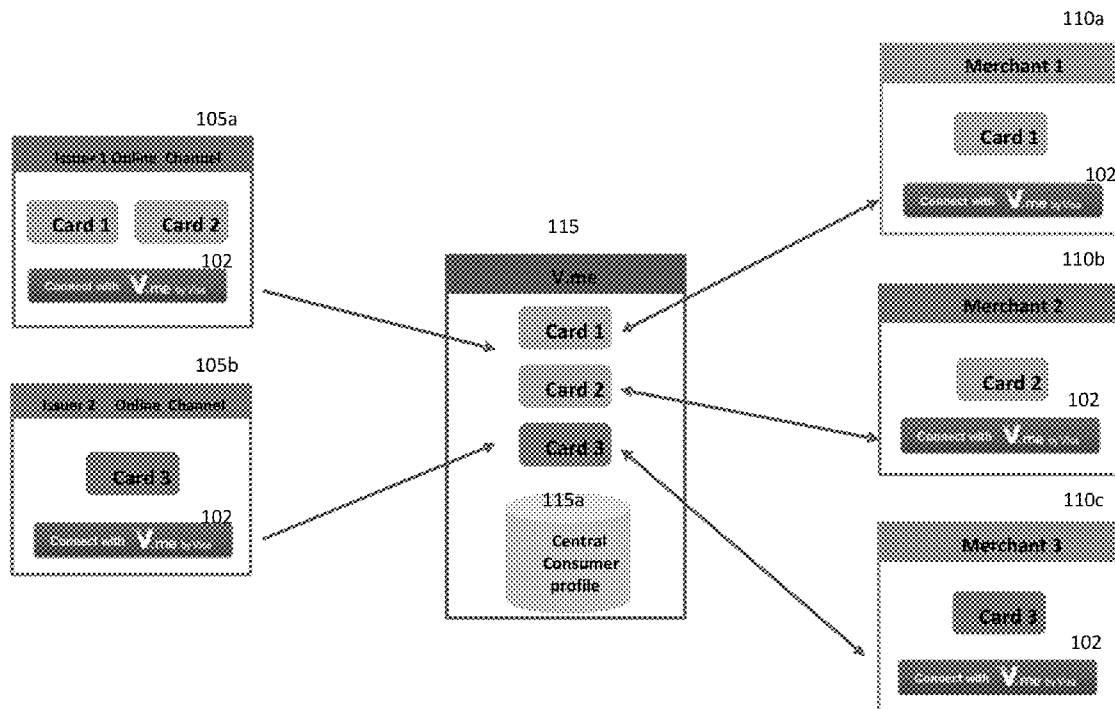




US 20130054454A1

(19) **United States**(12) **Patent Application Publication**
Purves et al.(10) **Pub. No.: US 2013/0054454 A1**(43) **Pub. Date: Feb. 28, 2013**(54) **WALLET SERVICE ENROLLMENT
PLATFORM APPARATUSES, METHODS AND
SYSTEMS**61/558,620, filed on Nov. 11, 2011, provisional applica-
tion No. 61/668,441, filed on Jul. 5, 2012.(71) Applicants: **Thomas Purves**, San Francisco, CA
(US); **Shilpak Mahadkar**, Foster City,
CA (US); **Ayman Hammad**, Pleasanton,
CA (US)(72) Inventors: **Thomas Purves**, San Francisco, CA
(US); **Shilpak Mahadkar**, Foster City,
CA (US); **Ayman Hammad**, Pleasanton,
CA (US)(21) Appl. No.: **13/624,779**(22) Filed: **Sep. 21, 2012****Related U.S. Application Data**(63) Continuation-in-part of application No. 13/589,053,
filed on Aug. 17, 2012.(60) Provisional application No. 61/525,168, filed on Aug.
18, 2011, provisional application No. 61/537,421,
filed on Sep. 21, 2011, provisional application No.**Publication Classification**(51) **Int. Cl.**
G06Q 20/36 (2012.01)(52) **U.S. Cl.** **705/41**(57) **ABSTRACT**

The WALLET SERVICE ENROLLMENT PLATFORM APPARATUSES, METHODS AND SYSTEMS ("WSEP") facilitates the enrollment of payment accounts in a consumer's virtual wallet. The consumer may be logged into their payment account issuer's web site and designate one or more payment accounts for enrollment in a virtual wallet. The issuer may then share account, billing and/or other relevant information with the virtual wallet provider to facilitate the enrollment of the designated payment accounts in the virtual wallet. The WSEP may also be configured to facilitate the creation and funding of pre-paid accounts in a consumer's virtual wallet.



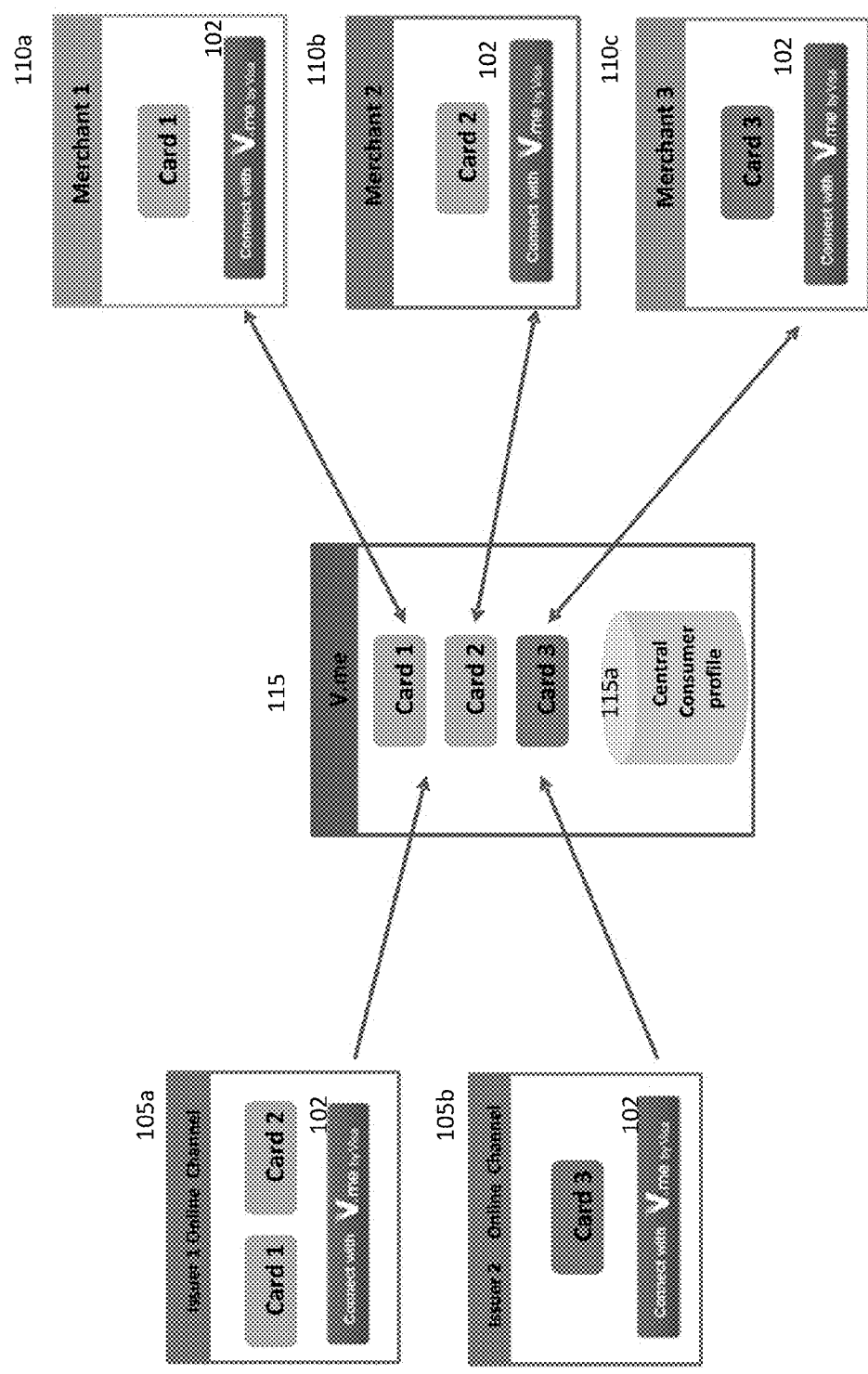


Figure 1: Example Service Connections

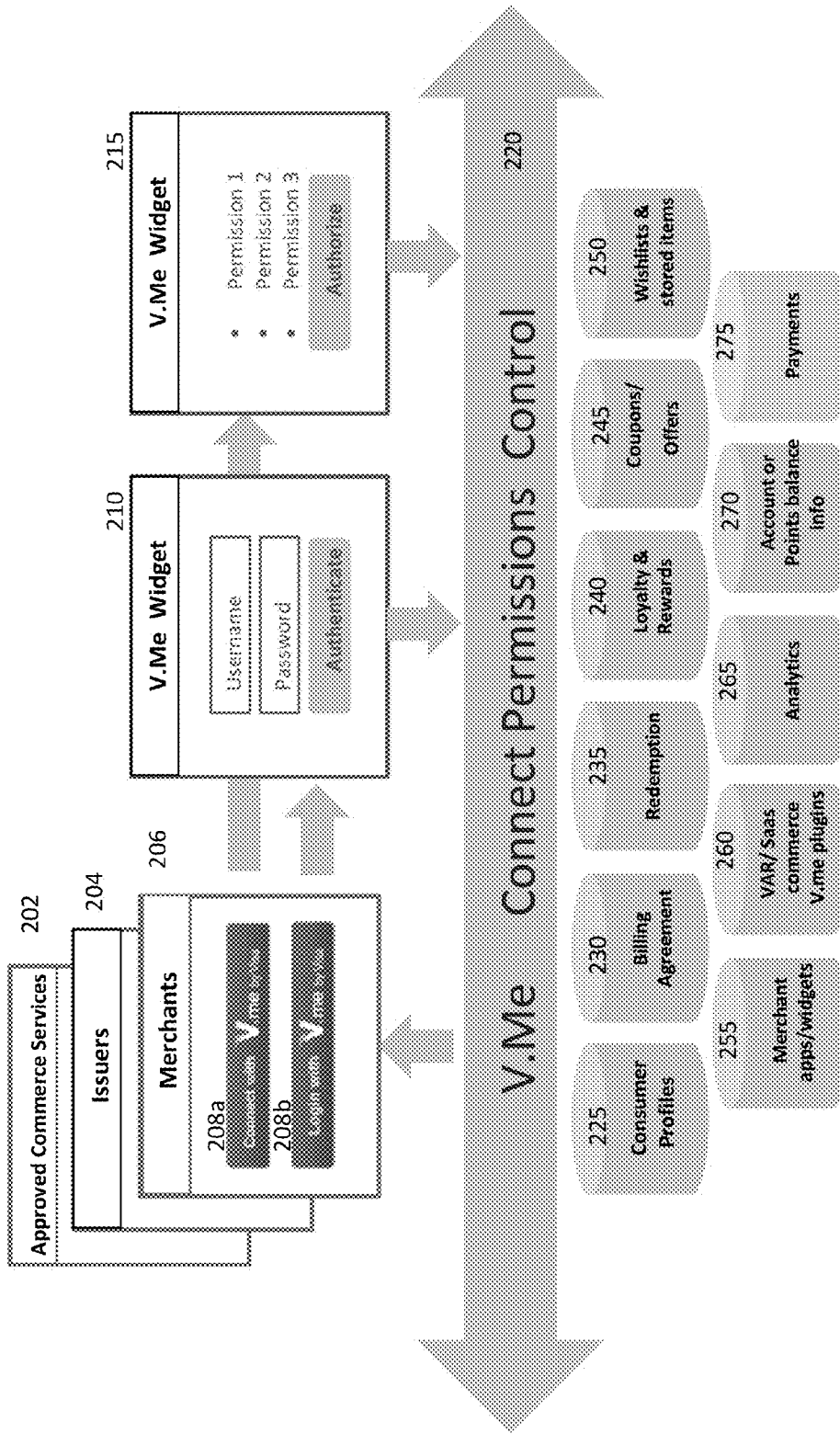


Figure 2: Example Connect Architecture

[NORDSTROM.COM](#) [https://www.nordstrom.com/Sign-up.aspx?utm_source=Referral&utm_medium=CPE](#) • fendi bags

SPEND OR STYLE. NOT SHIPPING. Free shipping and returns. [Get details](#)

Welcome to Nordstrom. Would you like to sign up? Your Account / Our Stores & Events / Gift Card Registry / Shop / Shopping Bags / 1 Item

NORDSTROM

FASHION REWARDS. On shopping get rewards. [Get details](#)

SIGN UP FOR THE NEW CUSTOMER > WELCOME TO NORDSTROM >

NEW CUSTOMER?

CREATE AN ACCOUNT 305

Create an account to take advantage of the features and benefits that make shopping easier and faster. Learn more about [Nordstrom Rewards](#).

Email Name:

E-mail:

Confirm E-mail:

Password:

Validation:
Your password is used throughout your shopping experience. To protect your information, we require that you include both letters and numbers.

Confirm Password:

Zip / Postal Code:

C Gender: Select

310

CREATE AN ACCOUNT WITH V.I.N.E

Do you have a V.I.N.E account? You can connect it to our website or discuss instantly with Nordstrom.

C Gender: Select

☒ YES! Send me e-mail updates from Nordstrom.com trends, products and promotions online and in store.

310a

[Create my V.I.N.E profile](#)

305

CREATE AN ACCOUNT 305

Create an account to take advantage of the features and benefits that make shopping easier and faster. Learn more about [Nordstrom Rewards](#).

Email Name:

E-mail:

Confirm E-mail:

Password:

Validation:
Your password is used throughout your shopping experience. To protect your information, we require that you include both letters and numbers.

Confirm Password:

Zip / Postal Code:

C Gender: Select

305a

☒ YES! Send me e-mail updates from Nordstrom.com about the latest trends, products and promotions online and in store.

305a

[Create my account](#)

