SYSTEM AND METHOD FOR TOKEN DOMAIN CONTROL

CROSS REFERENCE TO RELATED APPLICATIONS

[0001] This application is a non-provisional application of and claims the benefit of the filing date of U.S. Provisional Application No. 61/988,815, filed on May 5, 2014, which is herein incorporated by reference in its entirety for all purposes.

BACKGROUND

[0002] Current implementations of token systems could be improved. For example, in some cases, there may be an insufficient amount of possible unique token values. This may be an issue when multiple tokens are issued for one user account. A unique token may be issued for each domain in which the payment account is used, and tokens may be regularly refreshed (e.g., one-time use tokens). Current token formats only allow a certain amount of variation, so eventually all possible unique values may be exhausted.

[0003] Another issue is that users have historically been recognized by various entities through their primary account numbers. User activity tracking, loyalty programs, activity analysis, and other programs have been based on PAN usage. When the PAN is replaced with multiple tokens, each token may appear to be a different account to certain entities (e.g., acquirers), and thus these entities may not be able to determine the total activity for a given account. Accordingly, using tokens instead of primary account numbers can be disruptive.

[0004] Embodiments of the invention address these and other problems, individually and collectively.

SUMMARY

[0005] One embodiment of the invention is directed to a method. The method comprises receiving, by a data processor in a token service computer, a first token request from a first token requestor computer, the first token request including a

value credential and a first domain identifier. The method also comprises identifying a value token associated with the value credential, generating a first token code associated with the value token, assigning the value token and the first token code to the first domain identifier, and transmitting the value token and the first token code to the first token requestor computer. The first token requestor subsequently uses the value token for an interaction, and the first token requestor's subsequent use of the value token is valid if the value token is accompanied by the first token code.

[0006] In another embodiment, the method further comprises receiving a second token request from a second token requestor computer, the second token request including the value credential and a second domain identifier. The method also comprises identifying the value token associated with the value credential generating a second token code associated with the value token assigning the value token and the second token code to the second domain identifier, and transmitting the value token and the second token code to the second token requestor computer. The second token requestor subsequently uses the value token for an interaction, and the second token requestor's subsequent use of the value token is valid if the value token is accompanied by the second token code.

[0007] Another embodiment of the invention is directed to a token service computer configured to perform the above-described method.

[0008] Another embodiment of the invention is directed to a method comprising receiving, by a token service system, a first authorization request message including a value token, a first token code, and a first domain identifier. The method also comprises determining that the first token code is associated with the value token, determining that the value token and the first token code are assigned to the first domain identifier, identifying a value credential associated with the value token, adding the value credential to the first authorization request message, sending the first authorization request message to an authorizing entity computer, receiving a first authorization response message including the value credential from the authorizing entity computer, replacing the value credential with

the value token and the first token code in the first authorization response message, and forwarding the first authorization response message.

[0009] In another embodiment, the method further comprises receiving, by the token service system, a second authorization request message including a value token, a second token code, and a second domain identifier. The method also comprises determining that the second token code is associated with the value token, determining that the value token and the second token code are assigned to the second domain identifier, identifying a value credential associated with the value token, adding the value credential to the second authorization request message, sending the second authorization request message to the authorizing entity computer, receiving a second authorization response message including the value credential from the authorizing entity computer, replacing the value credential with the value token and the second token code in the authorization response message, and forwarding the authorization response message.

[0010] Another embodiment of the invention is directed to a token service system configured to perform the above-described method.

[0011] Further details regarding embodiments of the invention can be found in the Detailed Description and the Figures.

BRIEF DESCRIPTION OF THE DRAWINGS

[0012] **FIG. 1** shows a block diagram of a system according to an embodiment of the invention.

[0013] **FIG. 2** shows a block diagram of a token vault computer according to an embodiment of the invention.

[0014] **FIG. 3** shows a block diagram of an exemplary mobile device according to an embodiment of the invention.

[0015] **FIG. 4** shows a block diagram of a merchant computer according to an embodiment of the invention.

[0016] **FIG. 5** shows a block diagram illustrating the use of token codes for various domains, in accordance with some embodiments of the invention.

[0017] **FIG. 6** shows a flow diagram illustrating a method according to embodiments of the invention.

[0018] **FIG. 7** shows a flow diagram illustrating a method for sharing a token according to embodiments of the invention.

[0019] **FIG. 8** shows a block diagram of a computer apparatus according to an embodiment of the invention.

DETAILED DESCRIPTION

[0020] Embodiments of the present invention are directed to providing, in addition to a value token, a token code to a token requestor. The token code can serve as a shared secret between a token holder and a token vault. Both the value token and the token code can be provided together during a transaction, and the token code can be used to authenticate that the token holder is authorized to use the value token.

[0021] In some embodiments, one value token can be provided for use in multiple domains. Each domain may receive a different token code along with the value token. If the value token is received for a transaction through a certain domain, it may only be considered valid if accompanied by the appropriate token code for that domain.

[0022] As discussed above, using value tokens can lead to a token numeric issue. For example, a typical PAN (payment account number) is composed of 16 digits. If each digit can range from 0-9, than there are 10^{16} possible unique PANs. However, certain digits within a PAN are sometimes occupied by various types of identification information. For example, the first six digits may be used to indicate a BIN (bank identification number). Other digits may be used for other purposes, such as the final digit being used as a checksum. In some embodiments, the number of digits used to actually identify a specific account may be as low as (or lower than) 9

digits. In the example of 9 digits, there are now only 10^9, or one billion, possible unique values. Considering that there may be many millions of account holders, that there may be multiple tokens associated with each account at any given time, and that each token might be refreshed regularly (e.g., once a week, after each transaction, etc.), it can be seen that one billion unique values may not be enough to accommodate tokenization applications.

[0023] Embodiments of the invention solve this issue by providing one value token for use in multiple token domains, along with a different token code for each domain. Instead of increasing token proliferation (e.g., by issuing a different value token to every domain, by constantly refreshing value tokens, etc.), a single value token can be issued to each domain for a certain payment account. Current token formats may provide enough possible token values to accommodate a single unique value token per payment account. For example, one billion values may be sufficient for each account holder to have a unique token value. The token code allows the single value token to be widely distributed without sacrificing security, as a token code may be required when using the value token, and token codes may only be provided to (and valid for) domains that are authorized to use the value token.

[0024] This can also solve the issue of tracking users. As described above, incorporating token codes allows a payment account to be uniquely represented by a single value token across all domains. Accordingly, entities (e.g., acquirers and merchants) that previously used a PAN to track user activity can now simply use the value token instead (as it may be provided during transactions instead of the PAN). Thus, entities can essentially continue tracking user activity as usual. In some cases, the value token may be formatted similarly to a PAN, so entities may not be able to distinguish between a value token and a PAN, and they may not even notice that anything changed. For example, a value token be a 16 digit number, such as "4146 7238 9345 7895". Acquirers and merchants may use the entire number, or a portion of the number (such as field 2, the last four digits "7895", etc.) to identify an account and/or user. When communicating about the user, entities may refer to the user by the value token "4146 7238 9345 7895". Also, when identifying which user is

participating in a transaction, acquirers and merchants identify a user record based on the value token "4146 7238 9345 7895".

[0025] This token code system may be easily adopted by merchants, acquirers, and other relevant entities. Minimal changes are needed for existing systems (e.g., user activity tracking remains the same, as described above). Token holders may simply store an extra value (the token code), and other entities may simply pass the token code along in authorization request and response messages. In some embodiments, the token code may be able to fit within existing authorization message formats without requiring any changes. Thus, a token system that increases payment security can be introduced with minimal interruptions.

[0026] Prior to discussing specific embodiments of the invention, some terms may be described in detail.

[0027] An "interaction" may include a communication, contact, or exchange between parties, devices, and/or entities. In some embodiments, data can be exchanged in an interaction between two devices (e.g., an interaction between a mobile device and an access device).

[0028] A "value credential" may be information associated with worth. Examples of a value credential include payment credentials, information needed to obtain a promotional offer, etc.

[0029] "Payment credentials" may include any suitable information associated with an account (*e.g.* a payment account and/or payment device associated with the account). Such information may be directly related to the account or may be derived from information related to the account. Examples of account information may include a PAN (primary account number or "account number"), user name, expiration date, CVV (card verification value), dCVV (dynamic card verification value), CVV2 (card verification value 2), CVC3 card verification values, etc. CVV2 is generally understood to be a static verification value associated with a payment device. CVV2 values are generally visible to a user (*e.g.*, a consumer), whereas CVV and dCVV values are typically embedded in memory or authorization request messages and

are not readily known to the user (although they are known to the issuer and payment processors). Payment credentials may be any information that identifies or is associated with a payment account. Payment credentials may be provided in order to make a payment from a payment account. Payment credentials can also include a user name, an expiration date, a gift card number or code, and any other suitable information.

[0030] "Tokenization" is a process by which data is replaced with substitute data. For example, a payment account identifier (e.g., a primary account number (PAN)) may be tokenized by replacing the primary account identifier with a substitute number (e.g. a token) that may be associated with the payment account identifier. Further, tokenization may be applied to any other-information which may be replaced with a substitute value (i.e., token). Tokenization may be used to enhance transaction efficiency, improve transaction security, increase service transparency, or to provide a method for third-party enablement.

[0031] A "value token" may be an identifier for worth. Examples of a value token include payment tokens, a coupon identifier, etc.

