Abstract: Pass through service login for an application can include receiving, within a client system, a credential from a Web-based service responsive to a successful authentication of a user of the client system to the Web-based service. The user can be logged into the application executing within the client system using the credential.
PASS THROUGH SERVICE LOGIN TO APPLICATION LOGIN

BACKGROUND

[0001] A "browser" generally refers to a computer program that, when executed within a data processing system, retrieves electronic documents, such as Web pages, over a communication network and displays the retrieved electronic documents. Typically, the browser executes within, or as part of, a client system and retrieves information from a server system. Modern browsers can provide users with an interactive experience through which data, e.g., Web pages, can be accessed.

[0002] In some cases, a user is permitted to maintain an identity with an application. Having determined an identity for a user through some mechanism of authentication, an application such as a browser can provide the user with a customized user experience. The application can implement a customized environment that is stored in association with the identity.

BRIEF SUMMARY

[0003] One or more embodiments disclosed within this specification relate to using a login process for a service to log a user into an application executing within a client system.

[0004] An embodiment can include a method. The method can include receiving, within a client system, a credential from a Web-based service responsive to a successful authentication of a user of the client system to the Web-based service. The user can be logged into the application executing within the client system using the credential.

[0005] Another embodiment can include a system having a processor. The processor can be configured to initiate executable operations including receiving, within a client system, a credential from a Web-based service responsive to a successful authentication of a user of the client system to the Web-based service. The user can be logged into the application executing within the client system using the credential.

[0006] Another embodiment can include a computer program product. The computer
program product can include a computer readable storage medium having stored thereon program code that, when executed, configures a processor to perform operations including receiving, within a client system, a credential from a Web-based service responsive to a successful authentication of a user of the client system to the Web-based service. The user can be logged into the application executing within the client system using the credential.

BRIEF DESCRIPTION OF THE SEVERAL VIEWS OF THE DRAWINGS

[0007] FIG. 1 is a block diagram illustrating a communication system in accordance with an embodiment disclosed within this specification.

[0008] FIG. 2 is a block diagram illustrating an exemplary implementation of a client system shown in FIG. 1 in accordance with another embodiment disclosed within this specification.

[0009] FIG. 3 is a block diagram illustrating an exemplary implementation of a browser in accordance with another embodiment disclosed within this specification.

[0010] FIG. 4 is a flow chart illustrating a method of logging a user into an application within a client in accordance with another embodiment disclosed within this specification.

DETAILED DESCRIPTION

[0011] While the specification concludes with claims defining features of one or more embodiments that are regarded as novel, it is believed that the embodiment(s) will be better understood from a consideration of the description in conjunction with the drawings. As required, one or more detailed embodiments are disclosed within this specification. It should be appreciated, however, that the embodiment(s) are merely exemplary. Therefore, specific structural and functional details disclosed within this specification are not to be interpreted as limiting, but merely as a basis for the claims and as a representative basis for teaching one skilled in the art to variously employ the embodiment(s) in virtually any appropriately detailed structure. Further, the terms and
phrases used within this specification are not intended to be limiting, but rather to provide an understandable description of the embodiments disclosed herein.

One or more embodiments disclosed within this specification relate to using a login process for a service to log a user into an application executing within a client system. In accordance with the inventive arrangements disclosed within this specification, a login process used to authenticate a user to a service can be used to authenticate the user to an application executing within the user's client system. By leveraging the same login process used by the service, the user need not undertake a separate login process that is distinct from the service login process in order to login to the application. Once the user is logged into the application, the application can perform one or more operations with the service on behalf of the user. The operations can be performed using the identity with which the user has been associated within the application via the "shared" login process.

In one aspect, data received from the service responsive to a successful authentication of the user to the service can be used to establish an identity for the user within the application. For example, the application can intercept a credential sent from the service and store the credential for subsequent use. In general, the reception of the credential from a particular service, e.g., one that is considered a trusted service, indicates that the user has been successfully authenticated to that service as an authorized user. As such, the application can consider the user authorized and effectively logs the user into the application by providing the user access to an identity maintained within the application.

FIG. 1 is a block diagram illustrating a communication system 100 in accordance with an embodiment disclosed within this specification. As pictured, communication system 100 can include a client system (client) 105 coupled to a server system (server) 130 via a network 125.

