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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
4 August 2016 (04.08.2016)

WIPO | PCT

(10) International Publication Number
WO 2016/122623 A1

(51) International Patent Classification:
G06F 9/54 (2006.01) *G06F 21/00* (2013.01)

(21) International Application Number:
PCT/US2015/013864

(22) International Filing Date:
30 January 2015 (30.01.2015)

(25) Filing Language: English

(26) Publication Language: English

(71) Applicant: **HEWLETT PACKARD ENTERPRISE DEVELOPMENT LP** [US/US]; 11445 Compaq Center Drive West, Houston, TX 77070 (US).

(72) Inventor: **WHANG, Yeon**; 5555 Windward Pkwy, Alpharetta, Georgia 30004 (US).

(74) Agents: **POSSETT, Ramya** et al.; Hewlett Packard Enterprise, 3404 E. Harmony Road, Mail Stop 79, Fort Collins, CO 80528 (US).

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

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

Declarations under Rule 4.17:

— *as to the identity of the inventor (Rule 4.17(i))*

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

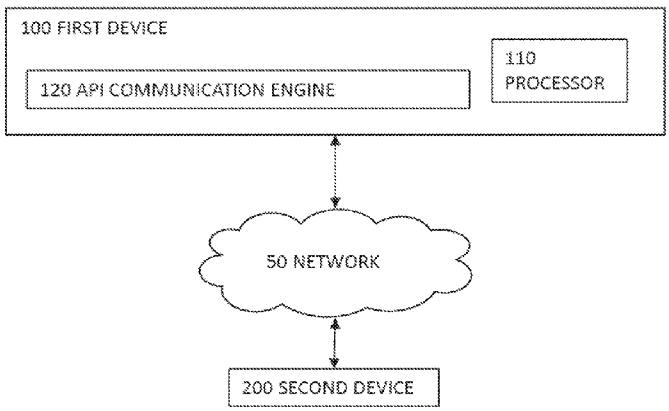
Published:

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

WO 2016/122623 A1

(54) Title: SECURE API COMMUNICATION

FIG. 1



(57) Abstract: An example computing device provides secure Application Programming Interface (API) communication. The example computing device may comprise an API communication engine to communicate with an application executing on a second computing device. The API communication engine may communicate with the application by sending an API call to the second device, the API call comprising a dynamic device access token and first meter information, the first meter information related to first usage of the application. The API communication engine may then receive, from the second computing device, an API response comprising a dynamic first device access token and second meter information, the second meter information related to second usage of the application. The API communication engine may determine whether the communication between the first device and the second device is secure based on a comparison of the API call and the API response.

SECURE API COMMUNICATION

Background

[0001] Devices communicate with applications using Application Programming Interfaces ("APIs"). An API for an application may comprise a set of API calls that a device may use to interact with the application. Each API call may be associated with a set of functionality and/or other API calls. Use of an API call may specify how an application should act responsive to receiving the call. A conversation between a device and an application (e.g., an application executing on a separate device) may comprise a plurality of API calls sent by the device to the application and a corresponding plurality of API responses sent by the application to the device.

Brief Description of the Drawings

[0002] The following detailed description references the drawings, wherein:

[0003] FIG. 1 is a block diagram of an example device engaging in secure API communication;

[0004] FIG. 2 is a block diagram of an example consumer device and provider device engaging in secure API communication;

[0005] FIG. 3A is a flow diagram of an example method for execution by an example consumer device and provider device engaging in secure API communication;

[0006] FIG. 3B is a flow diagram of an example method for execution by an example consumer device and provider device engaging in secure API communication;

[0007] FIG. 4 is a flowchart of an example method for execution by an example consumer device and provider device engaging in secure API communication;

[0008] FIG. 4A is a flowchart of an example method for execution by an example consumer device and provider device engaging in secure API communication;

[0009] FIG. 5 is a block diagram of an example device engaging in secure API communication; and

[0010] FIG. 6 is a block diagram of an example device engaging in secure API communication.

Detailed Description

[0011] The following detailed description refers to the accompanying drawings. Wherever possible, the same reference numbers are used in the drawings and the following description to refer to the same or similar parts. It is to be expressly understood, however, that the drawings are for the purpose of illustration and description only. While several examples are described in this document, modifications, adaptations, and other implementations are possible. Accordingly, the following detailed description does not limit the disclosed examples. Instead, the proper scope of the disclosed examples may be defined by the appended claims.

