SNAP MOBILE PAYMENT APPARATUSES, METHODS AND SYSTEMS

[0001] This patent application disclosure document (hereinafter "description" and/or "descriptions") describes inventive aspects directed at various novel innovations (hereinafter "innovation," "innovations," and/or "innovation(s)") 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 patent disclosure document 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. 61/443,624 filed February 16, 2011, entitled "MOBILE CAPTURE CHECKOUT APPARATUSES, METHODS AND SYSTEMS," attorney docket no. P-42032PRV|20270-127PV; United States provisional patent application serial no. 61/512,248 filed July 27, 2011, entitled "SNAP MOBILE PAYMENT APPARATUSES, METHODS AND SYSTEMS," attorney docket no. 10US01|20270-175PV; United States provisional patent application serial no. 61/522,213 filed August 10, 2011, entitled "UNIVERSAL MOBILE PAYMENT PLATFORM APPARATUSES, METHODS AND SYSTEMS," attorney docket no. 10US03|20270-175PV2; and United States provisional patent application serial no. 61/527,576 filed August 25, 2011, entitled "SNAP MOBILE PAYMENT APPARATUSES,

METHODS AND SYSTEMS," attorney docket no. 10US02|20270-175PV1. The entire contents of the aforementioned applications are expressly incorporated by reference herein.

FIELD

[0003] Embodiments of the present invention are directed generally to apparatuses, methods, and systems for electronic purchase transactions, and more particularly, to SNAP MOBILE PAYMENT APPARATUSES, METHODS AND SYSTEMS ("SNAP").

BACKGROUND

[0004] Consumer transactions typically require a customer to select a product from a store shelf or website, and then to check the out at a checkout counter or webpage. Product information is typically selected from a webpage catalog or entered into a point-of-sale terminal device. In a physical retail environemtn product information is automatically entered by scanning an item barcode with an integrated barcode scanner at a point-of-sale register, and the customer is usually provided with a number of payment options, such as cash, check, credit card or debit card. Once payment is made and approved, the point-of-sale register memorializes the transaction in the merchant's computer system, and a receipt is generated indicating the satisfactory consummation of the transaction.

SUMMARY OF THE INVENTION

[0004a] According to a first aspect of the present invention, there is provided a visual snap computer-implemented method, comprising: obtaining a user visual item information request (VITR); extracting a VITR image frame from the VITR obtained via an image acquisition device operatively connected to a user device; extracting VITR field attributes from the VITR image frame through a VITR processing component, wherein the VITR processing component may include barcode recognition and optical character recognition; querying a database with the extracted VITR field attributes; determining a VITR type from results from the querying; providing a visual item information response based on the determined VITR type; wherein a payment code is depicted within the extracted VITR image frame; wherein a purchase transaction request is generated using the identified payment code.

[0004b] According to a second aspect of the present invention, there is provided a visual snap computer-implemented system, comprising: a processor; and a memory disposed in communication with the processor and storing processor-executable instructions to: obtain a user visual item information request (VITR); extract a VITR image frame from the VITR obtained via an image acquisition device operatively connected to a user device; extract VITR field attributes from the VITR image frame through a VITR processing component, wherein the VITR processing component may include barcode recognition and optical character recognition; query a database with the extracted VITR field attributes; determine a VITR type from results from the querying; provide a visual item information response based on the determined VITR type; wherein a payment code is depicted within the extracted VITR image frame; wherein a purchase transaction request is generated using the identified payment code.

[0004c] According to a third aspect of the present invention, there is provided a computer-readable tangible medium storing computer-executable visual snap instructions to: obtain a user visual item information request (VITR); extract a VITR image frame from the VITR obtained via an image acquisition device operatively connected to a user device; extract VITR field attributes from the VITR image frame through a VITR processing component, wherein the VITR processing component may include barcode recognition and optical character recognition; query a database with the extracted VITR field attributes; determine a VITR type from results from the querying; provide a visual item information response based on the determined VITR type; wherein a payment code is depicted within the extracted VITR image frame; wherein a purchase transaction request is generated using the identified payment code.

[0004d] According to a fourth aspect of the present invention, there is provided a computer-readable tangible medium storing computer-executable snap payment instructions to: obtain a user visual item information request (VITR); acquire an VITR image frame via an image acquisition device operatively connected to the user device; extract VITR field attributes from the VITR image frame through a VITR processing component, wherein the VITR processing component may include barcode recognition and optical character recognition; query a database with the extracted VITR field attributes; determine a VITR type from results from the querying; provide a visual item information response based on the determined VITR type.

[0004e] According to a fifth aspect of the present invention, there is provided a visual snap computer-implemented means, comprising means for: obtaining a user visual item information request (VITR); extracting a VITR image frame from the VITR obtained via an image acquisition device operatively connected to a user device; extracting VITR field attributes from the VITR image frame through a VITR processing component, wherein the VITR processing component may include barcode recognition and optical character recognition; querying a database with the extracted VITR field attributes; determining a VITR type from results from the querying; providing a visual item information response based on the determined VITR type; wherein a payment code is depicted within the extracted VITR image frame; wherein a purchase transaction request is generated using the identified payment code.

BRIEF DESCRIPTION OF THE DRAWINGS

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

[0006] FIGURES 1A-F show block diagrams illustrating example aspects of a snap mobile payment-based purchase transaction in some embodiments of the SNAP;

[0007] FIGURES 2A-F shows application user interface diagrams illustrating example features of a snap mobile payment app facilitating snap mobile payment in some embodiments of the SNAP;

[0008] FIGURES 3A-E show application user interface diagrams illustrating example features of a snap mobile payment app for capturing product barcodes, securing user data and preventing fraud in some embodiments of the SNAP;

[0009] FIGURES 4A-D show data flow diagrams illustrating an example snap mobile payment procedure in some embodiments of the SNAP;

[0010] FIGURES 5A-E show logic flow diagrams illustrating example aspects of executing a snap mobile payment in some embodiments of the SNAP, e.g., a Snap Mobile Payment Execution ("SMPE") component 500;

[0011] FIGURES 6A-B show logic flow diagrams illustrating example aspects of processing a Quick Response code in some embodiments of the SNAP, e.g., a Quick Response Code Processing ("QRCP") component 600;

[0012] FIGURE 7 shows a user interface diagram illustrating an overview of example features of virtual wallet applications in some embodiments of the SNAP;

[0013] FIGURES 8A-G show user interface diagrams illustrating example features of virtual wallet applications in a shopping mode, in some embodiments of the SNAP;

[0014] FIGURES 9A-F show user interface diagrams illustrating example features of virtual wallet applications in a payment mode, in some embodiments of the SNAP;

[0015] FIGURE 10 shows a user interface diagram illustrating example features of virtual wallet applications, in a history mode, in some embodiments of the SNAP;

[0016] FIGURES 11A-F show user interface diagrams illustrating example features of virtual wallet applications in a snap mode, in some embodiments of the SNAP;

[0017] FIGURE 12 shows a user interface diagram illustrating example features of virtual wallet applications, in an offers mode, in some embodiments of the SNAP;

[0018] FIGURES 13A-B show user interface diagrams illustrating example features of virtual wallet applications, in a security and privacy mode, in some embodiments of the SNAP;

[0019] FIGURE 14 shows a block diagram illustrating embodiments of a SNAP controller.

[0020] 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

SNAP MOBILE PAYMENT (SNAP)

[0021] The SNAP MOBILE PAYMENT APPARATUSES, METHODS AND SYSTEMS (hereinafter "SNAP") transform real-time-generated merchant-product Quick Response codes, via SNAP components, into virtual wallet card-based transaction purchase notifications. FIGURES 1A-F show block diagrams illustrating example aspects of a snap mobile payment-based purchase transaction in some embodiments of the SNAP. With reference to FIGURE 1A, in some implementations, a user, e.g., 101a-b, may wish to purchase products at a merchant store, e.g., 103a, or at a merchant website, e.g., 103b. For example, at a merchant store, the user may scan barcodes for a number of products, e.g., 102a, at a point-of-sale ("POS") terminal in the store, e.g., 103a, and then indicate that the user wishes to checkout the scanned items. In some implementations, the POS terminal may generate a Quick Response ("QR") code, e.g., 105a, including information on the scanned product items, as well as merchant information for processing the purchase transaction via a payment network. The user may capture an image of the QR code generated by the POS terminal using a user device, such as a smartphone. For example, the user device may have executing on it an app for snapping the merchant-product QR code. The user device may utilize the information extracted from the QR code, along with information on a virtual wallet tied to the user device to initiate a purchase transaction. For example, the user device may utilize the product and merchant information extracted from the QR code, and financial

payment information from the virtual wallet, to create a purchase transaction request, and submit the request to a payment network (e.g., credit card processing network).

[0022] In some implementations, the user device may utilize methods alternative to capture of a QR code to obtain information from the POS terminal. For example, the POS terminal may communicate the information required for submitting a purchase transaction request to a payment network to user device via Bluetooth™, Wi-Fi, SMS, text message, electronic mail, and/or other communication methods.

[0023] In some implementations, a user 101b may wish to checkout items stored in a virtual shopping cart on an online shopping website, e.g., 102b. For example, the user may be viewing the website using a secure display (e.g., that is part of a trusted computing device of the user). Upon indicating that the user wishes to checkout the items in the virtual shopping cart, the website may provide a QR code including information on the products in the virtual shopping cart and merchant information. For example, in the scenario where the user utilizes a secure display, the QR code may be displayed at a random position within the secure display for security purposes. The user may capture a snapshot of the displayed QR code, and utilize payment information from the virtual wallet associated with the user device to create a purchase transaction request for processing by the payment network. Upon completion of the purchase transaction, the payment network may provide a purchase receipt, e.g., 107 directly to the user device 106, the POS terminal in the store and/or the secure display (for the secure online shopping scenario) as confirmation of completion of transaction processing. Thus, in some implementations, the merchant may be shielded from obtaining personal and/or private information of the user while processing the purchase

transaction, while ensuring integrity of the user's virtual wallet using a secure display for presenting the merchant-product QR code.

[0024] In various implementations, such payment processing may be utilized for a wide variety of transactions. For example, a user dining at a restaurant may obtain a bill including a QR pay code including detail on the dining charges included in the bill, and a merchant ID for the restaurant. The user may take a snapshot of the restaurant bill using the user's smartphone, and utilize the user's virtual wallet to pay for the restaurant bill, without revealing any financial or personal information about the user to the restaurant.

[0025] With reference to FIGURE 1B, in some implementations, e.g., 110, a user 111 may wish to checkout items stored in a (virtual) shopping cart in a (online) shopping store, e.g., 112, using a reverse snap mobile payment procedure. For example, the user may be viewing the website using a secure display that is part of a trusted computing device of the user, e.g., 113, or via a POS terminal in a brick-and-mortar store. Upon indicating that the user wishes to checkout the items in the virtual shopping cart, the user may generate, e.g., 114, a QR code 115b via a mobile app on the user's mobile device including information on the user payment methods, offers, rewards, and/or other aspects connected to the user's virtual wallet. The user may provide the QR code displayed on the user's mobile device to a webcam (or other QR code capture device and/or mechanism) installed on the trusted computing device (or POS terminal). The user's trusted computing device or POS terminal may obtain a snapshot of the QR code generated by the user's mobile device, e.g., 116, and utilize payment information from QR code generated by the user to create a purchase transaction request for processing

by the payment network. Upon completion of the purchase transaction, the payment network may provide a purchase receipt directly to the user mobile device, the POS terminal in the store and/or the secure display (for the online shopping scenario) as confirmation of completion of transaction processing. Thus, in some implementations, the user may be able to utilize a QR code generated by the user's mobile device as a replacement for a plastic payment card (e.g., credit, debit, prepaid card), or as a substitute for other financial information transport mechanisms such as near-field communications, Bluetooth®, etc. In some implementations, the QR code may be representative of a one-time anonymized credit card number (e.g., see the description associated with FIGURE 3B).

[0026] In some implementations, a first user 121b may desire to pay a second user 121a an amount of money (or a value equivalent, e.g., virtual currency, alternate real currency, rewards, miles, points, etc.), e.g., P2P snap mobile payment 120. The second user 121a may generate a limited-time-validity QR code, e.g., 122, including information on the amount of money to be transferred, as well as a privacy token/alias linked to a financial account of the second user. The second user may display the QR code generated to the first user (e.g., by holding the second user's mobile phone displaying the QR code to the first user; sending the QR code by email, social network message, tweet, etc.). The first user may take a snapshot of the QR code using the first user's mobile phone, e.g., 123, and utilize the amount of money, the second user's privacy token/alias linking to a financial account, and the first user's virtual wallet linked to the first user's mobile phone, to generate a purchase transaction request for processing by the payment network. Upon completion of the transaction, the payment network may provide transaction notification receipts to the users who are parties to the transaction.

In alternate implementations, the two users may share the data encoded in the QR code via methods alternate to the QR code, including but not limited to: near-field communications (NFC), Wi-Fi™, Bluetooth™, cellular network, SMS, email, text messages and/or the other communication protocols.