Client 105 is implemented as a data processing system operable to execute program code, e.g., an operating system (not shown) and an application such as a browser 110. Client 105 can be implemented as virtually any data processing system that is
capable of executing an application such as browser 110 and that is also capable of coupling to service 140 through network 125 via a wired or wireless connection.

[0016] Network 125 can represent any of a variety of communication networks or a combination of two or more communication networks coupled together. For example, network 125 can be implemented as, or include, a Wide Area Network, a local area network, a wireless network, a mobile network, the Internet, or various combinations thereof.

[0017] Server 130 is implemented as a single data processing system or as a group of two or more interconnected data processing systems. Server 130 is configured to store information and respond to requests received from client systems, e.g., client 105, over network 125. Server 130 can be configured to provide data stored therein to client 105 over network 125 responsive to the received requests. Accordingly, system 100 generally implements a "client-server" type of architecture.

[0018] As shown, client 105 executes browser 110. Browser 110 is implemented as a computer program that, when executed, configures client 105 to perform operations including, but not limited to, requesting data from server 130, retrieving data from server 130, and displaying data received from server 130. For example, client 105, while executing browser 110, can request, receive, and display electronic documents such as Web pages.

[0019] Server 130 executes service 140. Service 140 is any of a variety of Web-based services that are accessed by client systems. Accordingly, server 130, executing service 140, allows a user, e.g., a human being, using client 105 to login to service 140 and use one or more functions provided by service 140. Examples of service 140 can include, but are not limited to, electronic mail, photo management and sharing, electronic document generation and editing, calendaring, task management, or the like. Appreciably, one or more services can be linked together and made available to a user if the user is authenticated with the service(s).

[0020] Service 140 can maintain one or more user accounts. Each account maintained or stored as part of service 140 can be considered a different identity.
maintained by service 140. For purposes of discussion, an identity maintained in server 130, e.g., as part of or by service 140, is referred to as a "service account." A service account includes information relating to service 140 such as preferences and content for the service such as documents, electronic mails, tasks, etc., for a specific user and/or for a particular identity of a user in the event the user has more than one service account with a same service.

[0021] Browser 110 also can maintain one or more user accounts. Each account maintained or stored as part of browser 110 can be considered a different identity maintained by browser 110. For purposes of discussion, a user identity maintained in client 105, e.g., as part of or by browser 110, is referred to as an "application account." An application account includes information relating to browser 110 such as preferences and settings for browser 110 that, when implemented provide a customized application environment for a specific user and/or for a particular identity of the user in the event the user has more than one application account for browser 110.

[0022] As pictured, client 105, while executing browser 110, has navigated to a Web page 145 that is provided by service 140. Accordingly, client 105 is displaying Web page 145. In one aspect, browser 110 can detect an access, or attempted access, to service 140. As shown, Web page 145 is requesting that the user provide a username and a password to login, or "sign in," to service 140. It should be appreciated that while a username and a password is requested in this example, any of a variety of login procedures with lesser or greater security features can be used. As such, the one or more embodiments disclosed within this specification are not intended to be limited by the particular login information that is provided to authenticate a user to a service.

[0023] Browser 110, responsive to detecting an access to service 140, can display a notification 150. In this example, notification 150 is implemented as an information bar displayed across a top portion of a window of browser 110 in which Web page 145 is displayed. It should be appreciated that notification 150 can be presented or displayed using any of a variety of known user interface elements and that the use of an information bar is for purposes of illustration only. The one or more embodiments disclosed within
this specification are not intended to be limited by the particular user interface element
selected to implement notification 150.

[0024] Notification 150 queries the user whether he or she would like to sign into the
local application executing in client 105, which is browser 110 in this example. More
particularly, notification 150 asks the user whether he or she would like to login to
browser 110 using the same login process used by service 140. If not, e.g., the user
selects "no," the user can continue to login to service 140 in the conventional manner and
is not logged into browser 110. If the user decides to login to the local application, e.g.,
the user selects "yes," then browser 110 leverages the login process implemented by
service 140 to log the user into the application. Browser 110 uses authentication
information provided from service 140 responsive to successful authentication of the user
to log the user into browser 110.