[0012] Devices communicate with applications using Application Programming Interfaces ("APIs"). An API for an application may comprise a set of API calls that a device may use to interact with the application. Each API call may be associated with a set of functionality and/or other API calls. Use of an API call may specify how an application should act responsive to receiving the call. A conversation between a device and an application (e.g., an application executing on a separate device) may comprise a plurality of API calls sent by the device to the application and a corresponding plurality of API responses sent by the application to the device.

[0013] A provider device may provide the application for access by a user via a consumer device. As such API communication may occur between the provider device and the consumer device. A technical challenge may exist in this API communication, as the consumer device may have no knowledge of whether a security breach occurs in the communication, and may be reliant on the provider device to detect breaches and notify the user of the consumer device. Further, the only authentication provided by a user of the consumer device may comprise, for example, authentication via user name and password, authentication tokens, and/or other similar discrete authentication methods.

[0014] Examples disclosed herein provide technical solutions to these technical challenges. An example computing device provides secure Application Programming Interface (API) communication. The example

computing device may comprise an API communication engine that communicates with an application executing on a second computing device. The API communication engine may communicate with the application by sending an API call to the second device, the API call comprising a dynamic device access token and first meter information, the first meter information related to first usage of the application. The API communication engine may then receive, from the second computing device, an API response comprising a dynamic first device access token and second meter information, the second meter information related to second usage of the application. The API communication engine may determine whether the communication between the first device and the second device is secure based on a comparison of the API call and the API response.

[0015] The terminology used herein is for the purpose of describing particular embodiments only and is not intended to be limiting. As used herein, the singular forms "a," "an," and "the" are intended to include the plural forms as well, unless the context clearly indicates otherwise. The term "plurality," as used herein, is defined as two or more than two. The term "another," as used herein, is defined as at least a second or more. The term "coupled," as used herein, is defined as connected, whether directly without any intervening elements or indirectly with at least one intervening elements, unless otherwise indicated. Two elements can be coupled mechanically, electrically, or communicatively linked through a communication channel, pathway, network, or system. The term "and/or" as used herein refers to and encompasses any and all possible combinations of one or more of the associated listed items. It will also be understood that, although the terms first, second, third, etc. may be used herein to describe various elements, these elements should not be limited by these terms, as these terms are only used to distinguish one element from another unless stated otherwise or the context indicates otherwise. As used herein, the term "includes" means includes but not limited to, the term "including" means including but not limited to. The term "based on" means based at least in part on.

[0016] FIG. 1 is an example first computing device 100 that provides secure API communication. The first computing device 100 may be any type of computing device providing a user interface through which a user can interact

with a software application and communicate with a second computing device. For example, first computing device 100 may include a laptop computing device, a desktop computing device, an all-in-one computing device, a tablet computing device, a mobile phone, an electronic book reader, a network-enabled appliance such as a "Smart" television, a server, and/or other electronic device suitable for processing user interactions with an application.

[0017] The various components (e.g., components 110 and 120) depicted in FIG. 1 may be coupled to at least one other component via a network. According to various implementations, the API communication engine 120 and the various components described herein may be implemented in hardware and/or a combination of hardware and programming that configures hardware. The term "engine", as used herein, refers to a combination of hardware and programming that performs a designated function. The hardware of each engine, for example, may include one or both of a processor (e.g., processor 110) and a machine-readable storage medium, while the programming is instructions or code stored on the machine-readable storage medium and executable by the processor to perform the designated function. Furthermore, in FIG. 1 and other Figures described herein, different numbers of components, engines, or entities than depicted may be used.

[0018] API communication engine 120 may facilitate secure communication between the first device 100 and a second device. For example, the API communication engine 120 may facilitate secure communication between the first device 100 and an application executing on the second device. The API communication engine 120 may facilitate authentication and initial communication between the first device 100 and the second device, and may then provide for continued secure communication between the first device 100 and the second device. The API communication engine 120 may provide for secure communication by determining whether a security breach has occurred at the first device 100 and/or at the second device.

[0019] The API communication engine 120 may comprise a set of handlers that may interrupt communication from the first device 100 to the second device. The API communication engine 120 may use the set of handlers to interrupt the communication to insert an API call and/or an API response into the communication, to determine whether a security breach exists based on

communication between the first device 100 and the second device, and/or otherwise facilitate provision of secure communication between the first device 100 and the second device. As such, functionality performed by the API communication engine 120 as described herein may be performed responsive to the set of handlers interrupting the communication between the first device and the second device.

