Innovation, Sciences et Développement économique Canada
Office de la Propriété Intellectuelle du Canada

Innovation, Science and Economic Development Canada
Canadian Intellectual Property Office

CA 2991067 C 2021/01/12

(11)(21) 2 991 067

(12) BREVET CANADIEN
CANADIAN PATENT

(13) C

(86) Date de dépôt PCT/PCT Filing Date: 2016/07/26
(87) Date publication PCT/PCT Publication Date: 2017/02/02
(45) Date de délivrance/Issue Date: 2021/01/12
(85) Entrée phase nationale/National Entry: 2017/12/28
(86) N° demande PCT/PCT Application No.: US 2016/043964
(87) N° publication PCT/PCT Publication No.: 2017/019652
(30) Priorités/Priorities: 2015/07/27 (US14/809,762); 2015/07/27 (US14/809,757)

(51) Cl.Int./Int.Cl. *H04L 29/06* (2006.01), *H04W 12/06* (2009.01), *H04W 4/00* (2018.01)

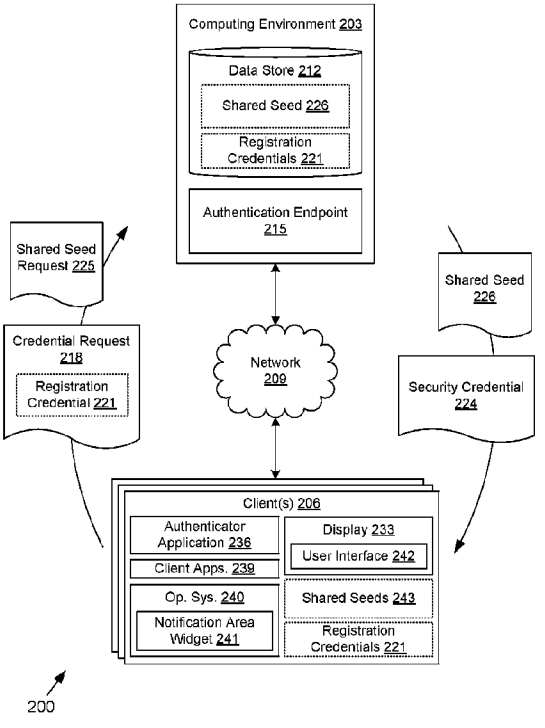
(72) Inventeurs/Inventors:
JOHANSSON, JESPER MIKAEL, US;
HITCHCOCK, DANIEL WADE, US;
CANAVOR, DARREN ERNEST, US;
BHIMANAIK, BHARATH KUMAR, US

(73) Propriétaire/Owner:
AMAZON TECHNOLOGIES, INC., US

(74) Agent: ROBIC

(54) Titre : FOURNITURE DE JUSTIFICATIFS D'AUTHENTIFICATION MULTI-FACTEURS PAR DES NOTIFICATIONS DE DISPOSITIF

(54) Title: PROVIDING MULTI-FACTOR AUTHENTICATION CREDENTIALS VIA DEVICE NOTIFICATIONS



(57) Abrégé/Abstract:
Disclosed are various embodiments for providing multi-factor authentication credentials. In one embodiment, in response to a request from an application, a notification is generated in a notification area of a display. Entry of a user approval is facilitated via the notification. In response to receiving the approval, a security credential is transferred to the application. In another embodiment, the security credential may be shown in the notification area so that a user may enter it in a form field of the application.

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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
2 February 2017 (02.02.2017)

WIPO | PCT

(10) International Publication Number
WO 2017/019652 A1

(51) **International Patent Classification**:
H04L 29/06 (2006.01) *H04W 4/00* (2009.01)
H04W 12/06 (2009.01)

(21) **International Application Number**:
PCT/US2016/043964

(22) **International Filing Date**:
26 July 2016 (26.07.2016)

(25) **Filing Language**: English

(26) **Publication Language**: English

(30) **Priority Data**:
14/809,757 27 July 2015 (27.07.2015) US
14/809,762 27 July 2015 (27.07.2015) US

(71) **Applicant**: **AMAZON TECHNOLOGIES, INC.** [US/US]; P. O. Box 81226, Seattle, WA 98108-1226 (US).

(72) **Inventors**: **JOHANSSON, Jesper, Mikael**; 410 Terry Avenue North, Seattle, WA 98109-5210 (US). **HITCHCOCK, Daniel, Wade**; 410 Terry Avenue North, Seattle, WA 98109-5210 (US). **CANAVOR, Darren, Ernest**; 410 Terry Avenue North, Seattle, WA 98109-5210 (US). **BHIMANAIK, Bharath, Kumar**; 410 Terry Avenue North, Seattle, WA 98109-5210 (US).

(74) **Agents**: **HILDEBRANDT, Thomas, B.** et al.; Thomas / Horsemeyer, Llp, 400 Interstate North Parkway, Ste. 1500, Atlanta, GA 30339 (US).

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

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

[Continued on next page]

(54) **Title**: PROVIDING MULTI-FACTOR AUTHENTICATION CREDENTIALS VIA DEVICE NOTIFICATIONS

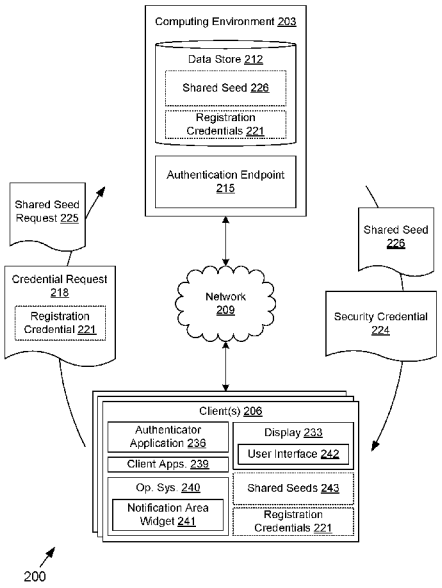


FIG. 2A

(57) **Abstract**: Disclosed are various embodiments for providing multi-factor authentication credentials. In one embodiment, in response to a request from an application, a notification is generated in a notification area of a display. Entry of a user approval is facilitated via the notification. In response to receiving the approval, a security credential is transferred to the application. In another embodiment, the security credential may be shown in the notification area so that a user may enter it in a form field of the application.

WO 2017/019652 A1

WO 2017/019652 A1

Published:

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

PROVIDING MULTI-FACTOR AUTHENTICATION CREDENTIALS VIA DEVICE NOTIFICATIONS

[0001] (INTENTIONALLY LEFT BLANK)

BACKGROUND

[0002] Multi-factor authentication is an access control approach that provides a higher degree of security than the single-factor approaches that have been traditionally employed. Multi-factor authentication may employ two or more authentication factors selected from what the user knows (*e.g.*, passwords, knowledge-based questions, *etc.*), what the user has (*e.g.*, a one-time password generated by a token, a one-time password sent via a different communication channel, *etc.*), and who the user is (*e.g.*, fingerprint scans or other biometric data). For example, to successfully log in to a network service, a user may have to provide a username, a password, and a one-time password generated by a hardware token. Thus, a malicious user who has compromised the username and password cannot log in without also having access to the hardware token. One-time passwords may be generated via time-based one-time password algorithms, hash chains, and other approaches.

SUMMARY

[0002a] According to an aspect, a system is provided. The system includes: at least one computing device; and a first application that, when executed by the at least one computing device, causes the at least one computing device to at least: generate a notification in a notification area of a display of the at least one computing device, the notification related to a requested authentication; obtain a user approval via the notification; and in response to receiving the user approval, transfer a security credential to a second application executed in the at least one computing device, the security credential corresponding to one of a plurality of authentication factors used to perform the requested authentication.

[0002b] According to an aspect, a method is provided. The method includes: determining, via at least one of one or more computing devices, that a first application has requested an authentication; causing, via the at least one of the

one or more computing devices, a user interface to be rendered by a second application, the user interface eliciting a user approval; receiving, via the at least one of the one or more computing devices, the user approval by the second application; automatically transferring, via the at least one of the one or more computing devices, a first authentication factor to a first application service in response to the user approval; and automatically transferring, via at least one of the one or more computing device, a second authentication factor to a second authentication service and not to the first authentication service in response to the user approval, wherein the first authentication factor and the second authentication factor correspond to a plurality of authentication factors used to perform the authentication, the second authentication service acts as a proxy for the first authentication service, and the first authentication service is configured to perform the authentication using the second authentication factor received from the second authentication service and the first authentication factor received from the one or more computing devices.

[0002c] According to an aspect, a non-transitory computer-readable medium is provided. The computer-readable medium embodies a first application and a second application executable in at least one computing device. When executed, the first application and the second application cause the at least one computing device to at least: determine that the first application has requested an authentication; cause a first user interface to be rendered by the second application, the first user interface eliciting a user approval of the authentication; generate a one-time password by the second application; automatically transfer the one-time password to the first application in response to receiving the user approval; cause a second user interface to be rendered by the first application, the second user interface including a first form field prepopulated with the one-time password and a second form field configured to receive a user-specified security credential; submit the user-specified security credential to a first authentication service; and submit the one-time password to a second authentication service and not directly to the first authentication service, the second authentication service acting as a proxy for the first authentication service, wherein the first authentication service is configured to perform the authentication using the one-time password received from the second

authentication service and the user-specified security credential received from the at least one computing device

[0002d] According to an aspect, a system is provided. The system includes: at least one computing device; a first application executable in the at least one computing device, wherein when executed the first application causes the at least one computing device to at least: send an authentication request to a first authentication service, the authentication request specifying a first authentication factor; and a second application executable in the at least one computing device, wherein when executed the second application causes the at least one computing device to at least: generate a user interface on a display of the at least one computing device, the user interface facilitating entry of a user approval; and in response to receiving the user approval, send a second authentication factor to a second authentication service and not directly to the first authentication service, wherein the second authentication service operates as a proxy for the first authentication service, and the first authentication service is configured to perform an authentication in response to the authentication request using the second authentication factor received from the second authentication service and the first authentication factor received from the first application.

BRIEF DESCRIPTION OF THE DRAWINGS

[0003] Many aspects of the present disclosure can be better understood with reference to the following drawings. The components in the drawings are not necessarily to scale, with emphasis instead being placed upon clearly illustrating the principles of the disclosure. Moreover, in the drawings, like reference numerals designate corresponding parts throughout the several views.

[0004] FIGS. 1A and 1B are pictorial diagrams of example user interfaces rendered by mobile devices according to various embodiments of the present disclosure.

[0005] FIGS. 2A and 2B are schematic block diagrams of networked environments according to various embodiments of the present disclosure.

[0006] FIGS. 3-5 are flowcharts illustrating examples of functionality implemented as portions of an authenticator application executed in a client in the networked environment of FIG. 2A according to various embodiments of the present disclosure.

[0007] FIG. 6 is a flowchart illustrating one example of functionality implemented as portions of a client application executed in a client in the networked environment of FIG. 2A according to various embodiments of the present disclosure.

[0008] FIG. 7 is a sequence diagram illustrating one example of interaction between a client application, an authenticator application, an authentication endpoint, and an authentication proxy in the networked environment of FIG. 2B according to various embodiments of the present disclosure.

[0009] FIG. 8 is a schematic block diagram that provides one example illustration of a computing environment employed in the networked environment of FIG. 2A according to various embodiments of the present disclosure.

DETAILED DESCRIPTION

[0010] The present disclosure relates to approaches for providing multi-factor authentication credentials. Multi-factor authentication is increasingly being deployed for use in logging into network sites or applications. While in times past a username and a password might have been all that was required to authenticate, network sites and applications now may require an additional authentication factor to be provided. In addition to having to prove what he or she knows (*i.e.*, a password), a user may now have to prove what he or she has. This may involve one-time passwords that are generated by a user's device and/or sent to the user via a trusted communication channel (*e.g.*, an email to a known address, a text message to a known telephone number, a voice call to a known telephone number, *etc.*).

[0011] The user may be asked to enter a one-time password into a network page or application form along with other security credentials, such as a username and password. This may be a simple procedure if the one-time password is displayed on one device and entered into a form on another device.

In such a scenario, the user may simply view and type the one-time password. However, if the one-time password is presented by a first application and entered by the user via a second application, the user may have to memorize the one-time password shown by the first application, switch to the second application, and then attempt to enter the one-time password from memory. Such a process may involve inconveniently switching back and forth between the second application and the first application to enter portions of the one-time password, as the one-time password may stretch the bounds of the user's working memory.