Figure 3: Example Merchant Account Creation UI

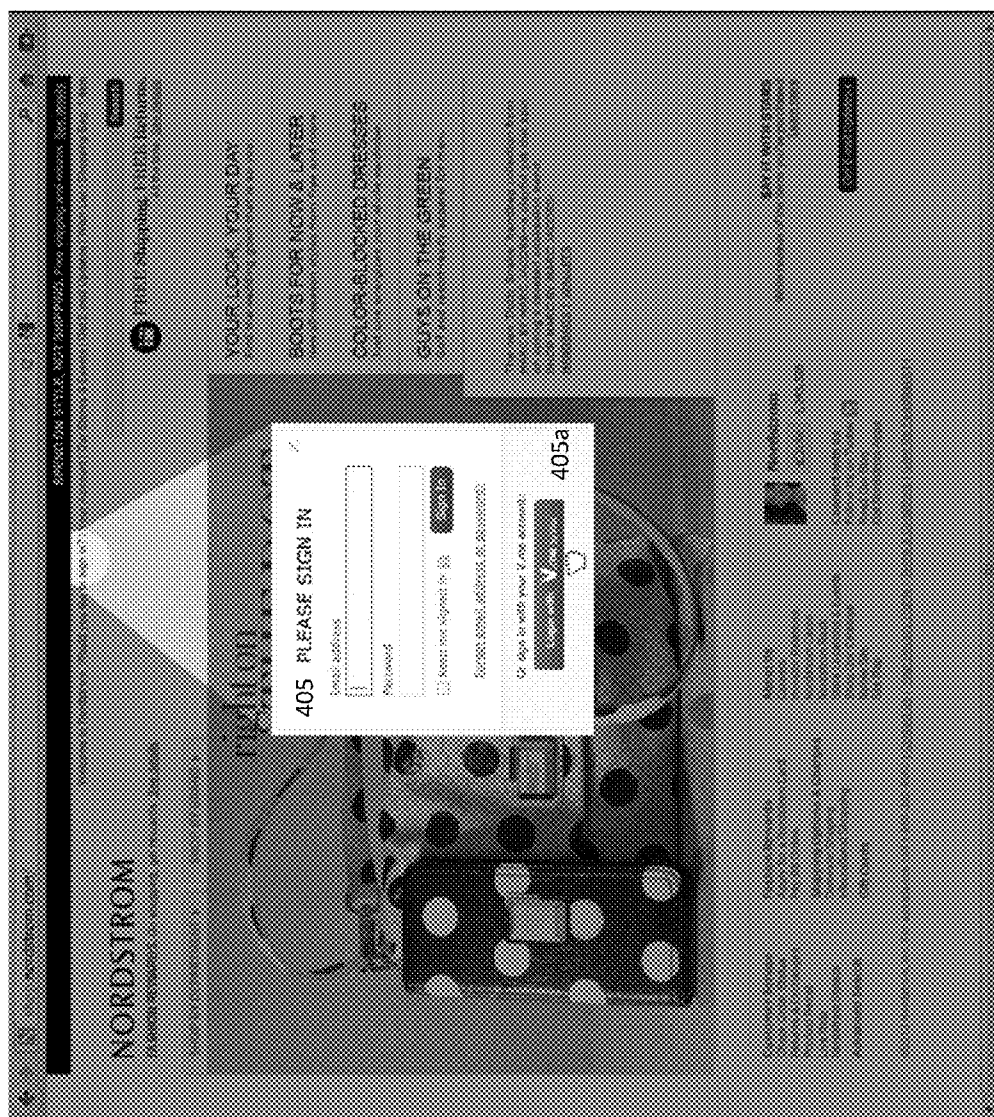


Figure 4: Example Merchant Account Login UI

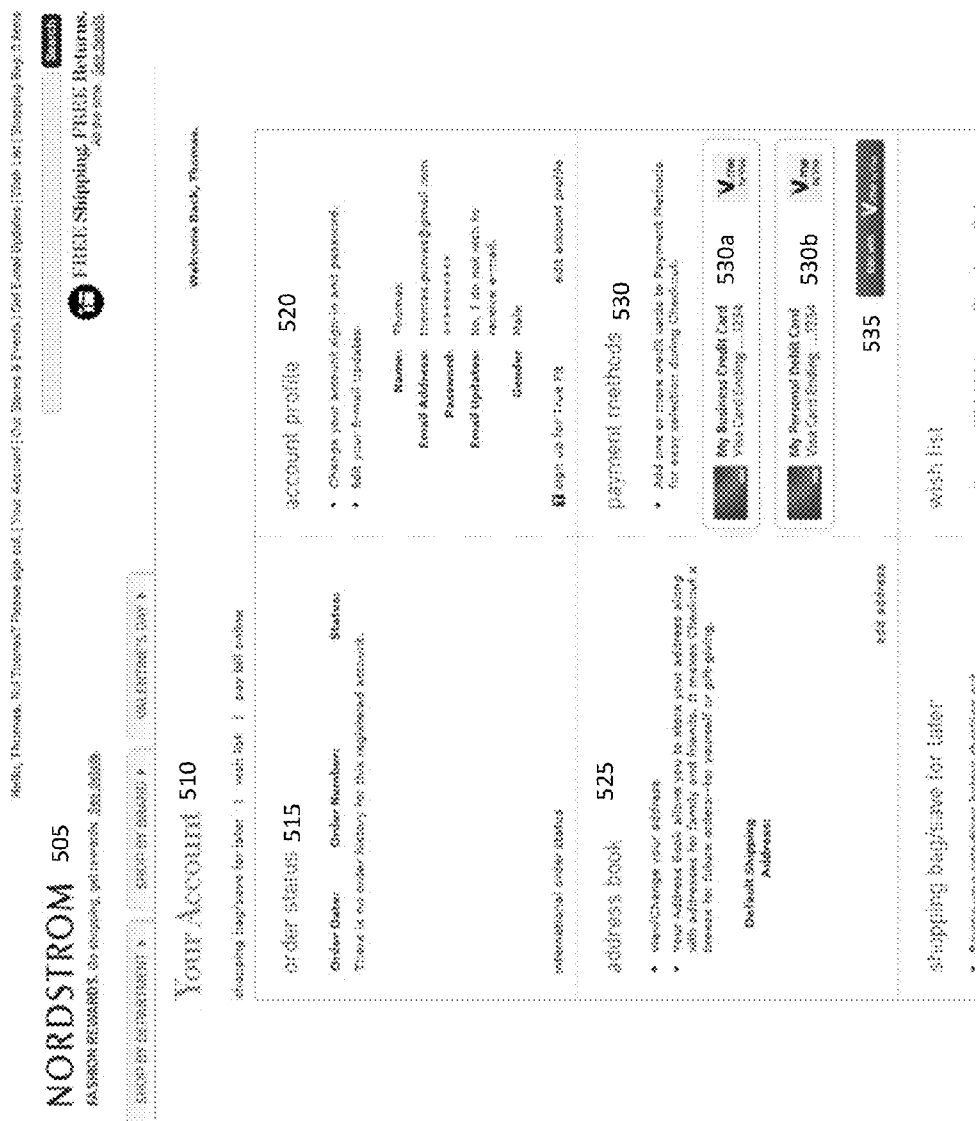


Figure 5: Example Account Preference Management UI

605

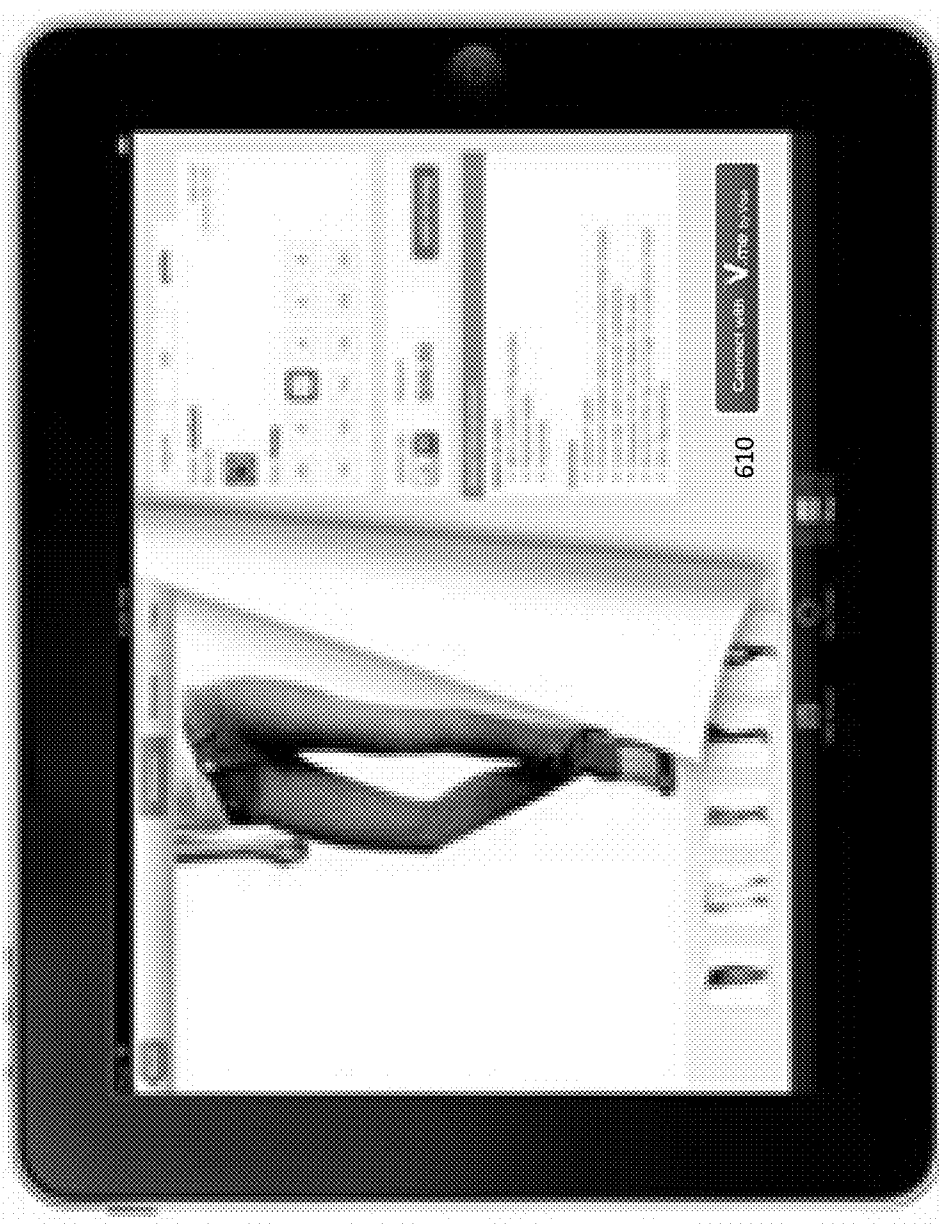


Figure 6: Example Cross Channel UI

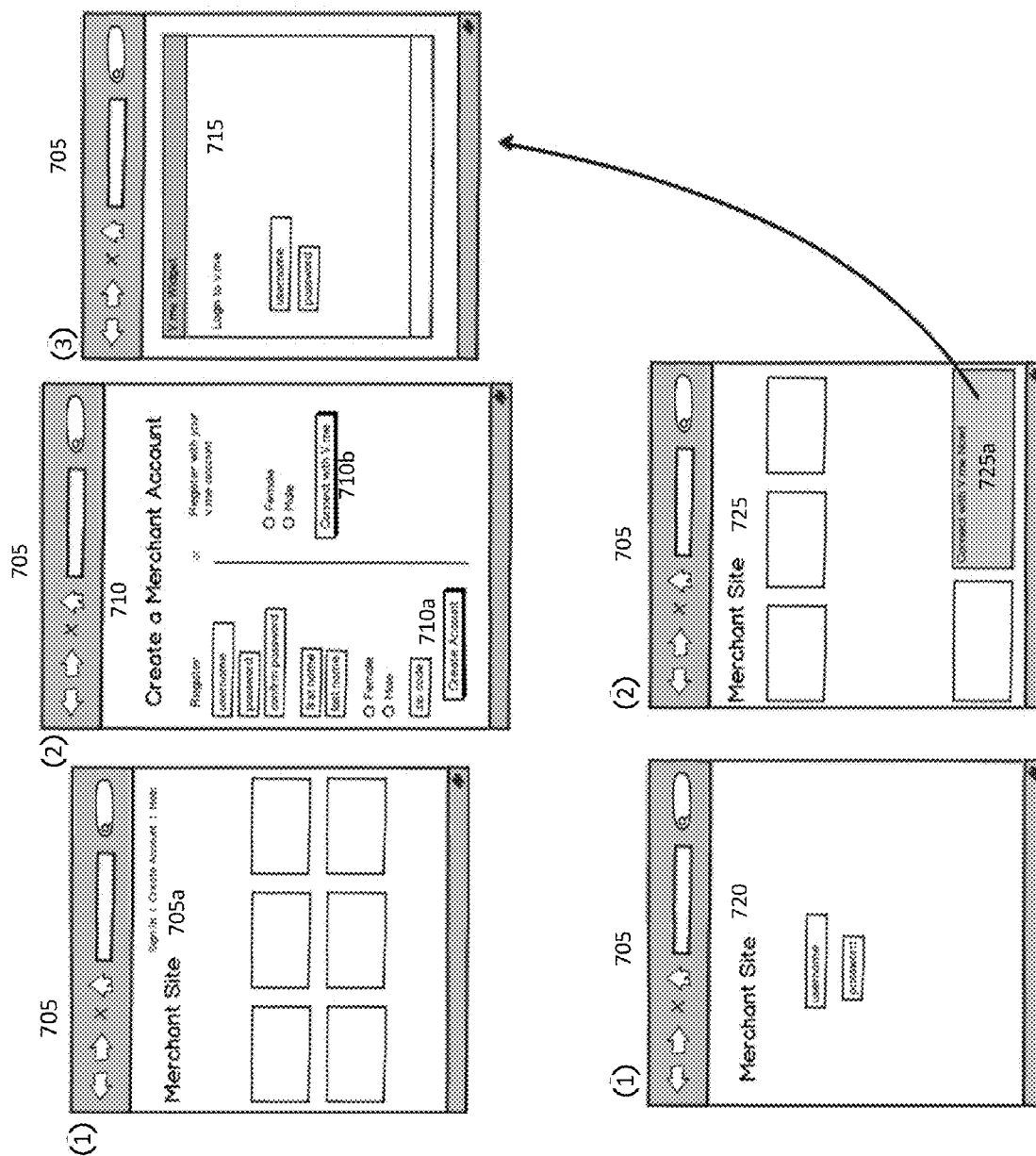


Figure 7a: Example Sign In UI

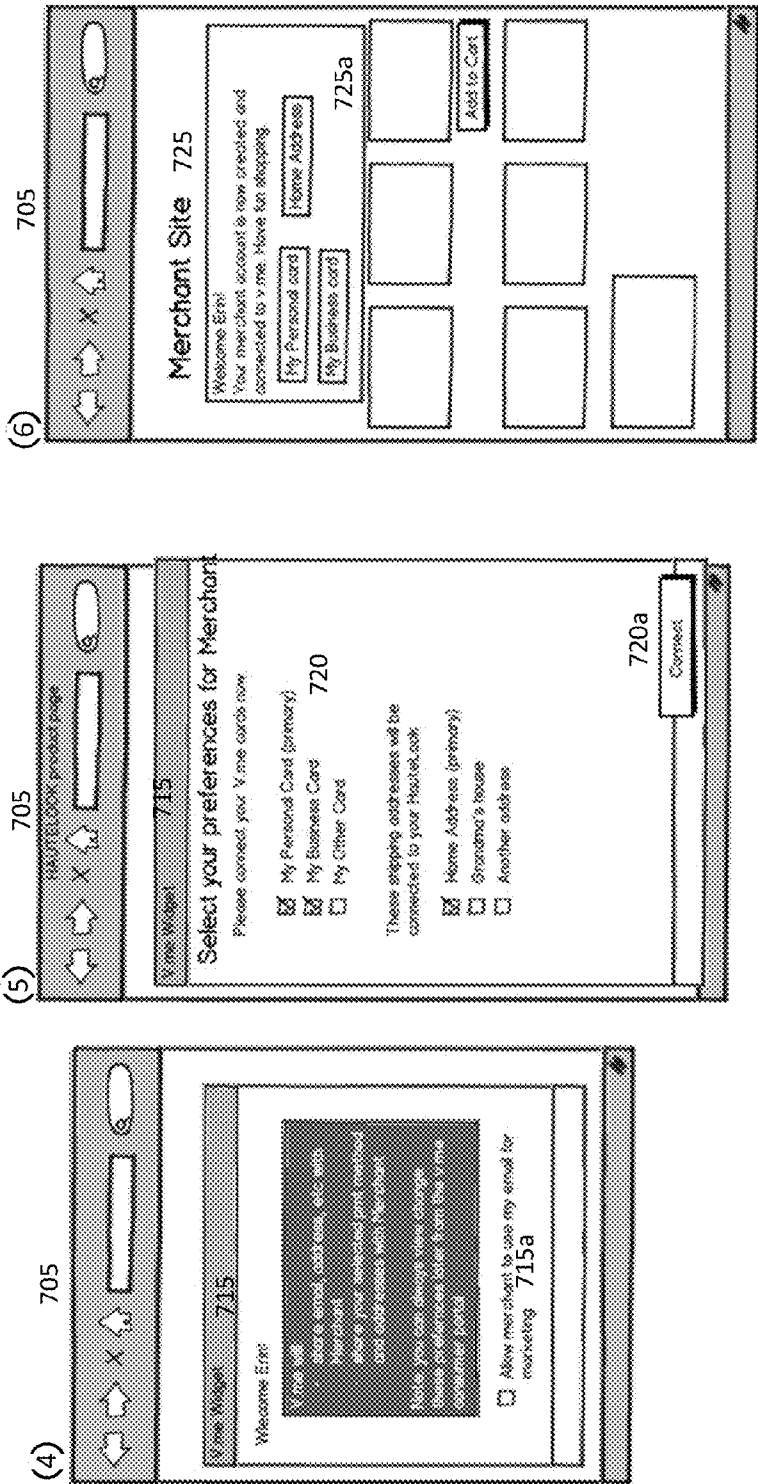


Figure 7b: Example Account Management UI

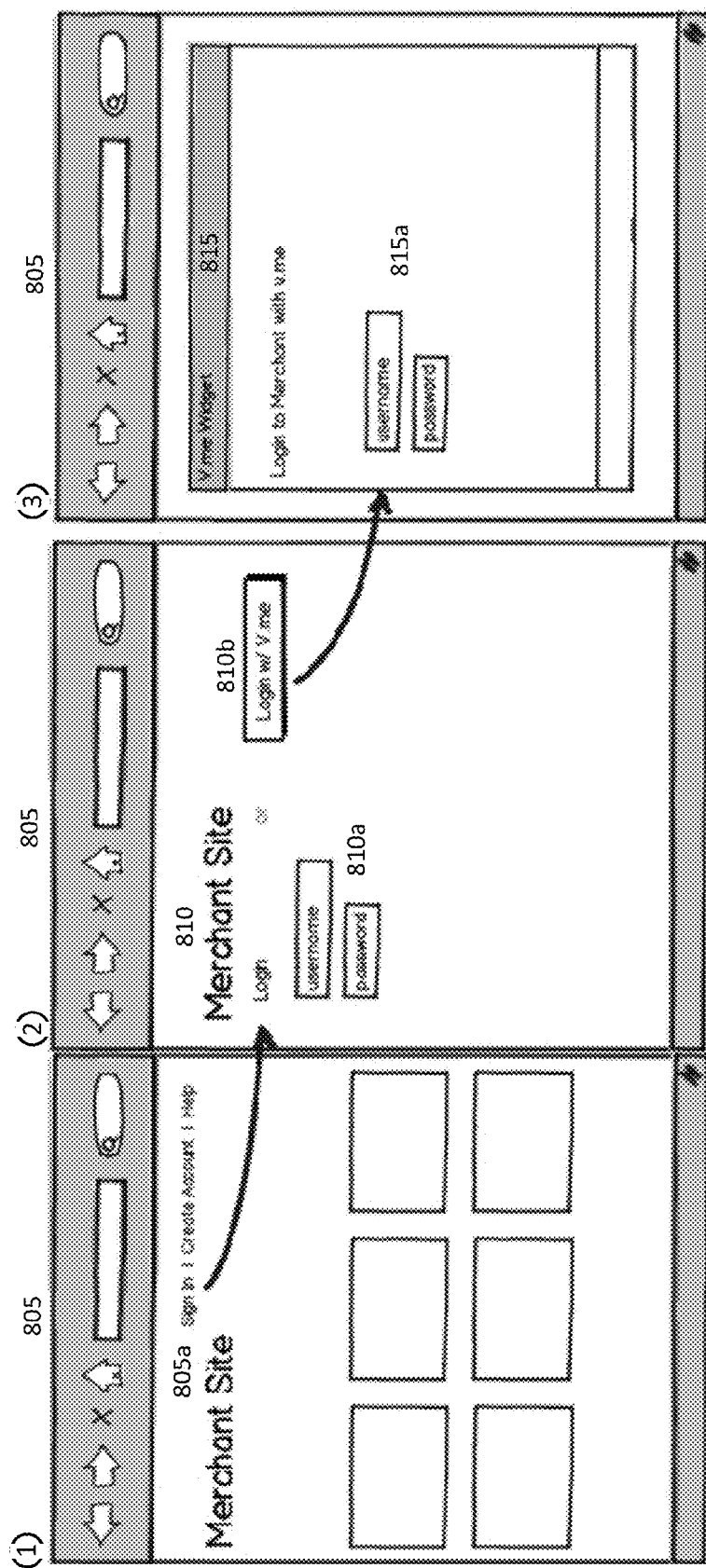


Figure 8a: Example Sign In UI

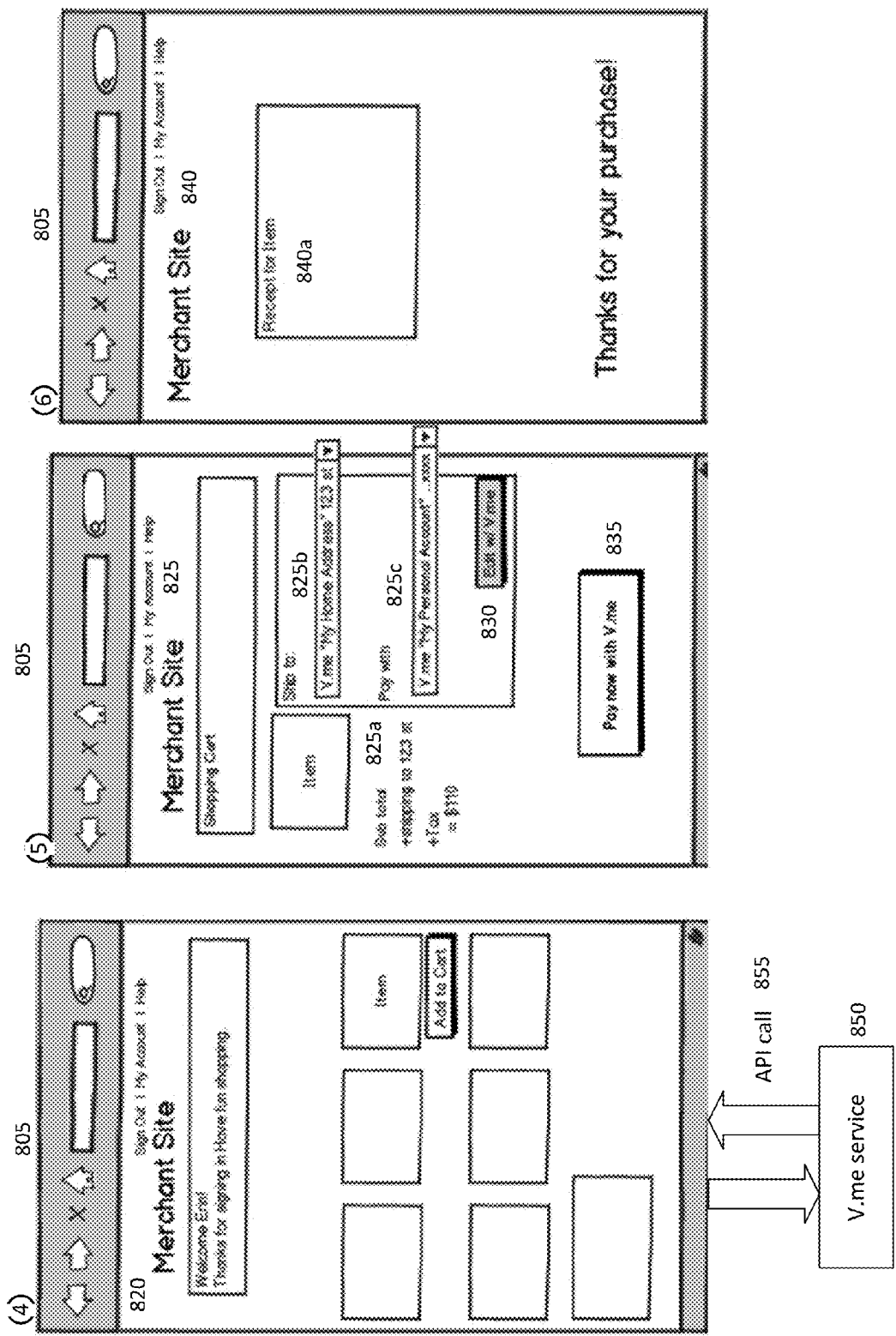


Figure 8b: Example Check-out UI

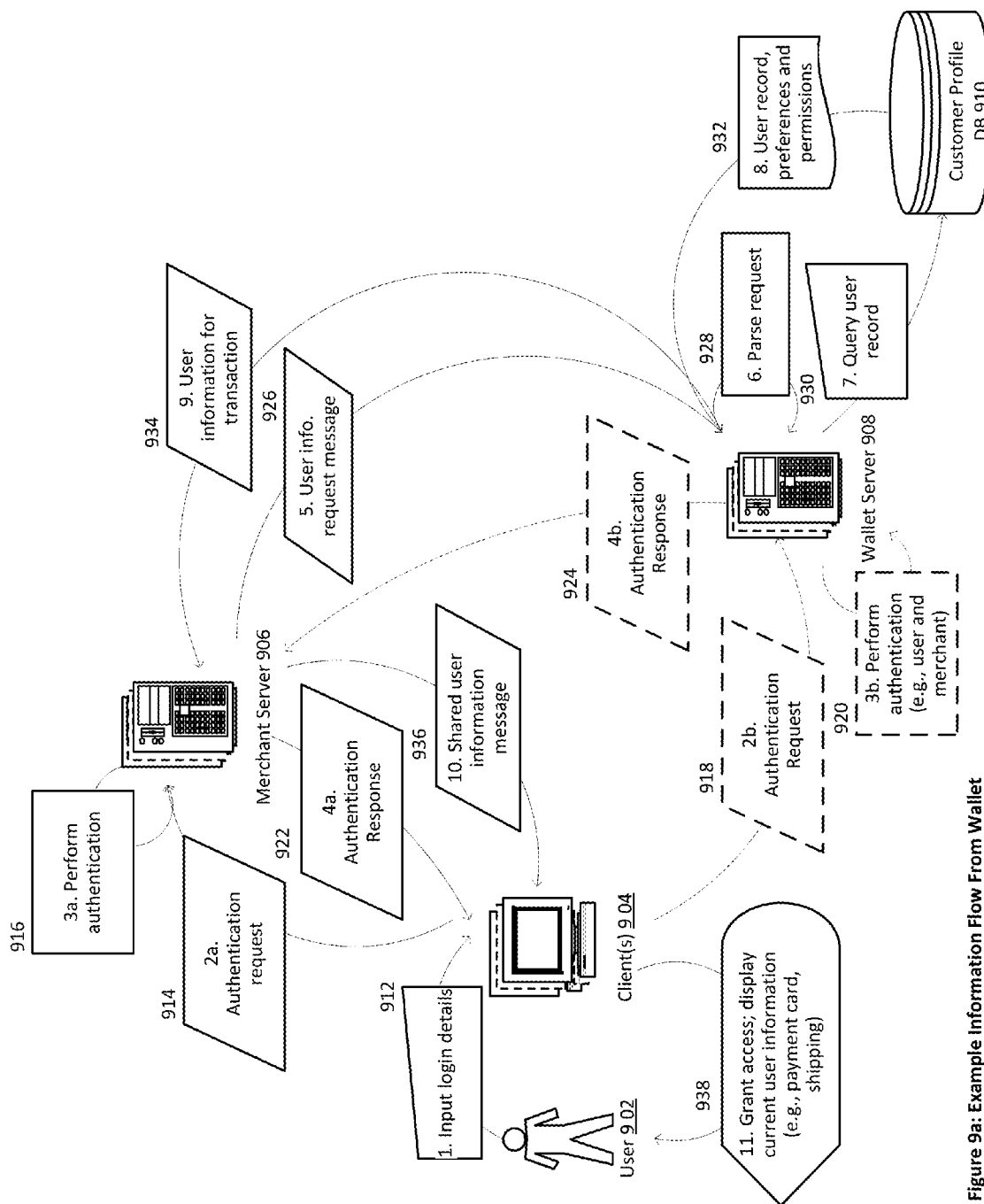


Figure 9a: Example Information Flow From Wallet

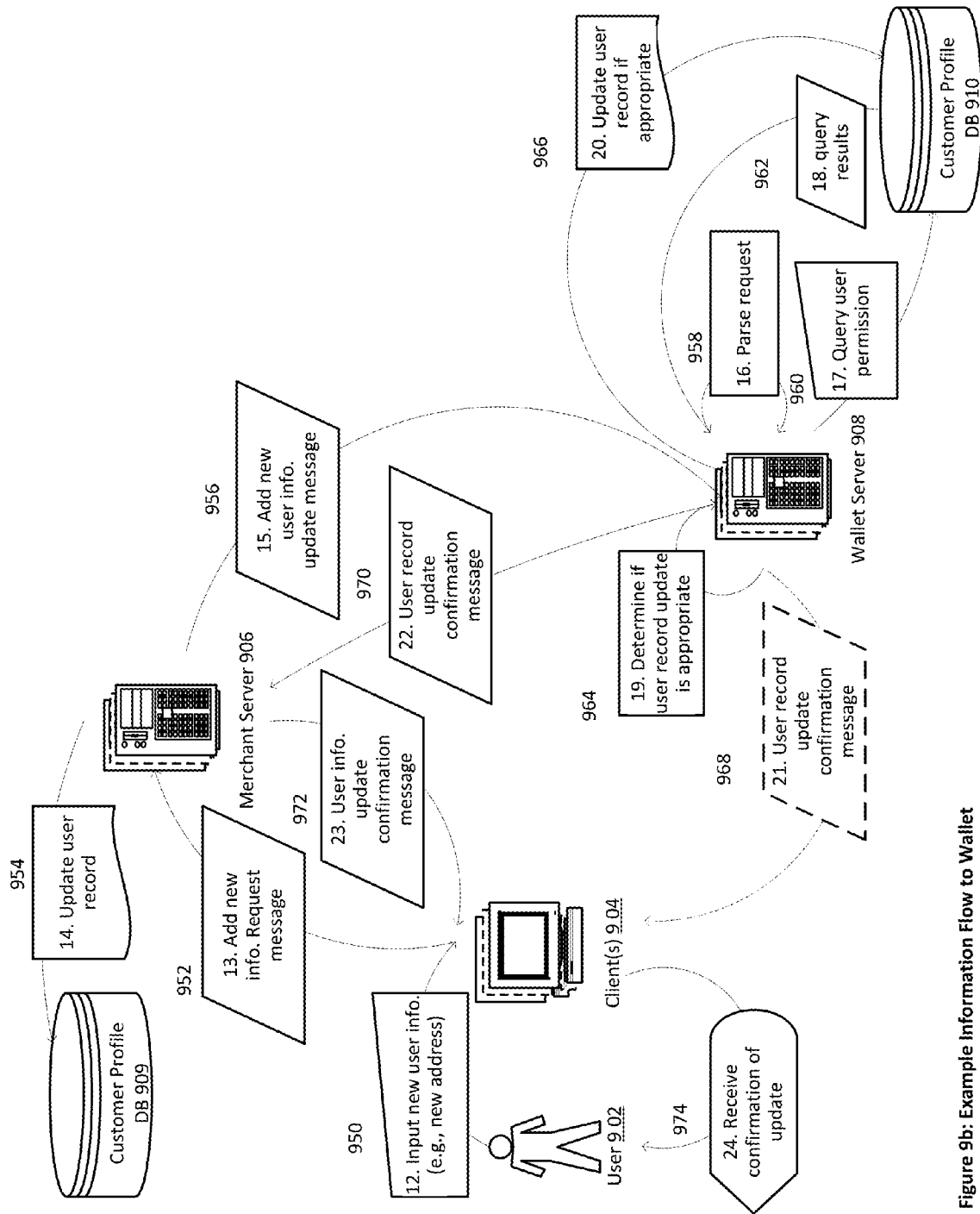


Figure 9b: Example Information Flow to Wallet

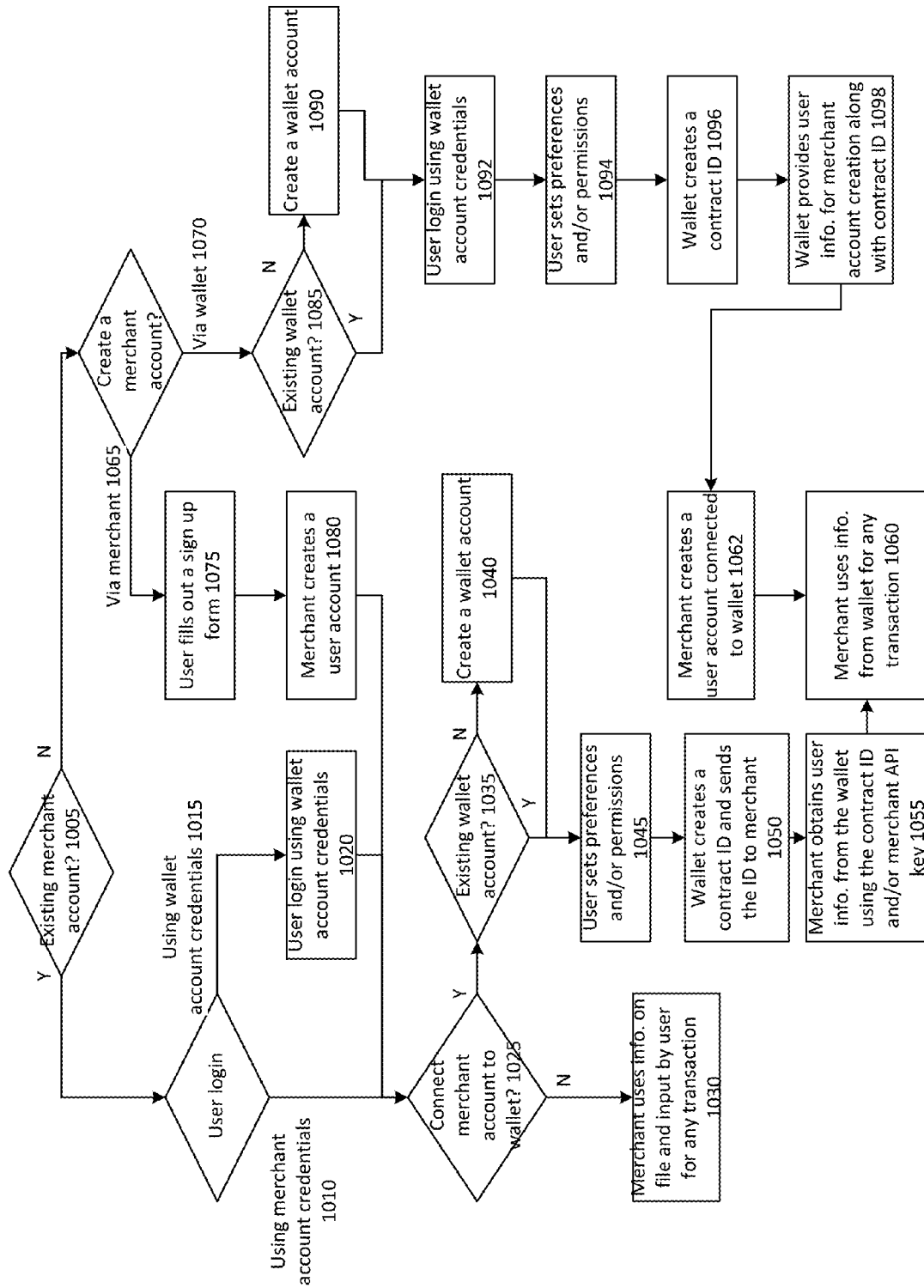


Figure 10: Example Account Creation and Management (ACM) Component

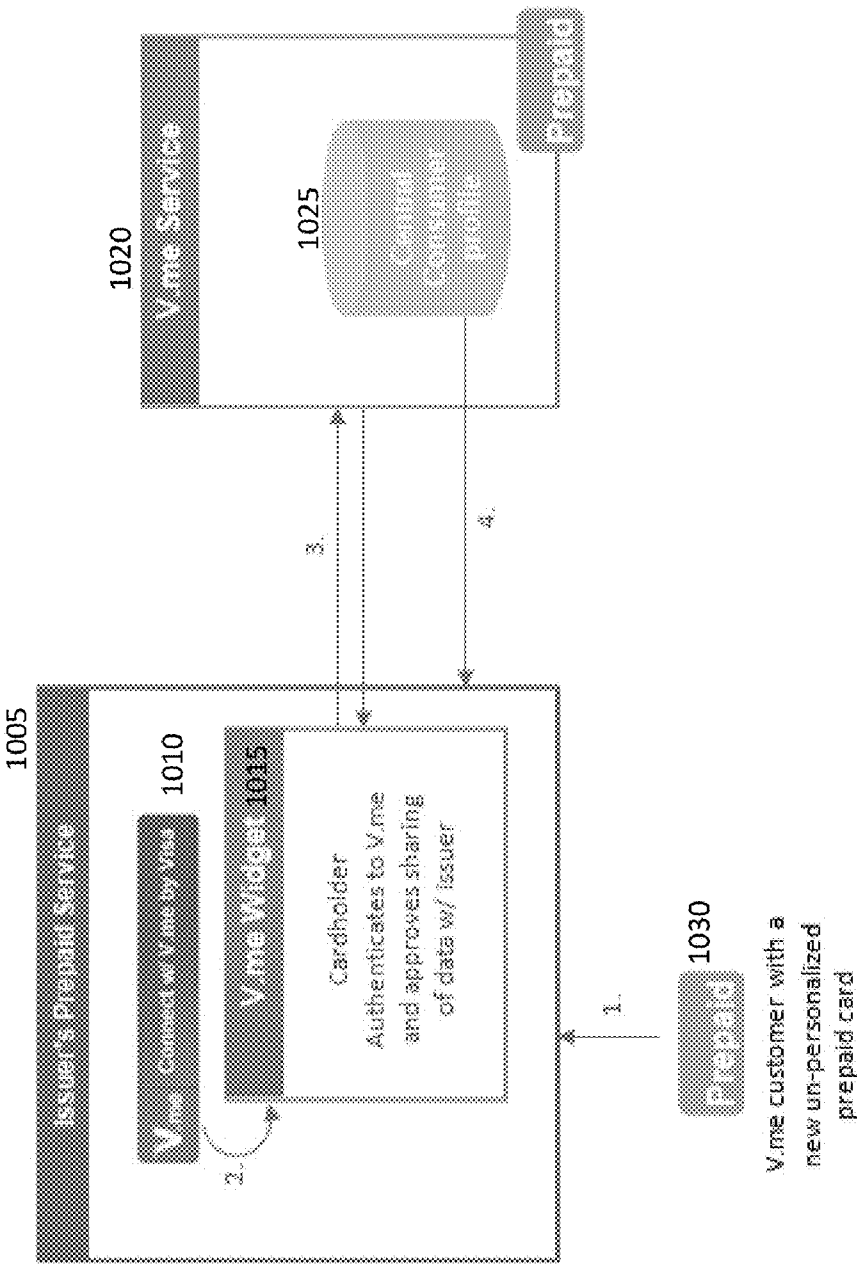


Figure 11: Example Prepaid Account Personalization

1205	1210	1215	1220	1225
Service providers	Payment methods	Shipping address	Share my	Permissions
1205a HauteLook	Card1 Image XXXX-XXXX-XXXX-XXXX ☐ My personal card	☐ Home Address (Primary) 200 Henry Street, Portland, Oregon 97126	☐ Name	☒ Execute transaction
1205b BestBuy	Card2 Image XXXX-XXXX-XXXX-XXXX ☐ My Visa cash back card	☐ Grandma's Address 224 Brookline Street, New York, NY 10112	☐ Primary Email address	☒ Confirm profile changes with me
1205c Nordstrom	Card3 Image XXXX-XXXX-XXXX-XXXX ☐ My prepaid card	☐ Work Address 204 GW Street, New York, NY 10086	☐ Work Email address	☒ Authorized to bill
1205d Gap	Card4 Image XXXX-XXXX-XXXX-XXXX ☐ My business card	☐ Add Another Address	☐ Information for account	☐ Subscription payments
1205e Ann Taylor	Card5 Image XXXX-XXXX-XXXX-XXXX ☐ Add new card		☒ Loyalty Programs	☐ AAA membership
Portland General 1205f Electric			☐ Hilton HHonors	☐ Shopkick
1205g Amazon			☒ Wish lists	☐ Amazon
1205h Verizon			☐ Points balance	☐ 1220h
1205i Home Depot			☐ Allow write access	
1205j Fresh Direct			☐ Allow write access	

Figure 12: Example W-Connect Settings Control Panel

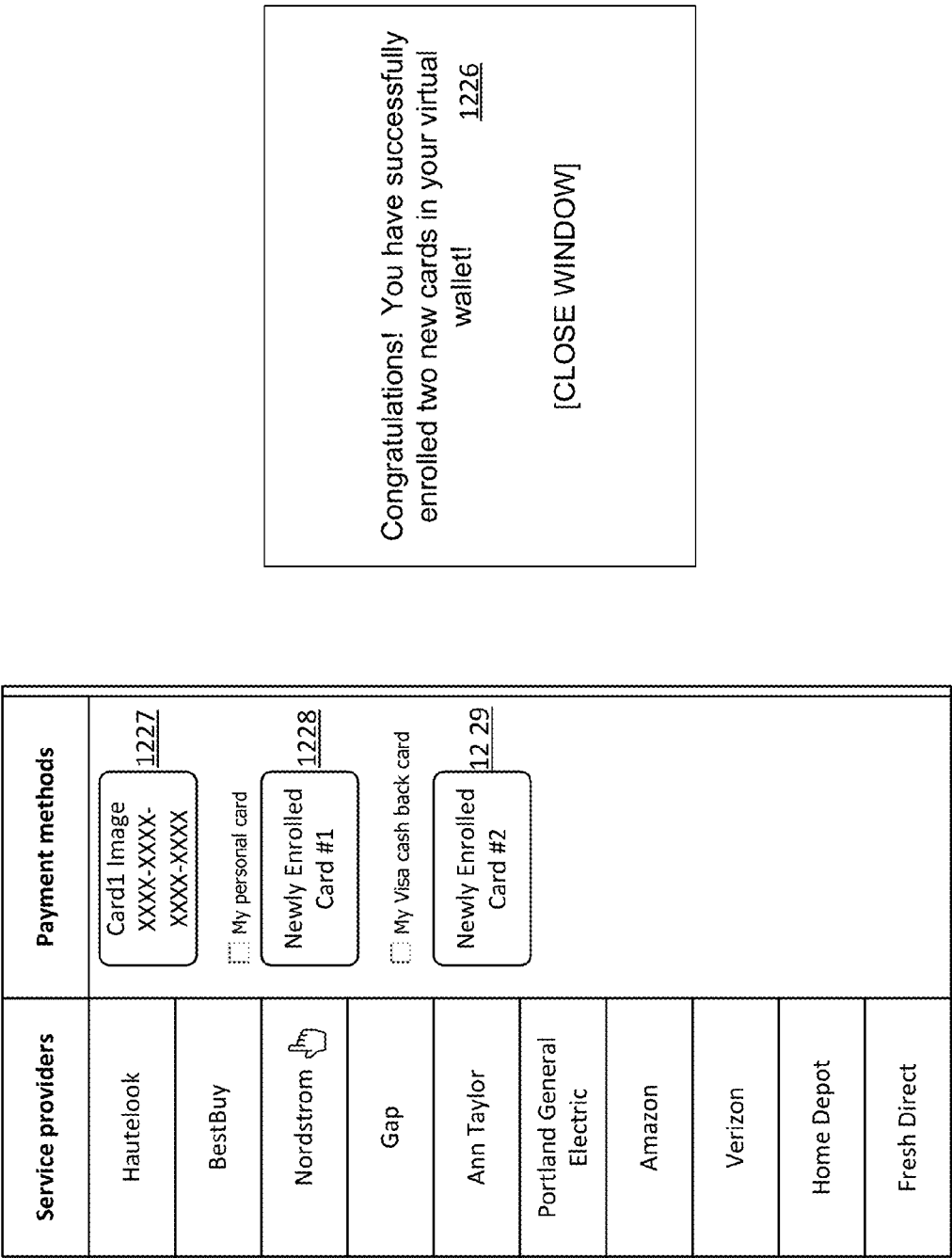


Figure 12a: Example New Card Enrollment Success

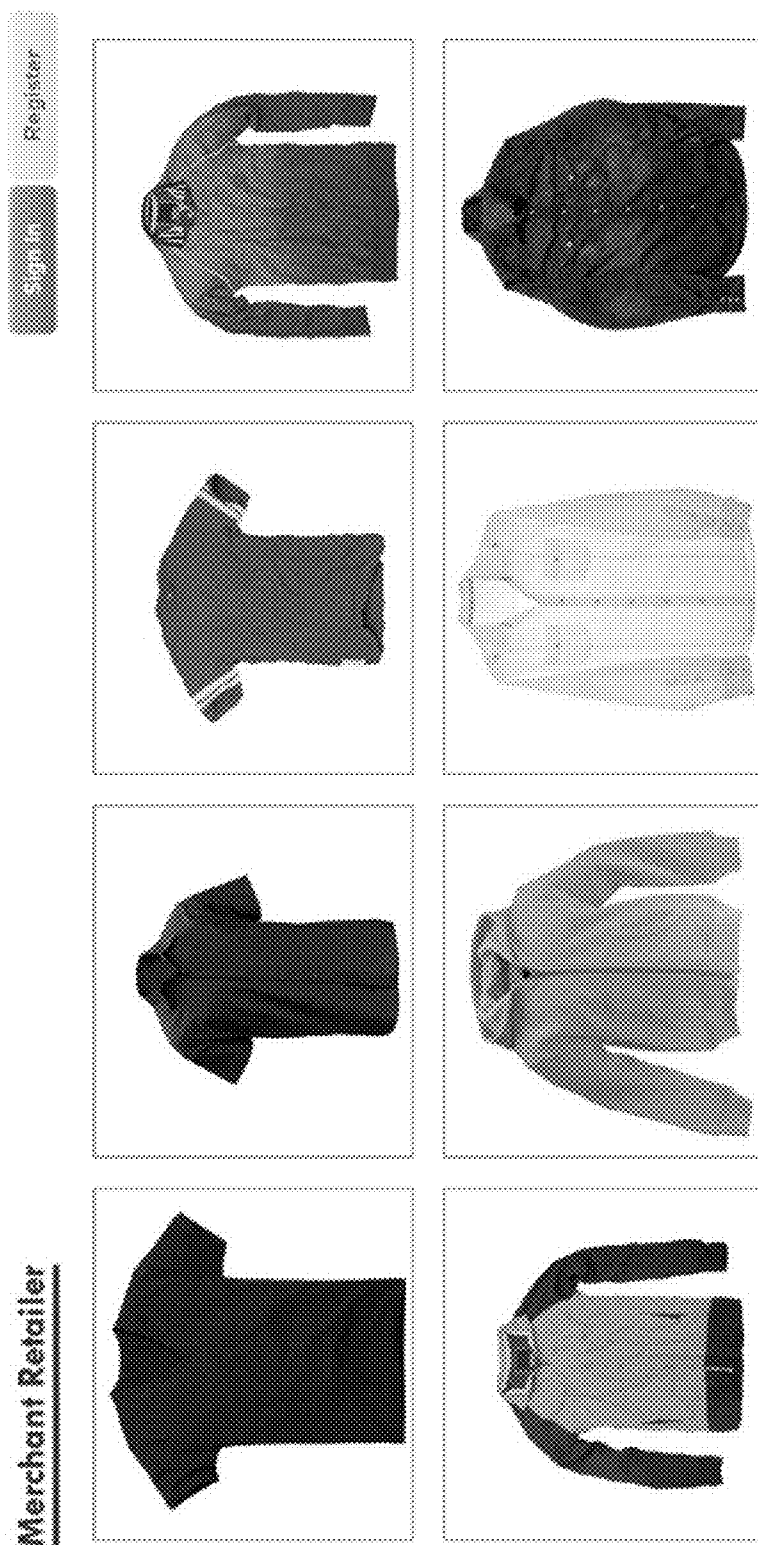


Figure 13a: Example W-Connect User Interface

Merchant Retailer

Login

username

password

Login

Or sign in with your V me account

sign in with **V** me by Visa

Figure 13b: Example W-Connect User Interface

Merchant Retailer

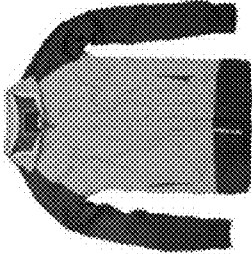
Checkout

Payment and Shipment

Payment Method:
TBD

Shipping Address:
TBD

Cart



Track Jacket

Qty:

Size: Small

\$59.00

Subtotal

\$59.00

Tax

\$5.23

Shipping

\$9.00

Order Total

\$72.23

W-Connect

Free to You

Figure 13c: Example W-Connect User Interface

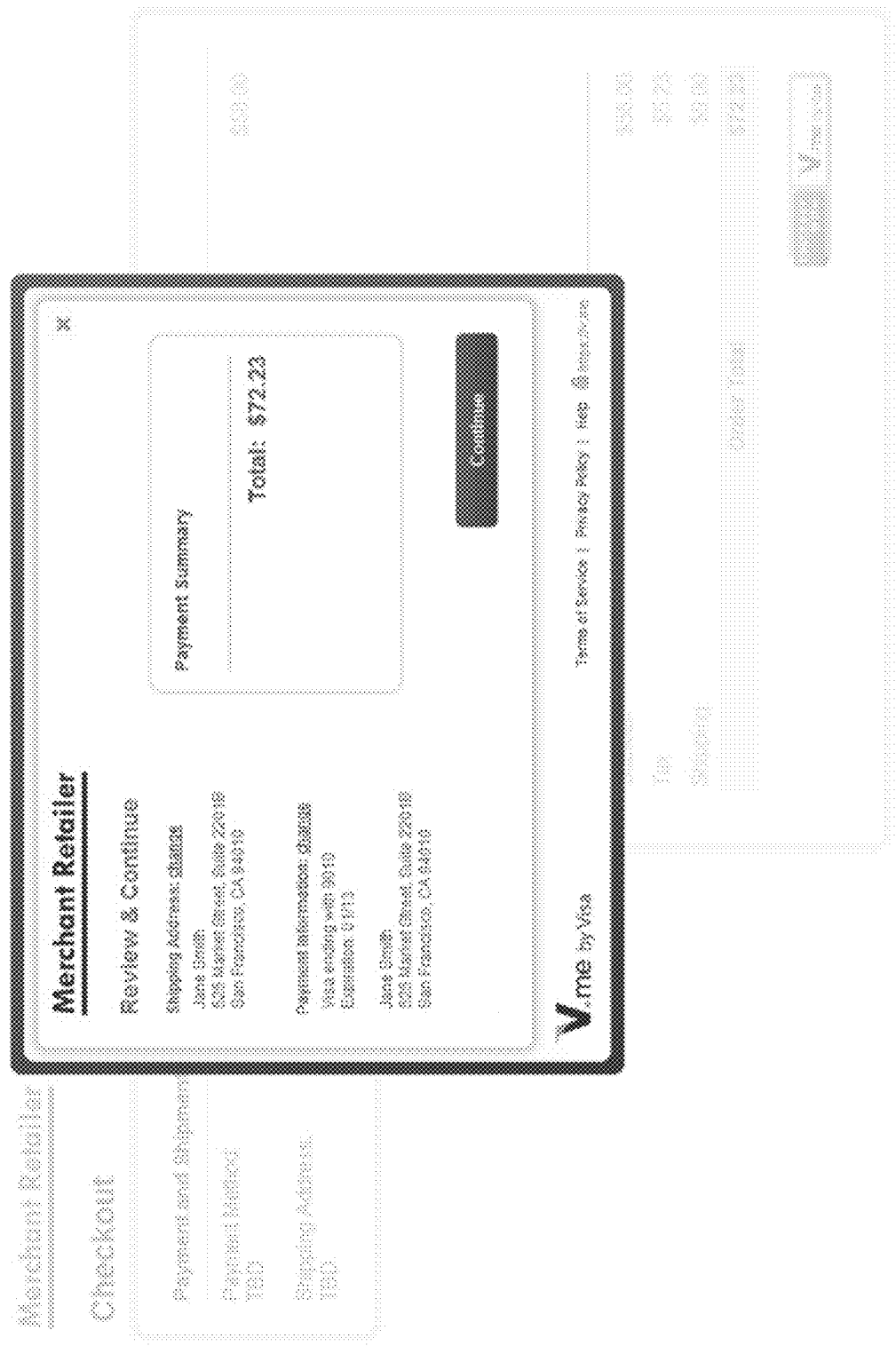


Figure 13d: Example W-Connect User Interface

Merchant Retailer

Checkout

Payment and Shipment

Payment Method:

V

me by you

Shipping Address:

Jane Smith
525 Market Street, Suite 22010
San Francisco, CA 94010

Cart

Track Jacket

Qty: 1

Size: Small

\$58.00

Subtotal

\$58.00

Tax

\$5.23

Shipping

\$9.00

Order Total

\$72.23

me by you

V

me by you

Figure 13e: Example W-Connect User Interface

Merchant Retailer

Order Confirmation

Ⓢ

Congratulation! Your order is successfully submitted. Your order confirmation ID is 675784327

Payment Information:

V

me by you

Ship To:

Jane Smith
525 Market Street, Suite 22018
San Francisco, CA 94010

Shipping Method:

First Class: (\$9.00)

Track Jacket

QTY

PRICE

TOTAL

1

\$58.00

\$58.00

Subtotal

\$58.00

Shipping

\$9.00

Tax

\$5.23

Total

\$72.23

Figure 13f: Example W-Connect User Interface

Merchant Retailer

Register

username

password

confirm password

first name

last name

☐ Female

☐ Male

zip code

Create Account

Or

Register with your V.me account

Email

☐ Friends

☐ Me

☐ Allow Merchant to use my Email for marketing

Log in with V.me by /os

Figure 13g: Example W-Connect User Interface

Merchant Retailer

Login to V.me

Email Address

Password

[Forgot your password?](#)

Sign In

V.me by Visa

[Terms of Service](#) | [Privacy Policy](#) | [Help](#) [https://v.me](#)

[Register](#)

[Merchant](#)

[Partner](#)

[Cardless pin](#)

[Pay using](#)

[Add more](#)

[Feature](#)

[Help](#)

[About Us](#)

[Contact Us](#)

Figure 13h: Example W-Connect User Interface

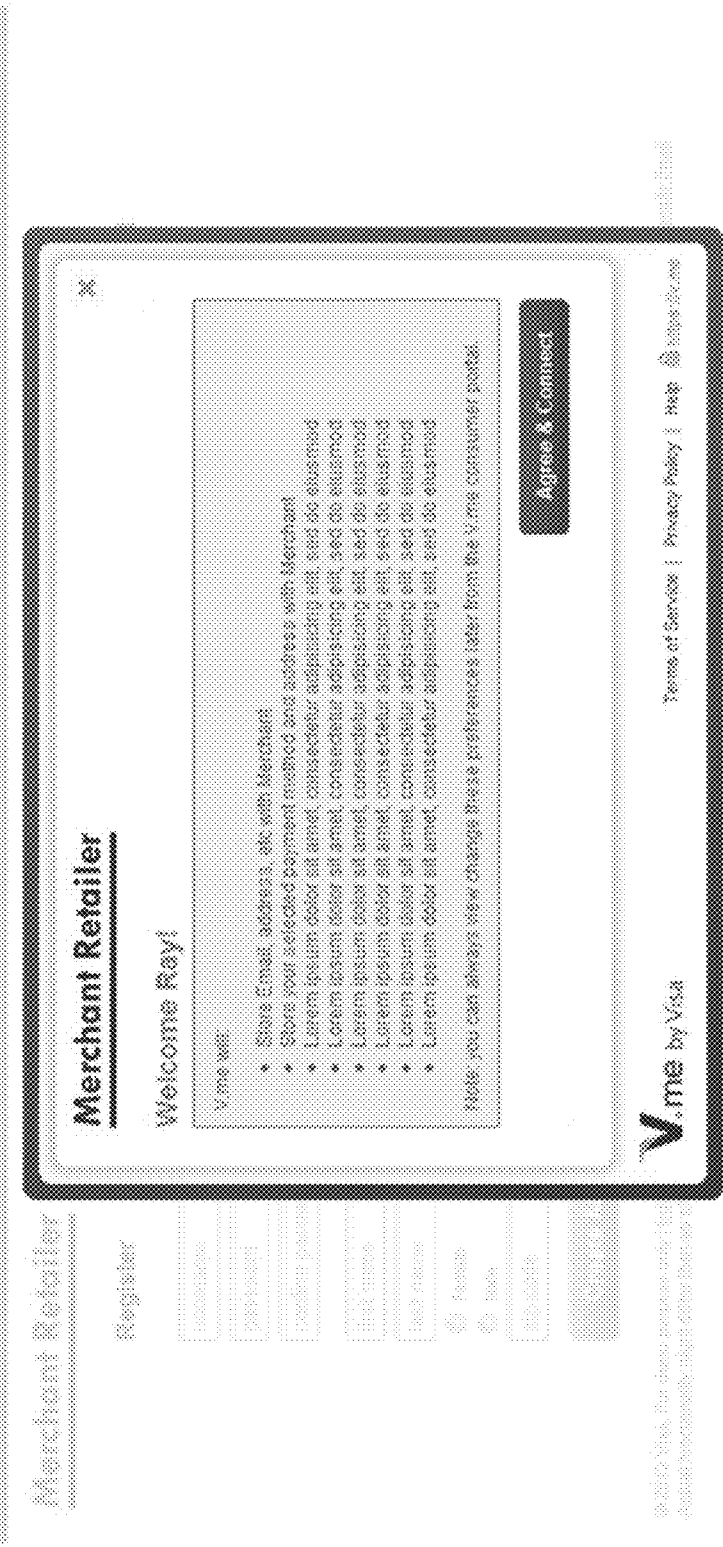


Figure 13i: Example W-Connect User Interface

Review Cart

Chat with a Sales Associate

Chat Now

Call

You're almost done! Simply review your cart below, then click the "Checkout" button.

If you're not ready to check out now, you can save your cart by entering your email below.

14 01

14 02

Due Today

Monthly

2 GB Shared Data Unlimited Talk & Text

Remove This Plan

Device 1



Device

Apple iPhone 4S, 16GB in White

Special Offer

One-time Activation Fee Due on Your First Bill

Change Device

\$199.99

\$35.00

2-year contract

Plan Details

Monthly Line Access, Share Everything - Smartphone Line Access

Change Plan

\$48.00

Connect with V

14 03

Figure 14: Example Reference Transaction Enrollment

Review Cart

Chat with a Sales Associate

Chat Now

Call

You're almost done! Simply review your cart below, then click the "Checkout" button.

If you're not ready to check out now, you can save your cart by entering your email below.

2 GB Shared Data Unlimited Talk & Text

Remove This Plan

Device 1

Device



Apple iPhone 4S 16GB in White

Special Offer!

One-time Activation Fee Due

Change Device

Plan Details

Monthly Line Access, Share

Change Plan

Due Today

Monthly

\$60.00

Email:

Password:

☐ Check here to create a v.me account

14 05

14 06

Login

Connect with v.me by text

14 04

Figure 14a: Example Reference Transaction Enrollment

Connect with **v.me** by Visa

Create an account with this merchant to complete your transaction:

Name:

Address:

City:

State:

ZIP:

Card Number:

Expiration Date:

☐ Click here to also create a v.me account

Create Account with Merchant

14 10

14 07

14 08

Connect with **v.me** by Visa

Due Today

Monthly

\$60.00

\$199.99

\$6.25.00

\$648.00

Figure 14b: Example Reference Transaction Enrollment

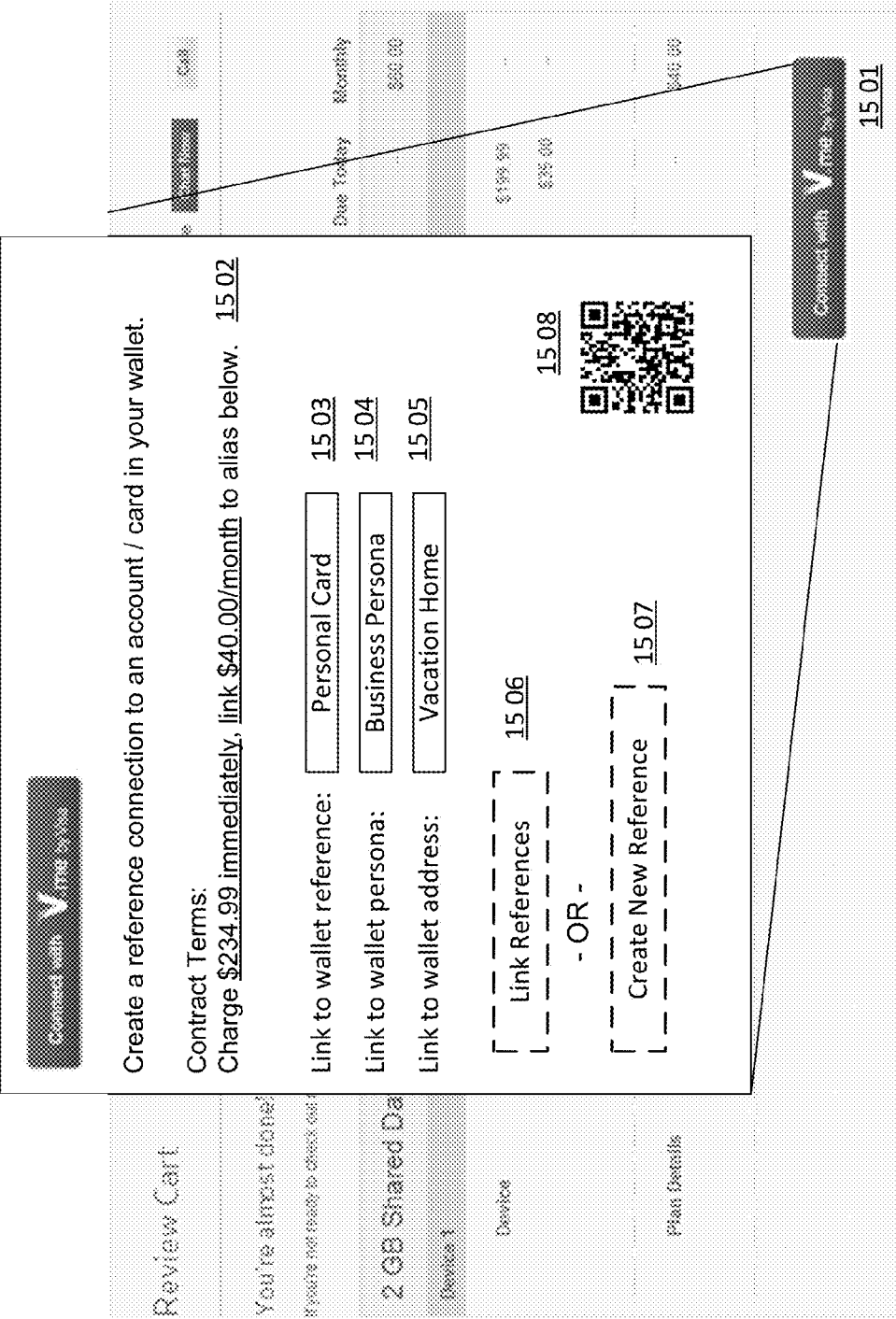


Figure 15: Example Reference Transaction Enrollment

<u>Manage Merchant Reference Contracts</u>	
Reference payment linked to: "Personal Card" <u>16 09</u>  New Card Number: New Exp. Date: <u>16 03</u> Update <u>16 02</u>	NORDSTROM Nickname at merchant: Fun Card <u>16 04</u> Terms: <u>16 05</u> Flexible recurring commerce Expiration: None History: <u>16 06</u> 2/13/11 - \$59.99 – view details 8/8/12 - \$132.21 – view details Revoke Access <u>16 07</u> amazon.com Nickname at merchant: Personal Card Terms: Managed Subscription - \$40.00/mo Expiration: July '14 History: 6/13/11 - \$40.00 – view details 9/18/12 - \$40.00 – view details Cancel Subscription <u>16 08</u> <u>16 01</u>

Figure 16: Example Reference Management Console

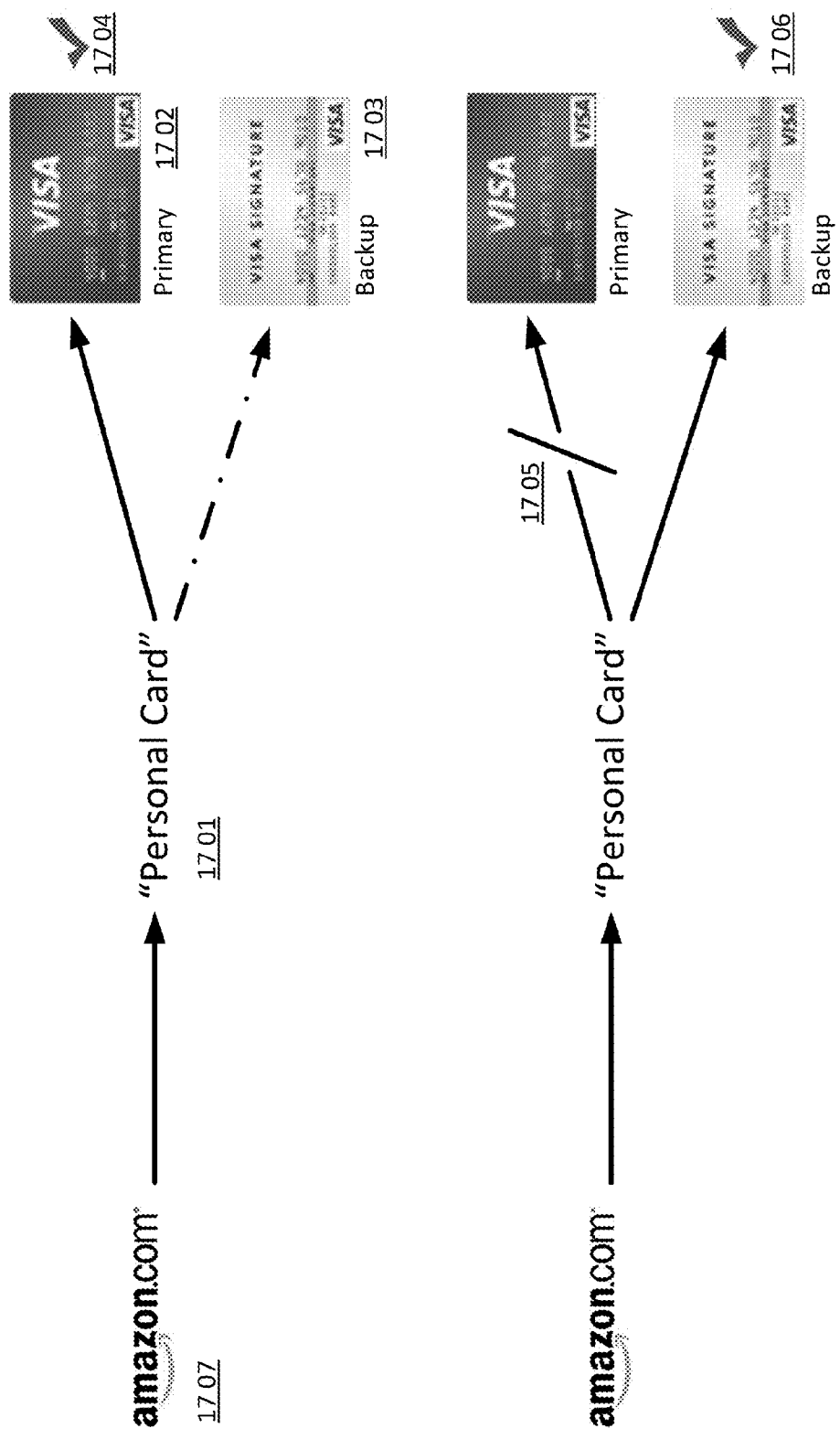


Figure 17: Example Reference Failover Transaction

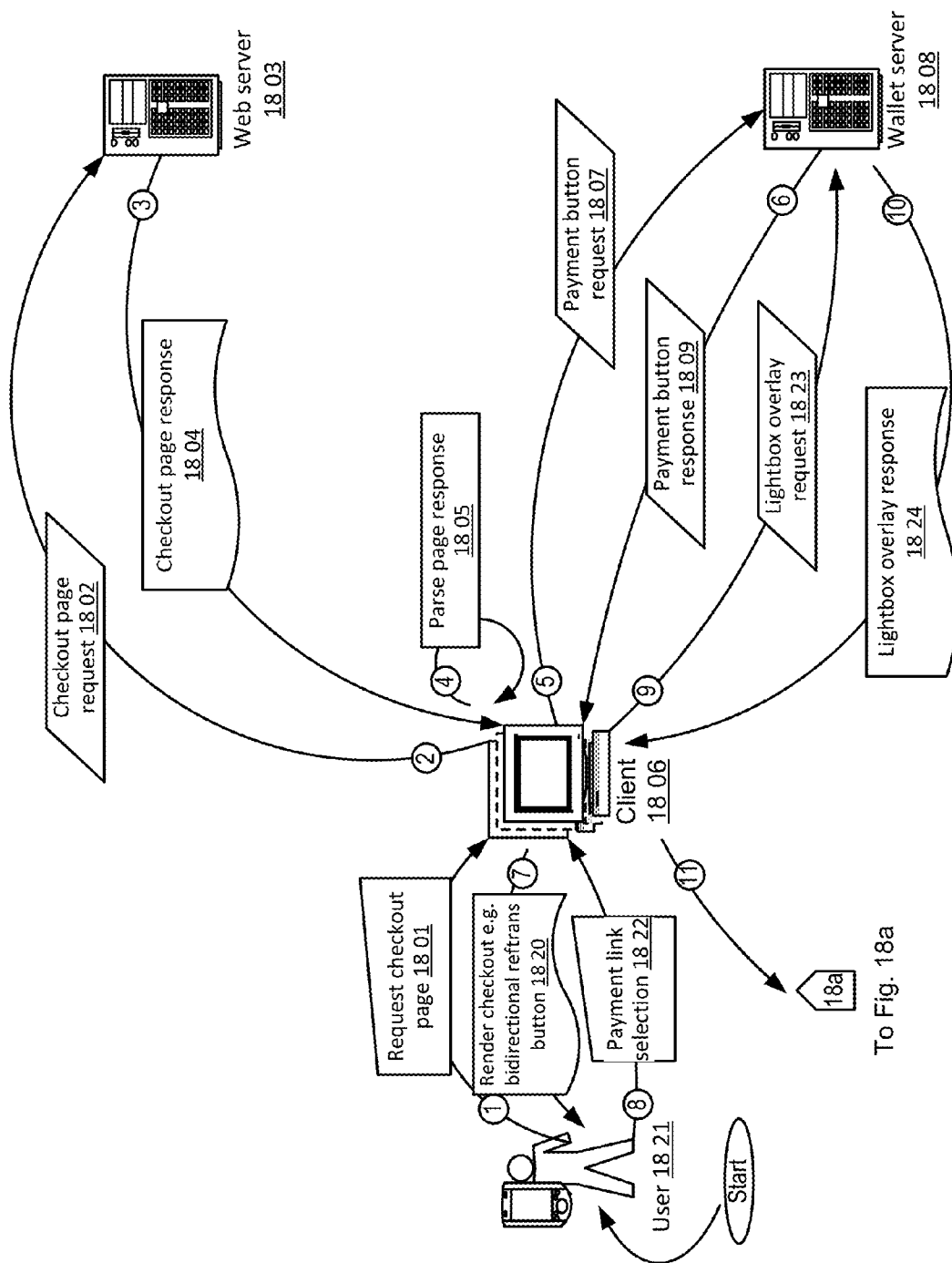


Figure 18: Example Reference Creation Datagram

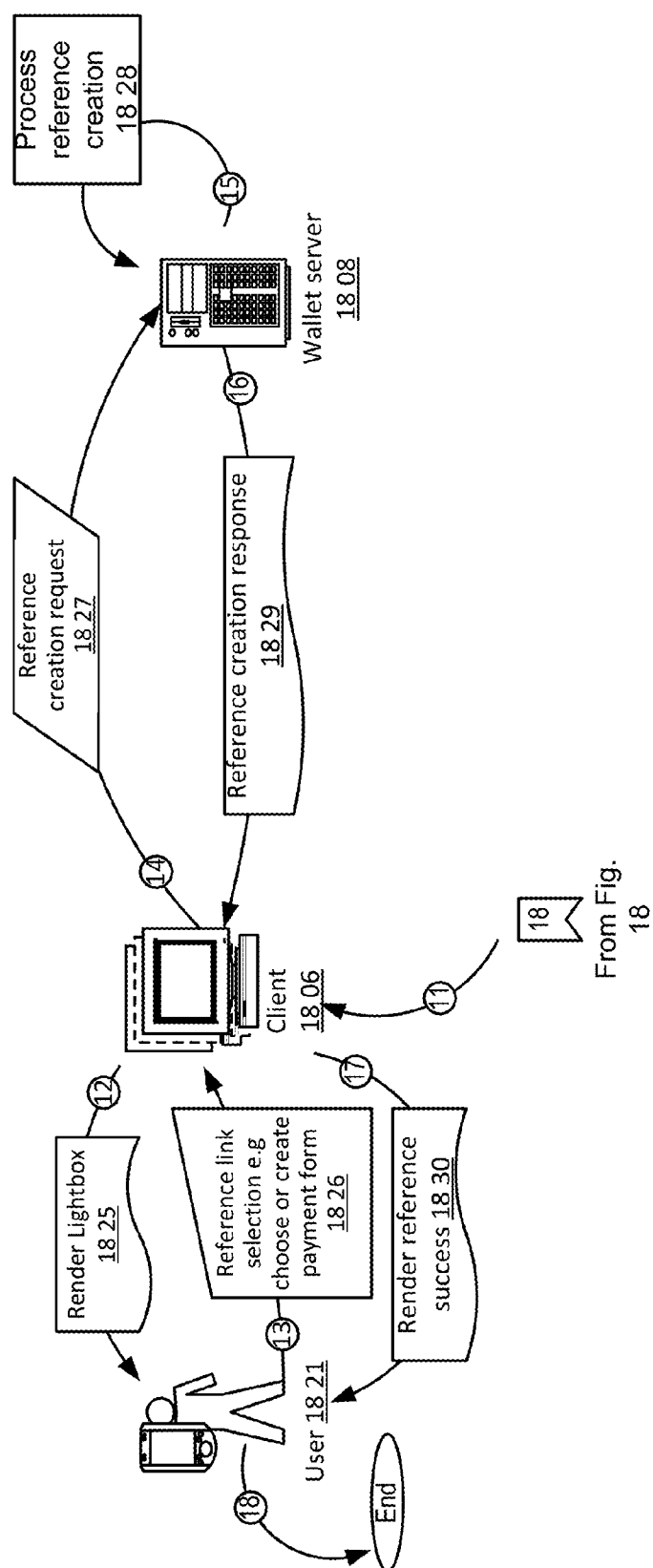


Figure 18a: Example Reference Creation Datagram

Manage Your Cards at BankIssuer

19 01

Bank Credit Card



Recent Transactions

Fix-a-car\$195.9519 04

Delta Airlines\$450.00

Balance\$645.95

19 03

☐ Click here to add this card to your v.me virtual wallet

Connect with

V

ITRP by Visa

Debit Card



Recent Transactions

McDonalds\$7.9519 04a

GroceryMart\$54.00

Balance\$61.95

19 03a

☐ Click here to add this card to your v.me virtual wallet

Connect with

V

ITRP by Visa

Figure 19: Example Issuer Side Wallet Service Enrollment

Manage Your

Bank Credit Card



Debit Card



Link Cards to your v.me Account
20 03

You have chosen to add **two cards** to your virtual wallet. Please log in to your v.me wallet account below to add these cards. If you don't have a wallet you can create a new wallet account and your **two cards** will be automatically added after setup.

Link to Existing Wallet 20 04

Username / Email:

Password:

20 05

Create New Wallet 20 07

☐ Click here to speed creation of a wallet account by pre-filling any information already on file with your bank (additional data will be shared) 20 08

☐ Force Two Factor Authentication 20 09

☐ Create a new Pre-paid card, link to wallet, and add funding. 20 10

[Create Wallet And Link Cards] 20 11

Link Cards to Wallet 20 06

Balance \$61.95

20 02 ☒ Click here to add this card to your v.me virtual wallet

Figure 20a: Example Issuer Side Wallet Service Enrollment

Associate Your Bank Issuer Cards With Your Wallet

Bank Credit Cards

Add to V.me Wallet

☒

20 12

20 13a



Card: xxxx-xxxx-xxxx-1234

Balance: \$645.95

20 14

20 15



Card: xxxx-xxxx-xxxx-5678

Balance: \$0.00

20 16

Continue >

20 15

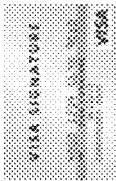
Debit Card

Add to V.me Wallet

☒

20 13b

20 13c

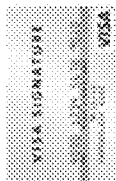


Card: xxxx-xxxx-xxxx-0246

Balance: \$61.95

20 13d

20 13e



Card: xxxx-xxxx-xxxx-1357

Balance: \$157.53

Figure 20b: Example Issuer Side Wallet Service Enrollment

Associate Your Bank Issuer Cards With Your Wallet

Bank Credit Cards

Add to V.me Wallet

☒

20 13a



20 14

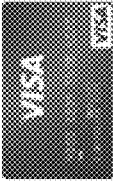
Card: xxxx-xxxx-xxxx-1234

Balance: \$645.95

Add to V.me Wallet

☐

20 12



Card: xxxx-xxxx-xxxx-5678

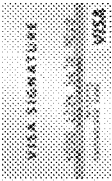
Balance: \$0.00

Debit Card

Add to V.me Wallet

☒

20 13b

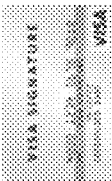


Card: xxxx-xxxx-xxxx-0246

Balance: \$61.95

Add to V.me Wallet

☒



Card: xxxx-xxxx-xxxx-1357

Balance: \$157.53

20 17

Create a V.me Wallet Account

Email Address:

Password:

Confirm Password:

20 16 | Continue >

Figure 20c: Example Issuer Side Wallet Service Enrollment

Associate Your Bank Issuer Cards With Your Wallet			<u>20 18</u>
Bank Credit Cards	Add to V.me Wallet	☒	
		☐	
Debit Card	Add to V.me Wallet	☒	
		☒	
		☒	

Send the Following Cards to Your Wallet

Please review the following selections and confirm that you would like them to be associated with your V.me Account:

	☒	Card: xxxx-xxxx-xxxx-1234 Balance: \$645.95
	☒	Card: xxxx-xxxx-xxxx-0246 Balance: \$61.95
	☒	Card: xxxx-xxxx-xxxx-1357 Balance: \$157.53

<u>20 19</u>	<u>20 20</u>	<u>20 21</u>
[Start Over]	[Complete]	

Card: xxxx-xxxx-xxxx-1357	Balance: \$157.53
---------------------------	-------------------

Figure 20d: Example Issuer Side Wallet Service Enrollment

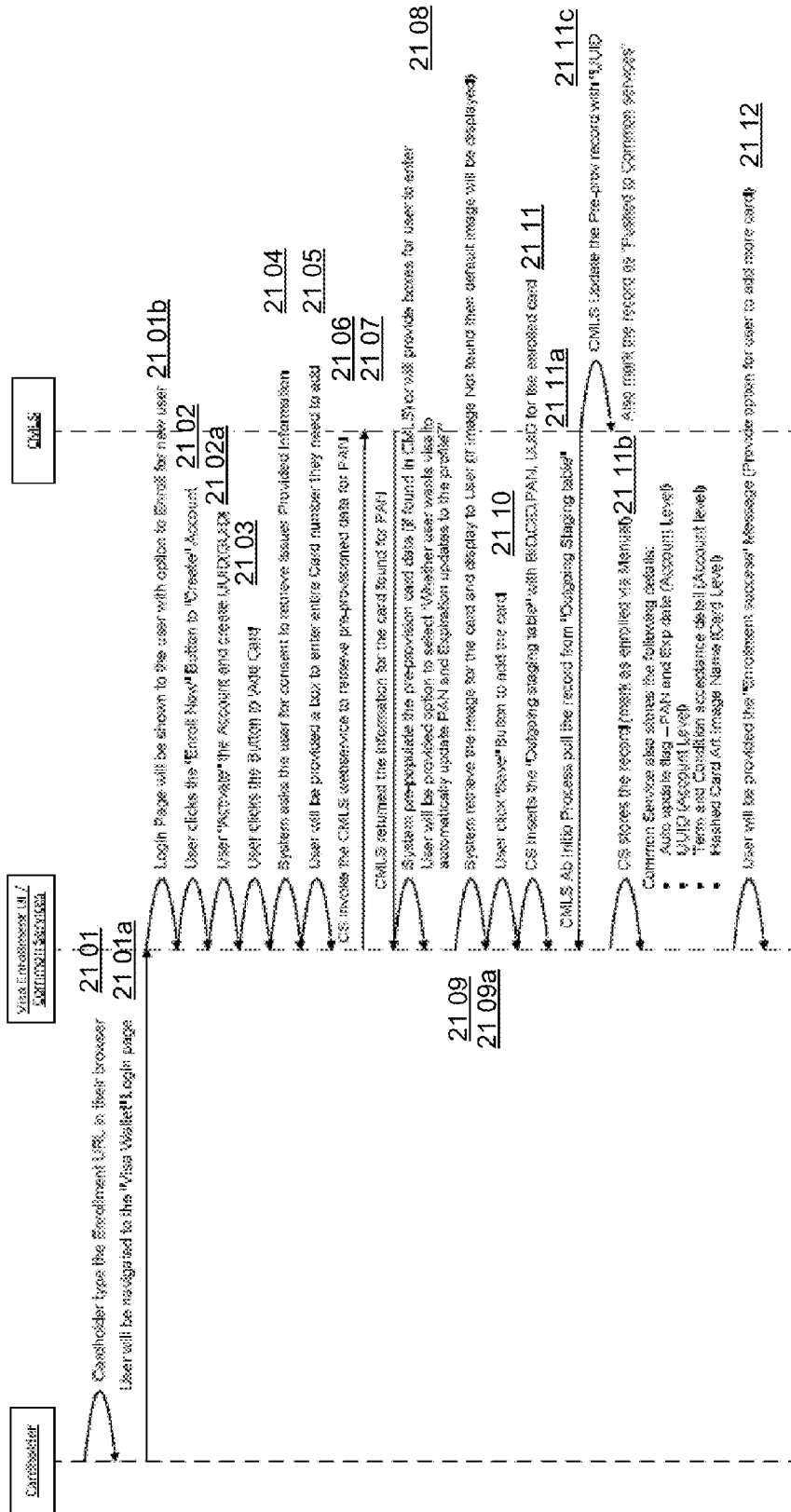


Figure 21: Example Wallet Service Enrollment (Issued)