[0025] Consider the case in which the user selects "yes" within notification 150. In
that case, the user has provided a username and a password within the fields illustrated
within Web page 145. Responsive to the user submitting the login information, e.g., the
user pressing the enter key or selecting the "sign in" interface element, client 105 sends
login information 155 to server 130 and, more particularly, to service 140 executing
within server 130. In this example, as noted, login information 155 includes the
username and the password provided by the user. In other embodiments, fewer or more
items of information, e.g., a certificate or the like, can be included as part of login
information 155.

[0026] Upon receipt of login information 155, service 140 authenticates the user
thereby correlating the user with a service account matching the login information.
Accordingly, the user is given access to the identified service account. Responsive to
successfully authenticating the user, service 140 sends authentication information
referred to as a credential 160 to client 105. Responsive to receiving credential 160,
browser 110 logs the user into browser 110. Browser 110 provides the user with access
to an application account maintained within browser 110 that is associated with the
credential 160 and/or service 140.
[0027] In one aspect, credential 160 is a bearer token. A bearer token refers to a digital object that is presented to an entity, e.g., a verifying entity such as service 140, in an authentication transaction. In the case of a bearer token, the bearer token need not be bound to a particular identity. Rather, the mere possession of the bearer token by a system such as client 105 authorizes the client for particular activities. In this case, possession of the bearer token can log the user into service 140 and provide the user, by way of client 105, access to a service account. In another aspect, credential 160 can be bound to an individual or identity to which the credential was issued.

[0028] An example of credential 160 can be a digital document such as a cookie or another file. As noted, credential 160 is used in authentication and/or access control and may bind an identity or an attribute to a user or some other property such as a network address. In this example, when service 140 detects credential 160 on client 105 in subsequent transactions, client 105 is presumed to be under the control of, or is recognized as, an authorized user and provided access to the service account associated with credential 160.

[0029] In another aspect, browser 110 can store credential 160 locally and persist, e.g., store, credential 160 in a long term storage so that the user, by virtue of having credential 160 stored locally in client 105, continues to be logged in to the application account as well as the service account. While the user is logged into the application account, browser 110 can perform one or more actions on behalf of the user. In one aspect, browser 110 can activate one or more user settings to implement a user-specific application environment. In another aspect, browser 110 can automatically back-up and/or store the application account of the user to service 140. In the case of a browser, the application account can include cookies, usernames and passwords, bookmarks, and the like.

[0030] The exemplary embodiments described with reference to FIG. 1 facilitate the automated login of a user to an application executing locally within a client using the same login process and authentication information used by a service. Further aspects of the embodiments will be described with reference to the remaining figures.
FIG. 2 is a block diagram illustrating an exemplary implementation of client 105 of FIG. 1 in accordance with another embodiment disclosed within this specification. Like numbers will be used to refer to the same items throughout this specification. Client 105 can include at least one processor 205 coupled to memory elements 210 through a system bus 215 or other suitable circuitry. As such, client 105 can store program code within memory elements 210. Processor 205 can execute the program code accessed from memory elements 210 via system bus 215.

Memory elements 210 can include one or more physical memory devices such as, for example, local memory 220 and one or more bulk storage devices 225. Local memory 220 refers to random access memory (RAM) or other non-persistent memory device(s) generally used during actual execution of the program code. Bulk storage device(s) 225 can be implemented as a hard disk drive (HDD), a solid state drive (SSD), or another persistent data storage device. Client 105 also can include one or more cache memories (not shown) that provide temporary storage of at least some program code in order to reduce the number of times program code must be retrieved from bulk storage device 225 during execution.

In one aspect, for example, client 105 can be implemented as a computer or communication device that is suitable for storing and/or executing program code. Client 105 can be implemented in the form of any system including a processor and memory that is capable of performing the functions and/or operations described within this specification. Examples of client system 105 can include any of a variety of computing systems such as desktop computers, laptop computers, thin clients, tablet computers, mobile communication devices such as mobile phones, or the like.

