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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
19 March 2020 (19.03.2020)

WIPO | PCT

(10) International Publication Number
WO 2020/055847 A1

(51) International Patent Classification:
H04L 29/08 (2006.01) *H04W 12/08* (2009.01)
H04L 29/06 (2006.01)

(21) International Application Number:
PCT/US2019/050386

(22) International Filing Date:
10 September 2019 (10.09.2019)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
16/128,424 11 September 2018 (11.09.2018) US

(71) Applicant: **CITRIX SYSTEMS, INC.** [US/US]; 851 West Cypress Creek Road, Fort Lauderdale, Florida 33309 (US).

(72) Inventor: **CHAUHAN, Abhishek**; c/o Citrix Systems, Inc., 4988 Great America Parkway, Santa Clara, California 95054 (US).

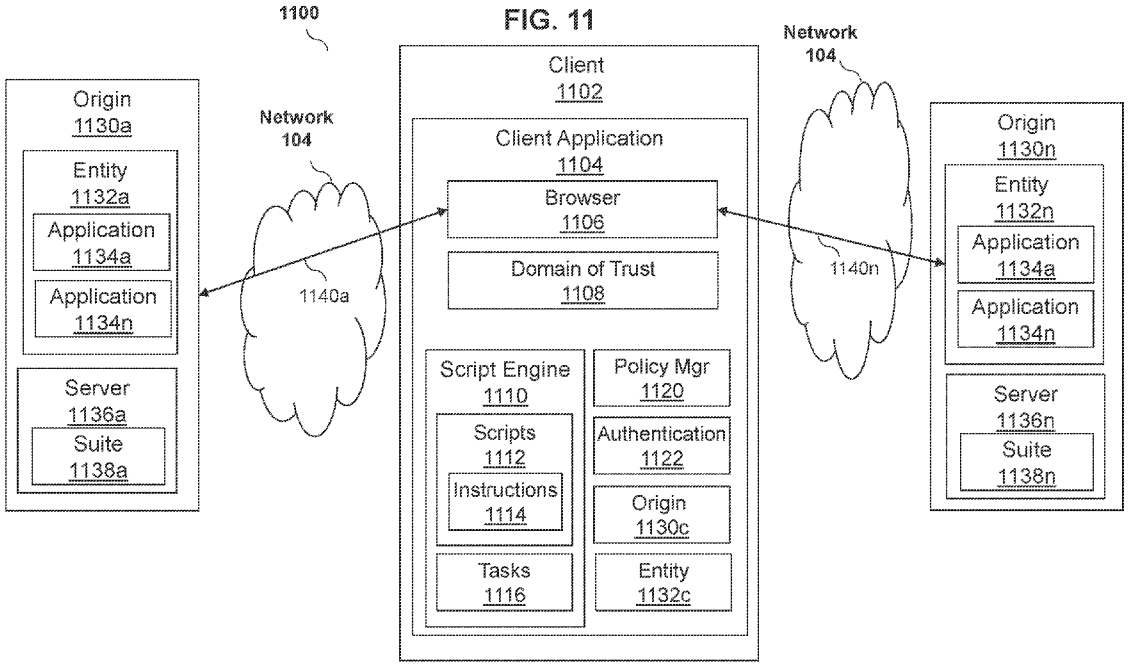
(74) Agent: **MCKENNA, Christopher J.** et al.; FOLEY & LARDNER LLP, 3000 K Street N.W., Suite 600, Washington, District of Columbia 20007-5109 (US).

(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, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP, KR, KW, 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.

WO 2020/055847 A1

(54) Title: APPLICATION SCRIPTS FOR CROSS-DOMAIN APPLICATIONS

FIG. 11



(57) **Abstract:** Embodiments described include systems and methods for executing in an embedded browser an application script for network applications of different origins. A client application can establish a first session with a first network application of a first entity at a first origin via an embedded browser within the client application and a second session with a second network application of a second entity at a second origin via the embedded browser within the client application. A scripting engine within the client application of a client device of a user at a third origin can identify an application script having instructions to interact with the first network application and the second network application, and can execute the instructions to perform a task across the first network application of the first entity at the first origin and the second network application of the second entity at the second origin.

[Continued on next page]

(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).

Declarations under Rule 4.17:

— *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*

— *as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii))*

Published:

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

— *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments (Rule 48.2(h))*

APPLICATION SCRIPTS FOR CROSS-DOMAIN APPLICATIONS

CROSS-REFERENCE TO RELATED APPLICATION

The present application claims priority to and the benefit of U.S. Patent Application No. 16/128,424, titled "SYSTEMS AND METHODS FOR APPLICATION SCRIPTS FOR CROSS-DOMAIN APPLICATIONS," and filed September 11, 2018, the contents of which are hereby incorporated herein by reference in its entirety for all purposes.

FIELD OF THE DISCLOSURE

The present application generally relates to management of applications, including but not limited to systems and methods for using an embedded browser to manage and monitor web and software-as-a-service (SaaS) applications.

BACKGROUND

As the workforce of an enterprise becomes more mobile and work under various conditions, an individual can use one or more client devices, including personal devices, to access network resources such as web applications. Due to differences between the client devices and the manner in which network resources can be accessed, there are significant challenges to the enterprise in managing access to network resources and monitoring for potential misuse of resources.

BRIEF SUMMARY

The present disclosure is directed towards systems and methods for executing in an embedded browser an application script for network applications of different origins. For example, a client application can execute on a client device via an embedded browser. The client application can establish or provide one or more sessions to one or more network applications via the embedded browser. The client application can establish a domain of trust between a client device and one or more network applications from one or more different entities and/or different origins. The domain of trust can provide a platform for a user of the client device to perform tasks, such as but not limited to, transferring data across network applications from different entities and/or different origins. For example, the client application can include or otherwise provide a scripting engine that generates one or more

application scripts. The application scripts can include instructions to perform the one or more tasks across network applications from different entities and/or different origins. The client application can use the application scripts to implement polices unique to the user of the client device on the network applications from different entities and/or different origins and thus, override or work around policies of the respective network applications, such as but not limited to, same origin policies, to allow cross application collaboration within the embedded browser of the client application.

Network applications coupled with or executing within the embedded browser of the client application can leverage the domain of trust for cross-application collaboration. For example, network application can include or be provided from a suite of applications that originate from or are hosted by different servers at different origins. The suits can include same-origin security policies that limit or prevent collaboration or integration between network applications from different suits, different servers and/or different origins. The systems and methods as described herein can establish a domain of trust that forms a platform for collaboration or integration between network applications from different suits, different servers and/or different origins. For example, a user of a client device can perform one or more tasks across the different network applications within the domain of trust using application scripts generated by a scripting engine of the client application.

The application scripts can work across multi-vendor network applications to perform the various tasks for a user of the client device. For example, the application scripts can be generated such that they meet the respective security policies of the network applications from different suits, different servers, different entities and/or different origins. Thus, the client application can operate as a scripting host for the client device to perform scripting, via a scripting engine and within the domain of trust, across the different applications from different suits, different servers, different entities and/or different origins. For example, the scripting engine of the client application can write an application script and/or automations tasks, such as but not limited to, pulling data from a first network application of a first entity at a first origin and transfer the data to a second network application of a second entity at a second origin.

In at least one aspect, a method for executing in an embedded browser an application script for network applications of different origins is provided. The method includes establishing, by a client application, a first one or more sessions with a first one or more

network applications of a first entity at a first origin via an embedded browser within the client application. The method includes establishing, by the client application, a second one or more sessions with a second one or more network applications of a second entity at a second origin via the embedded browser within the client application. The method includes identifying, by a scripting engine within the client application of a client device of a user at a third origin, an application script comprising instructions to interact with each the first one or more networks applications of the first entity at the first origin and the second one or more network applications of the second entity at the second origin. The method includes executing, by the scripting engine, the instructions of the application script to perform a task across the first one or more networks applications of the first entity at the first origin and the second one or more network applications of the second entity at the second origin.

The first one or more network applications can include a suite of applications originating from or hosted by a first one or more servers at the first origin. The second one or more network applications can include a suite of applications originating from or hosted by a second one or more servers at the second origin. The first origin and the second origin can be different origins that fail a same origin policy.

In some embodiments, the method can include establishing, by the client application, a domain of trust between the first one or more networks applications of the first entity at the first origin and the second one or more network applications of the second entity at the second origin. The domain of trust can be established responsive to authentication of the same user for the first one or more network applications and the second one or more network applications via the client application on the client device of a third entity of the user corresponding to the third origin. The method can include initiating execution of the application script by one of the first one or more network applications or the second one or more network applications.

The method can include allowing, by the client application responsive to a policy, the application script to interact across the first origin and the second origin. The policy can specify that the first origin and the second origin are trusted origins to interact across via the embedded browser of the client application.

In at least one aspect, a method for collaborating across network applications of different origins in an embedded browser is provided. The method can include establishing,

by a client application, a first one or more sessions with a first one or more network applications of first entity at a first origin via an embedded browser within the client application. The method can include establishing, by the client application, a second one or more sessions with a second one or more network applications of a second entity at a second origin via the embedded browser within the client application. The method can include determining, by the client application responsive to a policy, that the first origin and the second origin are origins to be trusted to interact across via the embedded browser. The method can include allowing, by the client application responsive to the determination, the first one or more networks applications of the first entity at the first origin to interact via the embedded browser with the second one or more network applications of the second entity at the second origin.

The first origin and the second origin can fail a same origin policy. The policies can specify a plurality of different origins to trust, the plurality of different origins comprising the first origin and the second origin. The method can include executing, by a scripting engine of the client application, a script to interact via the embedded browser between the first one or more networks applications of the first entity at the first origin and the second one or more network applications of the second entity at the second origin. In some embodiments, the method can include allowing interacting responsive to authentication of the same user to each of the first one or more networks applications and the second one or more network applications.

In at least one aspect, a system for collaborating across network applications of different origins in an embedded browser is provided. The system can include a client application executable on one or more processors of a client device. The client application can be configured to establish a first one or more sessions with a first one or more network applications of first entity at a first origin via an embedded browser within the client application and a second one or more sessions with a second one or more network applications of a second entity at a second origin via the embedded browser within the client application. The client application can be configured to determine, responsive to a policy, that the first origin and the second origin are origins to be trusted to interact across via the embedded browser. The client application can be configured to allow, responsive to the determination, the first one or more networks applications of the first entity at the first origin to interact via the embedded browser with the second one or more network applications of the second entity at the second origin.

The first origin and the second origin can be different origins that fail a same origin policy. The policy can specify a plurality of different origins to trust, the plurality of different origins comprising the first origin and the second origin.

In some embodiments, the system can further include a scripting engine configured to execute instructions of a script to perform via the embedded browser a task across the first one or more networks applications of the first entity at the first origin and the second one or more network applications of the second entity at the second origin. The first one or more network applications can include a suite of applications originating from or hosted by a first one or more servers at the first origin. The client application can be configured to allow the interaction responsive to authentication of the same user to each of the first one or more networks applications and the second one or more network applications.

BRIEF DESCRIPTION OF THE FIGURES

The foregoing and other objects, aspects, features, and advantages of the present solution will become more apparent and better understood by referring to the following description taken in conjunction with the accompanying drawings, in which:

FIG. 1 is a block diagram of embodiments of a computing device;

FIG. 2 is a block diagram of an illustrative embodiment of cloud services for use in accessing resources;

FIG. 3 is a block diagram of an example embodiment of an enterprise mobility management system;

FIG. 4 is a block diagram of a system 400 of an embedded browser;

FIG. 5 is a block diagram of an example embodiment of a system for using a secure browser;

FIG. 6 is an example representation of an implementation for browser redirection using a secure browser plug-in;

FIG. 7 is a block diagram of example embodiment of a system of using a secure browser;

FIG. 8 is a block diagram of an example embodiment of a system for using local embedded browser(s) and hosted secured browser(s);

FIG. 9 is an example process flow for using local embedded browser(s) and hosted secured browser(s);

FIG. 10 is an example embodiment of a system for managing user access to webpages;

FIG. 11 is block diagram of an example embodiment of a system for collaborating across network applications of different origins in an embedded browser;

FIG. 12 is a flow diagram of an example embodiment of a method for executing in an embedded browser an application script for network applications of different origins; and

FIG. 13 is a flow diagram of an example embodiment of a method for collaborating across network applications of different origins in an embedded browser.

The features and advantages of the present solution will become more apparent from the detailed description set forth below when taken in conjunction with the drawings, in which like reference characters identify corresponding elements throughout. In the drawings, like reference numbers generally indicate identical, functionally similar, and/or structurally similar elements.

DETAILED DESCRIPTION

For purposes of reading the description of the various embodiments below, the following descriptions of the sections of the specification and their respective contents may be helpful:

Section A describes a computing environment which may be useful for practicing embodiments described herein.

Section B describes systems and methods for an embedded browser.

Section C describes systems and methods for executing in an embedded browser an application script for network applications of different origins.

A. Computing Environment

Prior to discussing the specifics of embodiments of the systems and methods detailed herein in Section B, it may be helpful to discuss the computing environments in which such embodiments may be deployed.

As shown in FIG. 1, computer 101 may include one or more processors 103, volatile memory 122 (e.g., random access memory (RAM)), non-volatile memory 128 (e.g., one or more hard disk drives (HDDs) or other magnetic or optical storage media, one or more solid state drives (SSDs) such as a flash drive or other solid state storage media, one or more hybrid magnetic and solid state drives, and/or one or more virtual storage volumes, such as a cloud storage, or a combination of such physical storage volumes and virtual storage volumes or arrays thereof), user interface (UI) 123, one or more communications interfaces 118, and communication bus 150. User interface 123 may include graphical user interface (GUI) 124 (e.g., a touchscreen, a display, etc.) and one or more input/output (I/O) devices 126 (e.g., a mouse, a keyboard, a microphone, one or more speakers, one or more cameras, one or more biometric scanners, one or more environmental sensors, one or more accelerometers, etc.). Non-volatile memory 128 stores operating system 115, one or more applications 116, and data 117 such that, for example, computer instructions of operating system 115 and/or applications 116 are executed by processor(s) 103 out of volatile memory 122. In some embodiments, volatile memory 122 may include one or more types of RAM and/or a cache memory that may offer a faster response time than a main memory. Data may be entered using an input device of GUI 124 or received from I/O device(s) 126. Various elements of computer 101 may communicate via one or more communication buses, shown as communication bus 150.

Computer 101 as shown in FIG. 1 is shown merely as an example, as clients, servers, intermediary and other networking devices and may be implemented by any computing or processing environment and with any type of machine or set of machines that may have suitable hardware and/or software capable of operating as described herein. Processor(s) 103 may be implemented by one or more programmable processors to execute one or more executable instructions, such as a computer program, to perform the functions of the system. As used herein, the term "processor" describes circuitry that performs a function, an operation, or a sequence of operations. The function, operation, or sequence of operations may be hard coded into the circuitry or soft coded by way of instructions held in a memory

device and executed by the circuitry. A "processor" may perform the function, operation, or sequence of operations using digital values and/or using analog signals. In some embodiments, the "processor" can be embodied in one or more application specific integrated circuits (ASICs), microprocessors, digital signal processors (DSPs), graphics processing units (GPUs), microcontrollers, field programmable gate arrays (FPGAs), programmable logic arrays (PLAs), multi-core processors, or general-purpose computers with associated memory. The "processor" may be analog, digital or mixed-signal. In some embodiments, the "processor" may be one or more physical processors or one or more "virtual" (e.g., remotely located or "cloud") processors. A processor including multiple processor cores and/or multiple processors multiple processors may provide functionality for parallel, simultaneous execution of instructions or for parallel, simultaneous execution of one instruction on more than one piece of data.

Communications interfaces 118 may include one or more interfaces to enable computer 101 to access a computer network such as a Local Area Network (LAN), a Wide Area Network (WAN), a Personal Area Network (PAN), or the Internet through a variety of wired and/or wireless or cellular connections.

In described embodiments, the computing device 101 may execute an application on behalf of a user of a client computing device. For example, the computing device 101 may execute a virtual machine, which provides an execution session within which applications execute on behalf of a user or a client computing device, such as a hosted desktop session. The computing device 101 may also execute a terminal services session to provide a hosted desktop environment. The computing device 101 may provide access to a computing environment including one or more of: one or more applications, one or more desktop applications, and one or more desktop sessions in which one or more applications may execute.

Additional details of the implementation and operation of network environment, computer 101 and client and server computers may be as described in U.S. Patent No. 9,538,345, issued January 3, 2017 to Citrix Systems, Inc. of Fort Lauderdale, FL, the teachings of which are hereby incorporated herein by reference.

B. Systems and Methods for an Embedded Browser

The present disclosure is directed towards systems and methods of an embedded browser. A client application executing on a client device can allow a user to access applications (apps) that are served from and/or hosted on one or more servers, such as web applications and software-as-a-service (SaaS) applications (hereafter sometimes generally referred to as network applications). A browser that is embedded or integrated with the client application can render to the user a network application that is accessed or requested via the client application, and can enable interactivity between the user and the network application. The browser is sometimes referred to as an embedded browser, and the client application with embedded browser (CEB) is sometimes referred to as a workspace application. The client application can establish a secure connection to the one or more servers to provide an application session for the user to access the network application using the client device and the embedded browser. The embedded browser can be integrated with the client application to ensure that traffic related to the network application is routed through and/or processed in the client application, which can provide the client application with real-time visibility to the traffic (e.g., when decrypted through the client application), and user interactions and behavior. The embedded browser can provide a seamless experience to a user as the network application is requested via the user interface (shared by the client application and the embedded browser) and rendered through the embedded browser within the same user interface.

The client application can terminate one end of a secured connection established with a server of a network application, such as a secure sockets layer (SSL) virtual private network (VPN) connection. The client application can receive encrypted traffic from the network application, and can decrypt the traffic before further processing (e.g., rendering by the embedded browser). The client application can monitor the received traffic (e.g., in encrypted packet form), and also have full visibility into the decrypted data stream and/or the SSL stack. This visibility can allow the client application to perform or facilitate policy-based management (e.g., including data loss prevention (DLP) capabilities), application control (e.g., to improve performance, service level), and collection and production of analytics. For instance, the local CEB can provide an information technology (IT) administrator with a controlled system for deploying web and SaaS applications through the CEB, and allow the IT administrator to set policies or configurations via the CEB for performing any of the forgoing activities.

Many web and SaaS delivered applications connect from web servers to generic browsers (e.g., Internet Explorer, Firefox, and so on) of users. Once authenticated, the entire session of such a network application is encrypted. However, in this scenario, an administrator may not have visibility, analytics, or control of the content entering the network application from the user's digital workspace, or the content leaving the network application and entering the user's digital workspace. Moreover, content of a network application viewed in a generic browser can be copied or downloaded (e.g., by a user or program) to potentially any arbitrary application or device, resulting in a possible breach in data security.

This present systems and methods can ensure that traffic associated with a network application is channeled through a CEB. By way of illustration, when a user accesses a SaaS web service with security assertion markup language (SAML) enabled for instance, the corresponding access request can be forwarded to a designated gateway service that determines, checks or verifies if the CEB was used to make the access request. Responsive to determining that a CEB was used to make the access request, the gateway service can perform or provide authentication and single-sign-on (SSO), and can allow the CEB to connect directly to the SaaS web service. Encryption (e.g., standard encryption) can be used for the application session between the CEB and the SaaS web service. When the content from the web service is unencrypted in the CEB to the viewed via the embedded browser, and/or when input is entered via the CEB, the CEB can provide added services on selective application-related information for control and analytics for instance. For example, an analytics agent or application programming interface (API) can be embedded in the CEB to provide or perform the added services.

The CEB (sometimes referred to as workspace application or receiver) can interoperate with one or more gateway services, intermediaries and/or network servers (sometimes collectively referred to as cloud services or Citrix Cloud) to provide access to a network application. Features and elements of an environment related to the operation of an embodiment of cloud services are described below.

FIG. 2 illustrates an embodiment of cloud services for use in accessing resources including network applications. The cloud services can include an enterprise mobility technical architecture 200, which can include an access gateway 260 in one illustrative embodiment. The architecture can be used in a bring-your-own-device (BYOD)

environment for instance. The architecture can enable a user of a client device 202 (e.g., a mobile or other device) to both access enterprise or personal resources from a client device 202, and use the client device 202 for personal use. The user may access such enterprise resources 204 or enterprise services 208 via a client application executing on the client device 202. The user may access such enterprise resources 204 or enterprise services 208 using a client device 202 that is purchased by the user or a client device 202 that is provided by the enterprise to user. The user may utilize the client device 202 for business use only or for business and personal use. The client device may run an iOS operating system, and Android operating system, or the like. The enterprise may choose to implement policies to manage the client device 202. The policies may be implanted through a firewall or gateway in such a way that the client device may be identified, secured or security verified, and provided selective or full access to the enterprise resources. The policies may be client device management policies, mobile application management policies, mobile data management policies, or some combination of client device, application, and data management policies. A client device 202 that is managed through the application of client device management policies may be referred to as an enrolled device. The client device management policies can be applied via the client application for instance.

In some embodiments, the operating system of the client device may be separated into a managed partition 210 and an unmanaged partition 212. The managed partition 210 may have policies applied to it to secure the applications running on and data stored in the managed partition. The applications running on the managed partition may be secure applications. In other embodiments, all applications may execute in accordance with a set of one or more policy files received separate from the application, and which define one or more security parameters, features, resource restrictions, and/or other access controls that are enforced by the client device management system when that application is executing on the device. By operating in accordance with their respective policy file(s), each application may be allowed or restricted from communications with one or more other applications and/or resources, thereby creating a virtual partition. Thus, as used herein, a partition may refer to a physically partitioned portion of memory (physical partition), a logically partitioned portion of memory (logical partition), and/or a virtual partition created as a result of enforcement of one or more policies and/or policy files across multiple apps as described herein (virtual partition). Stated differently, by enforcing policies on managed apps, those apps may be restricted to only be able to communicate with other managed apps and trusted

enterprise resources, thereby creating a virtual partition that is not accessible by unmanaged apps and devices.