[0012] Various embodiments of the present disclosure leverage device notifications in order to ease user entry of one-time passwords and/or other multi-factor authentication credentials that are obtained through different applications. Under the mobile application paradigm, which may be employed by operating systems that are not executed in mobile devices, the user interface of an application fills most of the device's display. Multi-tasking thus involves switching which single application is rendered upon the display, rather than using a windowed paradigm that may be inefficient for smaller display sizes. Nonetheless, a notification area rendered by the operating system may be shown along with the user interface of the application having focus. The notification area may show the current time, wireless network signal strength, battery condition, and other status information. As will be described herein, a one-time password or other credential may be shown in the notification area, which the user may then view and type into an application form field in order to perform multi-factor authentication.

[0013] In another embodiment, an approval request may be shown in the notifications area, and upon approval, the one-time password or other credential may be automatically transferred to the application that requests it, or to a server on behalf of the application. This automatic transfer may avoid the multiple context switches that might otherwise be required, with a user manually switching to an authenticator application and then switching back to the client application.

[0014] In other approaches, a proxy server may be used to communicate one or more authentication factors to or from an authentication endpoint, while

one or more other authentication factors may be communicated by a client device to or from the authentication endpoint directly. The use of a proxy server that does not actually perform authentication functions may improve security. For example, if the proxy server were compromised by malicious actors, it may be that one-time passwords are intercepted. However, the one-time passwords are of limited utility by themselves, as they are time bound. Without the system to generate or validate the one-time passwords, a compromise of the proxy server alone would not allow authentication of malicious actors.

[0015] Referring to FIG. 1A, shown is an example mobile device 100a that facilitates entry of sign-in information in accordance with various embodiments of the present disclosure. The mobile device 100a includes a touchscreen display 103. Rendered upon the touchscreen display 103 are an application user interface 106 and a notification area 109. The application user interface 106 corresponds to the currently executing application that is given focus in the mobile device 100. In this example, the application user interface 106 corresponds to a mobile banking application ("Mobile Banking") and includes a sign-in form 112 that requests multiple authentication factors comprising a username, a password, and a verification code corresponding to a one-time password.

[0016] A user has selected the notification area 109 to be expanded. Within the expanded notification area 109 is a component 115 that, when selected, enables a one-time password corresponding to the mobile banking application to be entered into the sign-in form 112 for the verification code. Where multiple accounts are associated with the particular application or sign-in form, the user may select from among the multiple accounts using respective components 115 in the notification area 109. While in this example the component 115 is within an expanded notification area 109, in other examples the component 115 may be within a non-expanded notification area 109.

[0017] Turning now to FIG. 1B, shown is an example mobile device 100b that facilitates entry of sign-in information in accordance with various embodiments of the present disclosure. As compared to FIG. 1A, rather than including an approval component 115, the one-time password is itself displayed within the notification area 109 as component 118. In this example, the

notification area 109 is non-expanded and the sign-in form 112 has focus. Consequently, the user can view the one-time password in the component 118 at the same time as entering the one-time password in a form field of the sign-in form 112. In the following discussion, a general description of the system and its components is provided, followed by a discussion of the operation of the same.

[0018] With reference to FIG. 2A, shown is a networked environment 200 according to various embodiments. The networked environment 200 includes a computing environment 203 and one or more clients 206 in data communication via a network 209. The network 209 includes, for example, the Internet, intranets, extranets, wide area networks (WANs), local area networks (LANs), wired networks, wireless networks, cable networks, satellite networks, or other suitable networks, or any combination of two or more such networks.

[0019] The computing environment 203 may comprise, for example, a server computer or any other system providing computing capability. Alternatively, the computing environment 203 may employ a plurality of computing devices that are arranged, for example, in one or more server banks or computer banks or other arrangements. Such computing devices may be located in a single installation or may be distributed among many different geographical locations. For example, the computing environment 203 may include a plurality of computing devices that together may comprise a hosted or "cloud" computing resource, a grid computing resource, and/or any other distributed computing arrangement. In some cases, the computing environment 203 may correspond to an elastic computing resource where the allotted capacity of processing, network, storage, or other computing-related resources may vary over time.

[0020] Various applications and/or other functionality may be executed in the computing environment 203 according to various embodiments. Also, various data is stored in a data store 212 that is accessible to the computing environment 203. The data store 212 may be representative of a plurality of data stores 212 as can be appreciated. The data stored in the data store 212, for example, is associated with the operation of the various applications and/or functional entities described below.

[0021] The components executed on the computing environment 203, for example, include an authentication endpoint 215 and other applications,

services, processes, systems, engines, or functionality not discussed in detail herein. The authentication endpoint 215 is executed to facilitate authentication for a network resource. This authentication may take the form of multi-factor authentication. For example, the authentication endpoint 215 may receive traditional knowledge-based credentials such as username and password from the client 206, but the authentication endpoint 215 may also require an additional possession factor, such as a one-time password.

[0022] In one embodiment, the client 206 is configured to send a credential request 218 specifying a registration credential 221 to the authentication endpoint 215. Assuming that the registration credential 221 is valid, the authentication endpoint 215 may return a security credential 224 that may correspond to a one-time password. The registration credential 221 may correspond to a long-lived credential indicating authentication of the client 206 to receive the security credential 224. In another embodiment, the authentication endpoint 215 may receive shared seed requests 225 from clients 206 and then return shared seeds 226 for use in generating security credentials 224, assuming that the clients 206 are otherwise authenticated.

[0023] The authentication endpoint 215 may employ protocols such as hypertext transfer protocol (HTTP), simple object access protocol (SOAP), representational state transfer (REST), and/or other protocols. In one embodiment, the authentication endpoint 215 may comprise a commercially available HTTP server such as, for example, Apache® HTTP Server, Apache® Tomcat®, Microsoft® Internet Information Services (IIS), and others.

[0024] The data stored in the data store 212 includes, for example, shared seeds 226, registration credentials 221, and potentially other data. Additional stored data may include, for example, sign-in information, usernames, passwords, real names, network site names and domain names, uniform resource locators (URLs), payment instrument information (*e.g.*, credit card numbers and associated information, checking account information, and so on), identification numbers, security keys, birthdates, answers to knowledge-based security questions, names of pets, birthplaces, and/or other information.

[0025] The client 206 is representative of a plurality of client devices that may be coupled to the network 209. The client 206 may comprise, for example,

a processor-based system such as a computer system. Such a computer system may be embodied in the form of desktop computers, laptop computers, kiosks, personal digital assistants, cellular telephones, smartphones, set-top boxes, music players, web pads, tablet computer systems, game consoles, electronic book readers, point of sale (POS) terminals, credit card terminals, or other devices with like capability. The client 206 may include a display 233. The display 233 may comprise, for example, one or more devices such as touchscreens, liquid crystal display (LCD) displays, gas plasma-based flat panel displays, organic light emitting diode (OLED) displays, electrophoretic ink (E ink) displays, LCD projectors, or other types of display devices.

[0026] The client 206 may be configured to execute various applications and systems such as an authenticator application 236, client applications 239, an operating system 240, a notification area widget 241, and/or other applications. In one instance, the authenticator application 236 is executed to render a user interface 242 on the display 233 that enables users to configure the authenticator application 236 to provide security credentials 224, such as one-time passwords, for accounts. The authenticator application 236 may be configured to generate such security credentials 224 using shared seeds 243 or may be configured to obtain such credentials from a server, such as the authentication endpoint 215, by way of registration credentials 221.

[0027] The client applications 239 may be executed in a client 206, for example, to access network content served up by the computing environment 203 and/or other servers, thereby rendering a user interface 242 on the display 233. Each client application 239 may, for example, correspond to a browser, a mobile application, a shopping application, a banking application, a music playing application, a video application, a gaming application, a social application, and/or other applications that may request information from users to operate from time to time. The client applications 239 may correspond to native applications, web applications, hybrid applications, or other applications.

[0028] The authenticator application 236 may be integrated with the operating system 240 such that security credentials 224 are made available as users execute the various client applications 239. For example, the operating system 240 may include a notification area widget 241, such as the status bar

found on the Android® platform. The notification area widget 241 may be accessible via the display 233 while a user interface 242 of a client application 239 is being rendered upon the display 233. The authenticator application 236 may add one or more notification components to the notification area widget 241, such that selection of a notification component may approve security credentials 224 to be provided to the currently executing client application 239. Alternatively, the notification area widget 241 may be configured to render a security credential 224 so that a user may enter it into a form field of the user interface 242 of the client application 239 manually.

[0029] Techniques for providing stored account information to applications via a status bar are described in U.S. Patent Publication no. 2016/0337344 entitled "PROVIDING MULTI-FACTOR AUTHENTICATION CREDENTIALS VIA DEVICE NOTIFICATIONS". Techniques for rendering authentication codes upon a display are described in U.S. Patent no. 9,473,491, entitled "COMPUTING DEVICE WITH INTEGRATED AUTHENTICATION TOKEN".

[0030] With reference to FIG. 2B, shown is another networked environment 250 according to various embodiments. The networked environment 250 includes a computing environment 203, a computing environment 252, and one or more clients 206 in data communication via a network 209. The computing environment 252 may be similar to the computing environment 203 but may be operated by a different entity and/or may be maintained with some level of network separation from the computing environment 203.

[0031] Executed upon the computing environment 252 is an authentication proxy 254. The authentication proxy 254 may be used to provide a level of separation from generation and/or interpretation of security credentials 224 (FIG. 2A). In this embodiment, the authenticator application 236 on the client 206 communicates with the authentication proxy 254 via an authentication request 256. The authentication proxy 254 simply passes the authentication request 256 to the authentication endpoint 215 of the computing environment 203 via a proxied authentication request 258. The authentication endpoint 215 generates an authentication response 260, which is returned to the authentication proxy

254. The authentication proxy 254 returns the authentication response 260 to the client 206 as the proxied authentication response 262. Alternatively, rather than being initiated by the client 206, the communication described above may be initiated by the authentication endpoint 215.

[0032] The proxy architecture of FIG. 2B may be used to provide a layer of separation for the communication and receipt of one of multiple factors in multi-factor authentication. In one scenario, the authenticator application 236 may communicate with the authentication proxy 254 to authorize access by a client application 239, where the client application 239 has already provided a different authentication factor to the authentication endpoint 215. For example, a push notification may be sent from the authentication endpoint 215 to the authenticator application 236 by way of the authentication proxy 254.

[0033] The user may approve the authentication using the authenticator application 236, with the approval being sent via the authentication proxy 254 to the authentication endpoint 215. The approval may include a one-time password, or the use of a public-private key pair. In some examples, the public-private key pair may be built-in to the client 206, such as with ACTIVE DIRECTORY. The authentication proxy 254 does not perform the verification of the result returned by the client 206 (*i.e.*, the authentication request 256). To the contrary, the authentication proxy 254 may be incapable of validating or verifying the authentication factor supplied by the client 206. In this regard, the keys and/or shared seed 226 to perform this verification may be maintained at the computing environment 203. The authentication endpoint 215 may perform this verification, or the verification may be handled by a separate service (*e.g.*, a Remote Authentication Dial In User Service (RADIUS) server) in the computing environment 203.

[0034] Meanwhile, additional authentication factors (*e.g.*, a password) are provided by the client 206 to the authentication endpoint 215 directly, without proxying by the authentication proxy 254. In some implementations, the authentication endpoint 215 may enforce a requirement that the authenticator application 236 and the client application 239 to be authenticated be executed by separate clients 206. This may be enforced with reference to internet protocol (IP) address, media access control (MAC) address, operating system identifiers,

device identifiers, and/or other types of identifiers that may uniquely identify a client 206.

[0035] Referring next to FIG. 3, shown is a flowchart that provides one example of the operation of a portion of the authenticator application 236 according to an embodiment. It is understood that the flowchart of FIG. 3 provides merely an example of the many different types of functional arrangements that may be employed to implement the operation of the portion of the authenticator application 236 as described herein. As an alternative, the flowchart of FIG. 3 may be viewed as depicting an example of elements of a method implemented in the client 206 according to one or more embodiments.

[0036] Beginning with box 303, the authenticator application 236 acquires a shared seed 243 (FIG. 2A) for generating a security credential 224 (FIG. 2A). For example, the authenticator application 236 may acquire the shared seed 243 from an image containing a quick-response (QR) code, a barcode, machine readable text, *etc.* The shared seed 243 may also be transmitted via near-field communication (NFC), BLUETOOTH, WI-FI, infrared, acoustic coupling, and/or other wireless technologies. In some cases, acquiring a shared seed 243 may involve generating a seed and sharing it with the authentication endpoint 215 via the network 209 (FIG. 2A). In box 306, the authenticator application 236 associates the shared seed 243 with one or more client applications 239 (FIG. 2A) or accounts. It may be that different shared seeds 243 may be used for different accounts or client applications 239.