[0020] API communication engine 120 may facilitate authentication and initial communication between the first device 100 and the second device. For example, the API communication engine 120 may require an initial authentication of a user using the first device 100 based on a username and password, an authentication token, biometric authentication, any combination thereof, and/or other authentication techniques.

[0021] Responsive to the initial authentication being verified, the API communication engine 120 may send an initial API call from the first device 100 to the second device. Each API call (including an initial API call and subsequent API calls) may comprise, for example, a first dynamic access token related to the first device, metering information related to usage of the application, a second dynamic access token related to the second device, and a security label indicating a security status of the first device.

[0022] The initial API call may comprise a first dynamic access token generated by the first device 100, a security label indicating that the communication is new, and/or other information. The initial API call may have null or empty values for the metering information and the second dynamic access token. In some examples, the initial API call may also comprise a user authentication credential. The API communication engine 120 may store the first dynamic access token for later verification of an incoming initial API response from the second device.

[0023] The second device may authenticate the user (e.g., based on the user authentication credential).

[0024] Responsive to the second device not authenticating the user (e.g., based on the user authentication credential not being authorized), the second device may indicate that a breach has occurred. For example, the second device may indicate that a breach has occurred by facilitating provision of an alert to the user to notify the user of a security breach and/or force the user to

perform authentication again. The alert may be provided to the user via a communication method separate from the communication method between the first device 100 and the second device (e.g., via email, text, and/or other separate communication method). In some examples, the alert may be provided to a separate device of the user. The alert may provide information which may be needed for the authentication of the user. For example, the alert may comprise a token, information, and/or other data that may need to be entered at the first device 100 and verified by the first device 100 and/or the second device in order for the authentication to be validated. Responsive to the authentication being validated, the API communication engine 120 may again facilitate the initial communications between the first device 100 and the second device.

[0025] Responsive to authenticating the user, the second device may respond to the initial API call by saving the first dynamic access token received in the initial API call and by sending the first device an initial API response. Each API response (including an initial API response and subsequent API response) may comprise, for example, a first dynamic access token related to the first device, metering information related to usage of the application, a second dynamic access token related to the second device, and a security label indicating a security status of the second device. The initial API response may comprise first metering information, a second dynamic access token, and a security label. The initial API response may comprise a null or empty value for the first dynamic access token. The second device may store the second dynamic access token for later verification of an incoming API call from the first device.

[0026] Responsive to receiving the initial API response, the API communication engine 120 of the first device 100 may store the metering information, the second dynamic access token, and/or other information from the initial API response.

[0027] Responsive to receiving the initial API response and storing the relevant information, the API communication engine 120 may engage in continued communication with the second device. The API communication engine 120 may generate a first API call and send the first API call to the second device. As mentioned above, each API call may comprise a plurality of fields

including, for example, a first dynamic access token related to the first device, metering information related to usage of the application, a second dynamic access token related to the second device, and a security label indicating a security status of the first device.

[0028] An API call used in communication after the initial communication between the first device 100 and the second device may comprise values for each field of the API call. The API communication engine 120 may generate the first dynamic access token for the API call. The API communication engine 120 may generate the first dynamic access token using a random generator, based on information associated with the first device 100, a combination thereof, and/or by other token generation methods.

[0029] The API communication engine 120 may determine metering information related to usage of the application based on metering information received from a previous API response (e.g., from the initial API response or the immediately previously received API response). The metering information of the API call may comprise, for example, information related to a total amount of data communicated between the first device 100 and the second device, an amount of data communicated during the immediately prior API call and API response, a total amount of communications between the first device 100 and the second device, and/or any other measurement of the usage of the application. In some examples, the API communication engine 120 may also determine the metering information based on metering information stored from one or more previously received API responses.

[0030] The API communication engine 120 may select the previous second dynamic access token from the immediately previously received API response as the second dynamic access token for the API call.

[0031] The API communication engine 120 may determine a security label for the API call based on a determination of whether a security breach occurred with the immediately previously received API response. Examples of security labels include “new,” “no breach,” “breach,”, and/or other security labels that indicate a security status of the first device 100.

[0032] Responsive to receiving the API call from the first device 100, the second device may determine whether a security breach occurred regarding the received API call. The second device may determine whether a security breach

occurred by comparing the second dynamic access token of the received API call with the second dynamic access token generated for the previously sent API response from the second device. In some examples, the second device may store a set of second dynamic access tokens generated for a set of previously sent API responses. The second device may determine whether the security breach occurred by comparing the second dynamic access token of the received API call with the set of second dynamic access tokens. The set of second dynamic access tokens may comprise a predetermined number of second dynamic access tokens. The predetermined number of second dynamic access tokens may depend upon a latency of communication between the first device 100 and the second device.