[0027] In general, it is to be understood that such tokens, alias and/or handles may be advantageously utilized in various implementations of snap mobile payment. For example, a user wishing to engage in reverse snap mobile payment procedure (see, e.g., FIGURE 1B, element 110) may generate a QR code embodying information on a handle pointing to financial payment information stored on a server of a payment network system. For example, some implementations of snap mobile payment may utilize, to generate and/or process handles, a payment tokenization procedure similar to that described in U.S. application no. 13/153,301, titled "Payment tokenization apparatuses, methods and systems," the entire contents of which are expressly incorporated by reference herein. Further, in some implementations, the handle may encode information according to a compact messaging protocol, such as described in U.S. patent no. 6,837,425, titled "Compact protocol and solution for substantially offline messaging between portable consumer device and based device," the entire contents of which are expressly incorporated by reference herein. In some reverse snap mobile implementations, the user may provide the QR code embodying the handle and displayed on the user's mobile device to a webcam (or other QR code capture device and/or mechanism) installed on the trusted computing device (or POS terminal). The user's trusted computing device or POS terminal may obtain a snapshot of the QR code generated by the user's mobile device, e.g., 116, and provide the handle extracted from the QR code to a merchant server for purchase transaction request processing by the

payment network. The merchant server may generate a card authorization request (such as described further below in the discussion with reference to FIGURE 4A) for processing the purchase transaction using the handle, and may provide the card authorization request to a payment network. Upon completion of the purchase transaction, the payment network may provide a purchase receipt directly to the user mobile device, the POS terminal in the store, and/or the secure display (e.g., for the online shopping scenario) as confirmation of completion of transaction processing using the handle.

[0028] In some implementations, a user alert mechanism may be built into the snap mobile payment purchase transaction process flow. For example, in some implementations, a merchant server may embed a URL specific to the transaction into the card authorization request. For example, in some implementations, a POS terminal, remote device and/or desktop computer may embed the URL into optional level 3 data in the card authorization request. The URL may point to a webpage stored on the merchant's server dedicated to the transaction that is the subject of the card authorization request. For example, the webpage 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 implementations, 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.

[0029] In some implementations, a number of users may engage in group pay via snap mobile payment to split a tender, e.g., 130. In some implementations, one of the users 131a may obtain a snapshot, e.g., 132, of a QR pay code, e.g., 134, generated at a POS terminal (or, e.g., presented on paper such as a dining bill), e.g., 133. The user may in turn generate a QR split pay code, embodying information on the amounts that the tender has been split into. The user 131a may present the split tender QR code 135 to the other users 131b-c, who may obtain snapshots of the split tender QR code, e.g., 136. In some implementations, the users 131b-c may be reimbursing the user 131a for payment of the original QR code via the payment network, or the users 131b-c may be making direct payments via the split tender QR code to the merchant (e.g., when the user 131a took a snapshot of the merchant's QR code, no payment processing occurred immediately). In some implementations, the merchant may directly provide a split tender QR code for the users 131a-c.

[0030] In some implementations, group mobile payment may be implemented, instead of utilizing QR codes, via use of an alternate communication mechanism. For example, in some implementations, the POS terminal 133 may utilize a communication protocol such as Bluetooth™ to communicate with the users 131a-c. The POS terminal may, serially or in parallel, establish separate communication sessions with each of the users. Via the separate communication sessions that POS terminal may transmit the product and/or merchant data required by the users' devices to generate individual purchase transaction processing requests. Thus, via the separate communication

sessions, the POS terminal may split the group tender associated with the users 131a-c into individual payment amounts.

[0031] With reference to FIGURE 1C, in some implementations, snap mobile payment methods may be utilized for authentication/verification purposes, and for providing digital consent for disclosure of personal and/or private information. For example, a user 142 visiting his/her doctor 143 may be required to provide informed consent to disclosing personal information (e.g., medical records) to the doctor. The doctor's terminal may generate a QR code embodying the doctor's digital certificate as well as information on the type/content of medical records of the user that are requested, e.g., 144. The user may snap the QR code via the user's mobile device. The user's mobile device may generate a request for records release according to the QR code, and also serve as verification that the request is obtained from a personal trusted device (e.g., the user's mobile device). In alternate implementations, the user may be able to select the personal information that the user would like to reveal to the healthcare provider, and the user's mobile device may generate a QR code for the doctor's terminal to obtain a snapshot for retrieving the user's medical information. In some implementations, the QR code may also include payment information (e.g., the user's pay account information, or the doctor's acquirer information) along with the information on controlled release of personal information.

[0032] In some implementations, the SNAP may facilitate P2P transactions via pre-filled, modifiable QR payment codes, e.g., 150. For example, a first user having a public profile page, e.g., 151, may place an image of a QR code in the public profile, e.g., 152. For example, the QR code may include a predetermined payment amount for a

purchase transaction initiated by capturing a snapshot of the QR code. In some implementations, the predetermined amount may be $0 (e.g., a $0 QR pay code). A second user may capture a snapshot of the QR pay code using a mobile device, and may set an amount that the second user would like to pay the first user via the second user's mobile device. The second user's mobile device may provide the information encoded within the QR code along with the second-user-chosen payment amount to a payment network for transaction processing.

[0033] It is to be understood that the various aspects described herein of snap mobile payment may be utilized for any controlled exchange of information and/or payment. For example, with reference to FIGURE 1D, in some implementations, a user may obtain pay-per-view programming via snap mobile payment, e.g., 160. For example, a television display may provide an advertisement including programming information, e.g., 162, as well as a QR pay code for obtain the programming content, e.g., 161. The QR code may include information identifying the programming information, as well as information identifying the television subscriber account information, television machine address, and/or the like. The user may obtain a snapshot of the QR code, and provide the information embodied in the QR code along with information fro the user's mobile device (e.g., subscriber account number linked to the user's virtual wallet, pay account information, and/or the like). Upon processing of the payment information by the payment network, the payment network may provide an indication to the television-programming provider of the payment completion, and the television-programming provider may stream the programming content to the user's television. As another example, a similar flow may be utilized for in-flight entertainment, e.g., 170, wherein an in-flight screen may provide programming

information 172, as well as a QR pay code 171 for the user to snap for in-flight entertainment initiation. As another example, a billboard, wall hanging, poster, in-store advertisement, hoarding, etc., e.g., 180, may include an offer for a product/service, and a QR code including merchant information and product information identifying a purchase amount, and/or the like. The user may snap the QR code with the user's mobile device linked to the user's virtual wallet to purchase the product and/or service, and, if applicable, the product may be directly shipped to the user's address as specified by the purchase information exchanged with the payment network as part of the purchase request sent by the user's mobile device. As another example, newspapers, e.g., 185, may include offers, advertisements, job postings, and/or the like including QR codes, e.g., 186, embodying the information necessary for the user to initiate a purchase transaction with the payment network. It is to be understood that any aspects of implementing snap mobile payment discussed in any of the implementations herein, and/or their equivalents, may be utilized in any other implementations discussed herein, and/or their equivalents.

[0034] With references to FIGURES 1E-F, in some implementations, the data required for processing a purchase transaction may be provided via methods alternate to a QR code including, but not limited to: near-field communications (NFC), Wi-Fi™, Bluetooth™, cellular network, SMS, email, text messages and/or the other communication protocols. For example, in some implementations, a user shooping online via a web browser executing on a client device, e.g., 190, may desire to pay for a purchase of items from an online shopping website, e.g., 191. The website may include a user interface element that the user may activate to initiate shopping checkout and payment. Upon the user activating the user element, the client displaying the online

shopping website may provide a message to a server of the merchant to initiate secure purchase transaction processing. The server of the merchant operating the online shopping website may establish a secure connection (e.g., a Secure Sockets Layer connection) to a pay network server of a payment network, e.g., 192. Also, the pay network server may establish a secure connection to the client. For example, the client may include a secure I/O chip that only allows secure connections to be established by the client with pay network servers of the payment network. Via the secure connection, the pay network server may provide an instruction to the client to request the user to launch a virtual wallet mobile app on the user device of the user, see e.g., FIGURE 1F, 196. The client may accordingly provide a request to the user to launch a virtual wallet mobile app on the user device, e.g., 193, of the user. Upon the user launching the virtual wallet mobile app on the user device, the user device and the client may establish a secure connection with each other (e.g., via Bluetooth™, Wi-Fi, cellular, etc.) In some implementations, the client and user device may be preconfigured to rapidly establish the secure communication channel with each other. Via the secure communication channel, the client may provide data to the user's mobile device, or vice versa, to facilitate initiation of the purchase transaction. The virtual wallet app on the user's mobile device (or the client) may then generate a purchase transaction initiation message and provide it to the pay network server for processing the purchase transaction. Upon completion of transaction processing, the pay network server may provide a notification of payment completion to the client, e.g., FIGURE 1F, 197, or to the user device.

[0035] FIGURES 2A-F show application user interface diagrams illustrating example features of a snap mobile payment app facilitating snap mobile payment in

some embodiments of the SNAP. With reference to FIGURE 2A, in some implementations, a user may desire to checkout one or items stored in a virtual shopping cart of an online merchant website. For example, the user may be utilizing a browser application, e.g., 201, to visualize a checkout page of the merchant website, e.g., 202. The checkout webpage may depict details of the checkout order, e.g., 203, and may provide one or more options for the user to provide payment for the purchase of the store items. In some implementations, the checkout webpage may include an option to pay for the purchase using a snap mobile payment procedure, e.g., 204.

[0036] With reference to FIGURE 2B, in some implementations, upon selecting the option to utilize the snap mobile payment procedure, the merchant checkout webpage, e.g., 206, may provide via the browser application 205, a QR code, e.g., 209, including information on the items in the virtual shopping cart as well as merchant information for the payment network to process the purchase transaction (e.g., a privacy token/alias linked to an acquirer financial account of the merchant). In some implementations, the webpage may be displayed via a secure display of a trusted computing device of the user. For example, as a security measure, the position of the QR code frame, e.g., 207, within the display may be randomly varied to prevent a snapshot of the QR code from being obtained by fraudulent means (e.g., tampering with the trusted computing device). In some implementations, a security image, e.g., 208, pre-selected by the user may be displayed on the screen so that the user may verify as being accurate. In some implementations, the image may be encrypted by the SNAP before providing it to the trusted computing device. In some implementations, the trusted computing device may be the only device to hold a decryption key required to decrypt and successfully display the image on the secure display to the user.

[0037] With reference to FIGURE 2C, in some implementations, such merchant-product information embodying QR codes may be utilized by a point-of-sale ("POS") terminal, e.g., 210a-b. For example, in a brick-and-mortar store, the POS terminal may display a QR code, e.g., 211a-b, that includes the purchase payment amount, e.g., 212a-b, upon the user indicating that the user wishes to checkout the items in the user's physical shopping cart. For example, the QR code may include data formatted according to the extensible Markup Language ("XML"), such as the example data structure below:

```
<QR_data>
    <order_ID>4NFU4RG94</order_ID>
    <timestamp>2011-02-22 15:22:43</timestamp>
    <expiry_lapse>00:00:30</expiry_lapse>
    <transaction_cost>$34.78</transaction_cost>
    <user_ID>john.q.public@gmail.com</user_ID>
    <client_details>
        <client_IP>192.168.23.126</client_IP>
        <client_type>smartphone</client_type>
        <client_model>HTC Hero</client_model>
        <OS>Android 2.2</OS>
        <app_installed_flag>true</app_installed_flag>
    </client_details>
    <secure_element>www.merchant.com/securedyn/0394733/123.png</secure_element>
    <purchase_details>
        <num_products>1</num_products>
        <product>
            <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>
                <seller>bestbuybooks</seller>
            </product_params>
            <quantity>1</quantity>
        </product>
    </purchase_details>
    <merchant_params>
        <merchant_id>3FBCR4INC</merchant_id>
        <merchant_name>Books & Things, Inc.</merchant_name>
        <merchant_auth_key>1NNF484MCP59CHB27365</merchant_auth_key>
    </merchant_params>
<QR_data>
```

[0038] With reference to FIGURE 2D, in some implementations, the user may obtain a snapshot of the QR code displayed on the screen of the secure display or the

POS terminal using a smartphone, e.g., 213. For example, the user's smartphone may have an app, e.g., 214, executing on it to detect and capture QR codes, e.g., 216a. For example, the user may utilize registration features, e.g., 215, to align the QR code within the display of the smartphone. The app may, in some implementations, provide the user with the ability to zoom in, e.g., 217, or zoom out, e.g., 218, of the QR code to ensure that the image of the QR code fits within the dimensions of the screen of the smartphone. Upon aligning the QR code within the display of the smartphone, the user may be able to obtain a snapshot of the QR code using a user interface element, e.g., 219. The user may cancel the snap mobile payment procedure using a user interface element 220 on the display of the smartphone.

