the client. The client may render and display, e.g., 1029, the purchase receipt for the user. In some embodiments, the user's wallet device may also provide a notification of successful authorization to the user. For example, the PoS client/user device may render a webpage, electronic message, text / SMS message, buffer a voicemail, emit a ring tone, and/or play an audio message, etc., and provide output including, but not limited to: sounds, music, audio, video, images, tactile feedback, vibration alerts (e.g., on vibration-capable client devices such as a smartphone etc.), and/or the like.

[0093] FIGURES 11A-B show data flow diagrams illustrating example aspects of transforming a merchant transaction batch data query via a Purchase Transaction Clearance ("PTC") component into an updated payment ledger record. With reference to FIGURE 11A, in some embodiments, a merchant server, e.g., 1103a, may initiate clearance of a batch of authorized transactions. For example, the merchant server may generate a batch data request, e.g., 1111, and provide the request, to a merchant database, e.g., 1103b. For example, the merchant server may utilize PHP/SQL commands similar to the examples provided above to query a relational database. In response to the batch data request, the database may provide the requested batch data, e.g., 1112. The server may generate a batch clearance request, e.g., 1113, using the batch data obtained from the database, and provide, e.g., 1114, the batch clearance request to an acquirer server, e.g., 1107a. For example, the merchant server may provide a HTTP(S) POST message including XML-formatted batch data in the message body for the acquirer server. The acquirer server may generate, e.g., 1115, a batch payment request using the obtained batch clearance request, and provide, e.g., 1118, the batch payment request to the pay network server, e.g., 1105a. The pay network server may parse the batch payment request, and extract the transaction data for each transaction stored in the batch payment request, e.g., 1119. The pay network server may store the transaction data, e.g., 1120, for each transaction in a database, e.g., pay network database 1105b. In some embodiments, the pay network server may invoke a component to provide value-add analytics services based on analysis of the transactions of the merchant for whom the EWCP is clearing purchase transactions. Thus, in some embodiments, the pay network server may provide analytics-based value-added services for the merchant and/or the merchant's users.

[0094] With reference to FIGURE 11B, in some embodiments, for each extracted transaction, the pay network server may query, e.g., 1123, a database, e.g., pay network database 1105b, for an address of an issuer server. For example, the pay network server may utilize PHP/SQL commands similar to the examples provided above. The pay network server may generate an individual payment request, e.g., 1125, for each transaction for which it has extracted transaction data, and provide the individual payment request, e.g., 1125, to the issuer server, e.g., 1106a. For example, the pay network server may provide an individual payment request to the issuer server(s) as a

HTTP(S) POST message including XML-formatted data. An example listing of an individual payment request 1125, substantially in the form of a HTTP(S) POST message including XML-formatted data, is provided below:

```
POST /paymentrequest.php HTTP/1.1
Host: www.issuer.com
Content-Type: Application/XML
Content-Length: 788
<?XML version = “1.0” encoding = “UTF-8”?>
<pay_request>
   <request_ID>CNI4ICNW2</request_ID>
   <timestamp>2011-02-22 17:00:01</timestamp>
   <pay_amount>$72.89</pay_amount>
   <account_params>
          <account>
                 <account_type>debit</account_type>
                 <value_exchange_symbol>USD</value_exchange_symbol>
                 <account_number>123456789012345</account_number>
                 <account_name>John Q. Public</account_name>
                 <bill_add>987 Green St #456, Chicago, IL 94652</bill_add>
                 <ship_add>987 Green St #456, Chicago, IL 94652</ship_add>
                 <CVV>1234</CVV>
          </account>
   </account_params>
</pay_request>
```

[0095] In some embodiments, the issuer server may generate a payment command, e.g., 1127. For example, the issuer server may issue a command to deduct funds from the user's account (or add a charge to the user's credit card account). The issuer server may issue a payment command, e.g., 1127, to a database storing the user's account information, e.g., user profile database 1106b. The issuer server may provide an individual payment confirmation, e.g., 1128, to the pay network server, which may forward, e.g., 1129, the funds transfer message to the acquirer server. An example listing of an individual payment confirmation 1128, substantially in the form of a HTTP(S) POST message including XML-formatted data, is provided below:

```
POST /clearance.php HTTP/1.1
Host: www.acquirer.com
Content-Type: Application/XML
Content-Length: 206
<?XML version = “1.0” encoding = “UTF-8”?>
<deposit_ack>
    <request_ID>CNI4ICNW2</request_ID>
    <clear_flag>true</clear_flag>
    <timestamp>2011-02-22 17:00:02</timestamp>
    <deposit_amount>$72.89</deposit_amount>
</deposit_ack>
```

[0096] In some embodiments, the acquirer server may parse the individual payment confirmation, and correlate the transaction (e.g., using the request_ID field in the example above) to the merchant. The acquirer server may then transfer the funds specified in the funds transfer message to an account of the merchant. For example, the acquirer server may query, e.g. 1130, an acquirer database 1107b for payment ledger and/or merchant account data, e.g., 1131. The acquirer server may utilize payment ledger and/or merchant account data from the acquirer database, along with the individual payment confirmation, to generate updated payment ledger and/or merchant account data, e.g., 1132. The acquirer server may then store, e.g., 1133, the updated payment ledger and/or merchant account data to the acquire database.

[0097] FIGURES 12A-B show logic flow diagrams illustrating example aspects of transforming a merchant transaction batch data query via a Purchase Transaction Clearance (“PTC”) component into an updated payment ledger record. With reference to FIGURE 12A, in some embodiments, a merchant server may initiate clearance of a batch of authorized transactions. For example, the merchant server may generate a batch data request, e.g., 1201, and provide the request to a merchant database. In response to the batch data request, the database may provide the requested batch data, e.g., 1202. The server may generate a batch clearance request, e.g., 1203, using the batch data obtained from the database, and provide the batch clearance request to an acquirer server. The acquirer server may parse, e.g., 1204, the obtained batch clearance request, and generate, e.g., 1207, a batch payment request using the obtained batch clearance request to provide, the batch payment request to a pay network server. For example, the acquirer server may query, e.g., 1205, an acquirer database for an address of a payment network server, and utilize the obtained address, e.g., 1206, to forward the generated batch payment request to the pay network server.

[0098] The pay network server may parse the batch payment request obtained from the acquirer server, and extract the transaction data for each transaction stored in the batch payment request, e.g., 1208. The pay network server may store the transaction data, e.g., 1209, for each transaction in a pay network database. In some embodiments, the pay network server may invoke a component, e.g., 1210, to provide analytics based on the transactions of the merchant for whom purchase transaction are being cleared.

[0099] With reference to FIGURE 12B, in some embodiments, for each extracted transaction, the pay network server may query, e.g., 1211, a pay network database for an address of an issuer server. The pay network server may generate an individual payment request, e.g., 1213, for each transaction for which it has extracted transaction data, and provide the individual payment request to the issuer server. In some embodiments, the issuer server may parse the individual payment request, e.g., 1214, and generate a payment command, e.g., 1215, based on the parsed individual payment request. For example, the issuer server may issue a command to deduct funds from the user's account (or add a charge to the user's credit card account). The issuer server may issue a payment command, e.g., 1215, to a database storing the user's account information, e.g., a user profile database. The issuer server may provide an individual payment confirmation, e.g., 1217, to the pay network server, which may forward, e.g., 1218, the individual payment confirmation to the acquirer server.

[00100] In some embodiments, the acquirer server may parse the individual payment confirmation, and correlate the transaction (e.g., using the request_ID field in the example above) to the merchant. The acquirer server may then transfer the funds specified in the funds transfer message to an account of the merchant. For example, the acquirer server may query, e.g. 1219, an acquirer database for payment ledger and/or merchant account data, e.g., 1220. The acquirer server may utilize payment ledger and/or merchant account data from the acquirer database, along with the individual payment confirmation, to generate updated payment ledger and/or merchant account data, e.g., 1221. The acquirer server may then store, e.g., 1222, the updated payment ledger and/or merchant account data to the acquire database.

[00101] FIGURE 13 shows a user interface diagram illustrating an overview of example features of virtual wallet applications in some embodiments of the EWCP. FIGURE 13 shows an illustration of various exemplary features of a virtual wallet mobile application 1300. Some of the features displayed include a wallet 1301, social integration via TWITTER, FACEBOOK, etc., offers and loyalty 1303, snap mobile purchase 1304, alerts 1305 and security, setting and analytics 1396. These features are explored in further detail below.

[00102] FIGURES 14A-K show user interface and logic flow diagrams illustrating example features of virtual wallet applications in a shopping mode, in some embodiments of the EWCP. With reference to FIGURE 14A, some embodiments of the virtual wallet mobile app facilitate and greatly enhance the shopping experience of consumers. A variety of shopping modes, as shown in FIGURE 14A, may be available for a consumer to peruse. In one implementation, for example, a user may launch the shopping mode by selecting the shop icon 1410 at the bottom of the user interface. A user may type in an item in the search field 1412 to search and/or add an item to a cart 1411. A user may also use a voice activated shopping mode by saying the name or description of an item to be searched and/or added to the cart into a microphone 1413. In a further implementation, a user may also select other shopping options 1414 such as current items 1415, bills 1416, address book 1417, merchants 1418 and local proximity 1419.

[00103] In one embodiment, for example, a user may select the option current items 1415, as shown in the left most user interface of FIGURE 14A. When the current items 1415 option is selected, the middle user interface may be displayed. As shown, the middle user interface may provide a current list of items 1415a-h in a user's shopping cart 1411. A user may select an item, for example item 1415a, to view product description 1415j of the selected item and/or other items from the same merchant. The price and total payable information may also be displayed, along with a QR code 1415k that captures the information necessary to effect a snap mobile purchase transaction.

[00104] With reference to FIGURE 14B, in another embodiment, a user may select the bills 1416 option. Upon selecting the bills 1416 option, the user interface may display a list of bills and/or receipts 1416a-h from one or more merchants. Next to each of the bills, additional information such as date of visit, whether items from multiple stores are present, last bill payment date, auto-payment, number of items, and/or the like may be displayed. In one example, the wallet shop bill 1416a dated January 20, 2011 may be selected. The wallet shop bill selection may display a user interface that provides a variety of information regarding the selected bill. For example, the user interface may display a list of items 1416k purchased, <<1416i>>, a total number of items and the corresponding value. For example, 7 items worth $102.54 were in the selected wallet shop bill. A user may now select any of the items and select buy again to add purchase the items. The user may also refresh offers 1416j to clear any invalid offers from last time and/or search for new offers that may be applicable for the current purchase. As shown in FIGURE 14B, a user may select two items for repeat purchase. Upon addition, a message 1416l may be displayed to confirm the addition of the two items, which makes the total number of items in the cart 14.

[00105] With reference to FIGURE 14C, in yet another embodiment, a user may select the address book option 1417 to view the address book 1417a which includes a list of contacts 1417b and make any money transfers or payments. In one embodiment, the address book may identify each contact using their names and available and/or preferred modes of payment. For example, a contact Amanda G. may be paid via social pay (e.g., via FACEBOOK) as indicated by the icon 1417c. In another example, money may be transferred to Brian S. via QR code as indicated by the QR code icon 1417d. In yet another example, Charles B. may accept payment via near field communication 1417e, Bluetooth 1417f and email 1417g. Payment may also be made via USB 1417h (e.g., by physically connecting two mobile devices) as well as other social channels such as TWITTER.

[00106] In one implementation, a user may select Joe P. for payment. Joe P., as shown in the user interface, has an email icon 1417g next to his name indicating that Joe P. accepts payment via email. When his name is selected, the user interface may display his contact information such as email, phone, etc. If a user wishes to make a payment to Joe P. by a method other than email, the user may add another transfer mode 1417j to his contact information and make a payment transfer. With reference to FIGURE 14D, the user may be provided with a screen 1417k where the user can enter an amount to send Joe, as well as add other text to provide Joe with context for the payment transaction 1417l. The user can choose modes (e.g., SMS, email, social networking) via which Joe may be contacted via graphical user interface elements, 1417m. As the user types, the text entered may be provided for review within a GUI element 1417n. When the user has completed entering in the necessary information, the user can press the send button 1417o to send the social message to Joe. If Joe also has a virtual wallet application, Joe may be able to review 1417p social pay message within the app, or directly at the website of the social network (e.g., for Twitter™, Facebook®, etc.). Messages may be aggregated from the various social networks and other sources (e.g., SMS, email). The method of redemption appropriate for each messaging mode may be indicated along with the social pay message. In the illustration in FIGURE 14D, the SMS 1417q Joe received indicates that Joe can redeem the $5 obtained via SMS by replying to the SMS and entering the hash tag value '#1234'. In the same illustration, Joe has also received a message 1417r via Facebook®, which includes a URL link that Joe can activate to initiate redemption of the $25 payment.

[00107] With reference to FIGURE 14E, in some other embodiments, a user may select merchants 1418 from the list of options in the shopping mode to view a select list of merchants 1418a-e. In one implementation, the merchants in the list may be affiliated to the wallet, or have affinity relationship with the wallet. In another implementation, the merchants may include a list of merchants meeting a user-defined or other criteria. For example, the list may be one that is curated by the user, merchants where the user most frequently shops or spends more than an x amount of sum or shopped for three consecutive months, and/or the like. In one implementation, the user may further select one of the merchants, Amazon 1418a for example. The user may then navigate through the merchant's listings to find items of interest such as 1418f-j. Directly through the wallet and without visiting the merchant site from a separate page, the user may make a selection of an item 1418j from the catalog of Amazon 1418a. As shown in the right most user interface of FIGURE 14D, the selected item may then be added to cart. The message 1418k indicates that the selected item has been added to the cart, and updated number of items in the cart is now 13.

[00108] With reference to FIGURE 14F, in one embodiment, there may be a local proximity option 1419 which may be selected by a user to view a list of merchants that are geographically in close proximity to the user. For example, the list of merchants 1419a-e may be the merchants that are located close to the user. In one implementation, the mobile application may further identify when the user in a store based on the user's location. For example, position icon 1419d may be displayed next to a store (e.g., Walgreens) when the user is in close proximity to the store. In one implementation, the mobile application may refresh its location periodically in case the user moved away from the store (e.g., Walgreens). In a further implementation, the user may navigate the offerings of the selected Walgreens store through the mobile application. For example, the user may navigate, using the mobile application, to items 1419f-j available on aisle 5 of Walgreens. In one implementation, the user may select corn 1419i from his or her mobile application to add to cart 1419k.

[00109] With reference to FIGURE 14G, in another embodiment, the local proximity option 1419 may include a store map and a real time map features among others. For example, upon selecting the Walgreens store, the user may launch an aisle map 1419l which displays a map 1419m showing the organization of the store and the position of the user (indicated by a yellow circle). In one implementation, the user may easily configure the map to add one or more other users (e.g., user's kids) to share each other's location within the store. In another implementation, the user may have the option to launch a "store view" similar to street views in maps. The store view 1419n may display images/video of the user's surrounding. For example, if the user is about to enter aisle 5, the store view map may show the view of aisle 5. Further the user may manipulate the orientation of the map using the navigation tool 1419o to move the store view forwards, backwards, right, left as well clockwise and counterclockwise rotation.

[00110] FIGURES 14H-K show user interface and logic flow diagrams illustrating example aspects of virtual store injection into a virtual wallet application in some embodiments of the EWCP. In some implementations, the virtual wallet application may presents screens 1420 and 1430, respectively, as depicted in FIGURE 14H. In FIGURE 14H, 1420, the virtual wallet application displays a list of merchants participating in the virtual wallet of the EWCP, e.g., 601-605. Similarly, in FIGURE 14H, 1430, the virtual wallet application displays a list of merchants participating in the virtual wallet of the EWCP and at or nearby the approximate location of the user the user. The user may click on any of the merchants listed in the two screens 1420 and 1430, to be injected into the store inventory of the merchant. Upon injection, the user may be presented with a screen such as 1440. In some implementations, the virtual wallet application may be able to store, maintain and manage a plurality of shopping carts and/or wishlists for a user. The carts may be purely virtual or they may represent the contents of a physical cart in a merchant store. The user may activate any of the carts listed (e.g., 1440) to view the items currently stored in a cart (e.g., 1441-1446). In some implementations, the virtual wallet application may also provide wishlists, e.g., tech wishlist, with items that the user desires to be gifted. In some implementations, the virtual wallet may allow the user to quickly change carts or wishlists from another cart or wishlist, using a pop-up menu.

[00111] With reference to FIGURE 14I, in some embodiments, the user may be injected into a virtual reality 2D/3D storefront of the merchant, e.g., 1447. For example, the user may be presented with a plan map view of the store 1448. In some map views, the user may provided with the user's location (e.g., using GPS, or if not available, then using a coarse approximation using a cellular signal). In some implementations, the locations of the user's prior and current purchases may be provided for the user, if the user wishes (see 1449, the user can turn the indications off, in some implementations). In some implementations, the user may be provided with a 3D aisle view of an aisle within the virtual storefront. The user may point the view direction(s) at any of the objects to obtain virtual tools to obtain items from off the "virtual shelf," and place them in the user's virtual cart. The screen at 1450 shows an augmented reality view of an aisle, where user may see pins of items suggested by a concierge, or that were bookmarked in their cart/wishlist highlighted through a live video view 1453. In some embodiments, the color of a pin depicted in the augmented reality view may be indicative of an attribute of the suggestion, e.g., a discount offer, a warning not to buy, a prior purchase, etc. In still further embodiments, a color of a 3D viewer window may indicate additional attributes such as, without limitation, whether the product was recommended by the user's social graph, the product's rating (e.g., according to experts, the user's friends, Internet users, etc.), and/or the like.

[00112] In another view, a virtual store aisle view (e.g., akin to a Google map Street View) may be navigated 1451 when the consumer is not at the store, but would like to look for product; the directional control 1451 allows for navigation up and down the aisle, and rotation and views of items at the merchant location. Additionally, consumers may tap items in the shelves and create a new product pin, which may then be added 1452 to a cart or wishlist for further transacting.

[00113] FIGURE 14J shows a logic flow diagram illustrating example aspects of virtual store injection into a virtual wallet application in some embodiments of the EWCP, e.g., a Virtual Wallet Store Injection ("VWSI") component. In some embodiments, a user may provide a user input into a user device executing a virtual wallet application, e.g., 1461. The user device ("client") may obtain the user input, e.g., 1462. In various implementations, the user input may include, but not be limited to: keyboard entry, card swipe, activating a RFID/NFC enabled hardware device (e.g., electronic card having multiple accounts, smartphone, tablet, etc.), mouse clicks, depressing buttons on a joystick/game console, voice commands, single/multi-touch gestures on a touch-sensitive interface, touching user interface elements on a touch-sensitive display, and/or the like. The client may determine the type of user input, e.g., 1463. For example, the client may determine whether the user input is one that requests that the a virtual store of merchant(s) be injected into the virtual wallet application. If the user input constitutes a store injection request, e.g., 1464, option “Yes,” the client may generate a store injection request message, e.g., 1465. For example, the client may provide a store injection request message to a server as a HTTP(S) POST message including XML-formatted data. An example listing of a store injection request message, substantially in the form of a HTTP(S) POST message including XML-formatted data, is provided below:

```
POST /storeinjectionrequest.php HTTP/1.1
Host: www.merchant.com
Content-Type: Application/XML
Content-Length: 453
<?XML version = “1.0” encoding = “UTF-8”?>
<store_injection_request>
   <session_ID>ANAv483</session_ID>
   <timestamp>2052-01-01 12:12:12</timestamp>
   <user_id>john.q.public</user_id>
   <injection_data_request>
          <type>NEW STORE REQUEST</type>
          <merchant_id>JKHVHCGV456</merchant_id>
          <store_id>1234</store_id>
          <injection_point>ENTRY</injection_point>
          <augmented_reality_flag>ON</augmented_reality_flag>
          <view_type>street view</view_type>
          <alt_view_type>map view</alt_view_type>
   </injection_data_request>

```

[00114] In some embodiments, the server may obtain the store injection request from the client, and may parse the message, e.g., 1466. For example, the client may utilize a parser such as the example parsers discussed below in the description with reference to FIGURE 20. The client may extract the request parameters from the client’s message and generate a query for the requested store injection data, e.g., 1467. Examples of store injection data include, without limitation: product information, product images, product animations, videos, media content, animations, store wireframes, street view data, map data, lists of products (e.g., XML data), URLs pointing to other store injection data, augmented reality data, executable script (e.g., JavaScript™, Adobe Flash® object, .bundle files, HTML5 code, etc.), and/or the like. For example, the server may issue PHP/SQL commands to query a database table (such as FIGURE 20, Shop Sessions 2019i) for store injection data. An example store injection data query command, substantially in the form of PHP/SQL commands, is provided below:

```
<?PHP
header('Content-Type: text/plain');
mysql_connect(“254.93.179.112”,$DBserver,$password); // access database server
mysql_select_db(“EWCP_DB.SQL”); // select database table to search
//create query
$query = “SELECT product_information, product_images, product_animations,
    videos, media_content, animations, store_wireframes, street_view_data,
    map_data, product_list, pointer_URL_list, augmented_reality_data,
    executable_script_list FROM ShopSessionTable WHERE session_id LIKE '%'
    $sessionid”;
$result = mysql_query($query); // perform the search query
mysql_close(“EWCP_DB.SQL”); // close database access
?>
```

[00115] In some embodiments, in response to the query, a database of the server may provide the data requested by the server, e.g., 1468. Using the obtained data, the server may generate a store injection response message, e.g., 1469. For example, the server may provide a store injection response message to the client as a HTTP(S) POST message including XML-formatted data. An example listing of a store injection response message, substantially in the form of a HTTP(S) POST message including XML-formatted data, is provided below:

```
POST /storeinjectionresponse.php HTTP/1.1
Host: www.client.com
Content-Type: Application/XML
Content-Length: 1777
<?XML version = “1.0” encoding = “UTF-8”?>
<store_injection_response>
   <session_ID>ANAv483</session_ID>
   <timestamp>2052-01-01 12:12:15</timestamp>
   <user_id>john.q.public</user_id>
   <merchant_id>JKHVHCGV456</merchant_id>
   <store_id>1234</store_id>
   <injection_point>ENTRY</injection_point>
   <augmented_reality_flag>ON</augmented_reality_flag>
   <view_type>street view</view_type>
   <alt_view_type>map view</alt_view_type>
   <inventory_data>
          <categories>
                 <books>
                        ...
                        <product_params>
                               <product_type>Self Help</product_type>
                               <product_title>XML for dummies</product_title>
                               <ISBN>938-2-14-168710-0</ISBN>
                               <edition>2nd ed.</edition>
                               <cover>hardbound</cover>
                               <price>$59</price>
                               <inventory>70</ inventory>
                        </product_params>
                        ...
                 </books>
                 ...
                 <electronics>
                        <vendors>
                               ...
                               <Apple>
                                      ...
                                      <product_params>
                                             <product_type>tablet</product_type>
                        <product_name>iPad</product_name>
                        <serialno>12345678</ serialno >
                        <modelno>12345</modelno>
                        <description>64GB, 4G</description>
                        <price>$829</price>
                        <inventory>7</ inventory>
                    </product_params>
                    ...
                </Apple>
                ...
        </electronics>
    </categories>
    <products>
        ...
        <product_params>
            <publisher_params>
                <publisher_id>54TBRELF8</publisher_id>
                <publisher_name>McGraw-Hill,
Inc.</publisher_name>
            </publisher_params>
            <product_type>book</product_type>
            <product_params>
                <product_title>XML for dummies</product_title>
                <ISBN>938-2-14-168710-0</ISBN>
                <edition>2nd ed.</edition>
                <cover>hardbound</cover>
            </product_params>
            <inventory_level>2</inventory_level>
            <unit_cost>$14.46</unit_cost>
            <coupon_id>AY34567</coupon_id>
        </product_params>
        ...
        <product_params>
            <product_id>HJKFG345</product_id>
            <product_name>Philips Sonicare</product_name>
            <vendor_name>Philips, Inc.</vendor_name>
            <model>EH57</model>
            <product_type>Toothbrush</product_type>
            <inventory_level>12</inventory_level>
            <unit_cost>$34.78</unit_cost>
            <coupon_id>null</coupon_id>
        </product_params>
```

```
                 ...
          </products>
          ...
    </inventory_data>
    <store_injection_enhanced_interface_data>
          <floorplan_URL>www.inject.com?id= ANAv483&type=img</floorplan_URL>
          <UI_script_URL>www.inject.com?id= ANAv483&type=script</UI_script_URL>
          <ShopAssistant_UIbundle_url>www.inject.com?id=
    ANAv483&type=bundle</ShopAssistant_UIbundle_url>
    <AugmentedRealityFloorplanCartPinOverlayUI_html5_url>www.inject.com?id=
    ANAv483&type=html5</AugmentedRealityFloorplanCartPinOverlayUI_html5_url>
          <InteractiveStore_flash_url>www.inject.com?id=
    ANAv483&type=flash</InteractiveStore_flash_url>
    </store_injection_enhanced_interface_data>
</store_injection_response>
```

[00116] In some embodiments, the client may obtain the store injection response message, and parse the message, e.g., 1470. The client may render a visualization of the virtual store using the extracted store injection data, e.g., 1471, and display the rendered visualization for the user via a display device of the client, e.g., 1472.

[00117] With reference to FIGURE 14K, in some embodiments, the user may provide a user input into the virtual store visualization generated by the client, e..g, 1481. The client may obtain the user input, e.g., 1482, and may determine the type of input provided by the user into the client, e.g., 1483. If the user input represents a card addition request, e.g., 1484, option "Yes," the client may identify a product that the user desires to add to a shopping cart, e.g., 1485, and may add the user-selected product to a virtual shopping cart or wishlist, e.g., 1486. If the user input represents a store navigation request (e.g., walking through the aisle within a virtual store), e.g., 1487, option "Yes," the client may identify the store navigation action requested by the user, e.g., 1488, and may generate a store injection request message for the server to process the user's store navigation request (see, e.g., 1465-1472). If the user input represents a checkout request, e.g., 1489, option “Yes,” the client may generate a card authorization request, e.g., 1490, as a trigger for a purchase transaction, and may provide the card authorization request to a purchase transaction authorization component such as the example PTA component discussed in the description with reference to FIGURE 10A.

[00118] FIGURES 15A-F show user interface diagrams illustrating example features of virtual wallet applications in a payment mode, in some embodiments of the EWCP. With reference to FIGURE 15A, in one embodiment, the wallet mobile application may provide a user with a number of options for paying for a transaction via the wallet mode 1510. In one implementation, an example user interface 1511 for making a payment is shown. The user interface may clearly identify the amount 1512 and the currency 1513 for the transaction. The amount may be the amount payable and the currency may include real currencies such as dollars and euros, as well as virtual currencies such as reward points. The amount of the transaction 1514 may also be prominently displayed on the user interface. The user may select the funds tab 1516 to select one or more forms of payment 1517, which may include various credit, debit, gift, rewards and/or prepaid cards. The user may also have the option of paying, wholly or in part, with reward points. For example, the graphical indicator 1518 on the user interface shows the number of points available, the graphical indicator 1519 shows the number of points to be used towards the amount due 234.56 and the equivalent 1520 of the number of points in a selected currency (USD, for example).

[00119] In one implementation, the user may combine funds from multiple sources to pay for the transaction. The amount 1515 displayed on the user interface may provide an indication of the amount of total funds covered so far by the selected forms of payment (e.g., Discover card and rewards points). The user may choose another form of payment or adjust the amount to be debited from one or more forms of payment until the amount 1515 matches the amount payable 1514. Once the amounts to be debited from one or more forms of payment are finalized by the user, payment authorization may begin.

[00120] In one implementation, the user may select a secure authorization of the transaction by selecting the cloak button 1522 to effectively cloak or anonymize some (e.g., pre-configured) or all identifying information such that when the user selects pay button 1521, the transaction authorization is conducted in a secure and anonymous manner. In another implementation, the user may select the pay button 1521 which may use standard authorization techniques for transaction processing. In yet another implementation, when the user selects the social button 1523, a message regarding the transaction may be communicated to one of more social networks (set up by the user) which may post or announce the purchase transaction in a social forum such as a wall post or a tweet. In one implementation, the user may select a social payment processing option 1523. The indicator 1524 may show the authorizing and sending social share data in progress.

[00121] In another implementation, a restricted payment mode 1525 may be activated for certain purchase activities such as prescription purchases. The mode may be activated in accordance with rules defined by issuers, insurers, merchants, payment processor and/or other entities to facilitate processing of specialized goods and services. In this mode, the user may scroll down the list of forms of payments 1526 under the funds tab to select specialized accounts such as a flexible spending account (FSA) 1527, health savings account (HAS), and/or the like and amounts to be debited to the selected accounts. In one implementation, such restricted payment mode 1525 processing may disable social sharing of purchase information.

[00122] In one embodiment, the wallet mobile application may facilitate importing of funds via the import funds user interface 1528. For example, a user who is unemployed may obtain unemployment benefit fund 1529 via the wallet mobile application. In one implementation, the entity providing the funds may also configure rules for using the fund as shown by the processing indicator message 1530. The wallet may read and apply the rules prior, and may reject any purchases with the unemployment funds that fail to meet the criteria set by the rules. Example criteria may include, for example, merchant category code (MCC), time of transaction, location of transaction, and/or the like. As an example, a transaction with a grocery merchant having MCC 5411 may be approved, while a transaction with a bar merchant having an MCC 5813 may be refused.

[00123] With reference to FIGURE 15B, in one embodiment, the wallet mobile application may facilitate dynamic payment optimization based on factors such as user location, preferences and currency value preferences among others. For example, when a user is in the United States, the country indicator 1531 may display a flag of the United States and may set the currency 1533 to the United States. In a further implementation, the wallet mobile application may automatically rearrange the order in which the forms of payments 1535 are listed to reflect the popularity or acceptability of various forms of payment. In one implementation, the arrangement may reflect the user’s preference, which may not be changed by the wallet mobile application.

[00124] Similarly, when a German user operates a wallet in Germany, the mobile wallet application user interface may be dynamically updated to reflect the country of operation 1532 and the currency 1534. In a further implementation, the wallet application may rearrange the order in which different forms of payment 1536 are listed based on their acceptance level in that country. Of course, the order of these forms of payments may be modified by the user to suit his or her own preferences.

[00125] With reference to FIGURE 15C, in one embodiment, the payee tab 1537 in the wallet mobile application user interface may facilitate user selection of one or more payees receiving the funds selected in the funds tab. In one implementation, the user interface may show a list of all payees 1538 with whom the user has previously transacted or available to transact. The user may then select one or more payees. The payees 1538 may include larger merchants such as Amazon.com Inc., and individuals such as Jane P. Doe. Next to each payee name, a list of accepted payment modes for the payee may be displayed. In one implementation, the user may select the payee Jane P. Doe 1539 for receiving payment. Upon selection, the user interface may display additional identifying information relating to the payee.

[00126] With reference to FIGURE 15D, in one embodiment, the mode tab 1540 may facilitate selection of a payment mode accepted by the payee. A number of payment modes may be available for selection. Example modes include, blue tooth 1541, wireless 1542, snap mobile by user-obtained QR code 1543, secure chip 1544, TWITTER 1545, near-field communication (NFC) 1546, cellular 1547, snap mobile by user-provided QR code 1548, USB 1549 and FACEBOOK 1550, among others. In one implementation, only the payment modes that are accepted by the payee may be selectable by the user. Other non-accepted payment modes may be disabled.

[00127] With reference to FIGURE 15E, in one embodiment, the offers tab 1551 may provide real-time offers that are relevant to items in a user's cart for selection by the user. The user may select one or more offers from the list of applicable offers 1552 for redemption. In one implementation, some offers may be combined, while others may not. When the user selects an offer that may not be combined with another offer, the unselected offers may be disabled. In a further implementation, offers that are recommended by the wallet application's recommendation engine may be identified by an indicator, such as the one shown by 1553. In a further implementation, the user may read the details of the offer by expanding the offer row as shown by 1554 in the user interface.

[00128] With reference to FIGURE 15F, in one embodiment, the social tab 1555 may facilitate integration of the wallet application with social channels 1556. In one implementation, a user may select one or more social channels 1556 and may sign in to the selected social channel from the wallet application by providing to the wallet application the social channel user name and password 1557 and signing in 1558. The user may then use the social button 1559 to send or receive money through the integrated social channels. In a further implementation, the user may send social share data such as purchase information or links through integrated social channels. In another embodiment, the user supplied login credentials may allow EWCP to engage in interception parsing.

[00129] FIGURE 16 shows a user interface diagram illustrating example features of virtual wallet applications, in a history mode, in some embodiments of the EWCP. In one embodiment, a user may select the history mode 1610 to view a history of prior purchases and perform various actions on those prior purchases. For example, a user may enter a merchant identifying information such as name, product, MCC, and/or the like in the search bar 1611. In another implementation, the user may use voice activated search feature by clicking on the microphone icon 1614. The wallet application may query the storage areas in the mobile device or elsewhere (e.g., one or more databases and/or tables remote from the mobile device) for transactions matching the search keywords. The user interface may then display the results of the query such as transaction 1615. The user interface may also identify the date 1612 of the transaction, the merchants and items 1613 relating to the transaction, a barcode of the receipt confirming that a transaction was made, the amount of the transaction and any other relevant information.

[00130] In one implementation, the user may select a transaction, for example transaction 1615, to view the details of the transaction. For example, the user may view the details of the items associated with the transaction and the amounts 1616 of each item. In a further implementation, the user may select the show option 1617 to view actions 1618 that the user may take in regards to the transaction or the items in the transaction. For example, the user may add a photo to the transaction (e.g., a picture of the user and the iPad the user bought). In a further implementation, if the user previously shared the purchase via social channels, a post including the photo may be generated and sent to the social channels for publishing. In one implementation, any sharing may be optional, and the user, who did not share the purchase via social channels, may still share the photo through one or more social channels of his or her choice directly from the history mode of the wallet application. In another implementation, the user may add the transaction to a group such as company expense, home expense, travel expense or other categories set up by the user. Such grouping may facilitate year-end accounting of expenses, submission of work expense reports, submission for value added tax (VAT) refunds, personal expenses, and/or the like. In yet another implementation, the user may buy one or more items purchased in the transaction. The user may then execute a transaction without going to the merchant catalog or site to find the items. In a further implementation, the user may also cart one or more items in the transaction for later purchase.

[00131] The history mode, in another embodiment, may offer facilities for obtaining and displaying ratings 1619 of the items in the transaction. The source of the ratings may be the user, the user's friends (e.g., from social channels, contacts, etc.), reviews aggregated from the web, and/or the like. The user interface in some implementations may also allow the user to post messages to other users of social channels (e.g., TWITTER or FACEBOOK). For example, the display area 1620 shows FACEBOOK message exchanges between two users. In one implementation, a user may share a link via a message 1621. Selection of such a message having embedded link to a product may allow the user to view a description of the product and/or purchase the product directly from the history mode.

[00132] In one embodiment, the history mode may also include facilities for exporting receipts. The export receipts pop up 1622 may provide a number of options for exporting the receipts of transactions in the history. For example, a user may use one or more of the options 1625, which include save (to local mobile memory, to server, to a cloud account, and/or the like), print to a printer, fax, email, and/or the like. The user may utilize his or her address book 1623 to look up email or fax number for exporting. The user may also specify format options 1624 for exporting receipts. Example format options may include, without limitation, text files (.doc, .txt, .rtf, iif, etc.), spreadsheet (.csv, .xls, etc.), image files (.jpg, .tff, .png, etc.), portable document format (.pdf), postscript (.ps), and/or the like. The user may then click or tap the export button 1627 to initiate export of receipts.

[00133] FIGURES 17A-E show user interface diagrams illustrating example features of virtual wallet applications in a snap mode, in some embodiments of the EWCP. With reference to FIGURE 17A, in one embodiment, a user may select the snap mode 2110 to access its snap features. The snap mode may handle any machine-readable representation of data. Examples of such data may include linear and 2D bar codes such as UPC code and QR codes. These codes may be found on receipts, product packaging, and/or the like. The snap mode may also process and handle pictures of receipts, products, offers, credit cards or other payment devices, and/or the like. An example user interface in snap mode is shown in FIGURE 17A. A user may use his or her mobile phone to take a picture of a QR code 1715 and/or a barcode 1714. In one implementation, the bar 1713 and snap frame 1715 may assist the user in snapping codes properly. For example, the snap frame 1715, as shown, does not capture the entirety of the code 1716. As such, the code captured in this view may not be resolvable as information in the code may be incomplete. This is indicated by the message on the bar 1713 that indicates that the snap mode is still seeking the code. When the code 1716 is completely framed by the snap frame 1715, the bar message may be updated to, for example, “snap found.” Upon finding the code, in one implementation, the user may initiate code capture using the mobile device camera. In another implementation, the snap mode may automatically snap the code using the mobile device camera.

[00134] With reference to FIGURE 17B, in one embodiment, the snap mode may facilitate payment reallocation post transaction. For example, a user may buy grocery and prescription items from a retailer Acme Supermarket. The user may, inadvertently or for ease of checkout for example, use his or her Visa card to pay for both grocery and prescription items. However, the user may have an FSA account that could be used to pay for prescription items, and which would provide the user tax benefits. In such a situation, the user may use the snap mode to initiate transaction reallocation.

[00135] As shown, the user may enter a search term (e.g., bills) in the search bar 2121. The user may then identify in the tab 1722 the receipt 1723 the user wants to reallocate. Alternatively, the user may directly snap a picture of a barcode on a receipt, and the snap mode may generate and display a receipt 1723 using information from the barcode. The user may now reallocate 1725. In some implementations, the user may also dispute the transaction 1724 or archive the receipt 1726.

[00136] In one implementation, when the reallocate button 1725 is selected, the wallet application may perform optical character recognition (OCR) of the receipt. Each of the items in the receipt may then be examined to identify one or more items which could be charged to which payment device or account for tax or other benefits such as cash back, reward points, etc. In this example, there is a tax benefit if the prescription medication charged to the user’s Visa card is charged to the user’s FSA. The wallet application may then perform the reallocation as the back end. The reallocation process may include the wallet contacting the payment processor to credit the amount of the prescription medication to the Visa card and debit the same amount to the user's FSA account. In an alternate implementation, the payment processor (e.g., Visa or MasterCard) may obtain and OCR the receipt, identify items and payment accounts for reallocation and perform the reallocation. In one implementation, the wallet application may request the user to confirm reallocation of charges for the selected items to another payment account. The receipt 1727 may be generated after the completion of the reallocation process. As discussed, the receipt shows that some charges have been moved from the Visa account to the FSA.