The secure applications may be email applications, web browsing applications, software-as-a-service (SaaS) access applications, Windows Application access applications, and the like. The client application can include a secure application launcher 218. The secure applications may be secure native applications 214, secure remote applications 222 executed by the secure application launcher 218, virtualization applications 226 executed by the secure application launcher 218, and the like. The secure native applications 214 may be wrapped by a secure application wrapper 220. The secure application wrapper 220 may include integrated policies that are executed on the client device 202 when the secure native application is executed on the device. The secure application wrapper 220 may include meta-data that points the secure native application 214 running on the client device 202 to the resources hosted at the enterprise that the secure native application 214 may require to complete the task requested upon execution of the secure native application 214. The secure remote applications 222 executed by a secure application launcher 218 may be executed within the secure application launcher application 218. The virtualization applications 226 executed by a secure application launcher 218 may utilize resources on the client device 202, at the enterprise resources 204, and the like. The resources used on the client device 202 by the virtualization applications 226 executed by a secure application launcher 218 may include user interaction resources, processing resources, and the like. The user interaction resources may be used to collect and transmit keyboard input, mouse input, camera input, tactile input, audio input, visual input, gesture input, and the like. The processing resources may be used to present a user interface, process data received from the enterprise resources 204, and the like. The resources used at the enterprise resources 204 by the virtualization applications 226 executed by a secure application launcher 218 may include user interface generation resources, processing resources, and the like. The user interface generation resources may be used to assemble a user interface, modify a user interface, refresh a user interface, and the like. The processing resources may be used to create information, read information, update information, delete information, and the like. For example, the virtualization application may record user interactions associated with a graphical user interface (GUI) and communicate them to a server application where the server application may use the user interaction data as an input to the application operating on the server. In this arrangement, an enterprise may elect to maintain the application on the

server side as well as data, files, etc., associated with the application. While an enterprise may elect to "mobilize" some applications in accordance with the principles herein by securing them for deployment on the client device (e.g., via the client application), this arrangement may also be elected for certain applications. For example, while some applications may be secured for use on the client device, others might not be prepared or appropriate for deployment on the client device so the enterprise may elect to provide the mobile user access to the unprepared applications through virtualization techniques. As another example, the enterprise may have large complex applications with large and complex data sets (e.g., material resource planning applications) where it would be very difficult, or otherwise undesirable, to customize the application for the client device so the enterprise may elect to provide access to the application through virtualization techniques. As yet another example, the enterprise may have an application that maintains highly secured data (e.g., human resources data, customer data, engineering data) that may be deemed by the enterprise as too sensitive for even the secured mobile environment so the enterprise may elect to use virtualization techniques to permit mobile access to such applications and data. An enterprise may elect to provide both fully secured and fully functional applications on the client device. The enterprise can use a client application, which can include a virtualization application, to allow access to applications that are deemed more properly operated on the server side. In an embodiment, the virtualization application may store some data, files, etc., on the mobile phone in one of the secure storage locations. An enterprise, for example, may elect to allow certain information to be stored on the phone while not permitting other information.

In connection with the virtualization application, as described herein, the client device may have a virtualization application that is designed to present GUIs and then record user interactions with the GUI. The virtualization application may communicate the user interactions to the server side to be used by the server side application as user interactions with the application. In response, the application on the server side may transmit back to the client device a new GUI. For example, the new GUI may be a static page, a dynamic page, an animation, or the like, thereby providing access to remotely located resources.

The secure applications may access data stored in a secure data container 228 in the managed partition 210 of the client device. The data secured in the secure data container may be accessed by the secure wrapped applications 214, applications executed by a secure

application launcher 222, virtualization applications 226 executed by a secure application launcher 218, and the like. The data stored in the secure data container 228 may include files, databases, and the like. The data stored in the secure data container 228 may include data restricted to a specific secure application 230, shared among secure applications 232, and the like. Data restricted to a secure application may include secure general data 234 and highly secure data 238. Secure general data may use a strong form of encryption such as Advanced Encryption Standard (AES) 128-bit encryption or the like, while highly secure data 238 may use a very strong form of encryption such as AES 256-bit encryption. Data stored in the secure data container 228 may be deleted from the device upon receipt of a command from the device manager 224. The secure applications may have a dual-mode option 240. The dual mode option 240 may present the user with an option to operate the secured application in an unsecured or unmanaged mode. In an unsecured or unmanaged mode, the secure applications may access data stored in an unsecured data container 242 on the unmanaged partition 212 of the client device 202. The data stored in an unsecured data container may be personal data 244. The data stored in an unsecured data container 242 may also be accessed by unsecured applications 248 that are running on the unmanaged partition 212 of the client device 202. The data stored in an unsecured data container 242 may remain on the client device 202 when the data stored in the secure data container 228 is deleted from the client device 202. An enterprise may want to delete from the client device selected or all data, files, and/or applications owned, licensed or controlled by the enterprise (enterprise data) while leaving or otherwise preserving personal data, files, and/or applications owned, licensed or controlled by the user (personal data). This operation may be referred to as a selective wipe. With the enterprise and personal data arranged in accordance to the aspects described herein, an enterprise may perform a selective wipe.

The client device 202 may connect to enterprise resources 204 and enterprise services 208 at an enterprise, to the public Internet 248, and the like. The client device may connect to enterprise resources 204 and enterprise services 208 through virtual private network connections. The virtual private network connections, also referred to as microVPN or application-specific VPN, may be specific to particular applications (e.g., as illustrated by microVPNs 250), particular devices, particular secured areas on the client device (e.g., as illustrated by O/S VPN 252), and the like. For example, each of the wrapped applications in the secured area of the phone may access enterprise resources through an application specific VPN such that access to the VPN would be granted based on attributes

associated with the application, possibly in conjunction with user or device attribute information. The virtual private network connections may carry Microsoft Exchange traffic, Microsoft Active Directory traffic, HyperText Transfer Protocol (HTTP) traffic, HyperText Transfer Protocol Secure (HTTPS) traffic, application management traffic, and the like. The virtual private network connections may support and enable single-sign-on authentication processes 254. The single-sign-on processes may allow a user to provide a single set of authentication credentials, which are then verified by an authentication service 258. The authentication service 258 may then grant to the user access to multiple enterprise resources 204, without requiring the user to provide authentication credentials to each individual enterprise resource 204.

The virtual private network connections may be established and managed by an access gateway 260. The access gateway 260 may include performance enhancement features that manage, accelerate, and improve the delivery of enterprise resources 204 to the client device 202. The access gateway may also re-route traffic from the client device 202 to the public Internet 248, enabling the client device 202 to access publicly available and unsecured applications that run on the public Internet 248. The client device may connect to the access gateway via a transport network 262. The transport network 262 may use one or more transport protocols and may be a wired network, wireless network, cloud network, local area network, metropolitan area network, wide area network, public network, private network, and the like.

The enterprise resources 204 may include email servers, file sharing servers, SaaS/Web applications, Web application servers, Windows application servers, and the like. Email servers may include Exchange servers, Lotus Notes servers, and the like. File sharing servers may include ShareFile servers, and the like. SaaS applications may include Salesforce, and the like. Windows application servers may include any application server that is built to provide applications that are intended to run on a local Windows operating system, and the like. The enterprise resources 204 may be premise-based resources, cloud based resources, and the like. The enterprise resources 204 may be accessed by the client device 202 directly or through the access gateway 260. The enterprise resources 204 may be accessed by the client device 202 via a transport network 262. The transport network 262 may be a wired network, wireless network, cloud network, local area network, metropolitan area network, wide area network, public network, private network, and the like.

Cloud services can include an access gateway 260 and/or enterprise services 208. The enterprise services 208 may include authentication services 258, threat detection services 264, device manager services 224, file sharing services 268, policy manager services 270, social integration services 272, application controller services 274, and the like. Authentication services 258 may include user authentication services, device authentication services, application authentication services, data authentication services and the like. Authentication services 258 may use certificates. The certificates may be stored on the client device 202, by the enterprise resources 204, and the like. The certificates stored on the client device 202 may be stored in an encrypted location on the client device, the certificate may be temporarily stored on the client device 202 for use at the time of authentication, and the like. Threat detection services 264 may include intrusion detection services, unauthorized access attempt detection services, and the like. Unauthorized access attempt detection services may include unauthorized attempts to access devices, applications, data, and the like. Device management services 224 may include configuration, provisioning, security, support, monitoring, reporting, and decommissioning services. File sharing services 268 may include file management services, file storage services, file collaboration services, and the like. Policy manager services 270 may include device policy manager services, application policy manager services, data policy manager services, and the like. Social integration services 272 may include contact integration services, collaboration services, integration with social networks such as Facebook, Twitter, and LinkedIn, and the like. Application controller services 274 may include management services, provisioning services, deployment services, assignment services, revocation services, wrapping services, and the like.

The enterprise mobility technical architecture 200 may include an application store 278. The application store 278 may include unwrapped applications 280, pre-wrapped applications 282, and the like. Applications may be populated in the application store 278 from the application controller 274. The application store 278 may be accessed by the client device 202 through the access gateway 260, through the public Internet 248, or the like. The application store may be provided with an intuitive and easy to use User Interface.

A software development kit 284 may provide a user the capability to secure applications selected by the user by providing a secure wrapper around the application. An application that has been wrapped using the software development kit 284 may then be

made available to the client device 202 by populating it in the application store 278 using the application controller 274.

The enterprise mobility technical architecture 200 may include a management and analytics capability. The management and analytics capability may provide information related to how resources are used, how often resources are used, and the like. Resources may include devices, applications, data, and the like. How resources are used may include which devices download which applications, which applications access which data, and the like. How often resources are used may include how often an application has been downloaded, how many times a specific set of data has been accessed by an application, and the like.

FIG. 3 depicts is an illustrative embodiment of an enterprise mobility management system 300. Some of the components of the mobility management system 200 described above with reference to Figure 2 have been omitted for the sake of simplicity. The architecture of the system 300 depicted in Figure 3 is similar in many respects to the architecture of the system 200 described above with reference to Figure 2 and may include additional features not mentioned above.

In this case, the left hand side represents an enrolled client device 302 with a client agent 304, which interacts with gateway server 306 to access various enterprise resources 308 and services 309 such as Web or SaaS applications, Exchange, Sharepoint, public-key infrastructure (PKI) Resources, Kerberos Resources, Certificate Issuance service, as shown on the right hand side above. The gateway server 306 can include embodiments of features and functionalities of the cloud services, such as access gateway 260 and application controller functionality. Although not specifically shown, the client agent 304 may be part of, and/or interact with the client application which can operate as an enterprise application store (storefront) for the selection and/or downloading of network applications.

The client agent 304 can act as a UI (user interface) intermediary for Windows apps/desktops hosted in an Enterprise data center, which are accessed using the High-Definition User Experience (HDX) or Independent Computing Architecture (ICA) display remoting protocol. The client agent 304 can also support the installation and management of native applications on the client device 302, such as native iOS or Android applications. For example, the managed applications 310 (mail, browser, wrapped application) shown in the figure above are native applications that execute locally on the device. Client agent 304 and application management framework of this architecture act to provide policy driven

management capabilities and features such as connectivity and SSO (single sign on) to enterprise resources/services 308. The client agent 304 handles primary user authentication to the enterprise, for instance to access gateway (AG) with SSO to other gateway server components. The client agent 304 obtains policies from gateway server 306 to control the behavior of the managed applications 310 on the client device 302.

The Secure interprocess communication (IPC) links 312 between the native applications 310 and client agent 304 represent a management channel, which allows client agent to supply policies to be enforced by the application management framework 314 "wrapping" each application. The IPC channel 312 also allows client agent 304 to supply credential and authentication information that enables connectivity and SSO to enterprise resources 308. Finally the IPC channel 312 allows the application management framework 314 to invoke user interface functions implemented by client agent 304, such as online and offline authentication.

Communications between the client agent 304 and gateway server 306 are essentially an extension of the management channel from the application management framework 314 wrapping each native managed application 310. The application management framework 314 requests policy information from client agent 304, which in turn requests it from gateway server 306. The application management framework 314 requests authentication, and client agent 304 logs into the gateway services part of gateway server 306 (also known as NetScaler access gateway). Client agent 304 may also call supporting services on gateway server 306, which may produce input material to derive encryption keys for the local data vaults 316, or provide client certificates which may enable direct authentication to PKI protected resources, as more fully explained below.

In more detail, the application management framework 314 "wraps" each managed application 310. This may be incorporated via an explicit build step, or via a post-build processing step. The application management framework 314 may "pair" with client agent 304 on first launch of an application 310 to initialize the Secure IPC channel and obtain the policy for that application. The application management framework 314 may enforce relevant portions of the policy that apply locally, such as the client agent login dependencies and some of the containment policies that restrict how local OS services may be used, or how they may interact with the application 310.

The application management framework 314 may use services provided by client agent 304 over the Secure IPC channel 312 to facilitate authentication and internal network

access. Key management for the private and shared data vaults 316 (containers) may be also managed by appropriate interactions between the managed applications 310 and client agent 304. Vaults 316 may be available only after online authentication, or may be made available after offline authentication if allowed by policy. First use of vaults 316 may require online authentication, and offline access may be limited to at most the policy refresh period before online authentication is again required.

Network access to internal resources may occur directly from individual managed applications 310 through access gateway 306. The application management framework 314 is responsible for orchestrating the network access on behalf of each application 310. Client agent 304 may facilitate these network connections by providing suitable time limited secondary credentials obtained following online authentication. Multiple modes of network connection may be used, such as reverse web proxy connections and end-to-end VPN-style tunnels 318.

The Mail and Browser managed applications 310 can have special status and may make use of facilities that might not be generally available to arbitrary wrapped applications. For example, the Mail application may use a special background network access mechanism that allows it to access Exchange over an extended period of time without requiring a full AG logon. The Browser application may use multiple private data vaults to segregate different kinds of data.

This architecture can support the incorporation of various other security features. For example, gateway server 306 (including its gateway services) in some cases might not need to validate active directory (AD) passwords. It can be left to the discretion of an enterprise whether an AD password is used as an authentication factor for some users in some situations. Different authentication methods may be used if a user is online or offline (i.e., connected or not connected to a network).

Step up authentication is a feature wherein gateway server 306 may identify managed native applications 310 that are allowed to have access to more sensitive data using strong authentication, and ensure that access to these applications is only permitted after performing appropriate authentication, even if this means a re-authentication is requested from the user after a prior weaker level of login.

Another security feature of this solution is the encryption of the data vaults 316 (containers) on the client device 302. The vaults 316 may be encrypted so that all on-device

data including clipboard/cache data, files, databases, and configurations are protected. For on-line vaults, the keys may be stored on the server (gateway server 306), and for off-line vaults, a local copy of the keys may be protected by a user password or biometric validation. When data is stored locally on the device 302 in the secure container 316, it is preferred that a minimum of AES 256 encryption algorithm be utilized.

Other secure container features may also be implemented. For example, a logging feature may be included, wherein all security events happening inside an application 310 are logged and reported to the backend. Data wiping may be supported, such as if the application 310 detects tampering, associated encryption keys may be written over with random data, leaving no hint on the file system that user data was destroyed. Screenshot protection is another feature, where an application may prevent any data from being stored in screenshots. For example, the key window's hidden property may be set to YES. This may cause whatever content is currently displayed on the screen to be hidden, resulting in a blank screenshot where any content would normally reside.

Local data transfer may be prevented, such as by preventing any data from being locally transferred outside the application container, e.g., by copying it or sending it to an external application. A keyboard cache feature may operate to disable the autocorrect functionality for sensitive text fields. SSL certificate validation may be operable so the application specifically validates the server SSL certificate instead of it being stored in the keychain. An encryption key generation feature may be used such that the key used to encrypt data on the device is generated using a passphrase or biometric data supplied by the user (if offline access is required). It may be XORed with another key randomly generated and stored on the server side if offline access is not required. Key Derivation functions may operate such that keys generated from the user password use KDFs (key derivation functions, notably Password-Based Key Derivation Function 2 (PBKDF2)) rather than creating a cryptographic hash of it. The latter makes a key susceptible to brute force or dictionary attacks.

Further, one or more initialization vectors may be used in encryption methods. An initialization vector might cause multiple copies of the same encrypted data to yield different cipher text output, preventing both replay and cryptanalytic attacks. This may also prevent an attacker from decrypting any data even with a stolen encryption key. Further, authentication then decryption may be used, wherein application data is decrypted only after the user has authenticated within the application. Another feature may relate to sensitive

data in memory, which may be kept in memory (and not in disk) only when it's needed. For example, login credentials may be wiped from memory after login, and encryption keys and other data inside objective-C instance variables are not stored, as they may be easily referenced. Instead, memory may be manually allocated for these.

An inactivity timeout may be implemented via the CEB, wherein after a policy-defined period of inactivity, a user session is terminated.

Data leakage from the application management framework 314 may be prevented in other ways. For example, when an application 310 is put in the background, the memory may be cleared after a predetermined (configurable) time period. When backgrounded, a snapshot may be taken of the last displayed screen of the application to fasten the foregrounding process. The screenshot may contain confidential data and hence should be cleared.

Another security feature relates to the use of an OTP (one time password) 320 without the use of an AD (active directory) 322 password for access to one or more applications. In some cases, some users do not know (or are not permitted to know) their AD password, so these users may authenticate using an OTP 320 such as by using a hardware OTP system like SecurID (OTPs may be provided by different vendors also, such as Entrust or Gemalto). In some cases, after a user authenticates with a user ID, a text is sent to the user with an OTP 320. In some cases, this may be implemented only for online use, with a prompt being a single field.

An offline password may be implemented for offline authentication for those applications 310 for which offline use is permitted via enterprise policy. For example, an enterprise may want storefront to be accessed in this manner. In this case, the client agent 304 may require the user to set a custom offline password and the AD password is not used. Gateway server 306 may provide policies to control and enforce password standards with respect to the minimum length, character class composition, and age of passwords, such as described by the standard Windows Server password complexity requirements, although these requirements may be modified.

Another feature relates to the enablement of a client side certificate for certain applications 310 as secondary credentials (for the purpose of accessing PKI protected web resources via the application management framework micro VPN feature). For example, an application may utilize such a certificate. In this case, certificate-based authentication using

ActiveSync protocol may be supported, wherein a certificate from the client agent 304 may be retrieved by gateway server 306 and used in a keychain. Each managed application may have one associated client certificate, identified by a label that is defined in gateway server 306.

Gateway server 306 may interact with an Enterprise special purpose web service to support the issuance of client certificates to allow relevant managed applications to authenticate to internal PKI protected resources.

The client agent 304 and the application management framework 314 may be enhanced to support obtaining and using client certificates for authentication to internal PKI protected network resources. More than one certificate may be supported, such as to match various levels of security and/or separation requirements. The certificates may be used by the Mail and Browser managed applications, and ultimately by arbitrary wrapped applications (provided those applications use web service style communication patterns where it is reasonable for the application management framework to mediate https requests).

Application management client certificate support on iOS may rely on importing a public-key cryptography standards (PKCS) 12 BLOB (Binary Large Object) into the iOS keychain in each managed application for each period of use. Application management framework client certificate support may use a HTTPS implementation with private in-memory key storage. The client certificate might never be present in the iOS keychain and might not be persisted except potentially in "online-only" data value that is strongly protected.

Mutual SSL or TLS may also be implemented to provide additional security by requiring that a client device 302 is authenticated to the enterprise, and vice versa. Virtual smart cards for authentication to gateway server 306 may also be implemented.

Both limited and full Kerberos support may be additional features. The full support feature relates to an ability to do full Kerberos login to Active Directory (AD) 322, using an AD password or trusted client certificate, and obtain Kerberos service tickets to respond to HTTP Negotiate authentication challenges. The limited support feature relates to constrained delegation in Citrix Access Gateway Enterprise Edition (AGEE), where AGEE supports invoking Kerberos protocol transition so it can obtain and use Kerberos service tickets (subject to constrained delegation) in response to HTTP Negotiate authentication challenges. This mechanism works in reverse web proxy (aka corporate virtual private

network (CVPN)) mode, and when http (but not https) connections are proxied in VPN and MicroVPN mode.

Another feature relates to application container locking and wiping, which may automatically occur upon jail-break or rooting detections, and occur as a pushed command from administration console, and may include a remote wipe functionality even when an application 310 is not running.

A multi-site architecture or configuration of enterprise application store and an application controller may be supported that allows users to be service from one of several different locations in case of failure.

In some cases, managed applications 310 may be allowed to access a certificate and private key via an API (example OpenSSL). Trusted managed applications 310 of an enterprise may be allowed to perform specific Public Key operations with an application's client certificate and private key. Various use cases may be identified and treated accordingly, such as when an application behaves like a browser and no certificate access is used, when an application reads a certificate for "who am I," when an application uses the certificate to build a secure session token, and when an application uses private keys for digital signing of important data (e.g., transaction log) or for temporary data encryption.

Referring now to FIG. 4, depicted is a block diagram of a system 400 of an embedded browser. In brief overview, the system 400 may include a client device 402 with a digital workspace for a user, a client application 404, cloud services 408 operating on at least one network device 432, and network applications 406 served from and/or hosted on one or more servers 430. The client application 404 can for instance include at least one of: an embedded browser 410, a networking agent 412, a cloud services agent 414, a remote session agent 416, or a secure container 418. The cloud services 408 can for instance include at least one of: secure browser(s) 420, an access gateway 422 (or CIS, e.g., for registering and/or authenticating the client application and/or user), or analytics services 424 (or CAS, e.g., for receiving information from the client application for analytics). The network applications 406 can include sanctioned applications 426 and non-sanctioned applications 428.

Each of the above-mentioned elements or entities is implemented in hardware, or a combination of hardware and software, in one or more embodiments. Each component of the system 400 may be implemented using hardware or a combination of hardware or

software detailed above in connection with FIG. 1. For instance, each of these elements or entities can include any application, program, library, script, task, service, process or any type and form of executable instructions executing on hardware of the client device 402, the at least one network device 432 and/or the one or more servers 430. The hardware includes circuitry such as one or more processors in one or more embodiments. For example, the at least one network device 432 and/or the one or more servers 430 can include any of the elements of a computing device described above in connection with at least FIG. 1 for instance.

The client device 402 can include any embodiment of a computing device described above in connection with at least FIG. 1 for instance. The client device 402 can include any user device such as a desktop computer, a laptop computer, a tablet device, a smart phone, or any other mobile or personal device. The client device 402 can include a digital workspace of a user, which can include file system(s), cache or memory (e.g., including electronic clipboard(s)), container(s), application(s) and/or other resources on the client device 402. The digital workspace can include or extend to one or more networks accessible by the client device 402, such as an intranet and the Internet, including file system(s) and/or other resources accessible via the one or more networks. A portion of the digital workspace can be secured via the use of the client application 404 with embedded browser 410 (CEB) for instance. The secure portion of the digital workspace can include for instance file system(s), cache or memory (e.g., including electronic clipboard(s)), application(s), container(s) and/or other resources allocated to the CEB, and/or allocated by the CEB to network application(s) 406 accessed via the CEB. The secure portion of the digital workspace can also include resources specified by the CEB (via one or more policies) for inclusion in the secure portion of the digital workspace (e.g., a particular local application can be specified via a policy to be allowed to receive data obtained from a network application).

The client application 404 can include one or more components, such as an embedded browser 410, a networking agent 412, a cloud services agent 414 (sometimes referred to as management agent), a remote session agent 416 (sometimes referred to as HDX engine), and/or a secure container 418 (sometimes referred to as secure cache container). One or more of the components can be installed as part of a software build or release of the client application 404 or CEB, or separately acquired or downloaded and installed/integrated into an existing installation of the client application 404 or CEB for

instance. For instance, the client device may download or otherwise receive the client application 404 (or any component) from the network device(s) 432. In some embodiments, the client device may send a request for the client application 404 to the network device(s) 432. For example, a user of the client device can initiate a request, download and/or installation of the client application. The network device(s) 432 in turn may send the client application to the client device. In some embodiments, the network device(s) 432 may send a setup or installation application for the client application to the client device. Upon receipt, the client device may install the client application onto a hard disk of the client device. In some embodiments, the client device may run the setup application to unpack or decompress a package of the client application. In some embodiments, the client application may be an extension (e.g., an add-on, an add-in, an applet or a plug-in) to another application (e.g., a networking agent 412) installed on the client device. The client device may install the client application to interface or inter-operate with the pre-installed application. In some embodiments, the client application may be a standalone application. The client device may install the client application to execute as a separate process.