Input/output (I/O) devices such as a keyboard 230, a display 235, and a pointing device 240 optionally can be coupled to client 105. The I/O devices can be coupled to client 105 either directly or through intervening I/O controllers. One or more network adapters 245 also can be coupled to client 105 to enable client 105 to become coupled to other systems, computer systems, remote printers, and/or remote storage devices through intervening private or public networks. Modems, cable modems,
Ethernet cards, and wireless transceivers are examples of different types of network adapters that can be used with client 105.

[0035] As pictured in FIG. 2, memory elements 210 can store browser 110. Browser 110, being implemented in the form of executable program code, can be executed by client 105 and, as such, can be considered part of client 105. In general, browser 110 can be configured to detect an access a service, e.g., a Web-based service. Responsive to detecting the access, browser 110 can query a user whether to log the user into the application using any authentication information that may be received from the service in consequence of a successful authentication of the user to the service.

[0036] FIG. 3 is a block diagram illustrating an exemplary implementation of browser 110 in accordance with another embodiment disclosed within this specification. As pictured, browser 110 can include a login module 305. Browser 110 further can include more than one application account. Application account 310 can include or specify information that is different from application account 315.

[0037] For purposes of illustration, consider the case in which application account 310 corresponds to a first user, whereas application account 315 corresponds to a second and different user. In another example, application account 310 can be a first identity of a user, while application account 315 is a second and different identity of the same user. For example, application account 310 is a "personal" account whereas application account 315 is a "work" account for browser 110.

[0038] In one embodiment, each of application accounts 310 and 315 can be associated with one or more different services, and as such, Web pages provided by such services. For example, each application account can be correlated with one or more Uniform Resource Identifiers (URIs), or portions thereof, for a service. In another example, each application account can be correlated with an element such as a tag or other identifier that is included within a Web page that is identifiable by browser 110 when received and/or displayed. In any case, login module 305 can coordinate and maintain the associations.

[0039] In one aspect, browser 110 can store a list of services for which the login
process can be used to login a user to browser 110. For example, login module 305 can maintain a list of approved or trusted services. Further, login module 305 can store associations between selected, e.g., approved, services and application accounts. For example, login module 305 can store an association between application account 310 and service 320. Login module 305 further can store an association between application account 315 and service 325.

[0040] Consider the case in which a user navigates to a Web page of service 320 and is asked to login to service 320. Browser 110 can detect an access of service 320. In one aspect, detecting access can include login module 305 detecting a request sent to service 320, e.g., using the URI, and identifying service 320 as one in which the login process can be used to login the user into browser 110. For example, the URI or a portion thereof can be matched to a trusted service on the list. In another aspect, detecting access can include login module 305 detecting an identifier within a Web page received from service 320 and identifying service 320 as one in which the login process can be used to login the user into browser 110. For example, the identifier can be matched to a trusted service on the list.

[0041] In response to detecting an access to service 320, browser 110 can query the user whether to login to an application account stored therein. In this example, the user has responded in the affirmative to login to an application account. Login module 305 recognizes the association between service 320 and application account 310. Accordingly, responsive to receiving credential 330 from service 320, login module 305 logs the user into application account 310 using credential 330.

[0042] FIG. 3 is presented to illustrate an example in which multiple application accounts are maintained within browser 110. In another embodiment, however, login module 305 need not store any associations as described. In that case, for example, login module 305 can present, e.g., display, a list of available application accounts from which the user can choose, e.g., application account 310 and application account 315. Login module 305 can log the user into application account selected by the user using credential 330. In still another embodiment, one of the plurality of application accounts can be
designated as a "default" account. In that case, login module 305 can log the user into the default account using credential 330. It should be appreciated that when a single application account is included within browser 110, that account can be selected automatically for purposes of logging the user in using credential 330. It should be appreciated that whether or not associations between particular services and particular application accounts are stored, browser 110 still can store information, e.g., a list, indicating trusted services for which the authentication processes can be used for purpose of logging a user into browser 110.

[0043] FIG. 4 is a flow chart illustrating a method 400 of logging a user into an application within a client in accordance with another embodiment disclosed within this specification. Method 400 can be performed by a system such as client 105 as described with reference to FIG. 1. Accordingly, the method 400 can begin in a state in which the browser is executing with the client.