[0037] In box 309, the authenticator application 236 obtains a request to generate a security credential 224. The request may be received from a client application 239. For example, the authenticator application 236 may be associated with a particular scheme name (*e.g.*, "authenticator"), and the client application 239 may invoke the authenticator application 236 via a uniform resource locator (URL) such as "authenticator:// generatePassword/appID=UID," where UID corresponds to a unique identifier for a particular client application 239. An account identifier may also be specified. In another scenario, the request may be pushed by way of the authentication endpoint 215 (FIG. 2A) to the registered authenticator application 236.

[0038] In box 312, the authenticator application 236 determines a shared seed 243 associated with the client application 239. In box 315, the authenticator application 236 generates a security credential 224 based at least in part on the current time and the determined shared seed 243. Other algorithms to generate the security credential 224 may be used in other embodiments.

[0039] In box 318, the authenticator application 236 causes an approval component to be rendered as a notification in the notification area widget 241 (FIG. 2A), similarly to the example of FIG. 1A. The approval component may indicate to the user that a particular client application 239 has requested access to a particular security credential 224. The user may choose to approve or deny the request to transfer the security credential 224 to the client application 239.

[0040] In box 321, the authenticator application 236 receives an indication of user approval. In one example, a user may simply select a button or other user interface component to indicate approval. In another example, the user may have to provide a fingerprint or other biometric identifier to further safeguard the approval process. The authenticator application 236 may be configured to verify the biometric identifier before deeming a user approval to have occurred. The client 206 may incorporate fingerprint recognition technology in the touchscreen display 233, or a separate device may be present in the client 206 to receive fingerprints. In some cases where fingerprint recognition is required, the authenticator application 236 may require recognition of specific fingers corresponding to a particular user. In yet another example, the user may be required to enter a personal identification number (PIN) or other identifier to facilitate user approval. In various scenarios, the user approval may correspond to a parental control, such that a fingerprint or other identifier of a parent may be required before a child user is able to access an application or resource via a transfer of security credentials 224.

[0041] In box 324, the authenticator application 236 transfers the security credential 224 to the client application 239 in response to the user approval. For example, the security credential 224 may be automatically transferred by the authenticator application 236 invoking a URL with a scheme name associated with the particular client application 239, *e.g.*, "clientApplication://code=314159,"

where "314159" is the security credential 224. In some cases, the authenticator application 236 may be configured to automatically transfer the security credential 224 to a server. Thereafter, the portion of the authenticator application 236 ends.

[0042] Moving on to FIG. 4, shown is a flowchart that provides one example of the operation of a portion of the authenticator application 236 according to another embodiment. It is understood that the flowchart of FIG. 4 provides merely an example of the many different types of functional arrangements that may be employed to implement the operation of the portion of the authenticator application 236 as described herein. As an alternative, the flowchart of FIG. 4 may be viewed as depicting an example of elements of a method implemented in the client 206 according to one or more embodiments.

[0043] Beginning with box 403, the authenticator application 236 acquires a shared seed 243 (FIG. 2A) for generating a security credential 224 (FIG. 2A). For example, the authenticator application 236 may acquire the shared seed 243 from an image containing a quick-response (QR) code, a barcode, machine readable text, *etc.* The shared seed 243 may also be transmitted via near-field communication (NFC), BLUETOOTH, WI-FI, infrared, acoustic coupling, and/or other wireless technologies. In some cases, acquiring a shared seed 243 may involve generating a seed and sharing it with the authentication endpoint 215 via the network 209 (FIG. 2A). In box 406, the authenticator application 236 associates the shared seed 243 with one or more client applications 239 (FIG. 2A) or accounts. It may be that different shared seeds 243 may be used for different accounts or client applications 239.

[0044] In box 409, the authenticator application 236 obtains a request to generate a security credential 224. The request may be received from a client application 239. For example, the authenticator application 236 may be associated with a particular scheme name (*e.g.*, "authenticator"), and the client application 239 may invoke the authenticator application 236 via a uniform resource locator (URL) such as "authenticator:// generatePassword/appID=UID," where UID corresponds to a unique identifier for a particular client application 239. An account identifier may also be specified. In another scenario, the

request may be pushed by way of the authentication endpoint 215 (FIG. 2A) to the registered authenticator application 236.

[0045] In box 412, the authenticator application 236 determines a shared seed 243 associated with the client application 239. In box 415, the authenticator application 236 generates a security credential 224 based at least in part on the current time and the determined shared seed 243. Other algorithms to generate the security credential 224 may be used in other embodiments.

[0046] In box 418, the authenticator application 236 causes the generated security credential 224 to be rendered as a notification in the notification area widget 241, similarly to the example of FIG. 1B. The user may then view the security credential 224 and enter it manually into a form field of a user interface 242 (FIG. 2A) rendered by the client application 239. Thus, the notification may be rendered concurrently on the display 233 (FIG. 2A) along with the user interface 242 of the client application 239.

[0047] This may be significant especially from the standpoint of operating systems 240 that are configured to render at most one application user interface 242 at a time upon the display 233, such as with mobile devices. For example, the user interface 242 and the notification area widget 241 may be configured to fill an entirety of the display 233 when the security credential 224 is rendered in the notification area widget 241.

[0048] In box 421, the authenticator application 236 may cause the notification containing the security credential 224 to be removed from the notification area widget 241. This may be done after an elapse of a predefined amount of time or in response to a communication indicating that the security credential 224 has been used. For example, the client application 239 or the authentication endpoint 215 (FIG. 2A) may have a callback to the authenticator application 236 that indicates that the security credential 224 has been used. The authenticity of the communication from the authentication endpoint 215 may be confirmed via the use of X.509 certificates, secure sockets layer (SSL), and/or other technologies. A push or pull communication arrangement between the authenticator application 236 and the authentication endpoint 215 or client application 239 may be used by the authenticator application 236 to receive this

communication. Thereafter, the portion of the authenticator application 236 ends.

[0049] Continuing to FIG. 5, shown is a flowchart that provides one example of the operation of a portion of the authenticator application 236 according to still another embodiment. It is understood that the flowchart of FIG. 5 provides merely an example of the many different types of functional arrangements that may be employed to implement the operation of the portion of the authenticator application 236 as described herein. As an alternative, the flowchart of FIG. 5 may be viewed as depicting an example of elements of a method implemented in the client 206 according to one or more embodiments.

[0050] Beginning with box 503, the authenticator application 236 receives a registration credential 221 (FIG. 2A) for obtaining a security credential 224 (FIG. 2A) from an authentication endpoint 215 (FIG. 2A). The registration credential 221 may be a long-lived credential. The client 206 may be authenticated in order to receive the registration credential 221. In box 506, the authenticator application 236 associates the registration credential 221 with one or more client applications 239 (FIG. 2A) or accounts.

[0051] In box 509, the authenticator application 236 receives a request for a security credential 224. For example, the authenticator application 236 may be associated with a particular scheme name (*e.g.*, "authenticator"), and the client application 239 may invoke the authenticator application 236 via a uniform resource locator (URL) such as "authenticator:// generatePassword/appID=UID," where UID corresponds to a unique identifier for a particular client application 239. An account identifier may also be specified. In another scenario, the request may be pushed by way of the authentication endpoint 215 (FIG. 2A) to the registered authenticator application 236. In box 512, the authenticator application 236 determines a registration credential 221 associated with the client application 239.

[0052] In box 515, the authenticator application 236 requests a security credential 224 from the authentication endpoint 215 (FIG. 2A) via the network 209 (FIG. 2A) based at least in part on the registration credential 221. In other words, the authenticator application 236 is able to authenticate with the authentication endpoint 215 using the registration credential 221 for the purpose

of receiving the security credential 224. In box 518, the authenticator application 236 either presents the security credential 224 to the user as in box 418 of FIG. 4 or transfers the security credential 224 to the client application 239 as in boxes 318 through 324 of FIG. 3. In some cases, the authenticator application 236 may be configured to automatically transfer the security credential 224 to a server. Thereafter, the portion of the authenticator application 236 ends.

[0053] Referring next to FIG. 6, shown is a flowchart that provides one example of the operation of a portion of the client application 239 according to various embodiments. It is understood that the flowchart of FIG. 6 provides merely an example of the many different types of functional arrangements that may be employed to implement the operation of the portion of the client application 239 as described herein. As an alternative, the flowchart of FIG. 6 may be viewed as depicting an example of elements of a method implemented in the client 206 (FIG. 2A) according to one or more embodiments.

[0054] Beginning with box 603, the client application 239 determines that a security credential 224 (FIG. 2A) is required. In box 603, the client application 239 may request the security credential 224 from an authenticator application 236 (FIG. 2A). Alternatively, the client application 239 may request the security credential 224 from a server. In some scenarios, the security credential 224 may be specific to the particular client application 239 or an account used by the client application 239. For example, the client application 239 may invoke the authenticator application 236 via a uniform resource locator (URL) such as "authenticator://generatePassword/ appID=UID," where UID corresponds to a unique identifier for the particular client application 239. Alternatively, the client application 239 may simply render a user interface 242 that prompts a user to enter the security credential 224, and the user may manually invoke the authenticator application 236 to provide the security credential 224. If the client application 239 and the authenticator application 236 are executed on different clients 206, a direct network connection may be established between the different clients 206 to request and/or exchange the security credential 224.

[0055] In box 606, the client application 239 determines whether the security credential 224 is to be automatically transferred. If the security credential 224 is to be automatically transferred, the client application 239 receives the security

credential 224 from the authenticator application 236 in box 612. For example, the authenticator application 236 may invoke a URL with a scheme name associated with the particular client application 239, *e.g.*, "clientApplication://code=314159," where "314159" is the security credential 224. With automatic transfer, unnecessary context switches between the client application 239 and the authenticator application 239 are avoided. In some cases, the security credential 224 may be received from a server that is in communication with the authenticator application 239 via the network 209. Upon automatic transfer, a form field of a user interface 242 may be prepopulated with the security credential 224. The client application 239 proceeds to box 615.

[0056] If the client application 239 does not use automatic transfer to receive the security credential 224, the client application 239 moves from box 609 to box 618 and receives the security credential 224 manually from the user via a form field of a user interface 242. The notification area widget 241 (FIG. 2A) may render a notification component showing the security credential 224, and the user may view and enter the security credential 224 at the same time. The client application 239 then moves to box 615.

[0057] In box 615, the client application 239 authenticates with the authentication endpoint 215 (FIG. 2A) based at least in part on the security credential 224. In one example, the security credential 224 is just one of several authenticator factors to be provided to the authentication endpoint 215 to successfully authenticate. The user interface 242 of the client application 239 may elicit other factors such as a username, password, biometric factors, and so on. Upon authentication, the client application 239 may be enabled to access secured network resources. Thereafter, the portion of the client application 239 ends.

[0058] Moving on to FIG. 7, shown is a sequence diagram 700 that provides one example of interaction between the client application 239, the authenticator application 236, the authentication endpoint 215, and the authentication proxy 254 according to various embodiments. It is understood that the sequence diagram 700 of FIG. 7 provides merely an example of the many different types of functional arrangements as described herein.

[0059] Beginning with box 703, the client application 239 generates an authentication request specifying a first authentication factor. The authentication request is then sent to the authentication endpoint 215. For example, the client application 239 may request authentication by a username and password combination. Alternatively, the client application 239 may merely inform the authentication endpoint 215 or the authenticator application 236 that authentication is requested.

[0060] In box 706, the authentication endpoint 215 requests a second authentication factor from the authenticator application 236. Alternatively, the request for the second authentication factor may originate in the client application 239. Alternatively, a user may explicitly invoke the authenticator application 236. Where the authenticator application 236 and the client application 239 are executed in different clients 206 (FIG. 2A), a discovery procedure on a local network may be employed to enable communication.

[0061] In box 709, the authenticator application 236 generates a user interface 242 configured to receive a user approval for the client application 239 to proceed with authentication. For example, the authenticator application 236 may render an approval button that when selected indicates that the user approves of the authentication of the client application 239. In box 712, the authenticator application 236 receives a user approval via the user interface 242. In box 715, the authenticator application 236 generates a second authentication factor, *e.g.*, a one-time password. Alternatively, the authenticator application 236 may receive the second authentication factor from a server.