The embedded browser 410 can include elements and functionalities of a web browser application or engine. The embedded browser 410 can locally render network application(s) as a component or extension of the client application. For instance, the embedded browser 410 can render a SaaS/Web application inside the CEB which can provide the CEB with full visibility and control of the application session. The embedded browser can be embedded or incorporated into the client application via any means, such as direct integration (e.g., programming language or script insertion) into the executable code of the client application, or via plugin installation. For example, the embedded browser can include a Chromium based browser engine or other type of browser engine, that can be embedded into the client application, using the Chromium embedded framework (CEF) for instance. The embedded browser can include a HTML5-based layout graphical user interface (GUI). The embedded browser can provide HTML rendering and JavaScript support to a client application incorporating various programming languages. For example, elements of the embedded browser can bind to a client application incorporating C, C++, Delphi, Go, Java, .NET / Mono, Visual Basic 6.0, and/or Python.

In some embodiments, the embedded browser comprises a plug-in installed on the client application. For example, the plug-in can include one or more components. One such components can be an ActiveX control or Java control or any other type and/or form of

executable instructions capable of loading into and executing in the client application. For example, the client application can load and run an Active X control of the embedded browser, such as in a memory space or context of the client application. In some embodiments, the embedded browser can be installed as an extension on the client application, and a user can choose to enable or disable the plugin or extension. The embedded browser (e.g., via the plugin or extension) can form or operate as a secured browser for securing, using and/or accessing resources within the secured portion of the digital workspace.

The embedded browser can incorporate code and functionalities beyond that available or possible in a standard or typical browser. For instance, the embedded browser can bind with or be assigned with a secured container 418, to define at least part of the secured portion of a user's digital workspace. The embedded browser can bind with or be assigned with a portion of the client device's cache to form a secured clipboard (e.g., local to the client device, or extendable to other devices), that can be at least part of the secured container 418. The embedded browser can be integrated with the client application to ensure that traffic related to network applications is routed through and/or processed in the client application, which can provide the client application with real-time visibility to the traffic (e.g., when decrypted through the client application). This visibility to the traffic can allow the client application to perform or facilitate policy-based management (e.g., including data loss prevention (DLP) capabilities), application control, and collection and production of analytics.

In some embodiments, the embedded browser incorporates one or more other components of the client application 404, such as the cloud services agent 414, remote session agent 416 and/or secure container 418. For instance, a user can use the cloud services agent 414 of the embedded browser to interoperate with the access gateway 422 (sometimes referred to as CIS) to access a network application. For example, the cloud services agent 414 can execute within the embedded browser, and can receive and transmit navigation commands from the embedded browser to a hosted network application. The cloud services agent can use a remote presentation protocol to display the output generated by the network application to the embedded browser. For example, the cloud services agent 414 can include a HTML5 web client that allows end users to access remote desktops and/or applications on the embedded browser.

The client application 404 and CEB operate on the application layer of the operational (OSI) stack of the client device. The client application 404 can include and/or execute one or more agents that interoperate with the cloud services 408. The client application 404 can receive, obtain, retrieve or otherwise access various policies (e.g., an enterprise's custom, specified or internal policies or rules) and/or data (e.g., from an access gateway 422 and/or network device(s) of cloud services 408, or other server(s), that may be managed by the enterprise). The client application can access the policies and/or data to control and/or manage a network application (e.g., a SaaS, web or remote-hosted application). Control and/or management of a network application can include control and/or management of various aspects of the network application, such as access control, session delivery, available features or functions, service level, traffic management and monitoring, and so on. The network application can be from a provider or vendor of the enterprise (e.g., salesforce.com, SAP, Microsoft Office 365), from the enterprise itself, or from another entity (e.g., Dropbox or Gmail service).

For example, the cloud services agent 414 can provide policy driven management capabilities and features related to the use and/or access of network applications. For example, the cloud services agent 414 can include a policy engine to apply one or more policies (e.g., received from cloud services) to determine access control and/or connectivity to resources such as network applications. When a session is established between the client application and a server 430 providing a SaaS application for instance, the cloud services agent 414 can apply one or more policies to control traffic levels and/or traffic types (or other aspects) of the session, for instance to manage a service level of the SaaS application. Additional aspects of the application traffic that can be controlled or managed can include encryption level and/or encryption type applied to the traffic, level of interactivity allowed for a user, limited access to certain features of the network application (e.g., print-screen, save, edit or copy functions), restrictions to use or transfer of data obtained from the network application, limit concurrent access to two or more network applications, limit access to certain file repositories or other resources, and so on.

The cloud services agent 414 can convey or feed information to analytics services 424 of the cloud services 408, such as information about SaaS interaction events visible to the CEB. Such a configuration using the CEB can monitor or capture information for analytics without having an inline device or proxy located between the client device and the server(s) 430, or using a SaaS API gateway 'out-of-band' approach. In some embodiments,

the cloud services agent 414 does not execute within the embedded browser. In these embodiments, a user can similarly use the cloud services agent 414 to interoperate with the access gateway (or CIS) 422 to access a network application. For instance, the cloud services agent 414 can register and/or authenticate with the access gateway (or CIS) 422, and can obtain a list of the network applications from the access gateway (or CIS) 422. The cloud services agent 414 can include and/or operate as an application store (or storefront) for user selection and/or downloading of network applications. Upon logging in to access a network application, the cloud services agent 414 can intercept and transmit navigation commands from the embedded browser to the network application. The cloud services agent can use a remote presentation protocol to display the output generated by the network application to the embedded browser. For example, the cloud services agent 414 can include a HTML5 web client that allows end users to access remote desktops and/or applications on the embedded browser.

In some embodiments, the cloud services agent 414 provides single sign on (SSO) capability for the user and/or client device to access a plurality of network applications. The cloud services agent 414 can perform user authentication to access network applications as well as other network resources and services, by communicating with the access gateway 422 for instance. For example, the cloud services agent 414 can authenticate or register with the access gateway 422, to access other components of the cloud services 408 and/or the network applications 406. Responsive to the authentication or registration, the access gateway 422 can perform authentication and/or SSO for (or on behalf of) the user and/or client application, with the network applications.

The client application 404 can include a networking agent 412. The networking agent 412 is sometimes referred to as a software-defined wide area network (SD-WAN) agent, mVPN agent, or microVPN agent. The networking agent 412 can establish or facilitate establishment of a network connection between the client application and one or more resources (e.g., server 430 serving a network application). The networking agent 412 can perform handshaking for a requested connection from the client application to access a network application, and can establish the requested connection (e.g., secure or encrypted connection). The networking agent 412 can connect to enterprise resources (including services) for instance via a virtual private network (VPN). For example, the networking agent 412 can establish a secure socket layer (SSL) VPN between the client application and a server 430 providing the network application 406. The VPN connections, sometimes

referred to as microVPN or application-specific VPN, may be specific to particular network applications, particular devices, particular secured areas on the client device, and the like, for instance as discussed above in connection with FIG. 3. Such VPN connections can carry Microsoft Exchange traffic, Microsoft Active Directory traffic, HyperText Transfer Protocol (HTTP) traffic, HyperText Transfer Protocol Secure (HTTPS) traffic, as some examples.

The remote session agent 416 (sometimes referred to as HDX engine) can include features of the client agent 304 discussed above in connection with FIG. 2 for instance, to support display a remoting protocol (e.g., HDX or ICA). In some embodiments, the remote session agent 416 can establish a remote desktop session and/or remote application session in accordance to any variety of protocols, such as the Remote Desktop Protocol (RDP), Appliance Link Protocol (ALP), Remote Frame Buffer (RFB) Protocol, and ICA Protocol. For example, the remote session agent 416 can establish a remote application session for a user of the client device to access an enterprise network application. The remote session agent 416 can establish the remote application session within or over a secure connection (e.g., a VPN) established by the networking agent 412 for instance.

The client application or CEB can include or be associated with a secure container 418. A secure container can include a logical or virtual delineation of one or more types of resources accessible within the client device and/or accessible by the client device. For example, the secure container 418 can refer to the entirety of the secured portion of the digital workspace, or particular aspect(s) of the secured portion. In some embodiments, the secure container 418 corresponds to a secure cache (e.g., electronic or virtual clipboard), and can dynamically incorporate a portion of a local cache of each client device of a user, and/or a cloud-based cache of the user, that is protected or secured (e.g., encrypted). The secure container can define a portion of file system(s), and/or delineate resources allocated to a CEB and/or to network applications accessed via the CEB. The secure container can include elements of the secure data container 228 discussed above in connection with FIG. 2 for example. The CEB can be configured (e.g., via policies) to limit, disallow or disable certain actions or activities on resources and/or data identified to be within a secure container. A secured container can be defined to specify that the resources and/or data within the secure container are to be monitored for misuse, abuse and/or exfiltration.

In certain embodiments, a secure container relates to or involves the use of a secure browser (e.g., embedded browser 410 or secure browser 420) that implements various

enterprise security features. Network applications (or web pages accessed by the secure browser) that are configured to run within the secure browser can effectively inherit the security mechanisms implemented by the secure browser. These network applications can be considered to be contained within the secure container. The use of such a secure browser can enable an enterprise to implement a content filtering policy in which, for example, employees are blocked from accessing certain web sites from their client devices. The secure browser can be used, for example, to enable client device users to access a corporate intranet without the need for a VPN.

In some embodiments, a secure container can support various types of remedial actions for protecting enterprise resources. One such remedy is to lock the client device, or a secure container on the client device that stores data to be protected, such that the client device or secure container can only be unlocked with a valid code provided by an administrator for instance. In some embodiments, these and other types of remedies can be invoked automatically based on conditions detected on the client device (via the application of policies for instance), or can be remotely initiated by an administrator.

In some embodiments, a secure container can include a secure document container for documents. A document can comprise any computer-readable file including text, audio, video, and/or other types of information or media. A document can comprise any single one or combination of these media types. As explained herein, the secure container can help prevent the spread of enterprise information to different applications and components of the client device, as well as to other devices. The enterprise system (which can be partially or entirely within a cloud network) can transmit documents to various devices, which can be stored within the secure container. The secure container can prevent unauthorized applications and other components of the client device from accessing information within the secure container. For enterprises that allow users to use their own client devices for accessing, storing, and using enterprise data, providing secure container on the client devices helps to secure the enterprise data. For instance, providing secure containers on the client devices can centralize enterprise data in one location on each client device, and can facilitate selective or complete deletion of enterprise data from each client device when desired.

The secure container can include an application that implements a file system that stores documents and/or other types of files. The file system can comprise a portion of a

computer-readable memory of the client device. The file system can be logically separated from other portions of the computer-readable memory of the client device. In this way, enterprise data can be stored in a secure container and private data can be stored in a separate portion of the computer-readable memory of the client device for instance. The secure container can allow the CEB, network applications accessed via the CEB, locally installed applications and/or other components of the client device to read from, write to, and/or delete information from the file system (if authorized to do so). Deleting data from the secure container can include deleting actual data stored in the secure container, deleting pointers to data stored in the secure container, deleting encryption keys used to decrypt data stored in the secure container, and the like. The secure container can be installed by, e.g., the client application, an administrator, or the client device manufacturer. The secure container can enable some or all of the enterprise data stored in the file system to be deleted without modifying private data stored on the client device outside of the secure container. The file system can facilitate selective or complete deletion of data from the file system. For example, an authorized component of the enterprise's system can delete data from the file system based on, e.g., encoded rules. In some embodiments, the client application can delete the data from the file system, in response to receiving a deletion command from the enterprise's system.

The secure container can include an access manager that governs access to the file system by applications and other components of the client device. Access to the file system can be governed based on document access policies (e.g., encoded rules) maintained by the client application, in the documents and/or in the file system. A document access policy can limit access to the file system based on (1) which application or other component of the client device is requesting access, (2) which documents are being requested, (3) time or date, (4) geographical position of the client device, (5) whether the requesting application or other component provides a correct certificate or credentials, (6) whether the user of the client device provides correct credentials, (7) other conditions, or any combination thereof. A user's credentials can comprise, for example, a password, one or more answers to security questions (e.g., What is the mascot of your high school?), biometric information (e.g., fingerprint scan, eye-scan), and the like. Hence, by using the access manager, the secure container can be configured to be accessed only by applications that are authorized to access the secure container. As one example, the access manager can enable enterprise

applications installed on the client device to access data stored in the secure container and to prevent non-enterprise applications from accessing the data stored in the secure container.

Temporal and geographic restrictions on document access may be useful. For example, an administrator may deploy a document access policy that restricts the availability of the documents (stored within the secure container) to a specified time window and/or a geographic zone (e.g., as determined by a GPS chip) within which the client device must reside in order to access the documents. Further, the document access policy can instruct the secure container or client application to delete the documents from the secure container or otherwise make them unavailable when the specified time period expires or if the client device is taken outside of the defined geographic zone.

Some documents can have access policies that forbid the document from being saved within the secure container. In such embodiments, the document can be available for viewing on the client device only when the user is logged in or authenticated via the cloud services for example.

The access manager can also be configured to enforce certain modes of connectivity between remote devices (e.g., an enterprise resource or other enterprise server) and the secure container. For example, the access manager can require that documents received by the secure container from a remote device and/or sent from the secure container to the remote device be transmitted through secured tunnels/connections, for example. The access manager can require that all documents transmitted to and from the secure container be encrypted. The client application or access manager can be configured to encrypt documents sent from the secure container and decrypt documents sent to the secure container. Documents in the secure container can also be stored in an encrypted form.

The secure container can be configured to prevent documents or data included within documents or the secure container from being used by unauthorized applications or components of the client device or other devices. For instance, a client device application having authorization to access documents from the secure container can be programmed to prevent a user from copying a document's data and pasting it into another file or application interface, or locally saving the document or document data as a new file outside of the secure container. Similarly, the secure container can include a document viewer and/or editor that do not permit such copy/paste and local save operations. Moreover, the access

manager can be configured to prevent such copy/paste and local save operations. Further, the secure container and applications programmed and authorized to access documents from the secure container can be configured to prevent users from attaching such documents to emails or other forms of communication.

One or more applications (e.g., applications installed on the client device, and/or network applications accessed via the CEB) can be programmed or controlled (e.g., via policy-based enforcement) to write enterprise-related data only into the secure container. For instance, an application's source code can be provided with the resource name of the secure container. Similarly, a remote application (e.g., executing on a device other than the client device) can be configured to send data or documents only to the secure container (as opposed to other components or memory locations of the client device). Storing data to the secure container can occur automatically, for example, under control of the application, the client application, and/or the secure browser. The client application can be programmed to encrypt or decrypt documents stored or to be stored within the secure container. In certain embodiments, the secure container can only be used by applications (on the client device or a remote device) that are programmed to identify and use the secure container, and which have authorization to do so.

The network applications 406 can include sanctioned network applications 426 and non-sanctioned network applications 428. By way of a non-limiting example, sanctioned network applications 426 can include network applications from Workday, Salesforce, Office 365, SAP, and so on, while non-sanctioned network applications 426 can include network applications from Dropbox, Gmail, and so on. For instance, FIG. 4 illustrates a case where sanctioned applications 426 are accessed via a CEB. In operation (1), a user instance of a client application 404, that is installed on client device 402, can register or authenticate with the access gateway 422 of cloud services 408. For example, the user can authenticate the user to the client device and login to the client device 402. The client application can automatically execute, or be activated by the user. In some embodiments, the user can sign in to the client application (e.g., by authenticating the user to the client application). In response to the login or sign-in, the client application can register or authenticate the user and/or the client application with the access gateway 422.

In operation (2), in response to the registration or authentication, the access gateway 422 can identify or retrieve a list of enumerated network applications available or pre-

assigned to the user, and can provide the list to the client application. For example, in response to the registration or authentication, the access gateway can identify the user and/or retrieve a user profile of the user. According to the identity and/or user profile, the access gateway can determine the list (e.g., retrieve a stored list of network applications matched with the user profile and/or the identity of the user). The list can correspond to a list of network applications sanctioned for the user. The access gateway can send the list to the client application or embedded browser, which can be presented via the client application or embedded browser to the user (e.g., in a storefront user interface) for selection.

In operation (3), the user can initiate connection to a sanctioned network application (e.g., a SaaS application), by selecting from the list of network applications presented to the user. For example, the user can click on an icon or other representation of the sanctioned network application, displayed via the client application or embedded browser. This user action can trigger the CEB to transmit a connection or access request to a server that provisions the network application. The request can include a request to the server (e.g., SaaS provider) to communicate with the access gateway to authenticate the user. The server can send a request to the access gateway to authenticate the user for example.

In operation (4), the access gateway can perform SSO with the server, to authenticate the user. For example, in response to the server's request to authenticate the user, the access gateway can provide credentials of the user to the server(s) 430 for SSO, to access the selected network application and/or other sanctioned network applications. In operation (5), the user can log into the selected network application, based on the SSO (e.g., using the credentials). The client application (e.g., the networking agent 412 and/or the remote session agent 416) can establish a secure connection and session with the server(s) 430 to access the selected network application. The CEB can decrypt application traffic received via the secure connection. The CEB can monitor traffic sent via the CEB and the secured connection to the servers 430.

In operation (6), the client application can provide information to the analytics services 424 of cloud services 408, for analytics processing. For example, the cloud services agent 414 of the client application 404 can monitor for or capture user interaction events with the selected network application. The cloud services agent 414 can convey the user interaction events to the analytics services 424, to be processed to produce analytics.

FIG. 5 depicts an example embodiment of a system for using a secure browser. In brief overview, the system includes cloud services 408, network applications 406 and client device 402. In some embodiments, various elements of the system are similar to that described above for FIG. 4, but that the client application (with embedded browser) is not available in the client device 402. A standard or typical browser may be available on the client device, from which a user can initiate a request to access a sanctioned network application for instance. A network application can be specified as being sanctioned or unsanctioned via policies that can be set by an administrator or automatically (e.g., via artificial intelligence).

For example, in operation (1), the user may log into the network application using the standard browser. For accessing a sanctioned network application, the user may access a predefined URL and/or corresponding webpage of a server that provisions the network application, via the standard browser, to initiate a request to access the network application. In some embodiments, the request can be forwarded to or intercepted by a designated gateway service (e.g., in a data path of the request). For example, the gateway service can reside on the client device (e.g., as an executable program), or can reside on a network device 432 of the cloud services 408 for instance. In some embodiments, the access gateway can correspond to or include the gateway service. The gateway service can determine if the requested network application is a sanctioned network application. The gateway service can determine if a CEB initiated the request. The gateway service can detect or otherwise determine that the request is initiated from a source (e.g., initiated by the standard browser) in the client device other than a CEB. In some embodiments, there is no requirement for a designated gateway service to detect or determine if the request is initiated from a CEB, for example if the requested network application is sanctioned, that user is initiating the request via a standard browser, and/or that the predefined URL and/or corresponding webpage is accessed.

In operation (2), the server may authenticate the user via the access gateway of the cloud services 408. The server may communicate with the access gateway to authenticate the user, in response to the request. For instance, the request can include an indication to the server to communicate with the access gateway to authenticate the user. In some embodiments, the server is pre-configured to communicate with the access gateway to authenticate the user, for requests to access a sanctioned network application. The server may send a request to the access gateway to authenticate the user. In response to the

server's request to authenticate the user, the access gateway can provide credentials of the user to the server 430.

In operation (3), the gateway service and/or the server can direct (or redirect) all traffic to a secure browser 420 which provides a secure browsing service. This may be in response to at least one of: a determination that the requested network application is a sanctioned network application, a determination that the request is initiated from a source other than a CEB, a determination that the requested network application is sanctioned, a determination that user is initiating the request via a standard browser, and/or a determination that the predefined URL and/or corresponding webpage is accessed.

The user's URL session can be redirected to the secure browser. For example, the server, gateway service and/or the access gateway can generate and/or send a URL redirect message to the standard browser, responsive to the determination. The secure browser plug-in of the standard browser can receive the URL redirect message, and can for example send a request to access the non-sanctioned network application, to the secure browser 420. The secure browser 420 can direct the request to the server of the non-sanctioned network application. The URL redirect message can instruct the standard browser (and/or the secure browser plug-in) to direct traffic (e.g., destined for the network application) from the standard browser to the secure browser 420 hosted on a network device. This can provide clientless access and control via dynamic routing though a secure browser service. In some embodiments, a redirection of all traffic to the secure browser 420 is initiated or configured, prior to performing authentication of the user (e.g., using SSO) with the server.

In some embodiments, the gateway service can direct or request the server of the requested network application to communicate with the secure browser 420. For example, the gateway service can direct the server and/or the secure browser to establish a secured connection between the server and the secure browser, for establishing an application session for the network application.

In some embodiments, the secured browser 420 comprises a browser that is hosted on a network device 432 of the cloud services 408. The secured browser 420 can include one or more features of the secured browser 420 described above in connection with at least FIG. 4 for instance. The hosted browser can include an embedded browser of a CEB that is hosted on the network device 432 instead of on the client device. The hosted browser can

include an embedded browser of a hosted virtualized version of the CEB that is hosted on the network device 432. Similar to the CEB installed on the client device, traffic is routed through the CEB hosted on the network device, which allows an administrator to have visibility of the traffic through the CEB and to remain in control for security policy control, analytics, and/or management of performance.

FIG. 6 illustrates an example implementation for browser redirection using a secure browser plug-in. In brief overview, the implementation includes a web browser 512 with a secure browser plug-in 516 operating on a client device, and a hosted web browser (or secure browser) 522 residing on a network device. The web browser 512 can correspond to a standard browser, instead of an embedded browser as discussed above in connection with FIG. 4 for example. The secure browser plug-in 516 can execute within a first network 510 and access a server 430 in a second network 530. The first network 510 and the second network 530 are for illustration purposes and may be replaced with fewer or additional computer networks. A secure browser plug-in 516 can be installed on the standard browser 512. The plug-in can include one or more components. One such component can include an ActiveX control or Java control or any other type and/or form of executable instructions capable of loading into and executing in the standard browser. For example, the standard browser can load and run an Active X control of the secure browser plug-in 516, in a memory space or context of the standard browser. In some embodiments, the secure browser plug-in can be installed as an extension on the standard browser, and a user can choose to enable or disable the plugin or extension. The secure browser plug-in can communicate and/or operate with the secured browser 420 for securing, using and/or accessing resources within the secured portion of the digital workspace.

By using the secure browser plug-in 516 operating within the standard browser 512 network applications accessed via the standard browser 512 can be redirected to a hosted secure browser. For instance, the secure browser plug-in 516 can be implemented and/or designed to detect that a network application is being accessed via the standard browser, and can direct/redirect traffic from the client device associated with the network application, to the hosted secure browser. The hosted secure browser can direct traffic received from the network application, to the secure browser plug-in 516 and/or a client agent 514 for rendering and/or display for example. The client agent 514 can execute within the web browser 512 and/or the secure browser plug-in, and can include certain elements or features of the client application 404 discussed above in connection with at least FIG. 4 for example.

For instance, the client agent 514 can include a remote session agent 416 for rendering the network application at the web browser 512. In some embodiments, the network application is rendered at the hosted secure browser, and the rendered data is conveyed or mirrored to the secure browser plug-in 516 and/or the client agent 514 for processing and/or display.