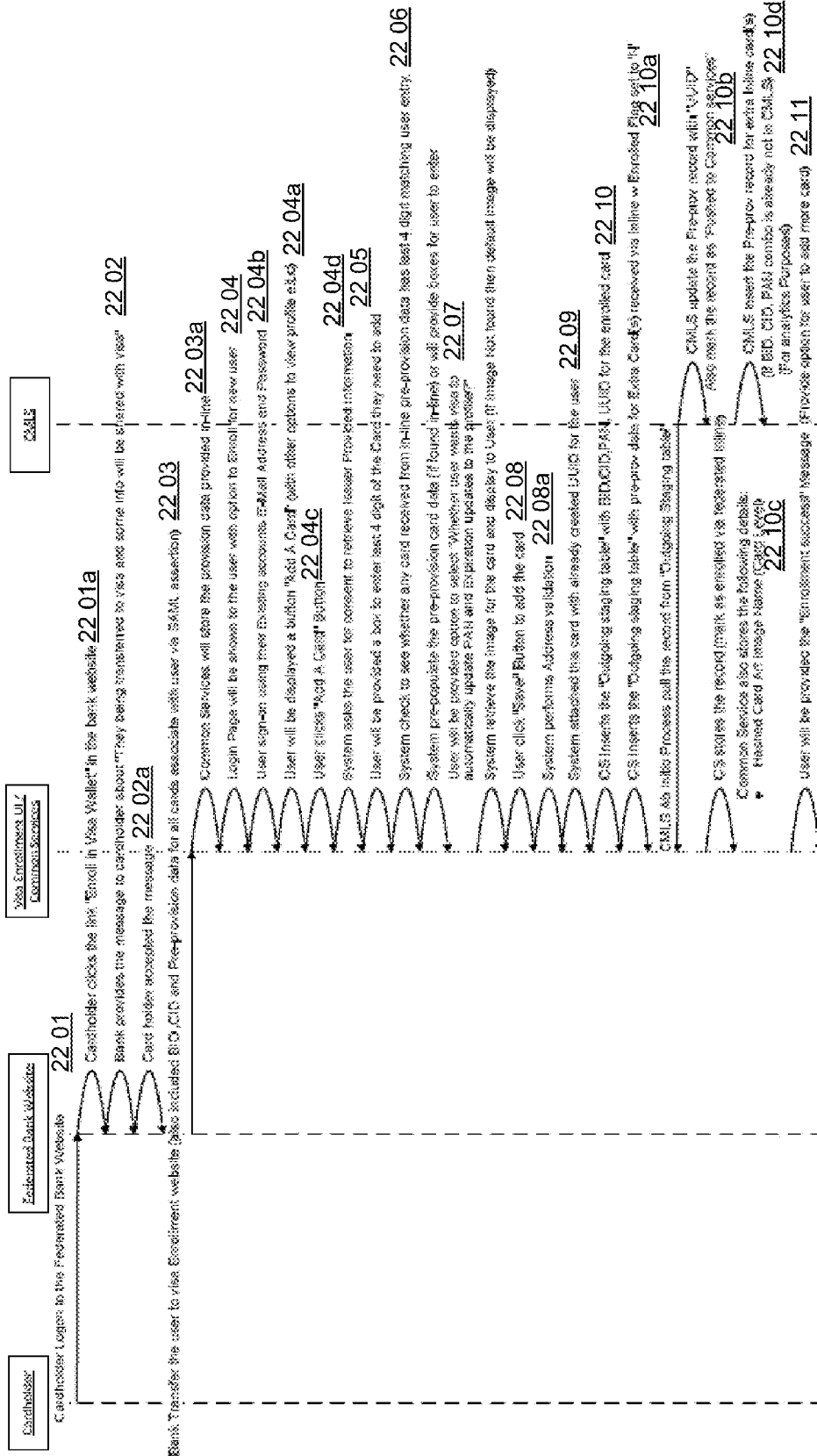


Figure 22a: Example Wallet Account Enrollment Using Issuer Data

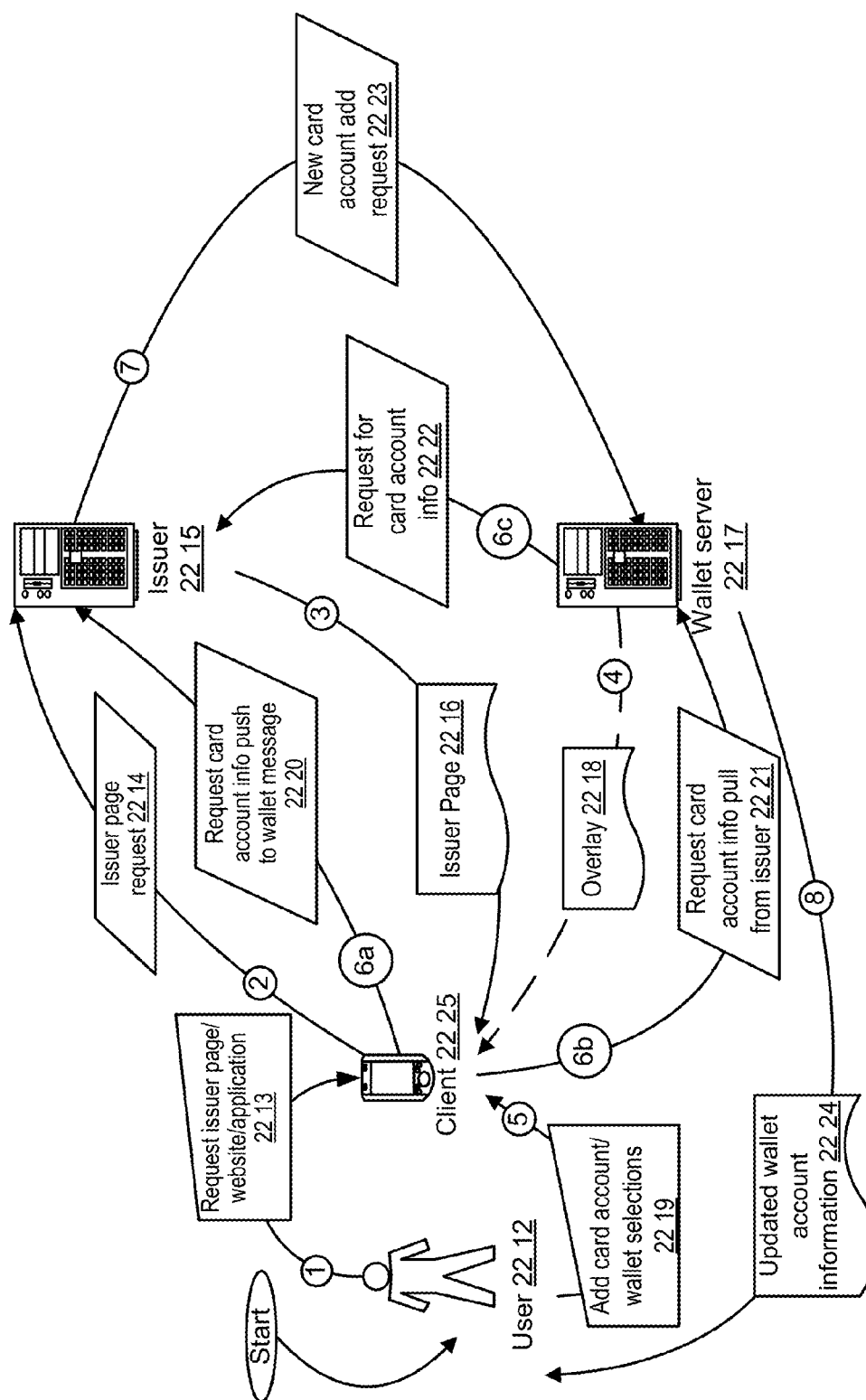


Figure 22b: Example Wallet Account Enrollment

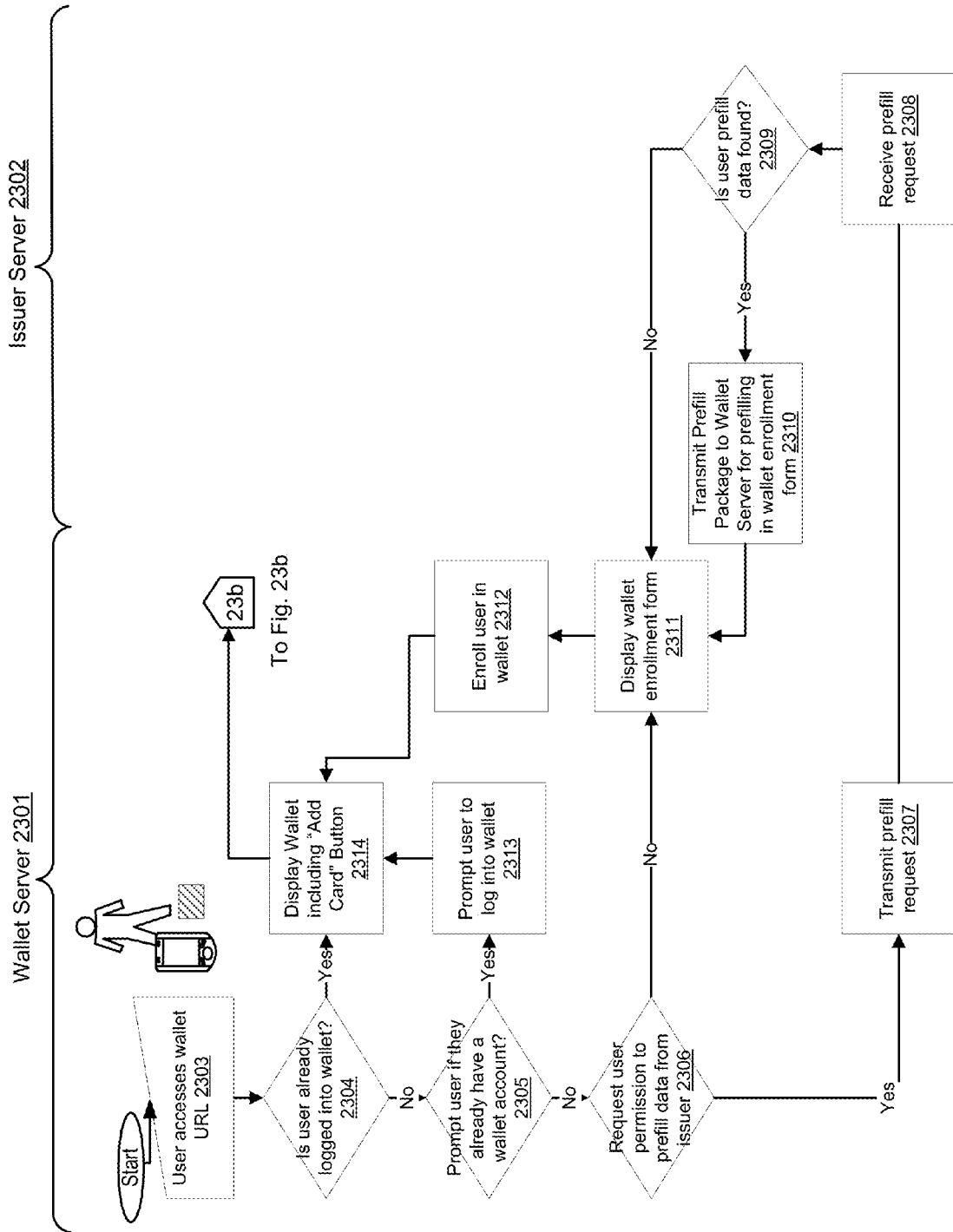


Figure 23a: Exemplary Wallet and Card Enrollment Logic Flow

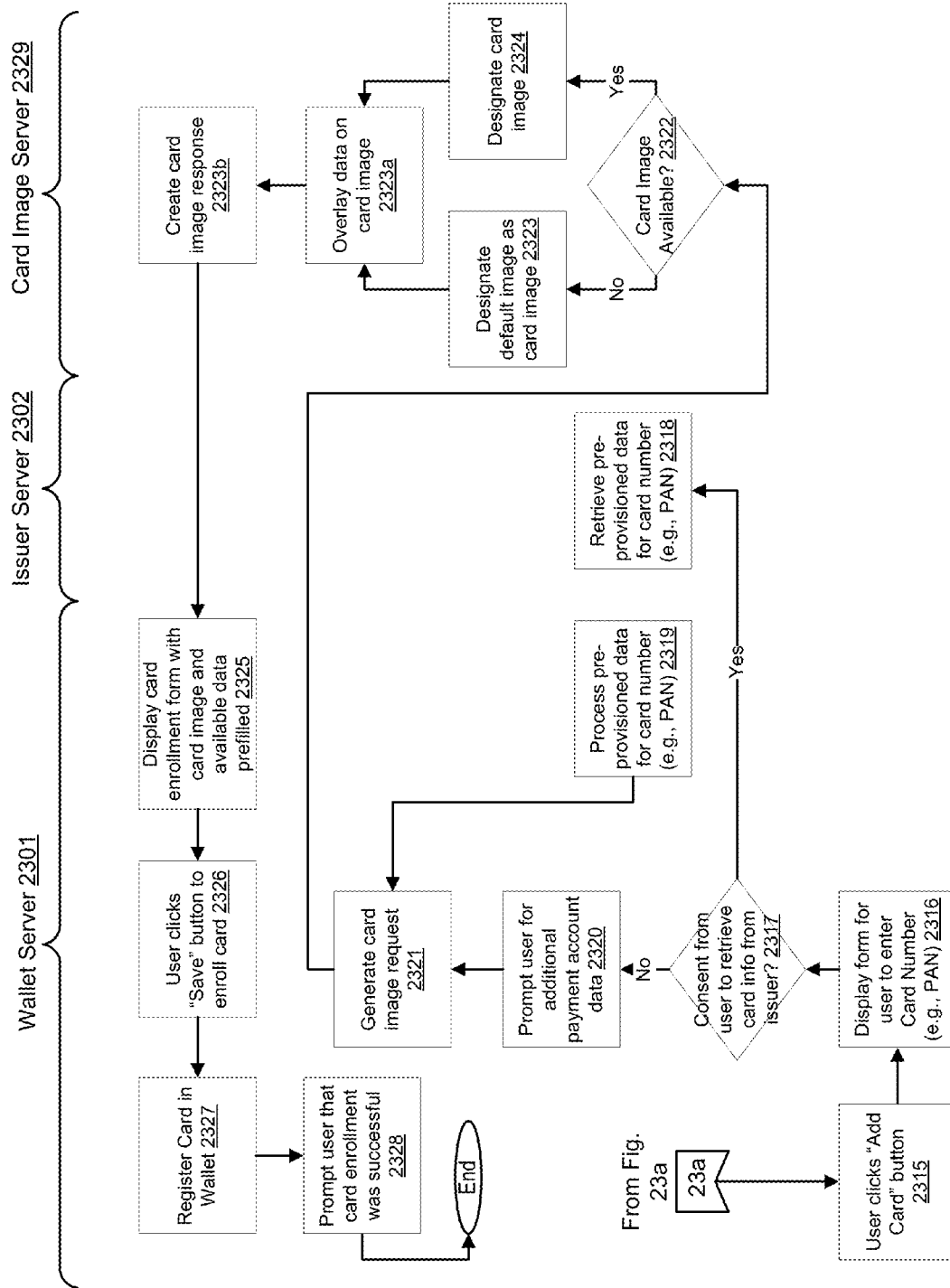


Figure 23b: Exemplary Wallet and Card Enrollment Logic Flow

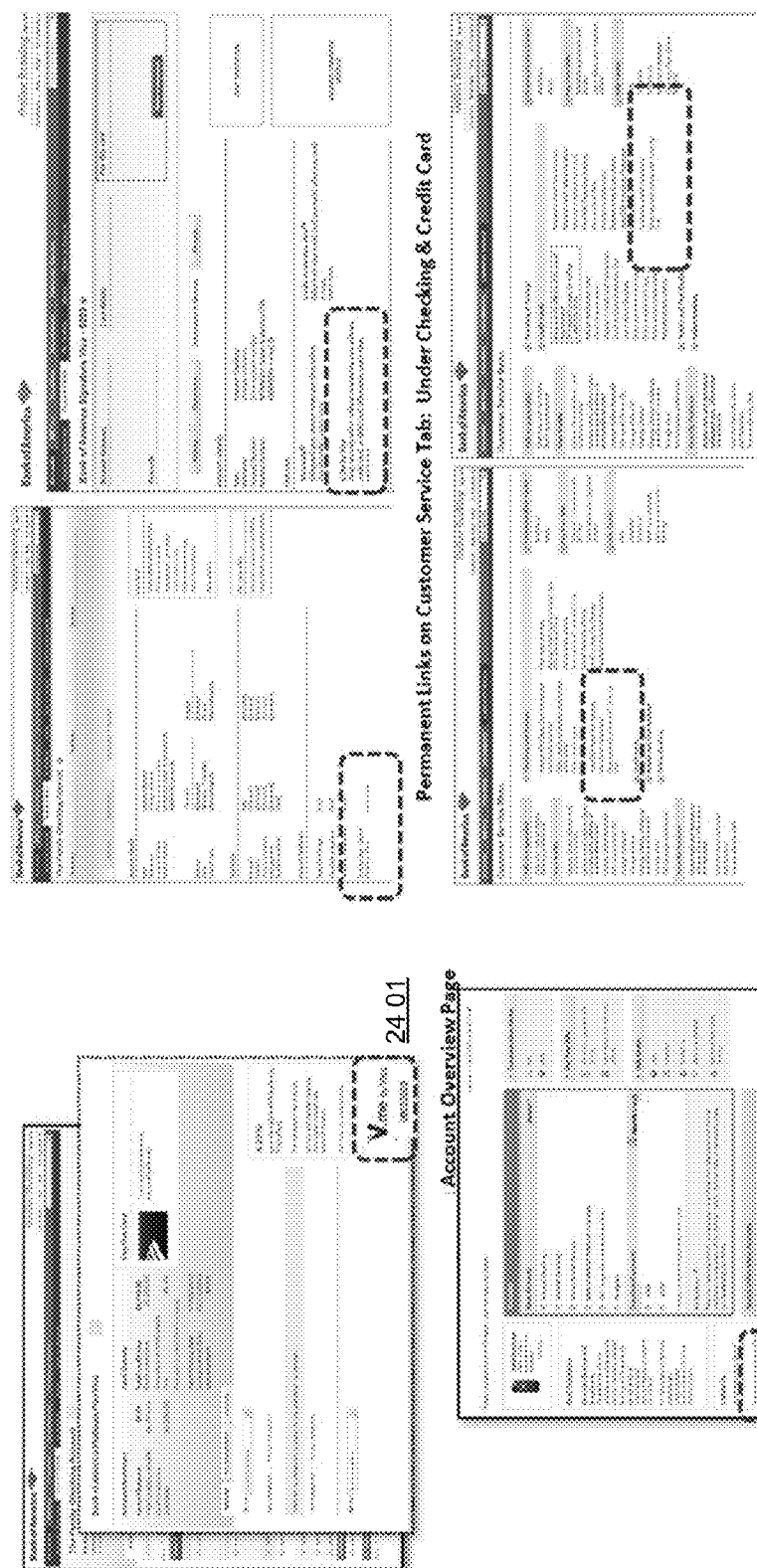


Figure 24a: Exemplary Wallet and Card Enrollment Screenshots

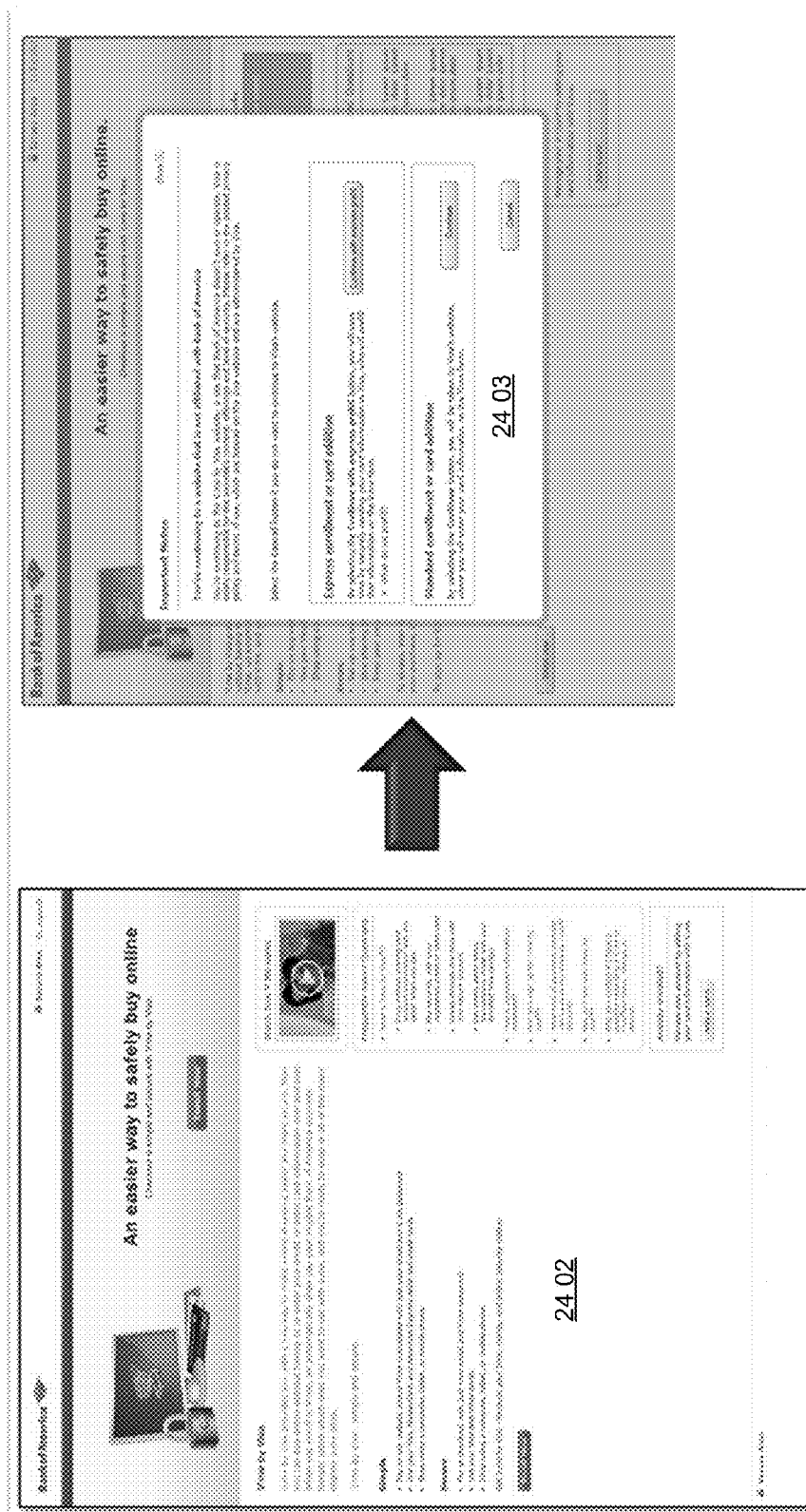


Figure 24b: Exemplary Wallet and Card Enrollment Screenshots

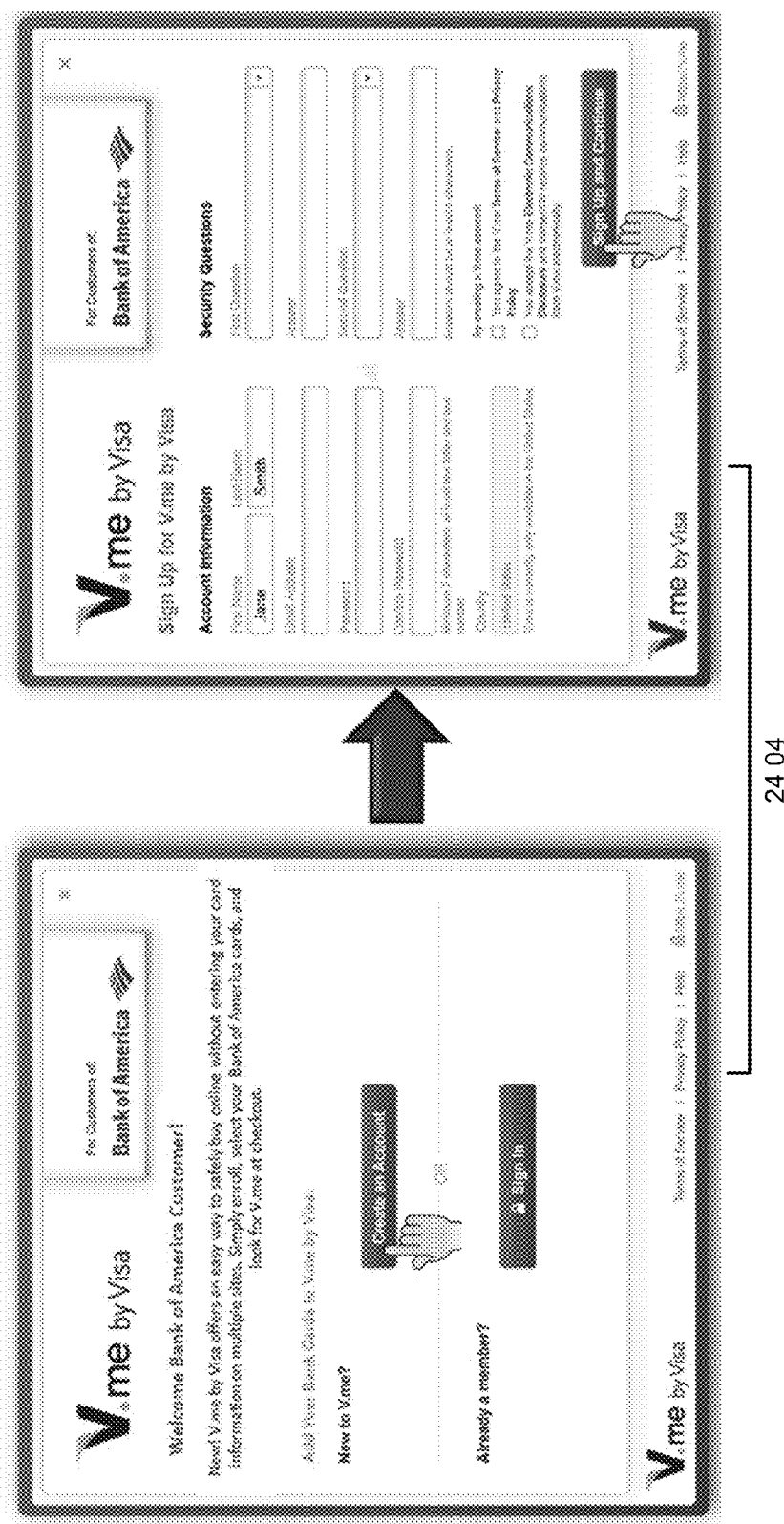


Figure 24c: Exemplary Wallet and Card Enrollment Screenshots

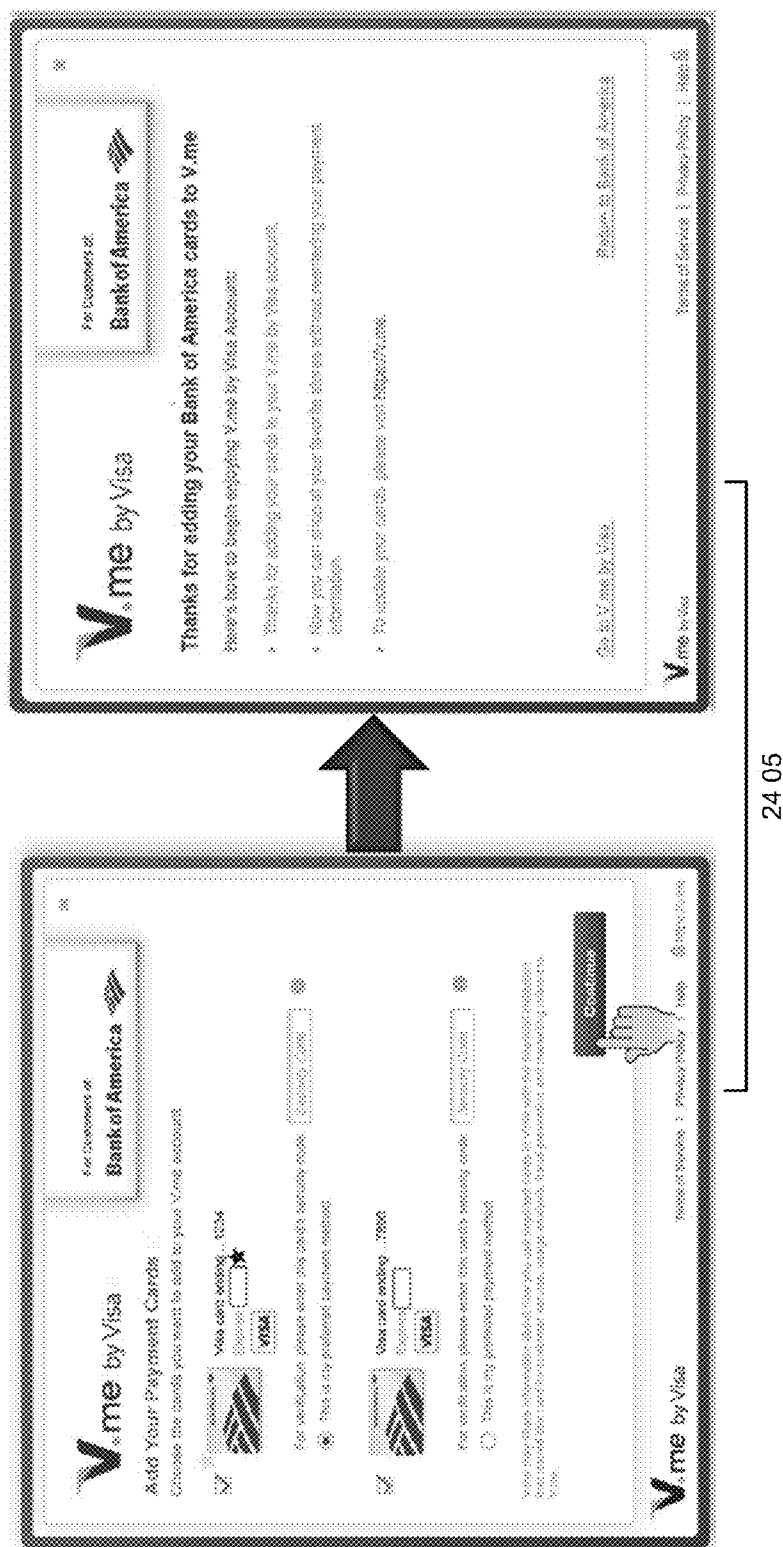


Figure 24d: Exemplary Wallet and Card Enrollment Screenshots

[illegible]

Figure 24e: Exemplary Wallet and Card Enrollment Screenshot

Account Set Up

Basic Information

Email Address:

Phone Number:

Address:

Security Questions

First Question:

Second Question:

Third Question:

First Answer:

Second Answer:

Third Answer:

Create Account

24 07

Figure 24f: Exemplary Wallet and Card Enrollment Screenshot

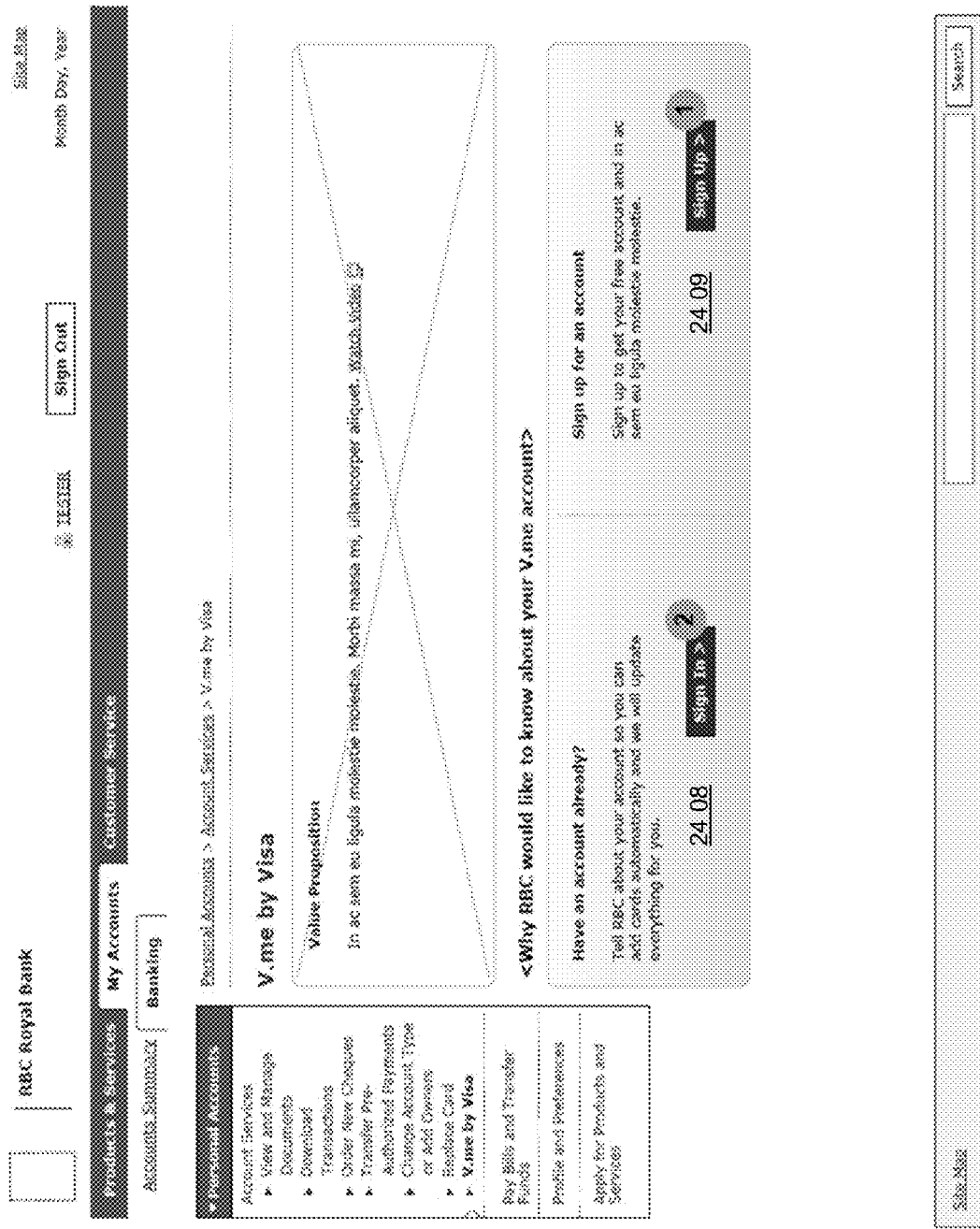


Figure 24g: Exemplary Wallet and Card Enrollment Screenshot

RBC Royal Bank

TESTER

Month Day, Year

1

Enter Information and select cards

2

Create Password and Security Questions

3

Complete process and go shopping.