[0032] A "payment token" may include an identifier for a payment account that is a substitute for an account identifier, such as a primary account number (PAN). For example, a token may include a series of alphanumeric characters that may be used as a substitute for an original account identifier. For example, a token "4900 0000 0000 0001" may be used in place of a PAN "4147 0900 0000 1234." In some embodiments, a token may be "format preserving" and may have a numeric format that conforms to the account identifiers used in existing transaction processing networks (e.g., ISO 8583 financial transaction message format). In some embodiments, a token may be used in place of a PAN to initiate, authorize, settle or resolve a payment transaction or represent the original credential in other systems where the original credential would typically be provided. In some embodiments, a token value may be generated such that the recovery of the original PAN or other account identifier from the token value may not be computationally derived. Further, in some embodiments, the token format may be configured to allow the entity

receiving the token to identify it as a token and recognize the entity that issued the token.

[0033] A "token code" may be any string of characters that serve as evidence of permission for using a value token (e.g., a payment token). In some embodiments, a token code may be a hexadecimal value. In other embodiments, a token code could be any other alphabetical, numeric, or alphanumeric value. For example, in some embodiments, a token code may be composed of five characters, or any other suitable number of characters. A token code may also be a random or otherwise un-guessable value. A token code may be issued to a holder of a value token, and the token code may be able to validate the value token. The combination of the token code and a value token may constitute a unique combination. A token code may also be associated with a domain ID (e.g., that indicates a token domain), such that a value token is validated by a token code and an associated domain ID. In some embodiments, a token code may be unique for a value token to which it is assigned. However, the token code may not necessarily be globally unique. For example, the same token code could possibly be randomly generated and assigned for a different value token. Thus, running out of unique token codes may not be a concern.

[0034] A "token service system" or "token service computer" can include a system that that services payment tokens. In some embodiments, a token service system can facilitate requesting, determining (e.g., generating) and/or issuing tokens, as well as maintaining an established mapping of tokens to primary account numbers (PANs) in a repository (e.g. token vault). In some embodiments, the token service system may establish a token assurance level for a given token to indicate the confidence level of the token to PAN binding. The token service system may include or be in communication with a token vault where the generated tokens are stored. The token service system may support token processing of payment transactions submitted using tokens by de-tokenizing the token to obtain the actual PAN. In some embodiments, a token service system may include a token service computer alone, or in combination with other computers such as a transaction

processing network computer. Various entities of a tokenization ecosystem may assume the roles of the token service provider. For example, payment networks and issuers or their agents may become the token service provider by implementing the token services according to embodiments of the present invention.

[0035] In addition to generating and processing payment tokens, a token service computer can generate, issue, and process token codes. For example, a token service computer can generate and provide a token code for each distributed payment token. In some embodiments, a single payment token can be distributed to multiple token requestors, but a different token code may be provided to each token requestor for the single payment token. Similarly, when de-tokenizing a payment token, the token service computer can verify that a payment token is accompanied by an appropriate token code and/or domain ID.

[0036] A "token vault" may be an example of a token service computer and can include a repository that maintains established token-to-PAN mappings. According to various embodiments, the token vault may also maintain other attributes of the token requestor that may be determined at the time of registration. The attributes may be used by the token service provider to apply domain restrictions or other controls during transaction processing. In some embodiments, the token vault may be a part of the token service system or the token service provider. Alternatively, the token vault may be a remote repository accessible to the token service provider. Token vaults, due to the sensitive nature of the data mappings that are stored and managed in them, may be protected by strong underlying physical and logical security.

[0037] "Token exchange" or "de-tokenization" can include a process of restoring the data that was substituted during tokenization. For example, a token exchange may include replacing a payment token with a corresponding primary account number (PAN) that was associated with the payment token during tokenization of the PAN. Thus, the de-tokenization may refer to the process of redeeming a token for the associated PAN value based on a token-to-PAN mapping stored, for example, in a token vault. The ability to retrieve a PAN in exchange for

the associated token may be restricted to specifically authorized entities, individuals, applications, or systems. Further, de-tokenization or token exchange may be applied to any other information. In some embodiments, token exchange may be achieved via a transactional message, such as an ISO message, an application programming interface (API), or another type of web interface (e.g., web request).

[0038] A "token requestor" include an entity that requests a token. The token requestor may initiate a request that a primary account number (PAN) be tokenized by submitting a token request message to the token service provider. According to various embodiments, a token requestor may no longer need to store a PAN associated with a token once the requestor has received a requested token. The requestor may be an application, a device, a process, or a system that is configured to perform actions associated with tokens. A token requestor can request registration with a network token system, request token generation, token activation, token de-activation, token exchange, token lifecycle management related processes, and/or any other token related processes. A requestor may interface with a network token system through any suitable communication networks and/or protocols (e.g., using HTTPS, SOAP and/or an XML interface among others). Some non-limiting examples of token requestors may include, for example, communication devices (e.g., mobile phones and computers) operated by users, card-on-file merchants, acquirers, acquirer processors, and payment gateways acting on behalf of merchants, payment enablers (e.g., original equipment manufacturers, mobile network operators, etc.), digital wallet providers, issuers, third party wallet providers, and/or transaction processing networks. In some embodiments, a token requestor can request tokens for multiple domains and/or channels. A token requestor may be registered and identified uniquely by the token service provider within the tokenization ecosystem. During token requestor registration, the token service provider may formally process the token requestor's application to participate in the token service system. The token service provider may collect information pertaining to the nature of the requestor and the relevant use of tokens to validate and formally approve the token requestor and establish appropriate domain restriction controls. Successfully registered token requestors may be assigned a domain ID that may

also be entered and maintained within the token vault. Token requestors may be revoked or assigned new domain ID. This information may be subject to reporting and audit by the token service provider.

[0039] A "token domain" may indicate an area and/or circumstance in which a token can be used. Examples of the token domain may include, but are not limited to, payment channels (e.g., e-commerce, physical point of sale, etc.), POS entry modes (e.g., contactless, magnetic stripe, etc.), and merchant identifiers to uniquely identify where the token can be used. A set of parameters (i.e. token domain restriction controls) may be established as part of token issuance by the token service provider that may allow for enforcing appropriate usage of the token in payment transactions. For example, the token domain restriction controls may restrict the use of the token with particular presentment modes, such as contactless or e-commerce presentment modes. In some embodiments, the token domain restriction controls may restrict the use of the token at a particular merchant that can be uniquely identified. Some exemplary token domain restriction controls may require the verification of the presence of a token cryptogram that is unique to a given transaction. In some embodiments, a token domain can be associated with a token requestor. For example, a token requestor may request tokens for a certain domain, and both the token requestor and domain may be identified by a certain domain ID.

[0040] A "domain identifier (ID)" may include an identifier for a token domain and/or a token requestor. It may include any characters, numerals, or other identifiers associated with an entity associated with a network token system. In some embodiments, a unique domain ID may be assigned for each domain associated with the same token requestor. For example, a domain ID can identify a pairing of a token requestor (e.g., a mobile device, a merchant, a mobile wallet provider, etc.) with a token domain (e.g., e-commerce, contactless, etc.). A domain ID may include any format or type of information. For example, in one embodiment, the domain ID may include an alphanumerical value such as a ten digit or an eleven digit letter and/or number (e.g., 4678012345). In some embodiments, a domain ID

may include a code for a token service provider (e.g., first 3 digits) such as the network token system and the remaining digits may be assigned by the token service provider for each requesting entity (e.g., mobile wallet provider) and the token domain (e.g., contactless, e-commerce, etc.).

[0041] "Token expiry date" may refer to the expiration date/time of the token. The token expiry date may be passed among the entities of the tokenization ecosystem during transaction processing to ensure interoperability. The token expiration date may be a numeric value (e.g. a 4-digit numeric value).

[0042] A "mobile device" may comprise any suitable electronic device that may be transported and operated by a user, which may also provide remote communication capabilities to a network. Examples of remote communication capabilities include using a mobile phone (wireless) network, wireless data network (*e.g.* 3G, 4G or similar networks), Wi-Fi, Wi-Max, or any other communication medium that may provide access to a network such as the Internet or a private network. Examples of mobile devices include mobile phones (*e.g.* cellular phones), PDAs, tablet computers, net books, laptop computers, personal music players, hand-held specialized readers, etc. Further examples of mobile devices include wearable devices, such as smart watches, fitness bands, ankle bracelets, rings, earrings, etc., as well as automobiles with remote communication capabilities. A mobile device may comprise any suitable hardware and software for performing such functions, and may also include multiple devices or components (*e.g.* when a device has remote access to a network by tethering to another device - *i.e.* using the other device as a modem – both devices taken together may be considered a single mobile device).

[0043] A mobile device may also include any suitable device that may be used to conduct a financial transaction, such as to provide payment credentials to a merchant. Such a mobile device may be in any suitable form. For example, suitable mobile devices can be hand-held and compact so that they can fit into a user's wallet and/or pocket (e.g., pocket-sized). They may include smart cards, magnetic stripe cards, keychain devices (such as the Speedpass™ commercially available from

Exxon-Mobil Corp.), etc. Other examples of mobile devices include pagers, payment cards, security cards, access cards, smart media, transponders, and the like. If the mobile device is in the form of a debit, credit, or smartcard, the mobile device may also optionally have features such as magnetic stripes. Such devices can operate in either a contact or contactless mode.

[0044] "Short range communication" or "short range wireless communication" may comprise any method of providing short-range contact or contactless communications capability, such as RFID, Bluetooth™, infra-red, or other data transfer capability that can be used to exchange data between a mobile device and an access device. In some embodiments, short range communications may be in conformance with a standardized protocol or data transfer mechanism (e.g., ISO 14443/NFC). Short range communication typically comprises communications at a range of less than 2 meters. In some embodiments, it may be preferable to limit the range of short range communications (e.g., to a range of less than 1 meter, less than 10 centimeters, or less than 2.54 centimeters) for security, technical, and/or practical considerations.

[0045] An "application" may be computer code or other data stored on a computer readable medium (e.g. memory element or secure element) that may be executable by a processor to complete a task.