By way of an example, a user may be working remotely and may want to access a network application that is internal to a secure corporate network while the user is working on a computing device connected to an unsecure network. In this case, the user may be utilizing the standard browser 512 executing in the first network 510, in which the first network 510 may comprise an unsecure network. The server 430 that the user wants to access may be on the second network 530, in which the second network 530 comprises a secure corporate network for instance. The user might not be able to access the server 430 from the unsecure first network 510 by clicking on an internal uniform record locator (URL) for the secure website 532. That is, the user may need to utilize a different URL (e.g., an external URL) while executing the standard browser 512 from the external unsecure network 510. The external URL may be directed to or may address one or more hosted web browsers 522 configured to access server(s) 430 within the second network 530 (e.g., secure network). To maintain secure access, the secure browser plug-in 516 may redirect an internal URL to an external URL for a hosted secure browser.

The secure browser plug-in 516 may be able to implement network detection in order to identify whether or not to redirect internal URLs to external URLs. The standard browser 512 may receive a request comprising an internal URL for a website executing within the secure network. For example, the standard browser 512 may receive the request in response to a user entering a web address (e.g., for secure website 532) in the standard browser. The secure browser plug-in 516 may redirect the user web browser application 512 from the internal URL to an external URL for a hosted web browser application. For example, the secure browser plug-in 516 may replace the internal URL with an external URL for the hosted web browser application 522 executing within the secure network 530.

The secure browser plug-in 516 may allow the client agent 514 to be connected to the hosted web browser application 522. The client agent 514 may comprise a plug-in component, such as an ActiveX control or Java control or any other type and/or form of executable instructions capable of loading into and executing in the standard browser 512. For example, the client agent 514 may comprise an ActiveX control loaded and run by a

standard browser 512, such as in the memory space or context of the user web browser application 512. The client agent 514 may be pre-configured to present the content of the hosted web browser application 522 within the user web browser application 512.

The client agent 514 may connect to a server or the cloud/hosted web browser service 520 using a thin-client or remote-display protocol to present display output generated by the hosted web browser application 522 executing on the service 520. The thin-client or remote-display protocol can be any one of the following non-exhaustive list of protocols: the Independent Computing Architecture (ICA) protocol developed by Citrix Systems, Inc. of Ft. Lauderdale, Fla.; or the Remote Desktop Protocol (RDP) manufactured by the Microsoft Corporation of Redmond, Wash.

The hosted web browser application 522 may navigate to the requested network application in full-screen mode, and can render the requested network application. The client agent 514 may present the content or rendition of the network application on the web browser application 512 in a seamless and transparent manner such that it appears that the content is being displayed by the standard browser 512, e.g., based on the content being displayed in full screen mode. In other words, the user may be given the impression that the website content is displayed by the user web browser application 512 and not by the hosted web browser application 522. The client agent 514 may transmit navigation commands generated by the user web browser application 512 to the hosted web browser application 522 using the thin-client or remote-display protocol. Changes to the display output of the hosted web browser application 522, due to the navigation commands, may be reflected in the user web browser application 512 by the client agent 514, giving the impression to the user that the navigation commands were executed by the user web browser application 512.

Referring again to FIG. 5, and in operation (4), a new browser tab can open on the standard browser, to render or display the secure browser session. The new browser tab can be established or opened by the secure browser plug-in for instance. The secure browser plug-in and/or a client agent can receive data from the secure browser session, and can render the network application within the new browser tab as discussed above in connection with FIG. 6 for instance.

In operation (5), the secure browser can feed all user interaction events via the network application, back to analytics service for processing. The secure browser plug-in

can monitor for and intercept any user interaction events directed to the rendition of the network application within the browser tab. Hence, a user can use a native (or standard) browser to access a network application while allowing visibility into the network application's traffic, via the interoperation of cloud services and a secure browser (in the absence of the client application).

FIG. 7 depicts another example embodiment of a system of using a secure browser. In brief overview, the system includes cloud services 408, network applications 406 and the client device 402. In some embodiments, various elements of the system are similar to that described above for FIG. 5. A client application with embedded browser is not available in the client device 402. A standard or typical (e.g., HTML5) browser is available on the client device, from which a user can initiate a request to access a non-sanctioned network application. A network application can be specified as being sanctioned or non-sanctioned via policies that can be set by an administrator or automatically (e.g., via artificial intelligence).

In operation (1), the user may attempt to log into a non-sanctioned network application using the standard browser. The user may attempt to access a webpage of a server that provisions the network application, and to initiate a request to access the network application. In some embodiments, the request can be forwarded to or intercepted by a designated gateway service (e.g., in a data path of the request). For example, the gateway service (sometimes referred to as SWG) can reside on the client device (e.g., as an executable program), or can reside on a network device 432 of the cloud services 408 for instance. The gateway service can detect or otherwise determine if the requested network application is a sanctioned network application. The gateway service can determine if a CEB initiated the request. The gateway service can detect or otherwise determine that the request is initiated from a source (e.g., initiated by the standard browser) in the client device other than a CEB.

In operation (2), the gateway service detects that the requested network application is a non-sanctioned network application. The gateway service can for instance extract information from the request (e.g., destination address, name of the requested network application), and compare the information against that from a database of sanctioned and/or non-sanctioned network applications. The gateway service can determine, based on the comparison, that the requested network application is a non-sanctioned network application.

In operation (3), responsive to the determination, the gateway service can block access to the requested network application, e.g., by blocking the request. The gateway service can generate and/or send a URL redirect message to the standard browser, responsive to the determination. The URL redirect message can be similar to a URL redirect message sent from the server to the standard browser in FIG. 5 in operation (3). A secure browser plug-in of the standard browser can receive the URL redirect message, and can for example send a request to access the non-sanctioned network application, to the secure browser 420. The secure browser 420 can direct the request to the server of the non-sanctioned network application.

The server of the non-sanctioned network application may authenticate the user via the access gateway of the cloud services 408, e.g., responsive to receiving the request from the secure browser. The server may communicate with the access gateway to authenticate the user, in response to the request. The server may send a request to the access gateway to authenticate the user. In response to the server's request to authenticate the user, the access gateway can provide credentials of the user to the server 430. Upon authentication, the secure browser (or a corresponding CEB) can establish a secured connection and an application session with the server.

In operation (4), a new browser tab can open on the standard browser, to render or display the secure browser's application session. The new browser tab can be established or opened by the secure browser plug-in for instance. The secure browser plug-in and/or a client agent can receive data from the secure browser session, and can render the network application within the new browser tab as discussed above in connection with FIGs. 5-6 for instance.

In operation (5), the secure browser can feed all user interaction events via the network application, back to analytics service for processing. The secure browser plug-in can monitor for and intercept any user interaction events directed to the rendition of the network application within the browser tab. Hence, a user can use a native (or standard) browser to access a network application while allowing visibility into the network application's traffic, via the interoperation of cloud services and a secure browser (in the absence of the client application).

In some embodiments, in the absence or non-availability of a CEB on the client device, browser redirection is performed so that each requested network application is accessed via a corresponding hosted secure browser (or hosted CEB) for handling, instead of having all traffic redirected through a single hosted secure browser (or hosted CEB). Each dedicated secure browser can provide compartmentalization and improved security.

The use of a CEB, whether hosted or local to the client device, can allow for end-to-end visibility of application traffic for analytics, service level agreement (SLA), resource utilization, audit, and so on. In addition to such visibility, the CEB can be configured with policies for managing and controlling any of these as well as other aspects. For example, DLP features can be supported, to control "copy and paste" activities, download of files, sharing of files, and to implement watermarking for instance. As another example, the CEB can be configured with policies for managing and controlling access to local drives and/or device resources such as peripherals.

Referring now to FIG. 8, an example embodiment of a system for using local embedded browser(s) and hosted secured browser(s) is depicted. An environment is shown where different types of client devices 402A, 402B may be used (e.g., in a BYOD context), such that one may be locally equipped with a suitable CEB, and another client device may not have a suitable local CEB installed. In such an environment, systems described in FIG. 4, 5 and 7 can be used to support each of the client devices based on the availability of a locally installed and suitable CEB.

FIG. 9 depicts an example process flow for using local embedded browser(s) and hosted secured browser(s). The process flow can be used in the environment described above in FIG. 8, to determine whether an embedded browser or a hosted secured browser should be used for each client device to access a network application. For example, in operation 901, a HTTP client can attempt to access a web service (e.g., server of a network application). In operation 903, the web service can redirect the HTTP client to a gateway service for authentication. In operation 905, the gateway service can determine if the HTTP client is a CEB. If so, in operation 909, the gateway service can determine if the CEB is a suitable CEB, e.g., capable of enforcing defined application policies. If so, in operation 911, the CEB is allowed access to the web service, and can enforce the defined policies.

If the gateway service determines that the HTTP client is not a CEB, the gateway service can cause a virtualized version of a CEB to be initialized and hosted on a remote server (e.g., a network device 432 of cloud services 408), in operation 907. In some embodiments, such a hosted CEB may already be available on a network device 432, and can be selected for use. For example in operation 911, the CEB is allowed access to the web service, and can enforce the defined policies.

If the gateway service determines that the HTTP client is a CEB, but that the CEB is not a suitable CEB, the gateway service can cause a virtualized version of a CEB to be initialized and hosted on a remote server (e.g., a network device 432 of cloud services 408), in operation 907. In some embodiments, such a hosted CEB may already be available on a network device 432, and can be selected for use. For example in operation 911, the CEB is allowed access to the web service, and can enforce the defined policies.

In some embodiments, if the user is requesting access to a web application located in a company data center, the gateway service (in cloud service or on premise) can allow access when the client application with CEB is detected. Otherwise, the request can be routed to a service with the hosted virtualized version of the CEB, and then access is authenticated and granted.

At operation 905 and/or operation 909 for instance, the decisions made on whether the HTTP client is a CEB and whether it is a suitable CEB may be determined by a number of factors. For example, to determine if the HTTP client is CEB, the gateway service may take into account factors, for example including at least one of: user Identity and strength of authentication, client Location, client IP Address, how trusted the user identity, client location, client IP are, jailbreak status of the client device, status of anti-malware software, compliance to corporate policy of the client device, and/or remote attestation or other evidence of integrity of the client software.

To determine if the CEB is able to honor or support all defined application policies (which may vary by client version, client OS platform and other factors), the client device's software and gateway service may perform capability negotiation and/or exchange version information. In some embodiments, the gateway service can query or check a version number or identifier of the CEB to determine if the CEB is a suitable CEB to use.

Driving all the traffic though the CEB then allows additional control of content accessing SaaS and Web based systems. Data Loss Prevention (DLP) of SaaS and Web traffic can be applied through the CEB app with features including copy and paste control to other CEB access applications or IT managed devices. DLP can also be enforced by enabling content to be downloaded only to designated file servers or services under IT control.

Referring now to FIG. 10, depicted is an example embodiment of a system for managing user access to webpages. Some webpages (or websites) are known to be safe while others may be suspect. A user may access a webpage via a corresponding URL through a standard browser. For example, the user may click on a link corresponding to the URL, which may be included in an email being viewed using a mail application. An access gateway (SWG) may intercept an access request generated by the clicking of the link, and can determine if the corresponding URL is safe or suspect. If the URL is known to be safe, the access gateway can allow the request to proceed to the corresponding website or web server. If the URL is suspect, the access gateway can redirect the request to be handled via a hosted secure browser. The secure browser can request access for, and access the webpage (on behalf of the standard browser), and can allow the webpage information to be conveyed to the standard browser, similar to the handling of a network application via browser redirection as discussed in connection with at least FIGs. 7 and 5.

C. Systems and Methods for Executing in an Embedded Browser an Application Script for Network Applications of Different Origins

The present disclosure is directed towards systems and methods for executing in an embedded browser an application script for network applications of different origins. A client application can execute on a client device via an embedded browser. The client application can establish or provide one or more sessions to one or more network applications via the embedded browser. The client application can establish policies and execute scripts to control operation and functionality of the embedded browser. For example, the network applications can be provided to a user of the client device within the embedded browser of the client application. The client application can establish policies and execute scripts, via a scripting engine, to control operation of the different network applications executing within the embedded browser.

The client application can generate and apply policies unique to network applications executing within the embedded browser than override, prevent or otherwise modify the policies of the entities and/or origins that the network applications originate from. In some embodiments, the client application can generate and apply policies to override same origin policies of one or more network applications executing within the embedded browser of the client application. The client application, through the embedded browser, can provide cross application collaboration between different network applications executing within the embedded browser of the client application such that network applications can leverage functionality of other network applications executing within the embedded browser even if the other network applications are of different entities and/or different origins. For example, a scripting engine of the client application can execute application scripts to perform tasks using network applications of different entities and/or of different origins. In some embodiments, a first one or more network applications of a first entity and/or a first origin can collaborate or otherwise share, transfer and/or receive data from a second one or more network applications of a second, different entity and/or a second, different origin to complete the corresponding task.

The client application can establish a domain of trust between a client device and one or more network applications from one or more different entities and/or different origins. The domain of trust can form a platform for a user of the client device to perform tasks, such as but not limited to, transferring data across network applications from different entities and/or different origins. The domain of trust can execute within the embedded browser of the client application. The domain of trust can provide a trusted environment for network applications of trusted origins to share data between each other and/or perform various tasks for a user of the client device using the functionality of multiple different network applications. The client application can include or otherwise provide a scripting engine that generates one or more application scripts. The application scripts can include instructions to perform the one or more tasks across network applications from different entities and/or different origins.

Network applications coupled with or executing within the embedded browser of the client application can leverage the domain of trust for cross-application collaboration. For example, network application can include or be provided from a suite of applications that originate from or are hosted by different servers at different origins. The suits can include same-origin security policies that limit or prevent collaboration or integration between

network applications from different suits, different servers and/or different origins. The systems and methods as described herein can establish a domain of trust that forms a platform for collaboration or integration between network applications from different suits, different servers and/or different origins. For example, a user of a client device can perform one or more tasks across the different network applications within the domain of trust using application scripts generated by a scripting engine of the client application.

The application scripts can work across multi-vendor network applications to perform the various tasks for a user of the client device. For example, the application scripts can be generated such that they meet the respective security policies of the network applications from different suits, different servers, different entities and/or different origins. Thus, the client application can operate as a scripting host for the client device to perform scripting, via a scripting engine and within the domain of trust, across the different applications from different suits, different servers, different entities and/or different origins. For example, the scripting engine of the client application can write an application script and/or automations tasks, such as but not limited to, pulling data from a first network application of a first entity at a first origin and transfer the data to a second network application of a second entity at a second origin.

Referring to FIG. 11, depicted is a block diagram of a system 1100 for collaborating across network applications of different origins in an embedded browser. The system 1100 can include one embodiment of a client application 1104 executing on a client device 1102. The client application 1104 can establish a domain of trust 1108 between one or more network applications 1134 of one or more different entities 1132 at one or more different origins 1130. For example, the client application 1104 can include a scripting engine 1110 to execute one or more application scripts 1112 across one or more network applications 1134 of one or more different entities 1132 at one or more different origins 1130 for a user of the client device 1102.

The client application 1104 includes an embedded browser 1106, the domain of trust 1108, a scripting engine 1110, a policy manager 1120, and an authentication server 1122. The client application 1104 can be hosted by or originate at an origin 1130c (e.g., third origin) and provided by an entity 1132c (e.g., third entity). The client application 1104 can establish one or more sessions 1140a-1140n to one or more network applications 1134a-1134n to perform one or more tasks 1116 across the network applications 1134a-1134n

using the application scripts 1112 of the scripting engine 1110. The client application 1104 may be an instance of any client application described herein. For example, the client application 1104 may be the same as or substantially similar to client application 404 of FIG. 4.

The client application 1104 can include the embedded browser 1106 integrated into the client application 1104 to access one or more network applications 1134 of the plurality of network applications 1134a-1134n. The client application 1104 with the embedded browser 1106 (CEB) can include any element of a CEB as described herein. For example, the embedded browser 1106 may be the same as or substantially similar to embedded browser 410 described above with respect to FIG. 4. The embedded browser 1106 can include elements and functionalities of a web browser application or engine. The embedded browser 1106 can locally render one or more of networks application 1134a-1134n as a component or extension of the client application 1104. For example, the embedded browser 1106 can render a SaaS/Web application inside the CEB which can provide the CEB with full visibility and control of an application session.

The client application 1104 can establish one or more of sessions 1140a-1140n to one or more of network applications 1134a-1134n for the client device 1102 through the embedded browser 1106. The sessions 1140a-1140n can include any type or form of a session as described herein. For example, sessions 1140a-1140n may include, but not limited to, an application session, an execution session, a desktop sessions, a hosted desktop session, a terminal services session, a browser session, a remote desktop session, and a remote application session. Sessions 1140a-1140n may include encrypted and/or secure sessions established between a network application 1134a-1134n and the client device 1102.

The network applications 1134a-1134n may include any type or form of a network application detailed herein. For example, the network applications 1134a-1134n may be the same as or substantially similar to network application 406 described above with respect to FIG. 4. The network applications 1134a-1134n may include applications (apps) that are served from and/or hosted on one or more servers (e.g., third part servers). The network applications 1134a-1134n can include an application hosted on at least one server accessed by the client device 1102 via a network 104. The network applications 1134a-1134n can include, but not limited to, a web application, a desktop application, remote-hosted application, a virtual application, a software as a service (SaaS) application, a mobile

application, an HDX application, a local application, a native application (e.g., native to the client device), and/or a device couple with the client device 1102. The network applications 1134a-1134n can include or execute one or more web pages. For example, each of the network applications 1134a-1134n can include a plurality of web pages. The web pages can include a web site, and/or a web server associated with at least one network application 1134. The web pages can provide content corresponding to at least network application 1134.

Each of the network applications 1134a-1134n can originate at least one origin 1130. The origins 1130a-1130n can refer to an origin of a web page or network application 1134. The origins 1130a-1130n include a domain name or combination of URI scheme, host name, and port number. For example, the origin 1130 can refer to a combination of a protocol, a host, and a port for a URL. In some embodiments, two or more URLs having the same origin 1130 can have the same protocol, host, and port. For example, two or more web pages or network applications 1134 can have the same origin 1130 if the protocol, the host, and the port for their respective URLs are the same.

Each of the network applications 1134a-1134n can be provided by at least one entity 1132. The entities 1132a-1132n can include a vendor or provider of one or more network applications 1134a-1134n. For example, the entities 1132a-1132n can generate and provide one or more network applications 1134a-1134n. The entities 1132a-1132n can include third party vendors executing on one or more third party servers 1136. The entities 1132a-1132n can group or otherwise provide multiple network applications 1134a-1134n together in one or more suites 1138 (or application suite). The suites 1138 can include multiple network applications 1134a-1134n that are combined and provided together to one or more client devices 1102. For example, a suite 1138 can include two or more network applications 1134a-1134n (e.g., software programs) delivered within a single executable and/or installable file. The suites 1138 can include multiple network applications 1134a-1134n from the same or common entity 1132. The suites 1138 can include multiple network applications 1134a-1134n from the same or common origin 1130.

The network applications 1134a-1134n can be hosted by servers 1136a-1136n (e.g., third party servers), respectively. The network applications 1134a-1134n can include an application hosted on at least one server 1136 accessed by the client device 1102 via a network 104. The servers 1136a-1136n can be separate from a server hosting the client

application 1104. Each of the origins 1130a-1130n can include or correspond to at least one server 1136. In some embodiments, an origin 1130 can include or correspond to multiple servers 1136. For example, a suite 1137 of applications can be originate from or hosted by a server 1138 of an origin 1130. For example, a first one or more network applications 1134a-1134n can include a first suite 1138a of applications originating from or hosted by a first one or more servers 1136a at a first origin 1130a. A second one or more network applications 1134a-1134n can include a second suite 1138b of applications originating from or hosted by a second one or more servers 1136b at a second origin 1130b. The servers 1136a-1136n can be same as or substantially similar to server 430 of FIG. 4.

The client application 1104 can establish, provide or include a domain of trust 1108. The domain of trust 1108 can include a platform or one or more connections between different network applications 1134a-1134n. The connections can include one or more trust relationships established between different network applications 1134a-1134n. In some embodiments, the domain of trust 1108 can include or be formed as a secure platform or connection between multiple network applications 1134a-1134n. The domain of trust 1108 connect two or more network applications 1134a-1134n through a secure platform such that the network applications 1134a-1134n of different entities 1132a-1132n and/or of different origins 1130a-1130n can share, exchange, transfer, and/or receive data or information from each other. In some embodiments, the domain of trust 1108 and the trust relationships between the plurality of applications 1134a-1134n can be encrypted using encryption techniques as described herein. The domain of trust 1108 can couple with a network 104 or multiple networks 104 (e.g., couple multiple networks 104) to allow communication, information exchange, and collaboration between different network applications 1134a-1134n.

The client application 1104 can establish a single domain of trust 1108 or multiple domains of trust 1108. For example, the client application 1104 can establish a single domain of trust 1108 that includes each network application 1134 that a client device accesses or requests access to. The client application 1104 can establish multiple domains of trust 1108 with each domain of trust 1108 having two or more network applications 1134a-1134n. The network applications 1134a-1134n can be grouped within common domains of trust 1108 based in part on their respective origin and/or entity. In some embodiments, the network applications 1134a-1134n can be grouped within common domains of trust 1108 based in part functions they perform, tasks 1116 they perform or can

execute, a type of the respective network application, and/or properties of the respective network application. The client application 1104 can establish multiple domains of trust 1108 for different suites 1138a-1138n of network applications 1134a-1134n.

Client application 1104 can include a scripting engine 1110. The scripting engine 1110 can generate and execute scripts in one or more different scripting languages to allow a user of the client device 1102 to interact within different network applications 1134 of different entities 1132a-1132n and/or different origins 1130a-1130n within the embedded browser 1106 of the client application 1104. For example, the scripting engine 1110 can generate and execute application scripts 1112 for network applications 1134 from different entities 1132a-1132n and/or different origins 1130a-1130b. Thus, the client application 1104 can perform tasks 1116 for a user of the client device 1102 using network applications 1134 from different entities 1132a-1132n and/or different origins 1130a-1130b for collaboration across the respective network applications 1134 of different entities 1132a-1132n and/or different origins 1130a-1130b in the embedded browser 1106. The scripting engine 1110 can include one or more processors to execute one or more applications scripts. The scripting engine 1110 can include one or more software programs to execute one or more applications scripts.

The application scripts 1112 can include a computer language or programming language that supports and/or executes on one or more network applications 1134a-1134n. The application scripts 1112 can include one or more instructions 1114 or series of commands within a file that can be executed. The instructions 1114 can include a series of commands or code to execute one or more operations of a processor. The application scripts 1112 can support and execute on network applications 1134a-1134n of different entities 1132a-1132n and/or different origins 1130a-1130b. For example, an application script 1112 can support and/or execute one or more tasks 1116 on, within or otherwise using a first network application 1134 of a first entity 1132 and/or a first origin 1130 and support and/or execute one or more tasks 1116 on, within or otherwise using a second, different network application 1134 of a second, different entity 1132 and/or a second, different origin 1130.