1

Account Information

Name:

TESTER LASTNAME

Email:

tester@tester.com

Country:

Canada

24 11

Select RBC cards to add to account

Set as default

Select/De-select All

my RBC Visa Card

4512-1234-5678-9090

my RBC Visa Debit

4512-1234-5678-0990

my RBC MasterCard

4512-1234-5678-0890

RBC Visa Bold

4512-1234-5678-0990

Please make your card information will be sent to Visa. This product is issued by Visa.

Back

Cancel

Continue >

Test size: A | A | A

12.100

Privacy & Security | Legal | Accessibility

Figure 24h: Exemplary Wallet and Card Enrollment Screenshot

RBC Royal Bank

TESTER

Month Day, Year

1

Enter Information and select cards

2

Create Password and Security Questions

3

Complete process and go shopping

1

Account Security

Password:

Confirm Password:

Security Questions:

Card Security Code:

Select One...

Answer

Select One...

Answer

my RBC Visa Card
4512 1234 5678 9010
123

my RBC MasterCard
4512 1234 5678 9010
123

24 12

24 13

Legal Terms and Conditions

☐ By creating an account you agree to review Terms of Service and Privacy Policy

☒ Please note your card information will be sent to Visa. This product is covered by Visa.

Back

Cancel

Continue >

Figure 24i: Exemplary Wallet and Card Enrollment Screenshot

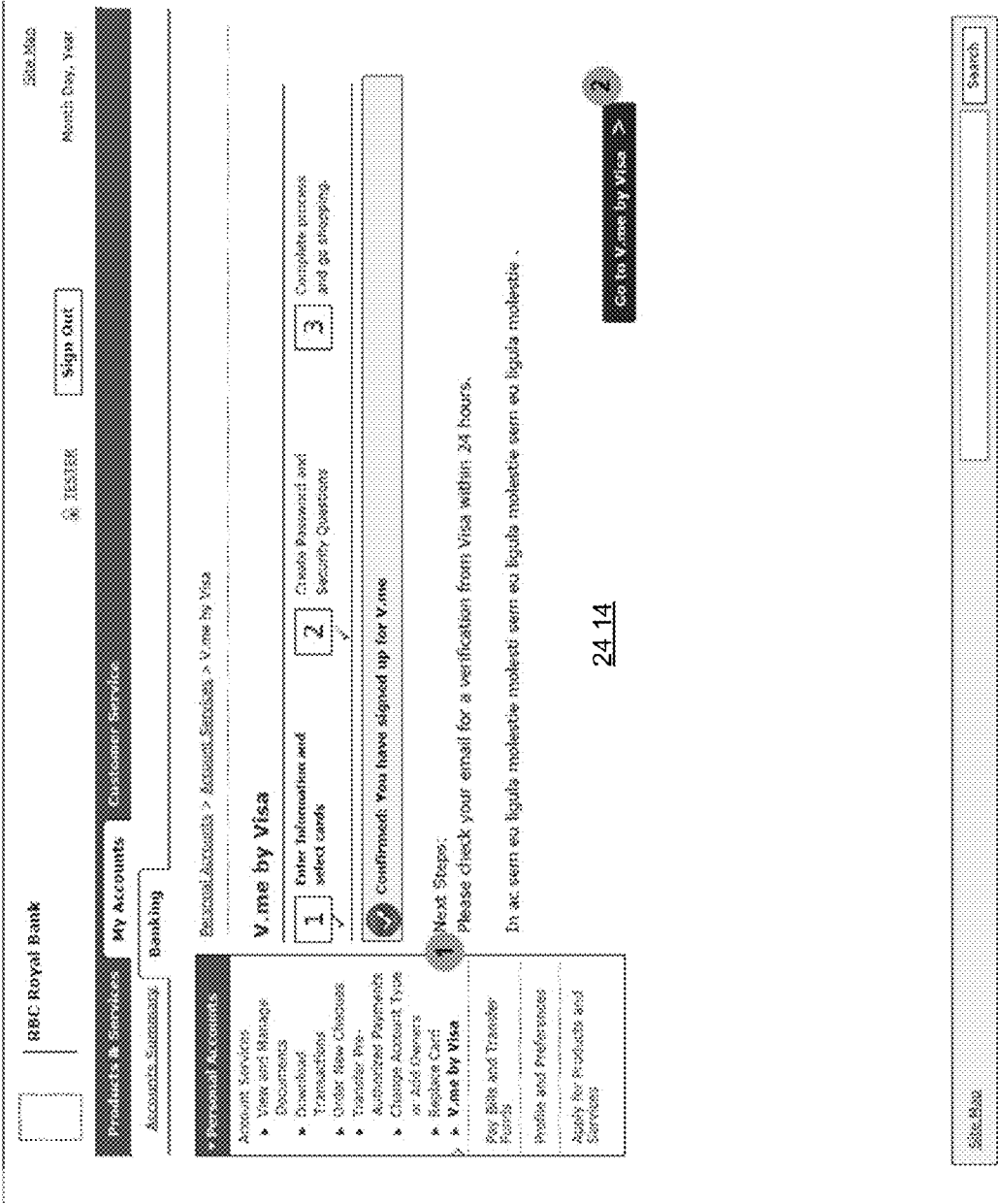


Figure 24j: Exemplary Wallet and Card Enrollment Screenshot

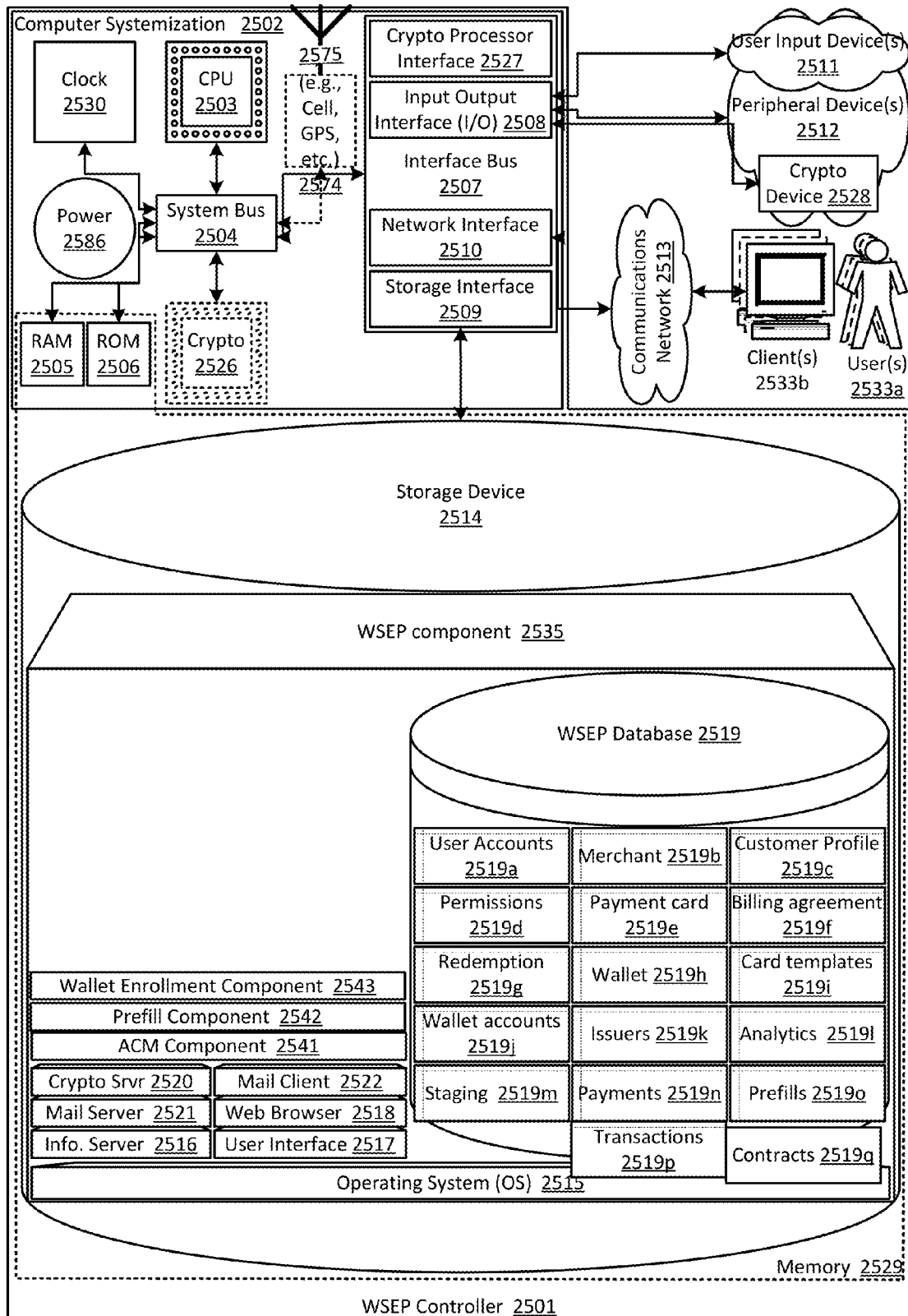


Figure 25

WALLET SERVICE ENROLLMENT PLATFORM APPARATUSES, METHODS AND SYSTEMS

PRIORITY CLAIM

[0001] This application is a continuation-in-part and claims priority under 35 U.S.C. §120 to U.S. patent application Ser. No. 13/589,053, filed Aug. 17, 2012 and entitled “WALLET SERVICE ENROLLMENT PLATFORM APPARATUSES, METHODS AND SYSTEMS,” which in turn claims priority under 35 USC §119 to: U.S. provisional patent application Ser. No. 61/525,168 filed Aug. 18, 2011, entitled “WALLET SERVICE ENROLLMENT PLATFORM APPARATUSES, METHODS AND SYSTEMS,” attorney docket no. 92US01|20270-186PV; U.S. provisional patent application Ser. No. 61/537,421 filed Sep. 21, 2011, entitled “CONSUMER WALLET ENROLLMENT APPARATUSES, METHODS AND SYSTEMS,” attorney docket no. 108US01|20270-192PV; U.S. provisional patent application Ser. No. 61/588,620 filed Jan. 19, 2012, entitled “CONSUMER WALLET ENROLLMENT APPARATUSES, METHODS AND SYSTEMS,” attorney docket no. 108US02|20270-192pv1; and U.S. provisional patent application Ser. No. 61/668,441 filed Jul. 25, 2012, entitled “REFERENCE TRANSACTION APPARATUSES, METHODS AND SYSTEMS.” The entire contents of the aforementioned applications are expressly incorporated by reference herein.

[0002] This application for letters patent disclosure document describes inventive aspects directed at various novel innovations (hereinafter “disclosure”) and contains material that is subject to copyright, mask work, and/or other intellectual property protection. The respective owners of such intellectual property have no objection to the facsimile reproduction of the disclosure by anyone as it appears in published Patent Office file/records, but otherwise reserve all rights.

FIELD

[0003] The present innovations are directed generally to digital wallets and more particularly, to WALLET SERVICE ENROLLMENT PLATFORM APPARATUSES, METHODS AND SYSTEMS or WSEP.

BACKGROUND

[0004] Consumers using the World Wide Web make purchases at electronic commerce merchants using credit cards. When consumers wish to make a purchase at a merchant web site they may provide an account number for future transactions. Accounts provided to merchants may expire.

BRIEF DESCRIPTION OF THE DRAWINGS

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

[0006] FIG. 1 shows a block diagram illustrating example service connections in some embodiments of the WSEP;

[0007] FIG. 2 shows a block diagram illustrating example WSEP architecture in some embodiments of the WSEP;

[0008] FIG. 3 shows a screen shot illustrating example account creation in some embodiments of the WSEP;

[0009] FIG. 4 shows a screen shot illustrating example merchant account login in some embodiments of the WSEP;

[0010] FIG. 5 shows a screen shot illustrating example account preference management in some embodiments of the WSEP;

[0011] FIG. 6 shows a screen shot illustrating an example cross-channel implementation of some embodiments of the WSEP;

[0012] FIGS. 7a-b show user interfaces illustrating example sign-in and account management in some embodiments of the WSEP;

[0013] FIGS. 8a-b show user interfaces illustrating example sign-in and checkout in some embodiments of the WSEP;

[0014] FIGS. 9a-b show data flow diagrams illustrating example bi-directional federation in some embodiments of the WSEP;

[0015] FIG. 10 shows a logic flow diagram illustrating an example account creation and management in some embodiments of the WSEP;

[0016] FIG. 11 shows a block diagram illustrating an example prepaid card personalization in some embodiments of the WSEP;

[0017] FIG. 12 shows a user interface illustrating an example WSEP settings control panel in some embodiments of the WSEP;

[0018] FIG. 12a is an example user interface illustrating a wallet account interface after the enrollment of new payment account(s).

[0019] FIG. 13a-i show example user interfaces in some embodiments of the WSEP.

[0020] FIG. 14 shows a screenshot illustrating an example checkout containing current transactions and a contract for future transactions.

[0021] FIG. 14a is a screenshot illustrating an example login form for accessing a WSEP account and optionally creating a virtual wallet account.

[0022] FIG. 14b is a screenshot illustrating the creation of an account with a merchant and, optionally, creating a virtual wallet account.

[0023] FIG. 15 shows a screenshot illustrating an example reference transaction enrollment.

[0024] FIG. 16 shows a user interface illustrating an example embodiment of a reference transaction management console.

[0025] FIG. 17 shows a block diagram depicting example reference failover transaction behavior.

[0026] FIGS. 18-18a shows a logic flow diagram illustrating a reference transaction link creation.

[0027] FIG. 19 shows a user interface illustrating an example embodiment of an issuer web site containing virtual wallet enrollment elements.

[0028] FIGS. 20a-d show a user interface illustrating an example embodiment of a wallet service enrollment interface.

[0029] FIG. 21 shows an example wallet service enrollment data flow.

[0030] FIG. 22a-b show example wallet account enrollment data flow.

[0031] FIGS. 23a-b show a logic flow diagram illustrating an example wallet and card enrollment logic flow.

[0032] FIGS. 24a-j show screenshot diagrams illustrating example wallet and card enrollment.

[0033] FIG. 25 shows a block diagram illustrating embodiments of a WSEP controller.

[0034] 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 FIG. 1. Reference number **201** is introduced in FIG. 2, etc.

DETAILED DESCRIPTION

[0035] Various embodiments of the WSEP may be configured to facilitate the creation of a virtual wallet account. For example, a financial institution may already have information in their records such as payment accounts, billing address, credit history reports and/or the like. By providing this information to the wallet service provider, a wallet account may be established on behalf of the user. In some embodiments, the information provided by the financial institution will be sufficient itself to enable the creation of a virtual wallet account. This would be the case where the information requirements of the virtual wallet provider are such that the financial institution is able to provide sufficient information about the user to enable creation of a wallet account. In other embodiments, the information provided by the financial institution will only partly fulfill the information requirements of the virtual wallet provider, in which case the user may be prompted for additional information before the virtual wallet is created.

[0036] Other embodiments of the WSEP enable frictionless enrollment of a consumer's payment accounts in a virtual wallet. In some embodiments, customers logged into a financial institution web site, such as an account issuer's web site, may desire to enroll payment accounts already established with that financial institution in their virtual wallet. In one embodiment, a consumer may be logged into the web site of its local bank and be able to access both a credit card and a debit card previously opened with that bank. Advantageously, the issuer bank may already have important information about the user that may facilitate the enrollment of the two payment accounts in a virtual wallet (e.g., billing address, PAN number, mother's maiden name, etc.) and/or the creation of a virtual wallet account. In one example, the consumer may indicate to the issuer that it desires for the issuer to transmit the account information the issuer has on file to a virtual wallet provider in order to pre-fill information in an enrollment form that may be used to enroll one or more payment accounts in a virtual wallet. The issuer may then share or transmit data to the wallet service provider to enable this enrollment. In one embodiment, the user will then provide additional information before the payment account is enrolled in the wallet. In other embodiments, no additional information will be provided by the user and the payment account will be automatically enrolled in the wallet after the issuer's transmission of the data. In still other embodiments, the issuer may be a merchant bank, pre-paid account provider, a non-financial institution, or an individual (i.e., a peer-to-peer enrollment facilitation).

[0037] In some embodiments of the WSEP, the creation of a virtual wallet account or the enrollment of a payment account in a virtual wallet account may be supplemented by allowing the user to create a pre-paid payment account. In doing so, the user may fund the pre-paid account immediately or open the pre-paid account with no funding. In one embodiment, the consumer desires to add an existing payment account to their virtual wallet while logged into an issuer's web site. The consumer may therefore select an established account for enrollment in the virtual wallet. Additionally, the

consumer may then also be prompted to create a pre-paid account in their virtual wallet. In some embodiments, after choosing to create a pre-paid account, the consumer may then choose an account with a financial institution from which to fund their pre-paid account. Advantageously, in this example, the consumer may also desire for the information about the pre-paid funding source account to be shared with the virtual wallet provider to enable the wallet provider to simultaneously create and fund a pre-paid account. In other embodiments, the WSEP may allow a wallet service provider to retain information (e.g., account number, routing number, billing address, and/or the like) to enable future funding of the pre-paid account to occur without additional sharing of data from financial institution to wallet service provider. In still other embodiments, the consumer may create a funding threshold rule that would indicate to the wallet service provider to re-fill or top-up the pre-paid account from a designated funding source on the occurrence of a certain event, such as low funds. In doing so, the WSEP enables a consumer to create a pre-paid account seamlessly while enrolling other payment accounts in the virtual wallet.

[0038] In other embodiments of the WSEP, the creation of the pre-paid account may happen independently of a consumer's interaction with a third-party financial institution. For example, in some embodiments the virtual wallet may be accessed through a mobile application. In this embodiment, the wallet application on the user's mobile phone may prompt the user to establish a pre-paid account when it detects that the consumer has just received a large credit to one of their financial accounts. In doing so, the establishment of pre-paid accounts may be encouraged and facilitated by the WSEP.

[0039] In some embodiments of the WSEP, the virtual wallet account enrollment facility may be configured to automatically retrieve an image of the payment account being enrolled in the virtual wallet. In doing so, consumers may be presented with an image of the card representation of the payment account being enrolled. In some embodiments, this image may be used by the consumer to verify the authenticity of the payment account being added. In other embodiments, the image will be displayed to facilitate the selection of payment accounts for addition to the virtual wallet.

[0040] Various embodiments of the WSEP facilitate the creation of persistent and re-assignable links between the consumer's virtual wallet and a merchant or other entity. In some embodiments, the WSEP may allow the customer to link their virtual wallet to a merchant using reference aliases that are not permanently linked to a single payment account or method. In doing so, a consumer's accounts may change over time without breaking the persistent reference links that have been created to various merchants. This capability may facilitate a low friction user experience for payment transactions. In some embodiments, the consumer may designate a reference for an account using a merchant's web site. In doing so, the consumer may agree to allow future transactions to occur without requiring future affirmative consent. The consumer may then manage the reference connection through a virtual wallet or web site and update the reference aliases without requiring another visit to the merchant's web site.

[0041] Alternative embodiments of the WSEP may also allow the consumer to create reference links between other information in their virtual wallet. For example, a consumer may desire to create a reference alias for an address frequently used in commerce transactions. Alternatively, the consumer may wish to create a reference alias to a name or persona that

they may use in commerce. In doing so, the WSEP may enable the consumer to maintain a degree of privacy while still enabling low friction commerce transactions.

[0042] In some embodiments of the WSEP, the consumer may agree to or designate certain payment options to be used in recurrent transactions. For example, the consumer may permit flexible recurring commerce, wherein future transactions from a merchant may be billed to the reference alias without further intervention from the user. In other embodiments, the consumer may permit managed subscription commerce wherein the consumer and/or merchant agrees to various terms or conditions that will govern the current and/or future reference transactions with the consumer's virtual wallet account. For example, the consumer may designate a pre-set amount which the merchant may bill through the reference link monthly. For example, a consumer may enroll in a "Jam of the Month" club. In one embodiment, the consumer may choose to create a reference transaction authorization of \$40.00 per month for 3 varieties of jam. In another embodiment, the jams may have variable prices (such as a rare Jam for \$199.00) and the consumer may authorize full payment or partial payment with the remainder billed later through a reference transaction or alternative mechanism. Alternatively, the consumer may agree to allow the merchant to bill a capped total amount to their virtual wallet reference account before requiring affirmative consent from the consumer for future transactions. For example, the user may authorize a one year "Jam of the Month" subscription for \$199.99 which will prompt the user in one year to optionally renew the subscription.

[0043] In some embodiments, the WSEP may provide payment security features to the merchant. For example, the merchant may be given assurances that at least one payment account will be available for a given period of time using a reference link. Alternatively, the merchant may be alerted when a reference link is updated or revoked by a consumer.

[0044] In some embodiments, the WSEP can enable the payment account issuer to update various parts of a reference transaction link without the intervention of the consumer. For example, if a consumer's card number is compromised as a result of fraud, the payment account issuer can automatically issue a new account number and update any references using that payment account. Additionally, a payment account issuer may change a consumer's account type (i.e. from 'Gold' to 'Platinum') and associate the updated account type with the reference transaction link. Advantageously, these capabilities may enable higher transaction clearance rates for consumers, merchants and payment account issuers.

[0045] In some embodiments, the WSEP may provide enhanced security features to the consumer. For example, the consumer may be given additional options for restricting reference transactions if the merchant is a new merchant, located in a foreign country, has a history of fraudulent transactions, or other conditions are present that may be cause for enhanced security. In alternative embodiments, the consumer may receive alerts when a transaction is posted through a reference link. For example, the consumer may be alerted after every transaction, or only if the transaction is suspicious. In some embodiments, the consumer may be given the option to approve or cancel the reference transaction.

[0046] In some embodiments, the WSEP may provide a control panel through which the consumer may manage the reference account links. For example, the consumer may desire to remove a payment account from their virtual wallet

and re-assign any reference connections previously using that payment account to instead use another payment account. In other embodiments, a consumer may desire to simultaneously add a new payment account to their virtual wallet and use the newly added account to replace another account in their virtual wallet. In some embodiments, when a consumer deletes a payment account from their wallet they may be prompted to update any reference transaction links that use that reference payment account. In doing so, the consumer can provide uninterrupted linkage to payment references. In still other embodiments, the consumer may be permitted to view reports regarding their historical usage of a reference alias or any accounts linked thereto. In some embodiments, the consumer can update, edit, or revoke links between reference account aliases and various merchants.

[0047] Various embodiments of the WSEP may enable the consumer to create rules governing the administration and use of reference aliases. As such, the consumer may be able to designate a hierarchy of payment accounts to be used for one reference alias in the event that some payment accounts are not available. In other embodiments, the consumer may be able to designate alternative reference payment methods such as frequent flyer accounts, merchant points accounts, coupons, virtual currencies, government benefits, future paychecks, accounts receivable, loans or lines of credit.

[0048] In some embodiments, the WSEP may enable a merchant offering a checkout option to display a button on their web page including enhanced information. For example, the button may include text indicating that the transaction will be fulfilled using a reference alias in the consumer's virtual wallet. Alternatively, the button may display a reference address that the consumer has previously designated for use in such transactions. In some embodiments the consumer may interact with the button directly to change, update or view reference transaction information.

[0049] Various embodiments of the WSEP facilitate a common, low friction user experience for consumers wishing to link a financial account, a merchant account, or any other participating commerce services to a digital wallet. In some embodiments, the WSEP provides a standardized common user experience and control panel for allowing customers to view, grant and manage permissions for financial institutions, merchants or participating commerce-related services to interact with their digital wallet. In other embodiments, the WSEP eliminates the need for consumers to remember and maintain multiple authentication passwords across many merchant, commerce and payment domains. In yet other embodiments, the WSEP maintains an up-to-date payment and other relevant personal data across multiple merchants and commerce-related services. Various WSEP embodiments may also solve for a usability friction for both merchants and consumers of having to authenticate twice, once to a merchant and once to wallet provider in order to conduct a wallet ecommerce transaction. Using WSEP, consumers may log in once either via the merchant or the wallet and conduct an ecommerce transaction.

[0050] Embodiments of the WSEP may also facilitate storage and management of customer identity and other relevant information for merchants and other commerce related services. Some WSEP embodiments may provide a faster and low friction new customer enrollment for customers who already have a wallet account. Other WSEP embodiments may provide consumers their own centralized cloud-based account having a master copy of commerce-related personal

and account information protected by a trusted brand. Some WSEP embodiments may provide issuers branding and/or communication opportunities with cardholders even in shopping experiences like card-on-file purchases.

[0051] Some embodiments of the WSEP may provide consumers facilities for easily and conveniently personalizing new prepaid accounts with their issuer using previously verified personal information stored in an online wallet, and expediting provisioning a prepaid account to a digital wallet. Once a prepaid card is connected with the wallet, the WSEP provides the consumer an easy to remember authentication tool to sign on to view and manage their prepaid account either at the wallet destination website/application or through limited federation to the prepaid issuers online (or mobile) prepaid service application.

[0052] These and other embodiments of the WSEP provide a secure and trusted bidirectional federation with a digital wallet by instituting a permissions system that allows services certain access privileges (e.g., read, write, transact, etc.) to the wallet only when appropriate and subject to both systematic and customer-managed controls.

WSEP

[0053] FIG. 1 shows a block diagram illustrating example service connections in some embodiments of the WSEP. In one implementation, the WSEP button **102** may be an OAuth based button that allows users to sign in and connect their wallet profile **115a** in the wallet **115** with their accounts at issuers **105a**, **105b** and merchants **110a**, **110b**, **110c**. Once connected, a bidirectional link may be established between the services (e.g., issuers, merchants, etc.; hereinafter “merchant”) and the wallet with ongoing permissions explicitly agreed to by the user. In some implementations, the bi-directional link may facilitate, for example, updating of card information (e.g., expire date, new identifier, increased spending limit, and/or the like) from the issuer to the corresponding card slot in the wallet, and from the wallet to the merchant. Similarly, in some other implementations, change in customer information initiated by the customer from a merchant interface may flow to the wallet and from the wallet to the issuer, for example. In some embodiments, WSEP may facilitate addition of an account or payment method and personal data to the wallet from an issuer website or an application, set up of default payment method and sharing of relevant info (e.g. contact and shipping information) with a merchant for an ongoing billing relationship, set up of one-way identity federation with a merchant to enable a customer to log in to a merchant through the wallet, real-time API calls for merchants to be able to display rich information about payment methods linked to customer relationship, and/or the like. In one implementation, for example, if a user provides a retailer Nordstrom with their nicknamed “personal card” and “business card”, Nordstrom would be able to display those nicknames and a thumbnail of the issuer card-art (if provided by a connected issuer). Similarly the user could provide Nordstrom with their wallet nicknamed “home shipping address” and “work shipping address”. If later on through the customer wallet application or portal, the customer updates their address or makes changes to their card nickname etc., those changes would be immediately reflected next time the customer visits Nordstrom because those accounts are connected by the WSEP. In some implementations, the same frame work

may facilitate any sort of customer-initiated unidirectional or bidirectional connection between the wallet and an outside service.

[0054] In some embodiments, various service providers may leverage the WSEP to provide a variety of services. For example, an issuer connected to the wallet may provision card accounts to a wallet, dynamically update account status, card art, and/or the like, provide real-time balance data, publish targeted offers to customers, publish and update issuer “apps” or gadgets to the customer’s wallet, and/or the like. A merchant connected to the wallet may allow customers to quickly link existing merchant accounts to a wallet account, allow customers to quickly create a merchant account by drawing information (with customer’s permission) from the customer’s wallet account, allow merchants to set up open authorization, recurring billing, subscription billing relationships with the customer, keep records up to date and access current information on file for their connected customers, show customers an inline display of current accounts (e.g. including card art) for accounts linked to their merchant relationship, allow returning customer to login to their merchant account with through wallet login widget, and/or the like. A loyalty provider connected to a wallet may add a loyalty account to a wallet, provide real-time points/currency balance, publish targeted rewards offers, access a loyalty account through a wallet login, and/or the like. A transit authority connected to a wallet may load or associate transit passes with the wallet, allow returning customer to login to their transit account or purse through the wallet login widget, allow redemption of transit passes or tickets from the wallet, and/or the like.

[0055] FIG. 2 shows a block diagram illustrating example WSEP architecture in some embodiments of the WSEP. In some embodiments, the WSEP architecture may be a cross-channel and cross-entity framework comprising widget-based authentication and permission management between various commerce solution components and the wallet. In one implementation, for example, various approved commerce services **202**, issuers **204**, merchants **206**, and/or the like may have embedded a WSEP button (e.g., **208a**, **208b**) in their native applications or sites. When the button is invoked on the web or on a mobile device, the button may trigger a WSEP widget (e.g., **210**, **215**) to either connect a new service (e.g., **202**, **204**, **206**) to the wallet or authenticate the user. A user may input username and password credentials into the wallet widget (e.g., **210**) to get authenticated. The user may have control (e.g., create, view, manage, cancel, etc.) over the individual relationships and may configure permissions for each service they connect to. In one embodiment, the WSEP may allow approved services, issuers and merchants permissions to obtain various information relating to the user and wallet such as consumer profile **225**, billing agreement **230**, redemption **235**, loyalty and rewards **240**, coupons/offers **245**, wish lists and stored items **250**, merchant applications/widgets **255**, Value Added Resellers (VAR)/Software-as-a-service (SaaS) commerce wallet plug-ins **260**, analytics **265**, account or points balance information **270**, payments **275**, and/or the like. In one implementation for example, the WSEP may manage which services can connect to the wallet. In a further implementation, the WSEP may pass along information from an approved and connected service such as a loyalty program (e.g., Star Woods Points program) to a merchant such that the merchant may provide the customer a special deal, offer or an opportunity to use or exchange points/currency when transacting. In one implementation, approved commerce services, issuers and merchants may be able to push information relating to any of the above to the wallet.