[0062] The authenticator application 236 then sends the second authentication factor to the authentication proxy 254. The authentication proxy 254 is an authentication service that acts as a proxy for the authentication endpoint 215. The authentication proxy 254 may be incapable of verifying the second authentication factor. In box 721, the authentication proxy 254 forwards the second authentication factor to the authentication endpoint 215. By now, the authentication endpoint 215 has received multiple authentication factors for the authentication request but from multiple sources – the client application 239 and the authenticator application 236 (by way of the authentication proxy 254).

[0063] In box 724, the authentication endpoint 215 validates the first and second authentication factors. Also, the authentication endpoint 215 may verify that the first and second authentication factors were provided by way of different clients 206. In box 727, having authenticated the client application 239, the authentication endpoint 215 may return an authentication token to the client application 239. Thereafter, the sequence diagram 700 ends.

[0064] With reference to FIG. 8, shown is a schematic block diagram of the client 206 according to an embodiment of the present disclosure. The client 206 includes at least one processor circuit, for example, having a processor 803 and a memory 806, both of which are coupled to a local interface 809. The local interface 809 may comprise, for example, a data bus with an accompanying address/control bus or other bus structure as can be appreciated. The computing environment 203 (FIG. 2A) may be made up of one or more computing devices with like structure.

[0065] Stored in the memory 806 are both data and several components that are executable by the processor 803. In particular, stored in the memory 806 and executable by the processor 803 are an authenticator application 236, client applications 239, and potentially other applications. Also stored in the memory 806 may be a data store. In addition, an operating system 240 with a notification area widget 241 may be stored in the memory 806 and executable by the processor 803.

[0066] It is understood that there may be other applications that are stored in the memory 806 and are executable by the processor 803 as can be appreciated. Where any component discussed herein is implemented in the form of software, any one of a number of programming languages may be employed such as, for example, C, C++, C#, Objective C, Java®, JavaScript®, Perl, PHP, Visual Basic®, Python®, Ruby, Flash®, or other programming languages.

[0067] A number of software components are stored in the memory 806 and are executable by the processor 803. In this respect, the term "executable" means a program file that is in a form that can ultimately be run by the processor 803. Examples of executable programs may be, for example, a compiled program that can be translated into machine code in a format that can be loaded

into a random access portion of the memory 806 and run by the processor 803, source code that may be expressed in proper format such as object code that is capable of being loaded into a random access portion of the memory 806 and executed by the processor 803, or source code that may be interpreted by another executable program to generate instructions in a random access portion of the memory 806 to be executed by the processor 803, *etc.* An executable program may be stored in any portion or component of the memory 806 including, for example, random access memory (RAM), read-only memory (ROM), hard drive, solid-state drive, USB flash drive, memory card, optical disc such as compact disc (CD) or digital versatile disc (DVD), floppy disk, magnetic tape, or other memory components.

[0068] The memory 806 is defined herein as including both volatile and nonvolatile memory and data storage components. Volatile components are those that do not retain data values upon loss of power. Nonvolatile components are those that retain data upon a loss of power. Thus, the memory 806 may comprise, for example, random access memory (RAM), read-only memory (ROM), hard disk drives, solid-state drives, USB flash drives, memory cards accessed via a memory card reader, floppy disks accessed via an associated floppy disk drive, optical discs accessed via an optical disc drive, magnetic tapes accessed via an appropriate tape drive, and/or other memory components, or a combination of any two or more of these memory components. In addition, the RAM may comprise, for example, static random access memory (SRAM), dynamic random access memory (DRAM), or magnetic random access memory (MRAM) and other such devices. The ROM may comprise, for example, a programmable read-only memory (PROM), an erasable programmable read-only memory (EPROM), an electrically erasable programmable read-only memory (EEPROM), or other like memory device.

[0069] Also, the processor 803 may represent multiple processors 803 and/or multiple processor cores and the memory 806 may represent multiple memories 806 that operate in parallel processing circuits, respectively. In such a case, the local interface 809 may be an appropriate network that facilitates communication between any two of the multiple processors 803, between any processor 803 and any of the memories 806, or between any two of the

memories 806, *etc.* The local interface 809 may comprise additional systems designed to coordinate this communication, including, for example, performing load balancing. The processor 803 may be of electrical or of some other available construction.

[0070] Although the authenticator application 236, the client applications 239, the operating system 240, and other various systems described herein may be embodied in software or code executed by general purpose hardware as discussed above, as an alternative the same may also be embodied in dedicated hardware or a combination of software/general purpose hardware and dedicated hardware. If embodied in dedicated hardware, each can be implemented as a circuit or state machine that employs any one of or a combination of a number of technologies. These technologies may include, but are not limited to, discrete logic circuits having logic gates for implementing various logic functions upon an application of one or more data signals, application specific integrated circuits (ASICs) having appropriate logic gates, field-programmable gate arrays (FPGAs), or other components, *etc.* Such technologies are generally well known by those skilled in the art and, consequently, are not described in detail herein.

[0071] The flowcharts of FIGS. 3-6 and the sequence diagram of FIG. 7 show the functionality and operation of an implementation of portions of the authenticator application 236, the client application 239, the authentication endpoint 215, and/or the authentication proxy 254. If embodied in software, each block may represent a module, segment, or portion of code that comprises program instructions to implement the specified logical function(s). The program instructions may be embodied in the form of source code that comprises human-readable statements written in a programming language or machine code that comprises numerical instructions recognizable by a suitable execution system such as a processor 803 in a computer system or other system. The machine code may be converted from the source code, *etc.* If embodied in hardware, each block may represent a circuit or a number of interconnected circuits to implement the specified logical function(s).

[0072] Although the flowcharts of FIGS. 3-6 and the sequence diagram of FIG. 7 show a specific order of execution, it is understood that the order of execution may differ from that which is depicted. For example, the order of

execution of two or more blocks may be scrambled relative to the order shown. Also, two or more blocks shown in succession in FIGS. 3-7 may be executed concurrently or with partial concurrence. Further, in some embodiments, one or more of the blocks shown in FIGS. 3-7 may be skipped or omitted. In addition, any number of counters, state variables, warning semaphores, or messages might be added to the logical flow described herein, for purposes of enhanced utility, accounting, performance measurement, or providing troubleshooting aids, *etc.* It is understood that all such variations are within the scope of the present disclosure.

[0073] Also, any logic or application described herein, including the authenticator application 236, the client applications 239, and the operating system 240, that comprises software or code can be embodied in any non-transitory computer-readable medium for use by or in connection with an instruction execution system such as, for example, a processor 803 in a computer system or other system. In this sense, the logic may comprise, for example, statements including instructions and declarations that can be fetched from the computer-readable medium and executed by the instruction execution system. In the context of the present disclosure, a "computer-readable medium" can be any medium that can contain, store, or maintain the logic or application described herein for use by or in connection with the instruction execution system.

[0074] The computer-readable medium can comprise any one of many physical media such as, for example, magnetic, optical, or semiconductor media. More specific examples of a suitable computer-readable medium would include, but are not limited to, magnetic tapes, magnetic floppy diskettes, magnetic hard drives, memory cards, solid-state drives, USB flash drives, or optical discs. Also, the computer-readable medium may be a random access memory (RAM) including, for example, static random access memory (SRAM) and dynamic random access memory (DRAM), or magnetic random access memory (MRAM). In addition, the computer-readable medium may be a read-only memory (ROM), a programmable read-only memory (PROM), an erasable programmable read-only memory (EPROM), an electrically erasable programmable read-only memory (EEPROM), or other type of memory device.

[0075] Further, any logic or application described herein, including the authenticator application 236, the client applications 239, and the operating system 240, may be implemented and structured in a variety of ways. For example, one or more applications described may be implemented as modules or components of a single application. Further, one or more applications described herein may be executed in shared or separate computing devices or a combination thereof. For example, a plurality of the applications described herein may execute in the same computing device, or in multiple computing devices in the same computing environment 203. Additionally, it is understood that terms such as "application," "service," "system," "engine," "module," and so on may be interchangeable and are not intended to be limiting.

[0076] Disjunctive language such as the phrase "at least one of X, Y, or Z," unless specifically stated otherwise, is otherwise understood with the context as used in general to present that an item, term, etc., may be either X, Y, or Z, or any combination thereof (*e.g.*, X, Y, and/or Z). Thus, such disjunctive language is not generally intended to, and should not, imply that certain embodiments require at least one of X, at least one of Y, or at least one of Z to each be present.

[0077] Embodiments of the disclosure can be described in view of the following clauses:

[0078] Clause 1. A non-transitory computer-readable medium embodying a first application executable in a computing device, comprising: code that acquires a shared seed from an image; code that associates the shared seed with a second application; code that generates a one-time password based at least in part on a current time and the shared seed; code that, in response to receiving a request from the second application via a first uniform resource locator (URL), generates a notification in an operating system notification area of a display of the computing device; code that facilitates entry of a user approval via the notification; and code that, in response to receiving the user approval, transfers the one-time password to the second application via a second URL.

[0079] Clause 2. The non-transitory computer-readable medium of clause 1, wherein the first URL includes a first scheme name associated with the first

application, and the second URL includes a second scheme name associated with the second application.

[0080] Clause 3. The non-transitory computer-readable medium of clauses 1 to 2, wherein the notification includes the one-time password.

[0081] Clause 4. A system, comprising: at least one computing device; and a first application executed in the at least one computing device, the first application comprising: logic that generates a notification in a notification area of a display of the at least one computing device; logic that facilitates entry of a user approval via the notification; and logic that, in response to receiving the user approval, transfers a security credential to a second application executed in the at least one computing device.

[0082] Clause 5. The system of clause 4, wherein the logic that generates the notification is configured to generate the notification in response to a request from the second application.

[0083] Clause 6. The system of clauses 4 to 5, wherein the logic that generates the notification is configured to generate the notification in response to a request from a server.

[0084] Clause 7. The system of clauses 4 to 6, wherein the security credential is a one-time password.

[0085] Clause 8. The system of clauses 4 to 7, wherein the first application further comprises logic that receives the one-time password from a server via a network.

[0086] Clause 9. The system of clauses 4 to 8, wherein the first application further comprises logic that generates the one-time password.

[0087] Clause 10. The system of clauses 4 to 9, wherein the first application further comprises logic that acquires a seed from an image, the seed being used to generate the one-time password.

[0088] Clause 11. The system of clauses 4 to 10, wherein the first application receives a request via a first uniform resource locator (URL) having a first scheme name associated with the first application, and the second application receives the security credential via a second URL having a second scheme name associated with the second application.

[0089] Clause 12. The system of clauses 4 to 11, wherein the notification area is rendered on the display concurrently with a user interface of the second application.

[0090] Clause 13. The system of clauses 4 to 12, wherein the security credential is specific to the second application.

[0091] Clause 14. The system of clauses 4 to 13, wherein the second application employs a plurality of authentication factors, and the security credential corresponds to one of the plurality of authentication factors.

[0092] Clause 15. The system of clauses 4 to 14, wherein the at least one computing device is configured to render one application user interface upon the display at a time.

[0093] Clause 16. A method, comprising: determining, by a first application executed via at least one of one or more computing devices, a security credential; causing the security credential to be rendered in a notification area upon a display of at least one of the one or more computing devices while a user interface of a second application executed via at least one of the one or more computing devices is also rendered upon the display; and receiving, by the second application executed via at least one of the one or more computing devices, the security credential by way of a form field of the user interface.

[0094] Clause 17. The method of clause 16, further comprising sending by the second application a request to display the security credential to the first application.

[0095] Clause 18. The method of clauses 16 to 17, wherein the request to display the security credential is sent via a uniform resource locator (URL) that includes a scheme name registered to the first application.

[0096] Clause 19. The method of clauses 16 to 18, wherein determining the security credential further comprises: generating, by the first application, a one-time password based at least in part on a current time and a shared seed; and identifying, by the first application, the shared seed from a plurality of seeds based at least in part on the second application.

[0097] Clause 20. The method of clauses 16 to 19, wherein the user interface and the notification area fill an entirety of the display when the security credential is rendered in the notification area.

[0098] Clause 21. A non-transitory computer-readable medium embodying a first application and a second application executable in at least one computing device, comprising: code that determines that the first application has requested an authentication; code that causes a first user interface to be rendered by the second application, the first user interface eliciting a user approval of the authentication; code that generates a one-time password by the second application; code that automatically transfers the one-time password to the first application in response to receiving the user approval; code that causes a second user interface to be rendered by the first application, the second user interface including a first form field prepopulated with the one-time password and a second form field configured to receive a user-specified security credential; and code that submits the one-time password and the user-specified security credential to at least one authentication service to complete the authentication of the first application.