The tasks 1116 can include an action to be performed or executed on or via the client device 1102. The tasks 1116 can include an action to be performed or executed on or via the client device 1102 using at least one network application 1134. The tasks 1116 can include a unit of programming, a unit of execution, a unit of work or process performed or

executed on or via at least one client device 1102 and/or network application 1134. For example, a task 1116 can include, but not limited to, sending an email, drafting a word document, transferring data from a first network application 1134 to a second, different network application 1134, or any action that includes interacting with any network application 1134, program, file, and/or device coupled with or executing on the client device 1102, for example, via the client application 1104 and embedded browser 1106. The tasks 1116 may include, but not limited to, printing functionality, storing data or files in particular locations (e.g., on the client device, on a network server), download functionality, upload functionality, camera functionality, viewing device functionality, taking a picture functionality, obtaining a signature functionality, and/or digital signature functionality. The tasks 1116 as provided here are listed for explanatory purposes and not intended to be limiting in anyway.

The client application can include a policy manager 1120. The policy manager 1120 can generate one or more policies for authenticating a user of a client device 1102 and/or a client device 1102. The policy manager 1120 generate one or more policies for establishing trust levels for one or more network applications 1134a-1134n, one or more origins 1130a-1130n, and/or one or more entities 1132a-1132n. The policy manager 1120 generate one or more policies for establishing trust relationships between one or more network applications 1134a-1134n, one or more origins 1130a-1130n, and/or one or more entities 1132a-1132n. For example, the trust levels can correspond to access privileges such as, but not limited to, access to different devices or systems of the client device, security privileges, functionalities of the respective network application within the embedded browser and/or tasks the respective network application or device can perform within the embedded browser. In some embodiments, responsive to a being assigned a particular trust level, a network application can be allowed to, but not limited to, transfer data to a second network application of a different entity and/or a different origin within the embedded browser of the client application. The policy manager 1120 can determine which one of the one or more network applications 1134a-1134n, one or more origins 1130a-1130n, and/or one or more entities 1132a-1132n can be trusted and included within or coupled with one or more domains of trust 1108.

In some embodiments, the policy manager 1120 to apply or otherwise provide policies for managing the sessions 1140a-1140n between the network applications 1134a-1134n and client device 1102. The policy manager 1120 can apply or otherwise provide

policies for managing or establishing the sessions 1140a-1140n between the network applications 1134a-1134n and the client device 1102. For example, the policy manager 1120 can establish a trust level for sessions 1140a-1140n between the network applications 1134a-1134n and client device 1102. The policy manager 1120 can generate and/or store a plurality of policies for the sessions 1140a-1140n. The policies can be used to control communication and information exchange between the network applications 1134a-1134n and the client device 1102. For example, the policies can include access policies and/or security policies. The client application 1104, through the policy manager 1120 can apply one or more policies to a session 1140 to secure the respective session 1140 between a network application 1134 and the client device 1102. The policies can be used to control traffic levels and/or traffic types through the different sessions 1140a-1140n. The policies can be used to manage a service level of one or more sessions 1140a-1140n. The policy manager 1120 can apply policies to encrypt or otherwise secure one or more sessions 1140a-1140n. Policies can be assigned each of the network applications 1134a-1134n, the client device 1102, and/or a user of the client device 1102 or network applications 1134a-1134n. The policies can be used to control and/or enhance usage of one or more sessions 1140a-1140n established between the network applications 1134a-1134n and the client device 1102.

The client application can include an authentication manager 1122. The authentication manager 1122 can authenticate one or more users of the client device 1102, one or more client devices 1102, one or more network applications 1134a-1134n, one or more origins 1130a-1130n, and/or one or more entities 1132a-1132n. For example, the authentication can authenticate receive authentication credentials and compare the received authentication credentials to stored authentication credentials to authenticate one or more users of the client device 1102, one or more client devices 1102, one or more network applications 1134a-1134n, one or more origins 1130a-1130n, and/or one or more entities 1132a-1132n. The authentication manager 1122 can store authentication credentials for one or more users of the client device 1102, one or more client devices 1102, one or more network applications 1134a-1134n, one or more origins 1130a-1130n, and/or one or more entities 1132a-1132n. In some embodiments, the authentication manager 1122 can couple with an authentication server executing on a third party server 1136 to retrieve, store, and/or receive authentication credentials for one or more users of the client device 1102, one or

more client devices 1102, one or more network applications 1134a-1134n, one or more origins 1130a-1130n, and/or one or more entities 1132a-1132n.

The client device 1102 may be an instance of any client device described herein. For example, the client device 1102 may be the same as or substantially similar to client device 202, 204 described above with respect to FIG. 2, client device 302 described above with respect to FIG. 3, and/or client device 402a described above with respect to FIG. 4. The client device 1102 can include, but not limited to, computing devices, desktop computing devices, and/or mobile computing devices. The client device 1102 can include or store context 1112 for one or more users of the client device 1102. The context of a user 1112 can include properties or characteristics of the user and/or the client device 1102. For example, the context of a user 1112 can include, but not limited to: an identify of the user, a user profile, a home location of the user, a location of one or more client devices 1102 associated with the user, information about the client device 1102 executing the client application 1104, information about one more network applications 1140a-1140n, and/or information about an entity associated with the user.

Network 104 may be a public network, such as a wide area network (WAN) or the Internet. In some embodiments, network 104 may be a private network such as a local area network (LAN) or a company Intranet. Network 104 may employ one or more types of physical networks and/or network topologies, such as wired and/or wireless networks, and may employ one or more communication transport protocols, such as transmission control protocol (TCP), internet protocol (IP), user datagram protocol (UDP) or other similar protocols.

Each of the above-mentioned elements or entities is implemented in hardware, or a combination of hardware and software, in one or more embodiments. Each component of the client application 1104 may be implemented using hardware or a combination of hardware or software detailed above in connection with FIG. 1. For instance, each of these elements or entities can include any application, program, library, script, task, service, process or any type and form of executable instructions executing on hardware of a client device 1102 (e.g., the client applications 404). The hardware includes circuitry such as one or more processors in one or more embodiments.

Referring to FIG. 12 depicts a flow diagram of one embodiment of a method 1200 for executing in an embedded browser an application script for network applications of different origins. The functionalities of the method may be implemented using, or performed by, the components detailed herein in connection with FIGs. 1–11. In brief overview, a first session can be established to a first network application by a client application (1205). A second session can be established to a second network application by the client application (1210). An application script can be identified for one or more network applications (1215). The application script can be executed (1220).

Referring now to operation (1205), and in some embodiments, a client application can establish a first session to a first network application. The client application can establish a first one or more sessions with a first one or more network applications of a first entity at a first origin via an embedded browser within the client application. The client application can be executing on a client device through an embedded browser. The client device can be associated with the user. The client application can include the embedded browser to provide access to one or more network applications for a user of a client device. The client application can establish for a user of a client device one or more sessions with one or more network applications via the embedded browser within the client application. The client application can be executing on a client device through an embedded browser. The client device can be associated with the user. The sessions can include any type or form of a session as described herein. For example, sessions may include, but not limited to, an application session, an execution session, a desktop sessions, a hosted desktop session, a terminal services session, a browser session, a remote desktop session, and a remote application session. Sessions may include encrypted and/or secure sessions established between a network application and a client device of the user.

The network applications can include an application hosted on a server (e.g., third party server) accessed by the client device via a network. The network applications can include, but not limited to, a web application, a desktop application, a virtual application, a software as a service (SaaS) application, a mobile application, an HDX application, a local application, a native application (e.g., native to the client device), and/or a device couple with the client device. The network applications may include applications (apps) that are served from and/or hosted on one or more servers. For example, each of the network applications can include one or more locations or one or more points of presence that can provide content associated with the respective network application. In some embodiments,

the network applications can include an application hosted on a server accessed by a client device of the user via one or more networks. The network applications can include or execute one or more web pages. The client application can provide access to local files or native files, for example, such as files executing on the client device.

Referring now to operation (1210), and in some embodiments, the client application can establish a second session to a second network application. For example, the client application can establish a second one or more sessions with a second one or more network applications of a second entity at a second origin via the embedded browser within the client application. The network applications can be provided from one or more different entities and/or one or more different origins. For example, each of the network applications can be hosted by or otherwise provided by at least one entity. The entities can include a vendor or provider of the respective network application or a webpage corresponding to the network application. Each of the entities can include one or more servers that host one or more network applications at one or more different origins. The origins can refer to an origin of the respective network application or a web page corresponding to the network application 1134. The origin can include a domain name or combination of URI scheme, host name, and port number. For example, the origin can refer to a combination of a protocol, a host, and a port for a URL.

The entities and/or origins can assign polices to each of their respective network applications to control access and/or operation of the respective network applications. The policies can include access policies and/or origin policies (e.g., same-origin policies). For example, the policies (e.g., same origin policies) can indicate what other network applications a particular network application can interact with or collaborate with, such as but not limited to, for sharing and/or transferring data between the respective network applications. In some embodiments, network applications from the same origin can share one or more policies (e.g., access policies, origin policies). The policies can indicate a trust level of an origin of the respective network application.

In some embodiments, a first origin and a second origin can be different origins that fail a same origin policy. For example, a first origin can refer to a different origin than a second origin. Each of the respective origins can have origin policies, such as but not limited to, same origin policies. Under a same origin policy, a first network application can be limited to collaborating with, accessing data from and/or transferring data to a second,

different network application of the same origin as the first network application. For example, a first network application of a first origin can collaborate with, access data from and/or transfer data to a second, different network application of the first origin, but the first network application of the first origin can be prevented, blocked or restricted from collaborating with, accessing data from and/or transferring data to a third, different network application of a second origin. Thus, different origins can fail same origin policies as they originate from different origins. Origins that are the same can pass same origin policies. In some embodiments, network applications from the same entity can share one or more policies (e.g., access policies, origin policies). Thus, network applications from the same origin can pass a same origin policy. Network applications from different origins can fail a same origin policy. In some embodiments, network applications from the same entity can have one or more different policies (e.g., access policies, origin policies).

The network applications can be grouped into suites (e.g., application suites) having multiple network applications. The entity providing the respective network applications or the origin and/or server hosting the respective network applications can group multiple network applications into a suite of applications. For example, the first one or more network applications can include a suite of applications originating from or hosted by a first one or more servers at the first origin. The second one or more network applications can include a suite of applications originating from or hosted by a second one or more servers at the second origin. The suites can include different but related network applications. For example, a suite can include network applications of the same origin and/or of the same entity. Each of the network applications in a common or same suite can perform one or more common functions tasks and/or one or more different functions or tasks. In some embodiments, a suite can include two or more network applications that are provided in a single executable format or file.

In some embodiments, a domain of trust can be established between different network applications. For example, a domain of trust can be established between different network applications executing within the embedded browser of the client application. The client application can establish a domain of trust between the first one or more networks applications of the first entity at the first domain and the second one or more network applications of the second entity at the second domain. The domain of trust can include and/or provide a common platform to share information between different network applications. For example, the client application can establish trust relationships (e.g.,

sessions, connections) between multiple network applications. For example, a first one or more network applications having a trust relationship with one or more second, different network applications can be included within a common domain of trust. Network applications within a common or same domain of trust can have one or more trust relationships established between the respective network applications. For example, network applications within a common or same domain of trust can collaborate, share, access, receive and/or transfer data between each other responsive to one or more application scripts. The domain of trust can provide a secure environment or secure connection to allow communication and information exchange between the diverse applications.

In some embodiments, the domain of trust can include or be formed as a secure platform or connection between multiple different network applications. For example, the domain of trust connect two or more network applications through a secure platform such that the network applications of different entities and/or of different origins can share, exchange, transfer, and/or receive data or information from each other. In some embodiments, the domain of trust and the trust relationships between the plurality of applications can be encrypted using encryption techniques as described herein. The domain of trust can allow communication, information exchange, and collaboration between different network applications.

The client application can establish one or more policies for the domain of trust. The policies can include, but not limited to, access policies, grouping policies, and/or origin polices. The policies can indicate what network applications can be grouped into common domains of trust with each other for a user of a client device and/or a client device. The policies can indicate what properties a particular first one or more network application may need to be grouped into a common domain of trust with a second, one or more different network applications. For example, the network applications can be provided to a user of the client device within the embedded browser of the client application. The client application can generate and assign one or more policies for a first one or more network applications executing within the embedded browser of the client application to control access, such as but not limited to, access by a second, different one or more network applications through the domain of trust. In some embodiments, the policies can be used to override same origin policies of the origins and/or servers that host the respective network applications. Thus, the policies and scripts used and applied to each of the network

applications can be generated and/or controlled by the client application through the embedded browser. Therefore, the client application, through the embedded browser, can provide cross application collaboration between different network applications executing within the embedded browser of the client application. For example, a scripting engine of the client application can execute application scripts to perform tasks using network applications of different entities and/or of different origins. For example, a first one or more network applications of a first entity and/or a first origin can collaborate or otherwise share, transfer and/or receive data from a second one or more network applications of a second, different entity and/or a second, different origin to complete the corresponding task.

The client application can establish policies to control the level of access a user of the client device can have with one or more network applications. For example, the client application can assign different policies and/or different levels of access for a user to one or more different network applications based in part on the credentials of the user and responsive to an authentication of the user. The levels of access can indicate what network applications the user of the client device can interact with. The levels of access can indicate how much access a user is allowed to one or more network applications. For example, the levels of access can indicate what functions of a network application a user can utilize and/or what one or more tasks a user can perform using a network application. The levels of access can indicate what other users and/or other client devices a particular user and/or a particular client device can interact with. applications the user of the client device can interact with. The levels of access can indicate a level of communication between different network applications.

In some embodiments, the client application can establish one or more domains of trust for a user of a client device. The client application can group or establish trust relationships between network applications based in part on network applications that a particular user has accessed previously and/or requested to access. The client application can group or establish trust relationships between network applications based in part on one or more tasks the particular user has performed via the client device previously and network applications corresponding to the tasks and/or network applications that can perform or execute the corresponding tasks.

In some embodiments, the client application can establish one or more domains of trust for a client device. The client application can group or establish trust relationships

between network applications based in part on network applications that a particular client device has interacted or accessed previously and/or requested to access. The client application can group or establish trust relationships between network applications based in part on one or more tasks the particular client device has performed or executed previously and network applications corresponding to the tasks and/or network applications that can perform or execute the corresponding tasks.

In some embodiments, the client application can establish one or more domains of trust for a group of users (e.g., office pool, family household). The client application can group or establish trust relationships between network applications based in part on network applications that a particular group of users have accessed previously and/or requested to access. The client application can group or establish trust relationships between network applications based in part on one or more tasks the particular group of users has performed via one or more client devices or computing devices previously and network applications corresponding to the tasks and/or network applications that can perform or execute the corresponding tasks.

In some embodiments, the client application can establish one or more domains of trust for a plurality of client devices or computing devices (e.g., office pool, family household). The client application can group or establish trust relationships between network applications based in part on network applications that a particular plurality of client devices or computing devices have interacted or accessed previously and/or requested to access. The client application can group or establish trust relationships between network applications based in part on one or more tasks the particular plurality of client devices or computing devices have performed or executed previously and network applications corresponding to the tasks and/or network applications that can perform or execute the corresponding tasks.

In some embodiments, a domain of trust can be established responsive to authentication of the same user for the first one or more network applications and the second one or more network applications via the client application on the client device of a third entity of the user corresponding to a third origin. For example, the client application can authenticate a user of the client device. The authentication can include authenticating the user for one or more network applications. The client application can establish one or more sessions with one or more network applications responsive to authentication of the user. For

example, the client application can receive authentication credentials for a user and identify which network applications the user has access or privileges to interact with. The client application can establish sessions with the network applications the user has access or privileges to interact with. The client application can establish one or more domains of trust for the network applications the user has access or privileges to interact with. The network applications can include multiple network applications of different entities and/or different origins. The client application can establish trust relationships between the network applications of different entities and/or different origins using the domain of trust. The client application can establish access policies for the domain of trust. For example, in some embodiments, the access policy for the access to the domain of trust to the user of the client device.

In some embodiments, the authentication can include providing the user a level of access to one or more network applications. The level of access can correspond and/or indicate which tasks, functions or operations the user can perform using the respective network application. For example, the user of the client device can be provided or assigned a first level of access to a first network application and a second level of access to a second network application. The first and second level of access can be different. The user of the client device can have the same level of access to each network application grouped in a common domain of trust. The user of the client device can have one or more different levels of access to one or more different network applications grouped in a common domain of trust.

Referring now to operation (1215), and in some embodiments, an application script can be identified for one or more network applications. For example, a scripting engine within the client application of a client device of a user at a third origin can identify an application script. The client application and/or the client device can be at or hosted by a third origin, different from the first origin and the second origin. For example, the third origin can be hosted a by a server different from the one or more servers hosting the first origin and/or the one or more servers hosting the second origin. The client application and/or the client device can be provided by a third entity corresponding to the third origin. The third entity can be different from the first entity and/or the second entity. The application script can include instructions to interact with each the first one or more networks applications of the first entity at the first origin and the second one or more network applications of the second entity at the second origin.

The scripting engine can be executing on the client device within the embedded browser of the client application. The scripting engine can generate one or more application scripts to interact with one or more network applications the client application establishes a session with for a user of the client device. In some embodiments, the scripting engine can store one or more applications scripts, for example, but not limited to, within a database of the client application. The application scripts can correspond to one or more tasks, operations or functions. The application scripts can correspond to one or more network applications. The application scripts can correspond to one or more tasks, operations or functions to be executed using one or more network applications. In some embodiments, the tasks, operations, and/or functions can be executed, responsive to at least one application script, using two or more network applications of different entities and/or different origins. The scripting engine can generate and execute scripts in one or more different scripting languages to allow a user of the client device to interact within different network applications of different entities and/or different origins within the embedded browser of the client application. The client application can perform tasks for a user of the client device using the application scripts to interact with and across different network applications from different entities and/or different origins.

The application scripts can include a computer language or programming language that supports and/or executes on one or more network applications. In some embodiments, the scripting engine can retrieve the application scripts from one or more third party servers. For example, the scripting engine can retrieve the application scripts from one or more third party servers hosting or providing one or more network applications. The retrieved application scripts can correspond to the network applications the respective server hosts or provides. The application scripts can include one or more instructions or series of commands within a file that can be executed, for example, by one or more network applications within the embedded browser of the client application. The application scripts can execute on or across network applications of different entities and/or different origins. For example, an application script can support and/or execute one or more tasks on, within or otherwise using a first network application of a first entity and/or a first origin and support and/or execute one or more tasks n, within or otherwise using a second, different network application of a second, different entity and/or a second, different origin. The application scripts can be executed to override same origin policies of one or more network applications such that network applications of different entities and/or different origins can

collaborate with each other to perform various tasks for a user of the client device within the embedded browser of the client application.

The scripting engine can generate or identify an application script responsive to a request to perform a task. For example, the client application can receive a request to perform a task within the embedded browser. The client application can transmit the task request to the scripting engine. The scripting engine can identify a network application or multiple network applications needed to perform the requested task. The scripting engine can generate or identify one or more application scripts corresponding to the network application or multiple network applications needed to perform the requested task. The client application can generate or identify an application script responsive to a request to access a network application. For example, the client application can receive a request for a network application within the embedded browser. The client application can transmit the network application request to the scripting engine. The scripting engine can generate or identify one or more application scripts corresponding to the network application or multiple network applications needed to perform the requested task.

Referring now to operation (1220), and in some embodiments, the application script can be executed. For example, the scripting engine can execute the instructions of the application script to perform a task across the first one or more networks applications of the first entity at the first origin and the second one or more network applications of the second entity at the second origin. The application script can include one can include one or more instructions or series of commands within a file that can be executed. The instructions can include a series of commands or code to be executed, for example, by the scripting engine of the client application. The instructions can include a series of commands or code to be executed to perform one or more tasks. The instructions can include or identify the task to be performed, one or more network applications to perform the respective task, the entity of the respective one or more network applications, the origin of the respective one or more network applications, and policies corresponding to the respective network applications, entities, and/or origins. A task can include, but not limited to, an action to be performed or executed on or via the client device. For example, a task can include an action to be performed or executed on or via the client device using at least one network application. The tasks can include a unit of programming, a unit of execution, a unit of work or process performed or executed on or via at least one client device and/or network application. The tasks may include, but not limited to, printing functionality, storing data or files in particular

locations (e.g., on the client device, on a network server), download functionality, upload functionality, camera functionality, viewing device functionality, taking a picture functionality, obtaining a signature functionality, and/or digital signature functionality. For example, a task can include, but not limited to, transferring data from a first network application of a first entity of a first origin to a second, different network application of a second, different entity of a second, different origin, via the client application and embedded browser.

In some embodiments, the scripting engine can initiate execution of the application script by one of the first one or more network applications or the second one or more network applications. One or more network applications can be executing within the embedded browser of the client application. The scripting engine can provide or other apply the application script to the network applications identified to perform a requested task for a user of the client device. For example, a first one or more network applications of a first entity and a first origin and a second one or more network applications of a second entity of a second origin can be executing within the embedded browser of the client application. The scripting engine can provide or apply the application script to the first one or more network applications and the second one or more network applications. The scripting engine can initiate execution of one or more applications scripts with the first one or more network applications and/or the second one or more network applications. The order of execution of the one or more application scripts can be included within the instructions of the respective application scripts and/or based in part on the task to be performed. In some embodiments, the scripting engine can initiate execution of one or more applications scripts with the first one or more network applications and initiate execution of one or more applications scripts with the second one or more network applications simultaneously. The scripting engine can initiate execution of one or more applications scripts with the first one or more network applications before initiating execution of one or more applications scripts with the second one or more network applications. The scripting engine can initiate execution of one or more applications scripts with the first one or more network applications after initiating execution of one or more applications scripts with the second one or more network applications. The application script can cause the respective one or more network applications to perform at least one task.

In some embodiments, a requested task can include a plurality of tasks or different set of tasks. The scripting engine can provide the application script to the first one or more

network applications, and, responsive to the application script, the first one or more network applications can perform a first set of tasks of the requested task. The scripting engine can provide the application script to the second one or more network applications, and, responsive to the application script, the second one or more network applications can perform a second set of tasks of the requested task. In some embodiments, the first one or more network applications can transfer data corresponding to the first set of tasks upon completing them to the second one or more network applications through the domain of trust. The second one or more network applications can receive the data corresponding to the first set of tasks. The second one or more network applications can use the data corresponding to the first set of tasks to complete or otherwise perform the second set of tasks. Thus, the first one or more network applications and the second one or more network applications can use the domain of trust within the embedded browser of the client application to share data and collaborate to perform one or more tasks. The client application, via the domain of trust, can provide ease of access and functionality for performing tasks using the different network applications of different entities and/or different origins. The domain of trust can provide systems and methods for different network applications of different entities and/or different origins to interact with each other to share and exchange information via the client application and the embedded browser.

In some embodiments, the client application can allow, responsive to a policy the application script to interact across the first origin and the second origin. The client application can generate and/or assign polices to specify which network applications or what properties network applications should have to be identified as trusted and connected or included within a domain of trust. In some embodiments, the policies can specify that network applications originating from a particular origin and/or of a particular entity can be trusted or not-trusted. For example, the policies can specify that network applications originating from a first origin and/or a first entity can be trusted and included within a domain of trust for a user of the client device. The policies can specify that network applications originating from a second origin and/or a second entity may not be trusted and thus, can be blocked or prevented from executing within a domain of trust for a user of the client device. In some embodiments, the policies can specify that network applications originating form the first origin can be trusted to interact and collaborate with network applications originating from a second, different origin within a domain of trust. For example, the policy can indicate or specify that the first origin and the second origin are

trusted origins to interact across via the embedded browser of the client application. The policy can indicate or specify that a first entity and a second entity are trusted entities to interact across via the embedded browser of the client application.