[0039] With reference to FIGURE 2E, in some implementations, upon obtaining a snapshot of the merchant-product QR code, the user's smartphone may extract the product and merchant data stored within the QR code, and utilize an account for the user's virtual wallet linked to the user's smartphone to generate a purchase transaction request for processing by a payment network. Upon completion of processing of the payment transaction by the payment network using the information provided by the user's smartphone, the merchant website 222 (via the browser application 221) may provide a purchase receipt 225 for the user. With reference to FIGURE 2F, in implementations where the user utilizes the snap mobile payment procedure at a brick-and-mortar store, the POS terminal may display a purchase receipt for the user. In some implementations, the payment network may provide a purchaser receipt directly to the smartphone of the user.

[0040] FIGURES 3A-E show application user interface diagrams illustrating example features of a snap mobile payment app for capturing product barcodes, securing user data and preventing fraud in some embodiments of the SNAP. With reference to FIGURE 3A, in some implementations, the app executing on the device of the user may include an app interface providing various features for the user. In some implementations, the app may be configured to recognize product identifiers (e.g., barcodes, QR codes, etc.), e.g., 301. For example, the app may be configured to capture merchant-product QR codes for snap mobile payment processing, as discussed above with reference to FIGURES 2A-F. In some implementations, the user may be required to sign in to the app to enable its features. Once enabled, the camera may provide in-person one tap purchasing features for the user. For example, the client device may have a camera via which the app may acquire images, video data, streaming live video, and/or the like, e.g., 303. The app may be configured to analyze the incoming data, and search, e.g., 301, for a product identifier, e.g., 304, such as QR codes 209, 211a-b, 216a and 227. In some implementations, the app may overlay cross-hairs, target box, and/or like alignment reference markers, e.g., 305, so that a user may align the product identifier using the reference markers so facilitate product identifier recognition and interpretation. In some implementations, the app may include interface elements to allow the user to switch back and forth between the product identification mode and product offer interface display screens (see, e.g., 306), so that a user may accurately study deals available to the user before capturing a product identifier. In some implementations, the app may provide the user with the ability to view prior product identifier captures (see, e.g., 307) so that the user may be able to better decide which product identifier the user desires to capture. In some implementations, the user may

desire to cancel product purchasing; the app may provide the user with a user interface element (e.g., 308) to cancel the product identifier recognition procedure and return to the prior interface screen the user was utilizing. In some implementations, the user may be provided with information about products, user settings, merchants, offers, etc. in list form (see, e.g., 309) so that the user may better understand the user's purchasing options. Various other features may be provided for in the app (see, e.g., 310).

[0041] With reference to FIGURE 3B, in some implementations, the app may include an indication of the location (e.g., name of the merchant store, geographical location, information about the aisle within the merchant store, etc.) of the user, e.g., 311. The app may provide an indication of a pay amount due for the purchase of the product, e.g., 312. In some implementations, the app may provide various options for the user to pay the amount for purchasing the product(s). For example, the app may utilize the GPS coordinates to determine the merchant store within the user is present, and direct the user to a website of the merchant. In some implementations, the SNAP may provide an API for participating merchants directly to facilitate transaction processing. In some implementations, a merchant-branded SNAP application may be developed with the SNAP functionality, which may directly connect the user into the merchant's transaction processing system. For example, the user may choose from a number of cards (e.g., credit cards, debit cards, prepaid cards, etc.) from various card providers, e.g., 313. In some implementations, the app may provide the user the option to pay the purchase amount using funds included in a bank account of the user, e.g., a checking, savings, money market, current account, etc., e.g., 314. In some implementations, the user may have set default options for which card, bank account, etc. to use for the purchase transactions via the app. In some implementations, such

setting of default options may allow the user to initiate the purchase transaction via a single click, tap, swipe, and/or other remedial user input action, e.g., 315a. In some implementations, when the user utilizes such an option, the app may utilize the default settings of the user to initiate the purchase transaction. In some implementations, the app may allow the user to utilize other accounts (e.g., Google™ Checkout, Paypal™ account, etc.) to pay for the purchase transaction, e.g., 316. In some implementations, the app may allow the user to utilize rewards points, airline miles, hotel points, electronic coupons, printed coupons (e.g., by capturing the printed coupons similar to the product identifier) etc., to pay for the purchase transaction, e.g., 317-318. In some implementations, the app may provide an option to provide express authorization before initiating the purchase transaction, e.g., 319. In some implementations, the app may provide a progress indicator provide indication on the progress of the transaction after the user has selected an option to initiate the purchase transaction, e.g., 320. In some implementations, the app may provide the user with historical information on the user's prior purchases via the app, e.g., 321. In some implementations, the app may provide the user with an option to share information about the purchase (e.g., via email, SMS, wall posting on Facebook®, tweet on Twitter™, etc.) with other users and/or control information shared with the merchant, acquirer, payment network etc., to process the purchase transaction, e.g., 322. In some implementations the app may provide the user an option to display the product identification information captured by the client device (e.g., in order to show a customer service representative at the exit of a store the product information), e.g., 324. In some implementations, the user, app, device and or SNAP may encounter an error in the processing. In such scenarios, the

user may be able to chat with a customer service representative (e.g., VerifyChat 323) to resolve the difficulties in the purchase transaction procedure.

[0042] In some implementations, the user may select to conduct the transaction using a one-time anonymized credit card number, see e.g., 315b. For example, the SNAP may utilize a pre-designated anonymized set of card details (see, e.g., "AnonCard1," "AnonCard2"). As another example, the SNAP may generate, e.g., in real-time, a one-time anonymous set of card details to securely complete the purchase transaction (e.g., "Anon It 1X"). In such implementations, the app may automatically set the user profile settings such that the any personal identifying information of the user will not be provided to the merchant and/or other entities. In some implementations, the user may be required to enter a user name and password to enable the anonymization features.

[0043] With reference to FIGURE 3C, in some implementations, the user interface elements of the snap mobile payment app may be advantageously arranged to provide the user the ability to process a purchase with customized payment parameters with a minimum number of user gestures applied to the user's mobile device. For example, the user may be provided with an overloaded user interface element, e.g., 325-326. For example, if the user has a QR pay code within the viewing angle of a camera included in the user's mobile device, the user may activate element 325 to snap the QR code and utilize predetermined default settings to process the purchase based on the QR code. However, if the user wishes to customize the payment parameters, the user may activate user interface element 326 (e.g., press and continue to hold). Upon doing so, the app may provide a pop-up menu, e.g., 327, providing a variety of payment

customization choices, such as those provided previously. The user may, e.g., drag the user's finger to the appropriate settings the user prefers, and release the user's finger from the touchscreen of the user's mobile device to select the setting for payment processing. In alternate implementations, the payment settings options, e.g., 330, and QR capture activation buttons, e.g., 328a-b (e.g., 328b may provide even more settings that those displayed in the initial screen), may be included in the user interface along with a window, e.g., 329, for capturing the QR code via the mobile device's camera. In alternate implementations, the user's mobile device may generate a hybrid QR code-payment settings graphic, and the POS terminal (or user's trusted computing device) may capture the entire graphic for payment processing.

[0044] With reference to FIGURE 3D, in some implementations, a user may be advantageously able to provide user settings into a device producing a QR code for a purchase transaction, and then capture the QR code using the user's mobile device. For example, a display device of a point-of-sale terminal may be displaying a checkout screen, such as a web browser executing on a client, e.g., 331, displaying a checkout webpage of an online shopping website, e.g., 332. In some implementations, the checkout screen may provide a user interface element, e.g., 333a-b, whereby the user can indicate the desire to utilize snap mobile payment. For example, if the user activates element 331a, the website may generate a QR code using default settings of the user, and display the QR code, e.g., 335, on the screen of the client for the user to capture using the user's mobile device. In some implementations, the user may be able to activate a user interface element, e.g., 333b, whereby the client may display a pop-up menu, e.g., 334, with additional options that the user may select from. For example, the website may provide user selection options similar to those discussed above in the description

with reference to FIGURES 3B-C. In some implementations, the website may modify the QR code 335 in real-time as the user modifies settings provided by activating the user interface element 333b. Once the user has modified the settings using the pop-up menu, the user may capture a snapshot of the QR code to initiate purchase transaction processing.

[0045] With reference to FIGURE 3E, in some implementations, the SNAP may provide the user with a user interface to modify the user's snap mobile payment settings. For example, the SNAP may provide a web interface, e.g., 341. For example, the user may be able to modify security settings of the user's virtual wallet, e.g., 342, using the web interface. For example, the user may review a list of trusted device, e.g., 344, via which the user may access the user's virtual wallet. In some implementations, the web interface may provide a user interface element to add a trusted device, e.g., 343. The web interface may also provide the user with additional security options. For example, the user be able to set a security passphrase, e.g., 345, modify settings for when the user should be challenged before authorizing a purchase transaction, e.g., 346, the type/style of presentation of the security features, e.g., 347, and a security image to be displayed on the terminal utilized in snap mobile payment, e.g., 348. In various implementations, the user may be able to access other services including modifying user profiles, accounts, account preferences, adding cards, obtaining offers and coupons, locating ATM machines, etc.

[0046] FIGURES 4A-D show data flow diagrams illustrating an example snap mobile payment procedure in some embodiments of the SNAP. With reference to FIGURE 4A, in some implementations, a user, e.g., 401, may desire to purchase a

product, service, offering, and/or the like ("product"), from a merchant, e.g., 403, via a merchant online site or in the merchant's store. The user may communicate with a merchant server, e.g., 403, 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., 402). For example, the user may provide user input, e.g., checkout input 411, into the client indicating the user's desire to purchase the product. For 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. The client may generate a checkout request, e.g., 412, and provide the checkout request, e.g., 413, to the merchant server. For example, the client may provide a (Secure) Hypertext Transfer Protocol ("HTTP(S)") GET message including the product details for the merchant server in the form of data formatted according to the eXtensible Markup Language ("XML"). Below is an example HTTP(S) GET message including an XML-formatted checkout request for the merchant server:

```
GET /checkout.php HTTP/1.1
Host: www.merchant.com
Content-Type: Application/XML
Content-Length: 718
<?XML version = "1.0" encoding = "UTF-8"?>
<checkout_request>
    <session_ID>4NFU4RG94</session_ID>
    <timestamp>2011-02-22 15:22:43</timestamp>
    <user_ID>john.q.public@gmail.com</user_ID>
    <client_details>
          <client_IP>192.168.23.126</client_IP>
          <client_type>smartphone</client_type>
          <client_model>HTC Hero</client_model>
          <OS>Android 2.2</OS>
          <app_installed_flag>true</app_installed_flag>
    </client_details>
    <purchase_details>
          <num_products>1</num_products>
          <product>
                <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>
                        <seller>bestbuybooks</seller>
                    </product_params>
                    <quantity>1</quantity>
                </product>
            </purchase_details>
        </checkout_request>
```

[0047] In some implementations, 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 14. The merchant server may extract the product data, as well as the client data from the checkout request. In some implementations, the merchant server may query, e.g., 414, a merchant database, e.g., 404, to obtain product data, e.g., 415, such as product pricing, sales tax, offers, discounts, rewards, and/or other information to process the purchase transaction. For example, the 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 the database for product data. An example PHP/SQL command listing, illustrating substantive aspects of querying the database, is provided below:

```
    <?PHP
    header('Content-Type: text/plain');
    mysql_connect("254.93.179.112",$DBserver,$password); // access database server
    mysql_select_db("PRODUCTS.SQL"); // select database table to search
    //create query
    $query = "SELECT product_info product_price tax_linfo_list offers_list
        discounts_list rewards_list FROM ProdTable WHERE product LIKE '%' $prod";
    $result = mysql_query($query); // perform the search query
    mysql_close("PRODUCTS.SQL"); // close database access
    ?>
```

[0048] In some implementations, in response to obtaining the product data, the merchant server may generate, e.g., 416a, a QR pay code, and/or secure display element according to the security settings of the user (see, e.g., 358). The merchant server may provide the QR code to the client, so that the client may display the QR code, and the user may capture the QR code using the user's device to obtain merchant and/or product data for generating a purchase transaction processing request. In alternate implementations, the merchant server may direct the client to communicate the product and/or merchant data required to process the transaction to the user's device via an alternate communication protocol, such as, but not limited to: Wi-Fi™, Bluetooth™, cellular network, SMS, email and/or like communication protocols. For example, the merchant server may direct the client to initiate a plug-in on its system to provide the alternate communication service, and transmit the product and/or merchant data to the user's device via the communication service.

[0049] In implementations utilizing a QR code, the merchant server may generate a QR code embodying the product information, as well as merchant information required by a payment network to process the purchase transaction. In some implementations, the QR code may include at least information required by the user device capturing the QR code to generate a purchase transaction processing request, such as a merchant identifier (e.g., a merchant ID number, merchant name, store ID, etc.) and a session identifier for a user shopping session associated with the shopping website/brick-and-mortar store.