[0099] Clause 22. The non-transitory computer-readable medium of clause 21, wherein the at least one authentication service comprises a first authentication service and a second authentication service, the user-specified security credential is submitted to the first authentication service, the one-time password is submitted to the second authentication service, and the second authentication service acts as a proxy for the first authentication service.

[0100] Clause 23. A system, comprising: at least one computing device; a first application executed in the at least one computing device, the first application comprising: logic that sends an authentication request to a first authentication service, the authentication request specifying a first authentication factor; and a second application executed in the at least one computing device, the second application comprising: logic that generates a user interface on a display of the at least one computing device, the user interface facilitating entry of a user approval; and logic that, in response to receiving the user approval, sends a second authentication factor to a second authentication service, wherein the second authentication service operates as a proxy for the first authentication service.

[0101] Clause 24. The system of clause 23, wherein the first authentication service and the second authentication service are operated by different entities.

[0102] Clause 25. The system of clauses 23 to 24, wherein the second authentication service is incapable of validating the second authentication factor.

[0103] Clause 26. The system of clauses 23 to 25, wherein the logic that generates the user interface is configured to generate the user interface in response to a request from the second authentication service.

[0104] Clause 27. The system of clauses 23 to 26, wherein the logic that generates the user interface is configured to generate the user interface in response to a request from the first application.

[0105] Clause 28. The system of clauses 23 to 27, wherein the at least one computing device comprises a first computing device and a second computing device, the first application is executed in the first computing device, and the second application is executed in the second computing device.

[0106] Clause 29. The system of clauses 23 to 28, wherein the first authentication service is configured to verify that the first application and second application are executed in different computing devices.

[0107] Clause 30. The system of clauses 23 to 29, wherein the at least one computing device comprises a single computing device, and both the first application and the second application are executed in the single computing device.

[0108] Clause 31. The system of clauses 23 to 30, wherein the user interface comprises a notification in a notification area.

[0109] Clause 32. A method, comprising: determining, via at least one of one or more computing devices, that a first application has requested an authentication; causing, via at least one of the one or more computing devices, a user interface to be rendered by a second application, the user interface eliciting a user approval; receiving, via at least one of the one or more computing devices, the user approval by the second application; and automatically transferring, via at least one of the one or more computing devices, an authentication factor to a recipient in response to the user approval, wherein the authentication factor corresponds to one of a plurality of authentication factors used to perform the authentication.

[0110] Clause 33. The method of clause 32, wherein the authentication factor is transferred to the recipient via a uniform resource locator (URL) that includes a scheme name registered to the recipient.

[0111] Clause 34. The method of clauses 32 to 33, wherein the authentication factor corresponds to a one-time password.

[0112] Clause 35. The method of clauses 32 to 34, wherein another one of the plurality of authentication factors corresponds to a user-provided password.

[0113] Clause 36. The method of clauses 32 to 35, further comprising generating, by the second application, the one-time password.

[0114] Clause 37. The method of clauses 32 to 36, further comprising receiving, by the second application, the one-time password from another computing device via a network.

[0115] Clause 38. The method of clauses 32 to 37, further comprising sending, by the first application, at least one of the plurality of authentication factors to an authentication service configured to perform the authentication.

[0116] Clause 39. The method of clauses 32 to 38, wherein the recipient corresponds to the first application.

[0117] Clause 40. The method of clauses 32 to 39, wherein the recipient corresponds to an authentication service.

[0118] It should be emphasized that the above-described embodiments of the present disclosure are merely possible examples of implementations set forth for a clear understanding of the principles of the disclosure. Many variations and modifications may be made to the above-described embodiment(s) without departing substantially from the spirit and principles of the disclosure. All such modifications and variations are intended to be included herein within the scope of this disclosure and protected by the following claims.

CLAIMS

1. A system, comprising:

at least one computing device; and

a first application that, when executed by the at least one computing device, causes the at least one computing device to at least:

generate a notification in a notification area of a display of the at least one computing device, the notification related to a requested authentication;

obtain a user approval via the notification; and

in response to receiving the user approval, transfer a security credential to a second application executed in the at least one computing device, the security credential corresponding to one of a plurality of authentication factors used to perform the requested authentication.

2. The system of claim 1, wherein the first application further causes the at least one computing device to generate the notification in response to a request from one of: the second application and a server.

3. The system of claim 1, wherein the security credential is a one-time password, the first application further causes the at least one computing device to generate the one-time password.

4. The system of claim 1, wherein the first application further causes the at least one computing device to at least acquire a shared seed from an image,

associate the shared seed with the second application, and generate a one-time password based at least in part on a current time and the shared seed, and wherein the first application receives a request via a first uniform resource locator (URL) having a first scheme name associated with the first application, and the second application receives the security credential via a second URL having a second scheme name associated with the second application.

5. The system of claim 1, wherein the security credential is specific to the second application, the second application employs a plurality of authentication factors, and the security credential corresponds to one of the plurality of authentication factors.

6. The system of claim 1, wherein the notification area is rendered on the display concurrently with a user interface of the second application and the at least one computing device is configured to render one application user interface upon the display at a time.

7. A method, comprising:

determining, via at least one of one or more computing devices, that a first application has requested an authentication;

causing, via the at least one of the one or more computing devices, a user interface to be rendered by a second application, the user interface eliciting a user approval;

receiving, via the at least one of the one or more computing devices, the user approval by the second application;

automatically transferring, via the at least one of the one or more computing devices, a first authentication factor to a first authentication service in response to the user approval; and

automatically transferring, via at least one of the one or more computing device, a second authentication factor to a second authentication service and not to the first authentication service in response to the user approval, wherein the first authentication factor and the second authentication factor correspond to a plurality of authentication factors used to perform the authentication, the second authentication service acts as a proxy for the first authentication service, and the first authentication service is configured to perform the authentication using the second authentication factor received from the second authentication service and the first authentication factor received from the one or more computing devices.

8. The method of claim 7, wherein the authentication factor is transferred to the first authentication service via a uniform resource locator (URL) that includes a scheme name registered to the first authentication service.

9. The method of claim 7, wherein the second authentication factor corresponds to a one-time password, the second application generates the one-time password, and the first authentication factor corresponds to a user-provided password.

10. The method of claim 9, further comprising receiving, by the second application, the one-time password from another computing device via a network.

11. The method of claim 7, wherein the second authentication service is incapable of verifying the second authentication factor.

12. The method of claim 7, wherein the first authentication service and the second authentication service are operated by different entities.

13. The method of claim 7, wherein the first authentication factor is not transferred to the second authentication service.

14. A non-transitory computer-readable medium having instructions stored thereon which, when executed by a computer, cause the computer to carry out the method of any one of claims 7 to 13.

15. A non-transitory computer-readable medium embodying a first application and a second application executable in at least one computing device, wherein when executed the first application and the second application cause the at least one computing device to at least:

determine that the first application has requested an authentication;

cause a first user interface to be rendered by the second application, the first user interface eliciting a user approval of the authentication;

generate a one-time password by the second application;

automatically transfer the one-time password to the first application in response to receiving the user approval;

cause a second user interface to be rendered by the first application, the second user interface including a first form field prepopulated with the one-time password and a second form field configured to receive a user-specified security credential;

submit the user-specified security credential to a first authentication service; and

submit the one-time password to a second authentication service and not directly to the first authentication service, the second authentication service acting as a proxy for the first authentication service, wherein the first authentication service is configured to perform the authentication using the one-time password received from the second authentication service and the user-specified security credential received from the at least one computing device.

16. The non-transitory computer-readable medium of claim 15, wherein the at least one computing device does not submit the user-specified security credential to the second authentication service.

17. A system, comprising:

at least one computing device;

a first application executable in the at least one computing device, wherein when executed the first application causes the at least one computing device to at least:

send an authentication request to a first authentication service, the authentication request specifying a first authentication factor; and

a second application executable in the at least one computing device, wherein when executed the second application causes the at least one computing device to at least:

generate a user interface on a display of the at least one computing device, the user interface facilitating entry of a user approval; and

in response to receiving the user approval, send a second authentication factor to a second authentication service and not directly to the first authentication service, wherein the second authentication service operates as a proxy for the first authentication service, and the first authentication service is configured to perform an authentication in response to the authentication request using the second authentication factor received from the second authentication service and the first authentication factor received from the first application.

18. The system of claim 17, wherein the first authentication service and the second authentication service are operated by different entities.

19. The system of claim 17, wherein the second authentication service is incapable of validating the second authentication factor.

20. The system of claim 17, wherein the user interface is generated in response to a request from the second authentication service.

21. The system of claim 17, wherein the user interface is generated in response to a request from the first application.

22. The system of claim 17, wherein the at least one computing device comprises a first computing device and a second computing device, the first application is executed in the first computing device, and the second application is executed in the second computing device.

23. The system of claim 17, wherein the first authentication service is configured to verify that the first application and second application are executed in different computing devices.

24. The system of claim 17, wherein the at least one computing device comprises a single computing device, and both the first application and the second application are executed in the single computing device.

25. The system of claim 17, wherein the user interface comprises a notification in a notification area.

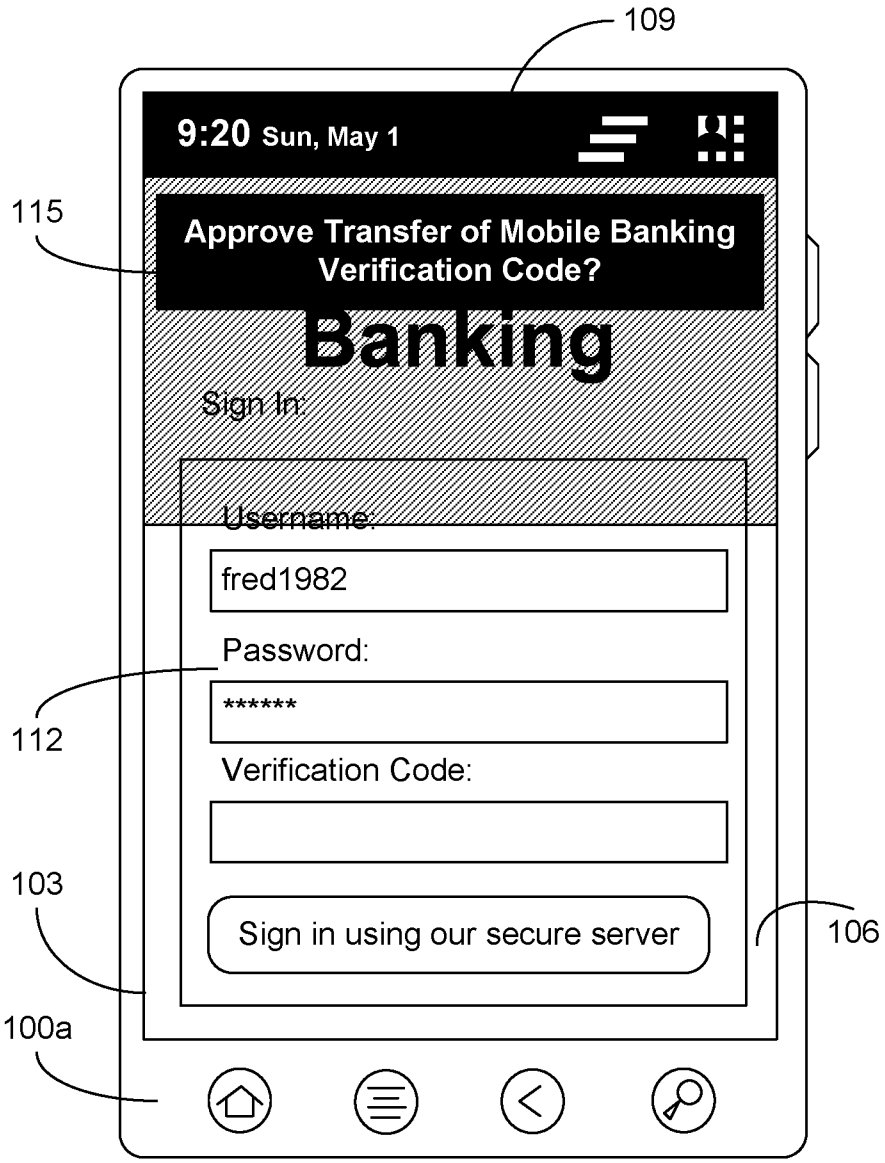

109
9:20 Sun, May 1
115
Approve Transfer of Mobile Banking Verification Code?
Banking
Sign In:
Username:
fred1982
Password:
******
112
Verification Code:
103
Sign in using our secure server
106
100a