[00137] With reference to FIGURE 17C, in one embodiment, the snap mode may facilitate payment via pay code such as barcodes or QR codes. For example, a user may snap a QR code of a transaction that is not yet complete. The QR code may be displayed at a merchant POS terminal, a web site, or a web application and may be encoded with information identifying items for purchase, merchant details and other relevant information. When the user snaps such as a QR code, the snap mode may decode the information in the QR code and may use the decoded information to generate a receipt 1732. Once the QR code is identified, the navigation bar 1731 may indicate that the pay code is identified. The user may now have an option to add to cart 1733, pay with a default payment account 1734 or pay with wallet 1735.

[00138] In one implementation, the user may decide to pay with default 1734. The wallet application may then use the user's default method of payment, in this example the wallet, to complete the purchase transaction. Upon completion of the transaction, a receipt may be automatically generated for proof of purchase. The user interface may also be updated to provide other options for handling a completed transaction. Example options include social 1737 to share purchase information with others, reallocate 1738 as discussed with regard to FIGURE 17B, and archive 1739 to store the receipt.

[00139] With reference to FIGURE 17D, in one embodiment, the snap mode may also facilitate offer identification, application and storage for future use. For example, in one implementation, a user may snap an offer code 1741 (e.g., a bar code, a QR code, and/or the like). The wallet application may then generate an offer text 1742 from the information encoded in the offer code. The user may perform a number of actions on the offer code. For example, the user use the find button 1743 to find all merchants who accept the offer code, merchants in the proximity who accept the offer code, products from merchants that qualify for the offer code, and/or the like. The user may also apply the offer code to items that are currently in the cart using the add to cart button 1744. Furthermore, the user may also save the offer for future use by selecting the save button 1745.

[00140] In one implementation, after the offer or coupon 1746 is applied, the user may have the option to find qualifying merchants and/or products using find, the user may go to the wallet using 1748, and the user may also save the offer or coupon 1746 for later use.

[00141] With reference to FIGURE 17E, in one embodiment, the snap mode may also offer facilities for adding a funding source to the wallet application. In one implementation, a pay card such as a credit card, debit card, pre-paid card, smart card and other pay accounts may have an associated code such as a bar code or QR code. Such a code may have encoded therein pay card information including, but not limited to, name, address, pay card type, pay card account details, balance amount, spending limit, rewards balance, and/or the like. In one implementation, the code may be found on a face of the physical pay card. In another implementation, the code may be obtained by accessing an associated online account or another secure location. In yet another implementation, the code may be printed on a letter accompanying the pay card. A user, in one implementation, may snap a picture of the code. The wallet application may identify the pay card 1751 and may display the textual information 1752 encoded in the pay card. The user may then perform verification of the information 1752 by selecting the verify button 1753. In one implementation, the verification may include contacting the issuer of the pay card for confirmation of the decoded information 1752 and any other relevant information. In one implementation, the user may add the pay card to the wallet by selecting the 'add to wallet' button 1754. The instruction to add the pay card to the wallet may cause the pay card to appear as one of the forms of payment under the funds tab 1516 discussed in FIGURE 15A. The user may also cancel importing of the pay card as a funding source by selecting the cancel button 1755. When the pay card has been added to the wallet, the user interface may be updated to indicate that the importing is complete via the notification display 1756. The user may then access the wallet 1757 to begin using the added pay card as a funding source.

[00142] FIGURE 18 shows a user interface diagram illustrating example features of virtual wallet applications, in an offers mode, in some embodiments of the EWCP. In some implementations, the EWCP may allow a user to search for offers for products and/or services from within the virtual wallet mobile application. For example, the user may enter text into a graphical user interface ("GUI") element 1811, or issue voice commands by activating GUI element 1812 and speaking commands into the device. In some implementations, the EWCP may provide offers based on the user's prior behavior, demographics, current location, current cart selection or purchase items, and/or the like. For example, if a user is in a brick-and-mortar store, or an online shopping website, and leaves the (virtual) store, then the merchant associated with the store may desire to provide a sweetener deal to entice the consumer back into the (virtual) store. The merchant may provide such an offer 1813. For example, the offer may provide a discount, and may include an expiry time. In some implementations, other users may provide gifts (e.g., 1814) to the user, which the user may redeem. In some implementations, the offers section may include alerts as to payment of funds outstanding to other users (e.g., 1815). In some implementations, the offers section may include alerts as to requesting receipt of funds from other users (e.g., 1816). For example, such a feature may identify funds receivable from other applications (e.g., mail, calendar, tasks, notes, reminder programs, alarm, etc.), or by a manual entry by the user into the virtual wallet application. In some implementations, the offers section may provide offers from participating merchants in the EWCP, e.g., 1817-1819, 1820. These offers may sometimes be assembled using a combination of participating merchants, e.g., 1817. In some implementations, the EWCP itself may provide offers for users contingent on the user utilizing particular payment forms from within the virtual wallet application, e.g., 1820.

[00143] FIGURES 19A-B show user interface diagrams illustrating example features of virtual wallet applications, in a security and privacy mode, in some embodiments of the EWCP. With reference to FIGURE 19A, in some implementations, the user may be able to view and/or modify the user profile and/or settings of the user, e.g., by activating a user interface element. For example, the user may be able to view/modify a user name (e.g., 1911a-b), account number (e.g., 1912a-b), user security access code (e.g., 1913-b), user pin (e.g., 1914-b), user address (e.g., 1915-b), social security number associated with the user (e.g., 1916-b), current device GPS location (e.g., 1917-b), user account of the merchant in whose store the user currently is (e.g., 1918-b), the user's rewards accounts (e.g., 1919-b), and/or the like. In some implementations, the user may be able to select which of the data fields and their associated values should be transmitted to facilitate the purchase transaction, thus providing enhanced data security for the user. For example, in the example illustration in FIGURE 19A, the user has selected the name 1911a, account number 1912a, security code 1913a, merchant account ID 1918a and rewards account ID 1919a as the fields to be sent as part of the notification to process the purchase transaction. In some implementations, the user may toggle the fields and/or data values that are sent as part of the notification to process the purchase transactions. In some implementations, the app may provide multiple screens of data fields and/or associated values stored for the user to select as part of the purchase order transmission. In some implementations, the app may provide the EWCP with the GPS location of the user. Based on the GPS location of the user, the EWCP may determine the context of the user (e.g., whether the user is in a store, doctor's office, hospital, postal service office, etc.). Based on the context, the user app may present the appropriate fields to the user, from which the user may select fields and/or field values to send as part of the purchase order transmission.

[00144] For example, a user may go to doctor's office and desire to pay the co-pay for doctor's appointment. In addition to basic transactional information such as account number and name, the app may provide the user the ability to select to transfer medical records, health information, which may be provided to the medical provider, insurance company, as well as the transaction processor to reconcile payments between the parties. In some implementations, the records may be sent in a Health Insurance Portability and Accountability Act (HIPAA)-compliant data format and encrypted, and only the recipients who are authorized to view such records may have appropriate decryption keys to decrypt and view the private user information.

[00145] With reference to FIGURE 19B, in some implementations, the app executing on the user's device may provide a "VerifyChat" feature for fraud prevention. For example, the EWCP may detect an unusual and/or suspicious transaction. The EWCP may utilize the VerifyChat feature to communicate with the user, and verify the authenticity of the originator of the purchase transaction. In various implementations, the EWCP may send electronic mail message, text (SMS) messages, Facebook® messages, Twitter™ tweets, text chat, voice chat, video chat (e.g., Apple FaceTime), and/or the like to communicate with the user. For example, the EWCP may initiate a video challenge for the user, e.g., 1921. For example, the user may need to present him/her-self via a video chat, e.g., 1922. In some implementations, a customer service representative, e.g., agent 1924, may manually determine the authenticity of the user using the video of the user. In some implementations, the EWCP may utilize face, biometric and/or like recognition (e.g., using pattern classification techniques) to determine the identity of the user. In some implementations, the app may provide reference marker (e.g., cross-hairs, target box, etc.), e.g., 1923, so that the user may the video to facilitate the EWCP's automated recognition of the user. In some implementations, the user may not have initiated the transaction, e.g., the transaction is fraudulent. In such implementations, the user may cancel the challenge. The EWCP may then cancel the transaction, and/or initiate fraud investigation procedures on behalf of the user.

[00146] In some implementations, the EWCP may utilize a text challenge procedure to verify the authenticity of the user, e.g., 1925. For example, the EWCP may communicate with the user via text chat, SMS messages, electronic mail, Facebook® messages, Twitter™ tweets, and/or the like. The EWCP may pose a challenge question, e.g., 1926, for the user. The app may provide a user input interface element(s) (e.g., virtual keyboard 1928) to answer the challenge question posed by the EWCP. In some implementations, the challenge question may be randomly selected by the EWCP automatically; in some implementations, a customer service representative may manually communicate with the user. In some implementations, the user may not have initiated the transaction, e.g., the transaction is fraudulent. In such implementations, the user may cancel the text challenge. The EWCP may cancel the transaction, and/or initiate fraud investigation on behalf of the user.

EWCP Controller

[00147] FIGURE 20 shows a block diagram illustrating example aspects of a EWCP controller 2001. In this embodiment, the EWCP controller 2001 may serve to aggregate, process, store, search, serve, identify, instruct, generate, match, and/or facilitate interactions with a computer through various technologies, and/or other related data.

[00148] Users, e.g., 2033a, which may be people and/or other systems, may engage information technology systems (e.g., computers) to facilitate information processing. In turn, computers employ processors to process information; such processors 2003 may be referred to as central processing units (CPU). One form of processor is referred to as a microprocessor. CPUs use communicative circuits to pass binary encoded signals acting as instructions to enable various operations. These instructions may be operational and/or data instructions containing and/or referencing other instructions and data in various processor accessible and operable areas of memory 2029 (e.g., registers, cache memory, random access memory, etc.). Such communicative instructions may be stored and/or transmitted in batches (e.g., batches of instructions) as programs and/or data components to facilitate desired operations. These stored instruction codes, e.g., programs, may engage the CPU circuit components and other motherboard and/or system components to perform desired operations. One type of program is a computer operating system, which, may be executed by CPU on a computer; the operating system enables and facilitates users to access and operate computer information technology and resources. Some resources that may be employed in information technology systems include: input and output mechanisms through which data may pass into and out of a computer; memory storage into which data may be saved; and processors by which information may be processed. These information technology systems may be used to collect data for later retrieval, analysis, and manipulation, which may be facilitated through a database program. These information technology systems provide interfaces that allow users to access and operate various system components.

[00149] In one embodiment, the EWCP controller 2001 may be connected to and/or communicate with entities such as, but not limited to: one or more users from user input devices 2011; peripheral devices 2012; an optional cryptographic processor device 2028; and/or a communications network 2013. For example, the EWCP controller 2001 may be connected to and/or communicate with users, e.g., 2033a, operating client device(s), e.g., 2033b, including, but not limited to, personal computer(s), server(s) and/or various mobile device(s) including, but not limited to, cellular telephone(s), smartphone(s) (e.g., iPhone®, Blackberry®, Android OS-based phones etc.), tablet computer(s) (e.g., Apple iPad™, HP Slate™, Motorola Xoom™, etc.), eBook reader(s) (e.g., Amazon Kindle™, Barnes and Noble's Nook™ eReader, etc.), laptop computer(s), notebook(s), netbook(s), gaming console(s) (e.g., XBOX Live™, Nintendo® DS, Sony PlayStation® Portable, etc.), portable scanner(s), and/or the like.

[00150] Networks are commonly thought to comprise the interconnection and interoperation of clients, servers, and intermediary nodes in a graph topology. It should be noted that the term "server" as used throughout this application refers generally to a computer, other device, program, or combination thereof that processes and responds to the requests of remote users across a communications network. Servers serve their information to requesting "clients." The term "client" as used herein refers generally to a computer, program, other device, user and/or combination thereof that is capable of processing and making requests and obtaining and processing any responses from servers across a communications network. A computer, other device, program, or combination thereof that facilitates, processes information and requests, and/or furthers the passage of information from a source user to a destination user is commonly referred to as a "node." Networks are generally thought to facilitate the transfer of information from source points to destinations. A node specifically tasked with furthering the passage of information from a source to a destination is commonly called a "router." There are many forms of networks such as Local Area Networks (LANs), Pico networks, Wide Area Networks (WANs), Wireless Networks (WLANs), etc.

For example, the Internet is generally accepted as being an interconnection of a multitude of networks whereby remote clients and servers may access and interoperate with one another.

[00151] The EWCP controller 2001 may be based on computer systems that may comprise, but are not limited to, components such as: a computer systemization 2002 connected to memory 2029.

Computer Systemization

[00152] A computer systemization 2002 may comprise a clock 2030, central processing unit ("CPU(s)" and/or "processor(s)" (these terms are used interchangeably throughout the disclosure unless noted to the contrary)) 2003, a memory 2029 (e.g., a read only memory (ROM) 2006, a random access memory (RAM) 2005, etc.), and/or an interface bus 2007, and most frequently, although not necessarily, are all interconnected and/or communicating through a system bus 2004 on one or more (mother)board(s) 2002 having conductive and/or otherwise transportive circuit pathways through which instructions (e.g., binary encoded signals) may travel to effectuate communications, operations, storage, etc. The computer systemization may be connected to a power source 2086; e.g., optionally the power source may be internal. Optionally, a cryptographic processor 2026 and/or transceivers (e.g., ICs) 2074 may be connected to the system bus. In another embodiment, the cryptographic processor and/or transceivers may be connected as either internal and/or external peripheral devices 2012 via the interface bus I/O. In turn, the transceivers may be connected to antenna(s) 2075, thereby effectuating wireless transmission and reception of various communication and/or sensor protocols; for example the antenna(s) may connect to: a

Texas Instruments WiLink WL1283 transceiver chip (e.g., providing 802.11n, Bluetooth 3.0, FM, global positioning system (GPS) (thereby allowing EWCP controller to determine its location)); Broadcom BCM4329FKUBG transceiver chip (e.g., providing 802.11n, Bluetooth 2.1 + EDR, FM, etc.), BCM28150 (HSPA+) and BCM2076 (Bluetooth 4.0, GPS, etc.); a Broadcom BCM4750IUB8 receiver chip (e.g., GPS); an Infineon Technologies X-Gold 618-PMB9800 (e.g., providing 2G/3G HSDPA/HSUPA communications); Intel's XMM 7160 (LTE & DC-HSPA), Qualcom's CDMA(2000), Mobile Data/Station Modem, Snapdragon; and/or the like. The system clock may have a crystal oscillator and generates a base signal through the computer systemization's circuit pathways. The clock may be coupled to the system bus and various clock multipliers that will increase or decrease the base operating frequency for other components interconnected in the computer systemization. The clock and various components in a computer systemization drive signals embodying information throughout the system. Such transmission and reception of instructions embodying information throughout a computer systemization may be referred to as communications. These communicative instructions may further be transmitted, received, and the cause of return and/or reply communications beyond the instant computer systemization to: communications networks, input devices, other computer systemizations, peripheral devices, and/or the like. It should be understood that in alternative embodiments, any of the above components may be connected directly to one another, connected to the CPU, and/or organized in numerous variations employed as exemplified by various computer systems.

[00153] The CPU comprises at least one high-speed data processor adequate to execute program components for executing user and/or system-generated requests.

Often, the processors themselves will incorporate various specialized processing units, such as, but not limited to: floating point units, integer processing units, integrated system (bus) controllers, logic operating units, memory management control units, etc., and even specialized processing sub-units like graphics processing units, digital signal processing units, and/or the like. Additionally, processors may include internal fast access addressable memory, and be capable of mapping and addressing memory 2029 beyond the processor itself; internal memory may include, but is not limited to: fast registers, various levels of cache memory (e.g., level 1, 2, 3, etc.), RAM, etc. The processor may access this memory through the use of a memory address space that is accessible via instruction address, which the processor can construct and decode allowing it to access a circuit path to a specific memory address space having a memory state/value. The CPU may be a microprocessor such as: AMD's Athlon, Duron and/or Opteron; ARM's classic (e.g., ARM7/9/11), embedded (Coretx-M/R), application (Cortex-A), embedded and secure processors; IBM and/or Motorola's DragonBall and PowerPC; IBM's and Sony's Cell processor; Intel's Atom, Celeron (Mobile), Core (2/Duo/i3/i5/i7), Itanium, Pentium, Xeon, and/or XScale; and/or the like processor(s). The CPU interacts with memory through instruction passing through conductive and/or transportive conduits (e.g., (printed) electronic and/or optic circuits) to execute stored instructions (i.e., program code). Such instruction passing facilitates communication within the EWCP controller and beyond through various interfaces. Should processing requirements dictate a greater amount speed and/or capacity, distributed processors (e.g., Distributed EWCP), mainframe, multi-core, parallel, and/or super-computer architectures may similarly be employed. Alternatively, should deployment requirements dictate greater portability, smaller mobile devices (e.g., smartphones, Personal Digital Assistants (PDAs), etc.) may be employed.

[00154] Depending on the particular implementation, features of the EWCP may be achieved by implementing a microcontroller such as CAST's R8051XC2 microcontroller; Intel's MCS 51 (i.e., 8051 microcontroller); and/or the like. Also, to implement certain features of the EWCP, some feature implementations may rely on embedded components, such as: Application-Specific Integrated Circuit ("ASIC"), Digital Signal Processing ("DSP"), Field Programmable Gate Array ("FPGA"), and/or the like embedded technology. For example, any of the EWCP component collection (distributed or otherwise) and/or features may be implemented via the microprocessor and/or via embedded components; e.g., via ASIC, coprocessor, DSP, FPGA, and/or the like. Alternately, some implementations of the EWCP may be implemented with embedded components that are configured and used to achieve a variety of features or signal processing.

[00155] Depending on the particular implementation, the embedded components may include software solutions, hardware solutions, and/or some combination of both hardware/software solutions. For example, EWCP features discussed herein may be achieved through implementing FPGAs, which are a semiconductor devices containing programmable logic components called "logic blocks", and programmable interconnects, such as the high performance FPGA Virtex series and/or the low cost Spartan series manufactured by Xilinx. Logic blocks and interconnects can be programmed by the customer or designer, after the FPGA is manufactured, to implement any of the EWCP features. A hierarchy of programmable interconnects allow logic blocks to be interconnected as needed by the EWCP system designer/administrator, somewhat like a one-chip programmable breadboard. An FPGA's logic blocks can be programmed to perform the operation of basic logic gates such as AND, and XOR, or more complex combinational operators such as decoders or simple mathematical operations. In most FPGAs, the logic blocks also include memory elements, which may be circuit flip-flops or more complete blocks of memory. In some circumstances, the EWCP may be developed on regular FPGAs and then migrated into a fixed version that more resembles ASIC implementations. Alternate or coordinating implementations may migrate EWCP controller features to a final ASIC instead of or in addition to FPGAs. Depending on the implementation all of the aforementioned embedded components and microprocessors may be considered the “CPU” and/or “processor” for the EWCP.

Power Source

[00156] The power source 2086 may be of any standard form for powering small electronic circuit board devices such as the following power cells: alkaline, lithium hydride, lithium ion, lithium polymer, nickel cadmium, solar cells, and/or the like. Other types of AC or DC power sources may be used as well. In the case of solar cells, in one embodiment, the case provides an aperture through which the solar cell may capture photonic energy. The power cell 2086 is connected to at least one of the interconnected subsequent components of the EWCP thereby providing an electric current to all the interconnected components. In one example, the power source 2086 is connected to the system bus component 2004. In an alternative embodiment, an outside power source 2086 is provided through a connection across the I/O 2008 interface. For example, a USB and/or IEEE 1394 connection carries both data and power across the connection and is therefore a suitable source of power.

Interface Adapters

[00157] Interface bus(ses) 2007 may accept, connect, and/or communicate to a number of interface adapters, frequently, although not necessarily in the form of adapter cards, such as but not limited to: input output interfaces (I/O) 2008, storage interfaces 2009, network interfaces 2010, and/or the like. Optionally, cryptographic processor interfaces 2027 similarly may be connected to the interface bus. The interface bus provides for the communications of interface adapters with one another as well as with other components of the computer systemization. Interface adapters are adapted for a compatible interface bus. Interface adapters may connect to the interface bus via expansion and/or slot architecture. Various expansion and/or slot architectures may be employed, such as, but not limited to: Accelerated Graphics Port (AGP), Card Bus, ExpressCard, (Extended) Industry Standard Architecture ((E)ISA), Micro Channel Architecture (MCA), NuBus, Peripheral Component Interconnect (Extended) (PCI(X)), PCI Express, Personal Computer Memory Card International Association (PCMCIA), Thunderbolt, and/or the like.

[00158] Storage interfaces 2009 may accept, communicate, and/or connect to a number of storage devices such as, but not limited to: storage devices 2014, removable disc devices, and/or the like. Storage interfaces may employ connection protocols such as, but not limited to: (Ultra) (Serial) Advanced Technology Attachment (Packet Interface) ((Ultra) (Serial) ATA(PI)), (Enhanced) Integrated Drive Electronics ((E)IDE), Institute of Electrical and Electronics Engineers (IEEE) 1394, Ethernet, fiber channel, Small Computer Systems Interface (SCSI), Thunderbolt, Universal Serial Bus (USB), and/or the like.

[00159] Network interfaces 2010 may accept, communicate, and/or connect to a communications network 2013. Through a communications network 2013, the EWCP controller is accessible through remote clients 2033b (e.g., computers with web browsers) by users 2033a. Network interfaces may employ connection protocols such as, but not limited to: direct connect, Ethernet (thick, thin, twisted pair 10/100/1000 Base T, and/or the like), Token Ring, wireless connection such as IEEE 802.11a-x, and/or the like. Should processing requirements dictate a greater amount speed and/or capacity, distributed network controllers (e.g., Distributed EWCP), architectures may similarly be employed to pool, load balance, and/or otherwise increase the communicative bandwidth required by the EWCP controller. A communications network may be any one and/or the combination of the following: a direct interconnection; the Internet; a Local Area Network (LAN); a Metropolitan Area Network (MAN); an Operating Missions as Nodes on the Internet (OMNI); a secured custom connection; a Wide Area Network (WAN); a wireless network (e.g., employing protocols such as, but not limited to a Wireless Application Protocol (WAP), I-mode, and/or the like); and/or the like. A network interface may be regarded as a specialized form of an input output interface. Further, multiple network interfaces 2010 may be used to engage with various communications network types 2013. For example, multiple network interfaces may be employed to allow for the communication over broadcast, multicast, and/or unicast networks.

[00160] Input Output interfaces (I/O) 2008 may accept, communicate, and/or connect to user input devices 2011, peripheral devices 2012, cryptographic processor devices 2028, and/or the like. I/O may employ connection protocols such as, but not limited to: audio: analog, digital, monaural, RCA, stereo, and/or the like; data: Apple Desktop Bus (ADB), Bluetooth, IEEE 1394a-b, serial, universal serial bus (USB); infrared; joystick; keyboard; midi; optical; PC AT; PS/2; parallel; radio; video interface: Apple Desktop Connector (ADC), BNC, coaxial, component, composite, digital, DisplayPort, Digital Visual Interface (DVI), high-definition multimedia interface (HDMI), RCA, RF antennae, S-Video, VGA, and/or the like; wireless transceivers: 802.11a/b/g/n/x; Bluetooth; cellular (e.g., code division multiple access (CDMA), high speed packet access (HSPA(+)), high-speed downlink packet access (HSDPA), global system for mobile communications (GSM), long term evolution (LTE), WiMax, etc.); and/or the like. One output device may be a video display, which may take the form of a Cathode Ray Tube (CRT), Liquid Crystal Display (LCD), Light Emitting Diode (LED), Organic Light Emitting Diode (OLED), Plasma, and/or the like based monitor with an interface (e.g., VGA, DVI circuitry and cable) that accepts signals from a video interface. The video interface composites information generated by a computer systemization and generates video signals based on the composited information in a video memory frame. Another output device is a television set, which accepts signals from a video interface. Often, the video interface provides the composited video information through a video connection interface that accepts a video display interface (e.g., an RCA composite video connector accepting an RCA composite video cable; a DVI connector accepting a DVI display cable, HDMI, etc.).

[00161] User input devices 2011 often are a type of peripheral device 2012 (see below) and may include: card readers, dongles, finger print readers, gloves, graphics tablets, joysticks, keyboards, microphones, mouse (mice), remote controls, retina readers, touch screens (e.g., capacitive, resistive, etc.), trackballs, trackpads, sensors (e.g., accelerometers, ambient light, GPS, gyroscopes, proximity, etc.), styluses, and/or the like.

[00162] Peripheral devices 2012 may be connected and/or communicate to I/O and/or other facilities of the like such as network interfaces, storage interfaces, directly to the interface bus, system bus, the CPU, and/or the like. Peripheral devices may be external, internal and/or part of the EWCP controller. Peripheral devices may include: antenna, audio devices (e.g., line-in, line-out, microphone input, speakers, etc.), cameras (e.g., still, video, webcam, etc.), dongles (e.g., for copy protection, ensuring secure transactions with a digital signature, and/or the like), external processors (for added capabilities; e.g., crypto devices 2028), force-feedback devices (e.g., vibrating motors), near field communication (NFC) devices, network interfaces, printers, radio frequency identifiers (RFIDs), scanners, storage devices, transceivers (e.g., cellular, GPS, etc.), video devices (e.g., goggles, monitors, etc.), video sources, visors, and/or the like. Peripheral devices often include types of input devices (e.g., microphones, cameras, etc.).

[00163] It should be noted that although user input devices and peripheral devices may be employed, the EWCP controller may be embodied as an embedded, dedicated, and/or monitor-less (i.e., headless) device, wherein access would be provided over a network interface connection.

[00164] Cryptographic units such as, but not limited to, microcontrollers, processors 2026, interfaces 2027, and/or devices 2028 may be attached, and/or communicate with the EWCP controller. A MC68HC16 microcontroller, manufactured by Motorola Inc., may be used for and/or within cryptographic units. The MC68HC16 microcontroller utilizes a 16-bit multiply-and-accumulate instruction in the 16 MHz configuration and requires less than one second to perform a 512-bit RSA private key operation. Cryptographic units support the authentication of communications from interacting agents, as well as allowing for anonymous transactions. Cryptographic units may also be configured as part of the CPU. Equivalent microcontrollers and/or processors may also be used. Other commercially available specialized cryptographic processors include: the Broadcom's CryptoNetX and other Security Processors; nCipher's nShield (e.g., Solo, Connect, etc.), SafeNet's Luna PCI (e.g., 7100) series; Semaphore Communications' 40 MHz Roadrunner 184; sMIP's (e.g., 208956); Sun's Cryptographic Accelerators (e.g., Accelerator 6000 PCIe Board, Accelerator 500 Daughtercard); Via Nano Processor (e.g., L2100, L2200, U2400) line, which is capable of performing 500+ MB/s of cryptographic instructions; VLSI Technology's 33 MHz 6868; and/or the like.

Memory

[00165] Generally, any mechanization and/or embodiment allowing a processor to affect the storage and/or retrieval of information is regarded as memory 2029. However, memory is a fungible technology and resource, thus, any number of memory embodiments may be employed in lieu of or in concert with one another. It is to be understood that the EWCP controller and/or a computer systemization may employ various forms of memory 2029. For example, a computer systemization may be configured wherein the operation of on-chip CPU memory (e.g., registers), RAM, ROM, and any other storage devices are provided by a paper punch tape or paper punch card mechanism; however, such an embodiment would result in an extremely slow rate of operation. In one configuration, memory 2029 may include ROM 2006, RAM 2005, and a storage device 2014. A storage device 2014 may employ any number of computer storage devices/systems. Storage devices may include a drum; a (fixed and/or removable) magnetic disk drive; a magneto-optical drive; an optical drive (i.e., Blueray, CD ROM/RAM/Recordable (R)/ReWritable (RW), DVD R/RW, HD DVD R/RW etc.); an array of devices (e.g., Redundant Array of Independent Disks (RAID)); solid state memory devices (USB memory, solid state drives (SSD), etc.); other processor-readable storage mediums; and/or other devices of the like. Thus, a computer systemization generally requires and makes use of memory.

Component Collection

[00166] The memory 2029 may contain a collection of program and/or database components and/or data such as, but not limited to: operating system component(s) 2015 (operating system); information server component(s) 2016 (information server); user interface component(s) 2017 (user interface); Web browser component(s) 2018 (Web browser); database(s) 2019; mail server component(s) 2021; mail client component(s) 2022; cryptographic server component(s) 2020 (cryptographic server); the EWCP component(s) 2035; and/or the like (i.e., collectively a component collection). These components may be stored and accessed from the storage devices and/or from storage devices accessible through an interface bus. Although nonconventional program components such as those in the component collection may be stored in a local storage device 2014, they may also be loaded and/or stored in memory such as: peripheral devices, RAM, remote storage facilities through a communications network, ROM, various forms of memory, and/or the like.

Operating System

[00167] The operating system component 2015 is an executable program component facilitating the operation of the EWCP controller. The operating system may facilitate access of I/O, network interfaces, peripheral devices, storage devices, and/or the like. The operating system may be a highly fault tolerant, scalable, and secure system such as: Apple Macintosh OS X (Server); AT&T Plan 9; Be OS; Unix and Unix-like system distributions (such as AT&T's UNIX; Berkley Software Distribution (BSD) variations such as FreeBSD, NetBSD, OpenBSD, and/or the like; Linux distributions such as Red Hat, Ubuntu, and/or the like); and/or the like operating systems. However, more limited and/or less secure operating systems also may be employed such as Apple Macintosh OS, IBM OS/2, Microsoft DOS, Microsoft Windows 2000/2003/3.1/95/98/CE/Millenium/NT/Vista/XP (Server), Palm OS, and/or the like. In addition, emobile operating systems such as Apple's iOS, Google's Android, Hewlett Packard's WebOS, Microsofts Windows Mobile, and/or the like may be employed. Any of these operating systems may be embedded within the hardware of the NICK controller, and/or stored/loaded into memory/storage. An operating system may communicate to and/or with other components in a component collection, including itself, and/or the like. Most frequently, the operating system communicates with other program components, user interfaces, and/or the like. For example, the operating system may contain, communicate, generate, obtain, and/or provide program component, system, user, and/or data communications, requests, and/or responses. The operating system, once executed by the CPU, may enable the interaction with communications networks, data, I/O, peripheral devices, program components, memory, user input devices, and/or the like. The operating system may provide communications protocols that allow the EWCP controller to communicate with other entities through a communications network 2013. Various communication protocols may be used by the EWCP controller as a subcarrier transport mechanism for interaction, such as, but not limited to: multicast, TCP/IP, UDP, unicast, and/or the like.

Information Server

[00168] An information server component 2016 is a stored program component that is executed by a CPU. The information server may be an Internet information server such as, but not limited to Apache Software Foundation's Apache, Microsoft's Internet Information Server, and/or the like. The information server may allow for the execution of program components through facilities such as Active Server Page (ASP), ActiveX, (ANSI) (Objective-) C (++), C# and/or .NET, Common Gateway Interface (CGI) scripts, dynamic (D) hypertext markup language (HTML), FLASH, Java, JavaScript, Practical Extraction Report Language (PERL), Hypertext Pre-Processor (PHP), pipes, Python, wireless application protocol (WAP), WebObjects, and/or the like. The information server may support secure communications protocols such as, but not limited to, File Transfer Protocol (FTP); HyperText Transfer Protocol (HTTP); Secure Hypertext Transfer Protocol (HTTPS), Secure Socket Layer (SSL), messaging protocols (e.g., America Online (AOL) Instant Messenger (AIM), Apple's iMessage, Application Exchange (APEX), ICQ, Internet Relay Chat (IRC), Microsoft Network (MSN) Messenger Service, Presence and Instant Messaging Protocol (PRIM), Internet Engineering Task Force's (IETF's) Session Initiation Protocol (SIP), SIP for Instant Messaging and Presence Leveraging Extensions (SIMPLE), open XML-based Extensible Messaging and Presence Protocol (XMPP) (i.e., Jabber or Open Mobile Alliance's (OMA's) Instant Messaging and Presence Service (IMPS)), Yahoo! Instant Messenger Service, and/or the like. The information server provides results in the form of Web pages to Web browsers, and allows for the manipulated generation of the Web pages through interaction with other program components. After a Domain Name System (DNS) resolution portion of an HTTP request is resolved to a particular information server, the information server resolves requests for information at specified locations on the EWCP controller based on the remainder of the HTTP request. For example, a request such as http://123.124.125.126/myInformation.html might have the IP portion of the request "123.124.125.126" resolved by a DNS server to an information server at that IP address; that information server might in turn further parse the http request for the "/myInformation.html" portion of the request and resolve it to a location in memory containing the information "myInformation.html." Additionally, other information serving protocols may be employed across various ports, e.g., FTP communications across port 21, and/or the like. An information server may communicate to and/or with other components in a component collection, including itself, and/or facilities of the like. Most frequently, the information server communicates with the EWCP database 2019, operating systems, other program components, user interfaces, Web browsers, and/or the like.

[00169] Access to the EWCP database may be achieved through a number of database bridge mechanisms such as through scripting languages as enumerated below (e.g., CGI) and through inter-application communication channels as enumerated below (e.g., CORBA, WebObjects, etc.). Any data requests through a Web browser are parsed through the bridge mechanism into appropriate grammars as required by the EWCP. In one embodiment, the information server would provide a Web form accessible by a Web browser. Entries made into supplied fields in the Web form are tagged as having been entered into the particular fields, and parsed as such. The entered terms are then passed along with the field tags, which act to instruct the parser to generate queries directed to appropriate tables and/or fields. In one embodiment, the parser may generate queries in standard SQL by instantiating a search string with the proper join/select commands based on the tagged text entries, wherein the resulting command is provided over the bridge mechanism to the EWCP as a query. Upon generating query results from the query, the results are passed over the bridge mechanism, and may be parsed for formatting and generation of a new results Web page by the bridge mechanism. Such a new results Web page is then provided to the information server, which may supply it to the requesting Web browser.

[00170] Also, an information server may contain, communicate, generate, obtain, and/or provide program component, system, user, and/or data communications, requests, and/or responses.

User Interface

[00171] Computer interfaces in some respects are similar to automobile operation interfaces. Automobile operation interface elements such as steering wheels, gearshifts, and speedometers facilitate the access, operation, and display of automobile resources, and status. Computer interaction interface elements such as check boxes, cursors, menus, scrollers, and windows (collectively and commonly referred to as widgets) similarly facilitate the access, capabilities, operation, and display of data and computer hardware and operating system resources, and status. Operation interfaces are commonly called user interfaces. Graphical user interfaces (GUIs) such as the Apple Macintosh Operating System's Aqua and iOS's Cocoa Touch, IBM's OS/2, Google's Android Mobile UI, Microsoft's Windows 2000/2003/3.1/95/98/CE/Millenium/ Mobile/NT/XP/Vista/7/8 (i.e., Aero, Metro), Unix's X-Windows (e.g., which may include additional Unix graphic interface libraries and layers such as K Desktop Environment (KDE), mythTV and GNU Network Object Model Environment (GNOME)), web interface libraries (e.g., ActiveX, AJAX, (D)HTML, FLASH, Java, JavaScript, etc. interface libraries such as, but not limited to, Dojo, jQuery(UI), MooTools, Prototype, script.aculo.us, SWFObject, Yahoo! User Interface, any of which may be used and) provide a baseline and means of accessing and displaying information graphically to users.

[00172] A user interface component 2017 is a stored program component that is executed by a CPU. The user interface may be a graphic user interface as provided by, with, and/or atop operating systems and/or operating environments such as already discussed. The user interface may allow for the display, execution, interaction, manipulation, and/or operation of program components and/or system facilities through textual and/or graphical facilities. The user interface provides a facility through which users may affect, interact, and/or operate a computer system. A user interface may communicate to and/or with other components in a component collection, including itself, and/or facilities of the like. Most frequently, the user interface communicates with operating systems, other program components, and/or the like. The user interface may contain, communicate, generate, obtain, and/or provide program component, system, user, and/or data communications, requests, and/or responses.

Web Browser

[00173] A Web browser component 2018 is a stored program component that is executed by a CPU. The Web browser may be a hypertext viewing application such as Goofle's (Mobile) Chrome, Microsoft Internet Explorer, Netscape Navigator, Apple's (Mobile) Safari, embedded web browser objects such as through Apple's Cocoa (Touch) object class, and/or the like. Secure Web browsing may be supplied with 128bit (or greater) encryption by way of HTTPS, SSL, and/or the like. Web browsers allowing for the execution of program components through facilities such as ActiveX, AJAX, (D)HTML, FLASH, Java, JavaScript, web browser plug-in APIs (e.g., Chrome, FireFox, Internet Explorer, Safari Plug-in, and/or the like APIs), and/or the like. Web browsers and like information access tools may be integrated into PDAs, cellular telephones, smartphones, and/or other mobile devices. A Web browser may communicate to and/or with other components in a component collection, including itself, and/or facilities of the like. Most frequently, the Web browser communicates with information servers, operating systems, integrated program components (e.g., plug-ins), and/or the like; e.g., it may contain, communicate, generate, obtain, and/or provide program component, system, user, and/or data communications, requests, and/or responses. Also, in place of a Web browser and information server, a combined application may be developed to perform similar operations of both. The combined application would similarly effect the obtaining and the provision of information to users, user agents, and/or the like from the EWCP equipped nodes. The combined application may be nugatory on systems employing standard Web browsers.

Mail Server

[00174] A mail server component 2021 is a stored program component that is executed by a CPU 2003. The mail server may be an Internet mail server such as, but not limited to Apple's Mail Server (3), dovect, sendmail, Microsoft Exchange, and/or the like. The mail server may allow for the execution of program components through facilities such as ASP, ActiveX, (ANSI) (Objective-) C (++), C# and/or .NET, CGI scripts, Java, JavaScript, PERL, PHP, pipes, Python, WebObjects, and/or the like. The mail server may support communications protocols such as, but not limited to: Internet message access protocol (IMAP), Messaging Application Programming Interface (MAPI)/Microsoft Exchange, post office protocol (POP3), simple mail transfer protocol (SMTP), and/or the like. The mail server can route, forward, and process incoming and outgoing mail messages that have been sent, relayed and/or otherwise traversing through and/or to the EWCP.