[0046] A "digital wallet" can include an electronic device that allows an individual to conduct electronic commerce transactions. A digital wallet may store user profile information, payment credentials, bank account information, one or more digital wallet identifiers and/or the like and can be used in a variety of transactions, such as but not limited to eCommerce, social networks, money transfer/ personal payments, mobile commerce, proximity payments, gaming, and/or the like for retail purchases, digital goods purchases, utility payments, purchasing games or gaming credits from gaming websites, transferring funds between users, and/or the like. A digital wallet may be designed to streamline the purchase and payment process. A digital wallet may allow the user to load one or more payment cards onto the digital

wallet so as to make a payment without having to enter an account number or present a physical card.

[0047] A “digital wallet provider” may include an entity, such as an issuing bank or third party service provider, that issues a digital wallet to a user that enables the user to conduct financial transactions. A digital wallet provider may provide standalone user-facing software applications that store account numbers, or representations of the account numbers (e.g., payment tokens), on behalf of a cardholder (or other user) to facilitate payments at more than one unrelated merchant, perform person-to-person payments, or load financial value into the digital wallet. A digital wallet provider may enable a user to access its account via a personal computer, mobile device or access device. Additionally, a digital wallet provider may also provide one or more of the following functions: storing multiple payment cards and other payment products on behalf of a user, storing other information including billing address, shipping addresses, and transaction history, initiating a transaction by one or more methods, such as providing a user name and password, NFC or a physical token, and may facilitate pass-through or two-step transactions.

[0048] A “user” may include an individual that may be associated with one or more personal accounts and/or mobile devices. The user may also be referred to as a cardholder, account holder, or consumer.

[0049] An “issuer” or “authorizing entity” may typically refer to a business entity (e.g., a bank) that maintains an account for a user.

[0050] A “merchant” may typically be an entity that engages in transactions and can sell goods or services, or provide access to goods or services.

[0051] An "acquirer" may typically be a business entity (e.g., a commercial bank) that has a business relationship with a particular merchant or other entity. Some entities can perform both issuer and acquirer functions. Some embodiments may encompass such single entity issuer-acquirers.

[0052] An "access device" may be any suitable device that provides access to a remote system. An access device may also be used for communicating with a merchant computer, a transaction processing network, an authentication computer, or any other suitable system. An access device may generally be located in any suitable location, such as at the location of a merchant. An access device may be in any suitable form. Some examples of access devices include POS or point of sale devices (e.g., POS terminals), cellular phones, PDAs, personal computers (PCs), tablet PCs, hand-held specialized readers, set-top boxes, electronic cash registers (ECRs), automated teller machines (ATMs), virtual cash registers (VCRs), kiosks, security systems, access systems, and the like. An access device may use any suitable contact or contactless mode of operation to send or receive data from, or associated with, a user mobile device. In some embodiments, where an access device may comprise a POS terminal, any suitable POS terminal may be used and may include a reader, a processor, and a computer-readable medium. A reader may include any suitable contact or contactless mode of operation. For example, exemplary card readers can include radio frequency (RF) antennas, optical scanners, bar code readers, or magnetic stripe readers to interact with a payment device and/or mobile device. In the examples provided herein, an access device and merchant computer may be referred to as separate system components. It should be appreciated, however, that the access device and merchant computer may be a single component, for example, one merchant mobile device or POS device.

[0053] An "authorization request message" may be an electronic message that is sent to a transaction processing network and/or an issuer of a payment card to request authorization for a transaction. An authorization request message according to some embodiments may comply with ISO 8583, which is a standard for systems that exchange electronic transaction information associated with a payment made by a user using a payment device or payment account. The authorization request message may include an issuer account identifier that may be associated with a payment device or payment account. An authorization request message may also comprise additional data elements corresponding to "identification information"

including, by way of example only: a service code, a CVV (card verification value), a dCVV (dynamic card verification value), a PAN (primary account number or "account number"), a payment token, a user name, an expiration date, etc. An authorization request message may also comprise "transaction information," such as any information associated with a current transaction, such as the transaction amount, merchant identifier, merchant location, acquirer bank identification number (BIN), card acceptor ID, etc., as well as any other information that may be utilized in determining whether to identify and/or authorize a transaction.

[0054] An "authorization response message" may be an electronic message reply to an authorization request message generated by an issuing financial institution or a transaction processing network. The authorization response message may include, by way of example only, one or more of the following status indicators: Approval -- transaction was approved; Decline -- transaction was not approved; or Call Center -- response pending more information, merchant must call the toll-free authorization phone number. The authorization response message may also include an authorization code, which may be a code that a credit card issuing bank returns in response to an authorization request message in an electronic message (either directly or through the transaction processing network) to the merchant's access device (e.g. POS equipment) that indicates approval of the transaction. The code may serve as proof of authorization. As noted above, in some embodiments, a transaction processing network may generate or forward the authorization response message to the merchant.

[0055] A "server computer" may include a powerful computer or cluster of computers. For example, the server computer can be a large mainframe, a minicomputer cluster, or a group of servers functioning as a unit. In one example, the server computer may be a database server coupled to a Web server. The server computer may be coupled to a database and may include any hardware, software, other logic, or combination of the preceding for servicing the requests from one or more client computers. The server computer may comprise one or more computational apparatuses and may use any of a variety of computing structures,

arrangements, and compilations for servicing the requests from one or more client computers.

[0056] **FIG. 1** shows a system **100** comprising a number of components. The system **100** comprises a token vault **110**, which may be an example of at least part of a token service provider computer. The token vault **110** may be in communication with one or more of a token requestor **115**, a merchant computer **130**, an acquirer computer **135**, a transaction processing network **140**, and an authorizing entity computer **150**. Each of the token requestor **115**, the merchant computer **130**, the acquirer computer **135**, the transaction processing network **140**, and the authorizing entity computer **150** may be embodied by one or more computers.

[0057] The user **120** (i.e., a device operated by the user **120**) may be in communication with the token requestor **115**, the authorizing entity computer **150**, and the merchant computer **130**. Furthermore, the merchant computer **130**, the acquirer computer **135**, the transaction processing network **140**, the authorizing entity computer **150**, and a token network **145** may all be in operative communication with each other through any suitable communication channel or communications network. A suitable 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.

[0058] The token vault **110** may be associated with the transaction processing network **140**, the authorizing entity computer **150**, the acquirer computer **135**, or the merchant computer **130**. For example, in some embodiments, a token service provider may comprise an associated token vault **110** and transaction processing network **140**. The token vault **110** may be able to issue payment tokens, token codes, and domain IDs. The token vault **110** may also be able to authenticate and de-tokenize payment tokens, token codes, and domain IDs.

[0059] An example of the token vault **110**, according to some embodiments of the invention, is shown in **FIG. 2**. The token vault **110** comprises a processor **110A**, a network interface **110B**, a token database **110C**, a token code database **110D**, a token requestor database **110E**, and a computer readable medium **110F**.

[0060] The computer readable medium **110F** may comprise an authentication module **110G**, a token generation module **110H**, a token identification module **110J**, a token code generation module **110K**, a communication module **110L**, a de-tokenization module **110M**, a registration module **110N**, and any other suitable module. It may also comprise code, executable by the processor **110A** for implementing a method comprising receiving a first token request from a first token requestor computer, wherein the first token request includes a value credential and a first domain identifier; identifying a value token associated with the value credential; generating a first token code associated with the value token; assigning the value token and the first token code to the first domain identifier; transmitting the value token and the first token code to the first token requestor computer, wherein the first token requestor subsequently uses the value token for an interaction, and wherein the first token requestor's subsequent use of the value token is valid if the value token is accompanied by the first token code; receiving a second token request from a second token requestor computer, wherein the second token request includes the value credential and a second domain identifier; identifying the value token associated with the value credential; generating a second token code associated with the value token; assigning the value token and the second token code to the second domain identifier; and transmitting the value token and the second token code to the second token requestor computer, wherein the second token requestor subsequently uses the value token for an interaction, and wherein the second token requestor's subsequent use of the value token is valid if the value token is accompanied by the second token code.

[0061] The authentication module **110G** may comprise code that causes the processor **110A** to authenticate a token requestor **115**. For example, the token vault **110** can receive a token request from the token requestor **115**, the request including

a domain ID that identifies the token requestor **115**, as well as a set of payment credentials that identify a payment account. The authentication module **110G** may contain logic that causes the processor **110A** to authenticate the domain ID or otherwise authenticate the token requestor. The authentication module **110G** may utilize information, such as stored domain IDs, from the token requestor database **110E**.

[0062] The token generation module **110H** may comprise code that causes the processor **110A** to generate payment tokens. For example, the token generation module **110H** may contain logic that causes the processor **110A** to issue a payment token that can be used as surrogate payment account information for a set of payment credentials included in a token request. Any suitable token generation process can be used. Such processes can utilize hashing, encryption, or random token generation processes. A record of the token may be stored at the token database **110C**, and the record may include the token, token expiration date, associated payment account information, token assurance information, token requestor information (e.g. a domain ID), and/or any other suitable information.

[0063] The token generation module **110H** may be able to periodically update payment tokens. For example, new payment tokens can be generated and sent to replace old payment tokens, and the payment tokens can be stored in token database **110C**.

[0064] The token identification module **110J** may comprise code that causes the processor **110A** to identify a payment token. For example, the token identification module **110J** may contain logic that causes the processor **110A** to identify a payment token that is associated with a set of payment credentials received in a token request. In some embodiments, a payment token may be assigned to more than one token requestor **115**. The payment token may be generated when a first token requestor **115** initially requests a token for the payment credentials. When a second token requestor **115** requests a token for the same payment credentials, the token identification module **110J** may be used to identify the existing payment token, instead of generating another payment token. The token

identification module **110J** also may update the record of the payment token at the token database **110C** to indicate that the second token requestor **115** (e.g. the second domain ID) is also associated with the payment token.