FIG. 1A

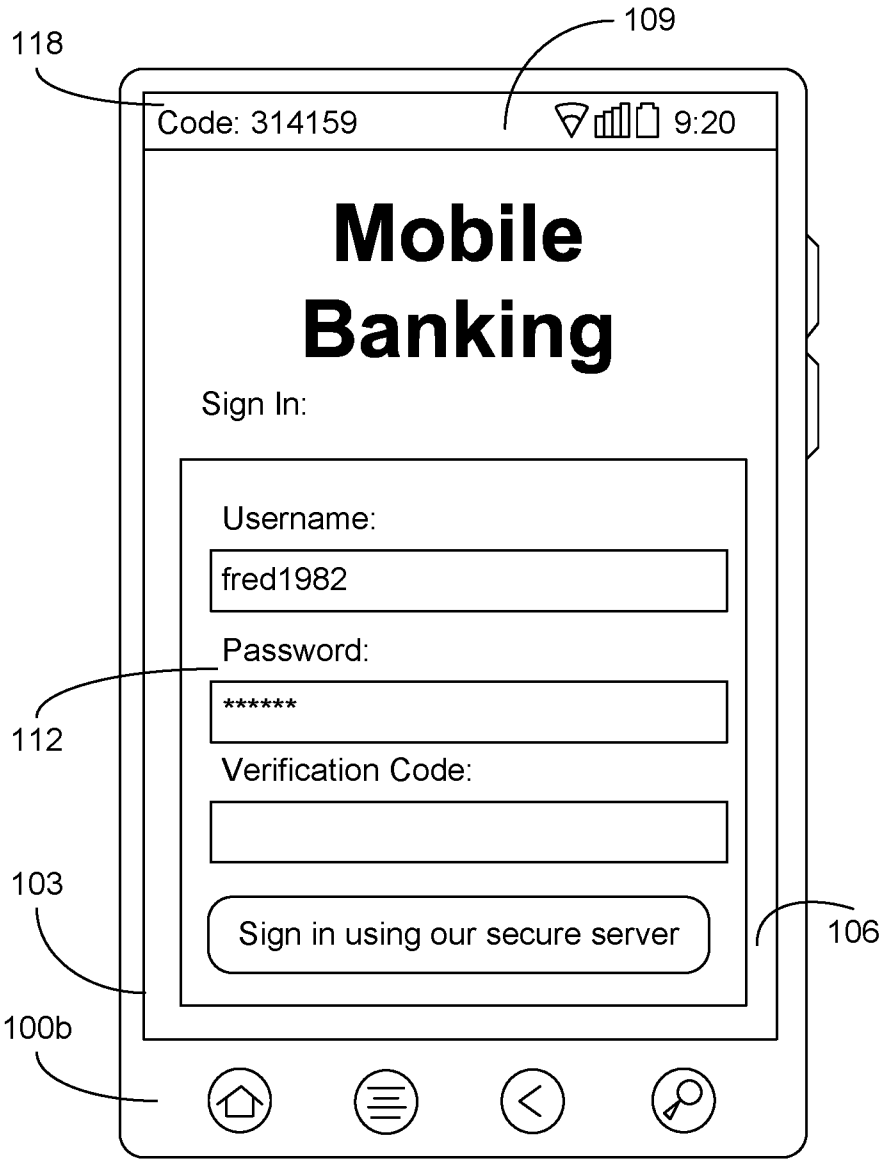

118
109
Code: 314159
9:20
Mobile Banking
Sign In:
Username:
fred1982
Password:
******
112
Verification Code:
103
Sign in using our secure server
106
100b

FIG. 1B

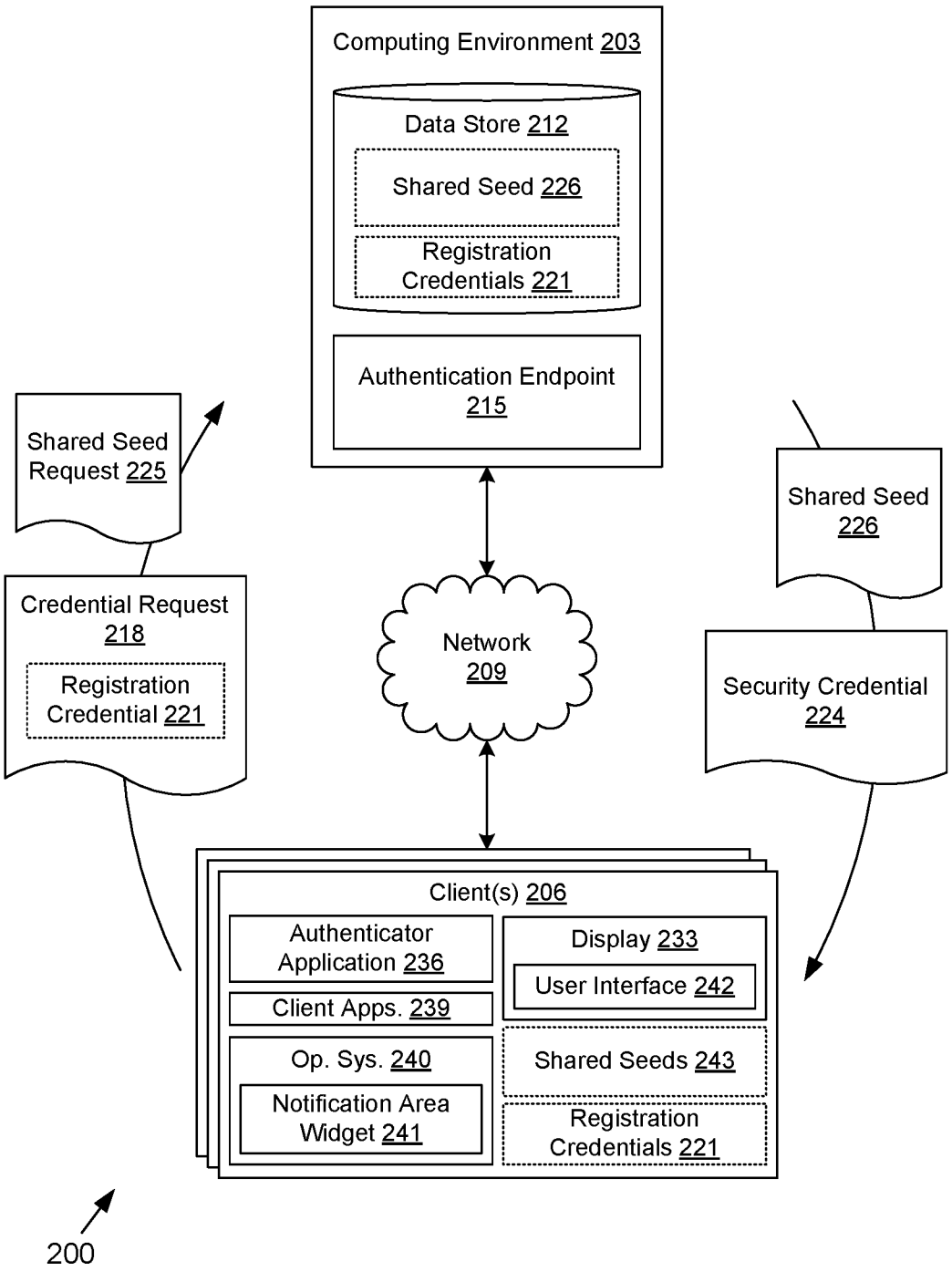

Computing Environment 203
Data Store 212
Shared Seed 226
Registration Credentials 221
Authentication Endpoint 215
Shared Seed Request 225
Credential Request 218
Registration Credential 221
Network 209
Shared Seed 226
Security Credential 224
Client(s) 206
Authenticator Application 236
Client Apps. 239
Op. Sys. 240
Notification Area Widget 241
Display 233
User Interface 242
Shared Seeds 243
Registration Credentials 221
200

FIG. 2A

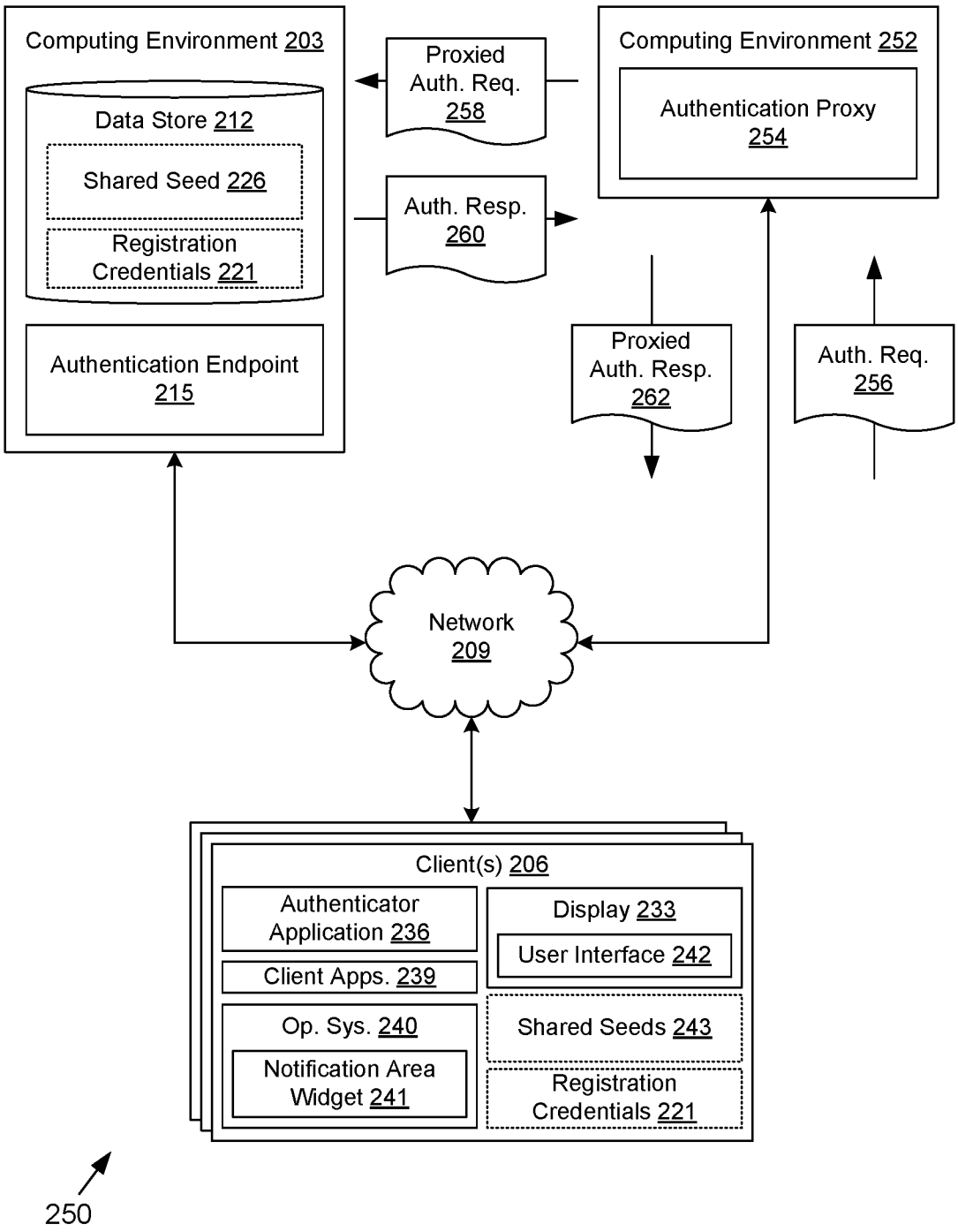

Computing Environment 203
Data Store 212
Shared Seed 226
Registration Credentials 221
Authentication Endpoint 215
Proxied Auth. Req. 258
Auth. Resp. 260
Computing Environment 252
Authentication Proxy 254
Proxied Auth. Resp. 262
Auth. Req. 256
Network 209
Client(s) 206
Authenticator Application 236
Client Apps. 239
Op. Sys. 240
Notification Area Widget 241
Display 233
User Interface 242
Shared Seeds 243
Registration Credentials 221
250