[0056] FIG. 3 shows a screen shot illustrating example account creation in some embodiments of the WSEP. In one embodiment, the WSEP may facilitate acceleration of an account creation with a merchant by drawing customer data such as name, addresses, email, etc., from the wallet. Once connected, the wallet may keep the customer data up to date and provide an easy way for the customer to sign in to the merchant account. For example, as shown in FIG. 3, a new customer may create an account with a merchant (e.g., Nordstrom) by filling out the form fields **305** (e.g., first name, last name, email, password, zip/postal code, gender, email preference, and/or the like). In one implementation, all of these fields may be replaced with information from the WSEP and persistently linked to the customer's wallet profile when the customer opts to create an account via the WSEP facilities of the wallet (e.g., V.me wallet). The data entry **205** for creating an account with the wallet is much less with the WSEP.

[0057] In some embodiments, the initial connection between an entity and V.me creates a customer identifier unique to that relationship. Unlike storing card information with a merchant, which, if compromised, could be used at any merchant, the customer identifier can only be used by the designated entity. Any other entity attempting to use another entity's identifier to access a customer's wallet account would be denied. In some implementations, the merchant may use this unique identifier to make calls to the wallet to retrieve and/or update commerce-relevant or other customer data. The customer has the option to maintain, in one place, address book, payment methods, and payment preferences. If the customer moves addresses for example, or obtains a new payment card, these changes may be remotely propagated to all the merchants they do ongoing business with. In some implementations, the merchant has a set of callbacks that the merchant can invoke to the wallet in order to offer seamless and uninterrupted service to the customer. Under the appropriate permissions, the merchant may make these calls independently and/or under certain triggers such as the appearance of the customer starting a new shopping session.

[0058] FIG. 4 shows a screen shot illustrating example merchant account login in some embodiments of the WSEP. The WSEP in some embodiments may facilitate expedited merchant sign in, where customers can skip merchant's login and password **405** with the click of the WSEP button **405a**. The one click WSEP check-in means customers log in with less friction and do not have to type, remember or forget and have to retrieve merchant passwords. The WSEP may return the merchant's customer ID (or contract ID) to the merchant, and facilitate the customer login to the merchant account.

[0059] FIG. 5 shows a screen shot illustrating example account preference management in some embodiments of the WSEP. The WSEP, in some embodiments, may maintain dynamic linkage and branding for issuers, merchants and the wallet whether or not a lightbox (i.e., a payment widget) is used for every purchase flow. For example, in a merchant site **505**, under the customer account **510**, information relating to order status **515**, account profile **520**, address book **525**, payment methods **530**, and/or the like may be displayed. The merchant may have their own set of customer information (e.g., order information or size information) that they maintain in their customer database. However, other information such as primary shipping address and payment methods may be dynamically linked and synced to WSEP such that the merchant has access to the customer's preferred shipping address and payment methods. For example, address book

525 may display the default shipping address and the payment methods **530** may display a list of payment methods that are stored with the merchant for faster checkout. Using callbacks, the WSEP may obtain not only payment methods and addresses, but also loyalty accounts, payment authorizations, entitlements, payment preferences, and/or the like.

[0060] In one implementation, each callback may include the customer ID that is unique to the customer-merchant relationship. In a further implementation, API calls to the WSEP may include one or more API keys such as a public key and/or a shared secret key. An API key may be a string value that identifies the general API access configuration and settings for the site. In some embodiments, callbacks for WSEP may include, without limitation, the following:

TABLE 1

Example Callbacks

Get Payment methods (returns card nicknames, brand and last 4 digits)
Get addresses (returns full addresses that customer has shared with merchant, address nickname, and indicator for default/primary address)
Get Loyalty accounts (returns active loyalty programs that customer has shared with merchant, program names and indicator for current default/primary loyalty program)
Make Payment authorizations (request to instantiate a purchase against the customer ID)
Get/Add Entitlements (retrieve and redeem previous purchase records for the customer, e.g. tickets, passes, pre-paid purchases, subscription codes, or other product codes defined by the merchant)
Get Payment preferences (e.g. receipting preferences and preferred shipping carriers)

[0061] Various methods of callbacks may be utilized. In some embodiments of the WSEP, API and inline widget methods, among others, may be implemented. Using the API method, the merchant server may make API calls to the V-Connect server to retrieve customer data. For example, a customer may log in to a merchant account to view their account preferences with the merchant. The merchant server may execute an API call to get payment methods from the WSEP server. The merchant may then display the currently active payment method is a wallet (e.g., V.me wallet) with account nickname and ending in digits xxxx. For example, referring to FIG. 5, the merchant may obtain payment methods **530a** and **530b** from WSEP and display them using their nicknames such as "My Business Credit Card Visa Card Ending . . . **1234**" (e.g., **530a**) and "My Personal Debit Card Visa Card Ending . . . **1234**" (e.g., **530b**). In this way, via API calls, the merchant may display rich, up to date account information including card art.

[0062] Using the inline widget method, the merchant may display a wallet rendered "window" into a user's wallet account. Inline widgets may display rendered or interactive elements that are injected into the merchant's website. An example would be a widget that displays the nickname and associated card-art for payment methods stored on file with a merchant, similar to **530a**, **530b** shown in FIG. 5. A JAVASCRIPT call from the merchant may indicate the type, parameters, and customer ID for rendering the widget.

[0063] Referring again to FIG. 5, a customer may also edit payment methods and other information in the wallet via the WSEP button **535**. Using the edit option, the customer may add, modify, delete, link/delink accounts and addresses, and, at a glance, confirm any new card they added to their wallet account last week is active with the merchant and their bill will process correctly.

[0064] FIG. 6 shows a screen shot illustrating an example cross-channel implementation of some embodiments of the WSEP. In one implementation, the WSEP button may be embedded in various channels including, for example, web sites, mobile devices, tablets, smart phones, web applications, mobile device application, and/or the like, as long as partners using the channels are enrolled in the WSEP, and in some implementations, agreed to access control restrictions. Referring to FIG. 6, a WSEP button 610 is placed in a tablet channel 605. Invoking the WSEP button may trigger a wallet widget to either authenticate the customer or authenticate and connect the service, merchant, or application to the customer's wallet account. In some embodiments, the WSEP button may be implemented in other channels and physical world scenarios such as point of sale interactions. For example, using a physical card swipe or chin/pin interaction may trigger a wallet account connection or login. As another example, using a quick response (QR) code scan, a near-field communication (NFC) tap or other mobile trigger in lieu of a WSEP button may also trigger a wallet connection or login. As yet another example, using a voice password, repeatable gestures or action, biometrics, and/or the like may trigger a wallet connection or login.

[0065] FIGS. 7a-b show user interfaces illustrating example sign-in and account management in some embodiments of the WSEP. Referring to FIG. 7a, in one implementation, a customer may launch a merchant site 705 and select the create account option 705a. Selection of the create account option may direct the customer to a merchant account creation page 710 in the merchant site 705. The customer may register for a merchant account by filling out the form 710a. Alternately, the customer may register a merchant account with the wallet account using the connect with wallet button 710b. When the connect with wallet button is selected, a wallet widget 715 may be launched within the merchant site 705. The customer may enter their wallet username and password (or other credentials) to gain access to the wallet widget configuration page 715a shown in FIG. 7b. Referring to FIG. 7a, in some implementations, the customer may already have a merchant account. The customer may enter their merchant site account credentials 720 and login to the merchant site page 725. The customer may, at this point, connect to the wallet by selecting the connect with wallet now button 725a which may launch the wallet widget 715.

[0066] Referring to FIG. 7b, the customer may configure merchant linkage to the wallet starting with option 715a for example. In one implementation, the customer may select preferences 720 for the merchant account in a more granular manner. For example, the customer may specify, for example, payment methods and shipping addresses to link to the merchant. Other preference management is discussed in further detail with respect to FIG. 12. Upon completing the preferences set up, the customer may select the connect button 720a to create the link between the merchant and the wallet. The wallet widget may then direct the customer to the merchant site 725. The wallet may also share or load or dynamically inject to the merchant site information according to the customer preferences. The merchant site 725 may obtain the shared information and display the shared payment methods, address, and other information 725a to the customer to confirm the connection between the merchant account and the wallet.

[0067] FIGS. 8a-b show user interfaces illustrating example sign-in and checkout in some embodiments of the WSEP. Referring to FIG. 8a, in one implementation, a customer may launch a merchant site 805 (or merchant application). Using the merchant sign in 805 option, the customer may be directed to a sign in page 810 in the merchant site 805, where the customer may login to the merchant site using username and password 810a for the merchant site. Alternately, the customer may login with the wallet using the login with wallet button 810b. When the login with wallet button is selected, a wallet widget 815 may be launched within the merchant site 805. The customer may provide wallet username and password 815a to login to the merchant site via the wallet. Referring to FIG. 8b, once the customer is authenticated via the wallet, the wallet may send the merchant the customer ID corresponding to the relationship between the customer and the merchant. The merchant, upon receiving the customer ID, and verifying that the customer ID corresponds to a customer record in their customer database, may allow the customer access to their merchant account 820. In one implementation, the customer sign in may be a trigger for the merchant to make an API/JAVASCRIPT call 855 to the wallet service 850 to obtain shipping details 825b, payment method 825c, and/or the like. The merchant site page 825 may use the shipping detail obtained from the wallet to calculate and display shipping and tax information. In one implementation, the payment method 825c obtained from the wallet may be a payment method nickname (e.g., my personal account). The merchant may not have the actual card or account number. The actual card or account number is resolved by the wallet once the customer selects the pay now with wallet button 835. In one implementation, the customer may also edit shipping address, payment method and other details directly from the merchant site using the edit with wallet button 830. Upon successful transaction authorization, the merchant site 805 may display the page 840, including information such as receipt 840a relating to the transaction.

[0068] FIGS. 9a-b show data flow diagrams illustrating example bi-directional federation in some embodiments of the WSEP. Referring to FIG. 9a, in one implementation, a user 902 may input login credentials (e.g., merchant account or wallet account username and password) at the merchant site or application on their client device 904 at 912. The client device may take the login credentials and generate an authentication request 914 for transmission to a merchant server 906. For example, the client may provide a (Secure) Hypertext Transfer Protocol ("HTTP(S)") POST message including data formatted according to the eXtensible Markup Language ("XML"). An example authentication request 914, substantially in the form of a HTTP(S) POST message including XML-formatted data, is provided below:

```
POST /authrequest.php HTTP/1.1
Host: www.merchant.com
Content-Type: Application/XML
Content-Length: 667
<?XML version = "1.0" encoding = "UTF-8"?>
<auth_request>
  <timestamp>2013-02-22 15:22:43</timestamp>
  <user_details>
    <user_name>JDoe@gmail.com</user_name>
    <password>Tomcat123</password>
  </user_details>
  <client_details>
```

-continued

```

<client__IP>192.168.23.233</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__detail>
</auth__request>

```

[0069] The merchant server **906** may receive the authentication request **914**, and may parse the request to obtain user and/or client details such as username and password. The merchant server may perform authentication of the user and/or client details at **916**. In one implementation, the merchant server may query its user/customer database to verify that the username and the password (or other credentials) are correct, and the user is authorized to access the account with the merchant (i.e., merchant account).

[0070] In another implementation, the user credentials may be authenticated by the wallet server **908**. The user may select sign in with wallet button and may input wallet credentials in the wallet widget launched. The client **904** may generate an authentication request **918** using the user provided login credentials. An example wallet authentication request **918**, substantially in the form of a HTTP(S) POST message including XML-formatted data, is provided below:

```

POST /authrequest.php HTTP/1.1
Host: www.wallet.com
Content-Type: Application/XML
Content-Length: 667
<?XML version = "1.0" encoding = "UTF-8"?>
<auth__request>
  <timestamp>2013-02-22 15:22:43</timestamp>
  <user__details>
    <user__name>JDoe1984</user__name>
    <password>thistryon56</password>
  </user__details>
  <widget__param>
    <apikey>aK2Lejj89j2A1_IOn4s2</apikey>
  </widget__param>
  <client__details>
    <client__IP>192.168.23.233</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__detail>
</auth__request>

```

[0071] At **920**, the wallet server may authenticate the user. In one implementation, OAuth protocol may be utilized to authenticate the user on behalf of the merchant. In one implementation, the wallet server may use the username and/or password, one or more widget parameters such as API key in the authorization request **918b**, and/or the like to obtain a customer ID associated with the user/customer and the merchant. The wallet server may send the customer ID in an authorization response **924** to the merchant. In one implementation, the authorization response **924** may be a back-end notification message sent from the wallet server to the merchant. An example notification message in POST method in XML format is provided below:

```

<?XML version = "1.0" encoding = "UTF-8"?>
<notification-auth>
  <timestamp>2013-02-22 15:22:43</timestamp>
  <customer__ID>56470898786687</customer__ID>
  <apikey>aK2Lejj89j2A1_IOn4s2</apikey>
</notification-auth>

```

[0072] The merchant server may receive the customer ID in the authorization response message **924**, and query their database to confirm that the customer ID matches a customer record in their customer database. Upon verification or successful authentication at **916**, the merchant server may send an authentication response **922** to the client **904**. The authentication response, in one implementation, may be the requested web page that is rendered by the client **904** and displayed to the user at **938**.

[0073] In one implementation, the merchant server may use the user sign as a trigger to request current user information from the wallet server. The merchant server may generate and send a user information request message **926** to the wallet server. The user information request message **926** may include, without limitation, the customer ID that is unique to the customer and the merchant relationship, a token, an API key, a digital certificate, and/or the like. In one implementation, the token may be generated using one or more parameters such as the merchant's API key, customer ID, merchant ID, merchant name, customer name, and/or the like. In a further implementation, the token may be encrypted. In one implementation, the token may be a string that is created by the MD5 Message Digest algorithm hash of one or more of the parameters listed above. In one implementation, the merchant server may utilize callbacks via APIs, inline widgets, etc., to pull user information from the wallet. For example, the merchant server may call the getPayment API to obtain payment method details such as card nicknames, brand, last 4 digits, etc. An exemplary GET request method for making the call is provided below.

```

http://server1.vwallet.com/wallet/api/getPayment?callid=100008&callno=
1&apikey=aK2Lejj89j2A1_IOn4s2&token=
u:c6a5941420cf67578986abe8e09a8299&customerid=
56470898786687

```

[0074] The wallet server may obtain the request **926** and may parse the request at **928**. In one implementation, the wallet server may validate the request by confirming the customer ID, API key and/or the token are correct. At **930**, the wallet server may use the customer ID, for example, to query one or more databases (e.g., customer profile database **910**) for user records. The wallet server may retrieve the user record, preferences, and/or permissions **932** from the customer profile database. In one implementation, the wallet server may use the associated preferences and permissions specified by the user to determine payment methods that the user has approved for sharing with the merchant. The wallet server may then generate the user information response message **934** for transmission to the merchant. An example response message **934** substantially in the form of a HTTP(S) POST message including XML-formatted data, is provided below:

```

<?XML version = "1.0" encoding = "UTF-8"?>
<payment_methods>
  <timestamp>2013-02-22 15:22:43</timestamp>
  <customer_ID>56470898786687</customer_ID>
  <call_ID>3</call_ID>
  <card1_details>
    <nickname>My personal card</nickname>
    <brand>Visa</brand>
    <digits>4554</digits>
  </card1_details>
  <card2_details>
    <nickname>My cashback card</nickname>
    <brand>Visa</brand>
    <digits>4557</digits>
  </card2_details>
  <card3_details>
    <nickname>My prepaid card</nickname>
    <brand>Amex</brand>
    <digits>5555</digits>
  </card3_details>
</payment_methods>

```

[0075] The merchant server may receive the response message 934, and may send the shared user information message 936 to the client, which renders the received message to display the current user information to the user at 928. Although only getPayment API call is discussed in detail, other API calls such as those listed in Table 1 may also be called by the merchant server to obtain information including address nick name, indicator for default/primary address, active loyalty programs, program names, indicator for current/primary loyalty program, request to instantiate a purchase against the customer ID, retrieve and redeem previous purchase records for the customer, and/or the like. In an alternate implementation, instead of the merchant making the API calls to obtain the user information, the wallet server may push user information to the merchant. In some implementations, the information push may be a one-time event, for example, when the user connects a new service (e.g., a merchant) to a wallet. In other implementations, the information push may be triggered by events such as the user signing in to a service account via the wallet.

[0076] Referring to FIG. 9b, in one implementation, the user may input new information to their merchant account. For example, the user may add a new shipping address to their merchant account. The client may take the user input and package it as an add new information request 952 to the merchant server. An example add new information request 952, substantially in the form of a HTTP(S) POST message including XML-formatted data, is provided below:

```

POST /addnewinfo.php HTTP/1.1
Host: www.merchant.com
Content-Type: Application/XML
Content-Length: 667
<?XML version = "1.0" encoding = "UTF-8"?>
<auth_request>
  <timestamp>2013-02-22 15:22:43</timestamp>
  <user_details>
    <user_name>JDoe@gmail.com</user_name>
    <password>Tomcat123</password>
  </user_details>
  <new_info>
    <shipping_address>
      <street_name>400 Turtle bay road</street_name>
      <apt_unit>6H</apt_unit>

```

-continued

```

      <city>New York</city>
      <zip_code>10086</zip_code>
    </shipping_address>
  </new_info>
  <client_details>
    <client_IP>192.168.23.233</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_detail>
</auth_request>

```

[0077] In one implementation, after receiving the new information request 952, the merchant server may parse the message, and retrieve the user record from the one or more databases and/or tables (e.g., customer profile database 909). The merchant server may then update the user record and store the updated user record 954 to the customer profile database 909. An exemplary listing, written substantially in the form of PHP/SQL commands, to update the user record 954 in the customer profile database, is provided below:

```

<?PHP
header('Content-Type: text/plain');
// store input data in a database
mysql_connect("201.408.185.132", $DBserver, $password); // access
database server
mysql_select("Customer_Profile_DB.SQL"); // select database to append
mysql_query("UPDATE UserTable
SET street_name = '400 Turtle bay road', apt_unit = '6H', city =
'New York', zip_code = '10086' timestamp = '2013-02-22 15:22:43'
WHERE username = 'JDoe@gmail.com'");
mysql_close("CSF_DB.SQL"); // close connection to database
?>

```

[0078] In one implementation, the merchant may send the new user information message 956 to the wallet server. An example new user information message 956, substantially in the form of a HTTP(S) POST message including XML-formatted data, is provided below:

```

POST /addnewinfo.php HTTP/1.1
Host: www.vwallet.com
Content-Type: Application/XML
Content-Length: 667
<?XML version = "1.0" encoding = "UTF-8"?>
<add_newinfo>
  <timestamp>2013-02-22 15:22:43</timestamp>
  <apikey>aK2Lejj89j2A1_On4s2</apikey>
  <token>uc6a5941420cf67578986abe8e09a8299</token>
  <customer_ID>56470898786687</customer_ID>
  <new_info>
    <shipping_address>
      <street_name>400 Turtle bay road</street_name>
      <apt_unit>6H</apt_unit>
      <city>New York</city>
      <zip_code>10086</zip_code>
    </shipping_address>
  </new_info>
</add_newinfo>

```

[0079] The wallet server may receive the new user information message 956 from the merchant, along with customer ID. The wallet server may parse the received information at 958. Using the customer ID extracted from the received infor-

mation, the wallet server may query one or more customer profile databases at **960**. At **962**, the server may obtain query results. In one implementation, the query may be performed to determine whether the field of new user information is a field that is permitted for updating using information from the merchant source. For example, in one implementation, shipping information may not be a field that is permitted for updating based on information from the connected service such as the merchant while other information such as a new telephone number received from the merchant may be used to update the customer record in the database (e.g., **910**). Such permissions for adding, removing, changing, updating, etc., information to and from the wallet may be specified by the user via the permission control panel discussed in detail with respect to FIG. 12. In some other implementations, whether information flowing from the merchant to the wallet server can be accepted by the wallet server, and used to update the customer records, may depend on the merchant trust level, how critical the update or change is (e.g., changing a payment method versus changing a telephone number), and/or the like. At **966**, depending on whether it is appropriate to update the customer record, the wallet server may or may not update the record. At **970**, the wallet server may send a confirmation message to the merchant server to confirm whether the new information was accepted, and the current information that is on the records in the wallet. At **972**, the merchant server may send the client a confirmation message whether the update was successful or not. The client may display the confirmation message at **974**. In one implementation, the wallet server may directly communicate with the user (e.g., via email, SMS, MMS, phone, etc.) at **968** and solicit and/or provide confirmation of the addition of the new information.

[0080] FIG. 10 shows a logic flow diagram illustrating an example account creation and management in some embodiments of the WSEP. In one implementation, at **1005**, if a customer has an existing merchant account, the customer may login using merchant account credentials **1010**. Alternately, the customer may login using their wallet account credentials **1015**. If the customer selects login via the wallet, a wallet widget may be provided at **1020** for the customer to enter their wallet credentials. At **1025**, if the customer does not wish to connect their wallet to their merchant account, the merchant may use the customer's information on file or solicit information from the customer to complete a transaction at **1030**. On the other hand, if the customer requests connection between the merchant account and the wallet account, and the customer is already authenticated by the wallet at **1035**, the customer may set preferences and permissions at **1045**. If the customer has not been authenticated, a wallet widget may be launched to obtain wallet credentials from the user for authentication at **1040**. At **1050**, the wallet may create a customer ID as a record of the relationship between the customer and the merchant, and the associated preferences and permissions. The customer ID may be sent to the merchant. Using the customer ID and/or API keys or tokens, the merchant may request customer information such as shipping address, payment method, and/or the like at **1055**. The wallet may provide the merchant the information that is permitted for sharing by the customer preferences and permissions. At **1060**, the merchant may use the information from the wallet to conduct a transaction. In one implementation, the transaction may be via the wallet. In another implementation, the transaction may be via a lightbox widget rendered within the merchant site.

[0081] In one implementation, if there is no existing merchant account as determined at **1005**, the customer may create a new merchant account. In one implementation, the customer may create a new merchant account via the merchant **1065** where the user may fill out a form with fields for name, address, email, username, password, and/or the like at **1075**. At **1080**, the merchant may use the customer provided information to create a new account for the customer and the decision may move to **1025**. If, on the other hand, the customer selects an option to create a new merchant account via the wallet **1070**, the WSEP may determine whether the customer has an existing wallet account at **1085**. If the customer does not have a wallet account, the WSEP may request the user to create a wallet account at **1090**. Once there is an existing wallet account, the WSEP may obtain customer wallet credentials, and may authenticate the user at **1092**. At **1094**, the WSEP may obtain preferences and/or permissions for the merchant account. At **1096**, the WSEP may create a customer ID that establishes the relationship between the merchant and the customer. In one implementation, the WSEP may store the preferences and/or permissions along with the customer ID in its customer database. At **1098**, the WSEP may provide user information allowed by the preferences and permissions to the merchant along with the customer ID. At **1062**, the merchant may receive the provided information and may create a merchant account for the customer. At **1060**, the merchant may use the wallet provided information to transact with the customer.

[0082] In some embodiments, the WSEP framework may be leveraged for prepaid card provisioning and personalization. An online wallet service such as V.me by Visa may store consumer information for a number of purposes including for expediting online shopping and checkout. Cardholder information (such as name, account number, contact information, billing and shipping addresses etc.) flows originally from an issuer through a provisioning process to the wallet and then by instruction of the consumer to a merchant at the time of checkout. Some embodiments of the WSEP entail reversing the flow of information, such that an online wallet may provision account information with an issuer and at the same time link the account records at the wallet with the account records of the prepaid issuer.

[0083] FIG. 11 shows a block diagram illustrating an example prepaid card personalization in some embodiments of the WSEP. In one implementation, a consumer having a wallet account may obtain a new gift card (open loop or closed loop) or a reloadable prepaid card **1130**. The consumer may personalize the card for online or offline usage and be able to view and service the account with the issuer. In one implementation, through the issuer's online or mobile service channel **1105**, the consumer may click a WSEP button **1110**. In a further implementation, the WSEP button may spawn a modal widget **1115** powered by the wallet. The consumer may authenticate to the wallet and may confirm their wish to personalize the new card and share the personalization information with the issuer. In one implementation, the personalization information may include information from the consumer profile in the wallet's central consumer profile database **1125** such as name, contact information, billing address, shipping address, card nickname, and/or the like. The wallet, upon receiving confirmation from the consumer, may share the consumer's personalization information with the issuer's prepaid platform service **1105**. The prepaid card may then be loaded and stored in the consumer's wallet

profile. In one implementation, once the prepaid card is linked to the wallet, the consumer may log in to the issuer's prepaid service using their wallet credentials (saving them having to remember additional usernames and passwords for every prepaid account). In a further implementation, the WSEP provides an option for prepaid platforms to integrate all prepaid card management and services directly into wallet platform. In some implementations, APIs for the wallet platform may be available to query current available balances and transaction history from issuer cards linked to the wallet service.

[0084] FIG. 12 shows a user interface illustrating an example WSEP settings control panel in some embodiments of the WSEP. The WSEP control panel may provide common customer experiences across different parties that are connected via the WSEP facilities to the wallet. Using the WSEP control panel, the customer may manage permissions and preferences for all parties connected to the wallet and establish a set of flexible standards to define which parties can read, write, update/modify or publish what customer profile information, which parties can execute transaction against the wallet account, or inject plug-ins and widgets to the wallet, and/or the like. Customers, including those who are concerned about how much data they should trust with various parties they do business with or use their services, may leverage the framework of the WSEP control panel to manage their identities and payments at various service providers such as merchants, utility providers, loyalty providers, money transfer services, and any other service providers ("merchants"). The components of the permissions/settings control panel may enforce terms of connection relations. For example all API calls by the service will be validated against the permissions and business rules expressly agreed to by the customer.

[0085] In one implementation, the WSEP control panel may include several panels such as service providers 1205, payment methods 1210, shipping address 1215, share 1220, permissions 1225, and/or the like. The service providers may include, without limitation, any party that a customer may do business with. The customer may have an identity, payment relationship, etc., established with such parties. The customer may select any one, multiple or all of the service providers 1205a-j for individual or group preference and permission management. In one implementation, the customer may select the merchant NORDSTROM 1205c. The customer may then configure each of the payment methods, shipping addresses, share, and permissions for the selected merchant 1205c. The payment methods panel 1210 may list one or more payment methods 1210a-d that are present in the wallet. The panel 1210 may display an image of the card (e.g., from the issuer), a nickname for the card, card identifier, card brand, and/or the like. The payment methods may also include bank or other financial accounts, debit cards, credit cards, prepaid cards, gift cards, and/or the like. In some implementations, the customer may also add new card to the wallet directly from the control panel interface. The customer may select one or more of these payment methods for sharing with the merchant 1205c. When the wallet provides the shared payment method to the selected service provider, only select information such as the nickname, brand, and last four digits of the card number, etc., may be shared. In some implementations, the actual card or account number may not be shared with the service provider.

[0086] The customer, using the permissions panel 1225, may authorize the service provider to execute transactions (option 1225a) against the wallet using the selected payment

methods. In some implementations, the customer may also set up, using the permissions panel 1225, recurring billing authorization 1225c, subscription payments 1225d, and/or the like. For example, at the end of a month, a merchant (e.g., AT&T) may request authorization from the wallet to bill a monthly charge amount (e.g., \$120.55) against the standing instructions for a "default" payment method by a customer having a customer ID. The wallet may be storing the standing payment instructions for "default" payment method in slot 1 of the wallet and a back up payment method in slot 2 of the wallet. The wallet may map slot 1 to an actual payment method and authorize billing using the actual payment method, without the merchant knowing the actual payment method. In one implementation, depending on the merchant request, a tiered authentication may be employed to more rigorously authenticate the merchant/customer. For example, a merchant that usually transacts against the primary card and primary shipping address may request to execute a transaction against another shipping address (e.g., grandma's address). Such a request may then cause the wallet to step up the authentication protocol (e.g., get customer confirmation, request digital certification, etc.) to ensure that the transaction being executed is not a fraudulent transaction.

[0087] In one embodiment, the WSEP may leverage its facilities to determine liability for transactions that happen based on trust relationships. For example, depending upon whether the merchant tries to bill the customer with or without popping up an extra widget to log on could affect the liability for the transaction. Using TSM (trusted service manager) protocols where a secure key from a issuer is passed to put on a phone or other client device, so that the wallet knows a secure key from the issuer was present during the transaction, may also prevent fraud and affect the liability for the transaction. Similar trust relationship could also be used for liabilities relating to change requests, for card not present transactions, and/or the like.

[0088] In some implementations, the customer may set up shipping address preferences for the service provider. The shipping address panel 1215 may display a list of shipping addresses 1215a-1215c stored in the customer profile with the wallet. Each of the shipping addresses may be nick named. The customer may select one or more of the shipping addresses for sharing with the merchant, and may add another address 1215d to the wallet directly from the shipping address panel 1215. In some implementations, the customer may allow shipping address to be a field which the service provider may have write access to by configuring the allow write access option 1215e. Such authorization for write access to the shipping address field of the customer profile record in the wallet's customer database may allow any changes the customer may make to the shipping information from the service provider interface to propagate to the wallet. Such a bi-directional flow of information may ensure true syncing of user information across various service providers and the wallet. In some implementations, the customer may configure, using the permissions panel 1225, that any profile changes must be confirmed with the customer (option 1225b). The wallet, in such a case, may send the customer a request to review and/or confirm the profile change, and may update its customer profile upon explicit approval from the customer.

[0089] In some implementations, the control panel's share panel 1220 may display a list of information fields that may be shared by the customer with the service provider. Examples of the fields of information include, without limitation, name

1220a, primary email address **1220b**, work email address **1220c**, information for account creation **1220d**, loyalty programs **1220e**, specific loyalty programs **122** of, wish lists **1220g**, points balance **1220h**, and/or the like. In one implementation, one or more of these fields may be configured for write access **1220i**. Using the permissions panel **1225**, the customer may further configure whether the service provider is allowed to execute transactions against the wallet **1225a**, authorized to bill the customer **1225c**, authorized the wallet to make/bill for subscription payments **1225d**, require confirmation before modifying the customer profile **1225b**, and/or the like. Various other permissions and panels for configuring and managing customer information federation are within the scope of the embodiments of the WSEP.

[0090] FIG. 12a is an example embodiment of a WSEP configured to display a success confirmation **1226** screen after the enrollment of new payment cards **1228**, **1229** in a virtual wallet account. In some embodiments, the wallet account may already been established and contain cards previously added **1227**.

[0091] FIG. 13a-i show example user interfaces in some embodiments of the WSEP.

[0092] FIG. 14 shows an exemplary screenshot depicting a merchant checkout system. In one embodiment, the WSEP may facilitate the administration of payments to merchants that contain a current transaction **1401** and a future transaction **1402**. In some embodiments, the merchant may place a button **1403** on their web page that may facilitate the creation of a reference account link. The button may, in some embodiments, contain information from the available reference transaction links previously created by the consumer. For example, the button may designate which reference account will be used for the transaction. In another example, the button may designate a reference for a shipping address to be used for the transaction or a persona that the user may wish to engage in the transaction using. Other embodiments may contain any manner of consumer information that may be subject to change over time.

[0093] FIG. 14a shows an exemplary screenshot depicting an inline login for accessing a consumer's WSEP account **1404**. In some embodiments, a user may log in using their email address and a password **1406**. In other embodiments, the user may optionally choose to create a virtual wallet account **1405** to facilitate future transactions with the current or other merchants.

[0094] FIG. 14b shows an exemplary screenshot depicting a merchant account creation screen facilitated by the WSEP. In this and other embodiments, the consumer may choose to create an account **1410** with the merchant and provide contact/shipping information **1407** and/or payment information **1408** to complete the transaction. Optionally, the consumer may choose to simultaneously create a virtual wallet account **1409** to facilitate future transactions with either the current merchant or other merchants.