[0044] In block 405, the client can navigate to a service and load a Web page from the service. For example, the client can request a Web page that can be received from the service. In block 410, the client can detect an access to a service for which the authentication process can be used to log the user into an application executing within the client, e.g., the browser. The client can detect an access of the service by the client using any of a variety of the different techniques previously described including variants or equivalents thereof.

[0045] In block 415, responsive to detecting access by the client to the service, the client can query the user whether to log the user into the application. In block 420, the client determines whether the user wishes to login to the application based upon a received response. It should be appreciated that a lack of a response can be interpreted by the system as a negative response indicating that the user does not wish to login to the application. For example, providing a username and a password without selecting yes or no to the notification would be construed as a negative response. If the user indicates that he or she would like to login to the application, method 400 can proceed to block 430. If the user indicates that he or she would not like to login to the application, method 400 can
continue to block 425. In block 425, the client can continue a login process of the service without logging the user into the application.

[0046] In block 430, where the user has indicated a desire to login to the application, the client can send login information to the service. As noted, the client can log the user into the application account using the same login process used to authenticate the user to the service. In block 435, the user can be logged into the service using the user login information. The service, for example, can authenticate the user and provide the user with access to the service account that is associated with the identity of the user as determined from the login information.

[0047] In block 440, the client can receive a credential from the service. In block 445, the client can store the received credential. The client can store the credential in a data storage medium within the client or that is accessible by the client. As discussed, in one aspect the client can store the received credential in a long-term data storage medium that persists data despite the browser terminating execution, the client system being turned off, restarting, e.g., power cycling, or the like.

[0048] In block 450, the client optionally can select a particular application account to which the user is to be given access upon login to the browser. Block 450 can be implemented when multiple application accounts are stored within the application and a particular application account must be selected for logging the user into the application. As noted, a default application account can be selected from a plurality of application accounts, a correlation can be stored between the service and a particular application account, or the user can manually select an application account.

[0049] In block 455, the client can log the user into the application, which is a browser in this example. The browser can authenticate the user using the received credential from the service and provide the user access to the application account that is associated with the identity indicated by the credential.

[0050] As noted, the reception of a credential from a particular service, e.g., one that is considered a trusted service by the client, indicates that the user has been successfully authenticated to that service as an authorized user. As such, the application, which is a
browser in this example, can consider the user authorized and provide the user access to an application account as described. In one aspect, the application account and the service can have same login information. For example, referring to FIG. 1, both the application account into which the user is logged in and the service may have a same username and a same password.

[0051] In block 460, the client, e.g., with the application executing therein, can perform one or more operations. The operations can be performed on behalf of the user using the identity of the application account to which the user has been given access within the application.

[0052] In illustration, when the application is a browser, in one aspect, the browser can store a copy of the application account, or a portion thereof, with the service. For example, the browser can backup the application account, settings of the application account, or the like, to the service. In another aspect, the browser can synchronize the application account, or a portion thereof, with the service. In another example, the user can be logged into the application and provided access to the application account in subsequent sessions of the application, e.g., any session in which the application is executed after a termination of execution of the application. The browser can continue to perform operations as described until the credential is no longer stored on the client or expires. In one embodiment, when the credential is no longer stored in the client or expires, the user can be logged out of the application.

[0053] Another exemplary operation can include activating or invoking one or more settings of the browser specified by the application account to implement a preferred and user-specific application environment. For example, particular cookies, bookmarks, visual themes for the user interface of the application, and the like can be made available to the user when logged into the application as specified by the particular application account to which the user has been given access.

[0054] The embodiment(s) disclosed within this specification illustrate various techniques for logging a user into an application using a login process for a service. While a browser has been used for purposes of illustration, it should be appreciated that
the one or more embodiments disclosed herein can be implemented with other types of applications. The one or more embodiments can be applied to any type of application that is capable of accessing a service, which includes accessing information from an "online" source or over a network. Further, as different varieties of applications incorporate embedded browsers or browser functionality, the one or more embodiments disclosed herein can be applied to such applications as well.

[0055] For example, in the event that the application is not a browser, but is configured to access information over a network, the application can issue a request to a service. A variety of applications such as, for example, financial applications, access financial data from sources over a network. In that case, the URI of the service can be recognized by the application. Alternatively, information received in response to a request can include an element identifiable by the application.