[00175] Access to the EWCP mail may be achieved through a number of APIs offered by the individual Web server components and/or the operating system.

[00176] Also, a mail server may contain, communicate, generate, obtain, and/or provide program component, system, user, and/or data communications, requests, information, and/or responses.

Mail Client

[00177] A mail client component 2022 is a stored program component that is executed by a CPU 2003. The mail client may be a mail viewing application such as Apple (Mobile) Mail, Microsoft Entourage, Microsoft Outlook, Microsoft Outlook Express, Mozilla, Thunderbird, and/or the like. Mail clients may support a number of transfer protocols, such as: IMAP, Microsoft Exchange, POP3, SMTP, and/or the like. A mail client may communicate to and/or with other components in a component collection, including itself, and/or facilities of the like. Most frequently, the mail client communicates with mail servers, operating systems, other mail clients, and/or the like; e.g., it may contain, communicate, generate, obtain, and/or provide program component, system, user, and/or data communications, requests, information, and/or responses. Generally, the mail client provides a facility to compose and transmit electronic mail messages.

Cryptographic Server

[00178] A cryptographic server component 2020 is a stored program component that is executed by a CPU 2003, cryptographic processor 2026, cryptographic processor interface 2027, cryptographic processor device 2028, and/or the like. Cryptographic processor interfaces will allow for expedition of encryption and/or decryption requests by the cryptographic component; however, the cryptographic component, alternatively, may run on a CPU. The cryptographic component allows for the encryption and/or decryption of provided data. The cryptographic component allows for both symmetric and asymmetric (e.g., Pretty Good Protection (PGP)) encryption and/or decryption. The cryptographic component may employ cryptographic techniques such as, but not limited to: digital certificates (e.g., X.509 authentication framework), digital signatures, dual signatures, enveloping, password access protection, public key management, and/or the like. The cryptographic component will facilitate numerous (encryption and/or decryption) security protocols such as, but not limited to: checksum, Data Encryption Standard (DES), Elliptical Curve Encryption (ECC), International Data Encryption Algorithm (IDEA), Message Digest 5 (MD5, which is a one way hash operation), passwords, Rivest Cipher (RC5), Rijndael, RSA (which is an Internet encryption and authentication system that uses an algorithm developed in 1977 by Ron Rivest, Adi Shamir, and Leonard Adleman), Secure Hash Algorithm (SHA), Secure Socket Layer (SSL), Secure Hypertext Transfer Protocol (HTTPS), and/or the like. Employing such encryption security protocols, the EWCP may encrypt all incoming and/or outgoing communications and may serve as node within a virtual private network (VPN) with a wider communications network. The cryptographic component facilitates the process of "security authorization" whereby access to a resource is inhibited by a security protocol wherein the cryptographic component effects authorized access to the secured resource. In addition, the cryptographic component may provide unique identifiers of content, e.g., employing and MD5 hash to obtain a unique signature for an digital audio file. A cryptographic component may communicate to and/or with other components in a component collection, including itself, and/or facilities of the like. The cryptographic component supports encryption schemes allowing for the secure transmission of information across a communications network to enable the EWCP component to engage in secure transactions if so desired. The cryptographic component facilitates the secure accessing of resources on the EWCP and facilitates the access of secured resources on remote systems; i.e., it may act as a client and/or server of secured resources. Most frequently, the cryptographic component communicates with information servers, operating systems, other program components, and/or the like. The cryptographic component may contain, communicate, generate, obtain, and/or provide program component, system, user, and/or data communications, requests, and/or responses.

The EWCP Database

[00179] The EWCP database component 2019 may be embodied in a database and its stored data. The database is a stored program component, which is executed by the CPU; the stored program component portion configuring the CPU to process the stored data. The database may be any of a number of fault tolerant, relational, scalable, secure databases, such as DB2, MySQL, Oracle, Sybase, and/or the like. Relational databases are an extension of a flat file. Relational databases consist of a series of related tables. The tables are interconnected via a key field. Use of the key field allows the combination of the tables by indexing against the key field; i.e., the key fields act as dimensional pivot points for combining information from various tables. Relationships generally identify links maintained between tables by matching primary keys. Primary keys represent fields that uniquely identify the rows of a table in a relational database. More precisely, they uniquely identify rows of a table on the “one” side of a one-to-many relationship.

[00180] Alternatively, the EWCP database may be implemented using various standard data-structures, such as an array, hash, (linked) list, struct, structured text file (e.g., XML), table, and/or the like. Such data-structures may be stored in memory and/or in (structured) files. In another alternative, an object-oriented database may be used, such as Frontier, ObjectStore, Poet, Zope, and/or the like. Object databases can include a number of object collections that are grouped and/or linked together by common attributes; they may be related to other object collections by some common attributes. Object-oriented databases perform similarly to relational databases with the exception that objects are not just pieces of data but may have other types of capabilities encapsulated within a given object. If the EWCP database is implemented as a data-structure, the use of the EWCP database 2019 may be integrated into another component such as the EWCP component 2035. Also, the database may be implemented as a mix of data structures, objects, and relational structures. Databases may be consolidated and/or distributed in countless variations through standard data processing techniques. Portions of databases, e.g., tables, may be exported and/or imported and thus decentralized and/or integrated.

[00181] In one embodiment, the database component 2019 includes several tables 2019a-p. A Users table 2019a may include fields such as, but not limited to: user_id, ssn, dob, first_name, last_name, age, state, address_firstline, address_secondline, zipcode, devices_list, contact_info, contact_type, alt_contact_info, alt_contact_type, and/or the like. The Users table may support and/or track multiple entity accounts on a EWCP. A Devices table 2019b may include fields such as, but not limited to: device_ID, device_name, device_IP, device_GPS, device_MAC, device_serial, device_ECID, device_UDID, device_browser, device_type, device_model, device_version, device_OS, device_apps_list, device_securekey, wallet_app_installed_ flag, and/or the like. An Apps table 2019c may include fields such as, but not limited to: app_ID, app_name, app_type, app_dependencies, app_access_code, user_pin, and/or the like. An Accounts table 2019d may include fields such as, but not limited to: account_number, account_security_code, account_name, issuer_acquirer_flag, issuer_name, acquirer_name, account_address, routing_number, access_API_call, linked_wallets_list, and/or the like. A Merchants table 2019e may include fields such as, but not limited to: merchant_id, merchant_name, merchant_address, store_id, ip_address, mac_address, auth_key, port_num, security_settings_list, and/or the like. An Issuers table 2019f may include fields such as, but not limited to: issuer_id, issuer_name, issuer_address, ip_address, mac_address, auth_key, port_num, security_settings_list, and/or the like. An Acquirers table 2019g may include fields such as, but not limited to: account_firstname, account_lastname, account_type, account_num, account_ balance_list, billingaddress_ line1, billingaddress_ line2, billing_zipcode, billing_state, shipping_preferences, shippingaddress_line1, shippingaddress_line2, shipping_ zipcode, shipping_state, and/or the like. A Pay Gateways table 2019h may include fields such as, but not limited to: gateway_ID, gateway_IP, gateway_MAC, gateway_secure_key, gateway_access_list, gateway_API_call_list, gateway_services_list, and/or the like. A Shop Sessions table 2019i may include fields such as, but not limited to: user_id, session_id, alerts_URL, timestamp, expiry_lapse, merchant_id, store_id, device_type, device_ID, device_IP, device_MAC, device_browser, device_serial, device_ECID, device_model, device_OS, wallet_app_installed, total_cost, cart_ID_list, product_params_list, social_flag, social_message, social_networks_list, coupon_lists, accounts_list, CVV2_lists, charge_ratio_list, charge_priority_list, value_exchange_symbols_list, bill_address, ship_address, cloak_flag, pay_mode, alerts_rules_list, and/or the like. A Transactions table 2019j may include fields such as, but not limited to: order_id, user_id, timestamp, transaction_cost, purchase_details_list, num_products, products_list, product_type, product_params_list, product_title, product_summary, quantity, user_id, client_id, client_ip, client_type, client_model, operating_system, os_version, app_installed_flag, user_id, account_firstname, account_lastname, account_type, account_num, account_priority_account_ratio, billingaddress_line1, billingaddress_line2, billing_zipcode, billing_state, shipping_preferences, shippingaddress_line1, shippingaddress_line2, shipping_ zipcode, shipping_state, merchant_id, merchant_name, merchant_auth_key, and/or the like. A Batches table 2019k may include fields such as, but not limited to: batch_id, transaction_id_list, timestamp_list, cleared_flag_list, clearance_trigger_ settings, and/or the like. A Ledgers table 2019l may include fields such as, but not limited to: request_id, timestamp, deposit_amount, batch_id, transaction_id, clear_flag, deposit_account, transaction_summary, payor_ name, payor_account, and/or the like. A Products table 2019m may include fields such as, but not limited to: product_ID, product_title, product_attributes_list, product_price, tax_info_list, related_products_ list, offers_list, discounts_list, rewards_list, merchants_list, merchant_availability_list, and/or the like. An Offers table 2019n may include fields such as, but not limited to: offer_ID, offer_title, offer_attributes_list, offer_price, offer_expiry, related_products_ list, discounts_list, rewards_list, merchants_list, merchant_availability_list, and/or the like. A Behavior Data table 2019o may include fields such as, but not limited to: user_id, timestamp, activity_type, activity_location, activity_attribute_list, activity_attribute_values_list, and/or the like. An Analytics table 2019p may include fields such as, but not limited to: report_id, user_id, report_type, report_algorithm_id, report_destination_address, and/or the like. In one embodiment, the EWCP database may interact with other database systems. For example, employing a distributed database system, queries and data access by search EWCP component may treat the combination of the EWCP database, an integrated data security layer database as a single database entity.

[00182] In one embodiment, user programs may contain various user interface primitives, which may serve to update the EWCP. Also, various accounts may require custom database tables depending upon the environments and the types of clients the EWCP may need to serve. It should be noted that any unique fields may be designated as a key field throughout. In an alternative embodiment, these tables have been decentralized into their own databases and their respective database controllers (i.e., individual database controllers for each of the above tables). Employing standard data processing techniques, one may further distribute the databases over several computer systemizations and/or storage devices. Similarly, configurations of the decentralized database controllers may be varied by consolidating and/or distributing the various database components 2019a-p. The EWCP may be configured to keep track of various settings, inputs, and parameters via database controllers.

[00183] The EWCP database may communicate to and/or with other components in a component collection, including itself, and/or facilities of the like. Most frequently, the EWCP database communicates with the EWCP component, other program components, and/or the like. The database may contain, retain, and provide information regarding other nodes and data.

The EWCPs

[00184] The EWCP component 2035 is a stored program component that is executed by a CPU. In one embodiment, the EWCP component incorporates any and/or all combinations of the aspects of the EWCP discussed in the previous figures. As such, the EWCP affects accessing, obtaining and the provision of information, services, transactions, and/or the like across various communications networks. The features and embodiments of the EWCP discussed herein increase network efficiency by reducing data transfer requirements the use of more efficient data structures and mechanisms for their transfer and storage. As a consequence, more data may be transferred in less time, and latencies with regard to transactions, are also reduced. In many cases, such reduction in storage, transfer time, bandwidth requirements, latencies, etc., will reduce the capacity and structural infrastructure requirements to support the EWCP's features and facilities, and in many cases reduce the costs, energy consumption/requirements, and extend the life of EWCP's underlying infrastructure; this has the added benefit of making the EWCP more reliable. Similarly, many of the features and mechanisms are designed to be easier for users to use and access, thereby broadening the audience that may enjoy/employ and exploit the feature sets of the EWCP; such ease of use also helps to increase the reliability of the EWCP. In addition, the feature sets include heightened security as noted via the Cryptographic components 2020, 2026, 2028 and throughout, making access to the features and data more reliable and secure.

[00185] The EWCP component may transform customer purchase requests triggering electronic wallet applications via EWCP components into electronic purchase confirmation and receipts, and/or the like and use of the EWCP. In one embodiment, the EWCP component 2035 takes inputs (e.g., customer purchase request 305; checkout request 711; product data 715; wallet access input 911; transaction authorization input 914; payment gateway address 918; payment network address 922; issuer server address(es) 925; funds authorization request(s) 926; user(s) account(s) data 928; batch data 1112;

payment network address 1116; issuer server address(es) 1124; individual payment request 1125; payment ledger, merchant account data 1131; and/or the like) etc., and transforms the inputs via various components (e.g., UPC 2041; PTA 2042; PTC 2043; EWCPT 2044; EWCPA 2045; EWC 2046; VWSI 2047; and/or the like), into outputs (e.g., customer purchase response 350; payment confirmation 340; checkout request message 713; checkout data 717; card authorization request 916, 923; funds authorization response(s) 930; transaction authorization response 932; batch append data 934; purchase receipt 935; batch clearance request 1114; batch payment request 1118; transaction data 1120; individual payment confirmation 1128, 1129; updated payment ledger, merchant account data 1133; and/or the like).

[00186] The EWCP 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, mapping tools, procedural and object oriented development tools, PERL, PHP, Python, shell scripts, SQL commands, web application server extensions, web development environments and libraries (e.g., Microsoft's ActiveX; Adobe AIR, FLEX & FLASH; AJAX; (D)HTML; Dojo, Java; JavaScript; jQuery(UI); MooTools; Prototype; script.aculo.us; Simple Object Access Protocol (SOAP); SWFObject; Yahoo! User Interface; and/or the like), WebObjects, and/or the like. In one embodiment, the EWCP server employs a cryptographic server to encrypt and decrypt communications. The EWCP component may communicate to and/or with other components in a component collection, including itself, and/or facilities of the like. Most frequently, the EWCP component communicates with the EWCP database, operating systems, other program components, and/or the like. The EWCP may contain, communicate, generate, obtain, and/or provide program component, system, user, and/or data communications, requests, and/or responses.

Distributed EWCPs

[00187] The structure and/or operation of any of the EWCP node controller components may be combined, consolidated, and/or distributed in any number of ways to facilitate development and/or deployment. Similarly, the component collection may be combined in any number of ways to facilitate deployment and/or development. To accomplish this, one may integrate the components into a common code base or in a facility that can dynamically load the components on demand in an integrated fashion.

[00188] The component collection may be consolidated and/or distributed in countless variations through standard data processing and/or development techniques. Multiple instances of any one of the program components in the program component collection may be instantiated on a single node, and/or across numerous nodes to improve performance through load-balancing and/or data-processing techniques. Furthermore, single instances may also be distributed across multiple controllers and/or storage devices; e.g., databases. All program component instances and controllers working in concert may do so through standard data processing communication techniques.

[00189] The configuration of the EWCP controller will depend on the context of system deployment. Factors such as, but not limited to, the budget, capacity, location, and/or use of the underlying hardware resources may affect deployment requirements and configuration. Regardless of if the configuration results in more consolidated and/or integrated program components, results in a more distributed series of program components, and/or results in some combination between a consolidated and distributed configuration, data may be communicated, obtained, and/or provided. Instances of components consolidated into a common code base from the program component collection may communicate, obtain, and/or provide data. This may be accomplished through intra-application data processing communication techniques such as, but not limited to: data referencing (e.g., pointers), internal messaging, object instance variable communication, shared memory space, variable passing, and/or the like.

[00190] If component collection components are discrete, separate, and/or external to one another, then communicating, obtaining, and/or providing data with and/or to other components may be accomplished through inter-application data processing communication techniques such as, but not limited to: Application Program Interfaces (API) information passage; (distributed) Component Object Model ((D)COM), (Distributed) Object Linking and Embedding ((D)OLE), and/or the like), Common Object Request Broker Architecture (CORBA), Jini local and remote application program interfaces, JavaScript Object Notation (JSON), Remote Method Invocation (RMI), SOAP, process pipes, shared files, and/or the like. Messages sent between discrete component components for inter-application communication or within memory spaces of a singular component for intra-application communication may be facilitated through the creation and parsing of a grammar. A grammar may be developed by using development tools such as lex, yacc, XML, and/or the like, which allow for grammar generation and parsing capabilities, which in turn may form the basis of communication messages within and between components.

[00191] For example, a grammar may be arranged to recognize the tokens of an HTTP post command, e.g.:

```
w3c -post http://... Value1
```

[00192] where Value1 is discerned as being a parameter because "http://" is part of the grammar syntax, and what follows is considered part of the post value. Similarly, with such a grammar, a variable "Value1" may be inserted into an "http://" post command and then sent. The grammar syntax itself may be presented as structured data that is interpreted and/or otherwise used to generate the parsing mechanism (e.g., a syntax description text file as processed by lex, yacc, etc.). Also, once the parsing mechanism is generated and/or instantiated, it itself may process and/or parse structured data such as, but not limited to: character (e.g., tab) delineated text, HTML, structured text streams, XML, and/or the like structured data. In another embodiment, inter-application data processing protocols themselves may have integrated and/or readily available parsers (e.g., JSON, SOAP, and/or like parsers) that may be employed to parse (e.g., communications) data. Further, the parsing grammar may be used beyond message parsing, but may also be used to parse: databases, data collections, data stores, structured data, and/or the like. Again, the desired configuration will depend upon the context, environment, and requirements of system deployment.

[00193] For example, in some implementations, the EWCP controller may be executing a PHP script implementing a Secure Sockets Layer ("SSL") socket server via the information server, which listens to incoming communications on a server port to which a client may send data, e.g., data encoded in JSON format. Upon identifying an incoming communication, the PHP script may read the incoming message from the client device, parse the received JSON-encoded text data to extract information from the JSON-encoded text data into PHP script variables, and store the data (e.g., client identifying information, etc.) and/or extracted information in a relational database accessible using the Structured Query Language ("SQL"). An exemplary listing, written substantially in the form of PHP/SQL commands, to accept JSON-encoded input data from a client device via a SSL connection, parse the data to extract variables, and store the data to a database, is provided below:

```
<?PHP
header('Content-Type: text/plain');

// set ip address and port to listen to for incoming data
$address = '192.168.0.100';
$port = 255;

// create a server-side SSL socket, listen for/accept incoming communication
$sock = socket_create(AF_INET, SOCK_STREAM, 0);
socket_bind($sock, $address, $port) or die('Could not bind to address');
socket_listen($sock);
$client = socket_accept($sock);

// read input data from client device in 1024 byte blocks until end of message
do {
    $input = "";
    $input = socket_read($client, 1024);
    $data .= $input;
} while($input != "");

// parse data to extract variables
$obj = json_decode($data, true);

// store input data in a database
mysql_connect("201.408.185.132",$DBserver,$password); // access database server
mysql_select("CLIENT_DB.SQL"); // select database to append
mysql_query("INSERT INTO UserTable (transmission)
VALUES ($data)"); // add data to UserTable table in  a CLIENT database
mysql_close("CLIENT_DB.SQL"); // close connection to database
```

```
?>
```

[00194] Also, the following resources may be used to provide example embodiments regarding SOAP parser implementation:

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

[00195] and other parser implementations:

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

[00196] all of which are hereby expressly incorporated by reference herein.

[00197] Non-limiting example embodiments highlighting numerous further advantageous aspects include:

1. An e-wallet checkout processor-implemented method embodiment, comprising:

receiving a merchant payment request;

determining a web-to-wallet transition payment protocol handler associated with the merchant payment request;

instantiating, via a processor, a wallet application via the payment protocol handler;

obtaining a payment method selection via the wallet application;

providing a transaction execution request for a transaction associated with the merchant payment request;

receiving a purchase response to the transaction execution request; and

outputting purchase response information derived from the received purchase response.

2. The method of embodiment 1, wherein the payment protocol handler includes a purchase transaction parameter.

3. The method of embodiment 2, wherein the purchase transaction parameter includes one of: a selected payment method; a user-selected merchandise; a merchandise price; an applicable discounts; and an offer.

4. The method of embodiment 1, further comprising:

receiving a merchant payment request resulting from a transaction initiated from an initiating webpage; and transistioning to the wallet application from the initiating webpage.

5. The method of embodiment 1, wherein the payment method selection is obtained by determining a default payment method.

6. The method of embodiment 1, wherein the payment method selection is an electronic wallet.

7. The method of embodiment 1, wherein the payment method selection is one of a credit card, a debit card, a gift card.

8. The method of embodiment 1, further comprising:

obtaining an authorization to use the payment method selection for the transaction.

9. The method of embodiment 1, wherein the wallet application is one of: a Java applet; an HTML application; and a Javascript application.

10. The method of embodiment 1, wherein the purchase response information comprises a receipt.

11. The method of embodiment 1, further comprising:

determining a discount associated with the payment method selection; and

calculating a purchase price for the transaction associated with the merchant payment request by applying the discount associated with the payment method selection.

12. The method of embodiment 1, further comprising:

displaying a default redirect page.

13. The method of embodiment 1, further comprising:

displaying a redirect page specified by a merchant associated with the merchant payment request.

14. The method of embodiment 1, further comprising:

obtaining a virtual merchant store injection package;

instantiating the virtual merchant store within the wallet application for display and interaction; and

wherein the transaction associated with the merchant payment request includes purchase of a merchandise displayed in the virtual merchant store.

15. An e-wallet checkout means embodiment, comprising means for:

receiving a merchant payment request;

determining a web-to-wallet transition payment protocol handler associated with the merchant payment request;

instantiating, via a processor, a wallet application via the payment protocol handler;

obtaining a payment method selection via the wallet application;

providing a transaction execution request for a transaction associated with the merchant payment request;

receiving a purchase response to the transaction execution request; and

outputting purchase response information derived from the received purchase response.

16. The means of embodiment 15, wherein the payment protocol handler includes a purchase transaction parameter.

17. The means of embodiment 16, wherein the purchase transaction parameter includes one of: a selected payment method; a user-selected merchandise; a merchandise price; an applicable discounts; and an offer.

18. The means of embodiment 15, further comprising means for:

receiving a merchant payment request resulting from a transaction initiated from an initiating webpage; and

transistioning to the wallet application from the initiating webpage.

19. The means of embodiment 15, wherein the payment method selection is obtained by determining a default payment method.

20. The means of embodiment 15, wherein the payment method selection is an electronic wallet.

21. The means of embodiment 15, wherein the payment method selection is one of a credit card, a debit card, a gift card.

22. The means of embodiment 15, further comprising means for:

obtaining an authorization to use the payment method selection for the transaction.

23. The means of embodiment 15, wherein the wallet application is one of: a Java applet; an HTML application; and a Javascript application.

24. The means of embodiment 15, wherein the purchase response information comprises a receipt.

25. The means of embodiment 15, further comprising means for:

determining a discount associated with the payment method selection; and

calculating a purchase price for the transaction associated with the merchant payment request by applying the discount associated with the payment method selection.

26. The means of embodiment 15, further comprising means for:

displaying a default redirect page.

27. The means of embodiment 15, further comprising means for:

displaying a redirect page specified by a merchant associated with the merchant payment request.

28. The means of embodiment 15, further comprising means for:

obtaining a virtual merchant store injection package;

instantiating the virtual merchant store within the wallet application for display and interaction; and

wherein the transaction associated with the merchant payment request includes purchase of a merchandise displayed in the virtual merchant store.

29. An e-wallet checkout apparatus embodiment, comprising:

a processor; and

a memory disposed in communication with the processor and storing processor-issuable instructions to:

receive a merchant payment request;

determine a web-to-wallet transition payment protocol handler associated with the merchant payment request;

instantiate, via the processor, a wallet application via the payment protocol handler;

obtain a payment method selection via the wallet application;

provide a transaction execution request for a transaction associated with the merchant payment request;

receive a purchase response to the transaction execution request; and

output purchase response information derived from the received purchase response.

30. The apparatus of embodiment 29, wherein the payment protocol handler includes a purchase transaction parameter.

31. The apparatus of embodiment 30, wherein the purchase transaction parameter includes one of: a selected payment method; a user-selected merchandise; a merchandise price; an applicable discounts; and an offer.

32. The apparatus of embodiment 29, the memory further storing instructions to:

receive a merchant payment request resulting from a transaction initiated from an initiating webpage; and

transistion to the wallet application from the initiating webpage.

33. The apparatus of embodiment 29, wherein the payment method selection is obtained by determining a default payment method.

34. The apparatus of embodiment 29, wherein the payment method selection is an electronic wallet.

35. The apparatus of embodiment 29, wherein the payment method selection is one of a credit card, a debit card, a gift card.

36. The apparatus of embodiment 29, the memory further storing instructions to:

obtain an authorization to use the payment method selection for the transaction.

37. The apparatus of embodiment 29, wherein the wallet application is one of: a Java applet; an HTML application; and a Javascript application.

38. The apparatus of embodiment 29, wherein the purchase response information comprises a receipt.

39. The apparatus of embodiment 29, the memory further storing instructions to:

determine a discount associated with the payment method selection; and

calculate a purchase price for the transaction associated with the merchant payment request by applying the discount associated with the payment method selection.

40. The apparatus of embodiment 29, the memory further storing instructions to:

display a default redirect page.

41. The apparatus of embodiment 29, the memory further storing instructions to:

display a redirect page specified by a merchant associated with the merchant payment request.

42. The apparatus of embodiment 29, the memory further storing instructions to:

obtain a virtual merchant store injection package;

instantiate the virtual merchant store within the wallet application for display and interaction; and

wherein the transaction associated with the merchant payment request includes purchase of a merchandise displayed in the virtual merchant store.

43. A computer-readable tangible medium embodiment storing computer-executable e-wallet checkout instructions to:

receive a merchant payment request;

determine a web-to-wallet transition payment protocol handler associated with the merchant payment request;

instantiate a wallet application via the payment protocol handler;

obtain a payment method selection via the wallet application;

provide a transaction execution request for a transaction associated with the merchant payment request;

receive a purchase response to the transaction execution request; and output purchase response information derived from the received purchase response.

44. The medium of embodiment 43, wherein the payment protocol handler includes a purchase transaction parameter.

45. The medium of embodiment 44, wherein the purchase transaction parameter includes one of: a selected payment method; a user-selected merchandise; a merchandise price; an applicable discounts; and an offer.

46. The medium of embodiment 43, further storing instructions to:

receive a merchant payment request resulting from a transaction initiated from an initiating webpage; and

transistion to the wallet application from the initiating webpage.

47. The medium of embodiment 43, wherein the payment method selection is obtained by determining a default payment method.

48. The medium of embodiment 43, wherein the payment method selection is an electronic wallet.

49. The medium of embodiment 43, wherein the payment method selection is one of a credit card, a debit card, a gift card.

50. The medium of embodiment 43, further storing instructions to:

obtain an authorization to use the payment method selection for the transaction.

51. The medium of embodiment 43, wherein the wallet application is one of: a Java applet; an HTML application; and a Javascript application.

52. The medium of embodiment 43, wherein the purchase response information comprises a receipt.

53. The medium of embodiment 43, further storing instructions to:

determine a discount associated with the payment method selection; and

calculate a purchase price for the transaction associated with the merchant payment request by applying the discount associated with the payment method selection.

54. The medium of embodiment 43, further storing instructions to:

display a default redirect page.

55. The medium of embodiment 43, further storing instructions to:

display a redirect page specified by a merchant associated with the merchant payment request.

56. The medium of embodiment 43, further storing instructions to:

obtain a virtual merchant store injection package;

instantiate the virtual merchant store within the wallet application for display and interaction; and

wherein the transaction associated with the merchant payment request includes purchase of a merchandise displayed in the virtual merchant store.

57. An e-wallet checkout platform processor-implemented method embodiment, comprising:

providing a webpage including a merchant payment request and a web-to-wallet transition payment protocol handler associated with the merchant payment request;

wherein the protocol handler is consumer-triggerable for instantiating a wallet application via the protocol handler;

obtaining a payment method selection in response to providing the webpage from an instantiated wallet application;

obtaining a transaction execution request for a transaction associated with the merchant payment request;

providing a purchase response to the transaction execution request.

58. The method of embodiment 57, wherein the transaction execution request is obtained at one of: a merchant server; an issuer server; an acquirer server; and a payment network server.

59. The method of embodiment 57, wherein the payment protocol handler includes a purchase transaction parameter.

60. The method of embodiment 59, wherein the purchase transaction parameter includes one of: a selected payment method; a user-selected merchandise; a merchandise price; an applicable discounts; and an offer.

61. The method of embodiment 57, further comprising:

providing a merchant payment request resulting from a transaction initiated from an initiating webpage; and

obtaining a response from a wallet application instantiated based on the provided webpage.

62. The method of embodiment 57, wherein the payment method selection is obtained by determining a default payment method.

63. The method of embodiment 57, wherein the payment method selection is an electronic wallet.

64. The method of embodiment 57, wherein the payment method selection is one of a credit card, a debit card, a gift card.

65. The method of embodiment 57, further comprising:

obtaining an authorization to use the payment method selection for the transaction.

66. The method of embodiment 57, wherein the provided purchase response to the transaction execution request includes a purchase receipt.

67. The method of embodiment 57, further comprising:

determining a discount associated with the payment method selection; and

calculating a purchase price for the transaction associated with the merchant payment request by applying the discount associated with the payment method selection.

68. The method of embodiment 57, further comprising:

providing a virtual merchant store injection package;

instantiating the virtual merchant store within the wallet application for display and interaction; and

wherein the transaction associated with the merchant payment request includes purchase of a merchandise displayed in the virtual merchant store.

69. An e-wallet checkout platform means embodiment, comprising means for:

providing a webpage including a merchant payment request and a web-to-wallet transition payment protocol handler associated with the merchant payment request;

wherein the protocol handler is consumer-triggerable for instantiating a wallet application via the protocol handler;

obtaining a payment method selection in response to providing the webpage from an instantiated wallet application;

obtaining a transaction execution request for a transaction associated with the merchant payment request;

providing a purchase response to the transaction execution request.

70. The means of embodiment 69, wherein the transaction execution request is obtained at one of: a merchant server; an issuer server; an acquirer server; and a payment network server.

71. The means of embodiment 69, wherein the payment protocol handler includes a purchase transaction parameter.

72. The means of embodiment 71, wherein the purchase transaction parameter includes one of: a selected payment method; a user-selected merchandise; a merchandise price; an applicable discounts; and an offer.

73. The means of embodiment 69, further comprising means for:

providing a merchant payment request resulting from a transaction initiated from an initiating webpage; and

obtaining a response from a wallet application instantiated based on the provided webpage.

74. The means of embodiment 69, wherein the payment method selection is obtained by determining a default payment method.

75. The means of embodiment 69, wherein the payment method selection is an electronic wallet.

76. The means of embodiment 69, wherein the payment method selection is one of a credit card, a debit card, a gift card.

77. The means of embodiment 69, further comprising means for:

obtaining an authorization to use the payment method selection for the transaction.

78. The means of embodiment 69, wherein the provided purchase response to the transaction execution request includes a purchase receipt.

79. The means of embodiment 69, further comprising means for:

determining a discount associated with the payment method selection; and

calculating a purchase price for the transaction associated with the merchant payment request by applying the discount associated with the payment method selection.

80. The means of embodiment 69, further comprising means for:

providing a virtual merchant store injection package;

instantiating the virtual merchant store within the wallet application for display and interaction; and

wherein the transaction associated with the merchant payment request includes purchase of a merchandise displayed in the virtual merchant store.

81. An e-wallet checkout platform system embodiment, comprising:

a processor; and

a memory disposed in communication with the processor and storing processor-issuable instructions to:

provide a webpage including a merchant payment request and a web-to-wallet transition payment protocol handler associated with the merchant payment request;

wherein the protocol handler is consumer-triggerable for instantiating a wallet application via the protocol handler;

obtain a payment method selection in response to providing the webpage from an instantiated wallet application;

obtain a transaction execution request for a transaction associated with the merchant payment request;

provide a purchase response to the transaction execution request.

82. The system of embodiment 81, wherein the transaction execution request is obtained at one of: a merchant server; an issuer server; an acquirer server; and a payment network server.

83. The system of embodiment 81, wherein the payment protocol handler includes a purchase transaction parameter.

84. The system of embodiment 83, wherein the purchase transaction parameter includes one of: a selected payment method; a user-selected merchandise; a merchandise price; an applicable discounts; and an offer.

85. The system of embodiment 81, the memory further storing instructions to:

provide a merchant payment request resulting from a transaction initiated from an initiating webpage; and

obtain a response from a wallet application instantiated based on the provided webpage.

86. The system of embodiment 81, wherein the payment method selection is obtained by determining a default payment method.

87. The system of embodiment 81, wherein the payment method selection is an electronic wallet.

88. The system of embodiment 81, wherein the payment method selection is one of a credit card, a debit card, a gift card.

89. The system of embodiment 81, the memory further storing instructions to:

obtain an authorization to use the payment method selection for the transaction.

90. The system of embodiment 81, wherein the provided purchase response to the transaction execution request includes a purchase receipt.

91. The system of embodiment 81, the memory further storing instructions to:

determine a discount associated with the payment method selection; and

calculate a purchase price for the transaction associated with the merchant payment request by applying the discount associated with the payment method selection.

92. The system of embodiment 81, the memory further storing instructions to:

provide a virtual merchant store injection package;

instantiate the virtual merchant store within the wallet application for display and interaction; and

wherein the transaction associated with the merchant payment request includes purchase of a merchandise displayed in the virtual merchant store.

93. A computer-readable tangible medium embodiment storing computer-executable e-wallet checkout platform instructions to:

provide a webpage including a merchant payment request and a web-to-wallet transition payment protocol handler associated with the merchant payment request;

wherein the protocol handler is consumer-triggerable for instantiating a wallet application via the protocol handler;

obtain a payment method selection in response to providing the webpage from an instantiated wallet application;

obtain a transaction execution request for a transaction associated with the merchant payment request;

provide a purchase response to the transaction execution request.

94. The medium of embodiment 93, wherein the transaction execution request is obtained at one of: a merchant server; an issuer server; an acquirer server; and a payment network server.

95. The medium of embodiment 93, wherein the payment protocol handler includes a purchase transaction parameter.

96. The medium of embodiment 95, wherein the purchase transaction parameter includes one of: a selected payment method; a user-selected merchandise; a merchandise price; an applicable discounts; and an offer.

97. The medium of embodiment 93, further storing instructions to:

provide a merchant payment request resulting from a transaction initiated from an initiating webpage; and

obtain a response from a wallet application instantiated based on the provided webpage.

98. The medium of embodiment 93, wherein the payment method selection is obtained by determining a default payment method.

99. The medium of embodiment 93, wherein the payment method selection is an electronic wallet.

100. The medium of embodiment 93, wherein the payment method selection is one of a credit card, a debit card, a gift card.

101. The medium of embodiment 93, further storing instructions to:

obtain an authorization to use the payment method selection for the transaction.

102. The medium of embodiment 93, wherein the provided purchase response to the transaction execution request includes a purchase receipt.

103. The medium of embodiment 93, further storing instructions to:

determine a discount associated with the payment method selection; and calculate a purchase price for the transaction associated with the merchant payment request by applying the discount associated with the payment method selection.

104. The medium of embodiment 93, further storing instructions to:

provide a virtual merchant store injection package;

instantiate the virtual merchant store within the wallet application for display and interaction; and

wherein the transaction associated with the merchant payment request includes purchase of a merchandise displayed in the virtual merchant store.

[00198] In order to address various issues and advance the art, the entirety of this application for ELECTRONIC WALLET CHECKOUT PLATFORM APPARATUSES, METHODS AND SYSTEMS (including the Cover Page, Title, Headings, Field, Background, Summary, Brief Description of the Drawings, Detailed Description, Claims, Abstract, FIGUREs, Appendices and/or otherwise) shows by way of illustration various example embodiments in which the claimed innovations may be practiced. The advantages and features of the application are of a representative sample of embodiments only, and are not exhaustive and/or exclusive. They are presented only to assist in understanding and teach the claimed principles. It should be understood that they are not representative of all claimed innovations. As such, certain aspects of the disclosure have not been discussed herein. That alternate embodiments may not have been presented for a specific portion of the innovations or that further undescribed alternate embodiments may be available for a portion is not to be considered a disclaimer of those alternate embodiments. It will be appreciated that many of those undescribed embodiments incorporate the same principles of the innovations and others are equivalent. Thus, it is to be understood that other embodiments may be utilized and functional, logical, operational, organizational, structural and/or topological modifications may be made without departing from the scope and/or spirit of the disclosure. As such, all examples and/or embodiments are deemed to be non-limiting throughout this disclosure. Also, no inference should be drawn regarding those embodiments discussed herein relative to those not discussed herein other than it is as such for purposes of reducing space and repetition. For instance, it is to be understood that the logical and/or topological structure of any combination of any data flow sequence(s), program components (a component collection), other components and/or any present feature sets as described in the figures and/or throughout are not limited to a fixed operating order and/or arrangement, but rather, any disclosed order is exemplary and all equivalents, regardless of order, are contemplated by the disclosure. Furthermore, it is to be understood that such features are not limited to serial execution, but rather, any number of threads, processes, processors, services, servers, and/or the like that may execute asynchronously, concurrently, in parallel, simultaneously, synchronously, and/or the like are also contemplated by the disclosure. As such, some of these features may be mutually contradictory, in that they cannot be simultaneously present in a single embodiment. Similarly, some features are applicable to one aspect of the innovations, and inapplicable to others. In addition, the disclosure includes other innovations not presently claimed. Applicant reserves all rights in those presently unclaimed innovations, including the right to claim such innovations, file additional applications, continuations, continuations-in-part, divisions, and/or the like thereof. As such, it should be understood that advantages, embodiments, examples, functional, features, logical, operational, organizational, structural, topological, and/or other aspects of the disclosure are not to be considered limitations on the disclosure as defined by the claims or limitations on equivalents to the claims. It is to be understood that, depending on the particular needs and/or characteristics of a EWCP individual and/or enterprise user, database configuration and/or relational model, data type, data transmission and/or network framework, syntax structure, and/or the like, various embodiments of the EWCP may be implemented that allow a great deal of flexibility and customization. For example, aspects of the EWCP may be adapted for facilitating donations, creating web-based operating systems, and/or the like. While various embodiments and discussions of the EWCP have been directed to electronic purchase transactions, however, it is to be understood that the embodiments described herein may be readily configured and/or customized for a wide variety of other applications and/or implementations.

CLAIMS

What is claimed is:

1. An e-wallet checkout processor-implemented method, comprising:

receiving a merchant payment request;

determining a web-to-wallet transition payment protocol handler associated with the merchant payment request;

instantiating, via a processor, a wallet application via the payment protocol handler;

obtaining a payment method selection via the wallet application;

providing a transaction execution request for a transaction associated with the merchant payment request;

receiving a purchase response to the transaction execution request; and

outputting purchase response information derived from the received purchase response.