[0095] FIG. 15 shows an example enrollment lightbox for creating a WSEP link between a user's virtual wallet and a merchant. In some embodiments, the enrollment form may contain details about the transactions authorized **1502**. The transactions may be one-time transactions, periodic transactions, recurring transactions, or any combination thereof. Additional terms may be included or associated with the reference transaction link. For example, some reference transaction links may have expiration dates, frequency caps, amount caps, alert requirements, heightened security requirements, or other desired limitations. In some embodiments, the user may be prompted to agree to the requirements for the

current or future transactions. A consumer may designate a payment account reference **1503** to use for the transactions. In some embodiments, the consumer may choose more than one payment reference account for the transactions. In alternative embodiments, the consumer may choose one payment account reference for the current transaction and a different payment account reference for future transactions. The consumer may also designate other information by reference either alone or in combination with reference payment transactions. For example, the consumer may designate a reference persona **1504** for the transaction. In some embodiments, the reference persona may contain contact information for the consumer. In other embodiments, the reference persona may contain contact information for another party. In still other embodiments, the reference persona may contain privacy enhanced information that limits the merchant's knowledge of some of the consumer's personal information or details. In some embodiments, the consumer may designate a reference address **1505** for use in the transaction. The reference address may be a user's preferred shipping address for a transaction. In other embodiments, the reference address may contain multiple addresses for use in various parts of the transactional relationship with the merchant. In still other embodiments, the reference address may be a designation that resolves to a third party that may then forward shipments to the consumer. In this embodiment, the consumer may advantageously be able to receive shipments using a reference address from a merchant without disclosing their actual address information to the merchant. Third parties may act as intermediaries for different types of reference links in various embodiments. In some embodiments, the consumer may click a button in the lightbox **1506** to link the selected references to the merchant. In other embodiments, the consumer may click a button **1507** to create a new reference. In doing so, the consumer may be prompted for information required to establish the reference link, such as adding a card to the consumer's wallet, adding an address for the reference link, or adding a persona to a virtual wallet. In some embodiments, the consumer may be presented with a QR code **1508**, bar code, or other visual element suitable for scanning by a mobile device. In doing so, the user may be able to establish the reference link with heightened security, less user input, or by sharing less information directly with the merchant. In some embodiments, the reference transaction link may be established to facilitate future refunds to the consumer. For example, a user may enroll a reference transaction link with an insurance provider to facilitate future claim refunds to the user's virtual wallet. In other embodiments, the refund reference link may be used by a merchant that has previously charged the user for a transaction. In alternative embodiments, the refund reference link is only used to facilitate refunds and may not be used for payments.

[0096] FIG. 16 shows an example user interface illustrating a reference management console. In some embodiments, the consumer can see the merchants associated with a payment reference **1601**. A nickname for a payment reference **1609** may be displayed in some embodiments. A consumer may update the nickname associated with a reference **1609** or the payment account the reference uses **1603** by clicking a button **1602** in one embodiment of the interface. In some embodiments, multiple payment accounts may be linked to one reference account. The nickname the user has chosen for the reference payment link **1609** may also be customized for various merchants using the reference **1604** to facilitate rec-

ognition of the reference account in the context of a merchant's web site. In some embodiments, the reference management console will show the terms of the financial relationship **1605** that the consumer has established with the merchants. The terms, in other embodiments, may be other than payment terms. For instance, terms may be product specifications, shipment standards, on-account credit agreements, or other aspects of the consumer's relationship with a merchant. In some embodiments, a transaction history is available in the management console or elsewhere in the WSEP. A consumer may also administer the reference transaction links from within the reference management console or elsewhere in the WSEP. For example, the consumer may revoke access to a merchant linked to a reference payment **1607**. A consumer may also cancel a recurrent subscription with a merchant from within the WSEP. In alternative embodiments, the consumer may request more favorable payment terms, incentives, value added services, or a refund through the reference management console or elsewhere throughout the WSEP.

[0097] FIG. 17 shows a block level diagram depicting exemplary failover payment capabilities of a reference transaction payment link. In one embodiment, the user may designate a reference name for a collection of payment accounts **1701**. The user may choose a primary account to be used if sufficient funds are available **1702** and a backup account to be used in the event the primary account link fails **1703**. A failure may be caused by insufficient funds, account closure, or other events. In an example transaction, merchant **1707** may use reference **1701** to execute a transaction that resolves to payment account **1702** and successfully processes the payment **1704**. In another example, if the reference link to the primary payment method is broken **1705**, the transaction may still resolve to backup payment method **1706**. In alternative embodiments, the consumer may designate rules regarding the order in which payment accounts should be used by a reference link and what criteria should determine the order. For example, a consumer may decide that all transactions from a certain type of merchant (i.e., grocery transactions, foreign travel transactions, etc.) should be processed through one payment account associated with the reference payment link. The consumer may also designate other payment accounts to handle transactions of other types.

[0098] FIGS. 18 and 18a are exemplary datagrams depicting the creation of a reference payment link between a merchant and a user. In FIG. 18, user **1821** requests a checkout page using a client terminal **1806**. The checkout page request **1802** is dispatched to a merchant web server **1803**. The merchant web server then replies to client **1806** with a checkout page response **1804**. The checkout page response **1804** is embedded with code that causes client to initiate a second request to a wallet server. The client **1806** parses the checkout page response **1805**. The client then dispatches a second request **1807** to a wallet server for a payment button. The wallet server responds with a payment button **1809**, which is rendered by the client terminal **1820**. The user then designates the payment button using an input device such as a mouse or finger **1822**. The client **1806** then dispatches a request for a lightbox **1823** to wallet server **1808**. The wallet server replies with a lightbox response **1824** containing reference transaction link information. In some embodiments, the lightbox response is substantially in the form of an HTTP(S) message including XML-formatted data, as provided below:

```
Host: www.merchant.com
Content-Type: Application/XML
Content-Length: 667
<?XML version = "1.0" encoding = "UTF-8"?>
<lightbox_response>
  <timestamp>2013-02-22 15:22:43</timestamp>
  <user_details>
    <user_name>JDoe@gmail.com</user_name>
    <password>Tomcat123</password>
  </user_details>
  <reference>
    <refname>Personal Card</refname>
    <type>reference_payment</type>
    <contract_id>1Z4567248987321</contract_id>
    <contract_trms>234.99,immediate|40.00,permonth</contract_trms>
  </reference>
  <reference>
    <refname>Secret Name</refname>
    <type>reference_persona</type>
    <name>Alias Name</name>
  </reference>
  <reference>
    <refname>Vacation Address</refname>
    <type>reference_address</type>
    <addr>500 Main St.</addr>
    <city>Anycity</city>
    <state>NY</state>
    <zip>12345</zip>
  </reference>
</lightbox_response>
```

[0099] The datagram in FIG. 18 then continues in FIG. 18a. Client **1806** then renders the lightbox **1825**. In some embodiments, the lightbox appears overlaid on the merchant's web site. In other embodiments, the lightbox appears in a different window. Upon rendering of the lightbox, user **1821** is then presented with reference links that have already been created. In some embodiments, the user may re-use a previously created reference payment, persona, address, or other link by selecting its alias from the lightbox. In other embodiments, the user can create a new reference link from within the lightbox. In some embodiments, the reference creation request **1827** will be substantially in the form of an HTTP(S) message including XML-formatted data, as provided below:

```
Host: www.merchant.com
Content-Type: Application/XML
Content-Length: 667
<?XML version = "1.0" encoding = "UTF-8"?>
<reference_creation_request>
  <timestamp>2013-02-22 15:22:43</timestamp>
  <user_details>
    <user_name>JDoe@gmail.com</user_name>
    <password>Tomcat123</password>
  </user_details>
  <new_reference>
    <refname>New Business Card</refname>
    <type>reference_payment</type>
    <card_num>1234123412341234</card_num>
    <contract_trms>234.99,immediate|40.00,permonth</contract_trms>
  </new_reference>
</reference_creation_request>
```

[0100] In some embodiments, wallet server **1808** will then process the reference creation request. For example, the wallet server may verify that the reference payment may be linked to the merchant. The wallet server may also verify that

the reference payment account has sufficient funds to cover the current or future transactions. The wallet server **1808** then may reply to client **1806** with a reference creation response indication successful or failed reference creation. The client **1806** may then render response **1830**.

[0101] FIG. **19** illustrates an example issuer side wallet enrollment interface user interface. In some embodiments of the WSEP, a consumer may be logged into their bank issuer's web site or mobile application **1901**. The web site may provide a listing of accounts that are associated with the consumer **1902-1902a**. Additionally, recent transaction and balance information **1904-1904a** may be provided to the consumer. In one embodiment, a consumer may add one or more accounts to a virtual wallet by indicating which accounts from the accounts associated with the issuer should be added to the virtual wallet **1903-1903a**. In other embodiments, the consumer will be able to select multiple cards for simultaneous addition to a virtual wallet.

[0102] FIG. **20a** illustrates a lightbox window **2001** for linking payment accounts to a virtual wallet, creating a virtual wallet, and/or simultaneously creating a virtual wallet and linking payment accounts to the newly created wallet account. In some embodiments, the lightbox is generated from a third-party provider through the use of Server-Side-Includes, absolute URL's, JavaScript, or other like inclusion mechanism. In other embodiments, the lightbox may instead be displayed after forwarding the user to a third-party web site and/or in a form that encompasses an entire browser window. In some embodiments, the consumer may desire to enroll more than one card **2002** simultaneously in their wallet account. As such, the lightbox may facilitate through one interface the simultaneous addition **2003** of multiple cards to a wallet account. In some embodiments, the user may already have a virtual wallet account that they wish to associate the payment accounts with **2004**. As such, the lightbox may solicit from the user credentials sufficient to identify the virtual wallet account to which the payment accounts should be added. In some embodiments, the credentials may be in the form of a user name/password combination, a user name/Email combination, and/or the like **2005**. Once the user has entered the appropriate wallet credentials, they may then link the payment accounts to the wallet **2006**. This may result in the lightbox (e.g., from an issuer, merchant, and/or a like source) creating message **2221** and pulling the information from the issuer server (see FIG. **22b**). In other embodiments, the consumer may desire to simultaneously create a virtual wallet account and add the selected payment accounts to the wallet **2007**. Advantageously, in some embodiments the consumer may desire to allow the issuer of the payment accounts to send information regarding the consumer's financial account with the issuer and/or the consumer's payment accounts with the issuer to the virtual wallet account provider **2008**. In doing so, the consumer may be assisted in the creation of a virtual wallet account by avoiding the entry of repetitive data that the issuer already has on file. This pre-fill of data may also be advantageously used in the establishment of other account types, including pre-paid accounts, reward accounts, savings accounts, and/or the like. In other embodiments, the consumer may indicate that the virtual wallet account is to be set up with the requirement for two factor authentication **2009**. Two factor authentication is a form of authentication that requires two distinct types of information in order to authenticate a user. For example, a user may be required to provide a user name/password combination and a

one-time code generated by their mobile device. Alternatively, the user may be required to identify an image of a friend and provide a thumbprint. Any two types of information that are known to a consumer may be used to enable two-factor authentication using the WSEP. In other embodiments, the consumer may be prompted to simultaneously create a pre-paid payment account while they are creating a new wallet and/or linking payment accounts to an existing wallet. In some embodiments, if a consumer chooses to create a pre-paid account they will be prompted to select a payment account from which to fund the pre-paid account. In other embodiments, the consumer may then enter the account information (e.g., account number, billing address, etc.). In still other embodiments, the account information may be retrieved from the account issuer or from the issuer the consumer is currently logged into. In some embodiments, the consumer may desire to create a rule set that will define the conditions in which the pre-paid account may be replenished with funds. Some example rules include the re-filling of the pre-paid account when the account balance reaches a threshold, the re-filling of the pre-paid account when a user's chosen financial account(s) reach a certain balance amount and/or receive a deposit of a certain size, and/or the like. In doing so, the WSEP may enable a user to easily create a pre-paid account while linking another account to their virtual wallet, creating a virtual wallet, and/or the like. In some embodiments, the pre-paid card creation request **2010** will be substantially in the form of an HTTP(S) message including XML-formatted data, as provided below:

```
Host: www.foo.com
Content-Type: Application/XML
Content-Length: 667
<?XML version = "1.0" encoding = "UTF-8"?>
<prepaid_creation_request>
  <timestamp>2020-02-22 15:22:43</timestamp>
  <user_details>
    <user_name>JDoe@gmail.com</user_name>
    <password>Tomeat123</password>
    <billing_address>123 Main St.</billing_address>
    <billing_state>VA</billing_state>
    <billing_zip>12345</billing_zip>
  </user_details>
  <prefill_data_source>
    <type>prefill_from_issuer_account_data</type>
    <account_number>456456456456</account_number>
  </prefill_data_source>
  <new_prepaid_card>
    <name>Lunch Money Prepaid Card</name>
    <type>prepaid</type>
    <funding_source>
      <type>rewards_points_account</type>
      <initial_deposit>10000points</initial_deposit>
      <currency_value>$124.52</currency_value>
    </funding_source>
    <funding_source>
      <type>savings_account</type>
      <account_number>1234123412341234</account_number>
      <routing_number>012345678</routing_number>
      <initial_deposit>$500.50</initial_deposit>
    </funding_source>
    <funding_source>
      //...n-sources of funding...
    </funding_source>
    <replenishment_rule>
      <type>low_prepaid_balance_initiate_deposit</type>
      <trigger_value>$20.00</trigger_value>
      <expires>2010-01-01</expires>
    </replenishment_rule>
```

-continued

```

<replenishment_rule>
  <type>date</type>
  <frequency>monthly</frequency>
  <day>15</day>
  <expires>never</expires>
</replenishment_rule>
</new_prepaid_card>
</prepaid_creation_request>

```

[0103] In some embodiments, the user may desire to simultaneously pre-fill information at the virtual wallet provider, force two-factor authentication before using the virtual wallet account, and/or establish a pre-paid payment account **2011**.

[0104] FIGS. **20b-20d** show an example alternate embodiment of the interface as described in FIG. **20a**. In some implementations, the user may be presented to a card management screen (e.g., from an issuer, merchant, and/or like source) that allows the user to select **2012** bank credit cards **2013a** and/or debit cards **2013b** to be used in the user's virtual wallet. In some implementations, information **2014** related to each card may be displayed with the card selection, including the card number, the card balance, images of the card, and/or like identifying information. After entering sign-in information **2015** for the user's virtual wallet account (e.g., a username or email address, a password, and/or like information), the user may click a button **2016** to submit the chosen cards and to log into the user's virtual wallet account. This may result in the website (e.g., from an issuer, merchant, and/or a like source) creating message **2220** and pushing the information to the virtual wallet server (see FIG. **22b**).

[0105] If the user does not have a virtual wallet account, the user may sign up via filling out a form **2017** as shown in FIG. **20C**, which may ask the user for identification information (e.g., a name, username, and/or the like), an email address, a password for the account, other information (e.g. gender, address, and/or the like), and/or like information. Once the user has entered said information, the user may click the continue button **2016** to submit the request for an account and the card selections to be associated with the newly-created account.

[0106] In some implementations, the WSEP, before submitting the card selections, may present the user with lightbox **2018**, which may indicate which cards have been selected. The user may have the ability to confirm the card selections by leaving all of the selections **2019** as-is and clicking the complete button **2021**, may deselect one or more of the selected cards and click the complete button, or may click the start over button **2020** in order to clear all selections and to return to the card selection interface. As such, in such implementations, only the accounts checked or otherwise selected by the user will be passed to the virtual server and added to the user's virtual wallet. Once the user has clicked the complete button, the bank issuer may package the information received from the user, and may send it to the WSEP. The WSEP may then send a request to a virtual wallet server, authenticating the user's account via the submitted login data, and requesting that the virtual wallet server associate the specified cards with the user's virtual wallet. If the user submitted information for creating a new virtual wallet account, the WSEP may instead send a request that creates a virtual wallet account for the user and associates the specified cards with the user's virtual wallet.

[0107] FIG. **21** is an example data and logic flow illustrating the enrollment of a consumer account in a virtual wallet service and the utilization of a pre-fill service to pre-populate information necessary for wallet enrollment. In some embodiments, the consumer is directed to the virtual wallet enrollment page by directly typing the enrollment URL in a web browser **2101**. In some embodiments, the consumer is navigated to a wallet login page where they may log into a wallet or create a new wallet account **2101a**. In other embodiments, the consumer may enroll in the virtual wallet through a link in their issuer's web site, credit card company, rewards online access account, and/or the like. In some embodiments, the user may then create a virtual wallet account **2102**. In other embodiments, the user will log into their pre-existing virtual wallet account. The user may then activate the wallet account **2102a**. The user may then indicate that they desire to add a new payment account to their virtual wallet **2103**. The WSEP may then request that the user consent to the retrieval of their payment account information from the payment account issuer **2104**. The user may be asked to provide the account number of the payment account that the user wishes to link to their virtual wallet account **2105**. The WSEP may then use the user's account number or other credential such as a username/password combination or the like to initiate a request for retrieval of pre-provisioned data associated with the payment account **2106**. In some embodiments, the request for retrieval of pre-provisioned data **2106** (e.g., "prefill data") may be in the form of an HTTP(S) message including XML-formatted data containing fields substantially similar to the following:

Element Name	Description	Field Size	Element Type	Business Rule
BID	Business ID of the Issuer		Numeric	For Federated Scenarios BID and CID
CID	Customer ID of the Cardholder		Numeric	
PAN	PAN Number of the Cardholder		Numeric	For Manual scenario PAN entered by the user

In some embodiments, the request for retrieval of pre-provisioned data **2106** (e.g., "prefill data") will be substantially in the form of an HTTP(S) message including XML-formatted data, as provided below:

```

Host: www.server.com
Content-Type: Application/XML
Content-Length: 667
<?XML version = "1.0" encoding = "UTF-8"?>
<preprovisioned_prefill_request>
  <BID>247581</BID>
  <CID>9854254</CID>
  <PAN>1234123412341234</PAN>
  <wallet_id>RW987856</wallet_id>
</preprovisioned_prefill_request>

```

[0108] In some embodiments, the issuer may then use the data in the request to perform a lookup of account and/or prefill information that may be shared with the requesting service. In some embodiments, the issuer will have a permissions rule set that governs what data may be shared with requesting services. Example rules include, "Never share my business account number," "Default to my personal account,"

“Never share my billing address,” and/or the like. In some embodiments, the issuer may then respond to the virtual wallet server **2107** with a prefill data package containing user, user account, user financial account, and/or similar data for use in establishing a virtual wallet account, pre-paid account, enrolling a payment account in a virtual wallet, and/or the like. In some embodiments, the pre-provisioned data response **2107** (e.g., “prefill data”) may be in the form of an HTTP(S) message including XML-formatted data containing fields substantially similar to the following:

Element Name	Description	Field Size	Element Type	Business Rule
<User Details>				
BID	Business ID of the Issuer	8	Alpha Numeric	Identification of the bank
CID	Customer ID of the Cardholder	19	Numeric	The CID The Customer ID is a unique identifier the user for the given issuer. This field is used to link the accounts (PANs) for a given user for the BID
Name Prefix		5	Alpha Numeric	
First Name	Cardholder first name	15	Alpha Numeric	
Middle initial	Cardholder middle name initials	1	Alpha Numeric	
Last Name	Cardholder last name	25	Alpha Numeric	
Name Suffix	Cardholder suffix	5	Alpha Numeric	
Company Name		40	Alpha Numeric	Company name if the account is held by a company instead of an individual
Country Code		3	Alpha Numeric	Country or Residence of the cardholder Numeric Country code ISO Numeric Currency Code: USA: 840 Canada: 124
Language Code	Cardholder language as set with the issuer	8	Alpha Numeric	Cardholder language as set with the issuer
<Card Details>				
Account Number	PAN Number of the Cardholder	19	Alpha Numeric	Card Number
Card Expiry Date	Expiration date of the card	4	UN	The expiration date as provided on the card Format: YYMM
Card Brand		4	Alpha Numeric	Example of the card brand: Visa
Product Identifier		2	Alpha Numeric	
Company Name		40	Alpha Numeric	
Name on the Card		26	Alpha Numeric	
Phone Number on back of the card		10	UN	
Billing Cycle Start Date		8	UN	Account Billing Cycle start date, used for spend accumulations and reminders

-continued

Element Name	Description	Field Size	Element Type	Business Rule
Street Number		10	AN	Billing Address street number
Address Line 2		40	AN	
Street Name		40	AN	Billing Address street name
Unit Number		10	AN	
PO Box Number		10	AN	
City		30	AN	Billing Address City
State		2	AN	Billing Address state For U.S.
Province		10	AN	Billing Address province For Canada
ZIP		10	UN	Billing Address zip code
Country		3	AN	Billing Address country
Product type		10	AN	The product type as provided on the card: Credit Debit Prepaid
Card Image Name		50	AN	
Reason code				

In some embodiments, the pre-provisioned data response **2107** (e.g., “prefill data”) may be in the form of an HTTP(S) message including XML-formatted data substantially similar to the following:

```
Host: www.server.com
Content-Type: Application/XML
Content-Length: 667
<?XML version = "1.0" encoding = "UTF-8"?>
<preprovisioned_prefill_response>
  <BID>247581</BID>
  <CID></CID>
  <wallet_id>AK21574</wallet_id>
  <name_prefix></name_prefix>
  //reference link may be used in place of data
  <first_name>
    ref_link=http://visanet.com/?walletid=AK21574&user_id=9548field=
    first_name
  </first_name>
  <middle_initial></middle_initial>
  <last_name>Doe</last_name>
  <name_suffix></name_suffix>
  <company_name></company_name>
  <country_code></country_code>
  <language_code></language_code>
  <account_number>
    ref_link=http://visanet.com/?walletid=AK21574&user_id=9548field=
    account_number
  </account_number>
  //alternatively, parameters (e.g. card expiration date)
  //can be made a live link requiring no parsing
  <card_expiry_date>
    ref_link=http://visanet.com/?walletid=AK21574&user_id=9548field=
    card_expiry_date>09/2020</card_expiry_date>
  <card_brand>Visa</card_brand>
  <product_identifier></product_identifier>
  <company_name></company_name>
  <name_on_card>
    ref_link=http://visanet.com/?walletid=AK21574&user_id=9548field=
    name_on_card>John Doe</name_on_card>
  <phone_number_on_card></phone_number_on_card>
  <billing_cycle_start_date></billing_cycle_start_date>
  <street_number>
```

-continued

```

ref_link=http://visanet.com/?walletid=AK21574&user_id=9548field=
street_number>58</street_number>
  <address_line_2></address_line_2>
  <street_name
ref_link=http://visanet.com/?walletid=AK21574&user_id=9548field=
street_name>Main St.</street_name>
  <unit_number></unit_number>
  <pobox_number></pobox_number>
  <city
ref_link=http://visanet.com/?walletid=AK21574&user_id=9548field=
city_name>Anytown</city>
  <state
ref_link=http://visanet.com/?walletid=AK21574&user_id=9548field=
state>VA</state>
  <province></province>
  <zip
ref_link=http://visanet.com/?walletid=AK21574&user_id=9548field=
zip>11547</zip>
  <country></country>
  <product_type></product_type>
  <card_image
ref_link=http://visanet.com/?walletid=AK21574&user_id=9548field=
card_image>http://www.imageserver.com/DRESKKJHKUHU/
?764765765765</card_image>
  <reason_code></reason_code>
</preprovisioned_prefill_response>

```

[0109] In some embodiments, the pre-provisioned data response **2107** may contain reference links (e.g., **1503**, **1504**, **1505** and/or the like) allowing dynamic updating of the data in the virtual wallet and/or at the payment card issuer. In some embodiments, the virtual wallet may then pre-populate the provided information **2108** into a form for enrollment of the user's payment account, rewards account, and/or like in the user's virtual wallet. In some embodiments, the WSEP may then make a request to retrieve an image for the card and/or payment account being added to the virtual wallet **2109**. In some embodiments, the card image may be a default image. The wallet server may store the card images locally, in a cache, or retrieve the card images via a web service such as XML-RPC, SOAP, and/or the like. In some embodiments, the image retrieval request **2109** may be in the form of an HTTP (S) message including XML-formatted data containing fields substantially similar to the following:

Element Name	Description	Field Size	Element Type	Business Rule
Account Number	PAN Number of the Cardholder	19	Alpha Numeric	For Manual scenrio PAN entered by the user

In other embodiments, the image retrieval request **2109** will be substantially in the form of an HTTP(S) message including XML-formatted data, as provided below:

```

Host: www.accountcardimageserver.com
Content-Type: Application/XML
Content-Length: 667
<?XML version = "1.0" encoding = "UTF-8"?>
<retrieve_image_request>
  <timestamp>2020-02-22 15:22:43</timestamp>
  <account_number>1234123412341234</account_number>
  <user_identifier>987654874</user_identifier>
  <image_resolution_desired>400x200</image_resolution_desired>
  <image_formats_desired>

```

-continued

```

  <type preference=1>JPG</type>
  <type preference=2>PNG</type>
  <type preference=3>HTML</type>
</image_formats_desired>
<image_formats_accepted>
  <type>JPG</type>
  <type>PNG</type>
  <type>HTML</type>
  <type>GIF</type>
</image_formats_accepted>
</retrieve_image_request>

```

[0110] In some embodiments, the card image server will then query a data store for an image of the card. An example PHP/SQL listing for querying a database for a card image is provided below:

```

<?PHP
header('Content-Type: text/plain');
mysql_connect("254.93.179.112",$DBserver,$password); // access
database server
mysql_select_db("CARDIMAGES.SQL"); // select database table to
search
//create query for token arbitrators
$query = "SELECT card_id, file_location, file_format FROM
CardTemplate WHERE card_type LIKE '%" . $usercardtype . "'";
$result = mysql_query($query); // perform the search query
mysql_close("ARBTRATORS.SQL"); // close database access
?>

```

[0111] The card may be a card virtually identical to the card the consumer is enrolling, or the card may be of a similar kind but of a more generic type (e.g., "green card," "gold card," "loyalty card," and/or the like). The data store may have multiple versions of the card available in various size/pixel resolutions and/or image formats. In some embodiments, the card image most closely matching the user's request will be returned to the user. In other embodiments, all card images meeting any of the criteria may be returned. In still other embodiments, the card image server may create an image "on the fly" in real-time using a dynamic image creation tool and/or a template tool such as ImageMagik, Gimp, Photoshop droplets, and/or the like. In one embodiment of the invention, the card template image retrieved from **2419i** may be overlaid with a logo, photo of the user, or other similar data using Bash ImageMagik UNIX instructions substantially similar to:

```

#!/bin/bash
composite -compose atop -geometry -13-171 card_overlay.png
card_template.png
card_output.png

```

The card image server may then return a data package containing descriptive information about the images returned, user data, account data, actual image data, and/or the like. In some embodiments, the image retrieval response **2109a** will be substantially in the form of an HTTP(S) message including XML-formatted data containing fields substantially similar to the following:

-continued

Element Name	Description	Field Size	Element Type	Business Rule
BID	Business ID of the Issuer	8	Alpha	Identification
CID	Customer ID of the Cardholder	19	Numeric	The CID
				The Customer ID is a unique identifier for the user for the given issuer. This field is used to link the accounts (PANs) for a given user for the BID
Account Number	PAN Number of the Cardholder	19	Numeric	
Card Image File Name		50	Alpha	
Reason code			Numeric	

In still other embodiments, the image retrieval response **2109a** will be substantially in the form of an HTTP(S) message including XML-formatted data, as provided below:

```
Host: www.accountcardimageserver.com
Content-Type: Application/XML
Content-Length: 667
<?XML version = "1.0" encoding = "UTF-8"?>
<retrieve_image_response>
<timestamp>2020-02-22 15:22:43</timestamp>
  <account_number>1234123412341234</account_number>
  <image_format>JPG</image_format>
  <image_generated_type>on-the-fly-generated
</image_generated_type>
  <image_binary_data>
```

```
SDFRDTCXREERXFDGXFDXRESRXREX...TREE#W#E
JUGYTFTRCCBBJHGFEER'&&'YHGJNKOIBJJVH
NMJNKJYT%TYFVVYTYVVBGUGUYGUYERSESWCGVU
VDRTGCDSERFDCVUE$RDTYYYYGVTYFTDGUIHUNI
</image_binary_data>
<image_url>http://imageserver.com/abc/image.jpg</image_url>
<cache_available_until>2030-02-22 15:22:43
</cache_available_until>
</retrieve_image_response>
```

[0112] In some embodiments, the image response may contain a cache control indication. The image server may indicate that it will cache the image for use by the wallet server, user, and/or like until a certain date or time. Alternatively, the cache date may be set to a date in the past, which indicates that the image will not be cached. By using a cached version of the image, the card image server may advantageously be able to provide individually customized versions of the card images for card image requesters without having to frequently regenerate customized card images (e.g. images containing a logo, or the user's name and/or photo) frequently. After the card image has been retrieved, the user may click a "Save" button to enroll the card in the wallet. In other embodiments, no card image is retrieved. In still other embodiments, the payment account is automatically added to the wallet. Additional logging and/or data storage may take place on the wallet server and/or data may be stored in a staging table **2111**, such as delayed processing of card enrollment requests during heavy periods of load. In some embodiments, the enrolled payment account and/or wallet enrollment data will be stored in a staging table for later processing **2111a**. In some embodiments, the data stored in the staging table **2111a** will be substantially in the form of an HTTP(S) message including XML-formatted data containing fields substantially similar to the following:

Element Name	Description	Field Size	Element Type	Business Rule
BID	Business ID of the Issuer	8	Alpha	Numeric
CID	Customer ID of the Cardholder	19		Numeric
Account Number	PAN Number of the Cardholder	19	Alpha	Numeric
Replaced Account Number		19	Alpha	Numeric
URI				Old Account Number
Name Prefix		5	Alpha	Numeric
First Name	Cardholder first name	15	Alpha	Numeric
Middle Initial	Cardholder middle name initials	1	Alpha	Numeric
Last Name	Cardholder last name	25	Alpha	Numeric
Name Suffix	Cardholder suffix	5	Alpha	Numeric
Company Name		40	Alpha	Numeric
Country Code		3	Alpha	Numeric
Language Code	Cardholder language as set with the issuer	8	Alpha	Numeric
Primary E-Mail Address		50	Alpha	Numeric

/vManage/v1/account/(GUID)/payment-Instruments/(paymentInstrumentID)

Company name if the account is help by a company instead of an individual
Country of Residence of the cardholder
Numeric Country code
ISO Numeric Currency Code
USA: 840
Canada: 124

Cardholder language as set with the issuer
Cardholder primary email address, this field may be used as the user ID in the wallet

-continued

Element Name	Description	Field Size	Element Type	Business Rule
Primary E-Mail Address Verification		1	Alpha Numeric	This field indicates whether this email address has been verified as valid email address for the cardholder
Secondary E-Mail Address		50	Alpha Numeric	Cardholder alternate or secondary email address
Secondary E-Mail Address Verification		1	Alpha Numeric	This field indicates whether this email address has been verified as a valid email address for the cardholder
Home Phone Number Country Code		3	UN	Country Code prefix USA: 001 Canada: 001
Home Phone Number		10	Alpha Numeric	
Primary Mobile Phone Number		3	UN	Country Code prefix USA: 001 Canada: 001
Country Code				
Primary Mobile Number		10	UN	
Primary Mobile Number Verification		1	Alpha Numeric	This field indicates whether this mobile number has been verified as a valid mobile number for the cardholder
Alternate Mobile Phone Number		3	UN	Country Code prefix USA: 001 Canada: 001
Country Code				
Alternate Mobile Number		10	UN	
Alternate Mobile Number Verification		1	Alpha Numeric	This field indicates whether this mobile number has been verified as a valid mobile number for the cardholder
Work Phone Number Country Code		3	UN	Country Code prefix USA: 001 Canada: 001
Work Phone Number		10	UN	
Work Phone Number Extension		10	UN	
Fax Number		3	UN	Country Code prefix USA: 001 Canada: 001
Country Code				
Fax Number		10	UN	
Card Brand		4	Alpha Numeric	Example of the card brand: Visa
Product Identifier		2	Alpha Numeric	
Company name		40	Alpha Numeric	
Name on the Card		26	Alpha Numeric	
Phone Number on back of the card		10	UN	
Billing Cycle Start Date		8	UN	Account Billing Cycle start date, used for spend accumulations and reminders
Street Number		10	AN	Billing Address street number
Address Line 2		40	AN	
Street Name		40	AN	Billing Address street name
Unit Number		10	AN	
PO Box Number		10	AN	
City		30	AN	Billing Address City
State		2	AN	Billing Address state For U.S.
Province		10	AN	Billing Address province For Canada
ZIP		10	UN	Billing Address zip code For United States and Canada
Country		3	AN	Billing Address country
Product Type		10	AN	The product type as provided on the card: Credit Debit Prepaid
Card Image Name		50	Alpha Numeric	
Enrolled Indicator		1	Alpha Numeric	
Card Added Method		25	Alpha Numeric	Federated Manual

[0113] The pre-provisioned data record may then be updated with the new wallet UUID **2111C**. In some embodiments, the record will be marked with an indication of enrollment method (such as “manual”) and additional data will be associated with the record such as an auto-update flag used in reference transactions, an account level identifier for associ-

ating child accounts with a parent account, acceptance of a terms and conditions, and/or a hashed card art image name **2111b**. In some embodiments, the user will receive an indication that they have completed the payment account enrollment in the virtual wallet **2112**, creation of the wallet account, and/or the like.