[0065] The token code generation module **110K** may comprise code that causes the processor **110A** to generate token codes. For example, the token code generation module **110K** may contain logic that causes the processor **110A** to generate a token code associated with a payment token for a certain token requestor **115**. Any suitable token code generation process can be used. Such processes can utilize hashing, encryption, or random token generation processes. In some embodiments, when a payment token is provided to a token requestor **115**, a token code is also provided. The token code may serve as a passcode for utilizing the payment token. The token vault **110** may provide a different token code to each token requestor that requests the same payment token. Thus, multiple token requestors **115** may possess the same payment token, but they each may have a different token code for that payment token. In order to use the payment token, a token requestor **115** may also need to provide their token code.

[0066] The token code generation module **110K** may also store a record of the token code at the token code database **110D**. The record may indicate a domain ID and/or a payment token associated with the token code. In addition or alternatively, the token code generation module **110K** may cause the record of the payment token at the token database **110C** to indicate one or more token codes and/or domain IDs associated with the payment token.

[0067] The token code generation module **110K** may be able to periodically update token codes. For example, new token codes can be generated and sent to replace old token codes, and the new token codes can be stored in the token code database **110D**.

[0068] The communication module **110L** may comprise code that causes the processor **110A** to communicate with one or more entities. For example, the communication module **110L** may contain logic that causes the processor **110A** to receive token requests and send payment tokens to token requestors **115**. The

communication module **110L** may also be able to receive de-tokenization requests and send payment credentials to the transaction processing network **140** and any other suitable entity.

[0069] The de-tokenization module **110M** may comprise code that causes the processor **110A** to de-tokenize a payment token. For example, the de-tokenization module **110M** may contain logic that causes the processor **110A** to identify a set of payment credentials associated with a payment token (e.g., as indicated in the token database **110C**). The de-tokenization module **110M** may also be able to authenticate a payment token by validating that a received token code and/or domain ID are associated with the payment token (e.g., as indicated in the token code database **110D**). For example, the token vault **110** may receive requests for payment account information from the transaction processing network **140**, authorizing entity computer **150**, and any other suitable entities. The token vault **110** may receive such a de-tokenization request including a payment token, token code, and/or domain ID, the token vault **110** may authenticate the information and identify the associated payment credentials, and then the token vault **110** may provide the payment account information to the de-tokenization requestor.

[0070] The token vault **110** may also allow the token requestor **115** to register at the token vault **110**. The registration module **110N** may comprise code that causes the processor **110A** to register a token requestor **115**. For example, the registration module **110N** may contain logic that causes the processor **110A** to receive a request to register from the token requestor **115**, the request including information about the token requestor **115**, information about how tokens will be used (e.g. token domain information), a desired token assurance level, or any other suitable information. The token vault **110** may authenticate the identity of the token requestor **115**, and the token vault **110** may approve or reject the registration request. If approved, the token vault **110** may generate and issue a domain ID for the token requestor **115**. The domain ID may serve to identify the token requestor **115** as well as a token domain for which issued payment tokens will be used.

[0071] The registration module **110N** may be able to periodically update domain IDs. For example, new domain IDs can be generated and sent to replace old domain IDs, and the new domain IDs can be stored in the token requestor database **110E**.

[0072] Referring back to **FIG. 1**, a number of different entities can act as the token requestor **115**, according to embodiments of the invention. For example, in some embodiments, the token requestor **115** can be a mobile device associated with the user **120**. A user **120** may activate a mobile payment functionality on the mobile device and request that a certain set of payment credentials are usable for mobile payments. In this case, the mobile device may request a payment token associated with the payment credentials. Then, the user **120** can use the mobile device to provide a payment to the merchant computer **130**.

[0073] **FIG. 3** shows an example of such a mobile device **315** that can act as a token requestor **115** and/or be used for mobile payments, according to some embodiments of the invention. Mobile device **315** may include circuitry that is used to enable certain device functions, such as telephony. The functional elements responsible for enabling those functions may include a processor **315A** that can execute instructions that implement the functions and operations of the device. Processor **315A** may access memory **315E** (or another suitable data storage region or element) to retrieve instructions or data used in executing the instructions, such as provisioning scripts and mobile applications. Data input/output elements **315C**, such as a keyboard or touchscreen, may be used to enable a user to operate the mobile device **315** and input data (e.g., user authentication data). Data input/output elements may also be configured to output data (via a speaker, for example). Display **315B** may also be used to output data to a user. Communications element **315D** may be used to enable data transfer between mobile device **315** and a wired or wireless network (via antenna **315H**, for example) to assist in connectivity to the Internet or other network, and enabling data transfer functions. Mobile device **315** may also include contactless element interface **315F** to enable data transfer between contactless element **315G** and other elements of the device, where contactless

element **315G** may include a secure memory and a near field communications data transfer element (or another form of short range communications technology). As noted, a cellular phone or similar device is an example of a mobile device **315** that may be used in accordance with embodiments of the present invention. However, other forms or types of devices may be used without departing from the underlying concepts of the invention. For example, the mobile device **315** may alternatively be in the form of a payment card, a key fob, a tablet computer, a wearable device, etc.

[0074] The memory **315E** may comprise a digital wallet application **315J**, a payment module **315K**, a token request module **315L**, a credential storage **315M**, a payment token **315N**, a domain ID **315P**, a token code **315Q**, and any other suitable module or data. The mobile device **315** may have any number of mobile applications installed or stored on the memory **315E** and is not limited to that shown in **FIG. 3**.

[0075] The digital wallet application **315J** may comprise code that causes the processor **315A** to provide digital wallet services. For example, the digital wallet application **315J** may contain logic that causes the processor **315A** to provide a user interface for the user to provide input and initiate, facilitate, and manage transactions using the mobile device **315**. The digital wallet application **315F** may be able to access payment credentials, such as the token **315N**, the token code **315Q**, and the domain ID **315P** stored in the credential storage **315M**. Further, the digital wallet application **315F** may be able to obtain payment tokens from the token vault **110** (e.g., via the token request module **315L**). For example, the token request module **315**L may send a token request including a set of payment credentials (which may have been provided by the user **120**) as well as a domain ID **315P** associated with the mobile device **315**.

[0076] The digital wallet application **315F** may also be able to provide the payment credentials for a transaction (e.g., via the payment module **315L**). In some embodiments, the digital wallet application **315F** may be able to provide payment credentials for online payments and/or in-person transactions (e.g., via short-range communication). In one example, the digital wallet application **315F** may generate

single data element such as an NFC transmission packet or a QR code including the token **315N**, the token code **315Q**, and the domain ID **315P**, and any other suitable information.

[0077] The credential storage **315M** may be able to store one or more tokens **315N**, token codes **315Q**, and domain IDs **315P**. For example, the credential storage **315M** may store credentials for one or more payment accounts, and there may be a different token **315N** and token code **315Q** for each payment account. The mobile device **315** may also be able to request payment tokens for more than one token domain, and the mobile device **315** may use a different domain ID **315P** for each different token domain. If one token **315N** is used for multiple domains (e.g., in-person contactless payments and online payments), the mobile device **315** may possess a different token code **315Q** for each domain (and domain ID **315P**).

[0078] Referring back to the token requestor **115** in **FIG. 1**, in some embodiments, the merchant computer **130** can act as the token requester **115**. For example, the merchant computer **130** can have a card-on-file for the user **120**. Instead of storing the user's sensitive payment credentials, the merchant computer **130** may be able to request and store an associated payment token. In this case, when the user **120** makes a purchase at the merchant computer **130**, the merchant computer **130** can submit the stored payment token for the transaction. In other embodiments, the merchant computer **130** may not be a token holder, and may instead receive a payment token from the mobile device **315** during a transaction.

[0079] An example of the merchant computer **130** (e.g., a server computer operated by the merchant), according to some embodiments of the invention, is shown in **FIG. 4**. The merchant computer **130** comprises a processor **130A**, a communication interface **130B**, a network interface **130C**, a credential database **130D**, a user database **130E**, and a computer readable medium **130F**.

[0080] The computer readable medium **130F** may comprise a transaction processing module **130G**, a global analysis module **130H**, an authorization module **130J**, a token request module **130K**, a token receive module **130L**, and any other suitable module.

[0081] The transaction processing module **130G** may comprise code that causes the processor **130A** to process transactions. For example, the transaction processing module **130G** may contain logic that causes the processor **130A** to receive payment credentials (e.g., a payment token) from the user **120** for a transaction. In some embodiments, the merchant computer **130** may store a payment token associated with the user **120** (e.g., at the credential database **130D**), and the transaction processing module **130G** identify the payment token when the user **120** initiates a purchase. The transaction processing module **130G** may be able to initiate transaction authorization (using either received payment credentials or an identified payment token), and may also be able to finalize a transaction so that goods and/or services can be released.

[0082] The global analysis module **130H** may comprise code that causes the processor **130A** to analyze payment credentials. For example, the global analysis module **130H** may contain logic that causes the processor **130A** to use a payment token to track user **120** spending habits.

[0065] In some embodiments, the global analysis module **130H** may use a payment token as a user **120** identifier instead of a PAN. The global analysis module **130H** may keep a record (e.g., at the user database **130E**) of user **120** trends by recording transactions that involve the payment token. As explained above, in some embodiments, the same payment token that represents a certain payment account can be provided to multiple token requestors **115**. As a result, the payment token may be static, so that the same payment token is provided each time a user **120** uses a certain payment account at a merchant computer **130**, even if the token code, domain ID, and/or token domain can vary. For example, in some embodiments, one payment account may be associated with several digital wallets, and each digital wallet may have the same payment token associated with the same payment account. The merchant computer **130** may receive the same payment token whenever any of these different digital wallets are used. Thus, the global analysis module **130H** may be able to recognize a payment account by a received payment token.