FIG. 2B

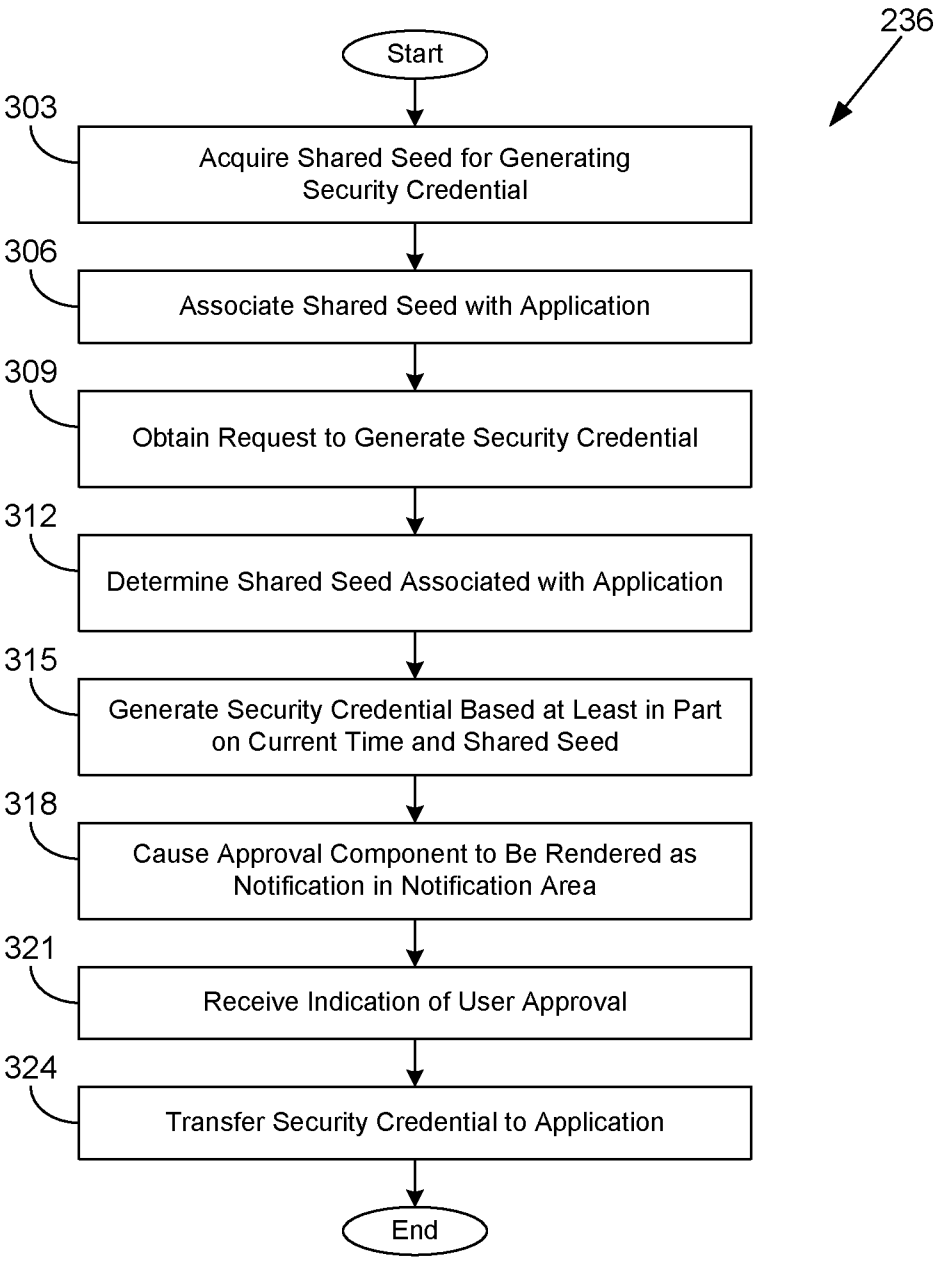

236
Start
303
Acquire Shared Seed for Generating Security Credential
306
Associate Shared Seed with Application
309
Obtain Request to Generate Security Credential
312
Determine Shared Seed Associated with Application
315
Generate Security Credential Based at Least in Part on Current Time and Shared Seed
318
Cause Approval Component to Be Rendered as Notification in Notification Area
321
Receive Indication of User Approval
324
Transfer Security Credential to Application
End

FIG. 3

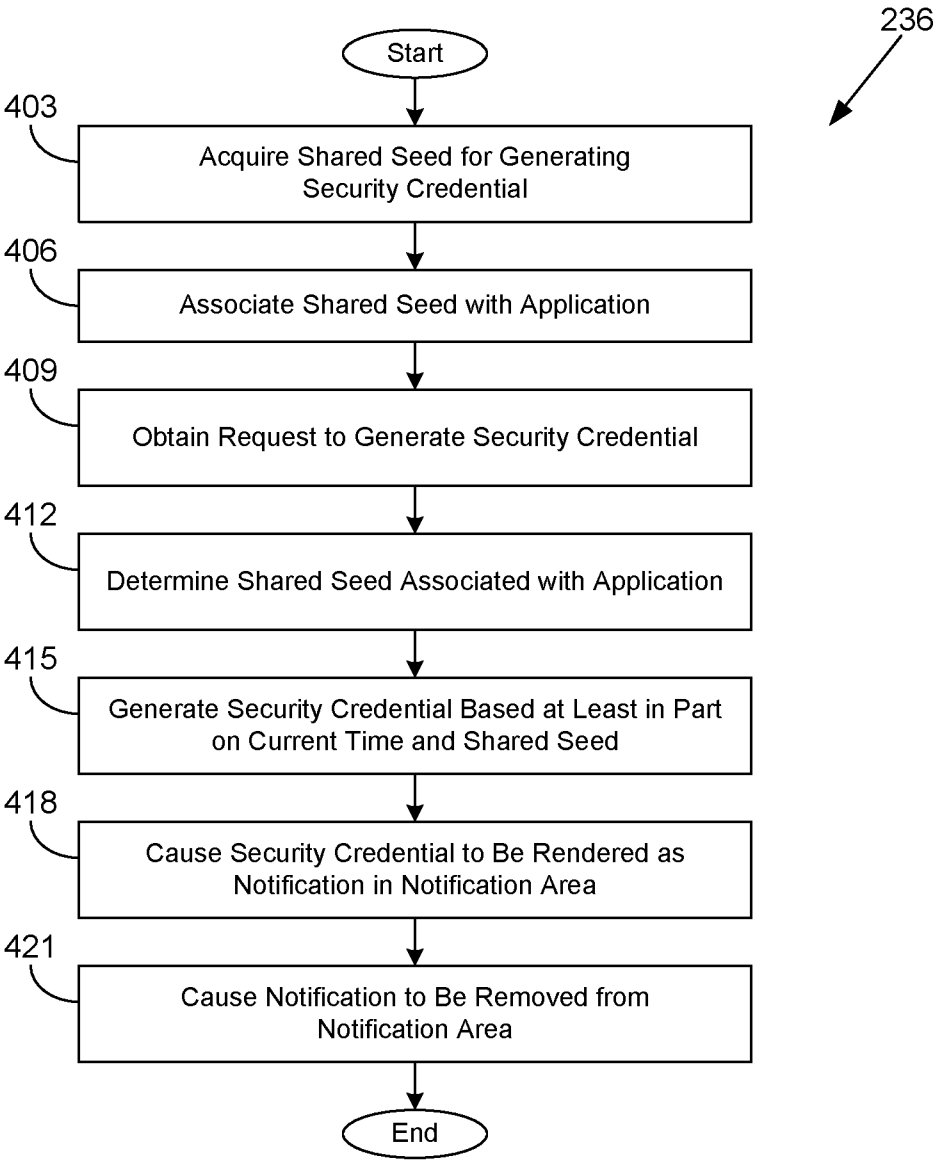

236
Start
403
Acquire Shared Seed for Generating Security Credential
406
Associate Shared Seed with Application
409
Obtain Request to Generate Security Credential
412
Determine Shared Seed Associated with Application
415
Generate Security Credential Based at Least in Part on Current Time and Shared Seed
418
Cause Security Credential to Be Rendered as Notification in Notification Area
421
Cause Notification to Be Removed from Notification Area
End

FIG. 4

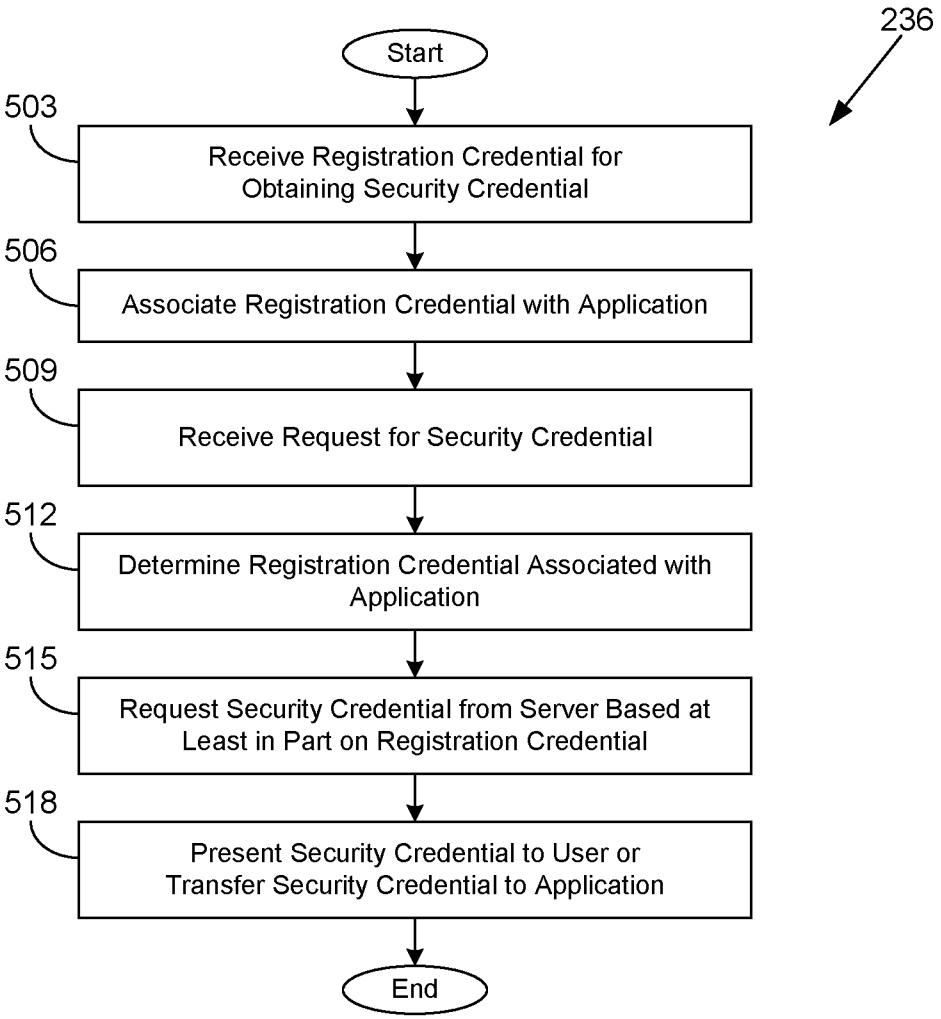

236
Start
503
Receive Registration Credential for Obtaining Security Credential
506
Associate Registration Credential with Application
509
Receive Request for Security Credential
512
Determine Registration Credential Associated with Application
515
Request Security Credential from Server Based at Least in Part on Registration Credential
518
Present Security Credential to User or Transfer Security Credential to Application
End