[0114] FIG. 22a is an example wallet account enrollment optionally using prefill data from a payment account issuer. In some embodiments, the consumer is logged into an issuer's web site 2201. The consumer may click a button indicating that they wish to enroll payment accounts associated with the issuer in a virtual wallet 2201a. The consumer may indicate that they wish to enroll some or all of their payment accounts with the issuer in a virtual wallet service 2201a. As such, the user may be asked to give their consent to their account information being transferred from the issuer to a virtual wallet provider 2202. The user may accept the message 2202a. In some embodiments, the issuer may then transfer the prefill and/or pre-provision data for all of the cards associated with a consumer user via a SAML assertion or other transfer mechanism 2203, which may be achieved using a data structure for each account similar to the above discussed pre-provisioned data response 2107. In some embodiments, payment account data may be stored by the wallet server 2203a. In other embodiments, the consumer will select which accounts information they desire to be transferred to the virtual wallet provider. In some embodiments, the consumer may then be transferred to the virtual wallet provider's web site 2203. A log-in page is then shown to the consumer 2204 to enable the consumer to log into their virtual wallet account. In some embodiments, the consumer may be automatically logged into their virtual wallet. In some embodiments, the consumer may log into their existing wallet using an email address and password and/or other similar means 2204b. A consumer may then indicate that they wish to enroll a card in a virtual wallet, such as by clicking an "Add Card" button 2204a, 2204c. The WSEP may request that the user consent to retrieve card prefill data from an issuer 2204d. In other embodiments, the consumer may be presented with a list of the payment accounts transferred from the issuer and/or images of the card accounts transferred and select which accounts to link to their virtual wallet. In some embodiments, the consumer may type the number of the account that they wish to add to their virtual wallet 2205. The WSEP may then verify that the account number is associated with one of the accounts with data transferred from the issuer as pre-fill and/or pre-provision data 2206. In some embodiments, the system may then pre-populate appropriate data in the enrollment form and request that the user indicate if they would like automatic updating of data after enrolled 2207. Examples of automatic updating (references) can include account number (e.g. PAN) and/or expiration dates 2207. In some embodiments, the WSEP may then advantageously pre-populate the pre-provision and/or pre-fill data into input boxes for the user to enroll their payment account. The user may then enroll their card in the virtual wallet by clicking a "Save" button 2208. In other embodiments, the payment account is automatically added to the virtual wallet without user interaction. The WSEP may perform address validation or verification 2208a prior to attaching the card to a virtual wallet. In some embodiments, the WSEP will then associate the added payment account(s) to the user's pre-existing virtual wallet 2209. In other embodiments, a new virtual wallet will be created. The WSEP may additionally create an entry in a staging table 2210, using means substantially similar to 2111, 2111a and/or 2111b. The prefilled/preprovisioned data may be inserted into the staging table with an enrolled designation 2210a. Later, records may be pulled from the staging table by an automated process and/or similar means, processed, and pushed to a common services platform 2210b. A record may be stored by the wallet server or otherwise indicating that the consumer was enrolled in a wallet account or payment accounts were enrolled via a federated bank website 2210c.

Additionally, data about additional cards may be stored for analytics purposes or other purposes 2210d. The consumer may be presented with a confirmation of successful enrollment after the payment account and/or wallet service has been enrolled and/or the staging table entry has been made 2211 (see FIG. 12a for an example card account success enrollment interface).

[0115] FIG. 22b is a block diagram showing an exemplary process of enrolling card accounts in a wallet account. In some embodiments, the user 2212 may request an issuer page, website, or application 2213 via their electronic device 2225. The device may send an issuer page request 2214 to the issuer's server 2215, which may return the issuer's page, website, and/or application 2216 to the electronic device (see FIGS. 24b and 24g).

[0116] In some embodiments, the user may provide card account selections to add to the wallet 2219 to the electronic device (see FIGS. 24e and 24h-i). The electronic device may send said selections to the issuer server via a request card account info push to a wallet message 2220. In some implementations, the XML-encoded push to wallet message 2220 may take a form similar to the following:

```

POST /pushtowalletrequest.php HTTP/1.1
Host: www.merchant.com
Content-Type: Application/XML
Content-Length: 667
<?XML version = "1.0" encoding = "UTF-8"?>
<push_request>
  <timestamp>2013-02-22 15:22:43</timestamp>
  <auth_params>
    <digital_cert>http://cert_request_server.com/
cert_request_1236789.cert</digital_cert>
  </auth_params>
  <wallet_params>
    <wallet_name>V.me Wallet</wallet_name>
    <wallet_address>http://v.me/</wallet_address>
    <wallet_type>online</wallet_type>
  </wallet_params>
  <wallet_auth_params>
    <wallet_auth_username>myusername
    </wallet_auth_username>
    <wallet_auth_password>mypassword
    </wallet_auth_password>
    <wallet_auth_ID>098382093YUI</wallet_auth_ID>
  </wallet_auth_params>
  <user_details>
    <user_name>JDoe@gmail.com</user_name>
    <password>Tomcat123</password>
    <user_ID>12348901</user_ID>
  </user_details>
  <card_details>
    <card_number>111122233334444</card_number>
    <card_security>123</card_security>
    <card_ID>135792</card_ID>
    <card_address>789 Main Street, AnyCity, AnyState
12345</card_address>
    <card_expire>2025-01-01</card_expire>
  </card_details>
  <card_details>
    <card_number>9999888877776666</card_number>
    <card_security>456</card_security>
    <card_ID>246801</card_ID>
    <card_address>222 Example Street, AnyCity, AnyState
55555</card_address>
    <card_expire>2025-05-10</card_expire>
  </card_details>
</push_request>

```

[0117] In some embodiments, the message may contain card selection information, user account information for the issuer, user account information for the wallet service, and/or the like. The issuer server may then push the selection information via a new card account add request **2223** to the wallet server **2217**. In some implementations, the XML-encoded request **2223** may take a form similar to the following:

```
POST /newcardrequest.php HTTP/1.1
Host: www.merchant.com
Content-Type: Application/XML
Content-Length: 667
<?XML version = "1.0" encoding = "UTF-8"?>
<new_card_request>
  <timestamp>2013-02-22 15:22:43</timestamp>
  <auth_params>
    <passcode>my_password</passcode>
  </auth_params>
  <user_details>
    <user_name>JDoe@gmail.com</account_name>
    <password>Tomcat123</password>
    <user_ID>12348901</user_ID>
  </user_details>
  <card_details>
    <card_number>1111222233334444</card_number>
    <card_security>123</card_security>
    <card_ID>135792</card_ID>
    <card_address>789 Main Street, AnyCity, AnyState
12345</card_address>
    <card_expire>2025-01-01</card_expire>
  </card_details>
  <card_details>
    <card_number>9999888877776666</card_number>
    <card_security>456</card_security>
    <card_ID>246801</card_ID>
    <card_address>222 Example Street, AnyCity, AnyState
55555</card_address>
    <card_expire>2025-05-10</card_expire>
  </card_details>
</new_card_request>
```

[0118] The wallet server may then use any information received from the issuer server to modify the user's wallet account via a MySQL database command similar to the following:

```
INSERT INTO user_cards (number, security_code, ID, address, expire)
VALUES (card_number, card_security, card_ID, card_address,
card_expire);
```

[0119] In some implementations, the electronic device may instead send the user selections to the wallet server via a request for a pull for card account information from the issuer **2221** that is sent by a wallet overlay **2218** (see FIG. **24c-d**). In some implementations, the XML-encoded pull request **2221** may resemble the following:

```
POST /pullrequest.php HTTP/1.1
Host: www.merchant.com
Content-Type: Application/XML
Content-Length: 667
<?XML version = "1.0" encoding = "UTF-8"?>
<pull_request>
  <timestamp>2013-02-22 15:22:43</timestamp>
  <auth_params>
    <auth_code>123ABC098</auth_code>
```

-continued

```
</auth_params>
<issuer_params>
  <issuer_name>Bank of America</issuer_name>
  <issuer_type>bank</issuer_type>
  <issuer_address>http://bankofamerica.com/accts
  </issuer_address>
</issuer_params>
<issuer_auth_params>
  <issuer_auth_username>my_username1
  </issuer_auth_username>
  <issuer_auth_password>my_password1
  </issuer_auth_password>
  <issuer_auth_ID>12678HJK</issuer_auth_ID>
</issuer_auth_params>
<user_details>
  <user_name>JDoe@gmail.com</user_name>
  <password>Tomcat123</password>
  <user_ID>12348901</user_ID>
</user_details>
<card_details>
  <card_number>1111222233334444</card_number>
</card_details>
<card_details>
  <card_number>9999888877776666</card_number>
</card_details>
</pull_request>
```

[0120] The wallet server may use any identifying information (such as the user's account number with the issuer, the user's card number(s), and/or the like) provided in the request for card account information to create a new request **2222** to the issuer server. The wallet server may request any information necessary to link the card account to the wallet service, including permission from the issuer, more information about the card account not provided by the user (e.g., a card account ID, and/or the like). The issuer server may, after receiving such a request, send a new card account add request **2223** which may include all information requested by the wallet server.

[0121] FIG. **23a** is an exemplary virtual wallet and card enrollment logic and data flow. In some embodiments, the user accesses a wallet URL using a mobile device **2303**. In other embodiments, the wallet URL is accessed from the user's computer, the user's issuer web site, and/or the like. In some embodiments, the wallet may be accessed either via a wallet-implemented JavaScript overlay, via the issuer's site directly, and/or the like. If the wallet is accessed via the overlay, the wallet may pull card account information from the issuer. If the wallet is accessed via the issuer's website, the wallet may push the card account data to the wallet server. If the user is already logged into their wallet account **2304**, as indicated in one embodiment by a cookie on the user's computer, the user is directed to a wallet display including an "Add Card" button **2314**. If the user is not logged into a virtual wallet account, the WSEP may then prompt the user to indicate if they already have a virtual wallet account **2305** and if so, prompt the user to log into their account **2313**. In some embodiments, the consumer may be asked to consent to the retrieval of pre-fill data from a payment account issuer **2306**. If the user does not consent, they may be directed to a wallet enrollment form with no pre-fill data pre-populated. Should the user consent to the issuer sharing pre-fill data, the wallet server **2301** may transmit a request to the issuer for data **2307** and the issuer server may receive **2308** and process the request. If the user account and/or pre-fill data is found by the issuer **2309**, the data may be transmitted to the wallet server **2310** for use in pre-filling/pre-populating fields in the wallet

enrollment form **2311**. If no pre-fill data is found by the issuer, the user is directed to the wallet enrollment form **2311**. In some embodiments, the issuer is a bank. In other embodiments, the issuer is a rewards account provider. In still other embodiments, the issuer is a non-financial company and/or an individual (as in peer-to-peer enrollment). The consumer may then complete any data required by the wallet enrollment form **2311**. In some embodiments, the pre-fill data is filled into the enrollment form for the user. Example data is user name, user billing address, user account identifier, mother's maiden name, security question and answer, and/or the like. In other embodiments, some fields of the enrollment form **2311** may be hidden if pre-fill data is available for those fields. Upon completion of the enrollment form, the user is enrolled into the wallet **2312** and logged into the wallet with an option to add accounts **2314**. The logic and data flow continues in FIG. **23b**.

[0122] FIG. **23b** is a continuation of an exemplary virtual wallet and card enrollment logic and data flow. In some embodiments, the user clicks a button to "Add Card" to their wallet. In other embodiments, no user interaction is required. A user may then be prompted to enter their card number, account number, PAN number, and/or similar **2316**. In some embodiments, a user will be asked to consent to the retrieval of the account information from an account issuer **2317**. In other embodiments, this user consent may be assumed. If the user does not consent to the retrieval of account information from the issuer **2317**, then the user may be prompted to input additional information about the payment account to facilitate enrollment of the account in the wallet **2320**. If the user does consent to the retrieval of card information from the issuer **2317**, account data such as billing address, user name, credit history, and/or the like is retrieved from the issuer **2318** and processed by the wallet server **2319**. In some embodiments, the WSEP may generate a request for a card image **2321**. The request may be sent to a card image server **2329**. If the card image is available **2322**, the card image server **2329** may designate a template image for the card **2324**. Alternatively, if no image is available a default template image may be used **2323**. In some embodiments, the card image server may create an "on the fly" image to represent the card and overlay that image with appropriate consumer specific data such as name, photo, and/or the like **2323a**. In some embodiments, confidential data such as PAN number, account number and/or the like may be obscured from the overlaid data using a tool such as ImageMagik. In doing so, the card image server may protect confidential consumer information. The image server may then create a card image response to send the card image data and/or card image(s) to the requesting service. In some embodiments, the card image response is substantially in the form described in **2109a**. In some embodiments, the WSEP may then display a payment account and/or card enrollment form with the retrieved card image and any data retrieved from the account issuer pre-filled **2325**. The consumer may then complete any remaining information required by the enrollment form and save the account in their virtual wallet **2326**. In some embodiments, the WSEP will then register or associate the payment account with the consumer's virtual wallet **2327** and prompt the user that the account has been linked to their wallet **2328** (see FIG. **12a** for an example card account success enrollment interface).

[0123] FIGS. **24a-j** illustrate alternate embodiments of wallet and card enrollment via the WSEP. In some embodiments, the wallet and card enrollment may occur on a normal

web interface, a mobile web interface, a voice-controlled interface, and/or other interfaces. FIG. **24a** illustrates alternate embodiments of linking **2401a** website for an issuer, merchant, and/or a like web service to the wallet service. FIGS. **24b-d** illustrate example embodiments of providing users a method of enrolling in a wallet program through an issuer's website. For example, in some embodiments, the user may access an introductory screen **2402** which may provide detail on the wallet service, and the user may be presented a number of options in enrolling in the wallet service **2403** (including an express enrollment or card addition option, a standard enrollment or card addition option, and/or the like). The user may then be presented with wallet-implemented overlays **2404** in which to enter wallet account information (either for a new or existing account), wallet-implemented overlays **2404** being alternative overlays to wallet overlay **2003**. The user may use card selection overlays **2405** to choose cards to associate with the wallet account, and may confirm the selection. The overlays **2404** and **2405** may send all collected information directly to the wallet server.

[0124] FIGS. **24e-f** illustrate further alternate example embodiments of providing users a way of enrolling in the wallet program. For example, the user may, while exclusively using the issuer's website, enter card selections **2406** (alternatively, the user may do so similar to the embodiment provided in FIGS. **20c-d**). While remaining on the issuer's website, the user may also provide information for logging into, or signing up for, a wallet account **2407**.

[0125] FIGS. **24g-j** illustrate further alternate example embodiments of the interface in FIGS. **20b-d**. In some embodiments, the wallet login and sign-up options **2015** and **2017** may resemble **2408** and **2409**, respectively. The sign-up form for a wallet account may be included on the main page as shown at **2410**. Similar to **2012**, the user may be presented with a set of available cards **2411** which may be selectable for a wallet account. The user may also be able to specify which card to set as a default card for the wallet. In addition to the information collected in FIGS. **20b-d**, the issuer may request that the user provide a set of security questions and answers **2412**, as well as security codes **2413**. The issuer may provide the user with a confirmation screen **2414** once the process has been completed.

WSEP Controller

[0126] FIG. **25** shows a block diagram illustrating embodiments of a WSEP controller. In this embodiment, the WSEP controller **2501** may serve to aggregate, process, store, search, serve, identify, instruct, generate, match, and/or facilitate interactions with a computer through various bi-directional linking technologies, and/or other related data.

[0127] Typically, users, 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 **2503** 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 **2529** (e.g., registers, cache memory, random access memory, etc.). Such communicative instructions may be stored and/or transmitted in batches (e.g., batches of instruc-

tions) 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.

[0128] In one embodiment, the WSEP controller **2501** may be connected to and/or communicate with entities such as, but not limited to: one or more users from user input devices **2511**; peripheral devices **2512**; an optional cryptographic processor device **2528**; and/or a communications network **2513**.

[0129] 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.

[0130] The WSEP controller **2501** may be based on computer systems that may comprise, but are not limited to, components such as: a computer systemization **2502** connected to memory **2529**.

Computer Systemization

[0131] A computer systemization **2502** may comprise a clock **2530**, central processing unit (“CPU(s)” and/or “processor(s)” (these terms are used interchangeable throughout the disclosure unless noted to the contrary)) **2503**, a memory **2529** (e.g., a read only memory (ROM) **2506**, a random access memory (RAM) **2505**, etc.), and/or an interface bus **2507**, and most frequently, although not necessarily, are all interconnected and/or communicating through a system bus

2504 on one or more (mother)board(s) **2502** 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 **2586**; e.g., optionally the power source may be internal. Optionally, a cryptographic processor **2526** and/or transceivers (e.g., ICs) **2574** 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 **2512** via the interface bus I/O. In turn, the transceivers may be connected to antenna(s) **2575**, 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 WSEP controller to determine its location)); Broadcom BCM4329FKUBG transceiver chip (e.g., providing 802.11n, Bluetooth 2.1+EDR, FM, etc.); a Broadcom BCM4750UB8 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.

[0132] 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 **2529** 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 WSEP controller and beyond through various interfaces. Should processing requirements dictate a greater amount speed and/or capacity, distributed processors (e.g., Distributed WSEP), 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.

[0133] Depending on the particular implementation, features of the WSEP 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 WSEP, 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 WSEP 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 WSEP may be implemented with embedded components that are configured and used to achieve a variety of features or signal processing.

[0134] 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, WSEP 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 WSEP features. A hierarchy of programmable interconnects allow logic blocks to be interconnected as needed by the WSEP 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 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 WSEP may be developed on regular FPGAs and then migrated into a fixed version that more resembles ASIC implementations. Alternate or coordinating implementations may migrate WSEP controller features to a final ASIC instead of or in addition to FPGAs. Depending on the implementation all of the aforementioned embedded components and micro-processors may be considered the "CPU" and/or "processor" for the WSEP.

Power Source

[0135] The power source **2586** 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 **2586** is connected to at least one of the interconnected subsequent components of the WSEP thereby providing an electric current to all subsequent components. In one example, the power source **2586** is connected to the system bus component **2504**. In an alternative embodiment, an outside power source **2586** is provided through a connection across the I/O **2508** 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

[0136] Interface bus(es) **2507** 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) **2508**, storage interfaces **2509**, network interfaces **2510**, and/or the like. Optionally, cryptographic processor interfaces **2527** 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.

[0137] Storage interfaces **2509** may accept, communicate, and/or connect to a number of storage devices such as, but not limited to: storage devices **2514**, 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.

[0138] Network interfaces **2510** may accept, communicate, and/or connect to a communications network **2513**. Through a communications network **2513**, the WSEP controller is accessible through remote clients **2533b** (e.g., computers with web browsers) by users **2533a**. 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 WSEP), architectures may similarly be employed to pool, load balance, and/or otherwise increase the communicative bandwidth required by the WSEP 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 **2510** may be used to engage with various communications network types **2513**. For example, multiple network interfaces may be employed to allow for the communication over broadcast, multicast, and/or unicast networks.

[**0139**] Input Output interfaces (I/O) **2508** may accept, communicate, and/or connect to user input devices **2511**, peripheral devices **2512**, cryptographic processor devices **2528**, 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.).

[**0140**] User input devices **2511** often are a type of peripheral device **512** (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.

[**0141**] Peripheral devices **2512** 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 WSEP 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 **528**), 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).

[**0142**] It should be noted that although user input devices and peripheral devices may be employed, the WSEP 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.

[**0143**] Cryptographic units such as, but not limited to, microcontrollers, processors **2526**, interfaces **2527**, and/or devices **2528** may be attached, and/or communicate with the WSEP 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: 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

[**0144**] Generally, any mechanization and/or embodiment allowing a processor to affect the storage and/or retrieval of information is regarded as memory **2529**. 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 WSEP controller and/or a computer systemization may employ various forms of memory **2529**. 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 **2529** will include ROM **2506**, RAM **2505**, and a storage device **2514**. A storage device **2514** 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., Blu-ray, 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

[**0145**] The memory **2529** may contain a collection of program and/or database components and/or data such as, but not limited to: operating system component(s) **2515** (operating system); information server component(s) **2516** (information server); user interface component(s) **2517** (user interface); Web browser component(s) **2518** (Web browser); database(s)

2519; mail server component(s) **2521**; mail client component (s) **2522**; cryptographic server component(s) **2520** (cryptographic server); the WSEP component(s) **2535**; the ACM component **1441**; 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 **2514**, 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

[0146] The operating system component **2515** is an executable program component facilitating the operation of the WSEP 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 Nan 9; Be OS; Unix and Unix-like system distributions (such as AT&T's UNIX; Berkeley 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/Millennium/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 WSEP controller to communicate with other entities through a communications network **2513**. Various communication protocols may be used by the WSEP 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

[0147] An information server component **2516** 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 WSEP 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 WSEP database **2519**, operating systems, other program components, user interfaces, Web browsers, and/or the like.

[0148] Access to the WSEP 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 WSEP. 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 WSEP 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.

[0149] 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

[0150] 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/Millennium/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.

[0151] A user interface component **2517** 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

[0152] A Web browser component **2518** 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 128 bit (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 WSEP enabled nodes. The combined application may be nugatory on systems employing standard Web browsers.

Mail Server

[0153] A mail server component **2521** is a stored program component that is executed by a CPU **2503**. 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 WSEP.

[0154] Access to the WSEP mail may be achieved through a number of APIs offered by the individual Web server components and/or the operating system.

[0155] 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

[0156] A mail client component **2522** is a stored program component that is executed by a CPU **2503**. 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

[0157] A cryptographic server component **2520** is a stored program component that is executed by a CPU **2503**, cryptographic processor **2526**, cryptographic processor interface **2527**, cryptographic processor device **2528**, 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 Pro-

tection (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 WSEP 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 WSEP component to engage in secure transactions if so desired. The cryptographic component facilitates the secure accessing of resources on the WSEP 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 WSEP Database

[0158] The WSEP database component **2519** 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.

[0159] Alternatively, the WSEP 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 WSEP database is implemented as a data-structure, the use of the WSEP database **2519** may be integrated into another component such as the WSEP component **2535**. 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.

[0160] In one embodiment, the database component **2519** includes several tables **2519a-q**. A user accounts **2519a** includes fields such as, but not limited to: a user ID, merchant identifier, name, home address, work address, telephone number, email, merchant ID and/or the like. The user table may support and/or track multiple entity accounts on a WSEP. A merchant/service provider table **2519b** includes fields such as, but not limited to: merchant ID, user ID, merchant name, merchant location, merchant address, merchant category code, merchant api key, loyalty program ID and/or the like. A customer profile table **2519c** includes fields such as, but not limited to: customer ID, user ID, merchant ID, payment card ID, preferred payment type, and/or the like. A permissions table **2519d** includes fields such as, but not limited to: customer ID, transaction execution authorization status, confirmation authorization status, billing authorization status, subscription payment authorization status, and/or the like. A payment card table **2519e** includes fields such as, but not limited to: payment_card_id, user_id, identifier, brand, expiration date, spending limit, billing address, issuer, name, nick name, loyalty program ID, and/or the like. A billing agreement table **2519f** includes fields such as, but not limited to: customer_id, billing_id, billing_date, billing_amount_limit, confirmation_requirement, authentication_level, billing_authorization_status, and/or the like. A redemption table **2519g** includes fields such as, but not limited to: customer_id, loyalty_program_id, coupon_id, redemption_date, redemption_time, redemption_amount, redemption_type, transaction_id, and/or the like. A wallet table **2519h** includes fields such as, but not limited to: wallet_id, user_id, prefill_id, billing_address, last_used_date, last_transaction_id, and/or the like. A card_templates table **2519i** includes fields such as, but not limited to: card_template_id, payment_card_id, card_type, file_card_front_location, file_card_back_location, card_front_template_location, card_back_template_location, template_type, and/or the like. A wallet accounts table **2519j** includes fields such as, but not limited to: wallet_account_id, wallet_id, account_number, issuer_name, issuer_id, issuer_routing_number, and/or the like. An issuers table **2519k** includes fields such as, but not limited to: issuer_id, payment_card_id, user_id, issuer_name, issuer_server_url, and/or the like. An analytics table **2519l** includes fields such as, but not

limited to: customer_id, merchant_id, transaction_volume, transaction_amount, transaction_type, transaction_id and/or the like. A staging table **2519m** includes fields such as, but not limited to: staging_table_id, BID, CID, account_number, user_id, customer_id, merchant_id, issuer_identifier, and/or the like. A payments table **2519n** includes fields such as, but not limited to: billing_id, billing_date, billing_amount, payment_card_id, authentication_level, and/or the like. A prefills table **2519o** includes fields such as, but not limited to: prefills_id, user_id, wallet_id, wallet_account_id, permissions, access rules, prefill_data, and/or the like. A transaction table **2519p** includes fields such as, but not limited to: transaction_id, merchant_id, user_id, session_id, date, time, item_model, manufacturer, price, item_id, and/or the like. A contracts table **1419q** includes fields such as, but not limited to: contract_id, contract_type, merchant_id, user_id, contract_expiration_date, total_authorized_charges, monthly_authorized_charges, and/or the like.

[0161] In one embodiment, the WSEP database may interact with other database systems. For example, employing a distributed database system, queries and data access by search WSEP component may treat the combination of the WSEP database, an integrated data security layer database as a single database entity.

[0162] In one embodiment, user programs may contain various user interface primitives, which may serve to update the WSEP. Also, various accounts may require custom database tables depending upon the environments and the types of clients the WSEP 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 **2519a-q**. The WSEP may be configured to keep track of various settings, inputs, and parameters via database controllers.

[0163] The WSEP database may communicate to and/or with other components in a component collection, including itself, and/or facilities of the like. Most frequently, the WSEP database communicates with the WSEP component, other program components, and/or the like. The database may contain, retain, and provide information regarding other nodes and data.

The WSEPs

[0164] The WSEP component **2535** is a stored program component that is executed by a CPU. In one embodiment, the WSEP component incorporates any and/or all combinations of the aspects of the WSEP that was discussed in the previous figures. As such, the WSEP affects accessing, obtaining and the provision of information, services, transactions, and/or the like across various communications networks.

[0165] The WSEP transforms inputs such as user accounts **2519a**, issuers **2519k**, prefills **2519p**, payment cards **2519e** and others using Prefill components **2542** and Wallet enrollment component **2543** into Wallet **2519h**, Wallet accounts **2519i** and Prefills **2519o** outputs.

[0166] The WSEP 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 WSEP server employs a cryptographic server to encrypt and decrypt communications. The WSEP component may communicate to and/or with other components in a component collection, including itself, and/or facilities of the like. Most frequently, the WSEP component communicates with the WSEP database, operating systems, other program components, and/or the like. The WSEP may contain, communicate, generate, obtain, and/or provide program component, system, user, and/or data communications, requests, and/or responses.

Distributed WSEPs

[0167] The structure and/or operation of any of the WSEP 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.

[0168] 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.

[0169] The configuration of the WSEP 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.

[0170] 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 component 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.

[0171] For example, a grammar may be arranged to recognize the tokens of an HTTP post command, e.g.:

[0172] w3c-post http:// . . . Value1

[0173] 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.

[0174] For example, in some implementations, the WSEP 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
?>
```

[0175] 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>

[0176] and other parser implementations:

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

[0177] all of which are hereby expressly incorporated by reference.

[0178] In order to address various issues and advance the art, the entirety of this application for WALLET SERVICE ENROLLMENT PLATFORM APPARATUSES, METHODS AND SYSTEMS (including the Cover Page, Title, Headings, Field, Background, Summary, Brief Description of the Drawings, Detailed Description, Claims, Abstract, Figures, Appendices, and 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 embodi-

ments 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 WSEP 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 WSEP, may be implemented that enable a great deal of flexibility and customization. For example, aspects of the WSEP may be adapted for transaction liability determination. While various embodiments and discussions of the WSEP have been directed to bi-direction federation of credentials and other information, 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.

What is claimed is:

1. A wallet service enrollment processor implemented method comprising:

receiving via a processor an indication of a consumer's desire to enroll a payment account in a virtual wallet and a payment account identifier;

determining a payment account issuer whereby the payment account issuer has previously created a payment account corresponding to the payment account identifier;

transmitting to the payment account issuer the payment account identifier; and

receiving from the payment account issuer additional account information, whereby the additional account information has been previously provided to the payment account issuer.

2. The method of claim 1 further comprising pre-populating the additional account information in a wallet payment account enrollment form.

3. The method of claim 1 further comprising automatically enrolling the payment account corresponding to the payment account identifier in a virtual wallet without requiring further user input.

4. The method of claim 1 whereby the payment account identifier is a payment account number.

5. The method of claim 1 whereby determining a payment account issuer further comprises:

submitting the payment account identifier to a server; and receiving an identifier of the issuer that issued the payment account corresponding to the payment account identifier.

6. The method of claim 1 whereby the transmitting to the issuer the payment account identifier is routed through a secure connection.

7. A wallet service enrollment processor-implemented system, comprising:

means to receive via a processor an indication of a consumer's desire to enroll a payment account in a virtual wallet and a payment account identifier;

means to determine a payment account issuer whereby the payment account issuer has previously created a payment account corresponding to the payment account identifier;

means to transmit to the payment account issuer the payment account identifier; and

means to receive from the payment account issuer additional account information, whereby the additional account information has been previously provided to the payment account issuer.

8. The system of claim 7 further comprising means to pre-populate the additional account information in a wallet payment account enrollment form.

9. The system of claim 7 further comprising means to automatically enroll the payment account corresponding to the payment account identifier in a virtual wallet without requiring further user input.

10. The system of claim 7 whereby the payment account identifier is a payment account number.

11. The system of claim 7 whereby the means to determine a payment account issuer further comprises:

means to submit the payment account identifier to a server; and

means to receive an identifier of the issuer that issued the payment account corresponding to the payment account identifier.

12. The system of claim 7 whereby the means to transmit to the issuer the payment account identifier is routed through a secure connection.

13. A wallet service enrollment apparatus, comprising:

a memory;

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

receive via a processor an indication of a consumer's desire to enroll a payment account in a virtual wallet and a payment account identifier;
 determine a payment account issuer whereby the payment account issuer has previously created a payment account corresponding to the payment account identifier;
 transmit to the payment account issuer the payment account identifier; and
 receive from the payment account issuer additional account information, whereby the additional account information has been previously provided to the payment account issuer.

14. The apparatus of claim **13** further comprising instructions to pre-populate the additional account information in a wallet payment account enrollment form.

15. The apparatus of claim **13** further comprising instructions to automatically enroll the payment account corresponding to the payment account identifier in a virtual wallet without requiring further user input.

16. The apparatus of claim **13** whereby the payment account identifier is a payment account number.

17. The apparatus of claim **13** whereby the instructions to determine a payment account issuer further comprises instructions to:
 submit the payment account identifier to a server; and
 receive an identifier of the issuer that issued the payment account corresponding to the payment account identifier.

18. The apparatus of claim **13** whereby the instructions to transmit to the issuer the payment account identifier is routed through a secure connection.

19. A non-transitory medium storing processor-issuable wallet service enrollment instructions to:
 receive via a processor an indication of a consumer's desire to enroll a payment account in a virtual wallet and a payment account identifier;
 determine a payment account issuer whereby the payment account issuer has previously created a payment account corresponding to the payment account identifier;

transmit to the payment account issuer the payment account identifier; and
 receive from the payment account issuer additional account information, whereby the additional account information has been previously provided to the payment account issuer.

20. The medium of claim **19** further comprising instructions to pre-populate the additional account information in a wallet payment account enrollment form.

21. The medium of claim **19** further comprising instructions to automatically enroll the payment account corresponding to the payment account identifier in a virtual wallet without requiring further user input.

22. The medium of claim **19** whereby the payment account identifier is a payment account number.

23. The medium of claim **19** whereby the instructions to determine a payment account issuer further comprises:
 means to submit the payment account identifier to a server;
 and
 means to receive an identifier of the issuer that issued the payment account corresponding to the payment account identifier.

24. The medium of claim **19** whereby the instructions to transmit to the issuer the payment account identifier is routed through a secure connection.

25. A wallet service enrollment processor implemented method comprising:
 receiving via a processor an indication of a consumer's desire to enroll at least one payment device associated with a payment account issuer in a virtual wallet account;
 receiving at least one selection of at least one payment device associated with the payment account issuer and virtual wallet account information;
 retrieving at least one payment account identifier associated with the at least one payment device; and
 transmitting the at least one payment account identifier and the virtual wallet account information to a virtual wallet provider associated with the virtual wallet account.

* * * * *