[0065] In some embodiments, the payment token may comprise 16, 18, or 19 digits and can be formatted as a PAN so that merchants **130** can easily integrate a payment token into an existing user-tracking system that typically uses a PAN for user tracking. For example, users are sometimes identified by a portion of the PAN known as "field 2". Payment tokens may also include a "field 2" that can similarly identify a payment account or user. The payment token may replace a PAN for any suitable identification purpose. A payment token may also be formatted to indicate an authentication entity **150** and/or a token vault **110**.

[0066] In some embodiments, the merchant computer **130** (e.g., via the global analysis module **130H**) may use the payment token and associated user purchase record for other applications including value added services such as loyalty programs, backend applications, and reporting. For example, the payment token may function as a user **120** loyalty identifier for tracking loyalty points (e.g., at the user database **130E**) and providing rewards or special offers.

[0085] The merchant computer **130** (or other entity) may also use the payment token for fraud/risk checks during authorization. For example, the merchant computer **130** (e.g., via the global analysis module **130H**) may be able to detect high-velocity transactions. If too many transactions associated with the payment token occur within a certain amount of time, the merchant computer **130** may suspect fraudulent activity. The merchant computer **130** may then reject further transactions involving the payment token. Alternatively, the merchant computer **130** may mark the payment token as high-risk or put the payment token on a "blacklist." The merchant computer **130** may also use the payment token to review fraud/risk levels after a transaction is authorized.

[0086] The merchant computer **130** may use the payment token to access past transactions. For example, the merchant computer **130** may use the payment token to identify a past transaction (e.g., stored in the user database **130E**) for a refund or an inquiry.

[0087] The payment token may also be used for providing a transaction feed to third party value added applications. For example, the merchant computer **130**

may inform third party applications about user **120** purchases and trends that are tracked via the payment token. A third-party may use a transaction feed for providing rewards or offers to the user **120** in real time or offline (e.g., in a statement of credit).

[0083] The authorization module **130J** may comprise code that causes the processor **130A** to send and receive authorization messages. For example, the authorization module **130J** may contain logic that causes the processor **130A** to generate an authorization request message including payment credentials (e.g., a payment token, token code, domain ID, name, etc.), transaction data, and any other relevant information, and then send the authorization request message to the acquirer computer **135**. An authorization response message from the acquirer computer **135** can also be received and processed.

[0084] The token request module **130K** may comprise code that causes the processor **130A** to request payment tokens. For example, the token request module **130K** may contain logic that causes the processor **130A** to send a token request to the token vault **110**, the token request including a set of payment credentials (which may have been provided by the user **120**) as well as a domain ID **315P** associated with the merchant computer **130**.

[0085] The token receive module **130L** may comprise code that causes the processor **130A** to receive and store payment tokens and other associated credentials. For example, the token receive module **130L** may contain logic that causes the processor **130A** to receive a requested payment token along with an associated token code, domain ID, and/or any other suitable information. The token receive module **130L** may be able to store any received payment tokens and associated information in the credential database **130D**. The merchant computer **130** may store payment tokens for users **120** that wish to store payment account information at the merchant computer **130** (e.g., for convenience). The merchant computer **130** may be able to submit a stored payment token for one or more purchases initiated by the user **120**.

[0086] The credential database **130D** may be able to store one or more payment tokens, token codes, and/or domain IDs. For example, the credential storage **130D** may store credentials for one or more payment accounts, and there may be a different payment token and token code for each payment account. The merchant computer **130** may also be able to request payment tokens for more than one token domain, and the merchant computer **130** may use a different domain ID for each different token domain.

[0078] Referring back to the token requestor **115** in **FIG. 1**, in some embodiments, the token requestor **115** may be a token aggregator. For example, a token aggregator may request tokens on behalf of one or more digital wallet providers. A digital wallet provider may prefer not to register with a token vault **110** and instead receive tokens from the token aggregator. Digital wallet providers may receive tokens from a token aggregator and then provide the tokens to users and mobile devices that use a digital wallet. In some embodiments, a token requestor **115** may be able to request and collect tokens from multiple token vaults **110** or token service providers, and may request tokens that are valid for multiple transaction processing networks **140**. The token requestor **115** may register with each token vault **110** separately.

[0087] Referring back to **FIG. 1**, the acquirer computer **135** may be associated with the merchant computer **130**, and may manage authorization requests on behalf of the merchant computer **130**. The acquirer computer **135** may receive an authorization request message including payment information from the merchant computer **130** and send the authorization request message to the transaction processing network **140**.

[0088] As explained above, in some embodiments, the same payment token that represents a certain payment account can be held and used by different token requestors **115** (e.g., mobile devices **315**, merchants **130**, third-party token providers, etc.). As a result, each time the payment account is used for a transaction, the acquirer computer **135** may receive the same payment token in an authorization request message, regardless of the payment domain. As a result, the

acquirer computer **135** may also be able use the payment token to track user **120** activity, as already described for the merchant computer **130** above.

[0089] For example, the acquirer computer **135** may use the payment token for online fraud analysis, offline fraud analysis, loyalty services, third party loyalty programs, reporting to merchants, or any other suitable purpose. For example, in some embodiments, the acquirer computer **135** may flag a payment account (identified by a payment token) and an associated merchant computer **130** for fraud/risk level scoring. In another example, a payment token may be used for providing card-linked offers. In another example, the acquirer computer **135** may provide a user transaction report to merchants, the report including transactions that involved a certain payment token.

[0090] As shown in **FIG. 1,** the transaction processing network **140** may be disposed between the acquirer computer **135** and the authorizing entity computer **150**. The transaction processing network **140** may include data processing subsystems, networks, and operations used to support and deliver authorization services, exception file services, and clearing and settlement services. For example, the transaction processing network **140** may comprise a server coupled to a network interface (e.g., by an external communication interface), and a databases of information. An exemplary transaction processing network **140** may include VisaNet™. Transaction processing networks such as VisaNet™ are able to process credit card transactions, debit card transactions, and other types of commercial transactions. VisaNet™, in particular, includes a VIP system (Visa Integrated Payments system) which processes authorization requests and a Base II system which performs clearing and settlement services. The transaction processing network **140** may use any suitable wired or wireless network, including the Internet.

[0091] The transaction processing network **140** may be capable of de-tokenizing a token in an authorization request message. For example, the transaction processing network **140** may receive an authorization request message including a payment token, a token code, and/or a domain ID. The transaction processing network **140** may be able to send the received information to the token

vault **110**, then receive associated payment account information from the token vault **110**, and then forward the authorization request message to the authorizing entity computer **150** with the payment account information. The transaction processing network **140** may also receive an authorization response message with the payment account information, and replace some or all of the payment account information with the payment token, token code, and/or domain ID (e.g., via the token vault **110**) before forwarding the message to the acquirer computer **135**.

[0092] In some embodiments, the token vault **110** and the transaction processing network **140** may form a token service system.

[0093] The transaction processing network **140**, the acquirer computer **135**, and the authorizing entity computer **150** may operate suitable routing tables to route authorization request messages using account identifiers such as PANs or tokens. Token routing data may be provided or maintained by the token vault **110**, and may be communicated to any of the entities in **FIG. 1**.

[0094] Referring back to **FIG. 1**, The authorizing entity computer **150** may manage a payment account of the user **120**. The authorizing entity computer **150** may be able authorize transactions that involve the payment account.

[0095] The token network **145** may also be able to provide de-tokenization services. The token network **145** may include another transaction processing network, another token vault, and/or another token requestor. For example, in some embodiments, the transaction processing network **140** may receive a payment token that is associated the token network **145** instead of the token vault **110**. In order to de-tokenize the token for authorization, the transaction processing network **140** may send the token, token code, and/or domain ID to the token network **145**, which may then authenticate the information and return the payment credentials. In some embodiments, the token network **145** may wish to de-tokenize a token that is associated with the transaction processing network **140**, and may obtain associated payment account information by similarly communicating with the transaction processing network **140** and token vault **110**.

[0096] **FIG. 5** shows a block diagram illustrating the use of token codes for various domains, in accordance with some embodiments of the invention. **FIG. 5** depicts a payment card **510**, a payment token **520**, and a plurality of merchants (which could be other types of token requestors, in some embodiments), including a first merchant **530**, second merchant **540**, and third merchant **550**. The payment card **510** may be associated with a user and a PAN, e.g., "4123 4567 8910 1112". Each merchant may wish to store payment credentials associated the user. Instead of storing the PAN, the merchant each may request a payment token. A token service provider (e.g., token vault **110**) may generate a payment token **520** to be associated with the PAN, as described with respect to **FIG. 1**. For example, the payment token **520** associated with the PAN could be "4999 1111 2222 3333." Each merchant may receive the payment token **520**, and then the payment token **520** may be passed in various authorization request and response messages during a transaction, instead of the PAN. In some embodiments, one or more of the merchants may instead be a mobile device **315**. To enable a contactless transactions on the mobile device **315**, the user may enroll the payment card **510** with a digital wallet application or payment application on the mobile device **315**, and the payment token **520** may be provisioned to the mobile device **315**. Also, in some embodiments, the domains can be various acquirers involved in transactions.

[0097] As can be seen in **FIG. 5**, the same payment token **520** can be assigned to each domain (e.g., first domain **530**, second domain **540**, third domain **550**). This may be in contrast to existing implementations in which a different payment token **520** may need to be generated for each separate domain. Each domain may be associated with a domain ID. For example, the first domain **530** may have a domain ID of 123456789101 (also referred to as "**101**"), the second domain **540** may have a domain ID of 123456789102 (also referred to as "**102**"), and the third domain **550** may have a domain ID of 123456789103 (also referred to as "**103**").