[0056] In any case, the application can present a user interface, as opposed to a Web page, for example, through which login information is requested from the user for the service. The user interface can be any of a variety of interface elements into which a user can enter login information. The user interface further can include a notification as described that asks the user whether he or she would also like to login to the application. The process can proceed as described with a credential received from the service being used to log the user into the application.

[0057] For purposes of explanation, specific nomenclature is set forth to provide a thorough understanding of the various inventive concepts disclosed herein. The terminology used herein, however, is for the purpose of describing particular embodiments only and is not intended to be limiting. For example, reference throughout this specification to "one embodiment," "an embodiment," or similar language means that a particular feature, structure, or characteristic described in connection with the embodiment is included in at least one embodiment disclosed within this specification. Thus, appearances of the phrases "in one embodiment," "in an embodiment," and similar language throughout this specification may, but do not necessarily, all refer to the same embodiment.
The terms "a" and "an," as used herein, are defined as one or more than one. The term "plurality," as used herein, is defined as two or more than two. The term "another," as used herein, is defined as at least a second or more. The term "coupled," as used herein, is defined as connected, whether directly without any intervening elements or indirectly with one or more intervening elements, unless otherwise indicated. Two elements also can be coupled mechanically, electrically, or communicatively linked through a communication channel, pathway, network, or system.

The term "and/or" as used herein refers to and encompasses any and all possible combinations of one or more of the associated listed items. It will be further understood that the terms "includes" and/or "including," when used in this specification, specify the presence of stated features, integers, steps, operations, elements, and/or components, but do not preclude the presence or addition of one or more other features, integers, steps, operations, elements, components, and/or groups thereof. It will also be understood that, although the terms first, second, etc. may be used herein to describe various elements, these elements should not be limited by these terms, as these terms are only used to distinguish one element from another.

The term "if" may be construed to mean "when" or "upon" or "in response to determining" or "in response to detecting," depending on the context. Similarly, the phrase "if it is determined" or "if [a stated condition or event] is detected" may be construed to mean "upon determining" or "in response to determining" or "upon detecting [the stated condition or event]" or "in response to detecting [the stated condition or event]." depending on the context.

One or more embodiments can be realized in hardware or a combination of hardware and software. One or more embodiments can be realized in a centralized fashion in one system or in a distributed fashion where different elements are spread across several interconnected systems. Any kind of data processing system or other apparatus adapted for carrying out at least a portion of the methods described herein is suited.

One or more embodiments further can be implemented in the form of a
computer program product, which includes all the features enabling the implementation of the methods described herein. The computer program product includes a data storage medium, e.g., a non-transitory computer-usable or computer-readable storage medium, storing program code that, when executed in a system including a processor, causes the system to perform at least a portion of the functions described within this specification. Examples of data storage media can include, but are not limited to, optical media, magnetic media, magneto-optical media, computer memory such as RAM, a bulk storage device, e.g., hard disk, or the like.

[0063] Accordingly, the flowchart and block diagrams in the Figures illustrate the architecture, functionality, and operation of possible implementations of systems, methods and computer program products according to various embodiments disclosed herein. In this regard, each block in the flowchart or block diagrams may represent a module, segment, or portion of code, which includes one or more executable instructions for implementing the specified logical function(s). It should also be noted that, in some alternative implementations, the functions noted in the blocks may occur out of the order noted in the figures. For example, two blocks shown in succession may be executed substantially concurrently, or the blocks may sometimes be executed in the reverse order, depending upon the functionality involved. It will also be noted that each block of the block diagrams and/or flowchart illustration, and combinations of blocks in the block diagrams and/or flowchart illustration, can be implemented by special purpose hardware-based systems that perform the specified functions or acts, or combinations of special purpose hardware and computer instructions.

[0064] The terms "computer program," "software," "application," "computer-usable program code," "program code," "executable code," variants and/or combinations thereof, in the present context, mean any expression, in any language, code or notation, of a set of instructions intended to cause a system having an information processing capability to perform a particular function either directly or after either or both of the following: a) conversion to another language, code, or notation; b) reproduction in a different material form. For example, program code can include, but is not limited to, a
subroutine, a function, a procedure, an object method, an object implementation, an executable application, an applet, a servlet, source code, object code, markup language, an extension, a shared library/dynamic load library and/or other sequence of instructions designed for execution on a computer system.