[0033] Responsive to the received second dynamic access token not matching the previously generated second dynamic access token (or any of the set of second dynamic access tokens), the second device may generate an alert indicating that a security breach has occurred and/or may force authentication of the user of the first device 100. The alert may be generated and used in a manner the same as or similar to the alert generated responsive to failed authentication at the first device. Authentication may be forced in a manner the same as or similar to that described above as well.

[0034] Responsive to the received second dynamic access token matching the previously generated second dynamic access token (or any of the set of second dynamic access tokens), the second device may determine whether the metering information received in the API call corresponds to metering information of the second device (e.g., metering information stored at the second device, metering information included in the immediately previously sent API response, metering information related to current usage of the application executing on the second device, and/or some other measurement of usage of the application). The received metering information may correspond to the metering information of the second device based on calculations performed based on the received metering information and the metering information of the second device. For example, the second device may determine that the metering information corresponds responsive to an addition of an amount of usage of the application since the immediately previously sent API response

and the received metering information equal the current amount of usage of the application.

[0035] Responsive to the received metering information not corresponding to the metering information of the second device, the second device may generate an alert indicating that a security breach has occurred and/or may force authentication of the user of the first device 100. Again, the alert may be generated used in a manner the same as or similar to the alert generated responsive to failed authentication at the first device and authentication may be forced in a manner the same as or similar to that described above.

[0036] Responsive to the received metering information corresponding to the metering information of the second device, the second device may determine that no security breach has occurred.

[0037] Responsive to determining that no security breach has occurred, the second device may then generate and send a first API response to the first device 100. The second device may also store the first dynamic access token of the API call. As mentioned above, each API response may comprise, for example, a first dynamic access token related to the first device, metering information related to usage of the application, a second dynamic access token related to the second device, and a security label indicating a security status of the first device.

[0038] An API response used in communication after the initial communication between the first device 100 and the second device may comprise values for each field of the API response. The second device may select the previous first dynamic access token from the immediately previously received API call as the first dynamic access token for the API response.

[0039] The second device may determine metering information related to usage of the application based on metering information received from a previous API response (e.g., from the initial API response or the immediately previously received API response), metering information stored at the second device, and/or may otherwise determine metering information. The metering information of the API response may comprise, for example, information related to a total amount of data communicated between the first device 100 and the second device, an amount of data communicated during the immediately prior API call and API response, a total amount of communications between the first

device 100 and the second device, and/or any other measurement of the usage of the application.

[0040] The second device may generate the second dynamic access token for the API response. The second device may generate the second dynamic access token using a random generator, based on information associated with the second device, a combination thereof, and/or by other token generation methods.

[0041] The second device may determine a security label for the API response based on a determination of whether a security breach occurred with the immediately previously received API call. Examples of security labels include "new," "no breach," "breach,", and/or other security labels that indicate a security status of the second device.

[0042] The second device may send the generated API response to the first device 100. Responsive to receiving the generated API response from the second device, the API communication engine 120 of the first device 100 may determine whether a security breach occurred regarding the received API response. The API communication engine 120 may determine whether a security breach occurred by comparing the first dynamic access token of the received API response with the first dynamic access token generated for the previously sent API call from the first device 100. In some examples, the API communication engine 120 may store a set of first dynamic access tokens generated for a set of previously sent API calls. The API communication engine 120 may determine whether the security breach occurred by comparing the first dynamic access token of the received API call with the set of first dynamic access tokens. The set of first dynamic access tokens may comprise a predetermined number of first dynamic access tokens. The predetermined number of first dynamic access tokens may depend upon a latency of communication between the first device 100 and the second device.

[0043] Responsive to the received first dynamic access token not matching the previously generated first dynamic access token (or any of the set of first dynamic access tokens), the API communication engine 120 may generate an alert indicating that a security breach has occurred and/or may force authentication of the user of the first device 100. The alert may be generated and used in a manner the same as or similar to the alert generated responsive

to failed authentication at the first device 100. Authentication may be forced in a manner the same as or similar to that described above as well.