[0050] In some implementations, the merchant server may generate in real-time, a custom, user-specific merchant-product XML data structure having a time-limited validity period, such as the example 'QR_data' XML data structure provided below:

```
<QR_data>
    <order_ID>4NFU4RG94</order_ID>
    <timestamp>2011-02-22 15:22:43</timestamp>
    <expiry_lapse>00:00:30</expiry_lapse>
    <transaction_cost>$34.78</transaction_cost>
    <alerts_URL>www.merchant.com/shopcarts.php?sessionID=AEBB4356</alerts_URL>
    <user_ID>john.q.public@gmail.com</user_ID>
    <client_details>
        <client_IP>192.168.23.126</client_IP>
        <client_type>smartphone</client_type>
        <client_model>HTC Hero</client_model>
        <OS>Android 2.2</OS>
        <app_installed_flag>true</app_installed_flag>
    </client_details>
    <secure_element>www.merchant.com/securedyn/0394733/123.png</secure_element>
    <purchase_details>
        <num_products>1</num_products>
        <product>
            <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>
                <seller>bestbuybooks</seller>
            </product_params>
            <quantity>1</quantity>
        </product>
    </purchase_details>
    <merchant_params>
        <merchant_id>3FBCR4INC</merchant_id>
        <merchant_name>Books & Things, Inc.</merchant_name>
        <merchant_auth_key>1NNF484MCP59CHB27365</merchant_auth_key>
    </merchant_params>
<QR_data>
```

[0051] In some implementations, the XML data may include a handle, alias, token, or pointer to information stored on a payment network server, rather than encoding all of the actual data required to initiate the transaction, so that the information encoded into the QR code may be advantageously minimized. In some implementations, the merchant may generate a QR code using the XML data. For example, the merchant server may utilize the PHP QR Code open-source (LGPL) library for generating QR Code, 2-dimensional barcode, available at

http://phpqrcode.sourceforge.net/. For example, the merchant server may issue PHP commands similar to the example commands provided below:

```
<?PHP
header('Content-Type: text/plain');
// Create QR code image using data stored in $data variable
QRcode::png($data, 'qrcodeimg.png');
?>
```

[0052] In alternate implementations, the merchant server may provide, e.g., 416b the XML data to a pay network server, e.g., 406, along with a request to generate a QR code. For example, the merchant server may utilize an API call to the pay network server to request generation of the QR code. The pay network server may generate the QR code for the merchant server, e.g., 416c, and provide, e.g., 416d, the QR code to the merchant server. For example, the pay network server may encode the information provided by the merchant into the QR code, and may also advantageously encode security information, time validity information, digital certificate information, anonymous shipping information, QR code generation/processing fee information, etc. into the QR code.

[0053] In some implementations, the pay network server may provide the merchant server with an encryption key (e.g., a Rivest-Shamir-Adleman ("RSA") private/public key, digital certificate). The merchant may encrypt the custom, user-specific merchant-product XML data structure using the encryption key to generate encrypted purchase data (e.g., using the RSA algorithm). The merchant server may then encode the encrypted data into the QR code. Such a scheme may be employed advantageously, in various embodiments, by the pay network server to authenticate the merchant for any transaction processing requests related to the user-merchant shopping session.

[0054] In some implementations, pre-designed QR codes associated with authenticated with pre-authenticated merchants may be provided to the user device. For example, a user may be browsing an online website on the user's device. The user device may make a HTTP(S) GET request for a webpage from a web server. In some implementations, the web server may, in response to the user device's request for a webpage, generate a query for advertisements to display on the webpage. For example, the web server may search a database, or provide a request to an ad network server (e.g., Akamai) to provide advertisements for embedding into the webpage. In some implementations, the ad network server may utilize keywords, metadata, etc. obtained from the web server (e.g., keywords or metadata associated with the webpage, user profile information, user ID, user browsing history from a cookie stored on the user device, etc.). The ad network may utilize the keywords to generate a query of database(s) for advertisements associated with the keywords, and may obtain advertisements to provide. In some implementations, the ad network server may provide information (e.g., via an API call) on such advertisements (e.g., merchant name, merchant ID, product name, product pricing information, related offers, etc.) to a pay network server. The pay network server may generate a QR code based on the information provide by the ad network server, such that a user device may snap the QR code to initiate a purchase transaction for the goods and/or services associated with the QR code (e.g., as provided by the ad network server to the pay network server). The ad network server may provide the QR as part of the advertisement to the web server, which may in turn embed the advertisement including the QR code into the webpage before providing it to the user device. In alternate implementations, the ad network server/web server may transmit a URL or other identifier of the QR code (ultimately) to

the user device, and the user device may make a call (e.g., a HTTP(S) GET request) using the URL of the QR code (e.g., hosted on the pay network server) to obtain the QR code and display it for the user.

[0055] In some implementations, the merchant server may provide the QR code to the client, e.g., 417. For example, the merchant server may provide a HyperText Markup Language ("HTML") page including a reference to the QR code image and/or secure element image, such as the example HTML page below:

```
<html>
    <img src="www.merchant.com/securedyn/0394733/qrcodeimg.png" alt="Merchant-
    Product QR code"/>
    <img src=" www.merchant.com/securedyn/0394733/123.png" alt="Secure
    Element"/>
</html>
```

[0056] In some implementations, the client may obtain the QR pay code, e.g., 417, and display the QR code, e.g., 418 on a display screen associated with the client device. In some implementations, the user may utilize a user device, e.g., 405, to capture the QR code presented by the client device for payment processing. For example, the user may provide payment input into the user device, e.g., 419. In various implementations, 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 enabled 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. For example, the user device may obtain track 1 data from the user's

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.)
```

[0057] In some implementation, the user device may determine whether an image it has captured depicts a QR code. Depending on whether or not a QR code has been captured, and also (optionally) depending on contents of the QR code, the user device may redirect the user (e.g., via a web browser application executing on the user device) to: a product, a merchant website, a product at a merchant website, a website and including a command to add an item to a purchasing cart of the user associated with the website, and/or the like. For example, the user device may execute a component such as the example Quick Response Code Processing ("QRCP") component 600 described below in the discussion with reference to FIGURES 6A-B.

[0058] In some implementations, upon obtaining the user payment input and capturing the QR code, the user device may generate a card authorization request 420 (e.g., if the QR code includes a purchasing coupon, offer, invoice, personal payment from another virtual wallet user, etc.), for providing to the pay network server. For example, the user device may provide a card authorization request, e.g., 421, on behalf of the user, a HTTP(S) GET message including the product order details for a pay network server, e.g., 406, in the form of XML-formatted data. Below is an example HTTP(S) GET message including an XML-formatted card authorization request for the pay network server:

```
GET /purchase.php HTTP/1.1
Host: www.merchant.com
```

```
Content-Type: Application/XML
Content-Length: 1306
<?XML version = "1.0" encoding = "UTF-8"?>
<purchase_order>
    <order_ID>4NFU4RG94</order_ID>
    <alerts_URL>www.merchant.com/shopcarts.php?sessionID=AEBB4356</alerts_URL>
    <timestamp>2011-02-22 15:22:43</timestamp>
    <user_ID>john.q.public@gmail.com</user_ID>
    <client_details>
        <client_IP>192.168.23.126</client_IP>
        <client_type>smartphone</client_type>
        <client_model>HTC Hero</client_model>
        <OS>Android 2.2</OS>
        <app_installed_flag>true</app_installed_flag>
    </client_details>
    <purchase_details>
        <num_products>1</num_products>
        <product>
            <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>
                <seller>bestbuybooks</seller>
            </product_params>
            <quantity>1</quantity>
        </product>
    </purchase_details>
    <merchant_params>
        <merchant_id>3FBCR4INC</merchant_id>
        <merchant_name>Books & Things, Inc.</merchant_name>
        <merchant_auth_key>1NNF484MCP59CHB27365</merchant_auth_key>
    </merchant_params>
    <account_params>
        <account_name>John Q. Public</account_name>
        <account_type>credit</account_type>
        <account_num>123456789012345</account_num>
        <billing_address>123 Green St., Norman, OK 98765</billing_address>
        <phone>123-456-7809</phone>
        <sign>/jqp/</sign>
        <confirm_type>email</confirm_type>
        <contact_info>john.q.public@gmail.com</contact_info>
    </account_params>
    <shipping_info>
        <shipping_adress>same as billing</shipping_address>
        <ship_type>expedited</ship_type>
        <ship_carrier>FedEx</ship_carrier>
        <ship_account>123-45-678</ship_account>
        <tracking_flag>true</tracking_flag>
        <sign_flag>false</sign_flag>
    </shipping_info>
</purchase_order>
```

[0059] In some implementations, 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 implementations, the card authorization request may include at least a merchant ID, a session ID for the user's shopping session with the merchant, and a device ID of a device (e.g., smartphone) of the user that is linked to the user's virtual wallet. In some implementations, the QR code and messages sent to/from the QR-code capturing device may include the source ID (e.g., identifier of the device generating the QR code), session ID, merchant ID, item ID (e.g., model number), the charge amount, and/or transacting device ID (e.g., the user's smartphone device).

[0060] In some implementations, the card authorization request may be provided by the merchant server or point of sale terminal, instead of the user device. In some implementations, the user, desiring security, may request, via the user device, the pay network server for a dynamically-generated card verification value code (dCVV™) to be utilized along with the user's primary account number ("PAN," e.g., credit card number) in the purchase transaction. In response, the payment network server may generate a dCVV™ code (e.g., using random number generation, MD5 hash of an input key, which may be generated using the user ID, merchant ID, session ID, timestamp, combinations thereof, and/or the like), and provide a session-specific dCVV™ code for the user to utilize along with the user's PAN number. For example, the session-specific dCVV™ code may have an expiry time (e.g., expiry in a one minute from issue). The user device may communicate (e.g., via Bluetooth™, NFC, Wi-Fi, cellular, QR code, etc.) the PAN and dCVV to the point-of-sale terminal, which may create the card authorization request. For example, the user device may generate a QR payment code embedding the PAN and dCVV numbers, and the point of sale terminal may snap an image of the user

device-generated QR payment code. The point of sale terminal may then generate and provide the card authorization request to the pay network server. The pay network server may then be able to validate the transaction by comparing the dCVV obtained from the merchant with the dCVV it provided to the user device before the purchase transaction was initiated. If the dCVV codes from the two sources (pay network server and merchant) correspond properly to each other, the pay network server may continue processing the purchase transaction.

[0061] In some implementations, the card authorization request from a user device may include encrypted data extracted from the QR code, which may have been encrypted by the merchant server as part of a merchant authentication scheme. In some implementations, the pay network server may obtain the encrypted data from the card authorization request provided by the user device, and attempt to decrypt the encrypted data, e.g., using a RSA private/public that is complementary to the key the pay network server initially provided to the merchant server for encrypting the purchase data before embedding it into the QR code. If the pay network server is able to decrypt the purchase data, then the merchant is authenticated as being a valid merchant. In some implementations, the pay network server may compare the purchase data decrypted from the card authorization with data provided by the user/user device, to determine whether the data from these different sources (user/user device, and merchant) correspond properly to each other. Thus, in some implementations, the pay network server may be able to authenticate the merchant, and correlate the merchant to a specific user session or user device before processing the transaction.

[0062] In some implementations, the pay network server may provide a notification to the user device that the transaction is authenticated and approved for transacting. In alternate implementations, the pay network server may proceed with transaction processing. In some implementations, upon identifying that the user is in a session with the merchant, the pay network server may communicate with the user device to provide additional features for the user. For example, in some implementations, the pay network server may provide a communication to the user device (e.g., via a HTTP(S) POST message) to provide: a virtual storefront of the merchant; a depiction of an aisle of the merchant associated with the products included in the card authorization request, a listing of related items; and/or the like (see, e.g., FIGURE 7E-G and description below of additional embodiments).

[0063] With reference to FIGURE 4B, in some implementations, 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.

[0064] In some implementations, the pay network server may generate a query, e.g., 422, 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., 408a-n, of the issuer(s) may maintain details of the user's account. In some implementations, a database, e.g., pay network database 407, may store details of the issuer server(s) associated with the issuer(s). For example, the database may be a relational database responsive to Structured Query Language ("SQL") commands. The pay network server may query the pay network database for issuer server(s) details. For example, the pay network server may execute a hypertext preprocessor ("PHP") script including SQL commands to query the database for details of the issuer server(s). An example PHP/SQL command listing, illustrating substantive aspects of querying the database, is provided below:

```
<?PHP
header('Content-Type: text/plain');
mysql_connect("254.93.179.112",$DBserver,$password); // access database server
mysql_select_db("ISSUERS.SQL"); // select database table to search
//create query for issuer server data
$query = "SELECT issuer_name issuer_address issuer_id ip_address mac_address
    auth_key port_num security_settings_list FROM IssuerTable WHERE account_num
    LIKE '%' $accountnum";
$result = mysql_query($query); // perform the search query
mysql_close("ISSUERS.SQL"); // close database access
?>
```

[0065] In response to obtaining the issuer server query, e.g., 422, the pay network database may provide, e.g., 423, the requested issuer server data to the pay network server. In some implementations, the pay network server may utilize the issuer server data to generate authorization request(s), e.g., 424, 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 card authorization request(s), e.g., 425a-n, to the issuer server(s), e.g., 408a-n. In some implementations, the 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. For example, the pay network server may

provide a HTTP(S) POST message including an XML-formatted authorization request similar to the example listing provided below:

```
POST /authorization.php HTTP/1.1
Host: www.issuer.com
Content-Type: Application/XML
Content-Length: 624
<?XML version = "1.0" encoding = "UTF-8"?>
<card_query_request>
    <query_ID>VNEI39FK</query_ID>
    <timestamp>2011-02-22 15:22:44</timestamp>
    <purchase_summary>
        <num_products>1</num_products>
        <product>
            <product_summary>Book - XML for dummies</product_summary>
            <product_quantity>1</product_quantity?
        </product>
    </purchase_summary>
    <transaction_cost>$22.61</transaction_cost>
    <account_params>
        <account_type>checking</account_type>
        <account_num>1234567890123456</account_num>
    </account_params>
    <merchant_params>
        <merchant_id>3FBCR4INC</merchant_id>
        <merchant_name>Books & Things, Inc.</merchant_name>
        <merchant_auth_key>1NNF484MCP59CHB27365</merchant_auth_key>
    </merchant_params>
</card_query_request>
```

[0066] In some implementations, an issuer server may parse the authorization request(s), and based on the request details may query a database, e.g., user profile database 409a-n, for data associated with an account linked to the user. For example, the issuer server may issue PHP/SQL commands similar to the example provided below:

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

[0067] In some implementations, on obtaining the user data, e.g., 427a-n, the issuer server may determine whether the user can pay for the transaction using funds

available in the account, e.g., 428a-n. 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 an authorization response, e.g., 429a-n, 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 implementations, if at least one issuer server determines that the user cannot pay for the transaction using the funds available in the account, see e.g., 430-431, the pay network server may request payment options again from the user (e.g., by providing an authorization fail message 431 to the user device and requesting the user device to provide new payment options), and re-attempt authorization for the purchase transaction. In some implementations, 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.

[0068] With reference to FIGURE 4C, in some implementations, the pay network server may obtain the authorization message including a notification of successful authorization, see e.g., 430, 433, and parse the message to extract authorization details. Upon determining that the user possesses sufficient funds for the transaction, the pay network server may generate a transaction data record, e.g., 432, 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 similar to the example listing below to store the transaction data in a database:

```
<?PHP
header('Content-Type: text/plain');
mysql_connect("254.92.185.103",$DBserver,$password); // access database server
mysql_select("TRANSACTIONS.SQL"); // select database to append
mysql_query("INSERT INTO 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("TRANSACTIONS.SQL"); // close connection to database
?>
```

[0069] In some implementations, the pay network server may forward an authorization success message, e.g., 433a-b, to the user device and/or merchant server. The merchant may obtain the authorization message, 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., 434, and store the XML data file, e.g., 435, 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>
```

[0070] In some implementations, the server may also generate a purchase receipt, e.g., 434, and provide the purchase receipt to the client, e.g., 436. The client may render and display, e.g., 437a, the purchase receipt for the user. In some implementations, the user device 405 may also provide a notification of successful authorization to the user, e.g., 437b. For example, the 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.

[0071] With reference to FIGURE 4D, in some implementations, the merchant server may initiate clearance of a batch of authorized transactions. For example, the merchant server may generate a batch data request, e.g., 438, and provide the request, e.g., 439, to a database, e.g., merchant database 404. 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., 440. The server may generate a batch clearance request, e.g., 441, using the batch data obtained from the database, and provide, e.g., 442, the batch clearance request to an acquirer server, e.g., 410. 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., 443, a batch payment request using the obtained batch clearance request, and provide the batch

payment request to the pay network server, e.g., 444. 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., 445. The pay network server may store the transaction data, e.g., 446, for each transaction in a database, e.g., pay network database 407. For each extracted transaction, the pay network server may query, e.g., 447-448, a database, e.g., pay network database 407, 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., 449, for each transaction for which it has extracted transaction data, and provide the individual payment request, e.g., 450, to the issuer server, e.g., 408. For example, the pay network server may provide a HTTP(S) POST request similar to the example below:

```
POST /requestpay.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>$34.78</pay_amount>
    <account_params>
        <account_name>John Q. Public</account_name>
        <account_type>credit</account_type>
        <account_num>123456789012345</account_num>
        <billing_address>123 Green St., Norman, OK 98765</billing_address>
        <phone>123-456-7809</phone>
        <sign>/jqp/</sign>
    </account_params>
    <merchant_params>
        <merchant_id>3FBCR4INC</merchant_id>
        <merchant_name>Books & Things, Inc.</merchant_name>
        <merchant_auth_key>1NNF484MCP59CHB27365</merchant_auth_key>
    </merchant_params>
    <purchase_summary>
        <num_products>1</num_products>
        <product>
            <product_summary>Book - XML for dummies</product_summary>
            <product_quantity>1</product_quantity?
        </product>
    </purchase_summary>
</pay_request>
```

[0072] In some implementations, the issuer server may generate a payment command, e.g., 451. 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., 452, to a database storing the user's account information, e.g., user profile database 409. The issuer server may provide a funds transfer message, e.g., 453, to the pay network server, which may forward, e.g., 454, the funds transfer message to the acquirer server. An example HTTP(S) POST funds transfer message 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>$34.78</deposit_amount>
</deposit_ack>
```

[0073] In some implementations, the acquirer server may parse the funds transfer message, 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, e.g., 455.

[0074] FIGURES 5A-E show logic flow diagrams illustrating example aspects of executing a snap mobile payment in some embodiments of the SNAP, e.g., a Snap Mobile Payment Execution ("SMPE") component 500. With reference to FIGURE 5A, in some implementations, 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 server via a client. For

example, the user may provide user input, e.g., 501, into the client indicating the user's desire to checkout shopping items in a (virtual) shopping cart. The client may generate a checkout request, e.g., 502, and provide the checkout request to the merchant server. The merchant server may obtain the checkout request from the client, and extract the checkout detail (e.g., XML data) from the checkout request, e.g., 503. For example, the merchant server may utilize a parser such as the example parsers described below in the discussion with reference to FIGURE 14. The merchant server may extract the product data, as well as the client data from the checkout request. In some implementations, the merchant server may query, e.g., 504, a merchant database to obtain product data, e.g., 505, such as product pricing, sales tax, offers, discounts, rewards, and/or other information to process the purchase transaction.

[0075] In response to obtaining the product data, the merchant server may generate, e.g., 506, a QR pay code, and/or secure display element according to the security settings of the user (see, e.g., 358). For example, the merchant server may generate a QR code embodying the product information, as well as merchant information required by a payment network to process the purchase transaction. For example, the merchant server may first generate in real-time, a custom, user-specific merchant-product XML data structure having a time-limited validity period, such as the example 'QR_data' XML data structure provided below:

```
<QR_data>
    <session_ID>4NFU4RG94</session_ID>
    <timestamp>2011-02-22 15:22:43</timestamp>
    <expiry_lapse>00:00:30</expiry_lapse>
    <transaction_cost>$34.78</transaction_cost>
    <user_ID>john.q.public@gmail.com</user_ID>
    <client_details>
        <client_IP>192.168.23.126</client_IP>
        <client_type>smartphone</client_type>
        <client_model>HTC Hero</client_model>
        <OS>Android 2.2</OS>
```

```
            <app_installed_flag>true</app_installed_flag>
        </client_details>
        <secure_element>www.merchant.com/securedyn/0394733/123.png</secure_element>
        <purchase_details>
            <num_products>1</num_products>
            <product>
                <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>
                    <seller>bestbuybooks</seller>
                </product_params>
                <quantity>1</quantity>
            </product>
        </purchase_details>
        <merchant_params>
            <merchant_id>3FBCR4INC</merchant_id>
            <merchant_name>Books & Things, Inc.</merchant_name>
            <merchant_auth_key>1NNF484MCP59CHB27365</merchant_auth_key>
        </merchant_params>
    <QR_data>
```

[0076] In some implementations, the merchant may generate QR code using the XML data. For example, the merchant server may utilize the PHP QR Code open-source (LGPL) library for generating QR Code, 2-dimensional barcode, available at http://phpqrcode.sourceforge.net/. For example, the merchant server may issue PHP commands similar to the example commands provided below:

```
    <?PHP
    header('Content-Type: text/plain');
    // Create QR code image using data stored in $data variable
    QRcode::png($data, 'qrcodeimg.png');
    ?>
```

[0077] The merchant server may provide the QR pay code to the client, e.g., 506. The client may obtain the QR pay code, and display the QR code, e.g., 507 on a display screen associated with the client device. In some implementations, the user may utilize a user device, e.g., 509, to capture the QR code presented by the client device for payment processing. The client device may decode the QR code to extract the information embedded in the QR code. For example, the client device may utilize an application such as the ZXing multi-format 1D/2D barcode image processing library,

available at http://code.google.com/p/zxing/ to extract the information from the QR code. In some implementations, the user may provide payment input into the user device, e.g., 508. Upon obtaining the user purchase input, the user device may generate a card authorization request, e.g., 509, and provide the card authorization request to a pay network server.

[0078] With reference to FIGURE 5B, in some implementations, the pay network server may parse the card authorization request, e.g., 510, and generate a query, e.g., 511, for issuer server(s) corresponding to the user-selected payment options. In some implementations, a pay network database may store details of the issuer server(s) associated with the issuer(s). In response to obtaining the issuer server query, the pay network database may provide, e.g., 512, the requested issuer server data to the pay network server. In some implementations, the pay network server may utilize the issuer server data to generate authorization request(s), e.g., 425134, for each of the issuer server(s), and provide the card authorization request(s) to the issuer server(s).

[0079] In some implementations, an issuer server may parse the authorization request(s), and based on the request details may query a user profile database for data associated with an account linked to the user. In some implementations, on obtaining the user data, the issuer server may determine whether the user can pay for the transaction using funds available in the account, e.g., 517. 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 an authorization response, e.g., 518, to the pay network server. In some implementations, if at least one issuer server

determines, e.g., 519, that the user cannot pay for the transaction using the funds available in the account, see e.g., 520, option "No," the pay network server may request payment options again from the user (see e.g., 521, option "No," 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 implementations, if the number of failed authorization attempts exceeds a threshold, see, e.g., 521, option "Yes," the pay network server may abort the authorization process, and provide an "authorization fail" message to the merchant server, user device and/or client, e.g., 522.

[0080] In some implementations, the pay network server may obtain the authorization message including a notification of successful authorization, see e.g., 520, option "Yes,", and parse the message to extract authorization details. Upon determining that the user possesses sufficient funds for the transaction, the pay network server may generate a transaction data record, e.g., 523, from the authorization request and/or authorization response, and store, e.g., 524, the details of the transaction and authorization relating to the transaction in a transactions database.

[0081] With reference to FIGURE 5C, in some implementations, the pay network server may forward an authorization success message, e.g., 525, to the user device and/or merchant server, sometimes via the acquirer server, e.g. 526. The merchant may parse the authorization message, e.g., 528, and determine from it that the user possesses sufficient funds in the card account to conduct the transaction, see, e.g., 529. The merchant server may add a record of the transaction for the user to a batch of transaction data relating to authorized transactions, see, e.g., 530-531. In some

implementations, the merchant server may also generate a purchase receipt, e.g., 532, and provide the purchase receipt to the client. The client may render and display, e.g., 534, the purchase receipt for the user. In some implementations, the user device 405 may also provide a notification of successful authorization to the user.

[0082] With reference to FIGURES 5D-E, in some implementations, the merchant server may initiate clearance of a batch of authorized transactions. For example, the merchant server may generate a batch data request, e.g., 535, and provide the request, e.g., 536, to a database, e.g., merchant database. In response to the batch data request, the database may provide the requested batch data, e.g., 536. The server may generate a batch clearance request, e.g., 537, using the batch data obtained from the database, and provide the batch clearance request to an acquirer server. The acquirer server may generate, e.g., 539, a batch payment request using the obtained batch clearance request, and provide the batch payment request to the pay network server. 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., 540-542. The pay network server may store the transaction data, e.g., 543-544, for each transaction in a database, e.g., pay network database. For each extracted transaction, the pay network server may query, e.g., 545-546, a database, e.g., pay network database, for an address of an issuer server. The pay network server may generate an individual payment request, e.g., 547, for each transaction for which it has extracted transaction data, and provide the individual payment request to the associated issuer server.

[0083] In some implementations, the issuer server may generate a payment command, e.g., 548-549. 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., 549, to a database storing the user's account information, e.g., user profile database. The issuer server may provide a funds transfer message, e.g., 551, to the pay network server, which may forward the funds transfer message to the acquirer server. In some implementations, the acquirer server may parse the funds transfer message, 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, e.g., 553-555.

[0084] FIGURES 6A-B show logic flow diagrams illustrating example aspects of processing a Quick Response code in some embodiments of the SNAP, e.g., a Quick Response Code Processing ("QRCP") component 600. With reference to FIGURE 6A, in some implementations, a virtual wallet application executing on a user device may determine whether a QR code has been captured in an image frame obtained by a camera operatively connected to the user device, and may also determine the type, contents of the QR code. Using such information, the virtual wallet application may redirect the user experience of the user and/or initiating purchases, update aspects of the virtual wallet application, etc. For example, the virtual wallet application may trigger the capture of an image frame by a camera operatively connected to the user device, 601. The virtual wallet application may utilize an image segmentation algorithm to identify a foreground in the image, 602, and may crop the rest of the image to reduce background noise in the image, 603. The virtual wallet application may determine whether the foreground image includes a QR code from which data can be reliably read (e.g., this may not be so if the image does not include a QR code, or the QR code is

partially cropped, blurred, etc.), 604. For example, the virtual wallet application may utilize a code library such as the ZXing multi-format 1D/2D barcode image processing library, available at http://code.google.com/p/zxing/ to try and extract the information from the QR code. If the virtual wallet application is able to detect a QR code (605, option "Yes"), the virtual wallet application may decode the QR code, and extract data from the QR code. If the virtual wallet application is unable to detect a QR code (605, option "No"), the virtual wallet application may attempt to perform Optical Character Recognition on the image. For example, the virtual wallet application may utilize the Tesseract C++ open source OCR engine, available at www.pixel-technology.com/freewarw/tessnet2, to perform the optical character recognition, 606. Thus, the virtual wallet application may obtain the data encoded into the image, and may continue if the data can be processed by the virtual wallet application. The virtual wallet application may query a database using fields identified in the extracted data, for a type of the QR code, 608. For example, the QR code could include an invoice/bill, a coupon, a money order (e.g., in a P2P transfer), a new account information packet, product information, purchase commands, URL navigation instructions, browser automation scripts, combinations thereof, and/or the like.

[0085] In some embodiments, the QR code may include data on a new account to be added to the virtual wallet application (see 609). The virtual wallet application may query an issuer of the new account (as obtained from the extracted data), for the data associated with the new account, 610. The virtual wallet application may compare the issuer-provided data to the data extracted from the QR code, 611. If the new account is validated (611, option "Yes"), the virtual wallet application may update the wallet

credentials with the details of the new account, 613, and update the snap history of the virtual wallet application using the data from the QR code, 614.

[0086] With reference to FIGURE 6B, in some embodiments, the QR code may include data on a bill, invoice, or coupon for a purchase using the virtual wallet application (see 615). The virtual wallet application may query merchant(s) associated with the purchase (as obtained from the extracted data), for the data associated with the bill, invoice, or coupon for a purchase (e.g., offer details, offer ID, expiry time, etc.), 616. The virtual wallet application may compare the merchant-provided data to the data extracted from the QR code, 617. If the bill, invoice, or coupon for a purchase is validated (618, option "Yes"), the virtual wallet application may generate a data structure (see e.g., XML QR_data structure in description above with reference to FIGURES 4-5) including the QR-encoded data for generating and providing a card authorization request, 619, and update the snap history of the virtual wallet application using the data from the QR code, 620.

[0087] In some embodiments, the QR code may include product information, commands, user navigation instructions, etc. for the virtual wallet application (see 621). The virtual wallet application may query a product database using the information encodd in the QR. The virtual wallet application may provide various features including, without limitation, displaying product information, redirecting the user to: a product page, a merchant website, a product page on a merchant website, add item(s) to a user shopping cart at a merchant website, etc. In some implementations, the virtual wallet application may perform a procedure such as described above for any image

frame pending to be processed, and/or selected for processing by the user (e.g., from the snap history).

[0088] FIGURE 7 shows a user interface diagram illustrating an overview of example features of virtual wallet applications in some embodiments of the SNAP. FIGURE 7 shows an illustration of various exemplary features of a virtual wallet mobile application 700. Some of the features displayed include a wallet 701, social integration via TWITTER, FACEBOOK, etc., offers and loyalty 703, snap mobile purchase 704, alerts 705 and security, setting and analytics 796. These features are explored in further detail below.

[0089] FIGURES 8A-G show user interface diagrams illustrating example features of virtual wallet applications in a shopping mode, in some embodiments of the SNAP. With reference to FIGURE 8A, 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 8A, 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 810 at the bottom of the user interface. A user may type in an item in the search field 812 to search and/or add an item to a cart 811. 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 813. In a further implementation, a user may also select other shopping options 814 such as current items 815, bills 816, address book 817, merchants 818 and local proximity 819.

[0090] In one embodiment, for example, a user may select the option current items 815, as shown in the left most user interface of FIGURE 8A. When the current

items 815 option is selected, the middle user interface may be displayed. As shown, the middle user interface may provide a current list of items 815a-h in a user's shopping cart 811. A user may select an item, for example item 815a, to view product description 815j 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 815k that captures the information necessary to effect a snap mobile purchase transaction.

[0091] With reference to FIGURE 8B, in another embodiment, a user may select the bills 816 option. Upon selecting the bills 816 option, the user interface may display a list of bills and/or receipts 816a-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 816a 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 816k purchased, <<816i>>, 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 816j 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 8B, a user may select two items for repeat purchase. Upon addition, a message 816l may be displayed to confirm the addition of the two items, which makes the total number of items in the cart 14.

[0092] With reference to FIGURE 8C, in yet another embodiment, a user may select the address book option 817 to view the address book 817a which includes a list of contacts 817b 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 817c. In another example, money may be transferred to Brian S. via QR code as indicated by the QR code icon 817d. In yet another example, Charles B. may accept payment via near field communication 817e, Bluetooth 817f and email 817g. Payment may also be made via USB 817h (e.g., by physically connecting two mobile devices) as well as other social channels such as TWITTER.

[0093] In one implementation, a user may select Joe P. for payment. Joe P., as shown in the user interface, has an email icon 817g 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 817j to his contact information and make a payment transfer. With reference to FIGURE 8D, the user may be provided with a screen 817k 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 817l. The user can choose modes (e.g., SMS, email, social networking) via which Joe may be contacted via graphical user interface elements, 817m. As the user types, the text entered may be provided for review within a GUI element 817n. When the user has completed entering in the necessary information, the user can press the send button 817o to send the social message to Joe. If Joe also has a virtual wallet application, Joe

may be able to review 817p 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 8D, the SMS 817q 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 817r via Facebook®, which includes a URL link that Joe can activate to initiate redemption of the $25 payment.

[0094] With reference to FIGURE 8E, in some other embodiments, a user may select merchants 818 from the list of options in the shopping mode to view a select list of merchants 818a-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 818a for example. The user may then navigate through the merchant's listings to find items of interest such as 818f-j. Directly through the wallet and without visiting the merchant site from a separate page, the user may make a selection of an item 818j from the catalog of Amazon 818a. As shown in the right most user interface of FIGURE 8D, the selected item may then be added to cart. The message 818k indicates that the selected item has been added to the cart, and updated number of items in the cart is now 13.

[0095] With reference to FIGURE 8F, in one embodiment, there may be a local proximity option 819 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 819a-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 819d 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 819f-j available on aisle 5 of Walgreens. In one implementation, the user may select corn 819i from his or her mobile application to add to cart 819k.

[0096] With reference to FIGURE 8G, in another embodiment, the local proximity option 819 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 819l which displays a map 819m 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 819n 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 8190 to move the store view forwards, backwards, right, left as well clockwise and counterclockwise rotation

[0097] FIGURES 9A-F show user interface diagrams illustrating example features of virtual wallet applications in a payment mode, in some embodiments of the SNAP. With reference to FIGURE 9A, 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 910. In one implementation, an example user interface 911 for making a payment is shown. The user interface may clearly identify the amount 912 and the currency 913 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 914 may also be prominently displayed on the user interface. The user may select the funds tab 916 to select one or more forms of payment 917, 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 918 on the user interface shows the number of points available, the graphical indicator 919 shows the number of points to be used towards the amount due 234.56 and the equivalent 920 of the number of points in a selected currency (USD, for example).

[0098] In one implementation, the user may combine funds from multiple sources to pay for the transaction. The amount 915 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 915 matches the amount payable 914. Once the amounts to be debited from one or more forms of payment are finalized by the user, payment authorization may begin.

[0099] In one implementation, the user may select a secure authorization of the transaction by selecting the cloak button 922 to effectively cloak or anonymize some (e.g., pre-configured) or all identifying information such that when the user selects pay button 921, the transaction authorization is conducted in a secure and anonymous manner. In another implementation, the user may select the pay button 921 which may use standard authorization techniques for transaction processing. In yet another implementation, when the user selects the social button 923, 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 923. The indicator 924 may show the authorizing and sending social share data in progress.

[00100] In another implementation, a restricted payment mode 925 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 926 under the funds tab to select specialized accounts such as a flexible spending account (FSA) 927, health savings account (HAS), and/or the like and amounts to be debited to the selected

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

[00101] In one embodiment, the wallet mobile application may facilitate importing of funds via the import funds user interface 928. For example, a user who is unemployed may obtain unemployment benefit fund 929 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 930. 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.

[00102] With reference to FIGURE 9B, 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 931 may display a flag of the United States and may set the currency 933 to the United States. In a further implementation, the wallet mobile application may automatically rearrange the order in which the forms of payments 935 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.

[00103] 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 932 and the currency 934. In a further implementation, the wallet application may rearrange the order in which different forms of payment 936 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.

[00104] With reference to FIGURE 9C, in one embodiment, the payee tab 937 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 938 with whom the user has previously transacted or available to transact. The user may then select one or more payees. The payees 938 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 939 for receiving payment. Upon selection, the user interface may display additional identifying information relating to the payee.

[00105] With reference to FIGURE 9D, in one embodiment, the mode tab 1940 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 941, wireless 942, snap mobile by user-obtained QR code 943, secure chip 944, TWITTER 945, near-field communication (NFC) 946, cellular 947, snap mobile by user-provided QR code 948, USB 949 and FACEBOOK 950, 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.

[00106] With reference to FIGURE 9E, in one embodiment, the offers tab 951 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 952 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 953. In a further implementation, the user may read the details of the offer by expanding the offer row as shown by 954 in the user interface.

[00107] With reference to FIGURE 9F, in one embodiment, the social tab 955 may facilitate integration of the wallet application with social channels 956. In one implementation, a user may select one or more social channels 956 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 957 and signing in 958. The user may then use the social button 959 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 SNAP to engage in interception parsing.

[00108] FIGURE 10 shows a user interface diagram illustrating example features of virtual wallet applications, in a history mode, in some embodiments of the SNAP. In one embodiment, a user may select the history mode 1010 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 1011. In another implementation, the user may use voice activated search feature by clicking on the microphone icon 1014. 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 1015. The user interface may also identify the date 1012 of the transaction, the merchants and items 1013 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.

[00109] In one implementation, the user may select a transaction, for example transaction 1015, 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 1016 of each item. In a further implementation, the user may select the show option 1017 to view actions 1018 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.

[00110] The history mode, in another embodiment, may offer facilities for obtaining and displaying ratings 1019 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 1020 shows FACEBOOK message exchanges between two users. In one implementation, a user may share a link via a message 1021. 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.

[00111] In one embodiment, the history mode may also include facilities for exporting receipts. The export receipts pop up 1022 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 1025, 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 1023 to look up email or fax number for exporting.

The user may also specify format options 1024 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 1027 to initiate export of receipts.

[00112] FIGURES 11A-F show user interface diagrams illustrating example features of virtual wallet applications in a snap mode, in some embodiments of the SNAP. With reference to FIGURE 11A, in some embodiments, a user may select a snap mode 1101 to access snap features. In various embodiments, the virtual wallet application may able to snap and identify a variety of items. For example, the virtual wallet application may be able to snap and identify a purchase invoice 1103, a coupon 104, money (e.g., sent in a person-to-person transfer) 1105, a bill (e.g., utilities, etc.) 1106, a receipt (e.g., for storing, expense reporting, etc.) 1107, a pay account (e.g., to add a new credit/debit/prepaid card to the virtual wallet application) 1108. The user may be able to return to a shopping screen at any time by activating a graphical user interface element 1102. In some embodiments, the user may be able to set a name of a cart or wishlist stored within the user's virtual wallet application to which the item snapped should be sent (see 1109). In some embodiments, the virtual wallet application may allow a user to create a new cart or wishlist to which the snapped items should be added.

[00113] In one embodiment, a user may select the snap mode 1110 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 11A. A user may use his or her mobile phone to take a picture of a QR code 1115 and/or a barcode 1114. In one implementation, the bar 1113 and snap frame 1115 may assist the user in snapping codes properly. For example, the snap frame 1115, as shown, does not capture the entirety of the code 1116. 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 1113 that indicates that the snap mode is still seeking the code. The user may modify the zoom level 1117 of the camera to facilitate snapping the QR code. When the code 1116 is completely framed by the snap frame 1115, 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 (see 1120). In another implementation, the snap mode may automatically snap the code using the mobile device camera (see 1119). In some implementations, the virtual wallet application may optionally apply a Global Positioning System tag (see 1118) to the QR code before storing it, or utilizing it in a transaction.

[00114] With reference to FIGURE 11B, 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.

[00115] 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 1122 the receipt 1123 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 1123 using information from the barcode. The user may now reallocate 1125. In some implementations, the user may also dispute the transaction 1124 or archive the receipt 1126.

[00116] In one implementation, when the reallocate button 1125 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 1127 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.