Referring to FIG. 13 depicts a flow diagram of one embodiment of a method 1300 for collaborating across network applications of different origins in an embedded browser. The functionalities of the method may be implemented using, or performed by, the components detailed herein in connection with FIGs. 1–11. In brief overview, a first session can be established to a first network application by a client application (1305). A second session can be established to a second network application by the client application (1310). A trust level of the network applications can be determined (1315). Interaction between the network applications can be allowed (1320).

Referring now to operation (1305), and in some embodiments, a client application can establish a first session to a first network application. The client application can establish a first one or more sessions with a first one or more network applications of a first entity at a first origin via an embedded browser within the client application. Referring now to operation (1310), and in some embodiments, a client application can establish a second session to a second network application. The client application can establish a second one or more sessions with a second one or more network applications of a second entity at a second origin via an embedded browser within the client application.

The client application can be executing on a client device through an embedded browser. The client device can be associated with the user. The client application can include the embedded browser to provide access to one or more network applications for a user of a client device. The client application can establish for a user of a client device one or more sessions with one or more network applications via the embedded browser within the client application. The client application can be executing on a client device through an embedded browser. The client device can be associated with the user. The sessions can include any type or form of a session as described herein. For example, sessions may include, but not limited to, an application session, an execution session, a desktop sessions, a hosted desktop session, a terminal services session, a browser session, a remote desktop session, and a remote application session. Sessions may include encrypted and/or secure sessions established between a network application and a client device of the user.

The network applications can include an application hosted on a server (e.g., third party server) accessed by the client device via a network. The network applications can include, but not limited to, a web application, a desktop application, a virtual application, a software as a service (SaaS) application, a mobile application, an HDX application, a local application, a native application (e.g., native to the client device), and/or a device couple with the client device. The network applications may include applications (apps) that are served from and/or hosted on one or more servers. For example, each of the network applications can include one or more locations or one or more points of presence that can provide content associated with the respective network application. In some embodiments, the network applications can include an application hosted on a server accessed by a client device of the user via one or more networks. The network applications can include or execute one or more web pages. The client application can provide access to local files or native files, for example, such as files executing on the client device.

Referring now to operation (1315), and in some embodiments, a client application can determine a trust level of the network applications. For example, the client application can determine, responsive to a policy, that the first origin and the second origin are origins to be trusted to interact across via the embedded browser. Network applications from different entities and/or different origins can be executing within the embedded browser of the client application. The client application can determine a trust level for each of the network applications executing within the embedded browser of the client application. The network can be assigned the same trust level. One or more network applications can be assigned one or more different trust levels. The trust levels can correspond to access privileges for a network application and/or for a user of the client device to interact with a respective one of the network applications. For example, the trust levels can correspond to access privileges such as, but not limited to, access to different devices or systems of the client device, security privileges, functionalities of the respective network application within the embedded browser and/or tasks the respective network application or device can perform within the embedded browser. In some embodiments, responsive to a being assigned a particular trust level, a network application can be allowed to, but not limited to, transfer data to a second network application of a different entity and/or a different origin within the embedded browser of the client application.

The client application can apply one or more policies to properties of one or more network applications to determine a trust level of the respective network applications. For

example, the trust levels can be assigned based in part on an entity that provides the particular network application, an origin of the particular network application, or a combination of both. For example, the trust level for a network application can be selected based in part on the origin the network application originates from. Network applications of different origins and/or of different entities can be assigned the same trust level. In some embodiments, network applications of different origins and/or of different entities can be assigned one or more different trust levels.

The policies can specify a plurality of different origins to trust. For example, the plurality of different origins can include the first origin and the second origin. The policies can indicate that a network application from a particular origin and/or entity can interact with and collaborate with one or more network applications from one or more different origins and/or different entities. The client application can apply the polices to one or more network applications to determine which other network applications executing within the embedded browser the particular network application can interact with based in part on their origins. For example, the client application can determine, responsive to a policy, that the first origin and the second origin are origins to be trusted to interact across via the embedded browser. Thus, responsive to the policy, network applications from the first origin can be allowed to collaborate with and interact with network applications from the second origin.

In some embodiments, the first origin and the second origin fail a same origin policy. For example, the origins of the network applications can include or apply same origin policies to the network applications they host or provide when executing within a browser of the respective origin. The same origin policy can indicate that network applications can only interact with network applications from the same origin. Thus, the first origin being different from the second origin can fail the same origin policy. However, the client application, via the embedded browser, can apply one or more policies to override or otherwise ignore the same origin policies of the origins of the network applications executing within the embedded browser of the client application. For example, as each of the network applications are executing within the embedded browser of the client application, the client application can generate and apply policies' to control operation and/or functionality of the respective network applications for a user of the client device. Therefore, the first origin and the second origin can fail the same origin policy, however, within the embedded browser of the client application, the first

Referring now to operation (1320), and in some embodiments, a client application can allow interaction between the network applications. For example, the client application can allow, responsive to the determination, the first one or more networks applications of the first entity at the first origin to interact via the embedded browser with the second one or more network applications of the second entity at the second origin. The interaction can include performing one or more tasks using one or more network applications executing within the embedded browser of the client application. The tasks may include, but not limited to, printing functionality, storing data or files in particular locations (e.g., on the client device, on a network server), download functionality, upload functionality, camera functionality, viewing device functionality, taking a picture functionality, obtaining a signature functionality, and/or digital signature functionality. For example, client application, responsive to the determination, can allow a first network application of a first entity of a first origin to transfer data from a second, different network application of a second, different entity of a second, different origin, via the client application and embedded browser. The different network applications of different origins and/or different entities can collaborate and interact with each to perform one or more tasks for a user of the client device.

In some embodiments, a scripting engine of the client application can execute a script to interact via the embedded browser between the first one or more networks applications of the first entity at the first origin and the second one or more network applications of the second entity at the second origin. The scripting engine can identify or generate an application script to perform a requested task for a user of the client device. The application scripts can correspond to the task to be performed and/or the network applications to perform the requested task. The application script can include one can include one or more instructions or series of commands. The instructions can include a series of commands or code to be executed to perform one or more tasks. The instructions can include or identify the task to be performed, one or more network applications to perform the respective task, the entity of the respective one or more network applications, the origin of the respective one or more network applications, and policies corresponding to the respective network applications, entities, and/or origins.

The scripting engine can provide or other apply the application script to the network applications identified to perform a requested task for a user of the client device. A first one or more network applications of a first entity and a first origin and a second one or more

network applications of a second entity of a second origin can be executing within the embedded browser of the client application. The scripting engine can provide or apply the application script to the first one or more network applications and the second one or more network applications. The scripting engine can initiate execution of one or more applications scripts with the first one or more network applications and/or the second one or more network applications. The order of execution of the one or more application scripts can be included within the instructions of the respective application scripts and/or based in part on the task to be performed.

In some embodiments, the client application can allow interacting responsive to authentication of the same user to each of the first one or more networks applications and the second one or more network applications. The client application can authenticate a user of the client device. The authentication can include authenticating the user for one or more network applications executing within the embedded browser of the client application. The client application can establish one or more sessions with one or more network applications responsive to authentication of the user. The client application can provide access to one or more sessions with one or more network applications responsive to authentication of the user of the client device within the embedded browser of the client application. The authentication can be performed responsive to receiving authentication credentials for a user of the client device. The client application can establish one or more domains of trust for the network applications the user has access or privileges to interact with. Responsive to establishing the domains of trust, the client application can establish sessions with the network applications the user has access or privileges to interact with though one or more domains of trust. The network applications can include multiple network applications of different entities and/or different origins. The client application can establish trust relationships between the network applications of different entities and/or different origins using the domain of trust. The client application can establish access policies for the domain of trust. For example, in some embodiments, the access policy for the access to the domain of trust to the user of the client device. The client application can perform one or more tasks for the user of the client device using the domain of trust. For example, the scripting engine can initiate execution of the application script by one of the first one or more network applications or the second one or more network applications within the domain of trust.

It should be understood that the systems described above may provide multiple ones of any or each of those components and these components may be provided on either a standalone machine or, in some embodiments, on multiple machines in a distributed system. The systems and methods described above may be implemented as a method, apparatus or article of manufacture using programming and/or engineering techniques to produce software, firmware, hardware, or any combination thereof. In addition, the systems and methods described above may be provided as one or more computer-readable programs embodied on or in one or more articles of manufacture. The term "article of manufacture" as used herein is intended to encompass code or logic accessible from and embedded in one or more computer-readable devices, firmware, programmable logic, memory devices (e.g., EEPROMs, ROMs, PROMs, RAMs, SRAMs, etc.), hardware (e.g., integrated circuit chip, Field Programmable Gate Array (FPGA), Application Specific Integrated Circuit (ASIC), etc.), electronic devices, a computer readable non-volatile storage unit (e.g., CD-ROM, USB Flash memory, hard disk drive, etc.). The article of manufacture may be accessible from a file server providing access to the computer-readable programs via a network transmission line, wireless transmission media, signals propagating through space, radio waves, infrared signals, etc. The article of manufacture may be a flash memory card or a magnetic tape. The article of manufacture includes hardware logic as well as software or programmable code embedded in a computer readable medium that is executed by a processor. In general, the computer-readable programs may be implemented in any programming language, such as LISP, PERL, C, C++, C#, PROLOG, or in any byte code language such as JAVA. The software programs may be stored on or in one or more articles of manufacture as object code.

While various embodiments of the methods and systems have been described, these embodiments are illustrative and in no way limit the scope of the described methods or systems. Those having skill in the relevant art can effect changes to form and details of the described methods and systems without departing from the broadest scope of the described methods and systems. Thus, the scope of the methods and systems described herein should not be limited by any of the illustrative embodiments and should be defined in accordance with the accompanying claims and their equivalents.

CLAIMS

We claim:

1. A method for executing in an embedded browser an application script for network applications of different origins, the method comprising:

(a) establishing, by a client application, a first one or more sessions with a first one or more network applications of a first entity at a first origin via an embedded browser within the client application;

(b) establishing, by the client application, a second one or more sessions with a second one or more network applications of a second entity at a second origin via the embedded browser within the client application;

(c) identifying, by a scripting engine within the client application of a client device of a user at a third origin, an application script comprising instructions to interact with each the first one or more networks applications of the first entity at the first origin and the second one or more network applications of the second entity at the second origin; and

(d) executing, by the scripting engine, the instructions of the application script to perform a task across the first one or more networks applications of the first entity at the first origin and the second one or more network applications of the second entity at the second origin.

2. The method of claim 1, wherein the first one or more network applications comprises a suite of applications originating from or hosted by a first one or more servers at the first origin.

3. The method of claim 1, wherein the second one or more network applications comprises a suite of applications originating from or hosted by a second one or more servers at the second origin.

4. The method of claim 1, wherein the first origin and the second origin are different origins that fail a same origin policy.

5. The method of claim 1, further comprising establishing, by the client application, a domain of trust between the first one or more networks applications of the first entity at the

first origin and the second one or more network applications of the second entity at the second origin.

6. The method of claim 5, wherein the domain of trust is established responsive to authentication of the same user for the first one or more network applications and the second one or more network applications via the client application on the client device of a third entity of the user corresponding to the third origin.

7. The method of claim 1, further comprising initiating execution of the application script by one of the first one or more network applications or the second one or more network applications.

8. The method of claim 1, wherein (d) further comprising allowing, by the client application responsive to a policy, the application script to interact across the first origin and the second origin.

9. The method of claim 8, wherein the policy specifies that the first origin and the second origin are trusted origins to interact across via the embedded browser of the client application.

10. A method for collaborating across network applications of different origins in an embedded browser, the method comprising:

(a) establishing, by a client application, a first one or more sessions with a first one or more network applications of first entity at a first origin via an embedded browser within the client application;

(b) establishing, by the client application, a second one or more sessions with a second one or more network applications of a second entity at a second origin via the embedded browser within the client application;

(c) determining, by the client application responsive to a policy, that the first origin and the second origin are origins to be trusted to interact across via the embedded browser; and

(d) allowing, by the client application responsive to the determination, the first one or more networks applications of the first entity at the first origin to interact via the

embedded browser with the second one or more network applications of the second entity at the second origin.

11. The method of claim 10, wherein the first origin and the second origin fail a same origin policy.

12. The method of claim 10, wherein the policies specifies a plurality of different origins to trust, the plurality of different origins comprising the first origin and the second origin.

13. The method of claim 10, further comprising executing, by a scripting engine of the client application, a script to interact via the embedded browser between the first one or more networks applications of the first entity at the first origin and the second one or more network applications of the second entity at the second origin.

14. The method of claim 10, wherein (d) further comprises allowing interacting responsive to authentication of the same user to each of the first one or more networks applications and the second one or more network applications.

15. A system for collaborating across network applications of different origins in an embedded browser, the system comprising:

a client application executable on one or more processors of a client device and configured to:

establish a first one or more sessions with a first one or more network applications of first entity at a first origin via an embedded browser within the client application and a second one or more sessions with a second one or more network applications of a second entity at a second origin via the embedded browser within the client application;

determine, responsive to a policy, that the first origin and the second origin are origins to be trusted to interact across via the embedded browser; and

allow, responsive to the determination, the first one or more networks applications of the first entity at the first origin to interact via the embedded browser with the second one or more network applications of the second entity at the second origin.

16. The system of claim 15, wherein the first origin and the second origin are different origins that fail a same origin policy.

17. The system of claim 15, wherein the policy specifies a plurality of different origins to trust, the plurality of different origins comprising the first origin and the second origin.

18. The system of claim 15, further comprising a scripting engine configured to execute instructions of a script to perform via the embedded browser a task across the first one or more networks applications of the first entity at the first origin and the second one or more network applications of the second entity at the second origin.

19. The system of claim 15, wherein the first one or more network applications comprises a suite of applications originating from or hosted by a first one or more servers at the first origin.

20. The system of claim 15, wherein the client application is configured to allow the interaction responsive to authentication of the same user to each of the first one or more networks applications and the second one or more network applications.

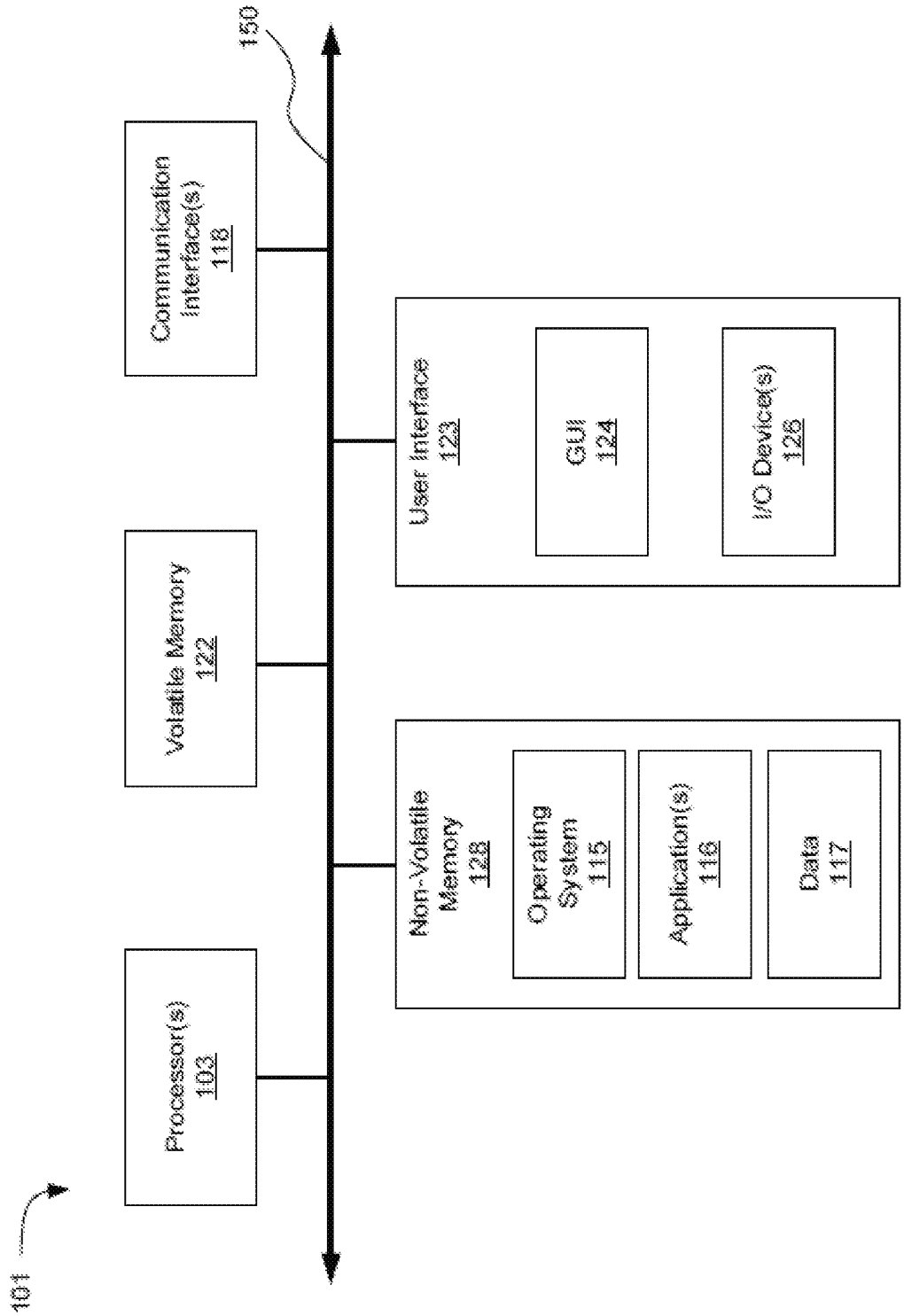

101
150
Processor(s)
103
Volatile Memory
122
Non-Volatile
Memory
128
Operating
System
115
Application(s)
116
Data
117
User Interface
123
GUI
124
I/O Device(s)
126
Communication
interface(s)
118

FIG. 1

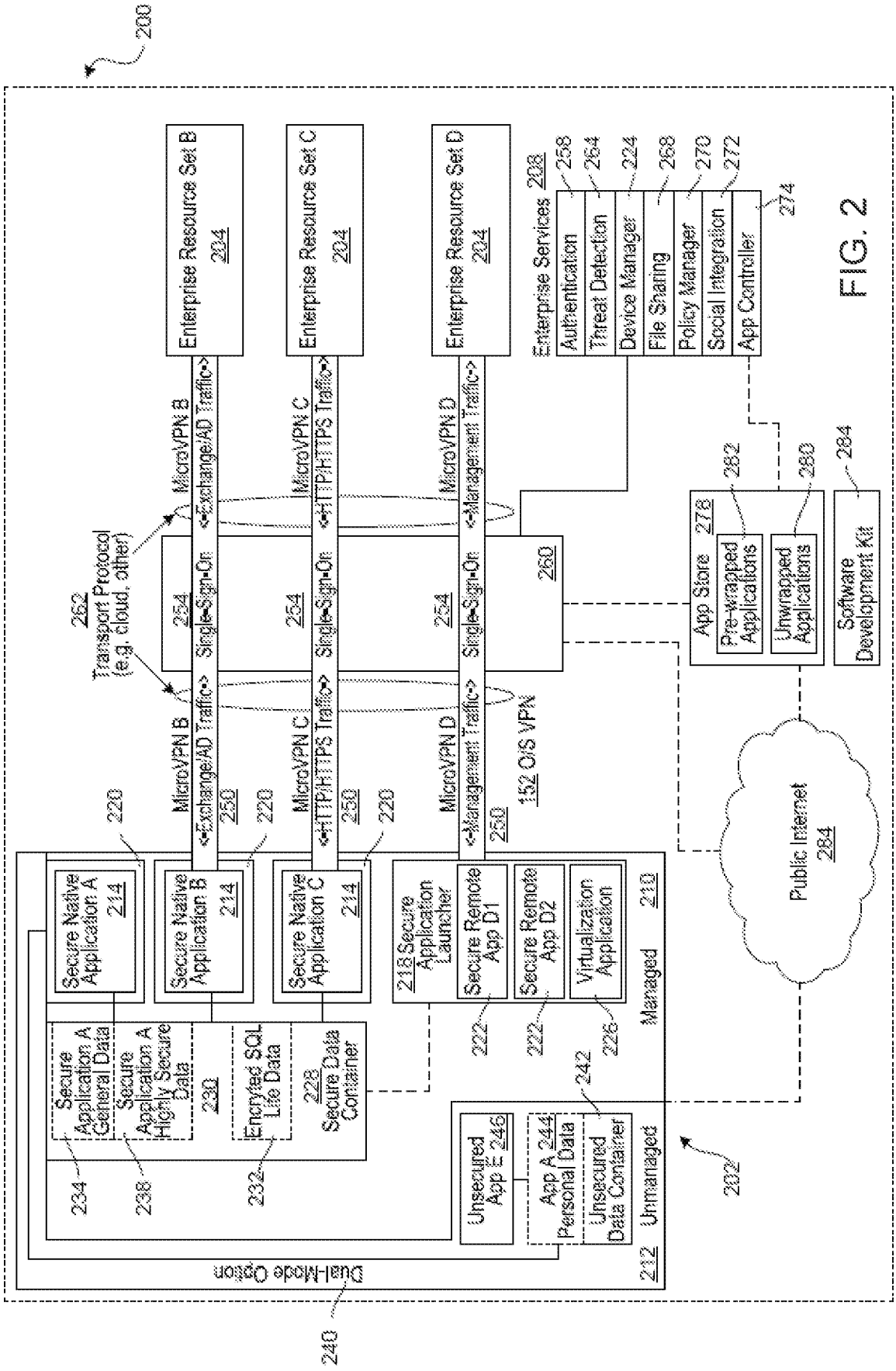

200
Enterprise Resource Set B 204
Enterprise Resource Set C 204
Enterprise Resource Set D 204
Enterprise Services 208
258 Authentication
264 Threat Detection
224 Device Manager
268 File Sharing
270 Policy Manager
272 Social Integration
274 App Controller
262 Transport Protocol (e.g. cloud, other)
MicroVPN B
MicroVPN C
MicroVPN D
<-Exchange/AD Traffic->
<-HTTP/HTTPS Traffic->
<-Management Traffic->
254 Single-Sign-On
250
220
260
152 OiS VPN
App Store 278
Pre-wrapped Applications
Unwrapped Applications
280
282
Software Development Kit
284
Public Internet 284
Secure Native Application A 214
Secure Native Application B 214
Secure Native Application C 214
218 Secure Application Launcher
Secure Remote App D1
Secure Remote App D2
Virtualization Application
222
226
Managed 210
Secure Application A General Data
Secure Application A Highly Secure Data
230
234
238
Encrypted SQL Lite Data
232
228 Secure Data Container
242
Unsecured App E 246
App A 244 Personal Data
Unsecured Data Container
Unmanaged 212
202
Dual-Mode Option
240