[0065] Thus, throughout this specification, statements utilizing terms such as "processing," "computing," "calculating," "determining," "displaying," or the like, refer to the action and processes of a data processing system, e.g., a computer system, or similar electronic computing device, that manipulates and transforms data represented as physical (electronic) quantities within the computer system's registers and/or memories into other data similarly represented as physical quantities within the computer system memories and/or registers or other such information storage, transmission or display devices.

[0066] The corresponding structures, materials, acts, and equivalents of all means or step plus function elements in the claims below are intended to include any structure, material, or act for performing the function in combination with other claimed elements as specifically claimed.

[0067] One or more embodiments disclosed within this specification can be embodied in other forms without departing from the spirit or essential attributes thereof. Accordingly, reference should be made to the following claims, rather than to the foregoing specification, as indicating the scope of the one or more embodiments.
CLAIMS

What is claimed is:

1. A method comprising:
   receiving, within a client system, a credential from a Web-based service
   responsive to a successful authentication of a user of the client system to the Web-based
   service; and
   logging the user into an application executing within the client system using the
   credential.

2. The method of claim 1, further comprising:
   detecting an access of the Web-based service; and
   responsive to detecting the access, querying a user whether to log the user into the
   application using the credential.

3. The method of claim 1, further comprising:
   storing the credential in a long term storage of the client system.

4. The method of claim 1, wherein the credential is associated with the Web-based
   service and an application account for the application.

5. The method of claim 4, wherein the application account is selected from a
   plurality of application accounts maintained within the application.

6. The method of claim 1, wherein the application is a browser configured to
   navigate to the Web-based service.

7. The method of claim 1, further comprising:
responsive to logging the user into the application, implementing at least one setting of the application specified by the application account.

8. The method of claim 1, further comprising:
responsive to logging the user into the application, performing an operation on behalf of the user with the Web-based service.

9. A system comprising:
a processor configured to invoke executable operations comprising:
receiving, within a client system, a credential from a Web-based service responsive to a successful authentication of a user of the client system to the Web-based service; and
logging the user into an application executing within the client system using the credential.

10. The system of claim 9, wherein the processor is further configured to invoke executable operations comprising:
detecting an access of the Web-based service; and
responsive to detecting the access, querying a user whether to log the user into the application using the credential.

11. The system of claim 9, wherein the processor is further configured to invoke an executable operation comprising:
    storing the credential in a long term storage of the client system.

12. The system of claim 9, wherein the credential is associated with the Web-based service and an application account for the application.
13. The system of claim 12, wherein the application account is selected from a plurality of application accounts maintained within the application.

14. The system of claim 9, wherein the application is a browser configured to navigate to the Web-based service.

15. The system of claim 9, wherein the processor is further configured to invoke an executable operation comprising:
   responsive to logging the user into the application, implementing at least one setting of the application specified by the application account.

16. The system of claim 9, wherein the processor is further configured to invoke an executable operation comprising:
   responsive to logging the user into the application, performing an operation on behalf of the user with the Web-based service.

17. A computer program product for accessing data, the computer program product comprising:
   a computer readable storage medium having stored thereon program code that, when executed, configures a processor to perform executable operations comprising:
   receiving, within a client system, a credential from a Web-based service responsive to a successful authentication of a user of the client system to the Web-based service; and
   logging the user into an application executing within the client system using the credential.

18. The computer program product of claim 17, wherein the computer readable storage medium stores further program code that, when executed, configures the processor to perform executable operations comprising:
detecting an access of the Web-based service; and
responsive to detecting the access, querying a user whether to log the user into the application using the credential.

19. The computer program product of claim 17, wherein the computer readable storage medium stores further program code that, when executed, configures the processor to perform an executable operation comprising:
   storing the credential in a long term storage of the client system.

20. The computer program product of claim 17, wherein the credential is associated with the Web-based service and an application account for the application.

21. The computer program product of claim 20, wherein the application account is selected from a plurality of application accounts maintained within the application.

22. The computer program product of claim 17, wherein the application is a browser configured to navigate to the Web-based service.