2. The method of claim 1, wherein the payment protocol handler includes a purchase transaction parameter.

3. The method of claim 2, wherein the purchase transaction parameter includes one of: a selected payment method; a user-selected merchandise; a merchandise price; an applicable discounts; and an offer.

4. The method of claim 1, further comprising:

receiving a merchant payment request resulting from a transaction initiated from an initiating webpage; and

transistioning to the wallet application from the initiating webpage.

5. The method of claim 1, wherein the payment method selection is obtained by determining a default payment method.

6. The method of claim 1, wherein the payment method selection is an electronic wallet.

7. The method of claim 1, wherein the payment method selection is one of a credit card, a debit card, a gift card.

8. The method of claim 1, further comprising:

obtaining an authorization to use the payment method selection for the transaction.

9. The method of claim 1, wherein the wallet application is one of: a Java applet; an HTML application; and a Javascript application.

10. The method of claim 1, wherein the purchase response information comprises a receipt.

11. The method of claim 1, further comprising:

determining a discount associated with the payment method selection; and

calculating a purchase price for the transaction associated with the merchant payment request by applying the discount associated with the payment method selection.

12. The method of claim 1, further comprising:

displaying a default redirect page.

13. The method of claim 1, further comprising:

displaying a redirect page specified by a merchant associated with the merchant payment request.

14. The method of claim 1, further comprising:

obtaining a virtual merchant store injection package;

instantiating the virtual merchant store within the wallet application for display and interaction; and

wherein the transaction associated with the merchant payment request includes purchase of a merchandise displayed in the virtual merchant store.

15. An e-wallet checkout means, comprising means for:

receiving a merchant payment request;

determining a web-to-wallet transition payment protocol handler associated with the merchant payment request;

instantiating, via a processor, a wallet application via the payment protocol handler;

obtaining a payment method selection via the wallet application;

providing a transaction execution request for a transaction associated with the merchant payment request;

receiving a purchase response to the transaction execution request; and

outputting purchase response information derived from the received purchase response.

16. The means of claim 15, wherein the payment protocol handler includes a purchase transaction parameter.

17. The means of claim 16, wherein the purchase transaction parameter includes one of: a selected payment method; a user-selected merchandise; a merchandise price; an applicable discounts; and an offer.

18. The means of claim 15, further comprising means for:

receiving a merchant payment request resulting from a transaction initiated from an initiating webpage; and

transistioning to the wallet application from the initiating webpage.

19. The means of claim 15, wherein the payment method selection is obtained by determining a default payment method.

20. The means of claim 15, wherein the payment method selection is an electronic wallet.

21. The means of claim 15, wherein the payment method selection is one of a credit card, a debit card, a gift card.

22. The means of claim 15, further comprising means for:

obtaining an authorization to use the payment method selection for the transaction.

23. The means of claim 15, wherein the wallet application is one of: a Java applet; an HTML application; and a Javascript application.

24. The means of claim 15, wherein the purchase response information comprises a receipt.

25. The means of claim 15, further comprising means for:

determining a discount associated with the payment method selection; and

calculating a purchase price for the transaction associated with the merchant payment request by applying the discount associated with the payment method selection.

26. The means of claim 15, further comprising means for:

displaying a default redirect page.

27. The means of claim 15, further comprising means for:

displaying a redirect page specified by a merchant associated with the merchant payment request.

28. The means of claim 15, further comprising means for:

obtaining a virtual merchant store injection package;

instantiating the virtual merchant store within the wallet application for display and interaction; and

wherein the transaction associated with the merchant payment request includes purchase of a merchandise displayed in the virtual merchant store.

29. An e-wallet checkout apparatus, comprising:

a processor; and

a memory disposed in communication with the processor and storing processor-issuable instructions to:

receive a merchant payment request;

determine a web-to-wallet transition payment protocol handler associated with the merchant payment request;

instantiate, via the processor, a wallet application via the payment protocol handler;

obtain a payment method selection via the wallet application;

provide a transaction execution request for a transaction associated with the merchant payment request;

receive a purchase response to the transaction execution request; and

output purchase response information derived from the received purchase response.

30. The apparatus of claim 29, wherein the payment protocol handler includes a purchase transaction parameter.

31. The apparatus of claim 30, wherein the purchase transaction parameter includes one of: a selected payment method; a user-selected merchandise; a merchandise price; an applicable discounts; and an offer.

32. The apparatus of claim 29, the memory further storing instructions to:

receive a merchant payment request resulting from a transaction initiated from an initiating webpage; and

transistion to the wallet application from the initiating webpage.

33. The apparatus of claim 29, wherein the payment method selection is obtained by determining a default payment method.

34. The apparatus of claim 29, wherein the payment method selection is an electronic wallet.

35. The apparatus of claim 29, wherein the payment method selection is one of a credit card, a debit card, a gift card.

36. The apparatus of claim 29, the memory further storing instructions to:

obtain an authorization to use the payment method selection for the transaction.

37. The apparatus of claim 29, wherein the wallet application is one of: a Java applet; an HTML application; and a Javascript application.

38. The apparatus of claim 29, wherein the purchase response information comprises a receipt.

39. The apparatus of claim 29, the memory further storing instructions to:

determine a discount associated with the payment method selection; and calculate a purchase price for the transaction associated with the merchant payment request by applying the discount associated with the payment method selection.

40. The apparatus of claim 29, the memory further storing instructions to:

display a default redirect page.

41. The apparatus of claim 29, the memory further storing instructions to:

display a redirect page specified by a merchant associated with the merchant payment request.

42. The apparatus of claim 29, the memory further storing instructions to:

obtain a virtual merchant store injection package;

instantiate the virtual merchant store within the wallet application for display and interaction; and

wherein the transaction associated with the merchant payment request includes purchase of a merchandise displayed in the virtual merchant store.

43. A computer-readable tangible medium storing computer-executable e-wallet checkout instructions to:

receive a merchant payment request;

determine a web-to-wallet transition payment protocol handler associated with the merchant payment request;

instantiate a wallet application via the payment protocol handler;

obtain a payment method selection via the wallet application;

provide a transaction execution request for a transaction associated with the merchant payment request;

receive a purchase response to the transaction execution request; and

output purchase response information derived from the received purchase response.

44. The medium of claim 43, wherein the payment protocol handler includes a purchase transaction parameter.

45. The medium of claim 44, wherein the purchase transaction parameter includes one of: a selected payment method; a user-selected merchandise; a merchandise price; an applicable discounts; and an offer.

46. The medium of claim 43, further storing instructions to:

receive a merchant payment request resulting from a transaction initiated from an initiating webpage; and

transistion to the wallet application from the initiating webpage.

47. The medium of claim 43, wherein the payment method selection is obtained by determining a default payment method.

48. The medium of claim 43, wherein the payment method selection is an electronic wallet.

49. The medium of claim 43, wherein the payment method selection is one of a credit card, a debit card, a gift card.

50. The medium of claim 43, further storing instructions to:

obtain an authorization to use the payment method selection for the transaction.

51. The medium of claim 43, wherein the wallet application is one of: a Java applet; an HTML application; and a Javascript application.

52. The medium of claim 43, wherein the purchase response information comprises a receipt.

53. The medium of claim 43, further storing instructions to:

determine a discount associated with the payment method selection; and

calculate a purchase price for the transaction associated with the merchant payment request by applying the discount associated with the payment method selection.

54. The medium of claim 43, further storing instructions to:

display a default redirect page.

55. The medium of claim 43, further storing instructions to:

display a redirect page specified by a merchant associated with the merchant payment request.

56. The medium of claim 43, further storing instructions to:

obtain a virtual merchant store injection package;

instantiate the virtual merchant store within the wallet application for display and interaction; and

wherein the transaction associated with the merchant payment request includes purchase of a merchandise displayed in the virtual merchant store.

57. An e-wallet checkout platform processor-implemented method, comprising:

providing a webpage including a merchant payment request and a web-to-wallet transition payment protocol handler associated with the merchant payment request;

wherein the protocol handler is consumer-triggerable for instantiating a wallet application via the protocol handler;

obtaining a payment method selection in response to providing the webpage from an instantiated wallet application;

obtaining a transaction execution request for a transaction associated with the merchant payment request;

providing a purchase response to the transaction execution request.

58. The method of claim 57, wherein the transaction execution request is obtained at one of: a merchant server; an issuer server; an acquirer server; and a payment network server.

59. The method of claim 57, wherein the payment protocol handler includes a purchase transaction parameter.

60. The method of claim 59, wherein the purchase transaction parameter includes one of: a selected payment method; a user-selected merchandise; a merchandise price; an applicable discounts; and an offer.

61. The method of claim 57, further comprising:

providing a merchant payment request resulting from a transaction initiated from an initiating webpage; and

obtaining a response from a wallet application instantiated based on the provided webpage.

62. The method of claim 57, wherein the payment method selection is obtained by determining a default payment method.

63. The method of claim 57, wherein the payment method selection is an electronic wallet.

64. The method of claim 57, wherein the payment method selection is one of a credit card, a debit card, a gift card.

65. The method of claim 57, further comprising:

obtaining an authorization to use the payment method selection for the transaction.

66. The method of claim 57, wherein the provided purchase response to the transaction execution request includes a purchase receipt.

67. The method of claim 57, further comprising:

determining a discount associated with the payment method selection; and

calculating a purchase price for the transaction associated with the merchant payment request by applying the discount associated with the payment method selection.

68. The method of claim 57, further comprising:

providing a virtual merchant store injection package;

instantiating the virtual merchant store within the wallet application for display and interaction; and

wherein the transaction associated with the merchant payment request includes purchase of a merchandise displayed in the virtual merchant store.

69. An e-wallet checkout platform means, comprising means for:

providing a webpage including a merchant payment request and a web-to-wallet transition payment protocol handler associated with the merchant payment request;

wherein the protocol handler is consumer-triggerable for instantiating a wallet application via the protocol handler;

obtaining a payment method selection in response to providing the webpage from an instantiated wallet application;

obtaining a transaction execution request for a transaction associated with the merchant payment request;

providing a purchase response to the transaction execution request.

70. The means of claim 69, wherein the transaction execution request is obtained at one of: a merchant server; an issuer server; an acquirer server; and a payment network server.

71. The means of claim 69, wherein the payment protocol handler includes a purchase transaction parameter.

72. The means of claim 71, wherein the purchase transaction parameter includes one of: a selected payment method; a user-selected merchandise; a merchandise price; an applicable discounts; and an offer.

73. The means of claim 69, further comprising means for:

providing a merchant payment request resulting from a transaction initiated from an initiating webpage; and

obtaining a response from a wallet application instantiated based on the provided webpage.

74. The means of claim 69, wherein the payment method selection is obtained by determining a default payment method.

75. The means of claim 69, wherein the payment method selection is an electronic wallet.

76. The means of claim 69, wherein the payment method selection is one of a credit card, a debit card, a gift card.

77. The means of claim 69, further comprising means for:

obtaining an authorization to use the payment method selection for the transaction.

78. The means of claim 69, wherein the provided purchase response to the transaction execution request includes a purchase receipt.

79. The means of claim 69, further comprising means for:

determining a discount associated with the payment method selection; and

calculating a purchase price for the transaction associated with the merchant payment request by applying the discount associated with the payment method selection.

80. The means of claim 69, further comprising means for:

providing a virtual merchant store injection package;

instantiating the virtual merchant store within the wallet application for display and interaction; and

wherein the transaction associated with the merchant payment request includes purchase of a merchandise displayed in the virtual merchant store.

81. An e-wallet checkout platform system, comprising:

a processor; and

a memory disposed in communication with the processor and storing processor-issuable instructions to:

provide a webpage including a merchant payment request and a web-to-wallet transition payment protocol handler associated with the merchant payment request;

wherein the protocol handler is consumer-triggerable for instantiating a wallet application via the protocol handler;

obtain a payment method selection in response to providing the webpage from an instantiated wallet application;

obtain a transaction execution request for a transaction associated with the merchant payment request;

provide a purchase response to the transaction execution request.

82. The system of claim 81, wherein the transaction execution request is obtained at one of: a merchant server; an issuer server; an acquirer server; and a payment network server.

83. The system of claim 81, wherein the payment protocol handler includes a purchase transaction parameter.

84. The system of claim 83, wherein the purchase transaction parameter includes one of: a selected payment method; a user-selected merchandise; a merchandise price; an applicable discounts; and an offer.

85. The system of claim 81, the memory further storing instructions to:

provide a merchant payment request resulting from a transaction initiated from an initiating webpage; and

obtain a response from a wallet application instantiated based on the provided webpage.

86. The system of claim 81, wherein the payment method selection is obtained by determining a default payment method.

87. The system of claim 81, wherein the payment method selection is an electronic wallet.

88. The system of claim 81, wherein the payment method selection is one of a credit card, a debit card, a gift card.

89. The system of claim 81, the memory further storing instructions to:

obtain an authorization to use the payment method selection for the transaction.

90. The system of claim 81, wherein the provided purchase response to the transaction execution request includes a purchase receipt.

91. The system of claim 81, the memory further storing instructions to:

determine a discount associated with the payment method selection; and

calculate a purchase price for the transaction associated with the merchant payment request by applying the discount associated with the payment method selection.

92. The system of claim 81, the memory further storing instructions to:

provide a virtual merchant store injection package;

instantiate the virtual merchant store within the wallet application for display and interaction; and

wherein the transaction associated with the merchant payment request includes purchase of a merchandise displayed in the virtual merchant store.

93. A computer-readable tangible medium storing computer-executable e-wallet checkout platform instructions to:

provide a webpage including a merchant payment request and a web-to-wallet transition payment protocol handler associated with the merchant payment request;

wherein the protocol handler is consumer-triggerable for instantiating a wallet application via the protocol handler;

obtain a payment method selection in response to providing the webpage from an instantiated wallet application;

obtain a transaction execution request for a transaction associated with the merchant payment request;

provide a purchase response to the transaction execution request.

94. The medium of claim 93, wherein the transaction execution request is obtained at one of: a merchant server; an issuer server; an acquirer server; and a payment network server.

95. The medium of claim 93, wherein the payment protocol handler includes a purchase transaction parameter.

96. The medium of claim 95, wherein the purchase transaction parameter includes one of: a selected payment method; a user-selected merchandise; a merchandise price; an applicable discounts; and an offer.

97. The medium of claim 93, further storing instructions to:

provide a merchant payment request resulting from a transaction initiated from an initiating webpage; and

obtain a response from a wallet application instantiated based on the provided webpage.

98. The medium of claim 93, wherein the payment method selection is obtained by determining a default payment method.

99. The medium of claim 93, wherein the payment method selection is an electronic wallet.

100. The medium of claim 93, wherein the payment method selection is one of a credit card, a debit card, a gift card.

101. The medium of claim 93, further storing instructions to:

obtain an authorization to use the payment method selection for the transaction.

102. The medium of claim 93, wherein the provided purchase response to the transaction execution request includes a purchase receipt.

103. The medium of claim 93, further storing instructions to:

determine a discount associated with the payment method selection; and

calculate a purchase price for the transaction associated with the merchant payment request by applying the discount associated with the payment method selection.

104. The medium of claim 93, further storing instructions to:

provide a virtual merchant store injection package;

instantiate the virtual merchant store within the wallet application for display and interaction; and

wherein the transaction associated with the merchant payment request includes purchase of a merchandise displayed in the virtual merchant store.

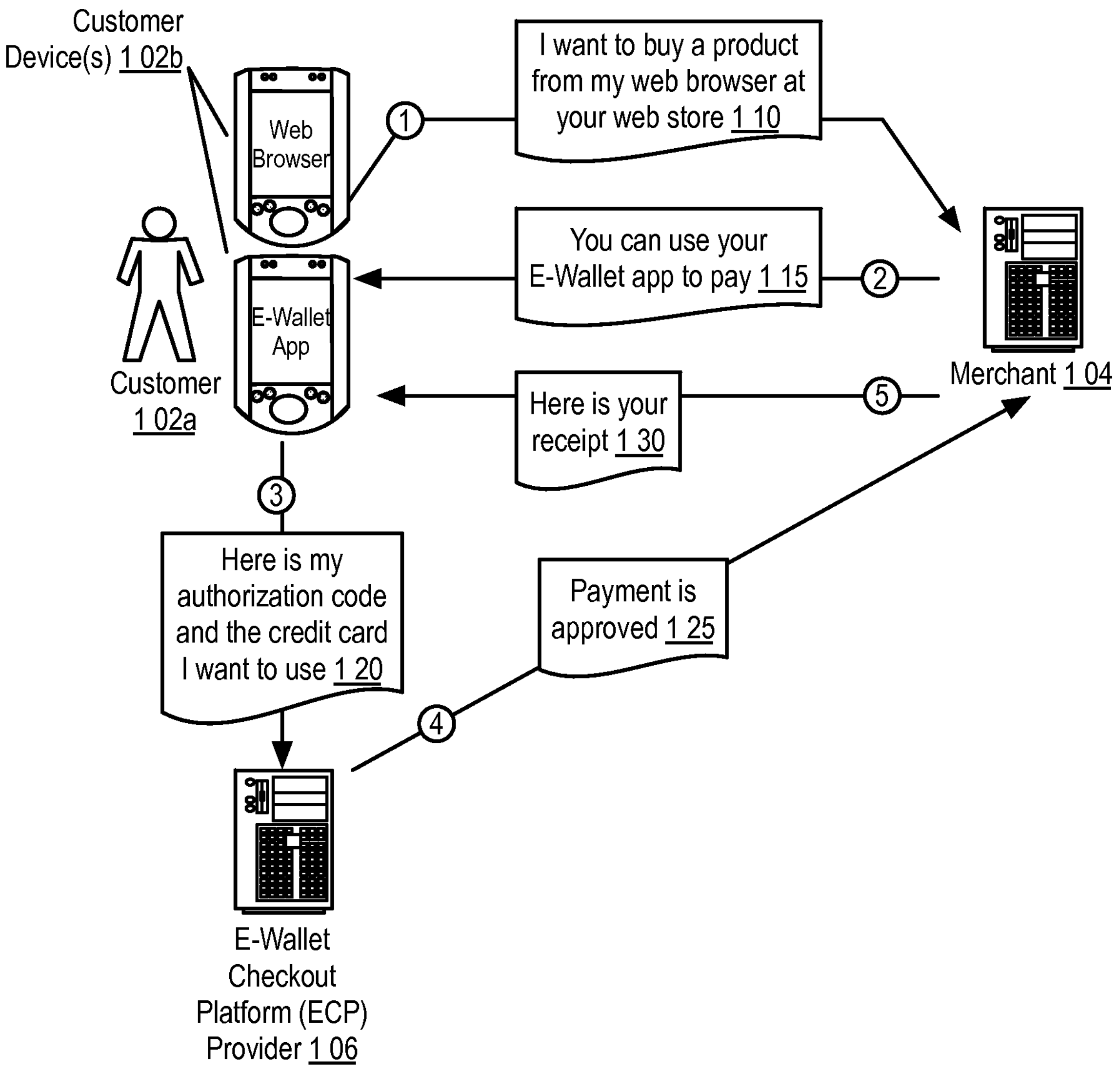
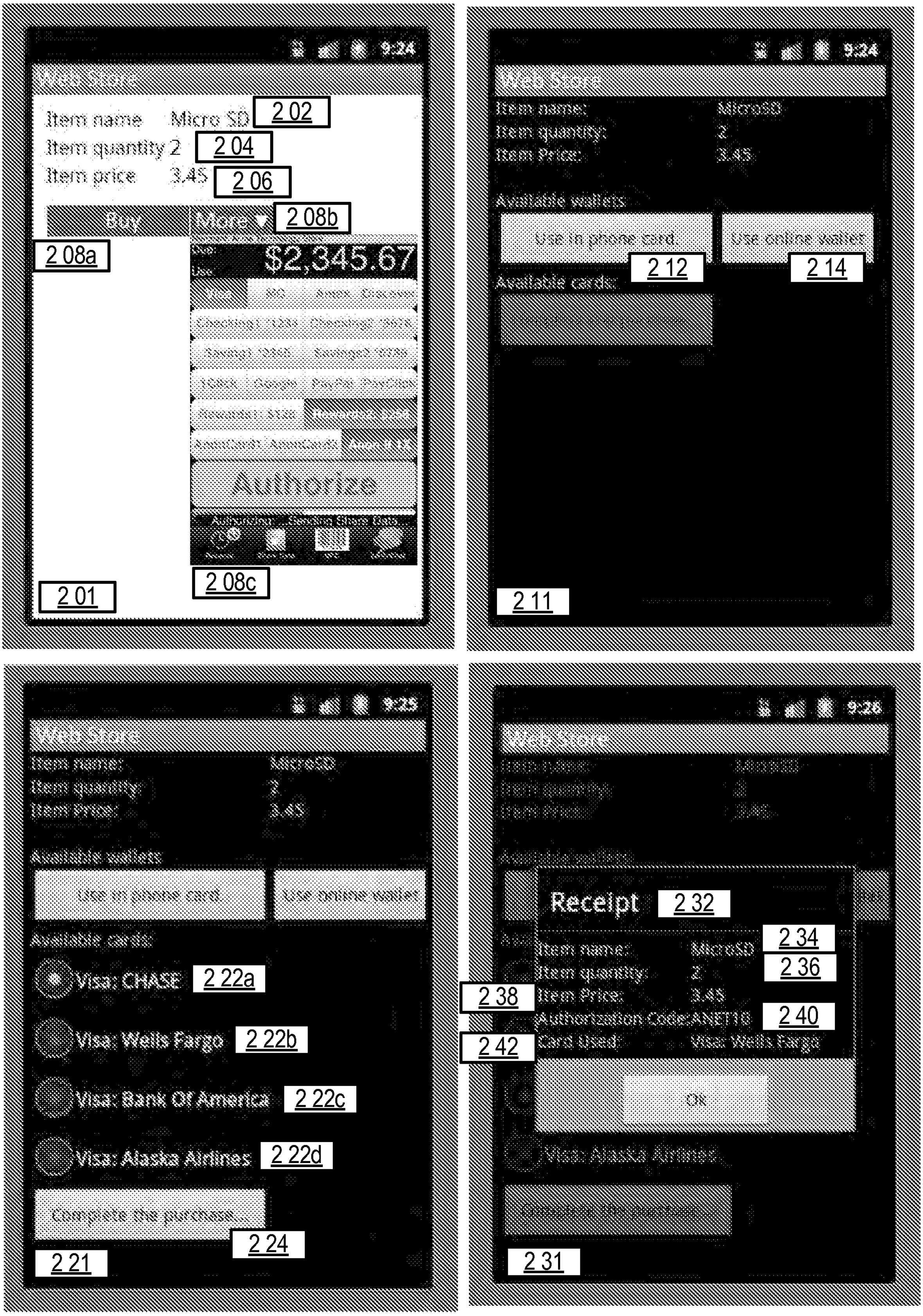

Customer
Device(s) 1 02b
Web
Browser
E-Wallet
App
Customer
1 02a
1
I want to buy a product from my web browser at your web store 1 10
2
You can use your E-Wallet app to pay 1 15
5
Here is your receipt 1 30
3
Here is my authorization code and the credit card I want to use 1 20
4
Payment is approved 1 25
Merchant 1 04
E-Wallet Checkout Platform (ECP) Provider 1 06

FIGURE 1

Example: E-Wallet Checkout Platform

FIGURE 2



9:24

Web Store

Item name Micro SD

Item quantity 2

Item price 3.45

Buy

More ▼

$2,345.67

Authorize

9:24

Web Store

Item name: MicroSD

Item quantity: 2

Item Price: 3.45

Available wallets:

Use in phone card

Use online wallet

Available cards:

9:25

Web Store

Item name: MicroSD

Item quantity: 2

Item Price: 3.45

Available wallets:

Use in phone card

Use online wallet

Available cards:

● Visa: CHASE

○ Visa: Wells Fargo

○ Visa: Bank Of America

○ Visa: Alaska Airlines

Complete the purchase

9:26

Web Store

Receipt

Item name: MicroSD

Item quantity: 2

Item Price: 3.45

Authorization Code:ANET10

Card Used: Visa: Wells Fargo

Ok

Visa: Alaska Airlines

Example: E-Wallet Checkout Mobile Application

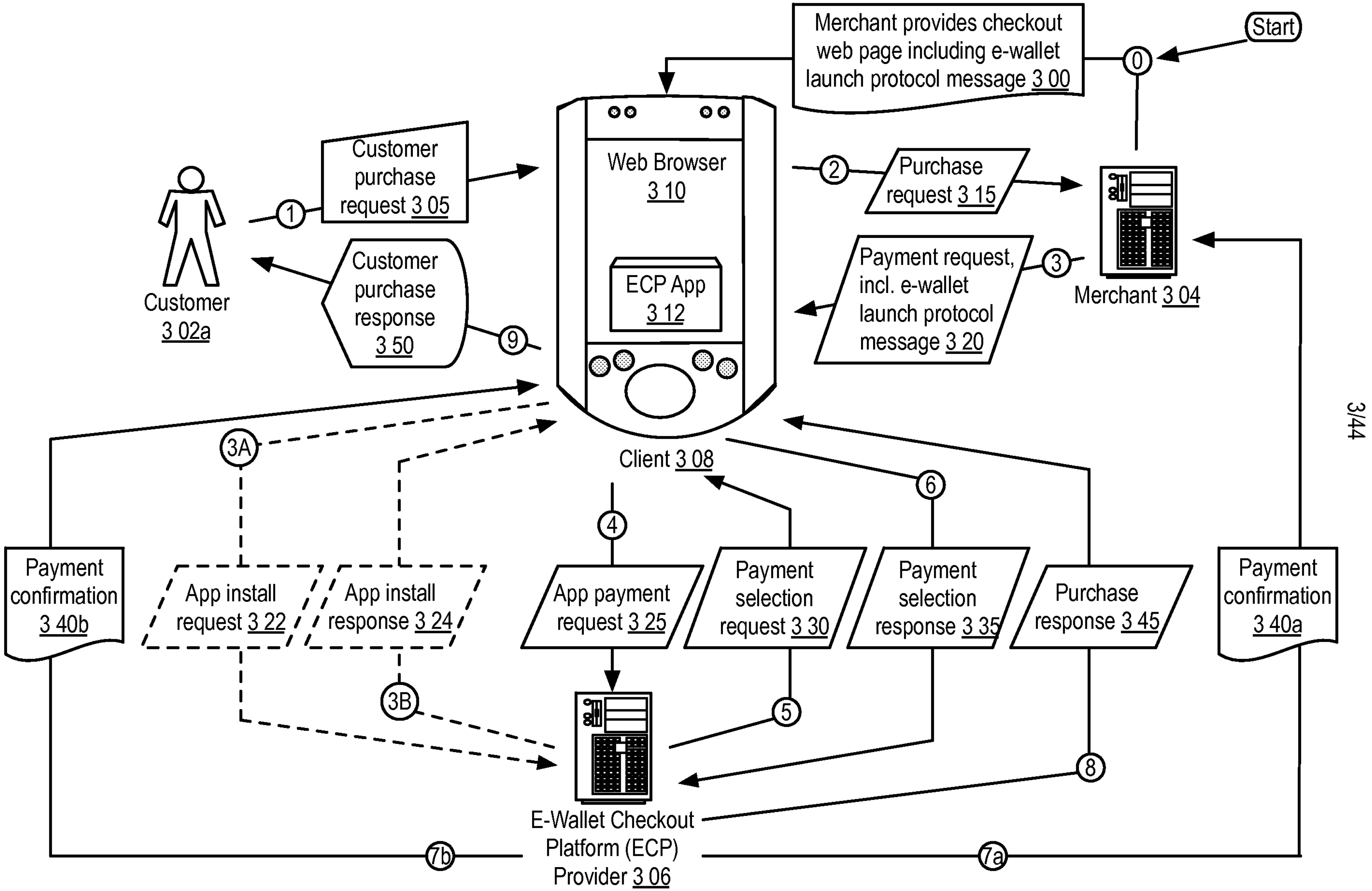
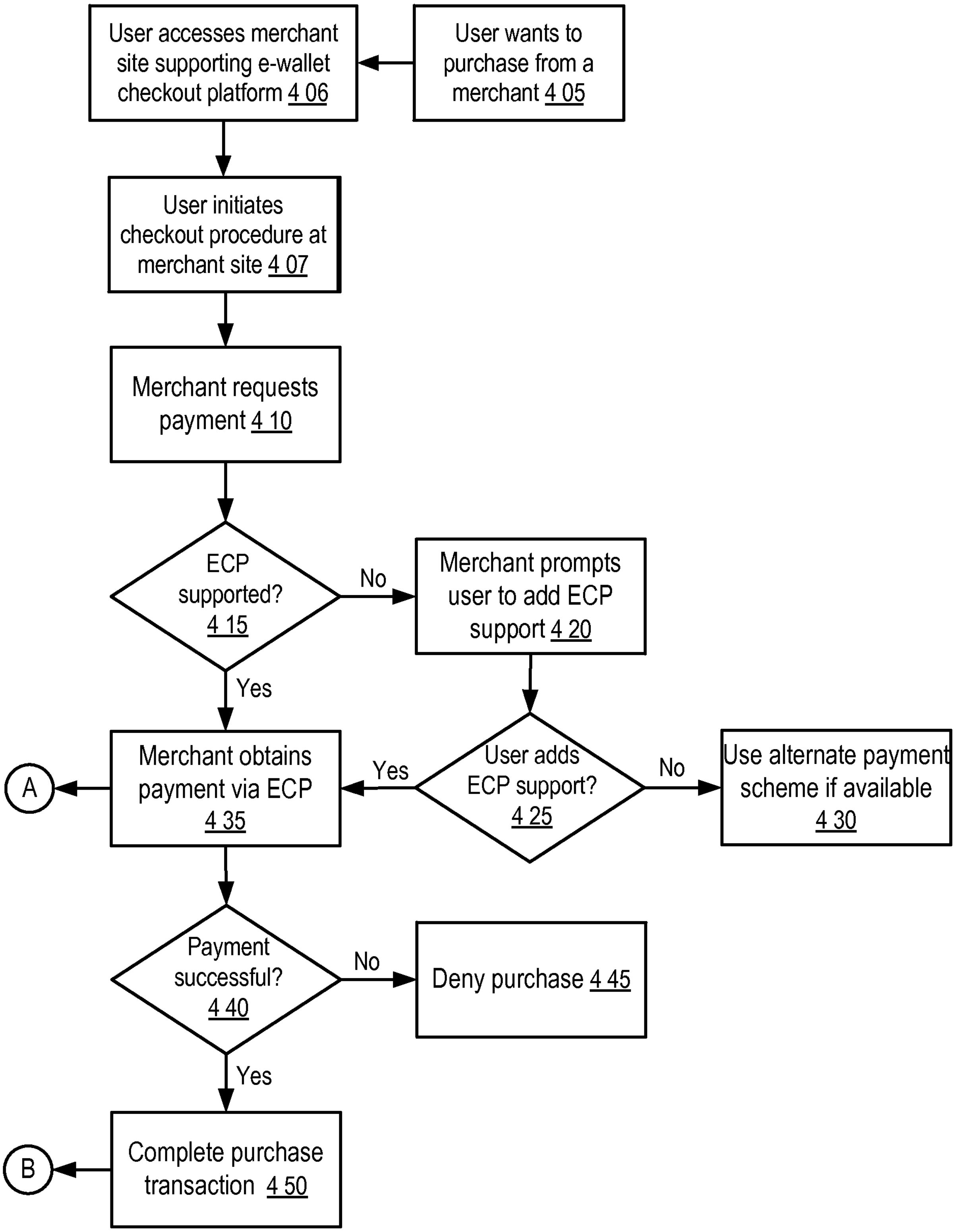

Start
0
Merchant provides checkout web page including e-wallet launch protocol message 3 00
Merchant 3 04
2
Purchase request 3 15
3
Payment request, incl. e-wallet launch protocol message 3 20
Web Browser 3 10
ECP App 3 12
Client 3 08
1
Customer purchase request 3 05
Customer 3 02a
Customer purchase response 3 50
9
3A
3B
App install request 3 22
App install response 3 24
4
App payment request 3 25
Payment selection request 3 30
5
6
Payment selection response 3 35
Purchase response 3 45
8
Payment confirmation 3 40a
Payment confirmation 3 40b
7a
7b
E-Wallet Checkout Platform (ECP) Provider 3 06

FIGURE 3

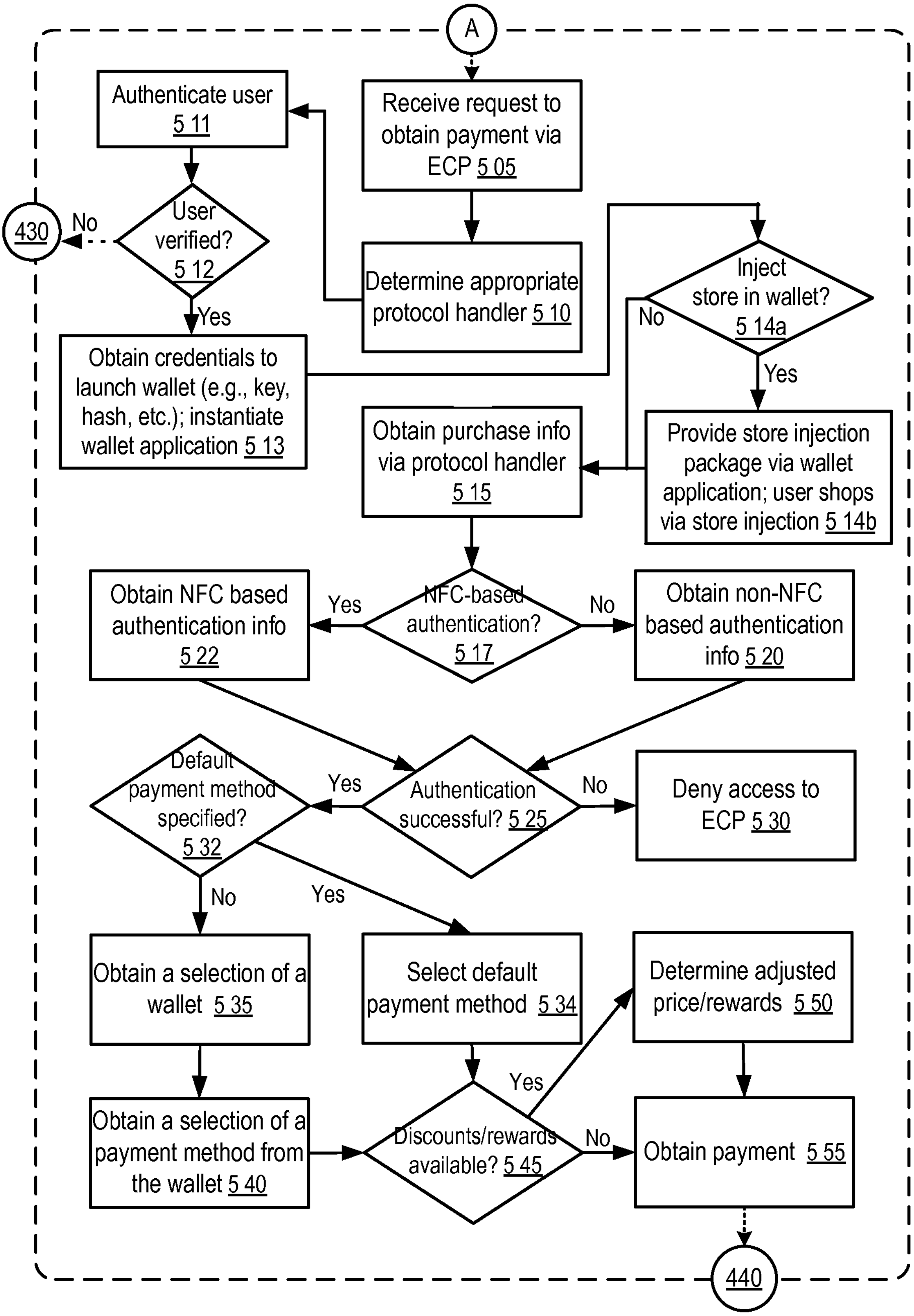
Example: E-Wallet Checkout Data Flow

FIGURE 4


User wants to purchase from a merchant 4 05
User accesses merchant site supporting e-wallet checkout platform 4 06
User initiates checkout procedure at merchant site 4 07
Merchant requests payment 4 10
ECP supported? 4 15
No
Merchant prompts user to add ECP support 4 20
Yes
User adds ECP support? 4 25
No
Use alternate payment scheme if available 4 30
Yes
Merchant obtains payment via ECP 4 35
A
Payment successful? 4 40
No
Deny purchase 4 45
Yes
Complete purchase transaction 4 50
B