FIG. 2

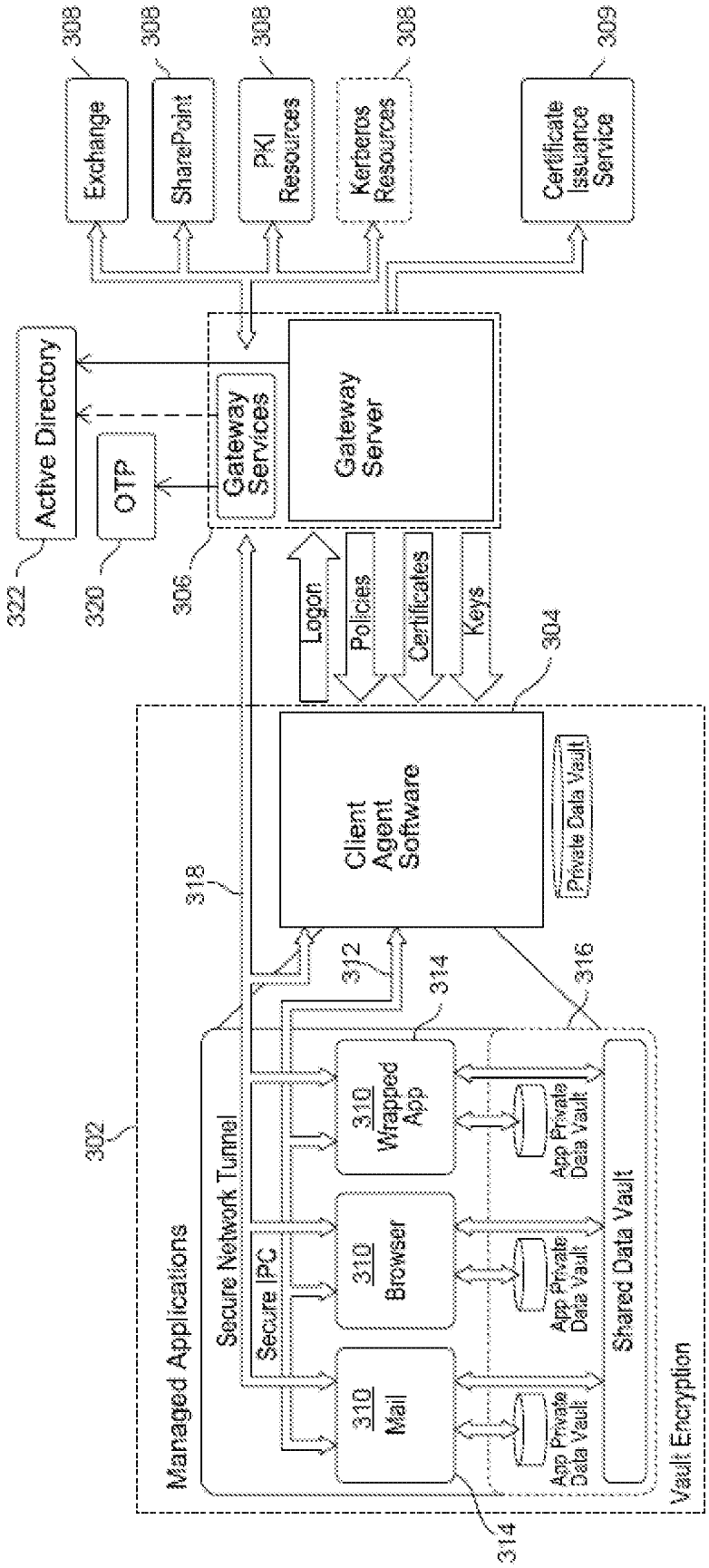

308
Exchange
308
SharePoint
308
PKI Resources
308
Kerberos Resources
309
Certificate Issuance Service
Active Directory
OTP
Gateway Services
Gateway Server
322
320
306
Logon
Policies
Certificates
Keys
304
Client Agent Software
Private Data Vault
318
312
314
316
302
Managed Applications
Secure Network Tunnel
Secure IPC
310 Wrapped App
310 Browser
310 Mail
App Private Data Vault
App Private Data Vault
App Private Data Vault
Shared Data Vault
Vault Encryption
314
300

FIG. 3

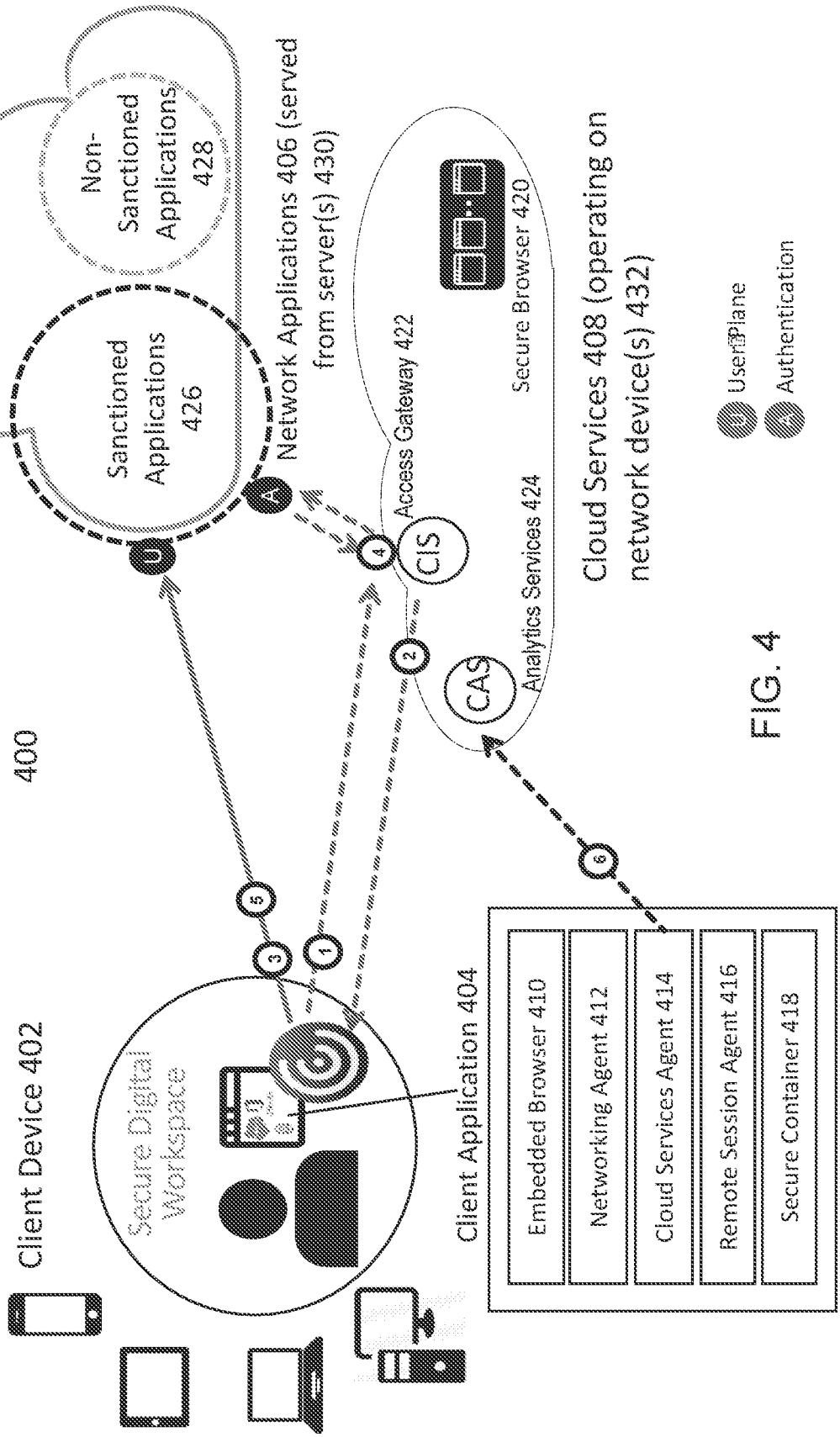

400
Client Device 402
Secure Digital Workspace
Client Application 404
Embedded Browser 410
Networking Agent 412
Cloud Services Agent 414
Remote Session Agent 416
Secure Container 418
Cloud Services 408 (operating on network device(s) 432)
Analytics Services 424
CAS
CIS
Access Gateway 422
Secure Browser 420
Network Applications 406 (served from server(s) 430)
Sanctioned Applications 426
Non-Sanctioned Applications 428
User Plane
Authentication

FIG. 4

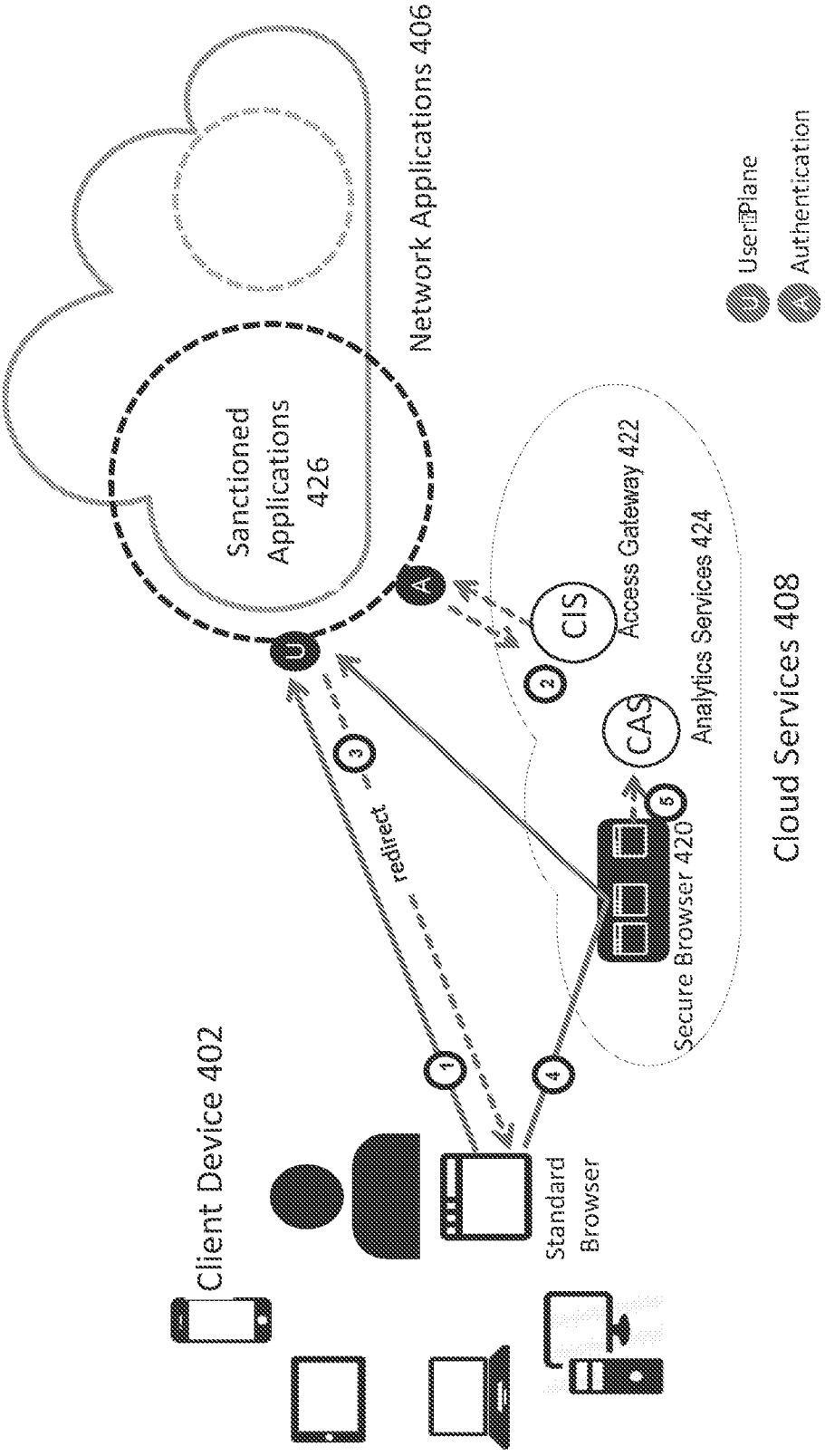

Network Applications 406
Sanctioned Applications 426
Cloud Services 408
Access Gateway 422
CIS
Analytics Services 424
CAS
Secure Browser 420
redirect
Client Device 402
Standard Browser
User Plane
Authentication

FIG. 5

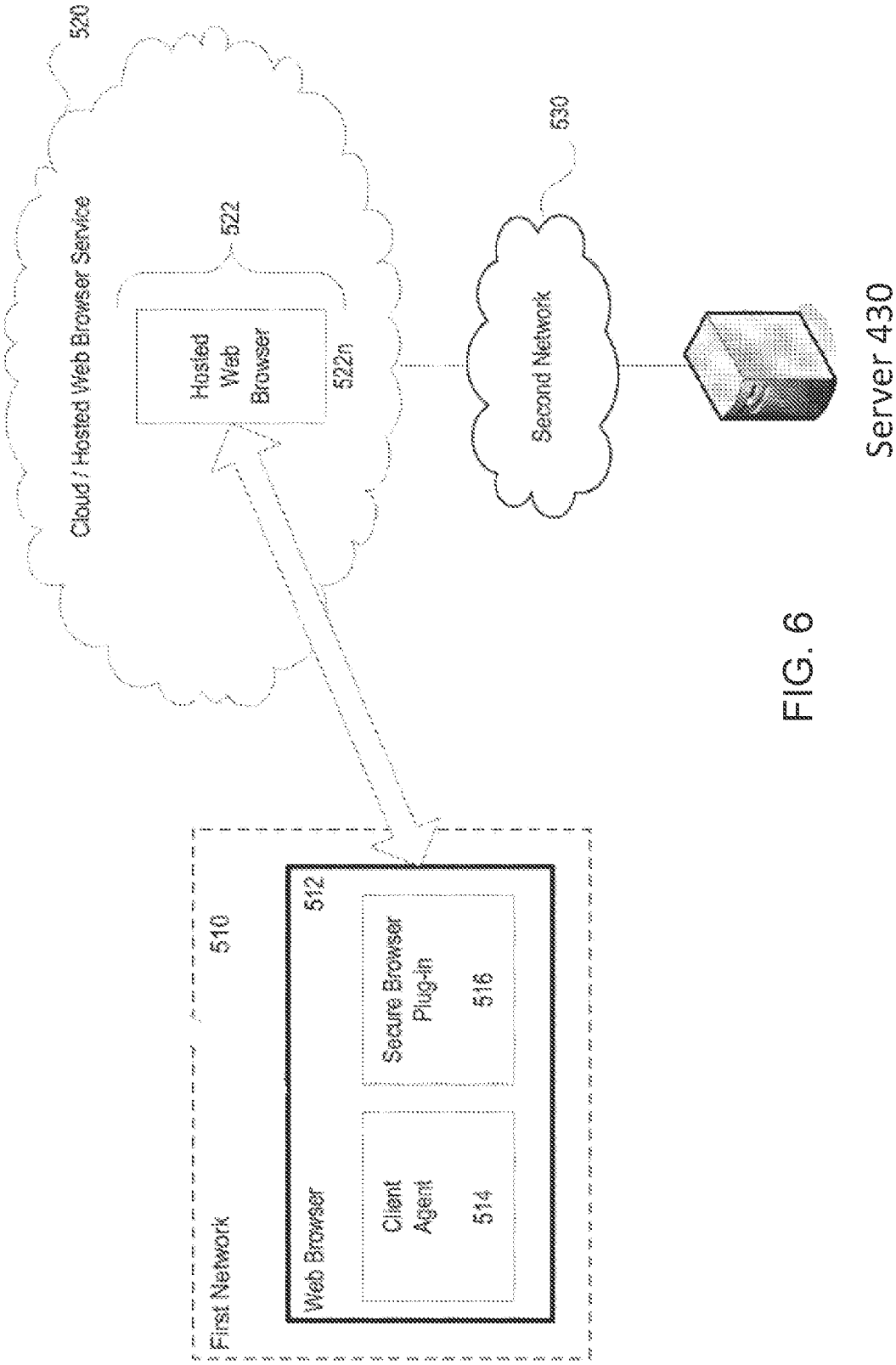

520
Cloud / Hosted Web Browser Service
522
Hosted Web Browser
522n
530
Second Network
Server 430
First Network
510
512
Web Browser
Client Agent
514
Secure Browser Plug-in
516

FIG. 6

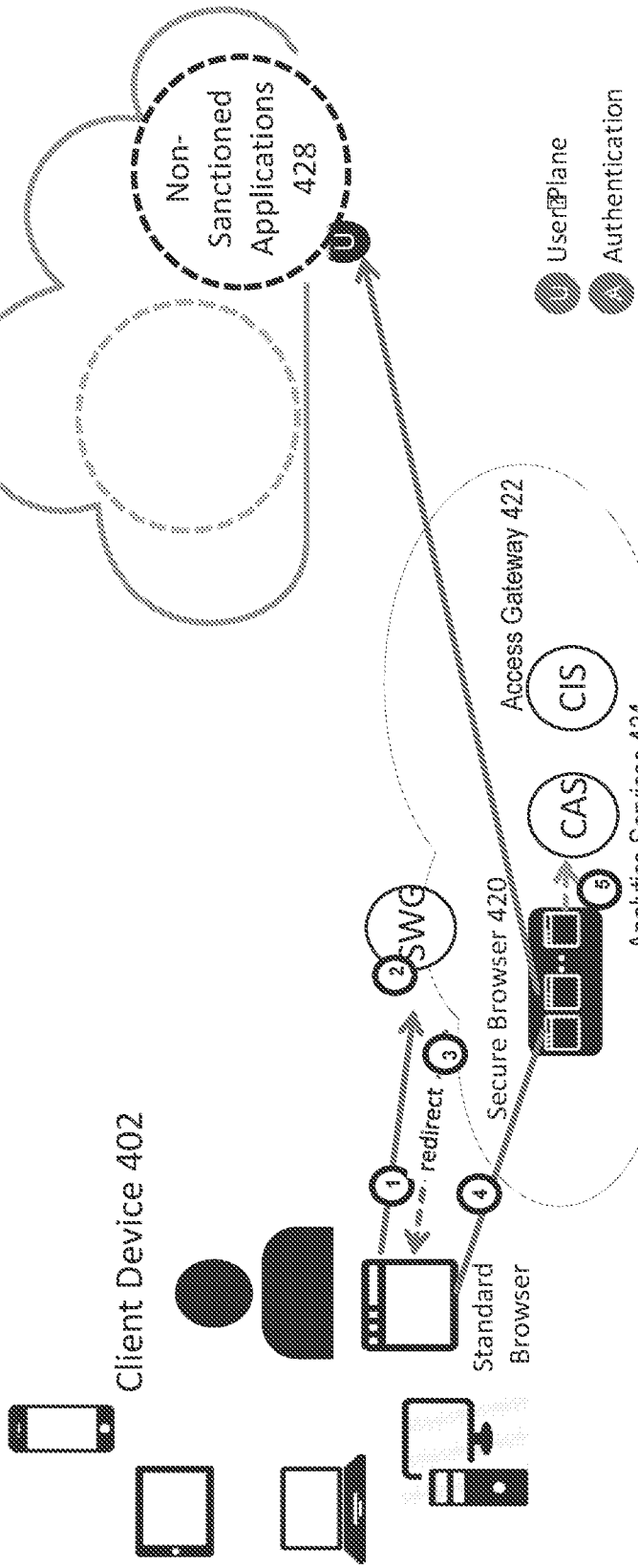

Network Applications 406
Non-Sanctioned Applications 428
User Plane
Authentication
Access Gateway 422
CIS
CAS
SWG
redirect
Secure Browser 420
Analytics Services 424
Cloud Services 408
Client Device 402
Standard Browser