[00117] With reference to FIGURE 11C, 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 1132. Once the QR code is identified, the navigation bar 1131 may indicate that the pay code is identified. The user may now have an option to add to cart 1133, pay with a default payment account 1134 or pay with wallet 1135.

[00118] In one implementation, the user may decide to pay with default 1134. 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 1137 to share purchase information with others, reallocate 1138 as discussed with regard to FIGURE 11B, and archive 1139 to store the receipt.

[00119] With reference to FIGURE 11D, 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 1141 (e.g., a bar code, a QR code, and/or the like). The wallet application may then generate an offer text 1142 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 1143 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 1144. Furthermore, the user may also save the offer for future use by selecting the save button 1145.

[00120] In one implementation, after the offer or coupon 1146 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 1148, and the user may also save the offer or coupon 1146 for later use.

[00121] With reference to FIGURE 11E, 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 1151 and may display the textual information 1152 encoded in the pay card. The user may then perform verification of the information 1152 by selecting the verify button 1153. In one implementation, the verification may include contacting

the issuer of the pay card for confirmation of the decoded information 1152 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 1154. 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 916 discussed in FIGURE 9A. The user may also cancel importing of the pay card as a funding source by selecting the cancel button 1155. 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 1156. The user may then access the wallet 1157 to begin using the added pay card as a funding source.

[00122] With reference to FIGURE 11F, in some implementations, the virtual wallet application may identify a product from processing the QR code, and may provide information about the product, as well as information about options for buying the product, assistive services, and/or the like. For example, the virtual wallet application may provide a window 1161, wherein the virtual wallet application may display images, product specification, prices, merchant information, and/or the like (see 1162). In some implementations, the virtual wallet application may provide a QR code including the displayed information, so that another user may quickly snap the information to import it into another virtual wallet application. In some implementations, the virtual wallet application may provide features so that a user may request concierge services (e.g., assistance while shopping), shipping services (e.g., so the user may leave a store without carrying the items out), 1164. In some implementations, the virtual wallet application may provide competitive prices of local merchants (e.g., using the GPS location of the user device) or merchants on the Internet (see 1165). In some implementations, the virtual wallet application may provide the user with features including, but not limited

to: viewing prior snaps, snapping a new code, adding GPS tags to codes, retrieving a previously snapped code for use, entering manual information about a QR code, attribute the QR code to an object (e.g., so that QR codes for home furniture products may be grouped into a "bedrrom furniture" object, for organization purposes), etc. (see 1166). In some embodiments, the user may be able to set a name of a cart or wishlist stored within the user's virtual wallet application to which the item snapped should be sent (see 1167). In some embodiments, the virtual wallet application may allow a user to create a new cart or wishlist to which the snapped items should be added.

[00123] FIGURE 12 shows a user interface diagram illustrating example features of virtual wallet applications, in an offers mode, in some embodiments of the SNAP. In some implementations, the SNAP 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 1211, or issue voice commands by activating GUI element 1212 and speaking commands into the device. In some implementations, the SNAP 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 1213. For example, the offer may provide a discount, and may include an expiry time. In some implementations, other users may provide gifts (e.g., 1214) 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., 1215). In some implementations, the offers section may

include alerts as to requesting receipt of funds from other users (e.g., 1216). 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 SNAP, e.g., 1217-1219, 1220. These offers may sometimes be assembled using a combination of participating merchants, e.g., 1217. In some implementations, the SNAP itself may provide offers for users contingent on the user utilizing particular payment forms from within the virtual wallet application, e.g., 1220.

[00124] FIGURES 13A-B show user interface diagrams illustrating example features of virtual wallet applications, in a security and privacy mode, in some embodiments of the SNAP. With reference to FIGURE 13A, 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., 1311a-b), account number (e.g., 1312a-b), user security access code (e.g., 1313-b), user pin (e.g., 1314-b), user address (e.g., 1315-b), social security number associated with the user (e.g., 1316-b), current device GPS location (e.g., 1317-b), user account of the merchant in whose store the user currently is (e.g., 1318-b), the user's rewards accounts (e.g., 1319-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 13A, the user has selected the name 1311a, account number 1312a, security code 1313a, merchant account ID 1318a and rewards account ID 1319a 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 SNAP with the GPS location of the user. Based on the GPS location of the user, the SNAP 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.

[00125] 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.

[00126] With reference to FIGURE 13B, in some implementations, the app executing on the user's device may provide a "VerifyChat" feature for fraud prevention. For example, the SNAP may detect an unusual and/or suspicious transaction. The SNAP 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 SNAP 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 SNAP may initiate a video challenge for the user, e.g., 1321. For example, the user may need to present him/her-self via a video chat, e.g., 1322. In some implementations, a customer service representative, e.g., agent 1324, may manually determine the authenticity of the user using the video of the user. In some implementations, the SNAP 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., 1323, so that the user may the video to facilitate the SNAP'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 SNAP may then cancel the transaction, and/or initiate fraud investigation procedures on behalf of the user.

[00127] In some implementations, the SNAP may utilize a text challenge procedure to verify the authenticity of the user, e.g., 1325. For example, the SNAP may communicate with the user via text chat, SMS messages, electronic mail, Facebook® messages, Twitter™ tweets, and/or the like. The SNAP may pose a challenge question, e.g., 1326, for the user. The app may provide a user input interface element(s) (e.g., virtual keyboard 1328) to answer the challenge question posed by the SNAP. In some implementations, the challenge question may be randomly selected by the SNAP 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 SNAP may cancel the transaction, and/or initiate fraud investigation on behalf of the user.

SNAP Controller

[00128] FIGURE 14 shows a block diagram illustrating embodiments of a SNAP controller 1401. In this embodiment, the SNAP controller 1401 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.

[00129] Typically, users, e.g., 1433a, 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 1403 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 1429 (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.

[00130] In one embodiment, the SNAP controller 1401 may be connected to and/or communicate with entities such as, but not limited to: one or more users from user input devices 1411; peripheral devices 1412; an optional cryptographic processor device 1428; and/or a communications network 1413. For example, the SNAP controller 1401 may be connected to and/or communicate with users, e.g., 1433a, operating client device(s), e.g., 1433b, 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.

[00131] 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.

[00132] The SNAP controller 1401 may be based on computer systems that may comprise, but are not limited to, components such as: a computer systemization 1402 connected to memory 1429.

Computer Systemization

[00133] A computer systemization 1402 may comprise a clock 1430, central processing unit ("CPU(s)" and/or "processor(s)" (these terms are used interchangeable

throughout the disclosure unless noted to the contrary)) 1403, a memory 1429 (e.g., a read only memory (ROM) 1406, a random access memory (RAM) 1405, etc.), and/or an interface bus 1407, and most frequently, although not necessarily, are all interconnected and/or communicating through a system bus 1404 on one or more (mother)board(s) 1402 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 1486; e.g., optionally the power source may be internal. Optionally, a cryptographic processor 1426 and/or transceivers (e.g., ICs) 1474 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 1412 via the interface bus I/O. In turn, the transceivers may be connected to antenna(s) 1475, 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 SNAP controller to determine its location)); Broadcom BCM4329FKUBG transceiver chip (e.g., providing 802.11n, Bluetooth 2.1 + EDR, FM, etc.); a Broadcom BCM4750IUB8 receiver chip (e.g., GPS); an Infineon Technologies X-Gold 618-PMB9800 (e.g., providing 2G/3G HSDPA/HSUPA communications); and/or the like. The system clock typically has a crystal oscillator and generates a base signal through the computer systemization's circuit pathways. The clock is typically 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 commonly 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.

[00134] 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: integrated system (bus) controllers, memory management control units, floating point units, 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 1429 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. The CPU may be a microprocessor such as: AMD's Athlon, Duron and/or Opteron; ARM's application, embedded and secure

processors; IBM and/or Motorola's DragonBall and PowerPC; IBM's and Sony's Cell processor; Intel's Celeron, Core (2) Duo, 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) according to conventional data processing techniques. Such instruction passing facilitates communication within the SNAP controller and beyond through various interfaces. Should processing requirements dictate a greater amount speed and/or capacity, distributed processors (e.g., Distributed SNAP), mainframe, multi-core, parallel, and/or super-computer architectures may similarly be employed.Alternatively, should deployment requirements dictate greater portability, smaller Personal Digital Assistants (PDAs) may be employed.

[00135] Depending on the particular implementation, features of the SNAP 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 SNAP, 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 SNAP 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 SNAP may be implemented with embedded components that are configured and used to achieve a variety of features or signal processing.

[00136] 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, SNAP 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 SNAP features. A hierarchy of programmable interconnects allow logic blocks to be interconnected as needed by the SNAP 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 SNAP may be developed on regular FPGAs and then migrated into a fixed version that more resembles ASIC implementations. Alternate or coordinating implementations may migrate SNAP 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 SNAP.

Power Source

[00137] The power source 1486 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 1486 is connected to at least one of the interconnected subsequent components of the SNAP thereby providing an electric current to all subsequent components. In one example, the power source 1486 is connected to the system bus component 1404. In an alternative embodiment, an outside power source 1486 is provided through a connection across the I/O 1408 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

[00138] Interface bus(ses) 1407 may accept, connect, and/or communicate to a number of interface adapters, conventionally although not necessarily in the form of adapter cards, such as but not limited to: input output interfaces (I/O) 1408, storage interfaces 1409, network interfaces 1410, and/or the like. Optionally, cryptographic processor interfaces 1427 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 conventionally connect to the interface bus via a slot architecture. Conventional slot architectures may be employed,

such as, but not limited to: Accelerated Graphics Port (AGP), Card Bus, (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), and/or the like.

[00139] Storage interfaces 1409 may accept, communicate, and/or connect to a number of storage devices such as, but not limited to: storage devices 1414, 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, fiber channel, Small Computer Systems Interface (SCSI), Universal Serial Bus (USB), and/or the like.

[00140] Network interfaces 1410 may accept, communicate, and/or connect to a communications network 1413. Through a communications network 1413, the SNAP controller is accessible through remote clients 1433b (e.g., computers with web browsers) by users 1433a. 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 SNAP), architectures may similarly be employed to pool, load balance, and/or otherwise increase the communicative bandwidth required by the SNAP 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 1410 may be used to engage with various communications network types 1413. For example, multiple network interfaces may be employed to allow for the communication over broadcast, multicast, and/or unicast networks.

[00141] Input Output interfaces (I/O) 1408 may accept, communicate, and/or connect to user input devices 1411, peripheral devices 1412, cryptographic processor devices 1428, 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), 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, 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 typical output device may include a video display, which typically comprises a Cathode Ray Tube (CRT) or Liquid Crystal Display (LCD) based monitor with an interface (e.g., DVI circuitry and cable) that accepts signals from a video interface, may be used. 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. Typically, 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, etc.).

[00142] User input devices 1411 often are a type of peripheral device 1412 (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.

[00143] Peripheral devices 1412 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 SNAP 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 1428), force-feedback devices (e.g., vibrating motors), network interfaces, printers, 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., cameras).

[00144] It should be noted that although user input devices and peripheral devices may be employed, the SNAP 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.

[00145] Cryptographic units such as, but not limited to, microcontrollers, processors 1426, interfaces 1427, and/or devices 1428 may be attached, and/or communicate with the SNAP 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, SafeNet's Luna PCI (e.g., 7100) series; Semaphore Communications' 40 MHz Roadrunner 184; 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

[00146] Generally, any mechanization and/or embodiment allowing a processor to affect the storage and/or retrieval of information is regarded as memory 1429. 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 SNAP controller and/or a computer systemization may employ various forms of memory 1429. 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 a typical configuration, memory 1429 will include ROM 1406, RAM 1405, and a storage device 1414. A storage device 1414 may be any conventional computer system storage. 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

[00147] The memory 1429 may contain a collection of program and/or database components and/or data such as, but not limited to: operating system component(s) 1415 (operating system); information server component(s) 1416 (information server); user interface component(s) 1417 (user interface); Web browser component(s) 1418 (Web browser); database(s) 1419; mail server component(s) 1421; mail client component(s) 1422; cryptographic server component(s) 1420 (cryptographic server);

the SNAP component(s) 1435; 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 non-conventional program components such as those in the component collection, typically, are stored in a local storage device 1414, 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

[00148] The operating system component 1415 is an executable program component facilitating the operation of the SNAP controller. Typically, the operating system facilitates 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. 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 SNAP controller to communicate with other entities through a communications network 1413. Various communication protocols may be used by the SNAP 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

[00149] An information server component 1416 is a stored program component that is executed by a CPU. The information server may be a conventional 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), 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 SNAP 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 SNAP database 1419, operating systems, other program components, user interfaces, Web browsers, and/or the like.

[00150] Access to the SNAP 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 SNAP. 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 SNAP 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.

[00151] 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

[00152] 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, IBM's OS/2, Microsoft's Windows 2000/2003/3.1/95/98/CE/Millenium/NT/XP/Vista/7 (i.e., Aero), 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.

[00153] A user interface component 1417 is a stored program component that is executed by a CPU. The user interface may be a conventional 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

[00154] A Web browser component 1418 is a stored program component that is executed by a CPU. The Web browser may be a conventional hypertext viewing application such as Microsoft Internet Explorer or Netscape Navigator. 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., FireFox, 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, 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 affect the obtaining and the provision of information to users, user agents,

and/or the like from the SNAP enabled nodes. The combined application may be nugatory on systems employing standard Web browsers.

Mail Server

[00155] A mail server component 1421 is a stored program component that is executed by a CPU 1403. The mail server may be a conventional Internet mail server such as, but not limited to 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 SNAP.

[00156] Access to the SNAP mail may be achieved through a number of APIs offered by the individual Web server components and/or the operating system.

[00157] 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

[00158] A mail client component 1422 is a stored program component that is executed by a CPU 1403. The mail client may be a conventional mail viewing application such as Apple 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

[00159] A cryptographic server component 1420 is a stored program component that is executed by a CPU 1403, cryptographic processor 1426, cryptographic processor interface 1427, cryptographic processor device 1428, 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 conventional 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 SNAP 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 SNAP component to engage in secure transactions if so desired. The cryptographic component facilitates the secure accessing of resources on the SNAP 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 SNAP Database

[00160] The SNAP database component 1419 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 a conventional, fault tolerant, relational, scalable, secure database such as Oracle or Sybase. 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.

[00161] Alternatively, the SNAP 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 SNAP database is implemented as a data-structure, the use of the SNAP database 1419 may be integrated into another component such as the SNAP component 1435. 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.

[00162] In one embodiment, the database component 1419 includes several tables 1419a-o. A Users table 1419a 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 SNAP. A Devices table 1419b may include fields such as, but not limited to: device_ID, device_name, device_IP, device_MAC, device_type, device_model, device_version, device_OS, device_apps_list, device_securekey, wallet_app_installed_flag, and/or the like. An Apps table 1419c may include fields such as, but not limited to: app_ID, app_name, app_type, app_dependencies, and/or the like. An Accounts table 1419d 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 1419e may include fields such as, but not limited to: merchant_id, merchant_name, merchant_address, ip_address, mac_address, auth_key, port_num, security_settings_list, and/or the like. An Issuers table 1419f 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 1419g 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 1419h 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 Transactions table 1419i 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 1419j 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 1419k 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 1419l 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 1419m 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 1419n 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 14190 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.

[00163] In one embodiment, the SNAP database may interact with other database systems. For example, employing a distributed database system, queries and data access by search SNAP component may treat the combination of the SNAP database, an integrated data security layer database as a single database entity.

[00164] In one embodiment, user programs may contain various user interface primitives, which may serve to update the SNAP. Also, various accounts may require custom database tables depending upon the environments and the types of clients the SNAP 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 1419a-o. The SNAP may be configured to keep track of various settings, inputs, and parameters via database controllers.

[00165] The SNAP database may communicate to and/or with other components in a component collection, including itself, and/or facilities of the like. Most frequently, the SNAP database communicates with the SNAP component, other program components, and/or the like. The database may contain, retain, and provide information regarding other nodes and data.

The SNAPs

[00166] The SNAP component 1435 is a stored program component that is executed by a CPU. In one embodiment, the SNAP component incorporates any and/or all combinations of the aspects of the SNAP discussed in the previous figures. As such, the SNAP affects accessing, obtaining and the provision of information, services, transactions, and/or the like across various communications networks.

[00167] The SNAP component may transform real-time-generated merchant-product Quick Response codes via SNAP components into virtual wallet card-based transaction purchase notifications, and/or the like and use of the SNAP. In one embodiment, the SNAP component 1435 takes inputs (e.g., checkout input 411; product data 414; payment input 419; issuer server data 423; user data 427a-n; and/or the like), and transforms the inputs via SNAP components (e.g., SMPE 1441; QRCP 1442; and/or

the like), into outputs (e.g., QR pay code 417; card authorization request 421; authorization response 429a-n; authorization success message 433a-b; batch append data 435; purchase receipt 436; and/or the like).

[00168] The SNAP 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 SNAP server employs a cryptographic server to encrypt and decrypt communications. The SNAP component may communicate to and/or with other components in a component collection, including itself, and/or facilities of the like. Most frequently, the SNAP component communicates with the SNAP database, operating systems, other program components, and/or the like. The SNAP may contain, communicate, generate, obtain, and/or provide program component, system, user, and/or data communications, requests, and/or responses.

Distributed SNAPs

[00169] The structure and/or operation of any of the SNAP 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.

[00170] 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.

[00171] The configuration of the SNAP 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.

[00172] 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.

[00173] For example, a grammar may be arranged to recognize the tokens of an HTTP post command, e.g.:

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

[00174] 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.

[00175] For example, in some implementations, the SNAP 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
?>
```