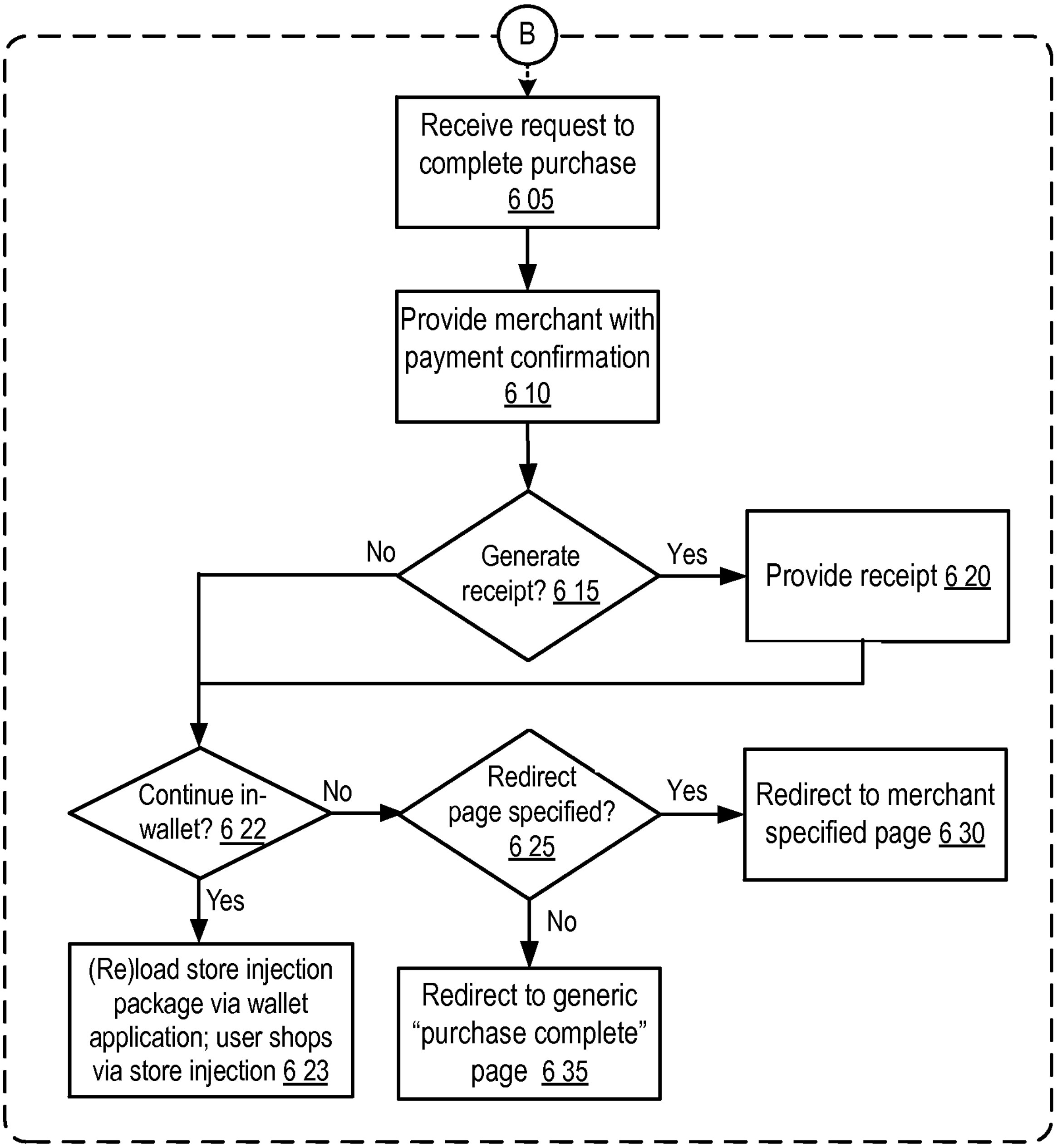
Example Logic Flow: E-Wallet Checkout (EWC) component

FIGURE 5


A
Receive request to obtain payment via ECP 5 05
Determine appropriate protocol handler 5 10
Authenticate user 5 11
User verified? 5 12
No
430
Yes
Obtain credentials to launch wallet (e.g., key, hash, etc.); instantiate wallet application 5 13
Inject store in wallet? 5 14a
No
Yes
Provide store injection package via wallet application; user shops via store injection 5 14b
Obtain purchase info via protocol handler 5 15
NFC-based authentication? 5 17
Yes
No
Obtain NFC based authentication info 5 22
Obtain non-NFC based authentication info 5 20
Authentication successful? 5 25
Yes
No
Deny access to ECP 5 30
Default payment method specified? 5 32
No
Yes
Obtain a selection of a wallet 5 35
Select default payment method 5 34
Obtain a selection of a payment method from the wallet 5 40
Discounts/rewards available? 5 45
Yes
No
Determine adjusted price/rewards 5 50
Obtain payment 5 55
440

Example Logic Flow: E-Wallet Checkout Payment Acquisition (EWCPA) component

FIGURE 6


B
Receive request to complete purchase 6 05
Provide merchant with payment confirmation 6 10
Generate receipt? 6 15
No
Yes
Provide receipt 6 20
Continue in-wallet? 6 22
No
Yes
Redirect page specified? 6 25
Yes
No
Redirect to merchant specified page 6 30
(Re)load store injection package via wallet application; user shops via store injection 6 23
Redirect to generic "purchase complete" page 6 35

Example Logic Flow: E-Wallet Checkout Purchase Transaction (EWCPT) component

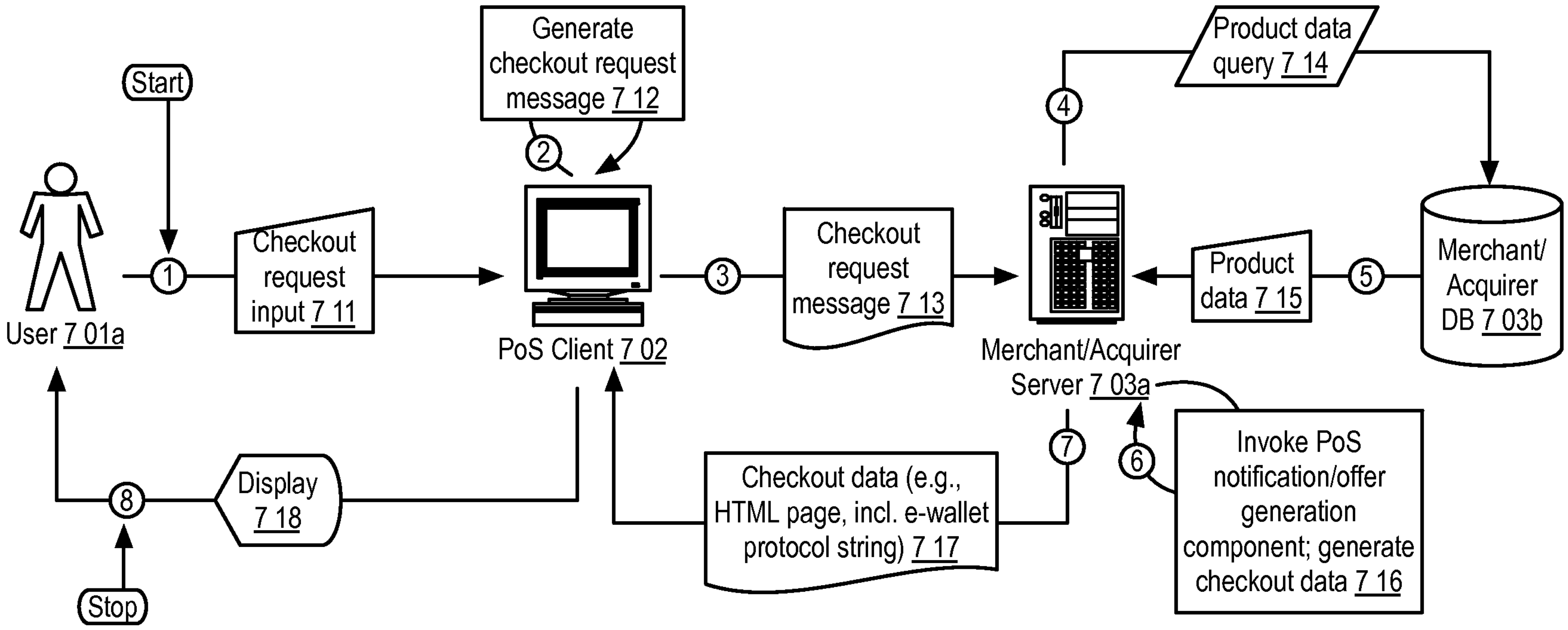

Start
Generate checkout request message 7 12
Product data query 7 14
1
2
4
User 7 01a
Checkout request input 7 11
PoS Client 7 02
3
Checkout request message 7 13
Merchant/Acquirer Server 7 03a
Product data 7 15
5
Merchant/ Acquirer DB 7 03b
Display 7 18
8
Stop
Checkout data (e.g., HTML page, incl. e-wallet protocol string) 7 17
7
6
Invoke PoS notification/offer generation component; generate checkout data 7 16

FIGURE 7

Example Datagraph: User Purchase Checkout ("UPC") component

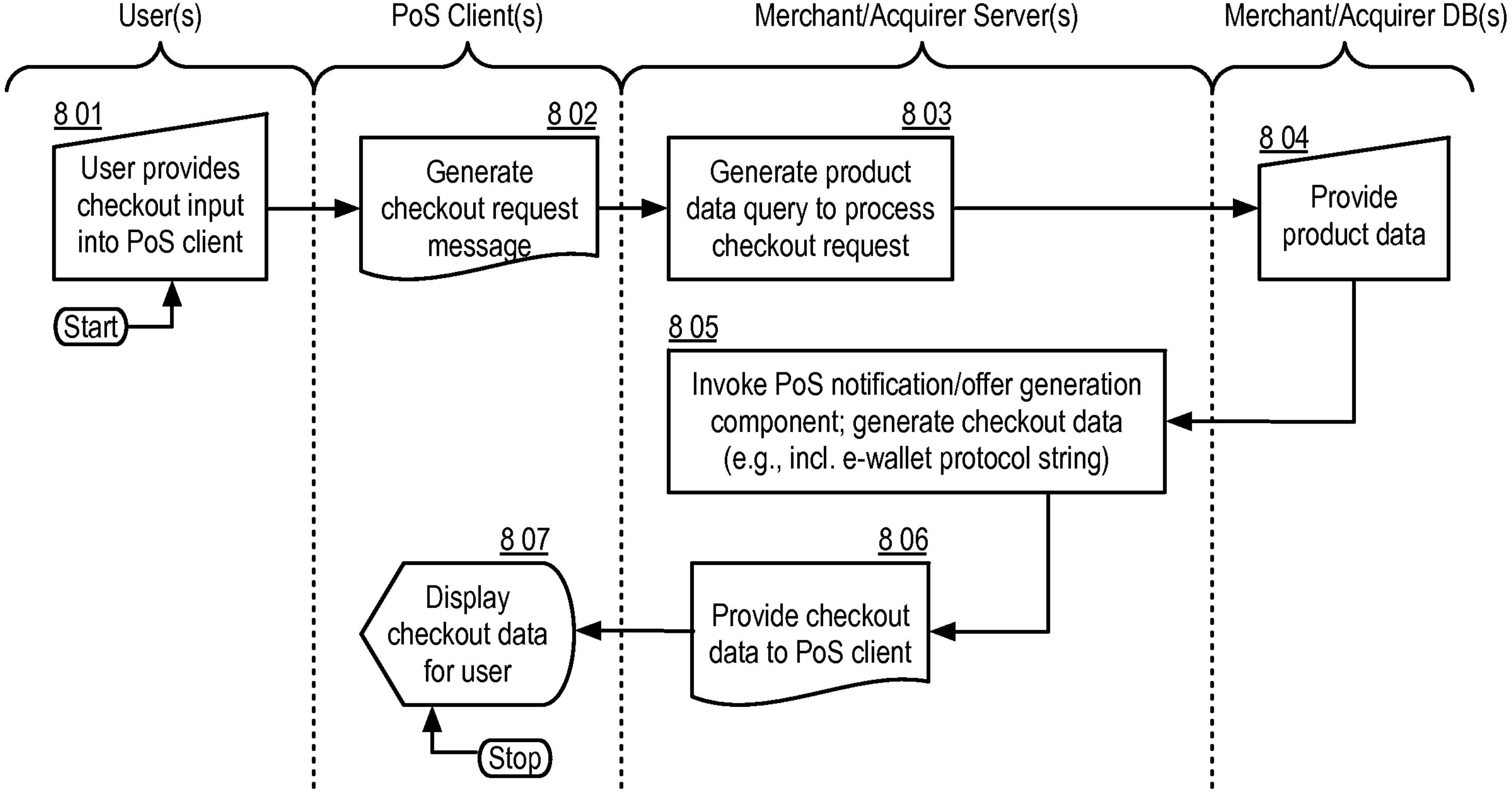

User(s)
PoS Client(s)
Merchant/Acquirer Server(s)
Merchant/Acquirer DB(s)
8 01
User provides checkout input into PoS client
Start
8 02
Generate checkout request message
8 03
Generate product data query to process checkout request
8 04
Provide product data
8 05
Invoke PoS notification/offer generation component; generate checkout data (e.g., incl. e-wallet protocol string)
8 06
Provide checkout data to PoS client
8 07
Display checkout data for user
Stop

FIGURE 8

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

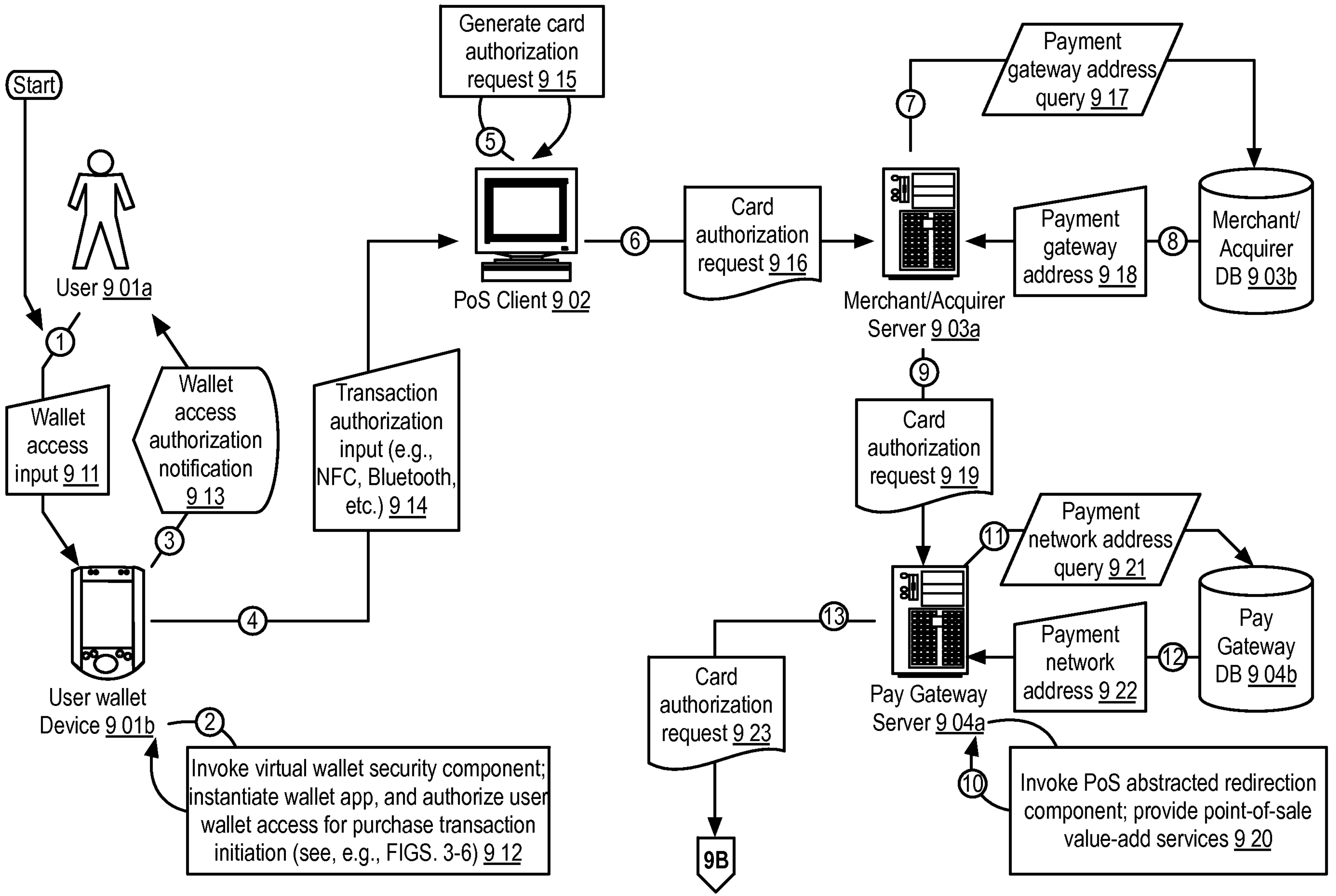

Start
User 9 01a
1
Wallet access input 9 11
Wallet access authorization notification 9 13
3
User wallet Device 9 01b
2
Invoke virtual wallet security component; instantiate wallet app, and authorize user wallet access for purchase transaction initiation (see, e.g., FIGS. 3-6) 9 12
4
Transaction authorization input (e.g., NFC, Bluetooth, etc.) 9 14
Generate card authorization request 9 15
5
PoS Client 9 02
6
Card authorization request 9 16
Merchant/Acquirer Server 9 03a
7
Payment gateway address query 9 17
Payment gateway address 9 18
8
Merchant/ Acquirer DB 9 03b
9
Card authorization request 9 19
11
Payment network address query 9 21
Pay Gateway DB 9 04b
Payment network address 9 22
12
13
Pay Gateway Server 9 04a
10
Invoke PoS abstracted redirection component; provide point-of-sale value-add services 9 20
Card authorization request 9 23
9B

FIGURE 9A

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

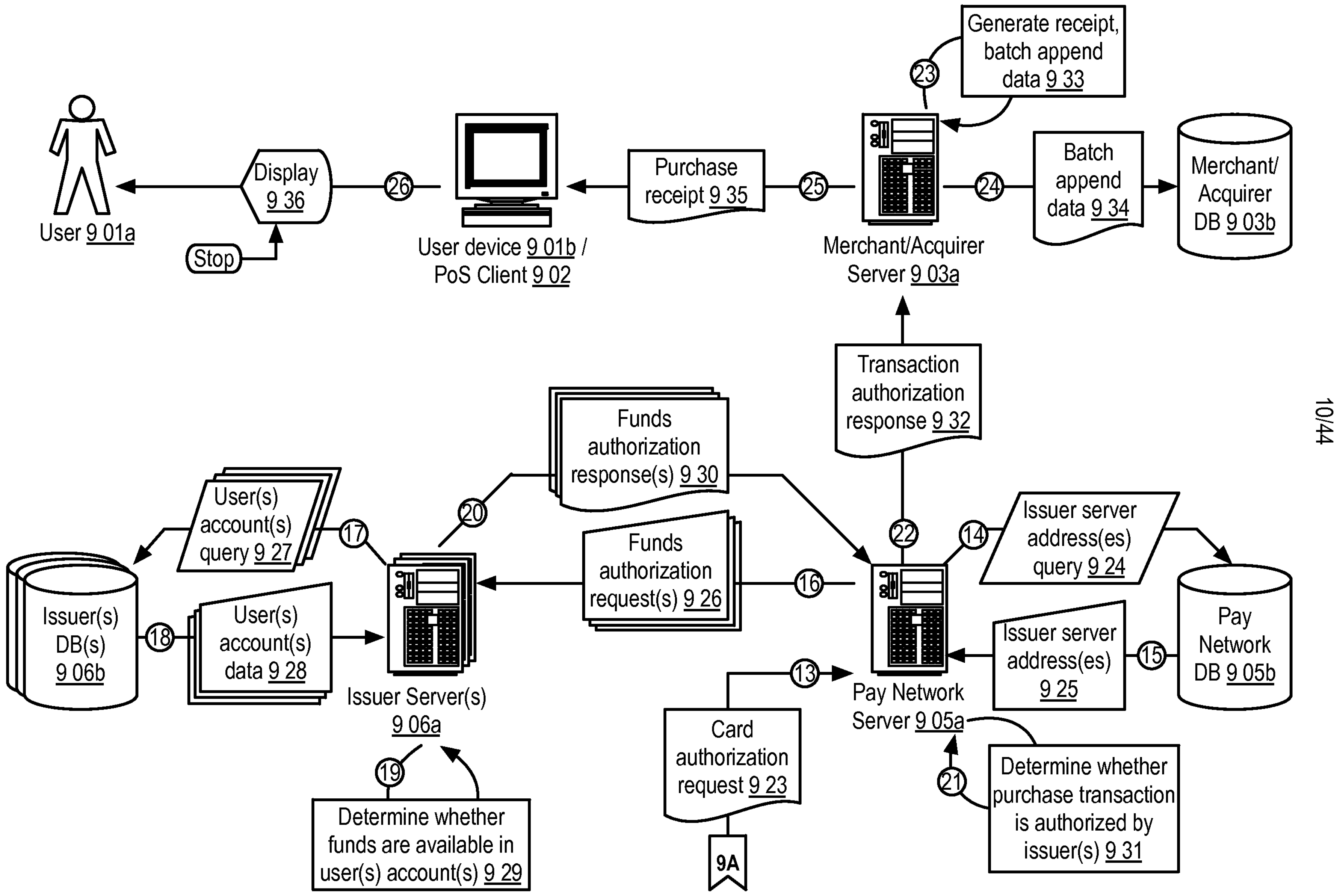

Generate receipt, batch append data 9 33
23
User 9 01a
Display 9 36
Stop
26
User device 9 01b / PoS Client 9 02
Purchase receipt 9 35
25
Merchant/Acquirer Server 9 03a
24
Batch append data 9 34
Merchant/ Acquirer DB 9 03b
Transaction authorization response 9 32
Funds authorization response(s) 9 30
20
User(s) account(s) query 9 27
17
Funds authorization request(s) 9 26
16
22
14
Issuer server address(es) query 9 24
Issuer(s) DB(s) 9 06b
18
User(s) account(s) data 9 28
Issuer Server(s) 9 06a
Pay Network Server 9 05a
Issuer server address(es) 9 25
15
Pay Network DB 9 05b
13
Card authorization request 9 23
9A
19
Determine whether funds are available in user(s) account(s) 9 29
21
Determine whether purchase transaction is authorized by issuer(s) 9 31

FIGURE 9B

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

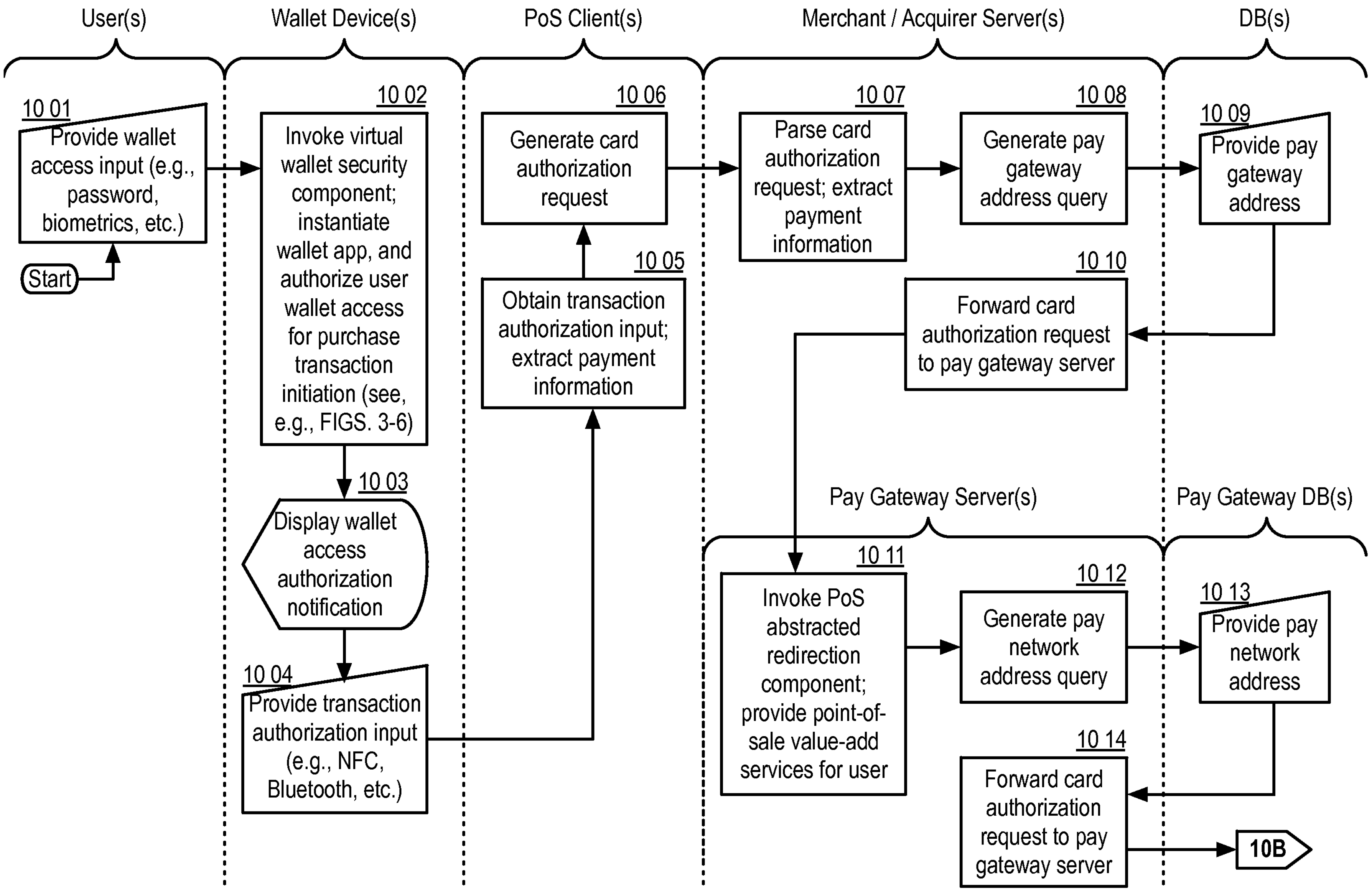
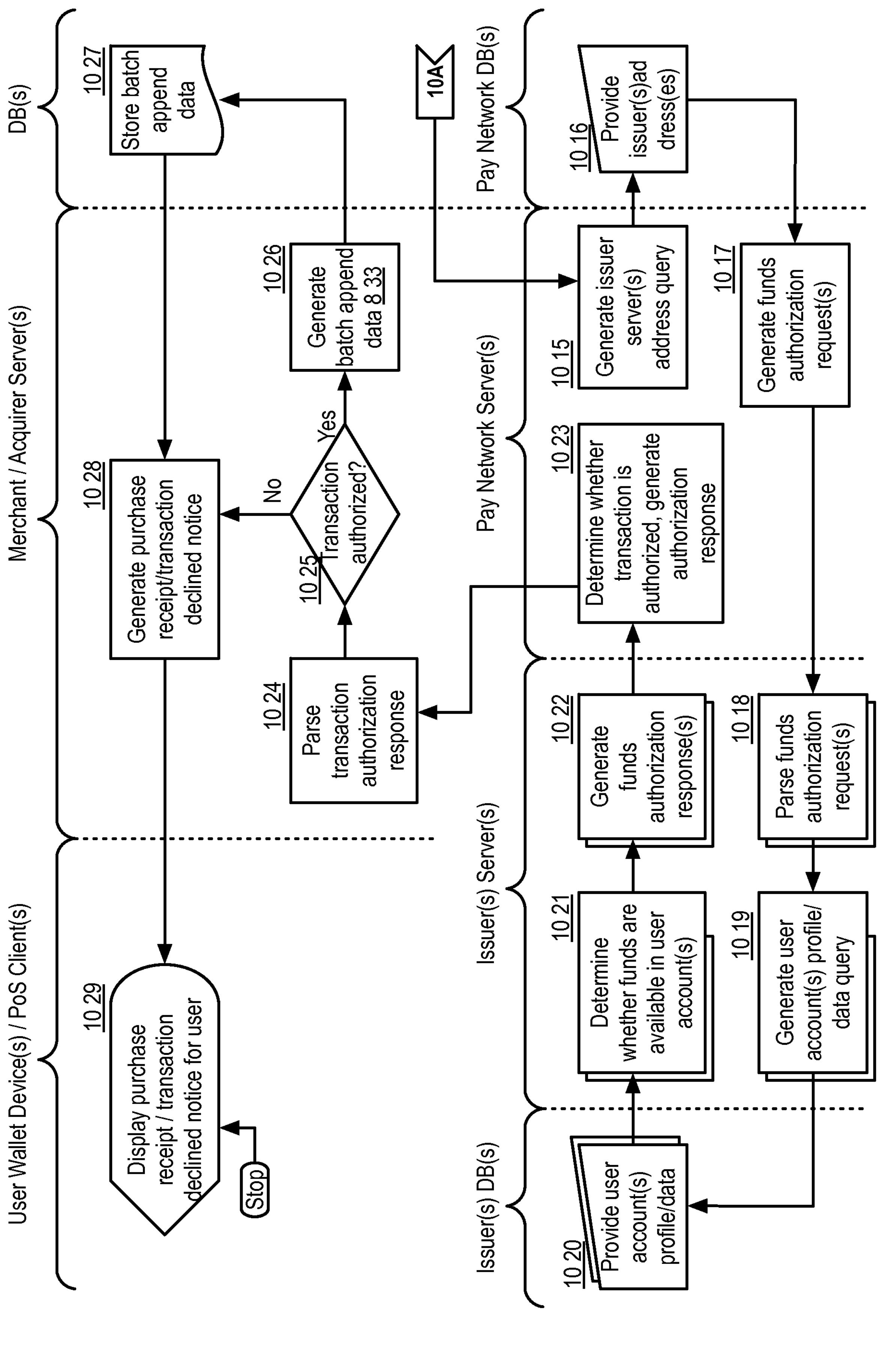

User(s)
Wallet Device(s)
PoS Client(s)
Merchant / Acquirer Server(s)
DB(s)
10 01
Provide wallet access input (e.g., password, biometrics, etc.)
Start
10 02
Invoke virtual wallet security component; instantiate wallet app, and authorize user wallet access for purchase transaction initiation (see, e.g., FIGS. 3-6)
10 03
Display wallet access authorization notification
10 04
Provide transaction authorization input (e.g., NFC, Bluetooth, etc.)
10 05
Obtain transaction authorization input; extract payment information
10 06
Generate card authorization request
10 07
Parse card authorization request; extract payment information
10 08
Generate pay gateway address query
10 09
Provide pay gateway address
10 10
Forward card authorization request to pay gateway server
Pay Gateway Server(s)
Pay Gateway DB(s)
10 11
Invoke PoS abstracted redirection component; provide point-of-sale value-add services for user
10 12
Generate pay network address query
10 13
Provide pay network address
10 14
Forward card authorization request to pay gateway server
10B

FIGURE 10A

Example Logic Flow: Purchase Transaction Authorization ("PTA") component


User Wallet Device(s) / PoS Client(s)
Merchant / Acquirer Server(s)
DB(s)
10 27
Store batch append data
10 26
Generate batch append data 8 33
Yes
No
10 25
Transaction authorized?
10 28
Generate purchase receipt/transaction declined notice
10 24
Parse transaction authorization response
10 29
Display purchase receipt / transaction declined notice for user
Stop
10A
Issuer(s) DB(s)
Issuer(s) Server(s)
Pay Network Server(s)
Pay Network DB(s)
10 16
Provide issuer(s)ad dress(es)
10 15
Generate issuer server(s) address query
10 17
Generate funds authorization request(s)
10 23
Determine whether transaction is authorized, generate authorization response
10 22
Generate funds authorization response(s)
10 18
Parse funds authorization request(s)
10 21
Determine whether funds are available in user account(s)
10 19
Generate user account(s) profile/ data query
10 20
Provide user account(s) profile/data

Example Logic Flow: Purchase Transaction Authorization ("PTA") component

FIGURE 10B

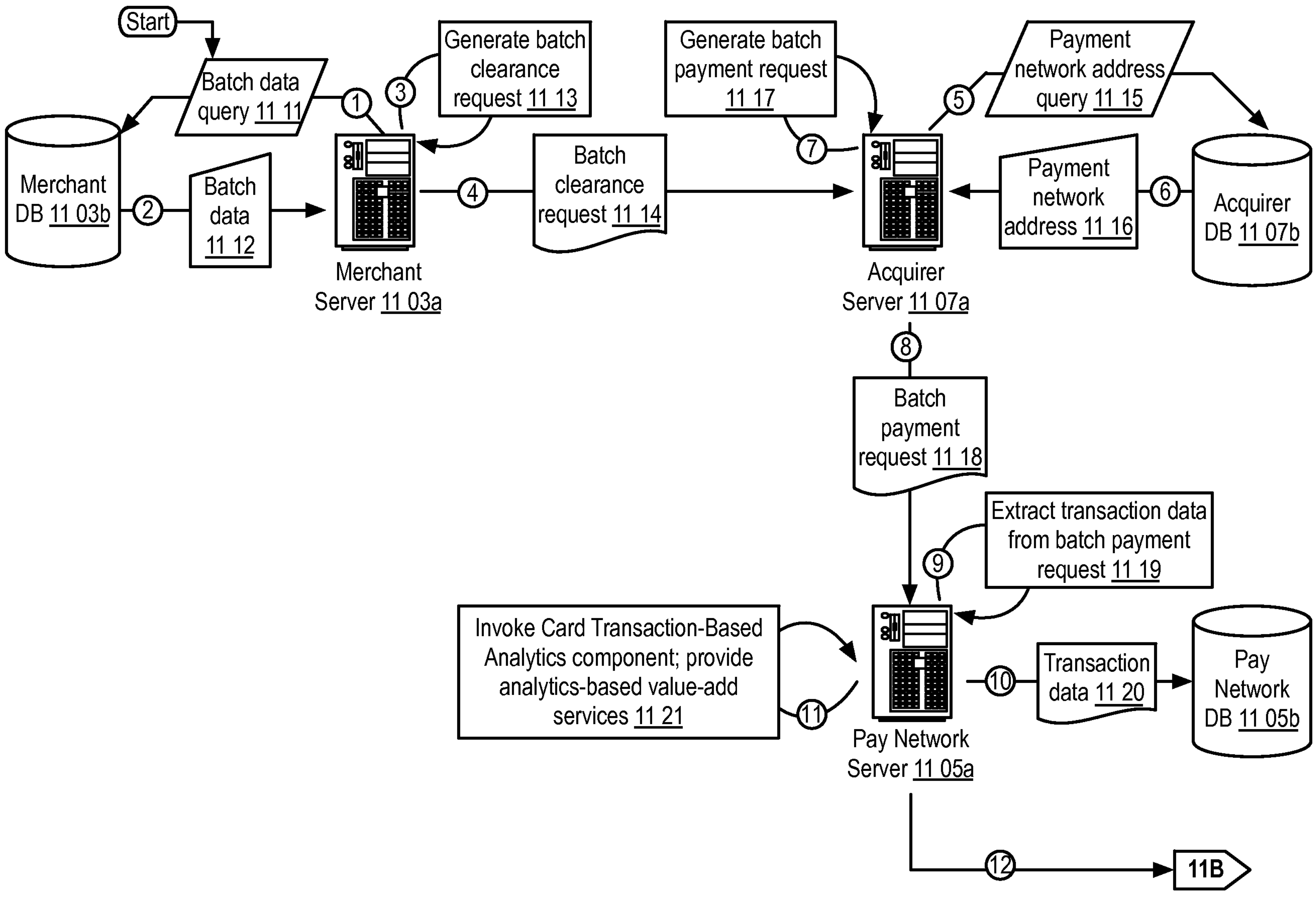

Start
Batch data query 11 11
Merchant DB 11 03b
Batch data 11 12
Merchant Server 11 03a
Generate batch clearance request 11 13
Batch clearance request 11 14
Generate batch payment request 11 17
Payment network address query 11 15
Payment network address 11 16
Acquirer DB 11 07b
Acquirer Server 11 07a
Batch payment request 11 18
Extract transaction data from batch payment request 11 19
Invoke Card Transaction-Based Analytics component; provide analytics-based value-add services 11 21
Pay Network Server 11 05a
Transaction data 11 20
Pay Network DB 11 05b
11B
1
2
3
4
5
6
7
8
9
10
11
12

FIGURE 11A

Example Datagraph: Purchase Transaction Clearance ("PTC") component

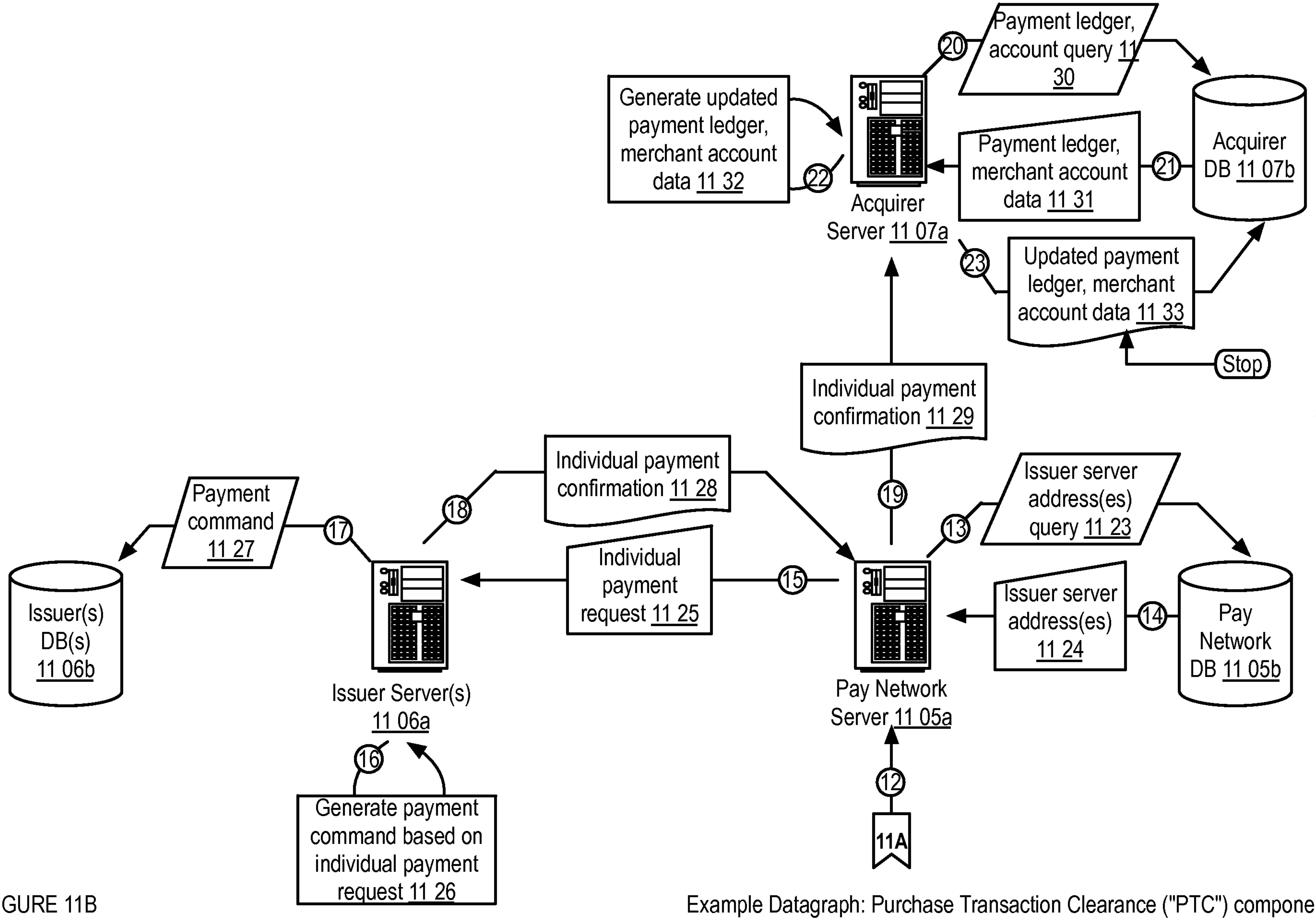

Generate updated payment ledger, merchant account data 11 32
22
Acquirer Server 11 07a
20
Payment ledger, account query 11 30
Acquirer DB 11 07b
21
Payment ledger, merchant account data 11 31
23
Updated payment ledger, merchant account data 11 33
Stop
Individual payment confirmation 11 29
19
Payment command 11 27
17
18
Individual payment confirmation 11 28
Individual payment request 11 25
15
Issuer(s) DB(s) 11 06b
Issuer Server(s) 11 06a
16
Generate payment command based on individual payment request 11 26
13
Issuer server address(es) query 11 23
Issuer server address(es) 11 24
14
Pay Network DB 11 05b
Pay Network Server 11 05a
12
11A