[0098] In addition to the payment token **520**, token codes associated with the payment token may also be generated. A different token code may be assigned to

each domain when requesting access to the payment token **520**. For example, the first domain **530** may have a token code of "A53B9", the second domain **540** may have a token code of "1D5F6", and the third domain **550** may have a token code of "6789A". A token code may be a shared secret between the domain and the token provider. Accordingly, token codes may be protected by encryption, secure storage, and any other suitable safeguarding. In some embodiments, the token code may also be used in the creation of a cryptogram. Token codes may also be periodically replaced and updated.

[0099] The combination of the payment token **520** and a token code may constitute a unique combination. Thus, while each domain may possess the same payment token **520**, each domain may possess a unique token code for the payment token **520**.

[0100] The token code may be a hexadecimal value, although the token code could be any other alphabetical, numeric, or alphanumeric value. In some embodiments, the token code may be indicative of the specific domain for which it was obtained.

[0101] The token code may be generated by the same token provider that generated the payment token **520**. The holder of the payment token **520** (e.g., merchant, digital wallet application, etc.) may receive and store both the payment token **520** and the token code at the same time. The payment token **520** and the token code may also be updated at any time for a number of reasons, including to prevent a security threat (e.g., fraudulent transactions).

[0102] The token code may be specific to the domain so that transaction authentication may occur on the combination of the payment token **520**, the token code, and/or the domain ID (along with merchant ID, acquirer BIN/PCR, etc.). Thus, in order for the first merchant **530** to utilize the payment token **520** for a transaction, the first merchant **530** may need to provide the token code "A53B9" and/or the domain ID "**101**" along with the payment token **520** for the transaction. Then, the token vault **110** can authenticate the payment token **520**, the token code "A53B9", and/or the domain ID "**101**". Further, the token vault **110** can verify that the payment

token **520**, the token code "A53B9", and/or the domain ID "**101**" are all associated with one another. The token vault **110** thereby verifies that the first merchant **530** is authorized to utilize the payment token **520**. Thus, the payment token **520** may be distributed to multiple entities, but the payment token **520** may only be usable if it is accompanied by an authentic token code and/or domain ID.

[0103] For example, the second merchant **540** may be able to use the payment token **520**, but the second merchant **540** may need to provide its token code "1D5F6" and its domain ID "**102**". If a fraudster attempts to use the payment token **520** and provides an illegitimate token code, an illegitimate domain ID, or a mismatching token code and domain ID (e.g., token code "A53B9" and domain ID "**103**"), the transaction may be rejected as potentially fraudulent.

[0104] Accordingly, the domains (e.g., merchants, acquirers, etc.) may be able to have a unique identifier (i.e. the payment token **520**) for which to track transactions for risk, fraud, and other analytical metrics. Transactions can be further tracked, divided, analyzed based on the different domain IDs used transactions.

[0105] Payment tokens can be a secure replacement for PANs, and may be formatted similarly to PANs. Accordingly, such an implementation would not require a great deal of infrastructure changes on the part of the domains, and may allow for risk and loyalty management on the part of the domains. Since the payment token **520** can be the same across the various domains, less infrastructure change may be required on the part of the domains than would be required for implementations involving providing a unique payment token **520** for each domain.

[0106] It can be appreciated that the implementation described with respect to **FIG. 5** may allow the use of a single payment token **520** across a plurality of domains. The combination of the single payment token **520** with the great number of possible token codes may allow for a large number of unique combinations. This may be effective for shared BINs.

[0107] Additionally, in some embodiments, it may be possible to use token codes in conjunction with a PAN. Instead of introducing the payment token **520**, the

PAN may be maintained under the condition that a token code should be provided with the PAN. Thus, similar to the payment token **520**, the PAN can be widely distributed, but only usable for authorized domains where a token code and/or domain ID have been assigned.

[0108] A method **600** according to embodiments of the invention can be described with respect to **FIG. 6**. Some elements in other Figures are also referred to. The steps shown in the method **600** may be performed sequentially or in any suitable order in embodiments of the invention. In some embodiments, one or more of the steps may be optional.

[0109] The various messages in **FIG. 6** may use any suitable form of communication. In some embodiments, a request or response may be in an electronic message format, such as an e-mail, a short messaging service (SMS) message, a multimedia messaging service (MMS) message, a hypertext transfer protocol (HTTP) request message, a transmission control protocol (TCP) packet, a web form submission. The request or response may be directed to any suitable location, such as an e-mail address, a telephone number, an internet protocol (IP) address, or a uniform resource locator (URL). In some embodiments, a request or response may comprise a mix of different message types, such as both email and SMS messages.

[0110] In the method **600** described below, the mobile device **315** is the token requestor. However, as discussed above, in some embodiments the token requestor may alternatively be a merchant, a third-party token provider, or any other suitable entity.

[0111] At step **S602**, the user **120** may indicate a desire to have payment credentials provisioned onto the mobile device **315**. For example, the user **120** may provide a PAN or other suitable payment credentials to the digital wallet application **315J**.

[0112] At step **S604**, the mobile device **315** may send a token request to the token vault **110** (e.g., via the token request module **315L**). The token request may

include the payment credentials (e.g., the PAN, expiration date, etc.). The token request may also include mobile device **315** identifying information, such as the domain ID **315P**, and any other suitable information.

[0113] At step **S606**, the token vault **110** may authenticate the mobile device **315**. The token vault 110 can check the payment credentials, device ID, etc. Also, the token vault **110** may verify that the domain ID **315P** is valid in the token requestor database **110E** (e.g., via the authentication module **110G**).

[0114] At step **S608**, the token vault **110** may identify a payment token that is associated with the payment credentials in the token database **110C** (e.g., via the token identification module **110J**). For example, a payment token may already have been generated for the payment credentials for another token requestor. If there is no payment token already associated with the payment credentials, the token vault **110** may generate a payment token, create an association between the payment token and the received payment credentials, and store a record of the payment token in the token database **110C** (e.g., via the token generation module **110H**).

[0115] At step **S610**, the token vault **110** may generate a token code for the mobile device **315** to use in conjunction with the payment token (e.g., via the token code generation module **110K**).

[0116] At step **S612**, the token vault **110** may store a record indicating an association between the token code, the payment token, and the domain ID at the token code database **110D** and/or the token database **110C** (e.g., via the token code generation module **110K**). For example, the token code and domain ID may be added to a list of token codes and domain IDs that are authorized to use the payment token. In another example, the payment token and token code may be added to a list of payment tokens and token codes that the domain ID is authorized to use.

[0117] At step **S614**, the token vault **110** may send a token response to the mobile device **315**. The token response may include the payment token, the token code, and/or the domain ID.

[0118] At step **S616**, the mobile device **315** may store the payment token **315N**, the token code **315Q**, and/or the domain ID **315P** (e.g., at the credential storage **315M**). This may conclude the token provisioning process, and the mobile device **315** may then be able to utilize the payment token **315N** for mobile transactions.

[0119] At step **S618**, the user **120** may initiate a purchase. For example, the user **120** may activate the digital wallet application **315J**, and may indicate a desire to make a payment (e.g., by selecting a "payment" option). The user **120** may wish to purchase a good or service at the merchant computer **130** via the mobile device **315**. Further, the user **120** may select a payment account within the digital wallet application **315J** (e.g., by selecting a visually displayed icon representing a payment account), and present the mobile device **315** to an access device for transmitting the payment information.

[0120] At step **S620**, the mobile device **315** may provide the payment token **315N**, the token code **315Q**, the domain ID **315P**, and/or any other suitable information to the merchant computer **130** for the transaction (e.g., via the payment module **315K**). For example, the mobile device **315** may provide the payment information via short-range communication to a merchant access device. The token code **315Q** may be encrypted or otherwise protected during transmission, such that only the token vault **110** can decrypt and view the token code **315Q**.

[0121] At step **S622**, the merchant computer **130** may analyze the payment token and/or the domain ID. For example, the merchant computer **130** may track user **120** activity based on the payment token, and may use the payment token to perform fraud risk analysis (e.g., check a blacklist and determine transaction velocity), process loyalty-related services, update a user **120** record, and perform any other suitable operations (e.g., via the global analysis module **130H**). The merchant computer **130** may not view or utilize the token code, and may simply pass the token code along in the authorization request message.

[0122] At step **S624**, the merchant computer **130** may send an authorization request message to the acquirer computer **135**. The authorization request message

may include transaction information, the payment token, the token code, the domain ID, and any other suitable information.

[0123] At step **S626**, the acquirer computer **135** may optionally analyze the payment token **315N** and/or the domain ID. For example, the acquirer computer **135** may track user **120** activity based on the payment token, and may use the payment token to perform fraud risk analysis (e.g., check a blacklist and determine transaction velocity), process loyalty-related services, update a user **120** record, and perform any other suitable operations. The acquirer computer **135** may not view or utilize the token code, and may simply pass the token code along in the authorization request message.

[0124] At step **S628**, the acquirer computer **135** may forward the authorization request message to the transaction processing network **140**.

[0125] At step **S630**, the transaction processing network **140** may determine that a payment token is included in the authorization request message. The transaction processing network **140** may send a de-tokenization request to the token vault **110**, the de-tokenization request including the payment token, the token code, the domain ID, and/or any other suitable information.

[0126] At step **S632**, the token vault **110** may authenticate the payment token, the token code, and/or the domain ID (e.g., via the authentication module **110G**, the token database **110C**, and/or the token code database **110D**). For example, the token vault **110** may validate that the payment token, the token code, and/or the domain ID are all associated with one another. The token vault **110** may confirm that the mobile device **315** is authorized to use the payment token. For example, the token vault **110** may confirm that the token code was issued to the mobile device **315** (e.g., it was assigned to the domain ID).