FIG. 5

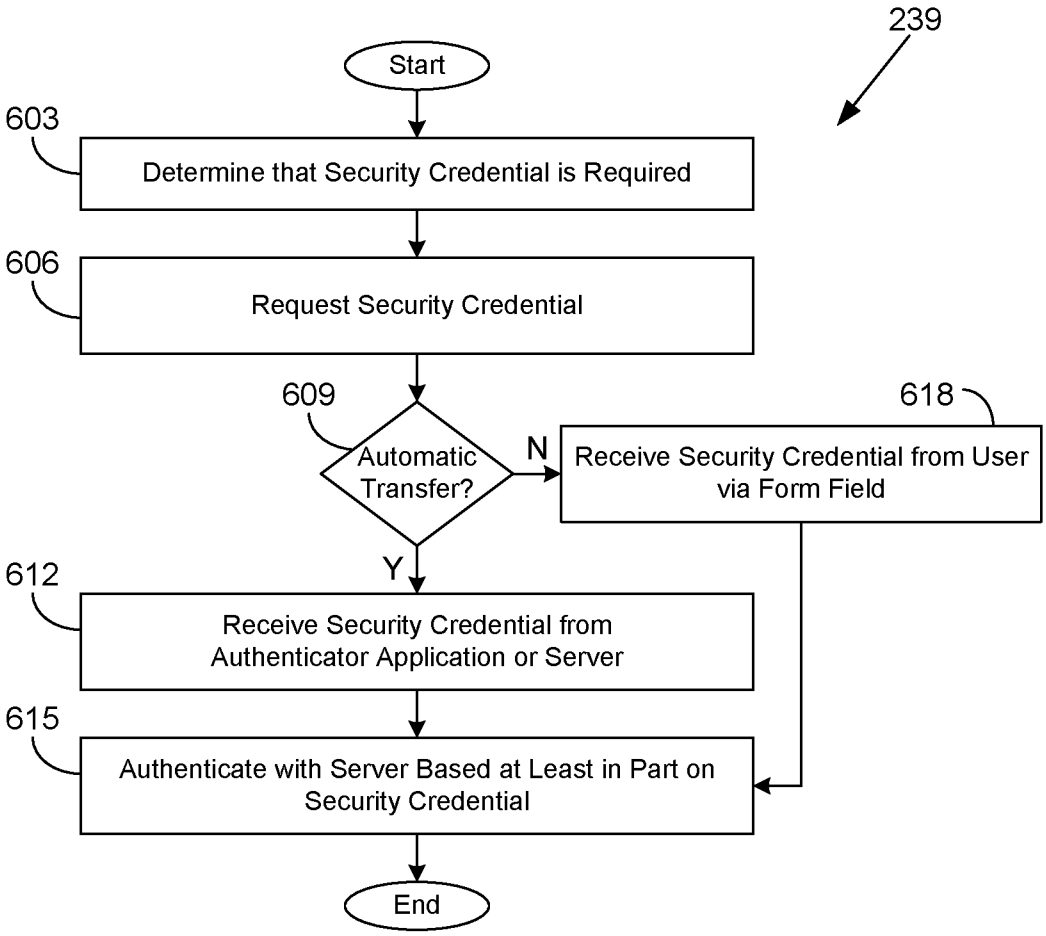

239
Start
603
Determine that Security Credential is Required
606
Request Security Credential
609
Automatic Transfer?
N
618
Receive Security Credential from User via Form Field
Y
612
Receive Security Credential from Authenticator Application or Server
615
Authenticate with Server Based at Least in Part on Security Credential
End

FIG. 6

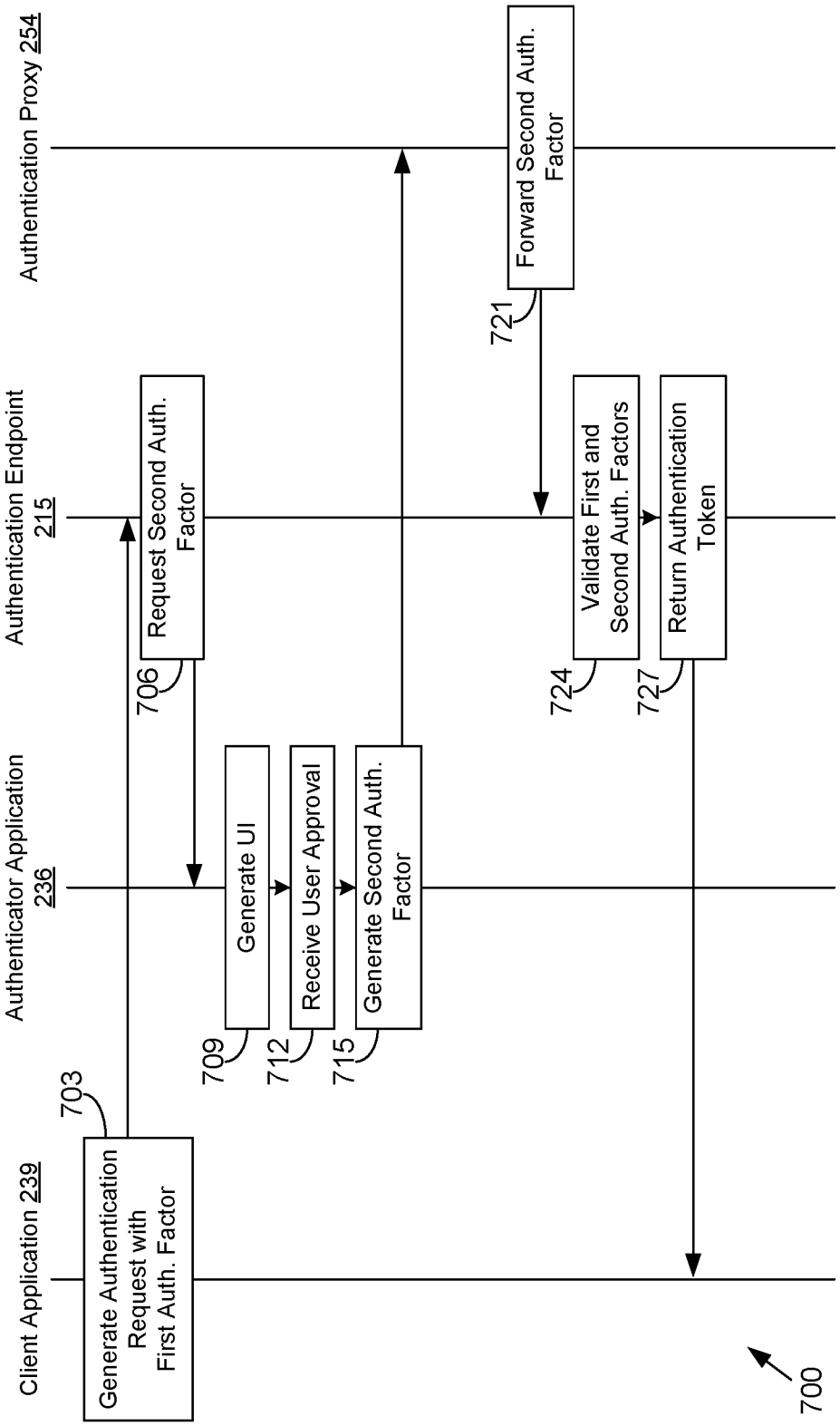

Client Application 239
Authenticator Application 236
Authentication Endpoint 215
Authentication Proxy 254
Generate Authentication Request with First Auth. Factor
703
Request Second Auth. Factor
706
Generate UI
709
Receive User Approval
712
Generate Second Auth. Factor
715
Forward Second Auth. Factor
721
Validate First and Second Auth. Factors
724
Return Authentication Token
727
700

FIG. 7

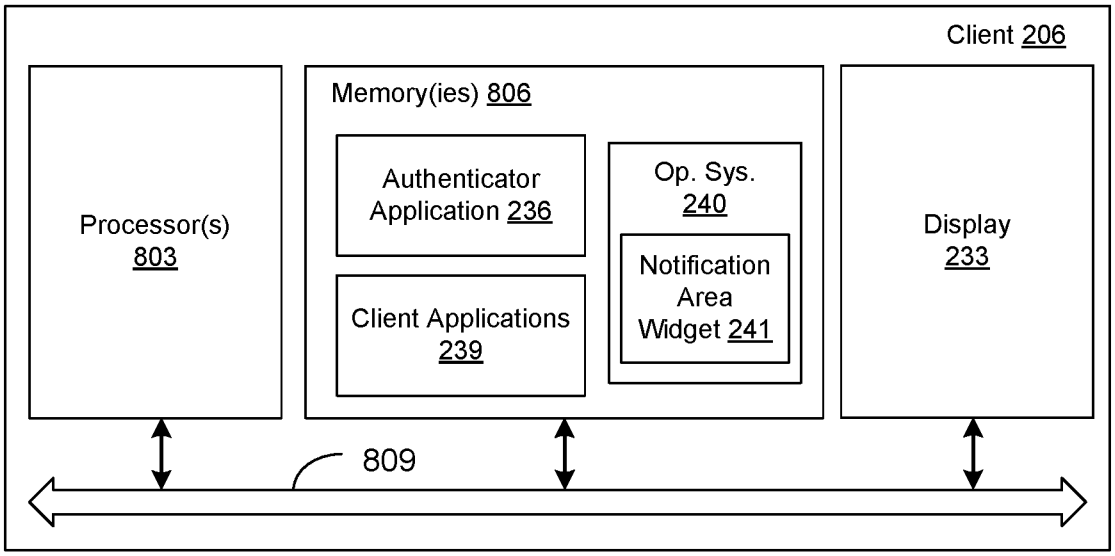
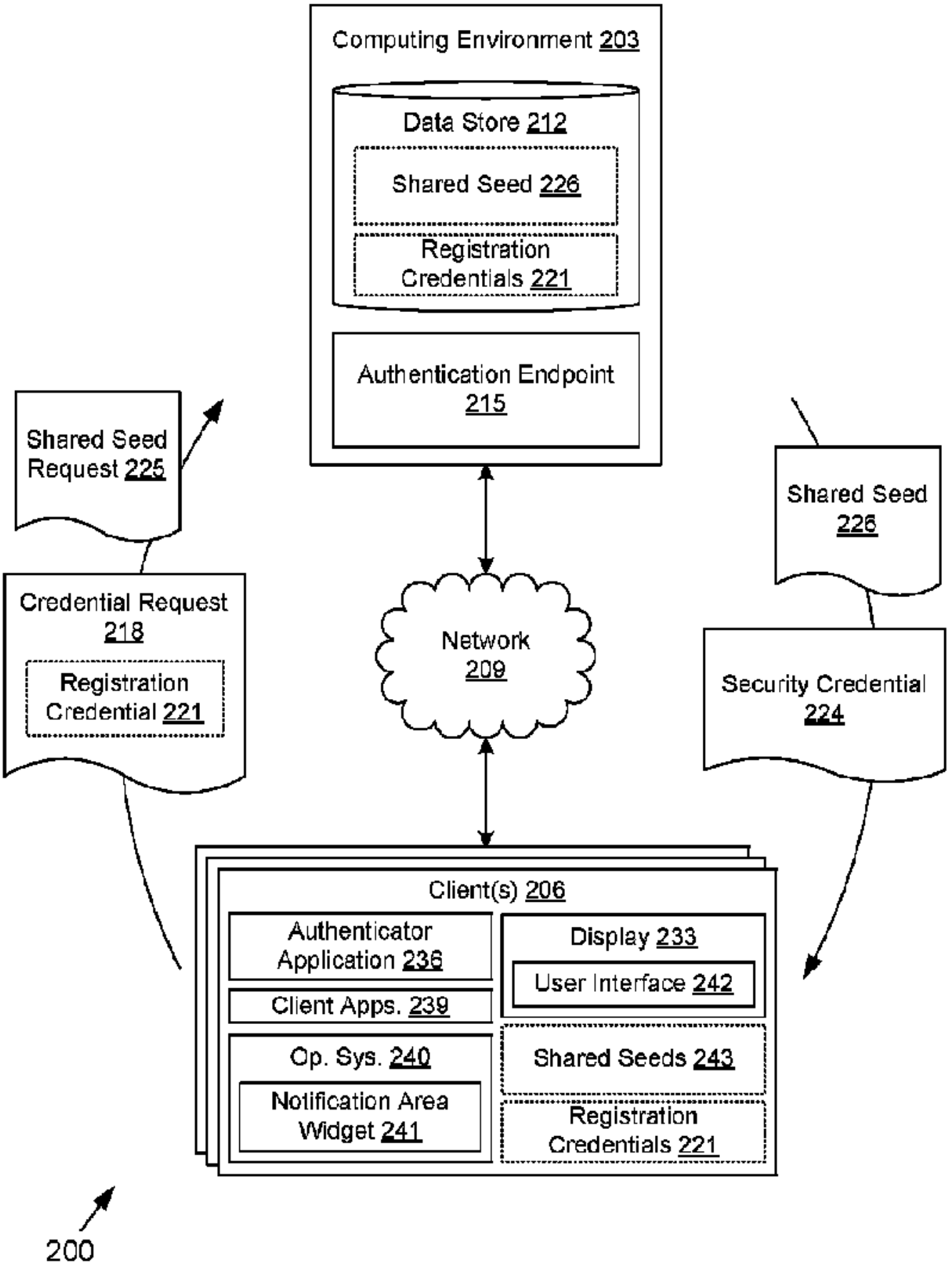

Client 206
Processor(s) 803
Memory(ies) 806
Authenticator Application 236
Client Applications 239
Op. Sys. 240
Notification Area Widget 241
Display 233
809

FIG. 8


Computing Environment 203
Data Store 212
Shared Seed 226
Registration Credentials 221
Authentication Endpoint 215
Shared Seed Request 225
Credential Request 218
Registration Credential 221
Network 209
Shared Seed 226
Security Credential 224
Client(s) 206
Authenticator Application 236
Client Apps. 239
Op. Sys. 240
Notification Area Widget 241
Display 233
User Interface 242
Shared Seeds 243
Registration Credentials 221
200