FIGURE 11B

Example Datagraph: Purchase Transaction Clearance ("PTC") component

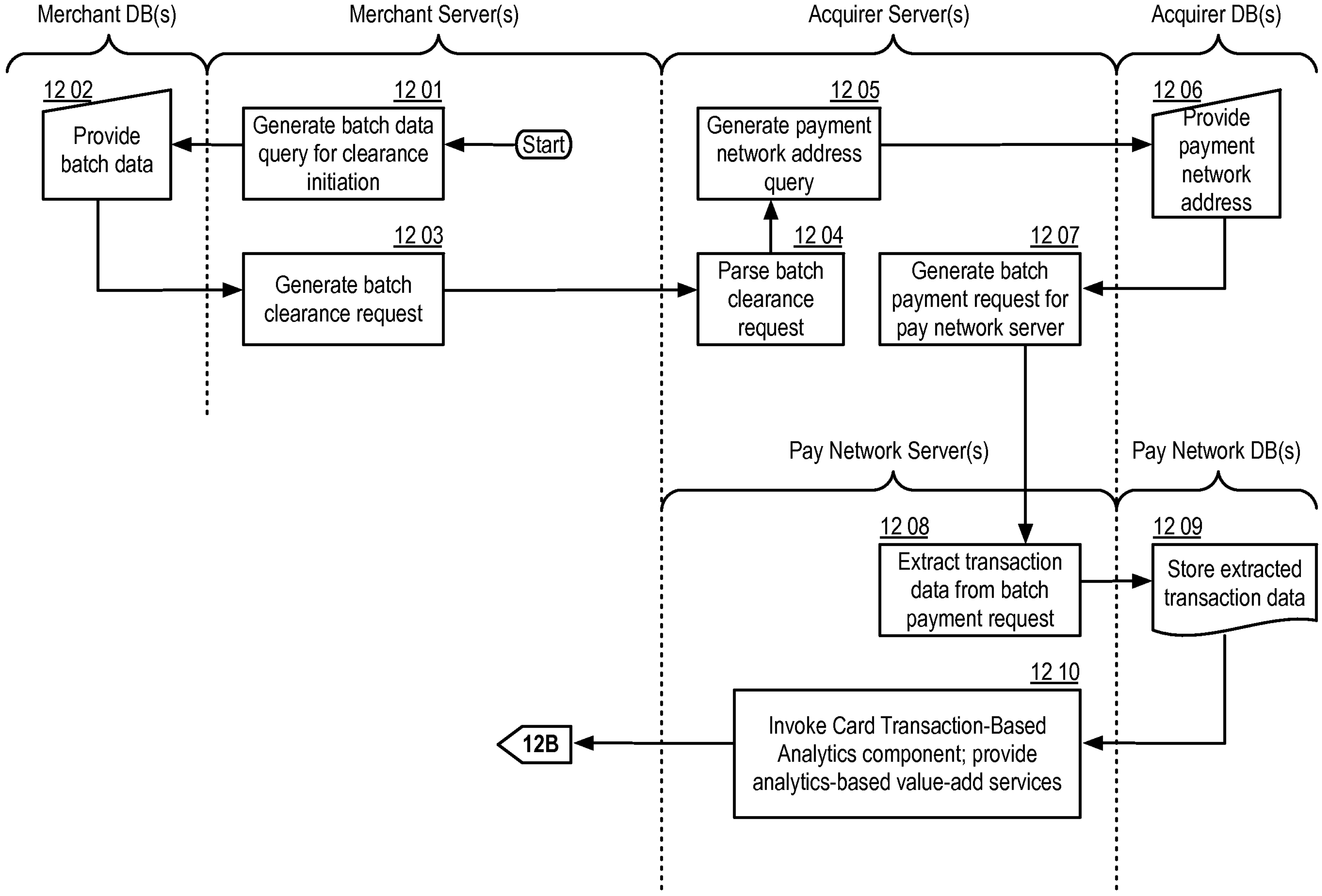

Merchant DB(s)
Merchant Server(s)
Acquirer Server(s)
Acquirer DB(s)
Pay Network Server(s)
Pay Network DB(s)
Start
12 01
Generate batch data query for clearance initiation
12 02
Provide batch data
12 03
Generate batch clearance request
12 04
Parse batch clearance request
12 05
Generate payment network address query
12 06
Provide payment network address
12 07
Generate batch payment request for pay network server
12 08
Extract transaction data from batch payment request
12 09
Store extracted transaction data
12 10
Invoke Card Transaction-Based Analytics component; provide analytics-based value-add services
12B

FIGURE 12A

Example Logic Flow: Purchase Transaction Clearance ("PTC") component

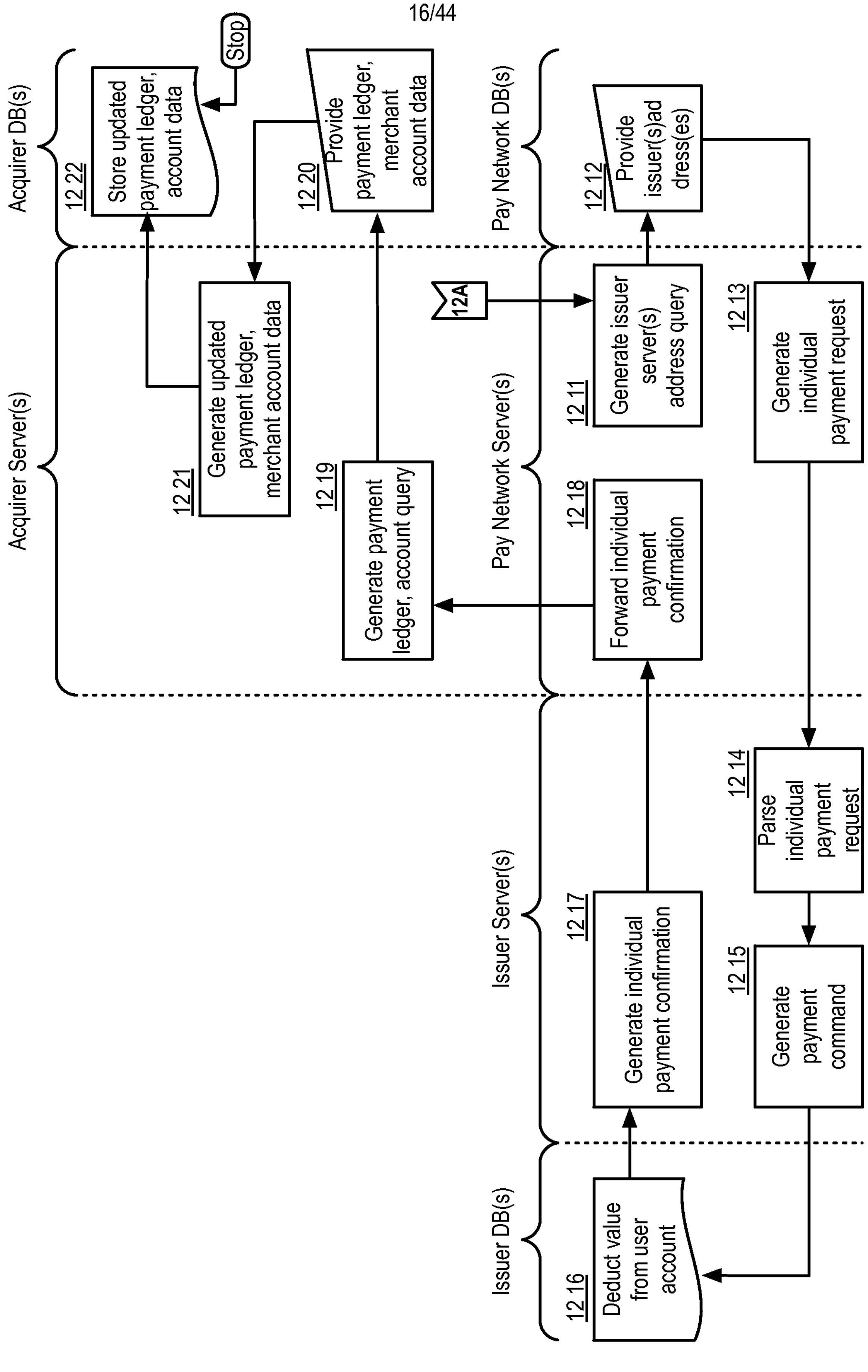

Acquirer DB(s)
Acquirer Server(s)
Pay Network DB(s)
Pay Network Server(s)
Issuer Server(s)
Issuer DB(s)
12A
12 11
Generate issuer server(s) address query
12 12
Provide issuer(s)address(es)
12 13
Generate individual payment request
12 14
Parse individual payment request
12 15
Generate payment command
12 16
Deduct value from user account
12 17
Generate individual payment confirmation
12 18
Forward individual payment confirmation
12 19
Generate payment ledger, account query
12 20
Provide payment ledger, merchant account data
12 21
Generate updated payment ledger, merchant account data
12 22
Store updated payment ledger, account data
Stop

FIGURE 12B

Example Logic Flow: Purchase Transaction Clearance ("PTC") component

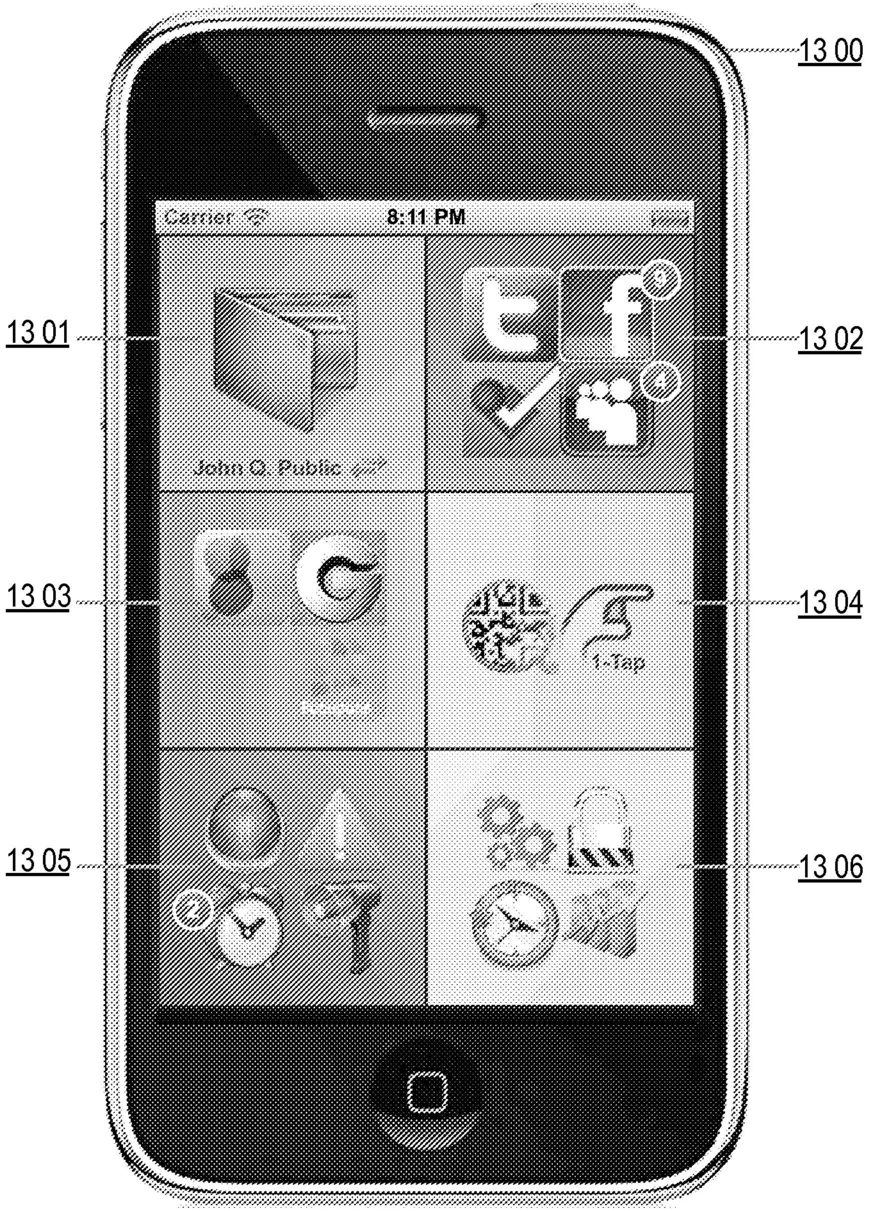
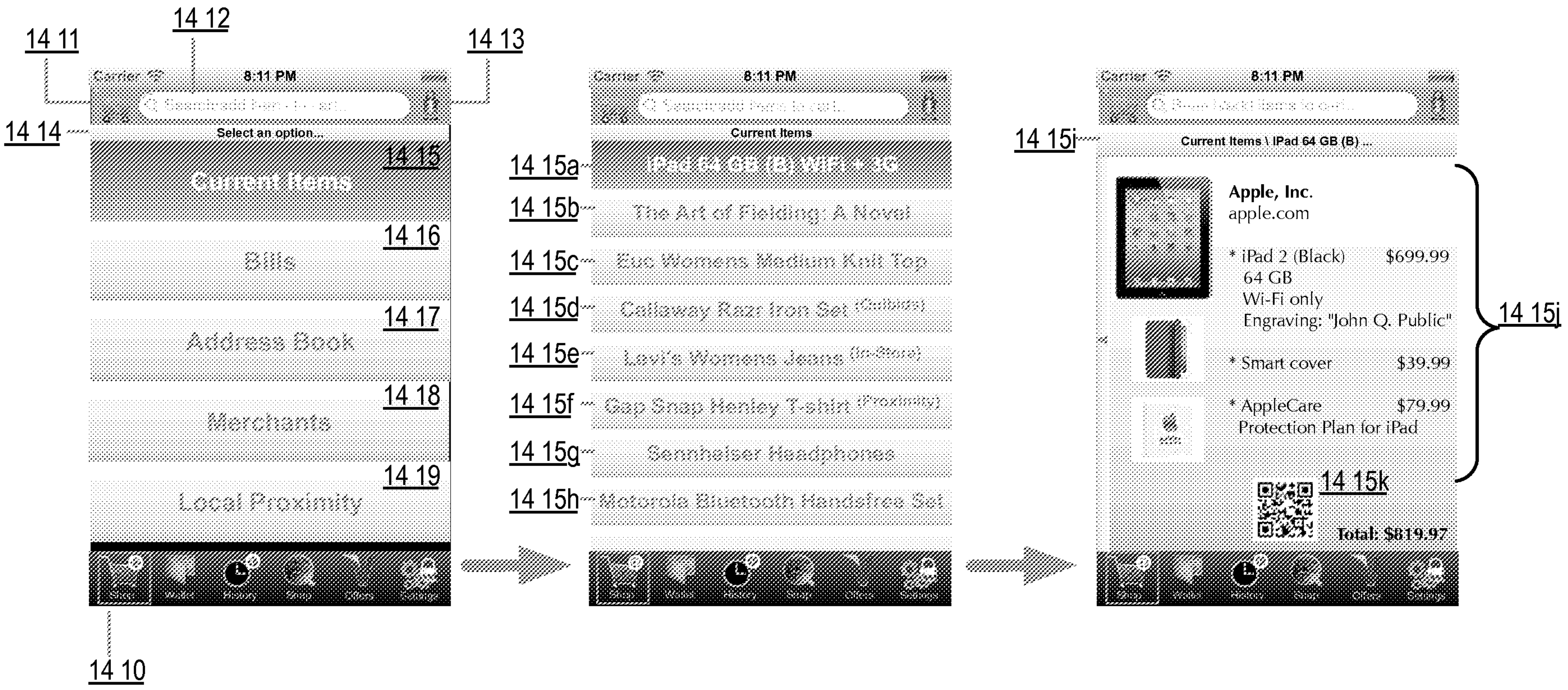


FIGURE 13

Example: Virtual Wallet Mobile App - Feature Overview



Apple, Inc.
apple.com

* iPad 2 (Black) 64 GB Wi-Fi only Engraving: "John Q. Public"	$699.99
* Smart cover	$39.99
* AppleCare Protection Plan for iPad	$79.99

Total: $819.97

FIGURE 14A

Example: Virtual Wallet Mobile App - Shopping Mode

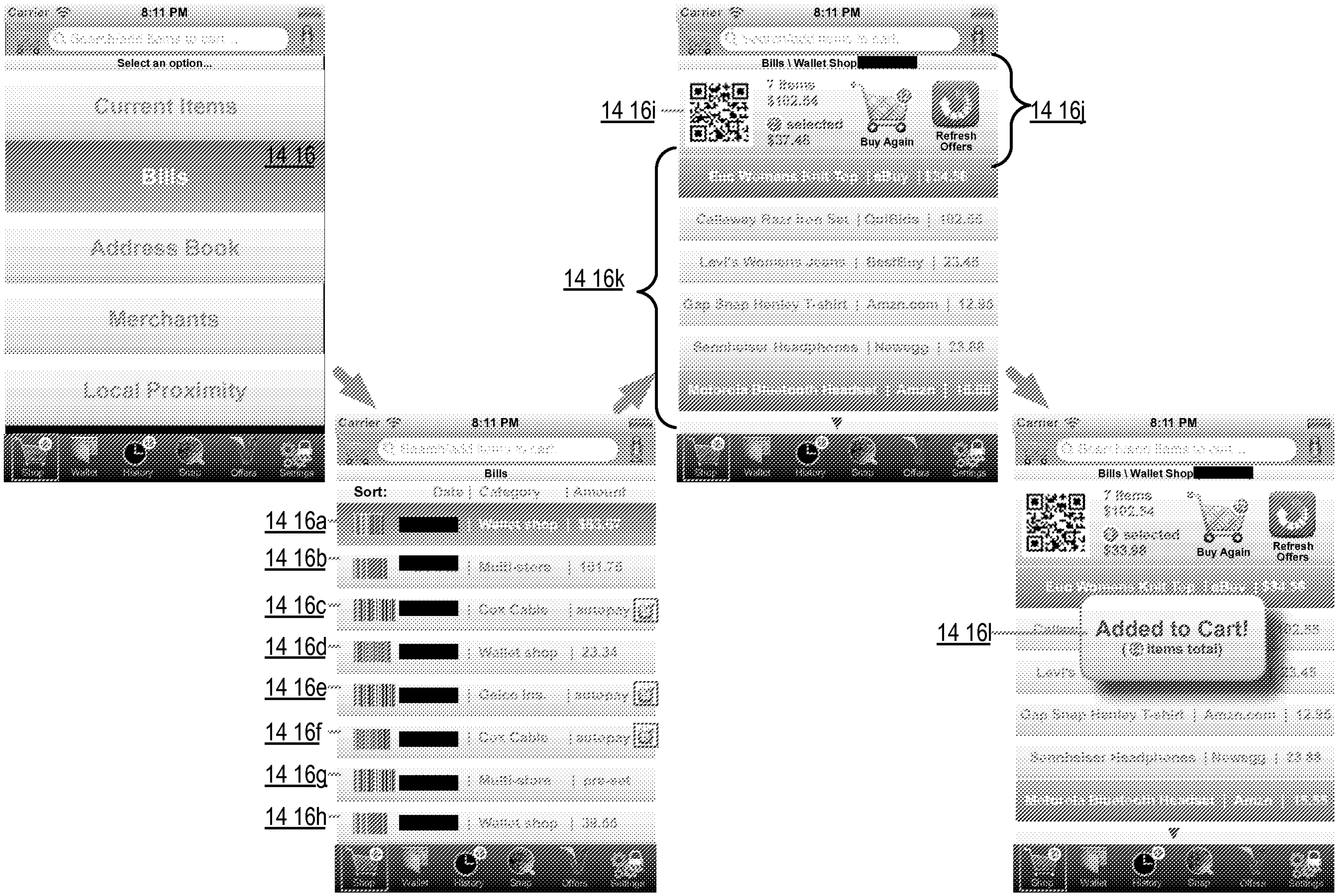
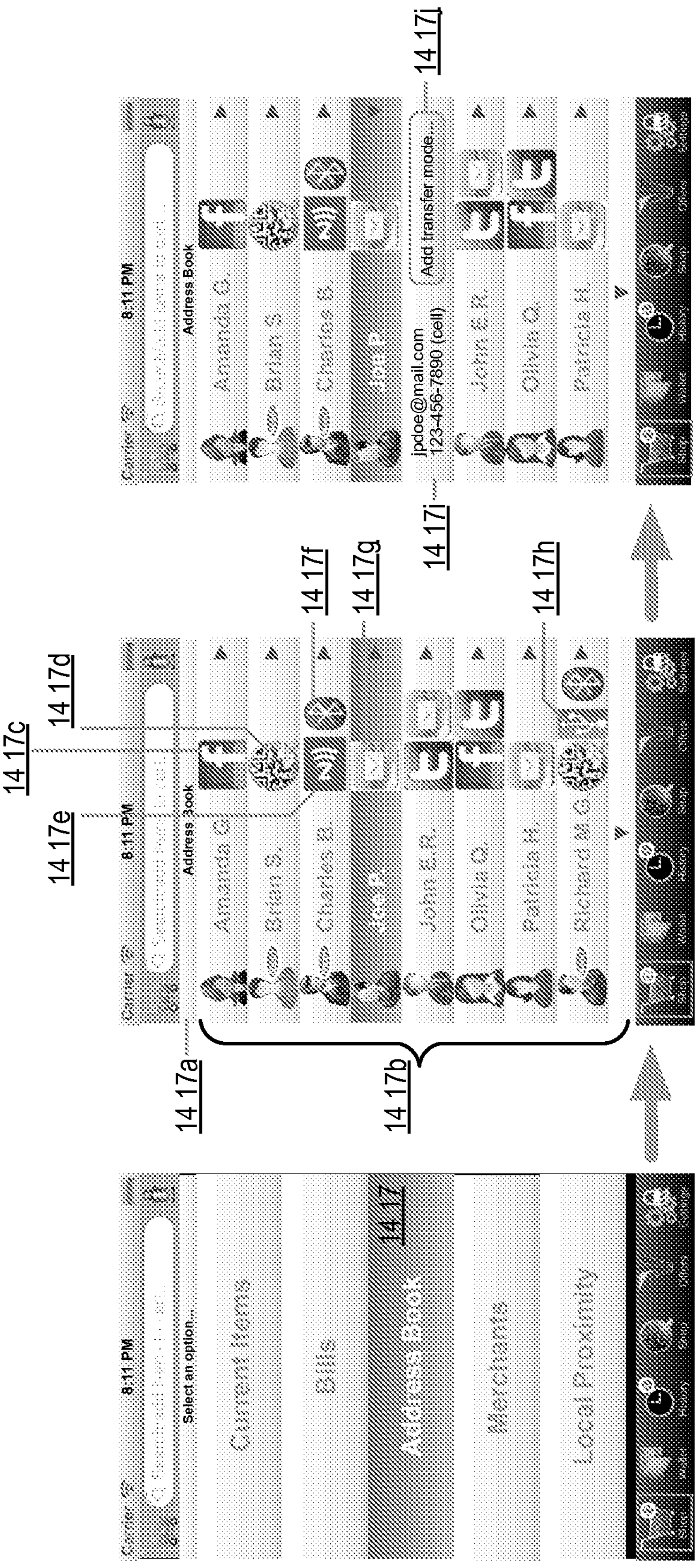


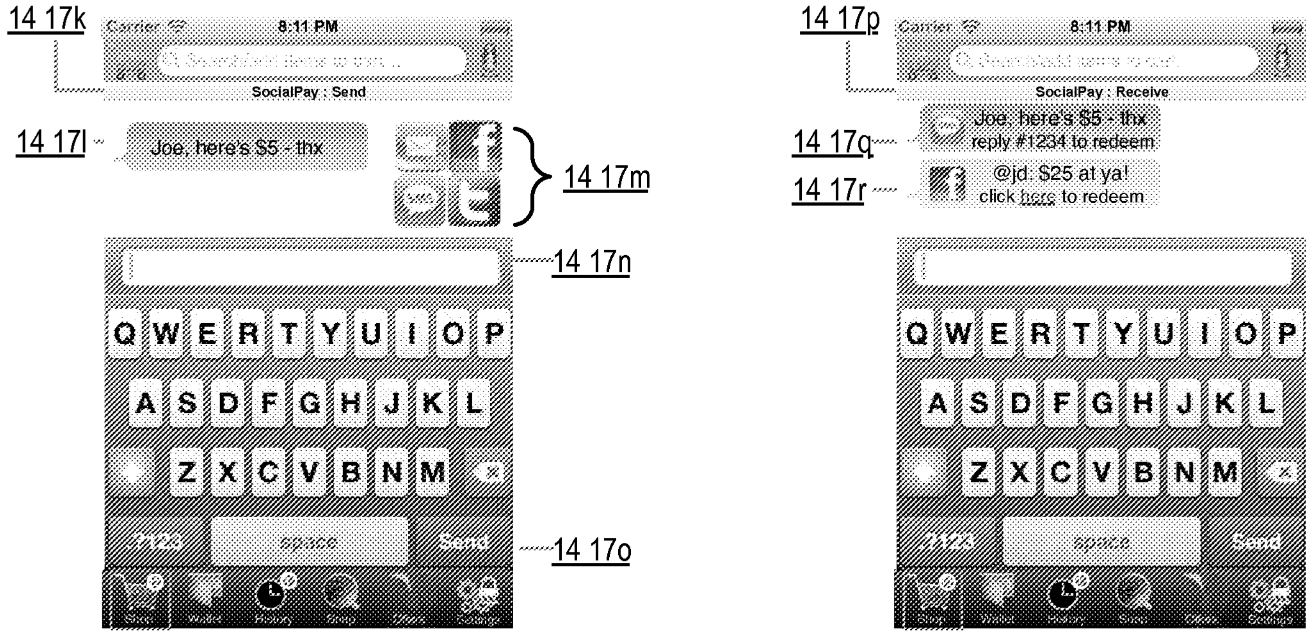
FIGURE 14B

Example: Virtual Wallet Mobile App - Shopping Mode



Example: Virtual Wallet Mobile App - Shopping Mode

FIGURE 14C



Joe, here's $5 - thx

Joe, here's $5 - thx
reply #1234 to redeem

@jd: $25 at ya!
click here to redeem

FIGURE 14D

Example: Virtual Wallet Mobile App - SocialPay Mode

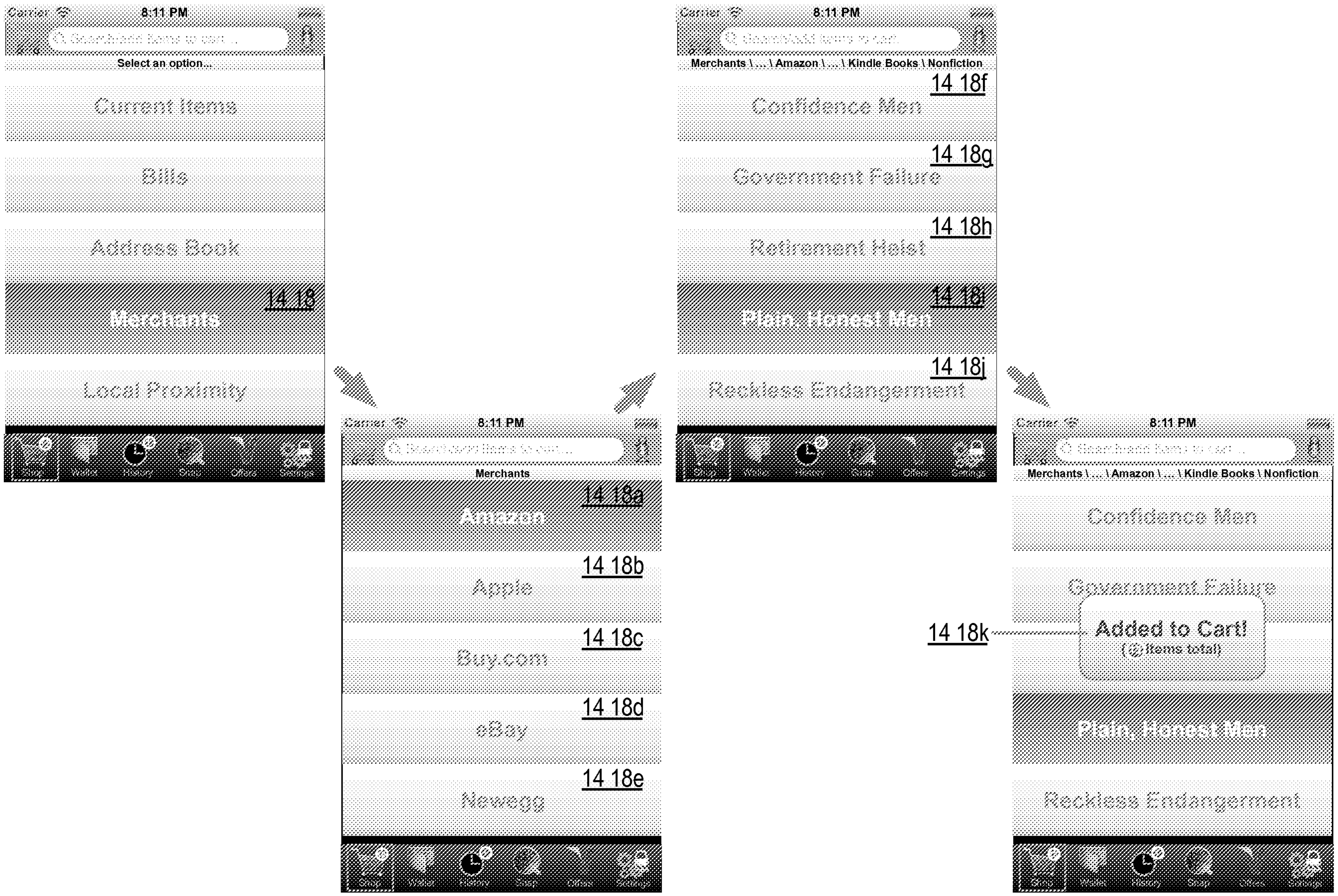


FIGURE 14E

Example: Virtual Wallet Mobile App - Shopping Mode

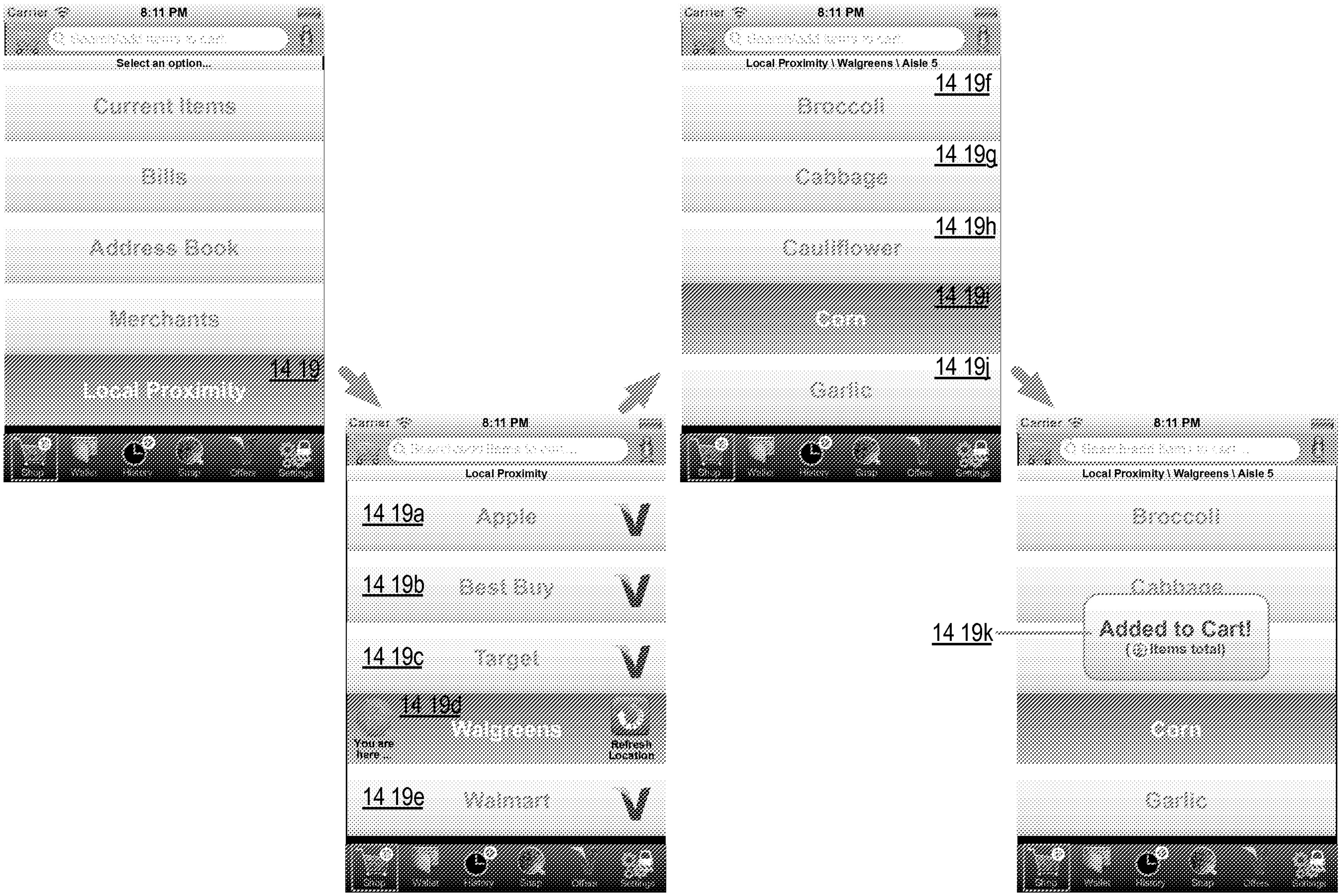


FIGURE 14F

Example: Virtual Wallet Mobile App - Shopping Mode
]