FIG. 7

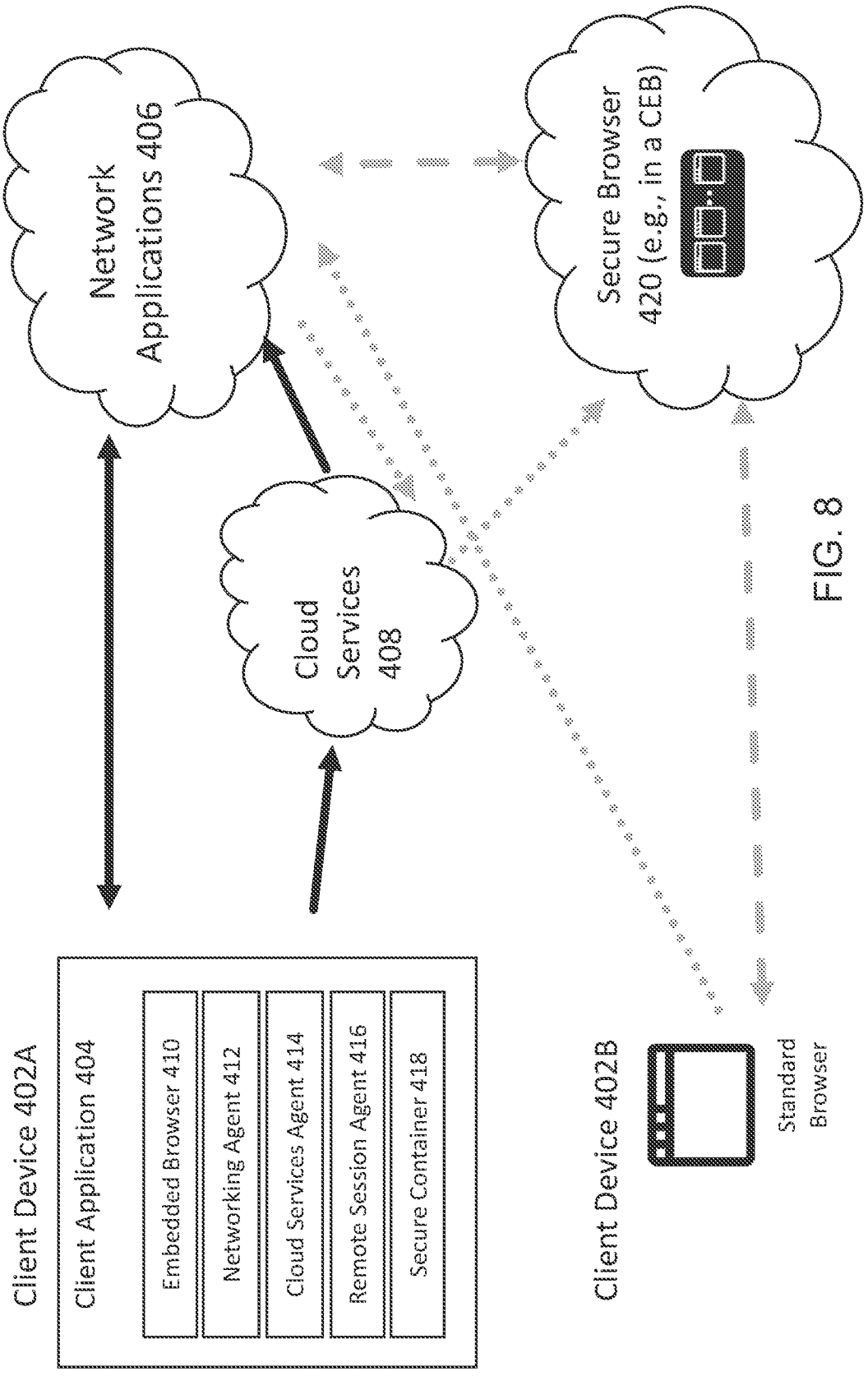
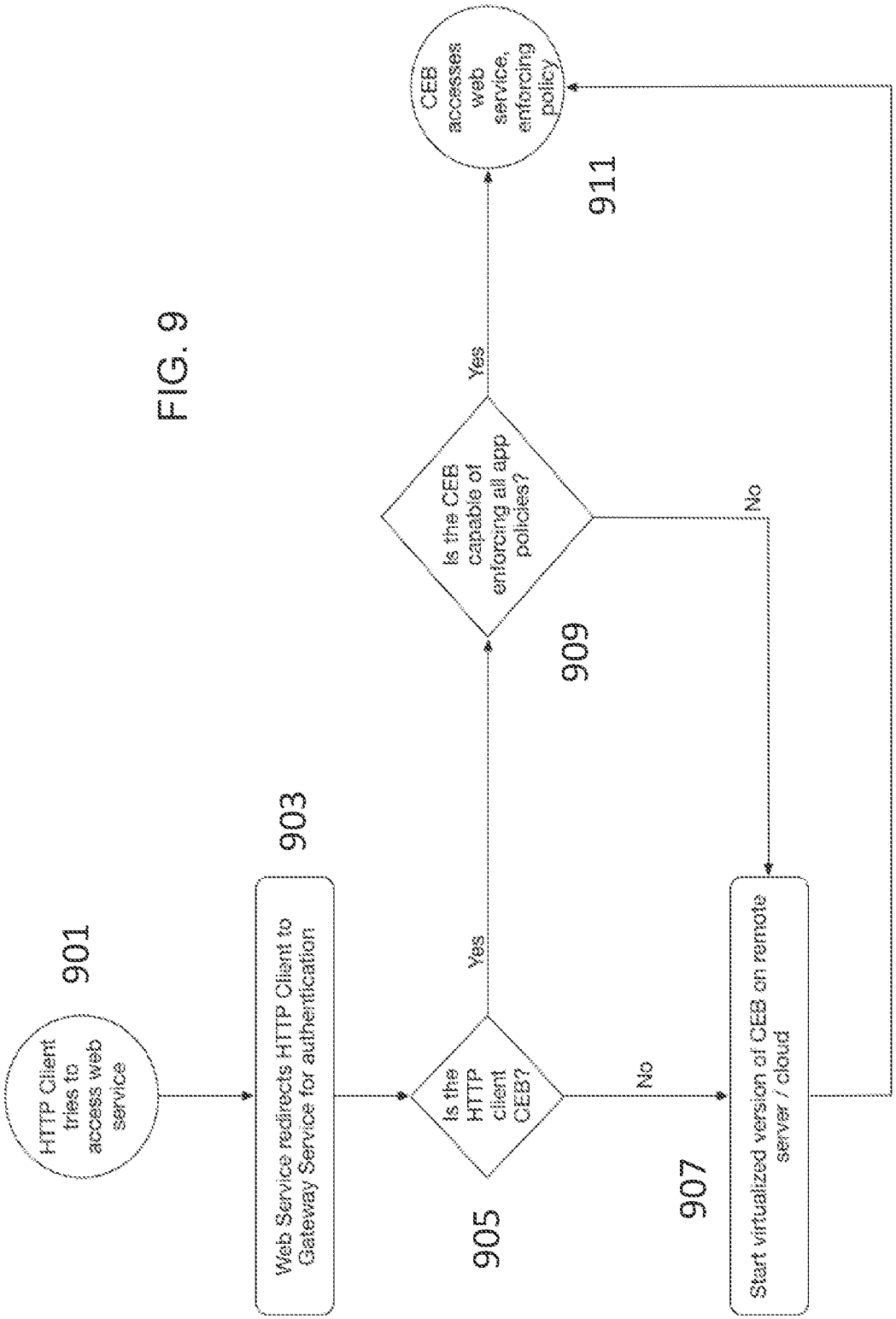

Network Applications 406
Secure Browser 420 (e.g., in a CEB)
Cloud Services 408
Client Device 402A
Client Application 404
Embedded Browser 410
Networking Agent 412
Cloud Services Agent 414
Remote Session Agent 416
Secure Container 418
Client Device 402B
Standard Browser

FIG. 8

FIG. 9


HTTP Client tries to access web service
901
Web Service redirects HTTP Client to Gateway Service for authentication
903
Is the HTTP client CEB?
905
Yes
No
Is the CEB capable of enforcing all app policies?
909
Yes
No
CEB accesses web service, enforcing policy
911
Start virtualized version of CEB on remote server / cloud
907

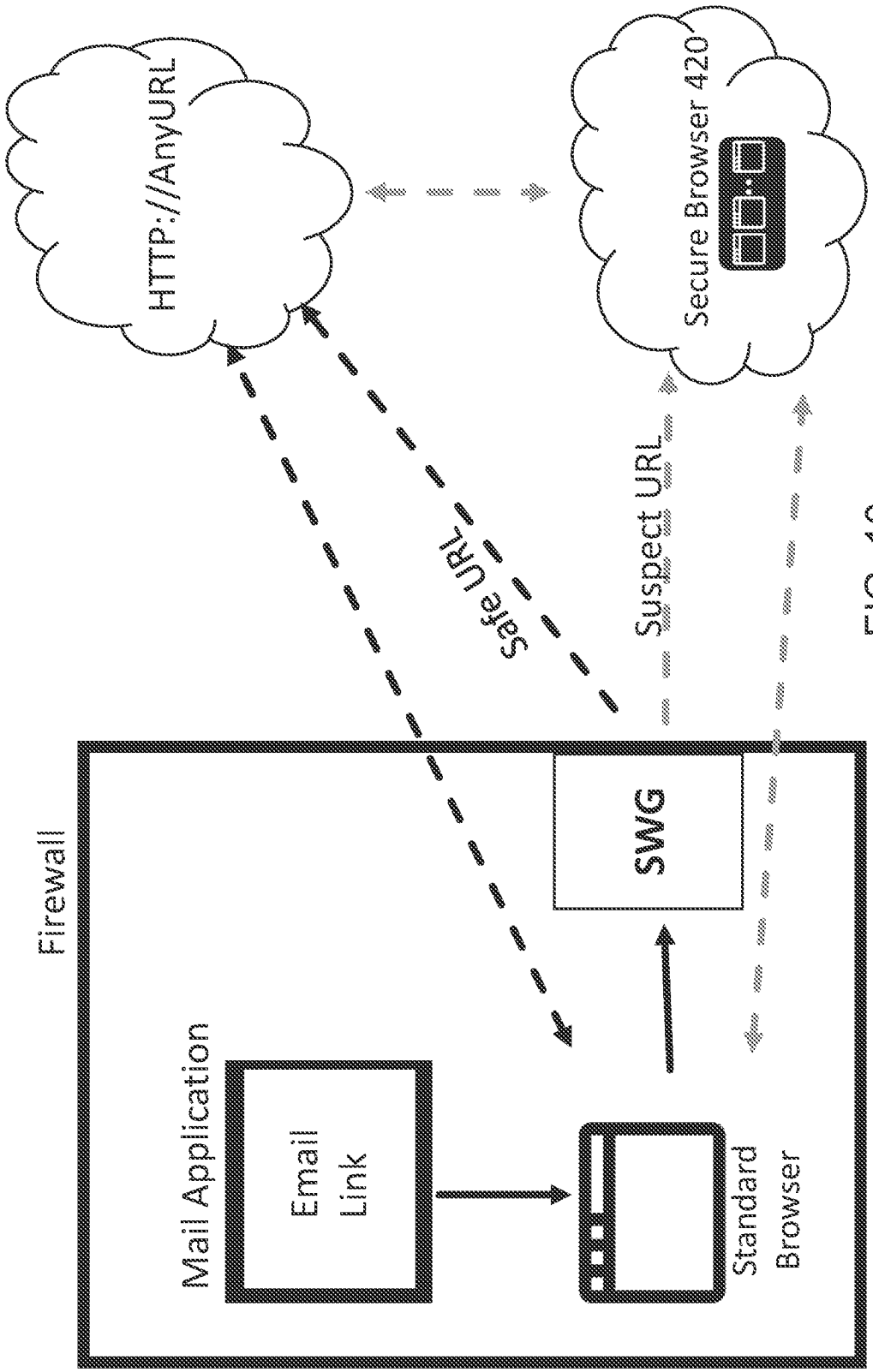

HTTP://AnyURL
Secure Browser 420
Safe URL
Suspect URL
Firewall
SWG
Mail Application
Email Link
Standard Browser

FIG. 10

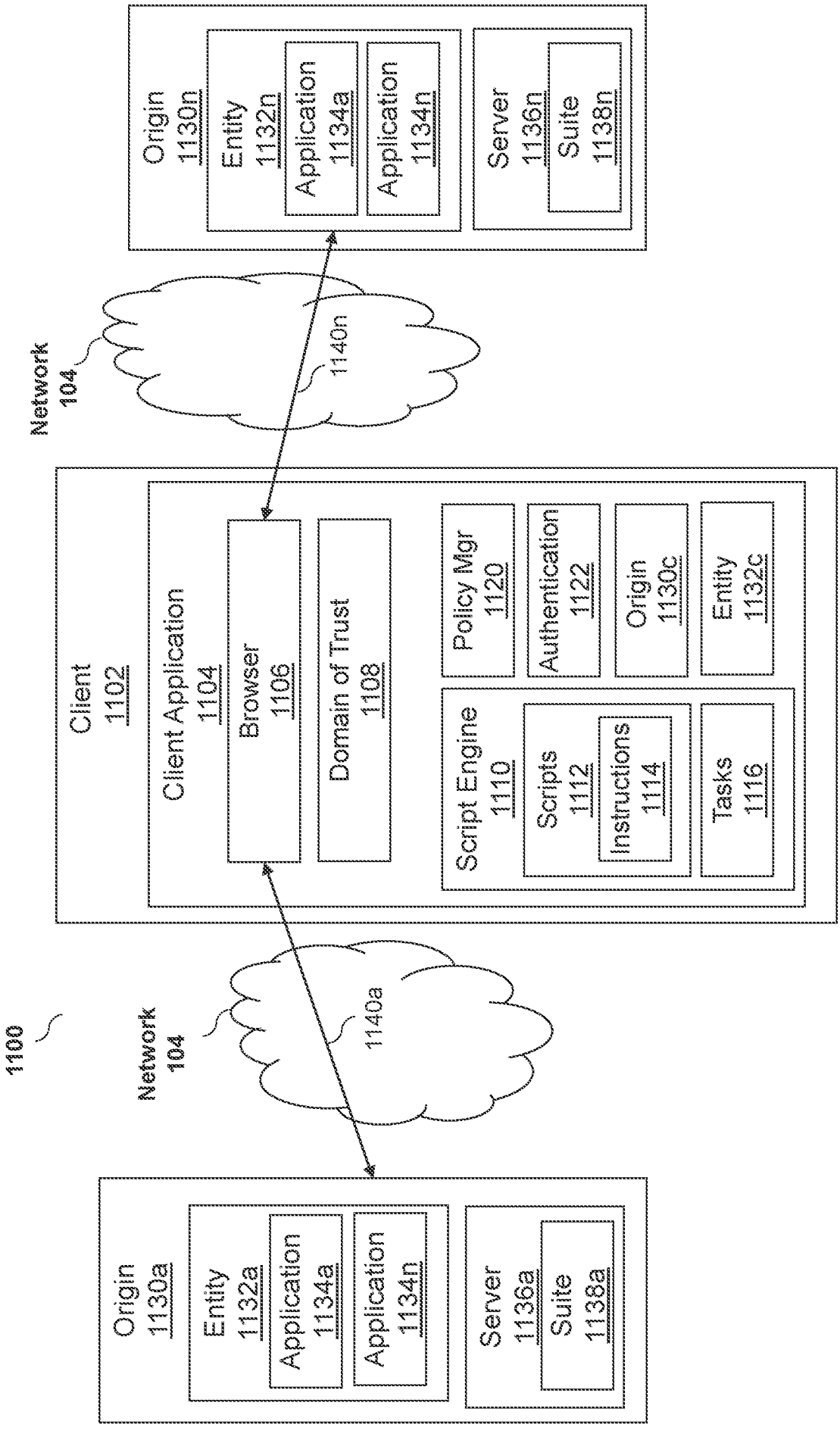

1100
Network 104
1140a
Origin 1130a
Entity 1132a
Application 1134a
Application 1134n
Server 1136a
Suite 1138a
Client 1102
Client Application 1104
Browser 1106
Domain of Trust 1108
Script Engine 1110
Scripts 1112
Instructions 1114
Tasks 1116
Policy Mgr 1120
Authentication 1122
Origin 1130c
Entity 1132c
Network 104
1140n
Origin 1130n
Entity 1132n
Application 1134a
Application 1134n
Server 1136n
Suite 1138n

FIG. 11

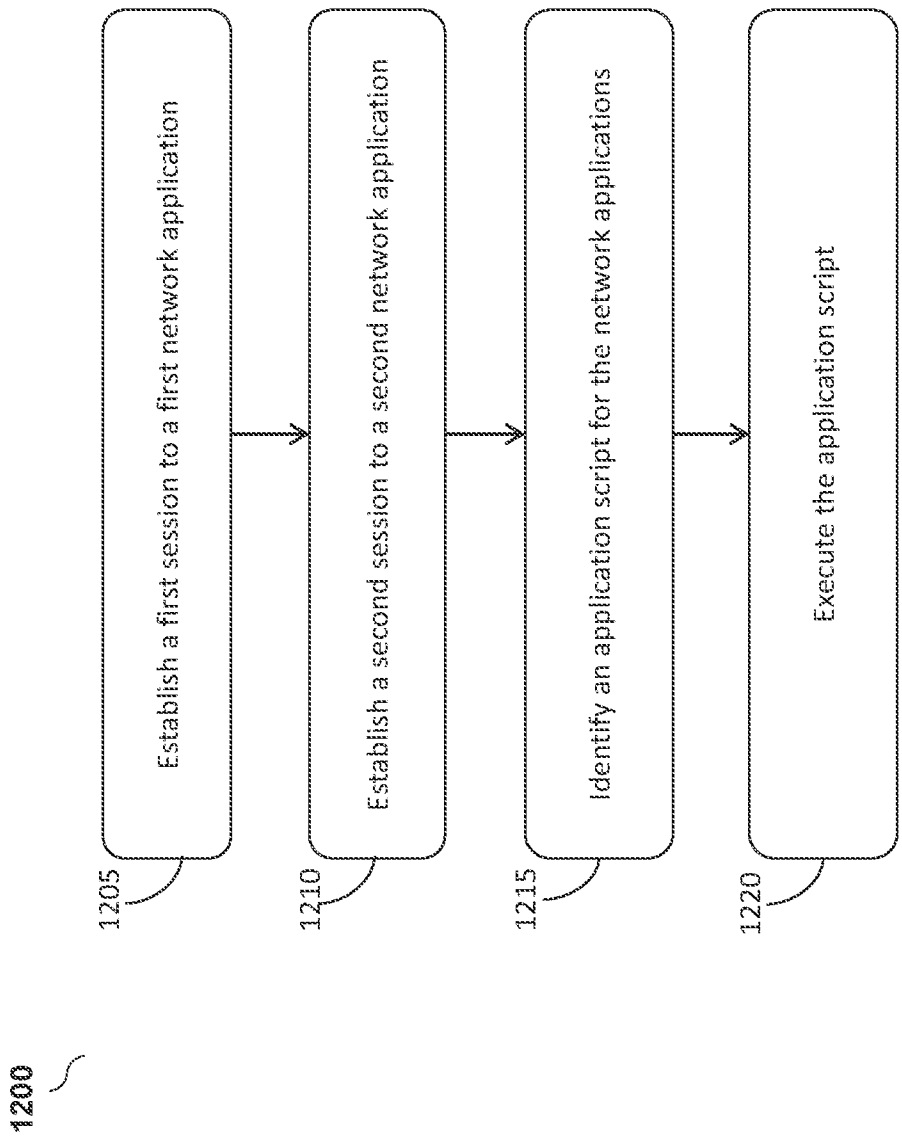

1200
1205
Establish a first session to a first network application
1210
Establish a second session to a second network application
1215
Identify an application script for the network applications
1220
Execute the application script

FIG. 12

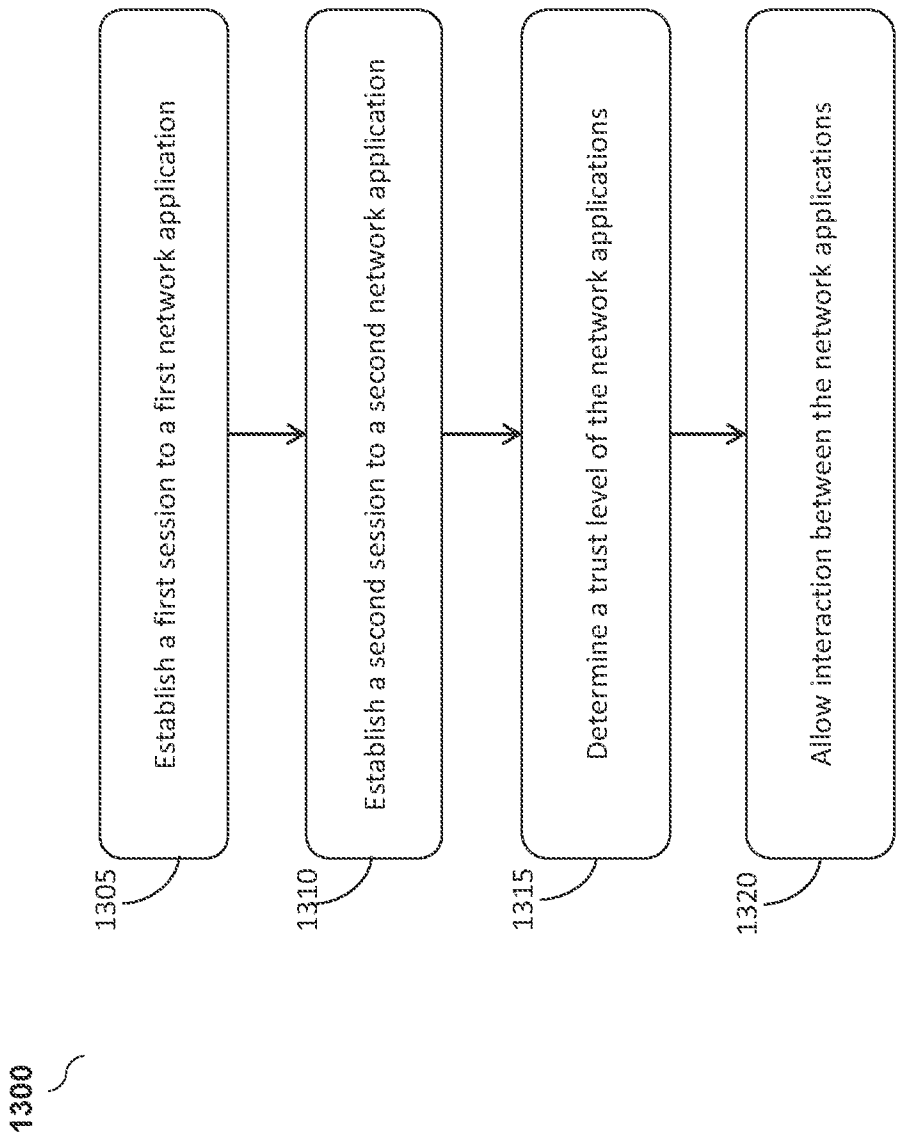

1300
1305
Establish a first session to a first network application
1310
Establish a second session to a second network application
1315
Determine a trust level of the network applications
1320
Allow interaction between the network applications

FIG. 13

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2019/050386

A. CLASSIFICATION OF SUBJECT MATTER
INV. H04L29/08 H04L29/06 H04W12/08
ADD.

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
H04L H04W

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

EPO-Internal, COMPENDEX, INSPEC, IBM-TDB, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	WO 2014/188049 A1 (NOKIA CORP [FI]) 27 November 2014 (2014-11-27) abstract paragraph [0006] - paragraph [0008] paragraph [0018] - paragraph [0028] paragraph [0031] - paragraph [0035] -----	1-4,7,8
A	US 2009/138937 A1 (ERLINGSSON ULFAR [US] ET AL) 28 May 2009 (2009-05-28) abstract paragraph [0014] - paragraph [0021] paragraph [0028] - paragraph [0029] paragraph [0038] - paragraph [0054] paragraph [0067] -----	1-4,7,8

☐ 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 another 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	Date of mailing of the international search report
22 November 2019	28/01/2020

Name and mailing address of the ISA/	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016	Aura Marcos, F

1

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2019/050386

Box No. II Observations where certain claims were found unsearchable (Continuation of item 2 of first sheet)

This international search report has not been established in respect of certain claims under Article 17(2)(a) for the following reasons:

1. ☐ Claims Nos.:
because they relate to subject matter not required to be searched by this Authority, namely:

2. ☐ Claims Nos.:
because they relate to parts of the international application that do not comply with the prescribed requirements to such an extent that no meaningful international search can be carried out, specifically:

3. ☐ Claims Nos.:
because they are dependent claims and are not drafted in accordance with the second and third sentences of Rule 6.4(a).

Box No. III Observations where unity of invention is lacking (Continuation of item 3 of first sheet)

This International Searching Authority found multiple inventions in this international application, as follows:

see additional sheet

1. ☐ As all required additional search fees were timely paid by the applicant, this international search report covers all searchable claims.

2. ☐ As all searchable claims could be searched without effort justifying an additional fees, this Authority did not invite payment of additional fees.

3. ☐ As only some of the required additional search fees were timely paid by the applicant, this international search report covers only those claims for which fees were paid, specifically claims Nos.:

4. ☒ No required additional search fees were timely paid by the applicant. Consequently, this international search report is restricted to the invention first mentioned in the claims; it is covered by claims Nos.:

1-4, 7, 8

Remark on Protest

☐ The additional search fees were accompanied by the applicant's protest and, where applicable, the payment of a protest fee.

☐ The additional search fees were accompanied by the applicant's protest but the applicable protest fee was not paid within the time limit specified in the invitation.

☐ No protest accompanied the payment of additional search fees.

FURTHER INFORMATION CONTINUED FROM PCT/ISA/ 210

This International Searching Authority found multiple (groups of) inventions in this international application, as follows:

1. claims: 1-4, 7, 8

   Enabling cross application collaboration within an embedded browser of the client application.

   ---

2. claims: 5, 6, 9-20

   Providing a secure environment for network application interaction.

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No
PCT/US2019/050386

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO 2014188049 A1	27-11-2014	CN 105556928 A	04-05-2016
		EP 3000219 A1	30-03-2016
		US 2016094673 A1	31-03-2016
		WO 2014188049 A1	27-11-2014
US 2009138937 A1	28-05-2009	NONE	