[00176] 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
```

[00177] and other parser implementations:

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

[00178] all of which are hereby expressly incorporated by reference herein.

[00179] In order to address various issues and advance the art, the entirety of this application for SNAP MOBILE PAYMENT 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 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 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, services, servers, and/or the like that may execute asynchronously, concurrently, in parallel, simultaneously, synchronously, and/or the like are 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 SNAP 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 SNAP may be implemented that enable a great deal of flexibility and customization. For example, aspects of the SNAP may be adapted for restaurant dining, online shopping, shopping in brick-and-mortar stores, secure information processing, healthcare information systems, and/or the like. While various embodiments and discussions of the SNAP 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. A visual snap computer-implemented method, comprising:

obtaining a user visual item information request (VITR);

extracting a VITR image frame from the VITR obtained via an image acquisition device operatively connected to a user device;

extracting VITR field attributes from the VITR image frame through a VITR processing component, wherein the VITR processing component may include barcode recognition and optical character recognition;

querying a database with the extracted VITR field attributes;

determining a VITR type from results from the querying;

providing a visual item information response based on the determined VITR type;

wherein a payment code is depicted within the extracted VITR image frame;

wherein a purchase transaction request is generated using the identified payment code.

2. The method of claim 1, wherein the VITR may be one of a visual coupon addition request, a visual account addition request, a visual bill addition request, a visual purchase request, and a visual purchase information request.

3. The method of claim 2, wherein the VITR and visual item information response are storable for later access and retrieval via a VITR history.

4. The method of claim 1, wherein the VITR is a visual purchase request, and the provided visual item information response is a purchase intake mechanism, and further comprising:

obtaining the user's indication to purchase a purchase item identified in the provided visual item information response;

obtaining a purchase transaction request for payment processing of the purchase item; and

providing a purchase receipt for the purchase transaction.

5. The method of claim 4, wherein the VITR, visual item information response, user's indication, purchase transaction request and purchase receipt are storable for later access and retrieval via a VITR history.

6. The method of claim 1, wherein a server performs VITR activities.

7. The method of claim 1, wherein a user client device performs VITR activities.

8. A visual snap computer-implemented system, comprising:

a processor; and

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

obtain a user visual item information request (VITR);

extract a VITR image frame from the VITR obtained via an image acquisition device operatively connected to a user device;

extract VITR field attributes from the VITR image frame through a VITR processing component, wherein the VITR processing component may include barcode recognition and optical character recognition;

query a database with the extracted VITR field attributes;

determine a VITR type from results from the querying;

provide a visual item information response based on the determined VITR type;

wherein a payment code is depicted within the extracted VITR image frame;

wherein a purchase transaction request is generated using the identified payment code.

9. The system of claim 8, wherein the VITR may be one of a visual coupon addition request, a visual account addition request, a visual bill addition request, a visual purchase request, and a visual purchase information request.

10. The system of claim 9, wherein the VITR and visual item information response are storable for later access and retrieval via a VITR history.

11. The system of claim 8, wherein the VITR is a visual purchase request, and the provided visual item information response is a purchase intake mechanism, and further comprising:

obtain the user's indication to purchase a purchase item identified in the provided visual item information response;

obtain a purchase transaction request for payment processing of the purchase item; and

provide a purchase receipt for the purchase transaction.

12. The system of claim 11, wherein the VITR, visual item information response, user's indication, purchase transaction request and purchase receipt are storable for later access and retrieval via a VITR history.

13. The system of claim 8, wherein a server performs VITR activities.

14. The system of claim 8, wherein a user client device performs VITR activities.

15. A computer-readable tangible medium storing computer-executable visual snap instructions to:

obtain a user visual item information request (VITR);

extract a VITR image frame from the VITR obtained via an image acquisition device operatively connected to a user device;

extract VITR field attributes from the VITR image frame through a VITR processing component, wherein the VITR processing component may include barcode recognition and optical character recognition;

query a database with the extracted VITR field attributes;

determine a VITR type from results from the querying;

provide a visual item information response based on the determined VITR type;

wherein a payment code is depicted within the extracted VITR image frame;

wherein a purchase transaction request is generated using the identified payment code.

16. The medium of claim 15, wherein the VITR may be one of a visual coupon addition request, a visual account addition request, a visual bill addition request, a visual purchase request, and a visual purchase information request.

17. The medium of claim 16, wherein the VITR and visual item information response are storable for later access and retrieval via a VITR history.

18. The medium of claim 15, wherein the VITR is a visual purchase request, and the provided visual item information response is a purchase intake mechanism, and further storing instructions to:

obtain the user's indication to purchase a purchase item identified in the provided visual item information response;

obtain a purchase transaction request for payment processing of the purchase item; and

provide a purchase receipt for the purchase transaction.

19. The medium of claim 18, wherein the VITR, visual item information response, user's indication, purchase transaction request and purchase receipt are storable for later access and retrieval via a VITR history.

20. The medium of claim 15, wherein a server performs VITR activities.

21. The medium of claim 15, wherein a user client device performs VITR activities.

22. A computer-readable tangible medium storing computer-executable snap payment instructions to:

obtain a user visual item information request (VITR);

acquire an VITR image frame via an image acquisition device operatively connected to the user device;

extract VITR field attributes from the VITR image frame through a VITR processing component, wherein the VITR processing component may include barcode recognition and optical character recognition;

query a database with the extracted VITR field attributes;

determine a VITR type from results from the querying;

provide a visual item information response based on the determined VITR type.

23. The medium of claim 22, wherein the VITR may be one of a visual coupon addition request, a visual account addition request, a visual bill addition request, a visual purchase request, and a visual purchase information request.

24. The medium of claim 23, wherein the VITR and visual item information response are storable for later access and retrieval via a VITR history.

25. The medium of claim 22, wherein the VITR is a visual purchase request, and the provided visual item information response is a purchase intake mechanism, and further storing instructions to:

obtain the user's indication to purchase a purchase item identified in the provided visual item information response;

provide a purchase transaction request for payment processing of the purchase item; and

obtain a purchase receipt for the purchase transaction.

26. The medium of claim 25, wherein the VITR, visual item information response, user's indication, purchase transaction request and purchase receipt are storable for later access and retrieval via a VITR history.

27. A visual snap computer-implemented means, comprising means for:

obtaining a user visual item information request (VITR);

extracting a VITR image frame from the VITR obtained via an image acquisition device operatively connected to a user device;

extracting VITR field attributes from the VITR image frame through a VITR processing component, wherein the VITR processing component may include barcode recognition and optical character recognition;

querying a database with the extracted VITR field attributes;

determining a VITR type from results from the querying;

providing a visual item information response based on the determined VITR type;

wherein a payment code is depicted within the extracted VITR image frame;

wherein a purchase transaction request is generated using the identified payment code.

28. The means of claim 27, wherein the VITR may be one of a visual coupon addition request, a visual account addition request, a visual bill addition request, a visual purchase request, and a visual purchase information request.

29. The means of claim 28, wherein the VITR and visual item information response are storable for later access and retrieval via a VITR history.

30. The means of claim 27, wherein the VITR is a visual purchase request, and the provided visual item information response is a purchase intake mechanism, and further comprising means for:

obtaining the user's indication to purchase a purchase item identified in the provided visual item information response;

obtaining a purchase transaction request for payment processing of the purchase item; and

providing a purchase receipt for the purchase transaction.

31. The means of claim 30, wherein the VITR, visual item information response, user's indication, purchase transaction request and purchase receipt are storable for later access and retrieval via a VITR history.

32. The means of claim 27, wherein a server performs VITR activities.

33. The means of claim 27, wherein a user client device performs VITR activities.