[0127] At step **S634**, the token vault **110** may identify the payment credentials (e.g., the PAN) that are associated with the payment token in the token database **110C** (e.g., via the de-tokenization module **110G**).

[0128] At step **S636**, the token vault **110** may send a de-tokenization response to the transaction processing network **140**, the de-tokenization response including the payment credentials.

[0129] At step **S638**, the transaction processing network **140** may add the payment credentials to the authorization request message and forward the authorization request message to the authorizing entity computer **150**. The transaction processing network **140** may optionally remove the payment token, the token code, and/or the domain ID from the authorization request message.

[0130] At step **S640**, the authorizing entity computer **150** then determines whether or not the transaction should be authorized. The authorizing entity computer **150** may check the status of the payment account, conduct any appropriate fraud or credit checks, and perform any other suitable processing in order to determine whether or not to approve of the transaction.

[0131] At step **S642**, the authorizing entity computer **150** then generates and sends an authorization response message back to the transaction processing network **140**.

[0132] At step **S644**, the transaction processing network **140** may remove the payment credentials from the authorization response message. The transaction processing network **140** may also reformat the authorization response message to include the payment token, the token code, and/or the domain ID. In some embodiments, the token code may not be included in the authorization response message, as the token code may not be readable or useful to the acquirer computer **135** or merchant computer **130**. In some embodiments, the transaction processing network **140** may obtain the payment token, the token code, and/or the domain ID from the token vault **110** in order to include the information in the authorization response message.

[0133] At step **S646**, the transaction processing network **140** may then forward the authorization response message to the acquirer computer **135**.

[0134] At step **S648**, the acquirer computer **135** may forward the authorization response message to the merchant computer **130**.

[0135] At step **S650**, the merchant computer **130** may release the purchased goods and/or services to the user **120** based on the authorization response message.

[0136] At the end of the day or at some other predetermined interval of time, a clearing and settlement process between the authorizing entity computer **150**, the transaction processing network **140**, and the acquirer computer **135** may occur. In the clearing and settlement process, account information and token exchanges that are similar to those in the above-described authorization processing steps can occur.

[0137] Accordingly, it may be possible to replace sensitive payment credentials with the payment token for purchases, but it may still be possible to track user **120** activity based on the payment token, as the payment token can remain static and can be submitted by multiple entities. The security of the payment token can be maintained by incorporating the token code and the domain ID, as the payment token may only be valid if accompanied by a valid token code and domain ID. Thus, with minimal changes to existing systems, payment security is improved without sacrificing the ability to track user **120** activity.

[0138] **FIG. 7** shows a flow diagram illustrating a method for sharing a token according to embodiments of the invention.

[0139] Before describing **FIG. 7**, it is helpful to discuss why a token might be shared. Merchants and other entities sometimes share user payment credentials. For example, a user may book a hotel room via a hotel reservation website (the first merchant), and the website may forward the user's payment credentials to the hotel (the second merchant). This can allow the hotel to process any room charges and otherwise utilize the user's payment credentials. Also, many merchant systems use payment credentials to generally keep track of a user. For example, instead of generating a new reservation number, the website and/or hotel may use the payment credentials as a unique reservation number. When otherwise communicating about

the user, the website and hotel may identify the user by the payment credentials. In addition to business agreements, many other scenarios exist where different payment credential holding entities share the user's payment credentials and track the user by payment credentials.

[0140] Accordingly, in payment token systems, token holding entities may want to share payment tokens in the same way that other payment credentials have been shared in the past. However, in typical payment token systems, payment tokens are uniquely assigned to a domain. Only the token holder can use the payment token, so the payment token cannot be functionally shared. If a payment token is used in the wrong domain, the transaction may be rejected. For example, the hotel reservation website may have been provisioned with a payment token associated with a user. The hotel reservation website could send the payment token to the hotel, but the hotel will not be able to use the payment token.

[0141] Another problem with typical tokenization systems is that the user can no longer be tracked. In the past, the user could be tracked by a PAN across merchants and acquirers. However, in token systems, there may by multiple payment tokens associated with the same PAN. Accordingly, different merchants and domains cannot identify the user by a single static number, as they have done with the PAN in the past.

[0142] Embodiments of the invention solve these problems by using a single payment token. The user can be identified with the single payment token across domains (e.g., merchants and acquirers), similarly to how users have been tracked with a PAN in the past. However, embodiments are more secure than previous systems that shared the PAN, as the single payment token can only be used in authorized domains. If a domain is authorized to use the payment token, it may have a token code and domain ID to prove it.

[0143] In some embodiments, token holders/requestors may be able to share payment tokens, as discussed above. However, to maintain security, if a token holder receives a payment token from another token holder, instead of from the token vault, that token holder may need to seek authorization to use the token from

the token vault. An example of such a token-sharing process, according to embodiments of the invention, can be described with respect to **FIG. 7**.

[0144] A first token requester **115** may possess a payment token, a first token code, a first domain ID, and any other suitable information associated with a user payment account. The first token requestor **115** may want to share the payment token with a second token requestor **715**. The first token requestor **115** and the second token requestor **715** may each be a merchant, mobile device, third-party token provider, or any other suitable entity.

[0145] At step **S702**, the first token requestor **115** may send the payment token to the second token requestor **715**. The first token code may be withheld from the second token requestor **715**, as it may be a shared secret that is only usable by the first token requestor **115**. The first domain ID also be withheld, as it may be an identifier for the first token requestor **115** alone, and it may only be valid for the first token requestor **115**.

[0146] At step **S704**, the second token requestor **715** may send a permission request to the token vault **110**. The permission request may include the payment token, a second domain ID associated with the second token requestor **715**, and any other suitable information. A permission request may differ from a token request in that a permission request includes a payment token, while a token request includes a PAN (or other suitable payment credentials).

[0147] At step **S706**, the token vault **110** may authenticate the second token requestor **715**. For example, the token vault **110** may verify that the second domain ID is valid in the token requestor database **110E** (e.g., via the authentication module **110G**), or otherwise authenticate that the second token requestor **715** is eligible to receive payment tokens. If the second token requestor **715** is not yet registered with the token vault **110** and/or does not have a domain ID, the second token requestor **715** may be prompted to register before a second token code is generated.

[0148] At step **S708**, the token vault **110** may generate a second token code for the second token requestor **715** to use in conjunction with the payment token (e.g., via the token code generation module **110K**).

[0149] At step **S710**, the token vault **110** may store a record indicating an association between the second token code, the payment token, and the second domain ID at the token code database **110D** and/or the token database **110C** (e.g., via the token code generation module **110K**). For example, the second token code and the second domain ID may be added to a list of token codes and domain IDs that are authorized to use the payment token. In another example, the payment token and the second token code may be added to a list of payment tokens and token codes that the domain ID is authorized to use.

[0150] At step **S712**, the token vault **110** may send a permission response to the second token requestor **715**. The token response may include the payment token, the second token code, and/or the second domain ID.

[0151] At step **S714**, the second token requestor **715** may store the payment token, the second token code, and/or the second domain ID. This may conclude the token provisioning process, and the second token requestor **715** may thereby be granted authorization to utilize the payment token for transactions.

[0152] Once the second token requestor **715** receives the payment token from the first token requestor **115**, the two entities may be able to use the payment token to identify the user **120** when communicating about the user **120**. Further, once the second token requestor **715** has the second token code, the second token requestor **715** may be able to use the payment token to submit charges associated with the user **120**.

[0153] Embodiments of the invention have a number of advantages. Token proliferation issues are solved, as introducing token codes allows payment tokens to be re-used for multiple domains and transactions. In previous systems, there was a risk that the total number of possible token values would be exhausted, as multiple tokens were being rapidly distributed for single payment accounts. In contrast, token

codes allow single payment tokens to be safely distributed to multiple token holders. In some embodiments, token codes can effectively serve to differentiate a single payment token into multiple unique (payment token + token code) pairs. Also, because introducing token codes reduces the amount of unique payment tokens needed, it may be easier to divide limited token BIN space among smaller issuers.

[0154] Additionally, embodiments allow tokens to be introduced without interrupting the user-activity tracking programs of several entities (e.g., acquirers, merchants). Entities that previously tracked a user's payment account activity based on a PAN can continue to track user activity based on a similarly formatted payment token that may now be present in each transaction involving the payment account.

[0155] Further, payment security can be increased. Instead of exposing sensitive payment credentials (e.g., a PAN), a payment token can be used. In some embodiments, additional authentication takes places when the payment token is used for a transaction. For example, a valid token code and corresponding domain ID may need be provided with the payment token. Token codes and domain IDs may only be available to registered token requestors. Thus, the entity submitting the payment token can be identified, and a shared secret (i.e., the token code) may protect the payment token from being fraudulently used. Additionally, payment tokens, token codes, and domain IDs can be refreshed periodically.

[0156] Another advantage in embodiments is that the system may be relatively easy to adopt. Merchants, acquirers, and other involved entities may be able to engage in the token system by simply passing along a token code and/or domain ID in an authorization message. Applications that previously relied on a PAN, such as loyalty, merchant-to-merchant communication, and other user tracking applications, may still function by utilizing the payment token (which may appear similar to the PAN). The token vault may be able to orchestrate issuing, authentication, and maintenance of payment tokens, token codes, and domain IDs such that other entities are not burdened. Also, in some embodiments, current authorization message formats may be able to accommodate the addition of the token code and/or domain ID without needing to add new fields or otherwise change

the format of the messages. Thus, embodiments may provide an easy transition from traditional payment systems into token systems.