[0044] Responsive to the received first dynamic access token matching the previously generated first dynamic access token (or any of the set of first dynamic access tokens), the API communication engine 120 may determine whether the metering information received in the API response corresponds to metering information of the first device 100 (e.g., metering information stored at the first device 100, metering information included in the immediately previously sent API call, metering information related to current usage of the application executing on the second device, and/or some other measurement of usage of the application). The received metering information may correspond to the metering information of the first device 100 based on calculations performed based on the received metering information and the metering information of the first device 100. For example, the API communication engine 120 may determine that the metering information corresponds responsive to an addition of an amount of usage of the application since the immediately previously sent API call and the received metering information equal the current amount of usage of the application.

[0045] Responsive to the received metering information not corresponding to the metering information of the first device 100, the API communication engine 120 may generate an alert indicating that a security breach has occurred and/or may force authentication of the user of the first device 100. Again, the alert may be generated and used in a manner the same as or similar to the alert generated responsive to failed authentication at the first device and authentication may be forced in a manner the same as or similar to that described above as well.

[0046] Responsive to the received metering information corresponding to the metering information of the first device 100, the API communication engine 120 may determine that no security breach has occurred.

[0047] The first device 100 and the second device may continue to engage in secured API communication in the manner described above.

[0048] In performing their respective functions, engine 120 may access data stored in a non-transitory machine-readable storage medium of first device 100. The non-transitory machine-readable storage medium of first device 100 may

represent any memory accessible to first device 100 that can be used to store and retrieve data. The non-transitory machine-readable storage medium of first device 100 and/or other database may comprise random access memory (RAM), read-only memory (ROM), electrically-erasable programmable read-only memory (EEPROM), cache memory, floppy disks, hard disks, optical disks, tapes, solid state drives, flash drives, portable compact disks, and/or other storage media for storing computer-executable instructions and/or data. In some examples, the first device 100 may access non-transitory machine-readable storage medium locally or remotely via a network.

[0049] The non-transitory machine-readable storage medium of first device 100 may include a database to organize and store data. The non-transitory machine-readable storage medium of first device 100 may be, include, or interface to, for example, a relational database. The database may reside in a single or multiple physical device(s) and in a single or multiple physical location(s). The database may store a plurality of types of data and/or files and associated data or file description, administrative information, or any other data.

[0050] In some examples, the non-transitory machine-readable storage medium of first device 100 may be any electronic, magnetic, optical, or other physical storage device that contains or stores executable instructions. In some implementations, the non-transitory machine-readable storage medium of first device 100 may be a non-transitory storage medium, where the term "non-transitory" does not encompass transitory propagating signals. The non-transitory machine-readable storage medium of first device 100 may be implemented in a single device or distributed across devices. Likewise, processor 110 may represent any number of processors capable of executing instructions stored by the machine-readable storage medium. The processor 110 may be integrated in a single device or distributed across devices. Further, the machine-readable storage medium may be fully or partially integrated in the same device as the processor 110, or it may be separate but accessible to that device and the processor 110.

[0051] In one example, the machine-readable storage medium may comprise instructions that may be part of an installation package that when installed can be executed by processor 110 to implement the functionality described herein. For example, the machine-readable storage medium may be

a portable medium such as a floppy disk, CD, DVD, or flash drive or a memory maintained by a server from which the installation package can be downloaded and installed. In another example, the program instructions may be part of an application or applications already installed. Here, the machine-readable storage medium may include a hard disk, optical disk, tapes, solid state drives, RAM, ROM, EEPROM, or the like.

[0052] The processor 110 may be at least one central processing unit (CPU), microprocessor, and/or other hardware device suitable for retrieval and execution of instructions stored in the machine-readable storage medium. The processor 110 may fetch, decode, and execute program instructions. As an alternative or in addition to retrieving and executing instructions, the processor 110 may include at least one electronic circuit comprising a number of electronic components for performing the functionality of at least one of instructions.

[0053] FIG. 2 is an example first computing device 100 and a second computing device 200 that each provide secure API communication. In this example, the first computing device 100 may comprise a consumer device that accesses an application executing on second computing device 200, which may act as a provider device. The first computing device 100 may be the same as or similar to the first device 100 of FIG. 1. The second computing device 200 may comprise an API communication engine 220 that performs functionality the same as or similar to the functionality performed by the second device described in FIG. 1.

[0054] The first computing device 100 may communicate with second computing device 200 over a network 50. In some examples, the first computing device 100 may be any type of computing device providing a user interface through which a user can interact with a software application and communicate with a second device. For example, first computing device 100 may include a laptop computing device, a desktop computing device, an all-in-one computing device, a tablet computing device, a mobile phone, an electronic book reader, a network-enabled appliance such as a "Smart" television, and/or other electronic device suitable for processing user interactions with the second device 200.