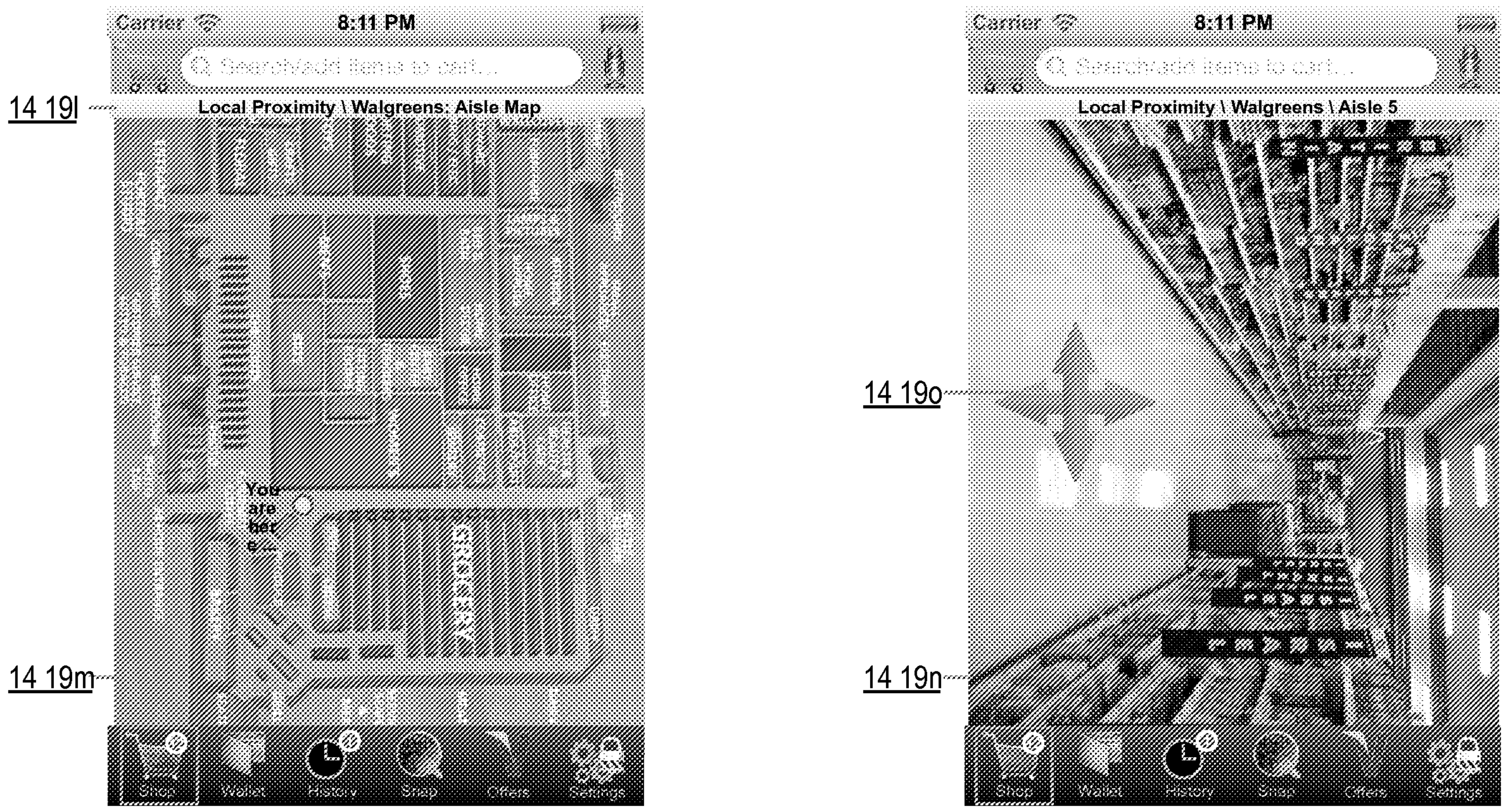


FIGURE 14G

Example: Virtual Wallet Mobile App - Shopping Mode

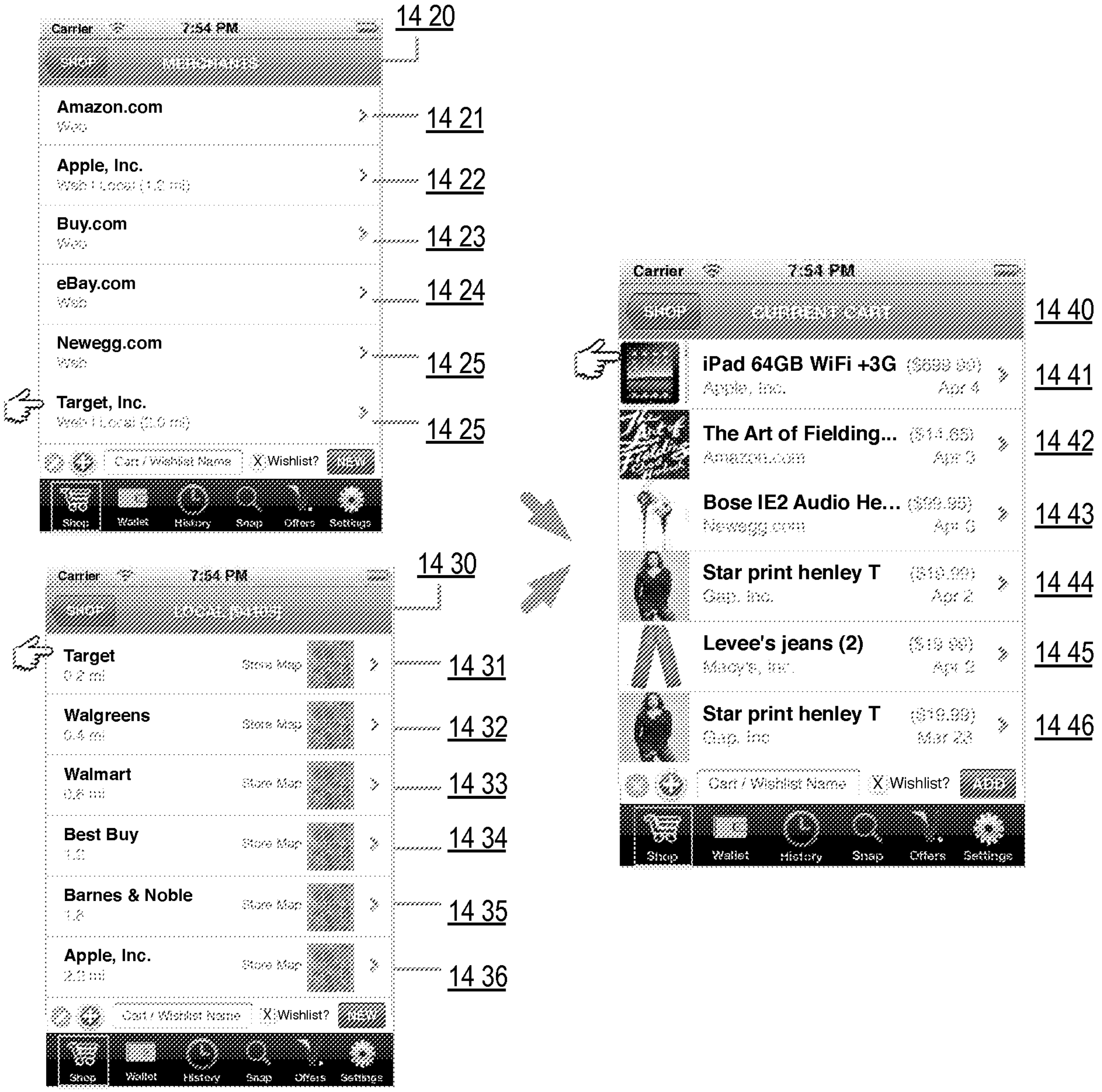


FIGURE 14H

Example: Virtual Wallet Application Embodiment - (Local) Merchant Shopping Mode

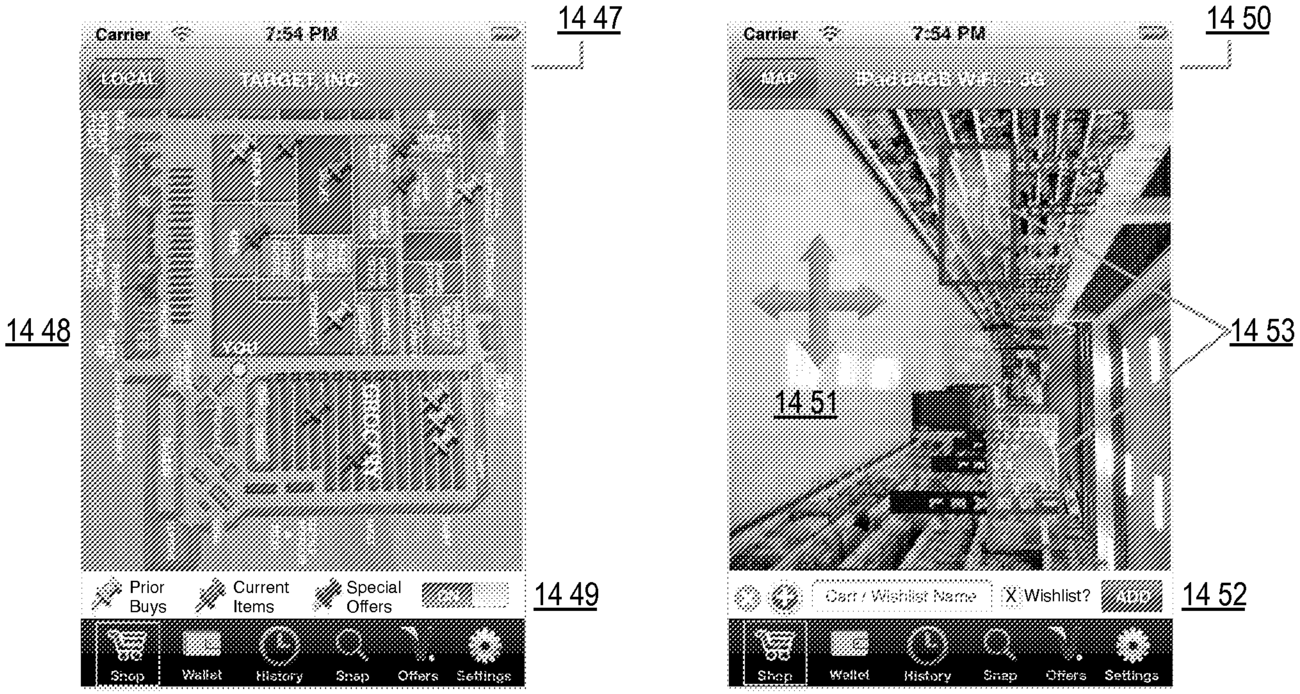


FIGURE 14I

Example: Virtual Wallet Application Embodiment - (Local) Merchant Shopping Mode: Virtual Store Injection

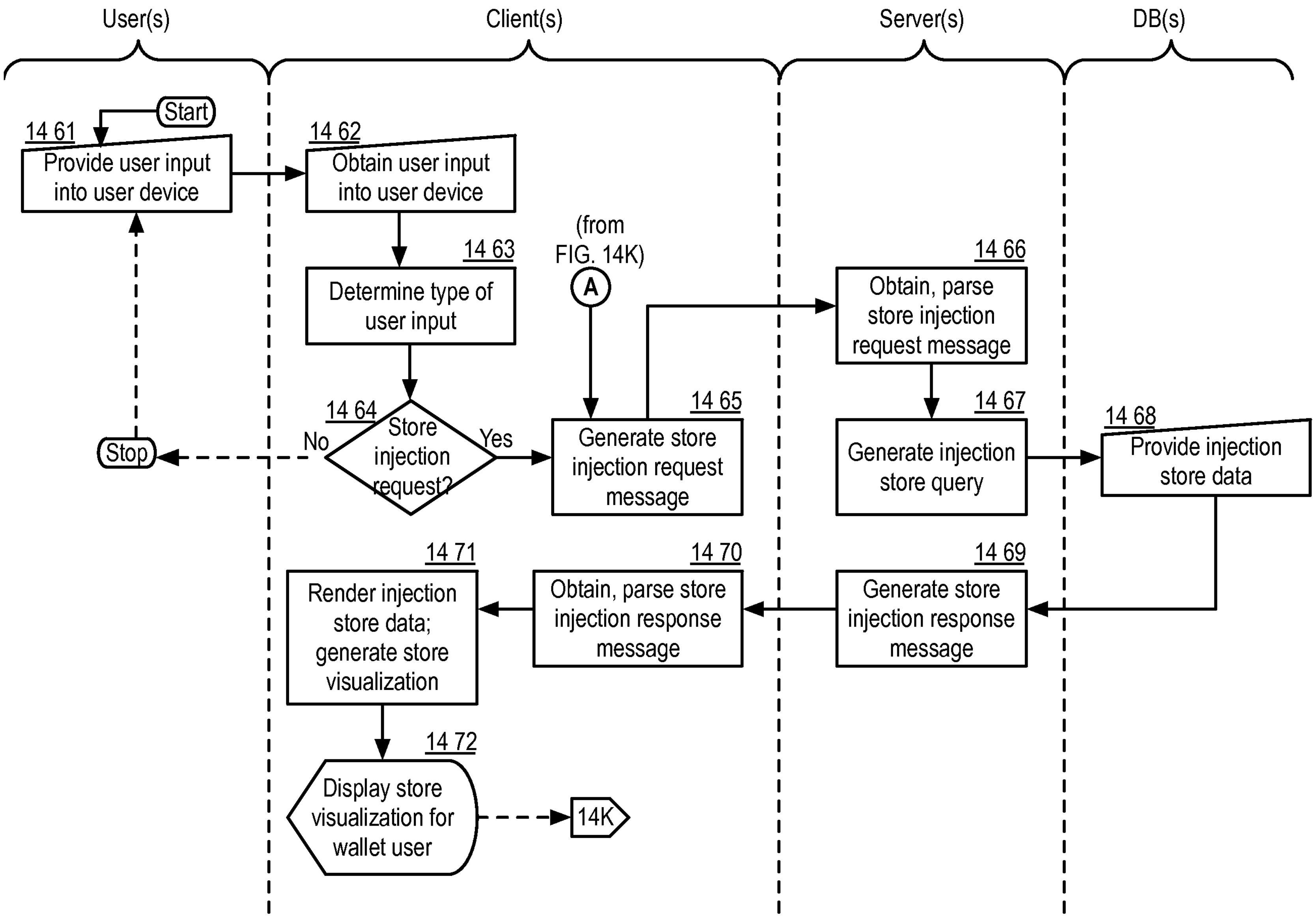

User(s)
Client(s)
Server(s)
DB(s)
Start
1461
Provide user input into user device
1462
Obtain user input into user device
1463
Determine type of user input
1464
Store injection request?
No
Yes
Stop
(from FIG. 14K)
A
1465
Generate store injection request message
1466
Obtain, parse store injection request message
1467
Generate injection store query
1468
Provide injection store data
1469
Generate store injection response message
1470
Obtain, parse store injection response message
1471
Render injection store data; generate store visualization
1472
Display store visualization for wallet user
14K

FIGURE 14J

Example: Virtual Wallet Store Injection ("VWSI") component

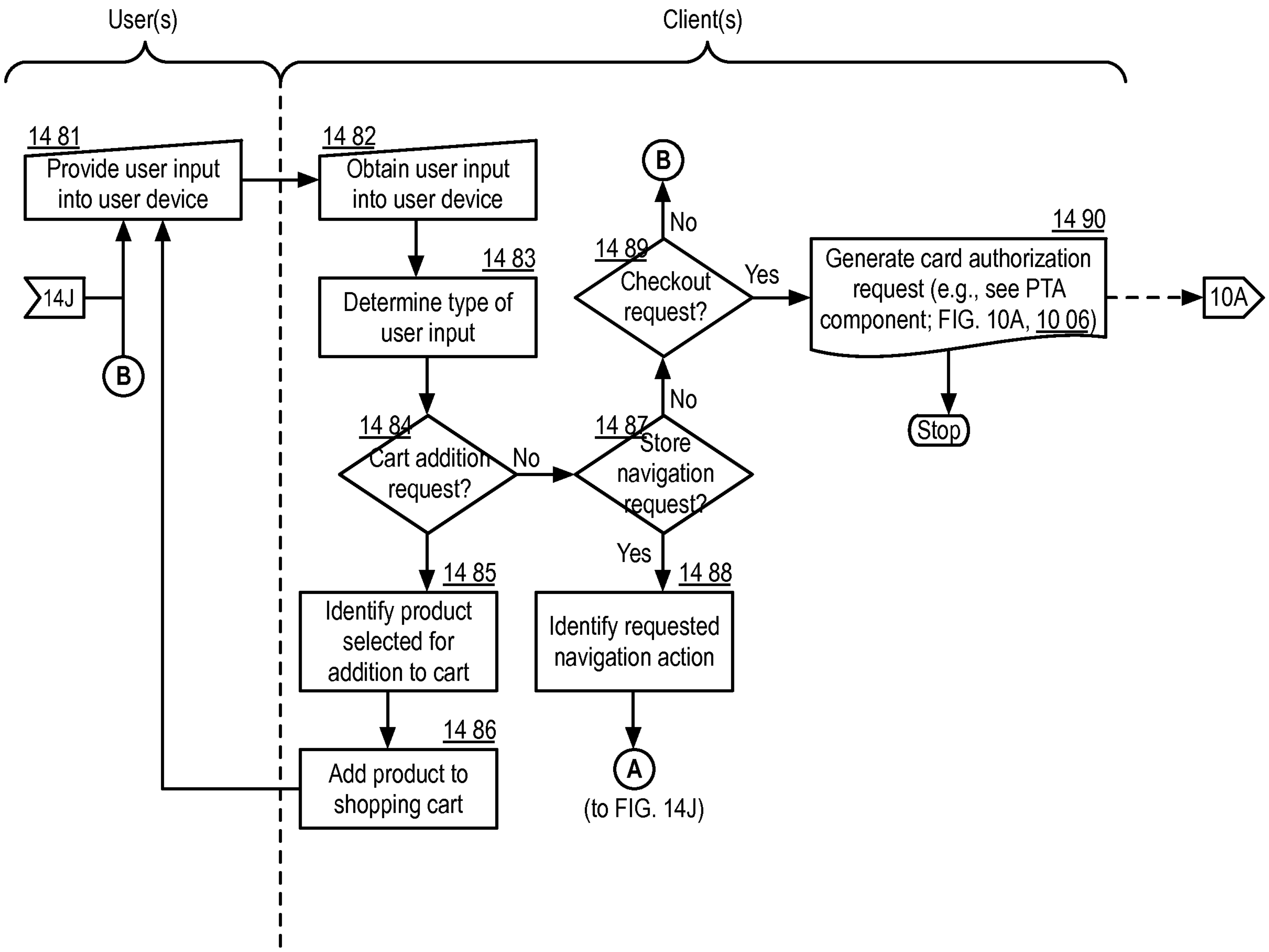

User(s)
Client(s)
14 81
Provide user input into user device
14J
B
14 82
Obtain user input into user device
14 83
Determine type of user input
14 84
Cart addition request?
No
14 85
Identify product selected for addition to cart
14 86
Add product to shopping cart
14 87
Store navigation request?
Yes
14 88
Identify requested navigation action
A
(to FIG. 14J)
No
14 89
Checkout request?
No
B
Yes
14 90
Generate card authorization request (e.g., see PTA component; FIG. 10A, 10 06)
Stop
10A

FIGURE 14K

Example: Virtual Wallet Store Injection ("VWSI") component

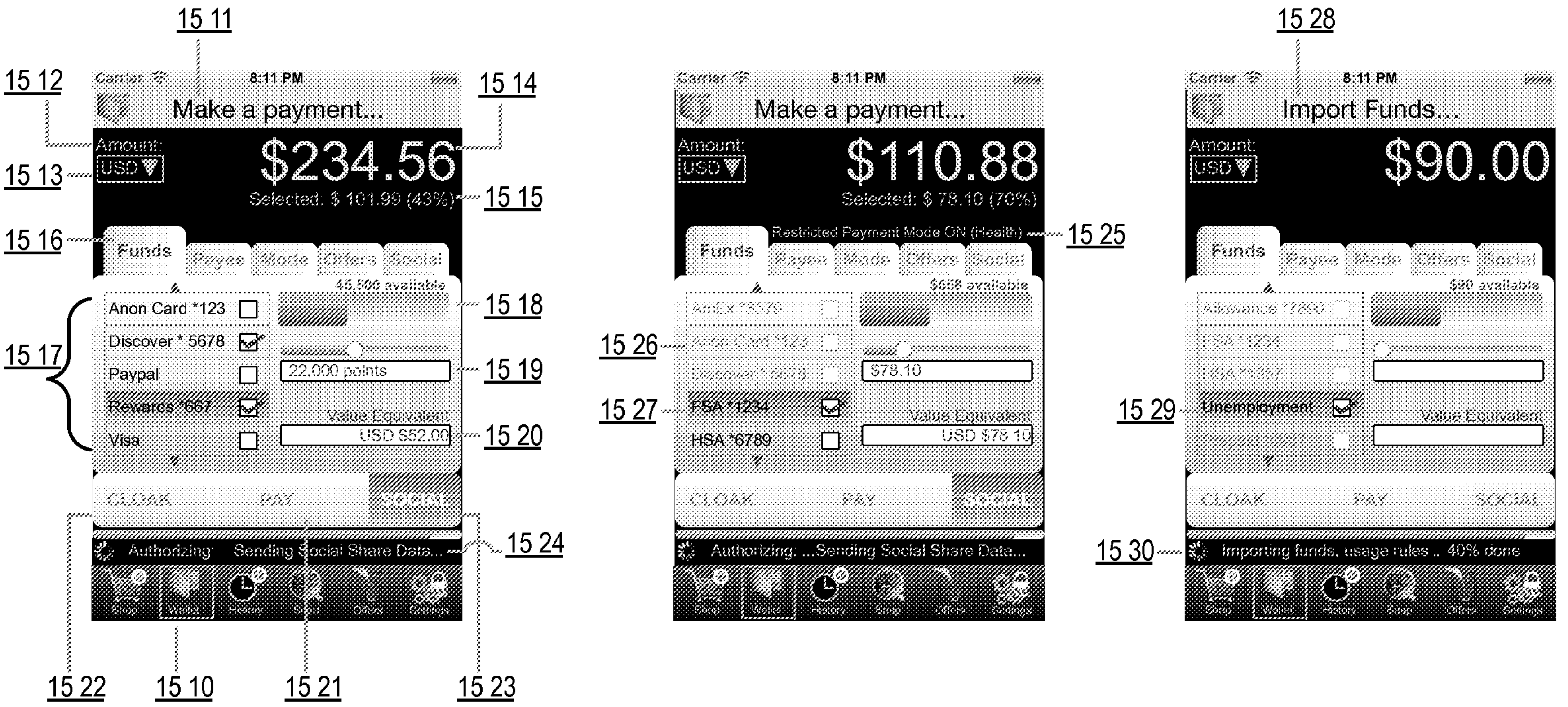
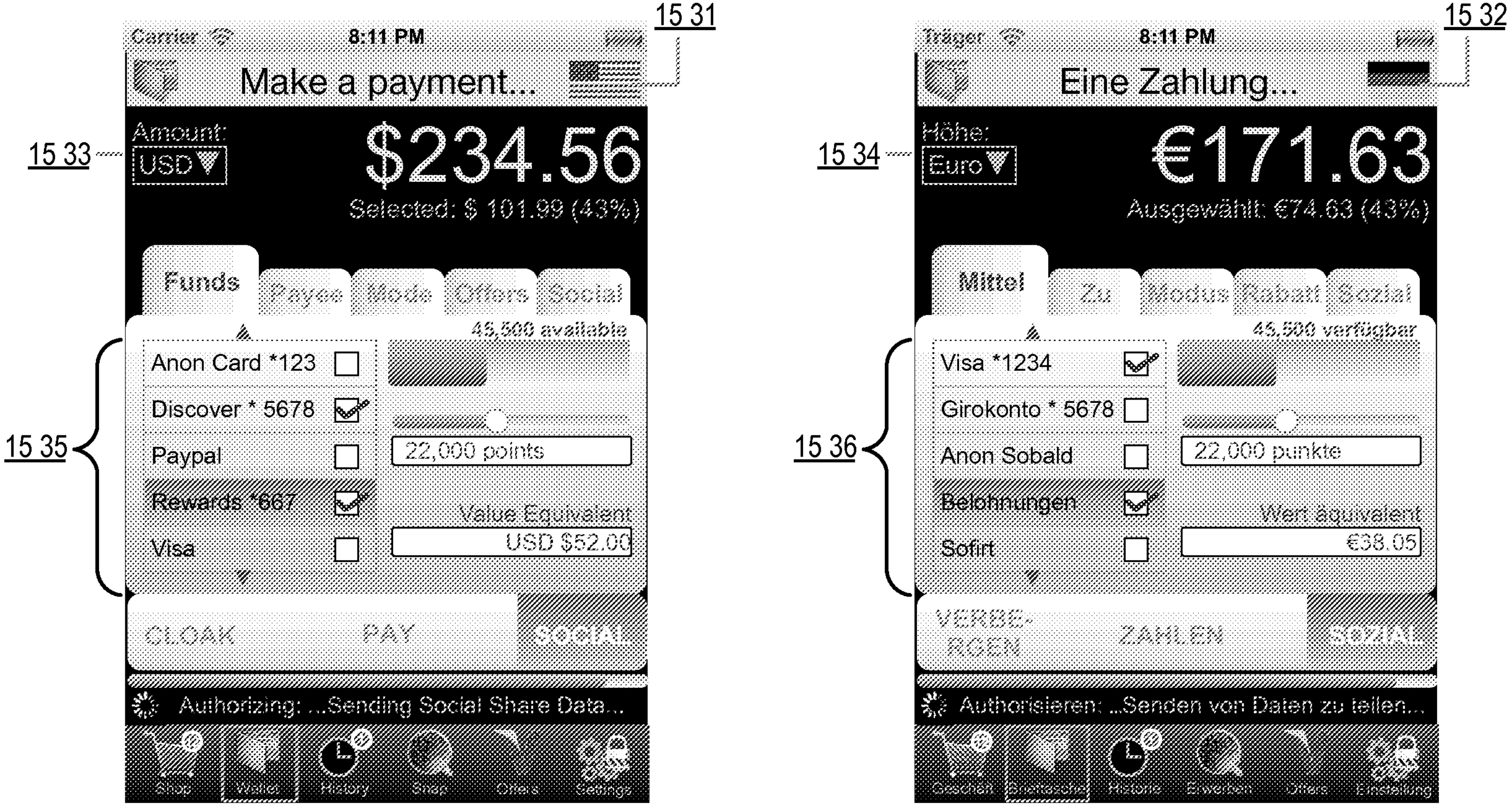


FIGURE 15A

Example: Virtual Wallet Mobile App - Payment Mode



$234.56
Selected: $ 101.99 (43%)
45,500 available

Anon Card *123 □
Discover * 5678 ☑
Paypal □
Rewards *667 ☑
Visa □

22,000 points
Value Equivalent
USD $52.00

Authorizing: ...Sending Social Share Data...

€171.63
Ausgewählt: €74.63 (43%)
45,500 verfügbar

Visa *1234 ☑
Girokonto * 5678 □
Anon Sobald □
Belohnungen ☑
Sofirt □

22,000 punkte
Wert äquivalent
€38.05

Authorisieren: ...Senden von Daten zu teilen...

FIGURE 15B

Example: Virtual Wallet Mobile App - Dynamic Payment Optimization

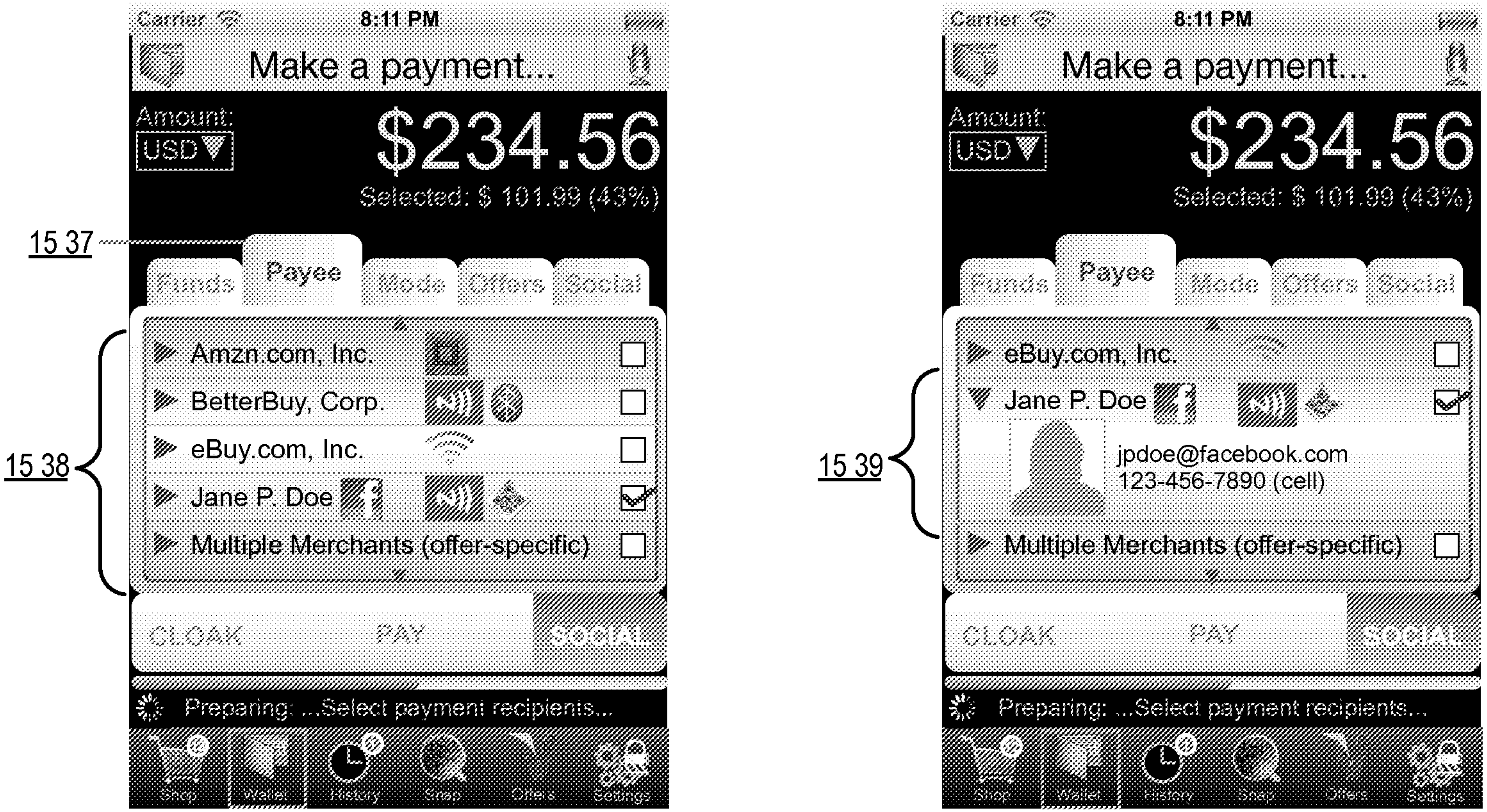
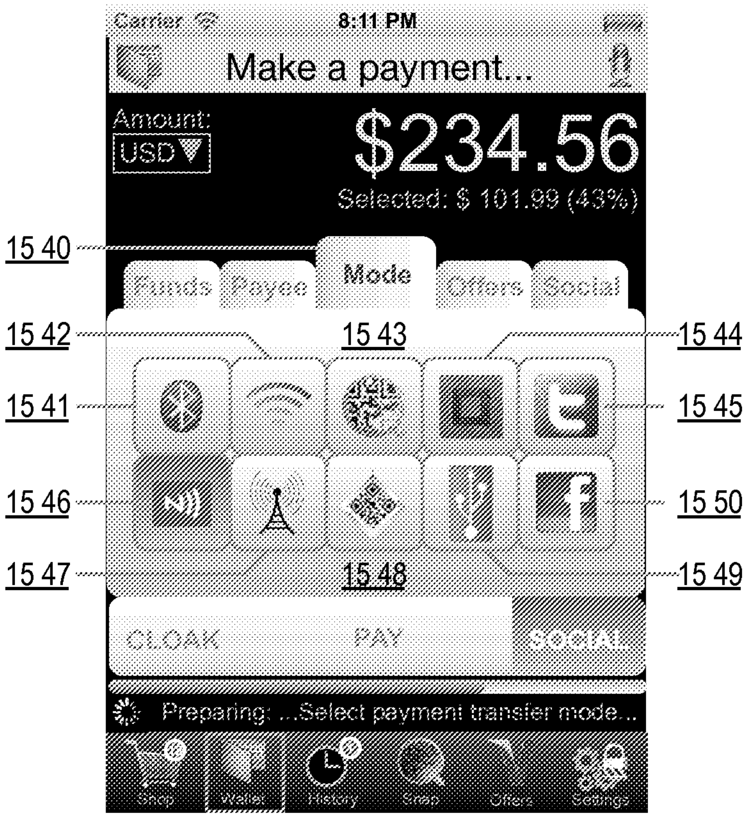


FIGURE 15C

Example: Virtual Wallet Mobile App



Make a payment...

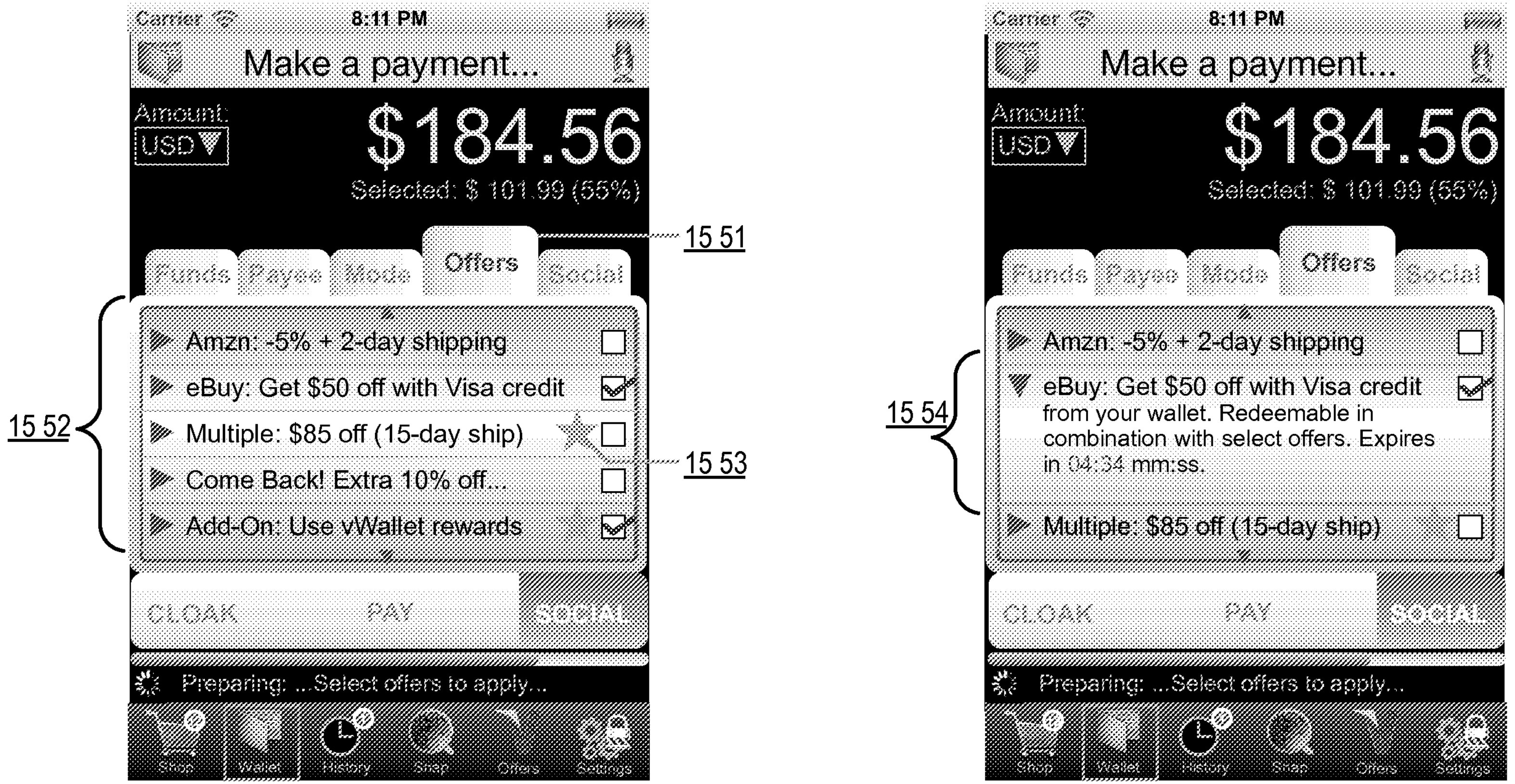
Amount: USD▼ $234.56

Selected: $ 101.99 (43%)

Preparing: ...Select payment transfer mode...

FIGURE 15D

Example: Virtual Wallet Mobile App



Amzn: -5% + 2-day shipping □
eBuy: Get $50 off with Visa credit ☑
Multiple: $85 off (15-day ship) □
Come Back! Extra 10% off... □
Add-On: Use vWallet rewards ☑

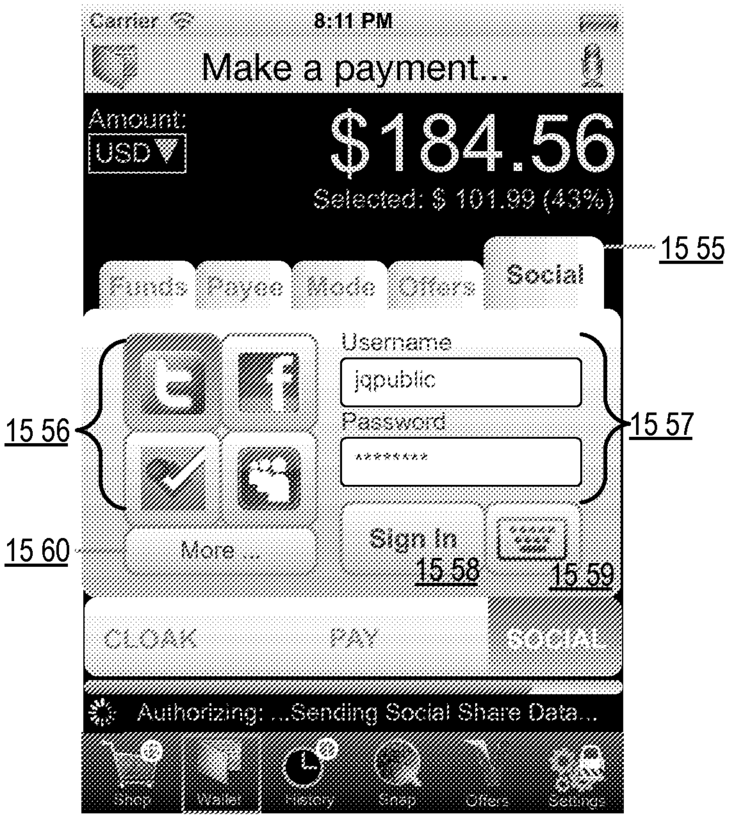
Preparing: ...Select offers to apply...

Amzn: -5% + 2-day shipping □
eBuy: Get $50 off with Visa credit from your wallet. Redeemable in combination with select offers. Expires in 04:34 mm:ss. ☑
Multiple: $85 off (15-day ship) □

Preparing: ...Select offers to apply...

FIGURE 15E

Example: Virtual Wallet Mobile App



Make a payment...

Amount: USD ▼ $184.56

Selected: $ 101.99 (43%)

Username jqpublic

Password ********

Authorizing: ...Sending Social Share Data...