[0157] **FIG. 8** is a high-level block diagram **11** of a computer system that may be used to implement any of the entities or components described above. The subsystems shown in **FIG. 8** are interconnected via a system bus **75**. Additional subsystems include a printer **74**, keyboard **78**, storage device **79**, and monitor **76**, which is coupled to display adapter **82**. Peripherals and input/output (I/O) devices, which couple to I/O controller **71**, can be connected to the computer system by any number of means known in the art, such as a serial port. For example, I/O port **77** or external interface **81** can be used to connect the computer apparatus to a wide area network such as the Internet, a mouse input device, or a scanner. The interconnection via system bus **75** allows the central processor **73** to communicate with each subsystem and to control the execution of instructions from system memory **72** or the storage device **79**, as well as the exchange of information between subsystems. The system memory **72** and/or the storage device may embody a computer-readable medium.

[0158] As described, the inventive service may involve implementing one or more functions, processes, operations or method steps. In some embodiments, the functions, processes, operations or method steps may be implemented as a result of the execution of a set of instructions or software code by a suitably-programmed computing device, microprocessor, data processor, or the like. The set of instructions or software code may be stored in a memory or other form of data storage element which is accessed by the computing device, microprocessor, etc. In other embodiments, the functions, processes, operations or method steps may be implemented by firmware or a dedicated processor, integrated circuit, etc.

[0159] Any of the software components or functions described in this application may be implemented as software code to be executed by a processor using any suitable computer language such as, for example, Java, C++ or Perl using, for example, conventional or object-oriented techniques. The software code may be stored as a series of instructions, or commands on a computer-readable

medium, such as a random access memory (RAM), a read-only memory (ROM), a magnetic medium such as a hard-drive or a floppy disk, or an optical medium such as a CD-ROM. Any such computer-readable medium may reside on or within a single computational apparatus, and may be present on or within different computational apparatuses within a system or network.

[0160] While certain exemplary embodiments have been described in detail and shown in the accompanying drawings, it is to be understood that such embodiments are merely illustrative of and not intended to be restrictive of the broad invention, and that this invention is not to be limited to the specific arrangements and constructions shown and described, since various other modifications may occur to those with ordinary skill in the art.

[0161] As used herein, the use of "a", "an" or "the" is intended to mean "at least one", unless specifically indicated to the contrary.

WHAT IS CLAIMED IS:

1. A method comprising:

receiving, by a processor in a token service computer, a first token request from a first token requestor computer, wherein the first token request includes a value credential and a first domain identifier;

identifying, by the processor in the token service computer, a value token associated with the value credential;

generating, by the processor in the token service computer, a first token code associated with the value token;

assigning, by the processor in the token service computer, the value token and the first token code to the first domain identifier; and

providing, by the processor in the token service computer, the value token and the first token code to the first token requestor computer, wherein the first token requestor subsequently uses the value token for an interaction, wherein the first token requestor's subsequent use of the value token is valid if the value token is accompanied by the first token code, and wherein the first token code is domain specific.

2. The method of claim 1, further comprising:

receiving, by the processor in the token service computer, a second token request from a second token requestor computer, wherein the second token request includes the value credential and a second domain identifier;

identifying, by the processor in the token service computer, the value token associated with the value credential;

generating, by the processor in the token service computer, a second token code associated with the value token, wherein the second token code is different than the first token code;

assigning, by the processor in the token service computer, the value token and the second token code to the second domain identifier; and

providing, by the processor in the token service computer, the value token and the second token code to the second token requestor computer, wherein

the second token requestor subsequently uses the value token for an interaction, wherein the second token requestor's subsequent use of the value token is valid if the value token is accompanied by the second token code, and wherein the second token code is domain specific.

3. The method of claim 2, wherein the first token code is a first hexadecimal value and the second token code is a second hexadecimal value.

4. The method of claim 2, wherein the first token code is specific to a first domain, wherein the second token code is specific to a second domain, and wherein the first domain is different than the second domain.

5. The method of claim 4, wherein the first domain includes e-commerce interactions and wherein the second domain includes in-person interactions.

6. A token service computer comprising:

a processor; and

a computer readable medium, the computer readable medium comprising code, executable by the processor, for implementing a method comprising:

receiving a first token request from a first token requestor computer, wherein the first token request includes a value credential and a first domain identifier;

identifying a value token associated with the value credential;

generating a first token code associated with the value token;

assigning the value token and the first token code to the first domain identifier; and

providing the value token and the first token code to the first token requestor computer, wherein the first token requestor subsequently uses the value token for an interaction, wherein the first token requestor's subsequent use of the value token is valid if the value token is accompanied by the first token code, and wherein the first token code is domain specific.

7. The token service computer of claim 6, wherein the method further comprises:

receiving a second token request from a second token requestor computer, wherein the second token request includes the value credential and a second domain identifier;

identifying the value token associated with the value credential;

generating a second token code associated with the value token, wherein the second token code is different than the first token code;

assigning the value token and the second token code to the second domain identifier; and

providing the value token and the second token code to the second token requestor computer, wherein the second token requestor subsequently uses the value token for an interaction, wherein the second token requestor's subsequent use of the value token is valid if the value token is accompanied by the second token code, and wherein the first token code is domain specific.

8. The token service computer of claim 7, wherein the first token code is a first hexadecimal value and the second token code is a second hexadecimal value.

9. The token service computer of claim 7, wherein the first token code is specific to a first domain, wherein the second token code is specific to a second domain, and wherein the first domain is different than the second domain.

10. The token service computer of claim 9, wherein the first domain includes e-commerce interactions and wherein the second domain includes in-person interactions.

11. A method comprising:

receiving, by a token service system, a first authorization request message including a value token, a first token code, and a first domain identifier;

determining, by the token service system, that the first token code is associated with the value token;

determining, by the token service system, that the value token and the first token code are assigned to the first domain identifier, wherein the first token code is domain specific;

identifying, by the token service system, a value credential associated with the value token;

adding, by the token service system, the value credential to the first authorization request message;

sending, by the token service system, the first authorization request message to an authorizing entity computer;

receiving, by the token service system, a first authorization response message including the value credential from the authorizing entity computer;

replacing, by the token service system, the value credential with the value token and the first token code in the first authorization response message; and

forwarding, by the token service system, the first authorization response message.

12. The method of claim 11, further comprising:

receiving, by the token service system, a second authorization request message including a value token, a second token code, and a second domain identifier;

determining, by the token service system, that the second token code is associated with the value token;

determining, by the token service system, that the value token and the second token code are assigned to the second domain identifier, wherein the second code is different than the first token code, and wherein the second token code is domain specific;

identifying, by the token service system, a value credential associated with the value token;

adding, by the token service system, the value credential to the second authorization request message;

sending, by the token service system, the second authorization request message to the authorizing entity computer;

receiving, by the token service system, a second authorization response message including the value credential from the authorizing entity computer;

replacing, by the token service system, the value credential with the value token and the second token code in the authorization response message; and

forwarding, by the token service system, the authorization response message.

13. The method of claim 11 wherein the token service system comprises a token service computer and a transaction processing network computer.

14. The method of claim 12, wherein the first token code is specific to a first domain, wherein the second token code is specific to a second domain, and wherein the first domain is different than the second domain.

15. The method of claim 14, wherein the first domain includes e-commerce interactions and wherein the second domain includes in-person interactions.

16. A token service system comprising:

a token service computer comprising a first processor and a first computer readable medium, the first computer readable medium comprising code, executable by the first processor to implement a method comprising:

receiving a first value credential request from a transaction processing network computer, the first value credential request including a value token, a first token code, and a first domain identifier;

determining that the first token code is associated with the value token;

determining that the value token and the first token code are assigned to the first domain identifier, wherein the first token code is domain specific;

identifying a value credential associated with the value token; and

sending a first value credential response including the value credential to the transaction processing network computer; and

a transaction processing network computer in communication with the token service computer, the transaction processing network computer comprising a second processor and a second computer readable medium, the second computer readable medium comprising code, executable by the second processor for implementing a method comprising:

receiving a first authorization request message including the value token, the first token code, and the first domain identifier;

sending the first value credential request to the token service computer, the first value credential request including the value token, the first token code, and the first token requestor;

receiving the first value credential response including the value credential associated with the value token from the token service computer;

adding the value credential to the first authorization request message;

sending the first authorization request message to an authorizing entity computer;

receiving a first authorization response message including the value credential from the authorizing entity computer;

replacing the value credential with the value token and the first token code in the first authorization response message; and

forwarding the first authorization response message;

17. The token service system of claim 16, wherein the method implemented by the first processor further comprises:

receiving a second value credential request from a transaction processing network computer, the second value credential request including the value token, a second token code, and a second domain identifier;

determining that the second token code is associated with the value token;

determining that the value token and the second token code are assigned to the second domain identifier, wherein the second code is different than the first token code, and wherein the second token code is domain specific;

identifying the value credential associated with the value token; and

sending a second value credential response including the value credential to the transaction processing network computer; and

wherein the method implemented by the second processor further comprises:

receiving a second authorization request message including the value token, the second token code, and the second domain identifier;

sending the second value credential request to the token service computer, the second value credential request including the value token, the second token code, and the second token requestor;

receiving the second value credential response including the value credential associated with the value token from the token service computer;

adding the value credential to the second authorization request message;

sending the second authorization request message to an authorizing entity computer;

receiving a second authorization response message including the value credential from the authorizing entity computer;

replacing the value credential with the value token and the second token code in the second authorization response message; and

forwarding the second authorization response message.

18. The token service system of claim 17, wherein the first token code is a first hexadecimal value and the second token code is a second hexadecimal value.

19. The token service system of claim 17, wherein the first token code is specific to a first domain, wherein the second token code is specific to a second domain, and wherein the first domain is different than the second domain.

20. The token service system of claim 19, wherein the first domain includes e-commerce interactions and wherein the second domain includes in-person interactions.