23. The computer program product of claim 17, wherein the computer readable storage medium stores further program code that, when executed, configures the processor to perform an executable operation comprising:
   responsive to logging the user into the application, implementing at least one setting of the application specified by the application account.

24. The computer program product of claim 17, wherein the computer readable storage medium stores further program code that, when executed, configures the processor to perform an executable operation comprising:
   responsive to logging the user into the application, performing an operation on behalf of the user with the Web-based service.
FIG. 1
Navigate to service

Detect access to service

Query whether to log user into application using login process of service

Yes

Log user into application?

No

Continue login process of service without logging user into application

Send login information to service

User is logged into service using received login information

Receive credential from service

Store credential

Optionally select application account

Log user into application

Perform one or more programmatic operations on behalf of user

End

FIG. 4
C. DOCUMENTS CONSIDERED TO BE RELEVANT

<table>
<thead>
<tr>
<th>Category</th>
<th>Citation of document, with indication, where appropriate, of the relevant passages</th>
<th>Relevant to claim No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>US 2012-0117634 Al (DAVID HALLS et al.) 10 May 2012</td>
<td>1-24</td>
</tr>
<tr>
<td></td>
<td>See paragraphs 6, 29-31, 48, 56-57, 94; and figure 3.</td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>US 2011-0296504 Al (LLOYD LEON BURCH et al.) 01 December 2011</td>
<td>1-24</td>
</tr>
<tr>
<td></td>
<td>See paragraphs 43-53; and figure 3.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>See paragraphs 4-28; and figures 1, 3-4.</td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>US 2009-0199276 Al (JAMES P. SCHNEIDER) 06 August 2009</td>
<td>1-24</td>
</tr>
<tr>
<td></td>
<td>See paragraphs 39-48; and figure 2A.</td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>US 2012-0072714 Al (MICHAEL GRANDCOLAS et al.) 22 March 2012</td>
<td>1-24</td>
</tr>
<tr>
<td></td>
<td>See paragraphs 48-53; and figures 3-6.</td>
<td></td>
</tr>
</tbody>
</table>

Further documents are listed in the continuation of Box C.

See patent family annex.

* Special categories of cited documents:
  "A" document defining the general state of the art which is not considered to be of particular relevance
  "E" earlier application or patent but published on or after the international filing date
  "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of citation or other special reason (as specified)
  "O" document referring to an oral disclosure, use, exhibition or other means
  "P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search: 19 November 2013 (19.11.2013)

Date of mailing of the international search report: 20 November 2013 (20.11.2013)

Name and mailing address of the ISA/KR:
Korean Intellectual Property Office
189 Cheongsa-ro, Seo-gu, Daejeon Metropolitan City,
341-701 Republic of Korea
Facsimile No. +82-42-472-7140

Authorized officer:
BYUN, Sung Cheal
Telephone No. +82-42-481-8262

Form PCT/ISA/210 (second sheet) (July 2009)
<table>
<thead>
<tr>
<th>Patent document cited in search report</th>
<th>Publication date</th>
<th>Patent family member(s)</th>
<th>Publication date</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>CA 2624623 Al</td>
<td>26/04/2007</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EP 1955251 A2</td>
<td>13/08/2008</td>
</tr>
<tr>
<td></td>
<td></td>
<td>US 2007-107048 Al</td>
<td>10/05/2007</td>
</tr>
<tr>
<td></td>
<td></td>
<td>US 8112789 B2</td>
<td>07/02/2012</td>
</tr>
<tr>
<td>US 2011-0296504 Al</td>
<td>01/12/2011</td>
<td>None</td>
<td></td>
</tr>
<tr>
<td>US 2009-0199276 Al</td>
<td>06/08/2009</td>
<td>None</td>
<td></td>
</tr>
<tr>
<td>US 2012-0072714 Al</td>
<td>22/03/2012</td>
<td>US 2013-097429 Al</td>
<td>18/04/2013</td>
</tr>
<tr>
<td></td>
<td></td>
<td>US 8146141 B1</td>
<td>27/03/2012</td>
</tr>
<tr>
<td></td>
<td></td>
<td>US 8302172 B2</td>
<td>30/10/2012</td>
</tr>
</tbody>
</table>