FIGURE 15F

Example: Virtual Wallet Mobile App

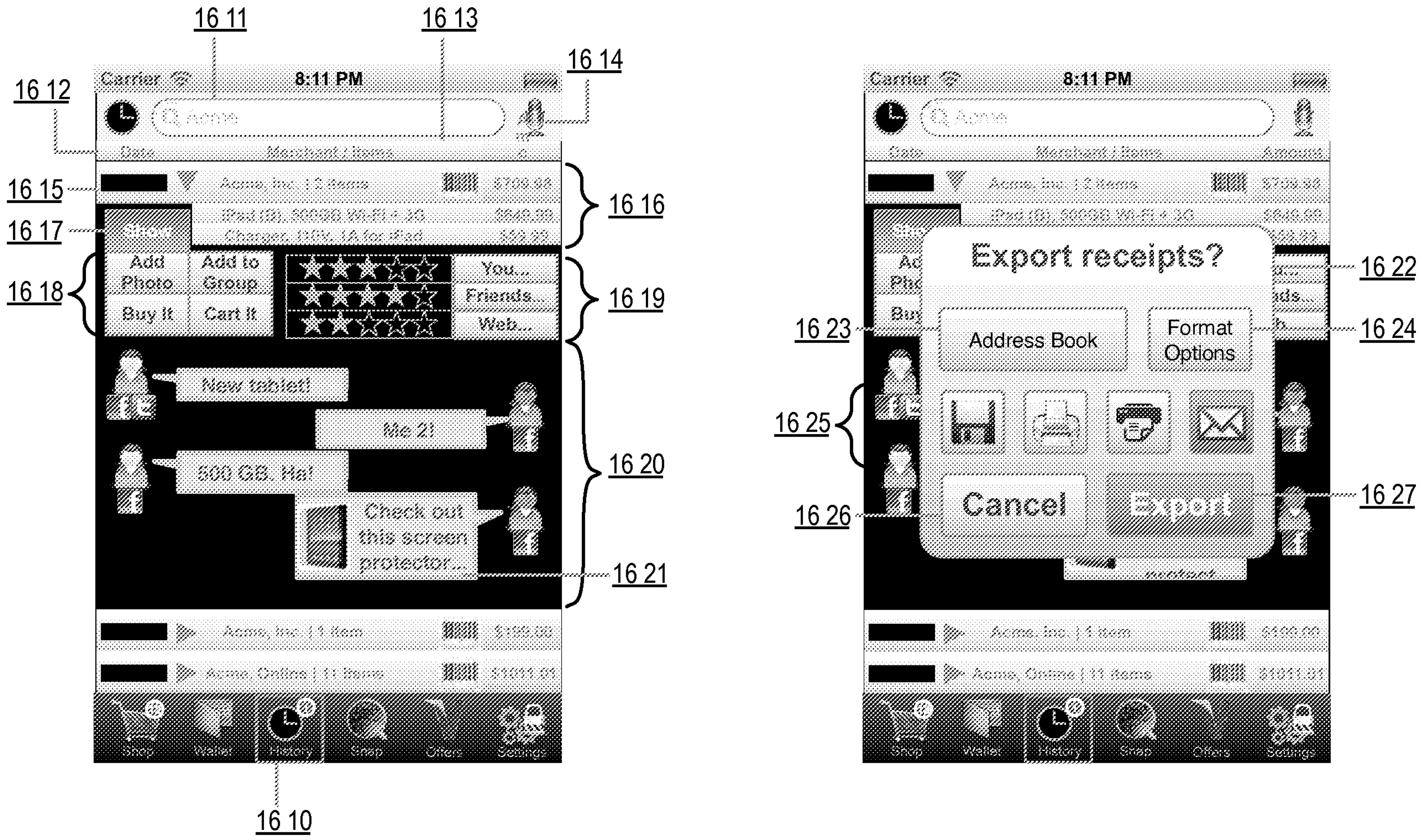
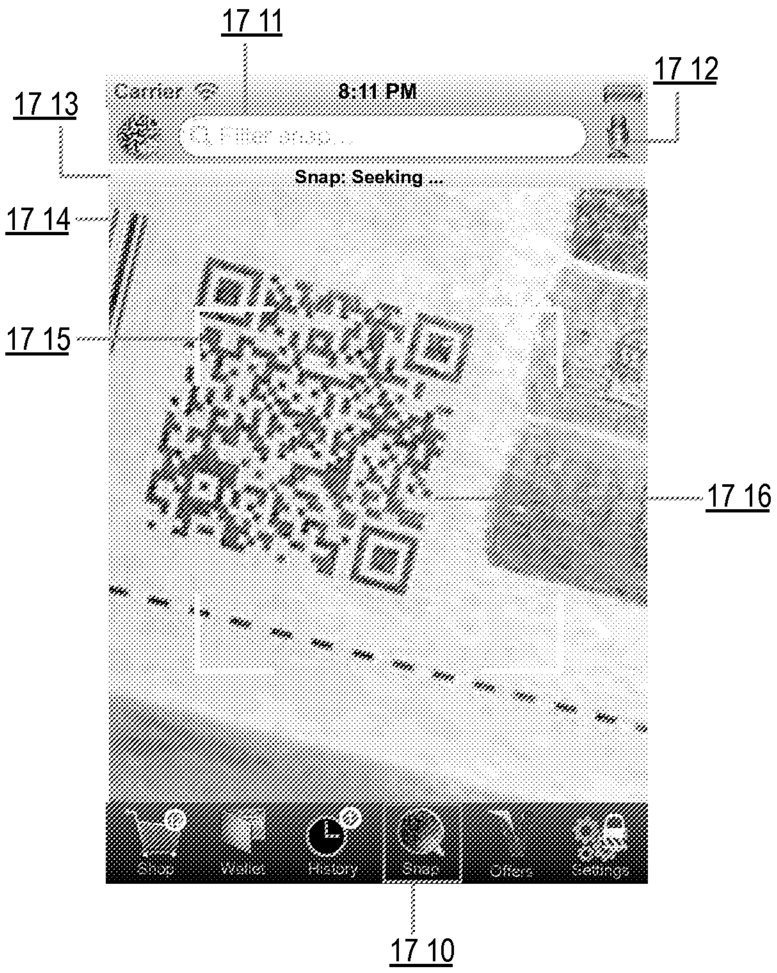


FIGURE 16

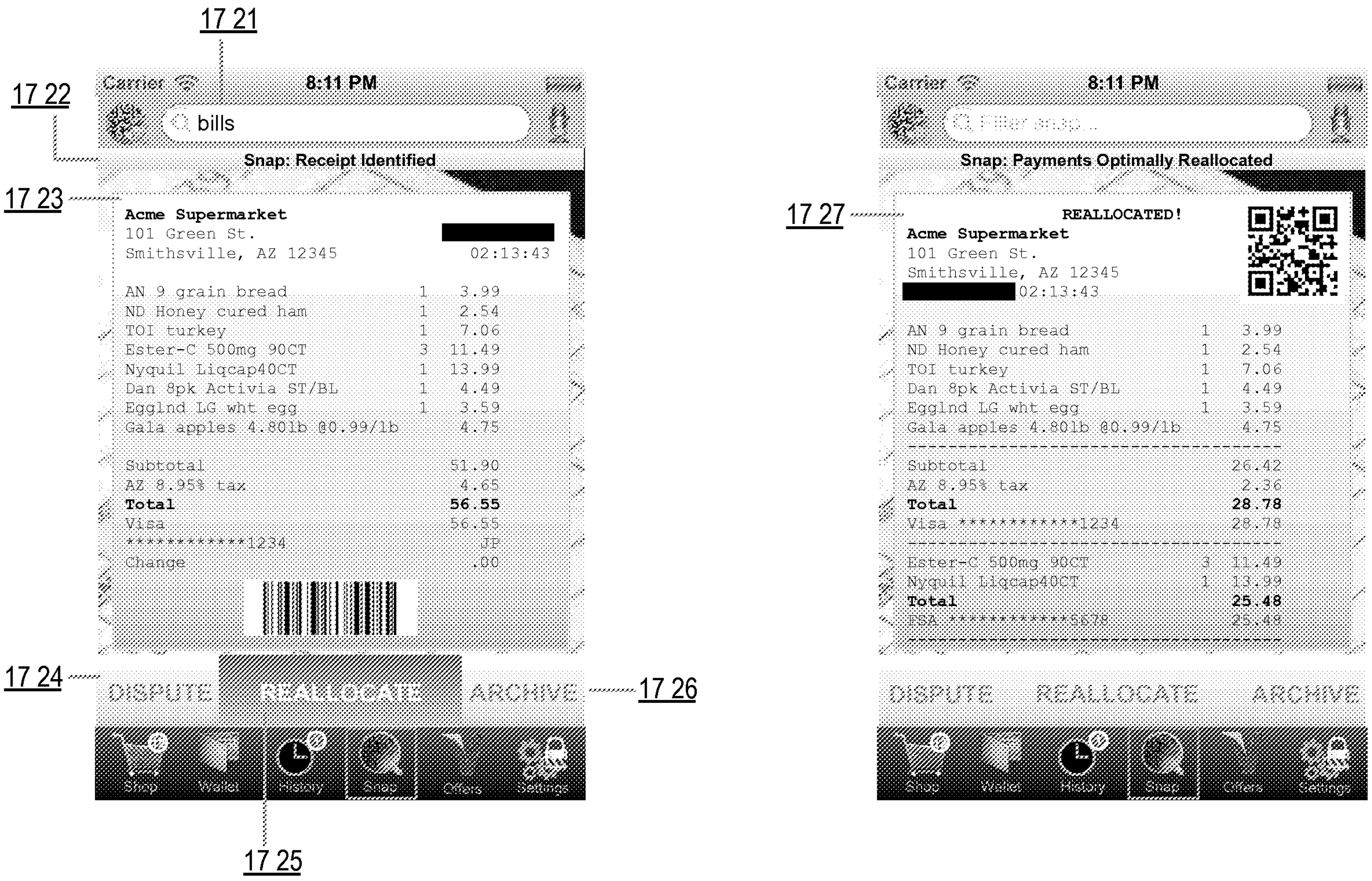
Example: Virtual Wallet Mobile App - History



Snap: Seeking ...

FIGURE 17A

Example: Virtual Wallet Mobile App - Snap Mode



Snap: Receipt Identified

```
Acme Supermarket
101 Green St.                         [redacted]
Smithsville, AZ 12345                 02:13:43

AN 9 grain bread                   1    3.99
ND Honey cured ham                 1    2.54
TOI turkey                         1    7.06
Ester-C 500mg 90CT                 3   11.49
Nyquil Liqcap40CT                  1   13.99
Dan 8pk Activia ST/BL              1    4.49
Egglnd LG wht egg                  1    3.59
Gala apples 4.801b @0.99/1b             4.75

Subtotal                               51.90
AZ 8.95% tax                            4.65
Total                                  56.55
Visa                                   56.55
************1234                          JP
Change                                   .00
```

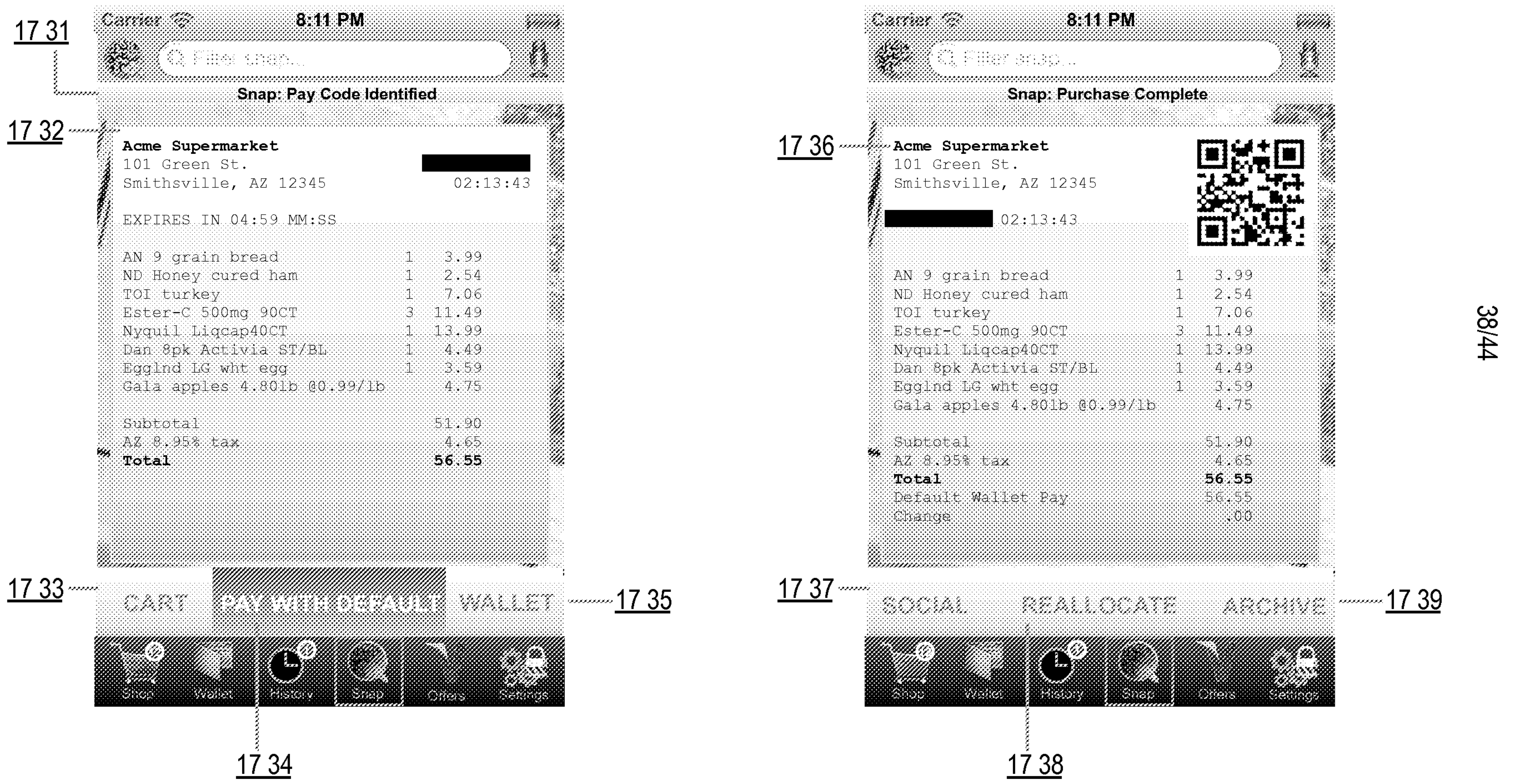
Snap: Payments Optimally Reallocated

```
                 REALLOCATED!
Acme Supermarket
101 Green St.
Smithsville, AZ 12345
[redacted]02:13:43

AN 9 grain bread                   1    3.99
ND Honey cured ham                 1    2.54
TOI turkey                         1    7.06
Dan 8pk Activia ST/BL              1    4.49
Egglnd LG wht egg                  1    3.59
Gala apples 4.801b @0.99/1b             4.75
-------------------------------------------
Subtotal                               26.42
AZ 8.95% tax                            2.36
Total                                  28.78
Visa ************1234                  28.78
-------------------------------------------
Ester-C 500mg 90CT                 3   11.49
Nyquil Liqcap40CT                  1   13.99
Total                                  25.48
FSA ************5678                   25.48
```

FIGURE 17B

Example: Virtual Wallet Mobile App - Snap Mode



Snap: Pay Code Identified

```
Acme Supermarket
101 Green St.                          [redacted]
Smithsville, AZ 12345                   02:13:43

EXPIRES IN 04:59 MM:SS

AN 9 grain bread               1   3.99
ND Honey cured ham             1   2.54
TOI turkey                     1   7.06
Ester-C 500mg 90CT             3  11.49
Nyquil Liqcap40CT              1  13.99
Dan 8pk Activia ST/BL          1   4.49
Egglnd LG wht egg              1   3.59
Gala apples 4.801b @0.99/1b        4.75

Subtotal                          51.90
AZ 8.95% tax                       4.65
Total                             56.55
```

Snap: Purchase Complete

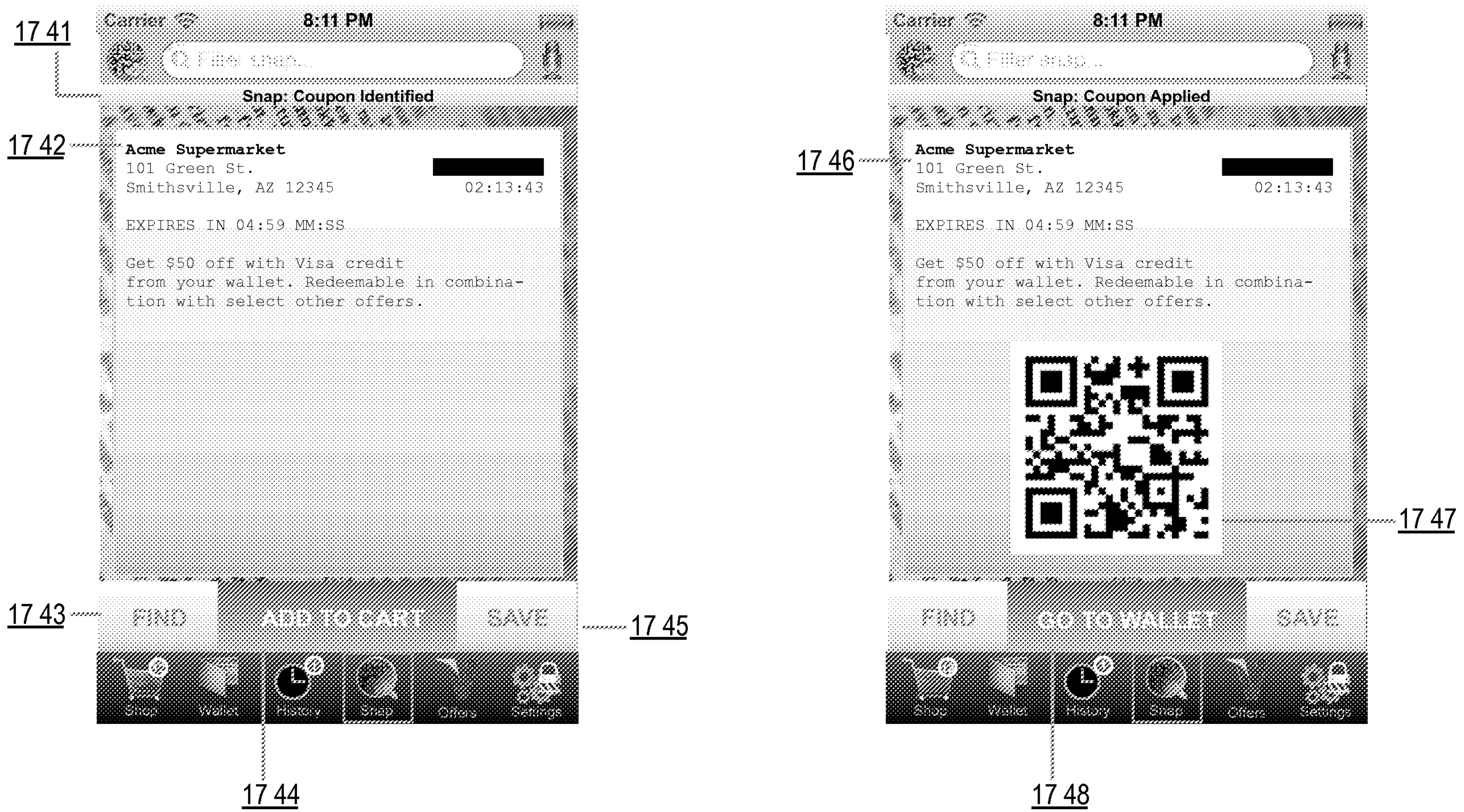
```
Acme Supermarket
101 Green St.
Smithsville, AZ 12345
[redacted] 02:13:43

AN 9 grain bread               1   3.99
ND Honey cured ham             1   2.54
TOI turkey                     1   7.06
Ester-C 500mg 90CT             3  11.49
Nyquil Liqcap40CT              1  13.99
Dan 8pk Activia ST/BL          1   4.49
Egglnd LG wht egg              1   3.59
Gala apples 4.801b @0.99/1b        4.75

Subtotal                          51.90
AZ 8.95% tax                       4.65
Total                             56.55
Default Wallet Pay                56.55
Change                              .00
```

FIGURE 17C

Example: Virtual Wallet Mobile App - Snap Mode



Acme Supermarket
101 Green St.
Smithsville, AZ 12345
02:13:43

EXPIRES IN 04:59 MM:SS

Get $50 off with Visa credit from your wallet. Redeemable in combination with select other offers.

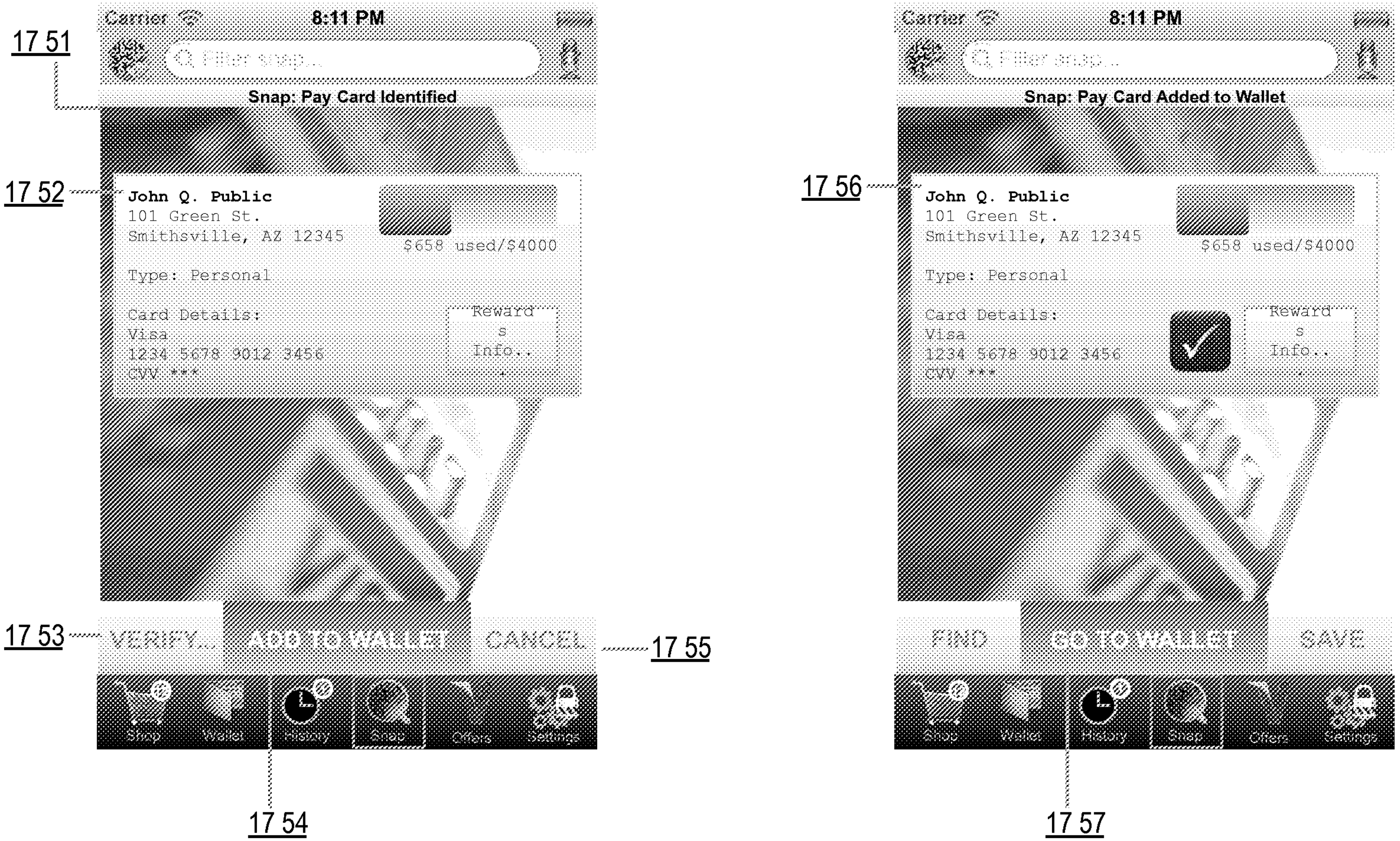
Acme Supermarket
101 Green St.
Smithsville, AZ 12345
02:13:43

EXPIRES IN 04:59 MM:SS

Get $50 off with Visa credit from your wallet. Redeemable in combination with select other offers.

FIGURE 17D

Example: Virtual Wallet Mobile App - Snap Mode



John Q. Public
101 Green St.
Smithsville, AZ 12345

$658 used/$4000

Type: Personal

Card Details:
Visa
1234 5678 9012 3456
CVV ***

Reward
s
Info..

John Q. Public
101 Green St.
Smithsville, AZ 12345

$658 used/$4000

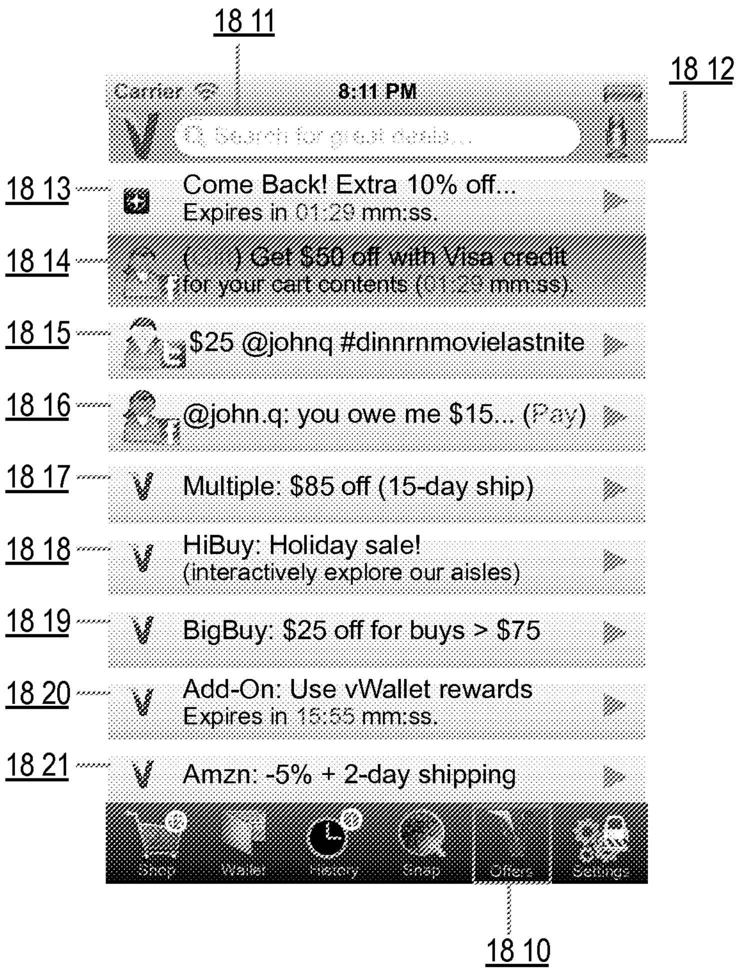
Type: Personal

Card Details:
Visa
1234 5678 9012 3456
CVV ***

Reward
s
Info..

FIGURE 17E

Example: Virtual Wallet Mobile App - Snap Mode



Carrier 8:11 PM

Search for great deals...

Come Back! Extra 10% off...
Expires in 01:29 mm:ss.

( ) Get $50 off with Visa credit
for your cart contents (01:29 mm:ss).

$25 @johnq #dinnrnmovielastnite

@john.q: you owe me $15... (Pay)

Multiple: $85 off (15-day ship)

HiBuy: Holiday sale!
(interactively explore our aisles)

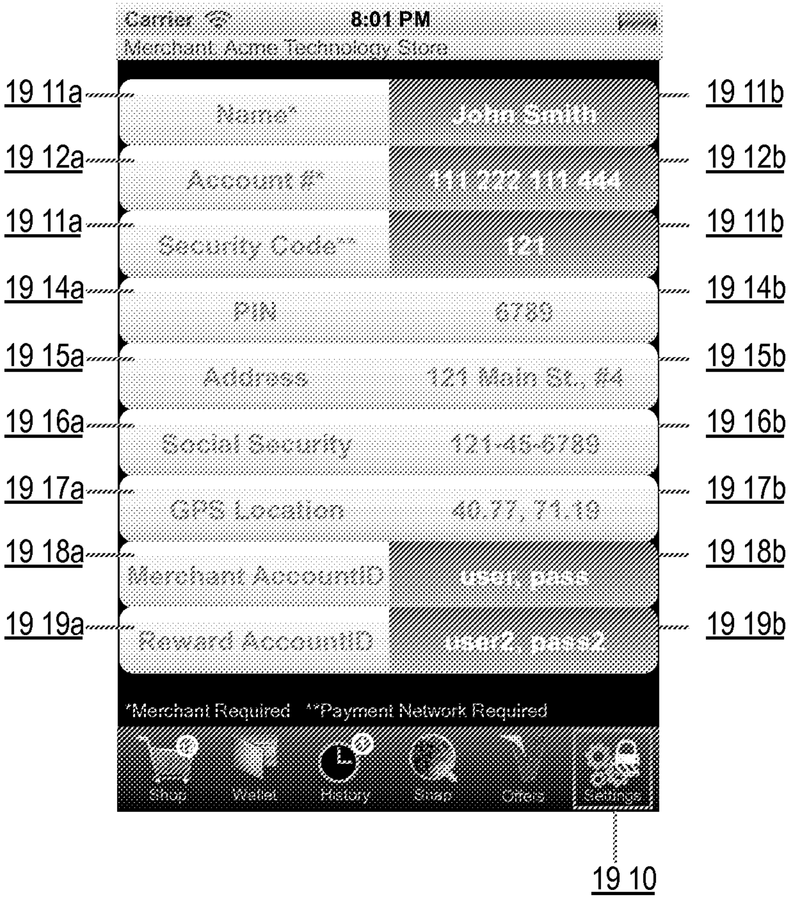
BigBuy: $25 off for buys > $75

Add-On: Use vWallet rewards
Expires in 15:55 mm:ss.

Amzn: -5% + 2-day shipping

FIGURE 18

Example: Virtual Wallet Mobile App - Offers



Carrier 8:01 PM

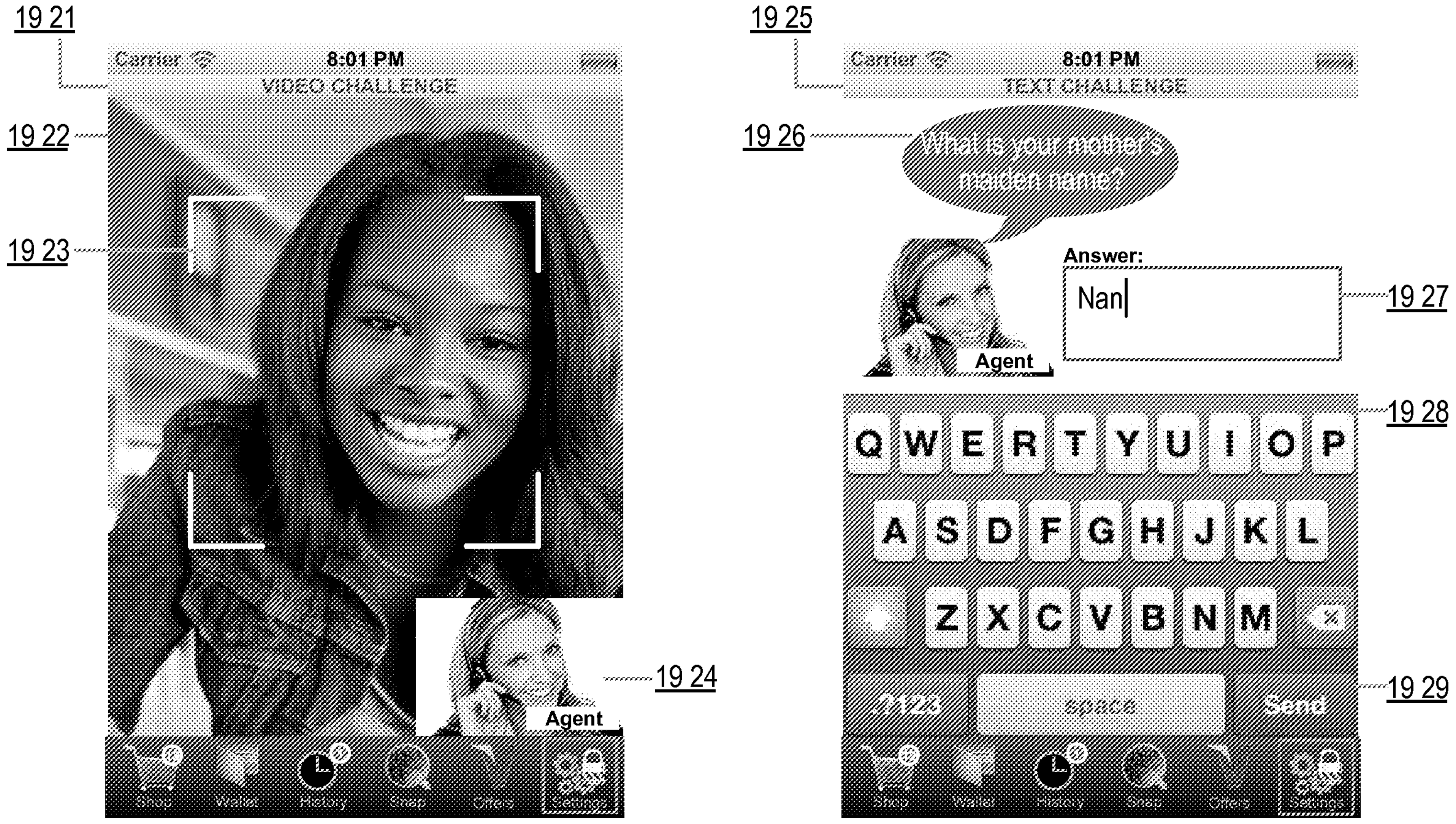
Merchant: Acme Technology Store

Name*	John Smith
Account #*	111 222 111 444
Security Code**	121
PIN	6789
Address	121 Main St., #4
Social Security	121-45-6789
GPS Location	40.77, 71.19
Merchant AccountID	user, pass
Reward AccountID	user2, pass2

*Merchant Required **Payment Network Required

FIGURE 19A

Example: Virtual Wallet Mobile App



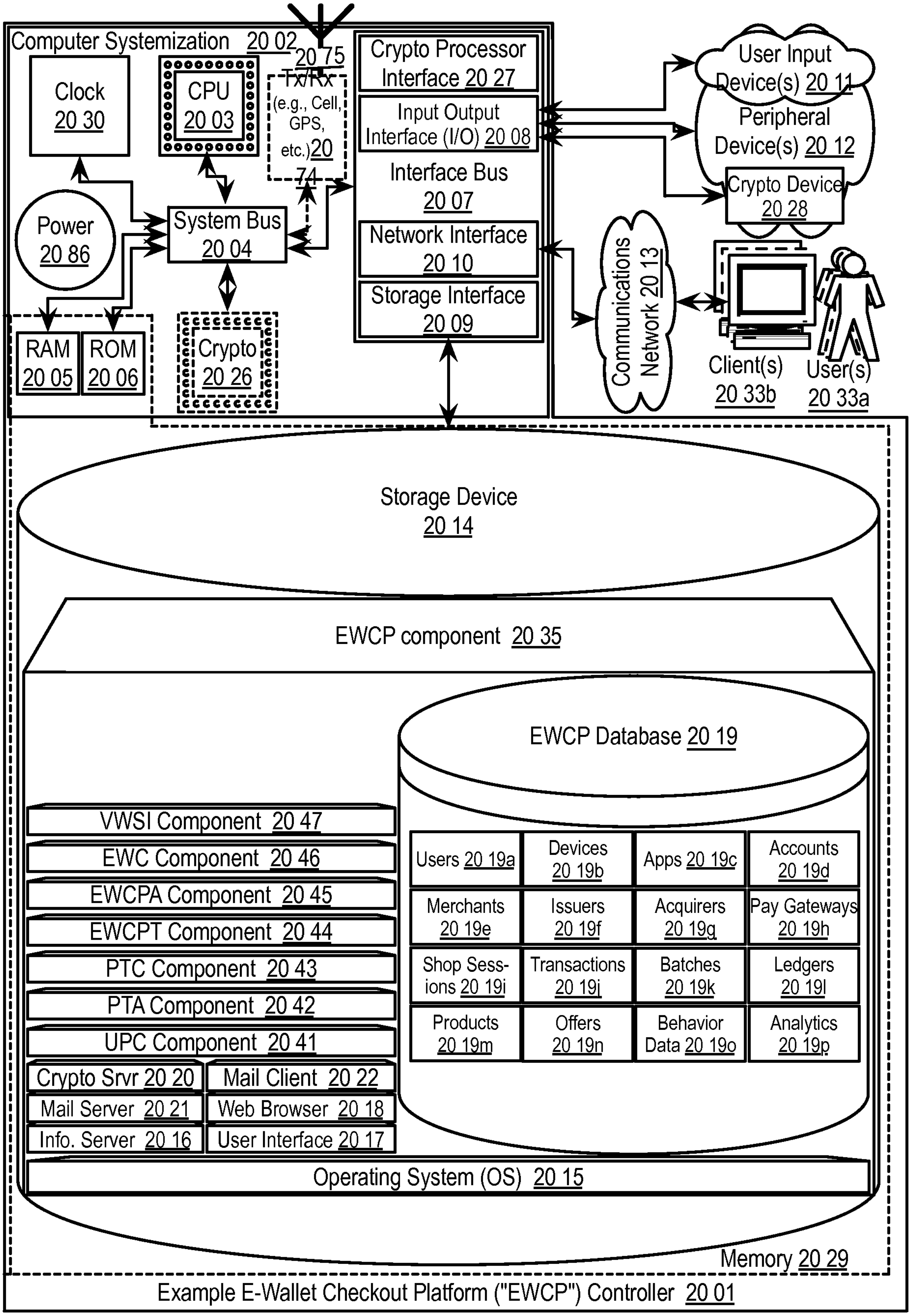
What is your mother's maiden name?

Answer:
Nan

FIGURE 19B

Example: Virtual Wallet Mobile App

FIGURE 20


Computer Systemization 20 02
Clock 20 30
CPU 20 03
20 75
Tx/Rx (e.g., Cell, GPS, etc.) 20 74
Crypto Processor Interface 20 27
Input Output Interface (I/O) 20 08
Interface Bus 20 07
Network Interface 20 10
Storage Interface 20 09
Power 20 86
System Bus 20 04
RAM 20 05
ROM 20 06
Crypto 20 26
User Input Device(s) 20 11
Peripheral Device(s) 20 12
Crypto Device 20 28
Communications Network 20 13
Client(s) 20 33b
User(s) 20 33a
Storage Device 20 14
EWCP component 20 35
EWCP Database 20 19
VWSI Component 20 47
EWC Component 20 46
EWCPA Component 20 45
EWCPT Component 20 44
PTC Component 20 43
PTA Component 20 42
UPC Component 20 41
Crypto Srvr 20 20
Mail Client 20 22
Mail Server 20 21
Web Browser 20 18
Info. Server 20 16
User Interface 20 17
Users 20 19a
Devices 20 19b
Apps 20 19c
Accounts 20 19d
Merchants 20 19e
Issuers 20 19f
Acquirers 20 19g
Pay Gateways 20 19h
Shop Sessions 20 19i
Transactions 20 19j
Batches 20 19k
Ledgers 20 19l
Products 20 19m
Offers 20 19n
Behavior Data 20 19o
Analytics 20 19p
Operating System (OS) 20 15
Memory 20 29
Example E-Wallet Checkout Platform ("EWCP") Controller 20 01