[0055] The second computing device 200 may similarly comprise a laptop computing device, a desktop computing device, an all-in-one computing device,

a tablet computing device, a mobile phone, an electronic book reader, a network-enabled appliance such as a "Smart" television, server, and/or other electronic device suitable for processing user interactions with the second device 200. While second computing device 200 is depicted as a single computing device, second computing device 200 may include any number of integrated or distributed computing devices serving at least one software application for consumption by first computing device 100.

[0056] Similar to the components of first device 100, the various components (e.g., components 210 and 220) depicted in second device 200 of FIG. 2 may be coupled to at least one other component via a network (e.g., network 50). According to various implementations, the API communication engine 220 and the various components described herein may be implemented in hardware and/or a combination of hardware and programming that configures hardware. As mentioned above, the term "engine", as used herein, refers to a combination of hardware and programming that performs a designated function. The hardware of each engine, for example, may include one or both of a processor (e.g., processor 210) and a machine-readable storage medium, while the programming is instructions or code stored on the machine-readable storage medium and executable by the processor to perform the designated function. The processor 210 and the machine-readable storage medium of second device 200 may comprise components and functionality the same as or similar to the respective processor 110 and machine-readable storage of the respective processor 110 and machine-readable storage medium of the first device 100.

[0057] API communication engine 220 may facilitate secure communication between the first device 100 and a second device. For example, the API communication engine 220 may facilitate secure communication between the first device 100 and an application executing on the second device 200. The API communication engine 220 may facilitate authentication and initial communication between the first device 100 and the second device 200, and may then provide for continued secure communication between the first device 100 and the second device 200. The API communication engine 220 may provide for secure communication by determining whether a security breach has occurred at the first device 100 and/or at the second device.

[0058] FIGS. 3A and 3B are flow diagrams of an example method for execution by an example consumer device and provider device engaging in secure API communication. Although execution of the methods described below are with reference to system described in FIG. 2, other suitable devices for execution of this method may be employed to practice the present techniques. The flow diagram described in FIG. 3A and other figures may be implemented in the form of executable instructions stored on a machine-readable storage medium, such as the storage medium of the hypervisor, by one or more components described herein, and/or in the form of electronic circuitry.

[0059] The various processing blocks and/or data flows depicted in FIG. 3A (and in the other drawing figures such as FIG. 4) are described in greater detail herein. The described processing blocks may be accomplished using some or all of the system components described in detail above and, in some implementations, various processing blocks may be performed in different sequences and various processing blocks may be omitted. Additional processing blocks may be performed along with some or all of the processing blocks shown in the depicted flow diagrams. Some processing blocks may be performed simultaneously. Accordingly, the operations depicted in the flow diagram as illustrated (and described in greater detail below) are meant be an example and, as such, should not be viewed as limiting.

[0060] In an operation 300, the first device may send a first API call to the second device. For example, the first device 100 (and/or the API communication engine 120, or other resource of the first device 100) may send the first API call. The first device may send the first API call in a manner similar or the same as that described above in relation to the execution of the API communication engine 120, and/or other resource of the first device 100. The first API call may be depicted as the request of the first API call in FIG. 3B. In some examples, and as depicted in FIG. 3B, an initial communication may occur between the first device and the second device prior to the first device sending a first API call to the second device. The initial communication, as depicted in FIG. 3B, may include initialization with a user name and password and/or other type of authentication from the user.

[0061] In an operation 310, the second device may send a first API response to the first device. For example, the second device 100 (and/or the API communication engine 220, or other resource of the second device 100) may send the first API response. The second device may send the first API response in a manner similar or the same as that described above in relation to the execution of the API communication engine 220, and/or other resource of the second device 100. The first API response may be depicted as the response of the first API call in FIG. 3B.

[0062] In an operation 320, the first device may determine whether communication between the first device and the second device is secure. For example, the first device 100 (and/or the API communication engine 120, or other resource of the first device 100) may determine whether communication between the first device and the second device is secure. The first device may whether communication between the first device and the second device is secure in a manner similar or the same as that described above in relation to the execution of the API communication engine 120, and/or other resource of the first device 100.

[0063] In an operation 330, the first device may send a second API call to the second device. For example, the first device 100 (and/or the API communication engine 120, or other resource of the first device 100) may send the second API call. The first device may send the second API call in a manner similar or the same as that described above in relation to the execution of the API communication engine 120, and/or other resource of the first device 100. The second API call may be depicted as the request of the second API call in FIG. 3B.

[0064] In an operation 340, the second device may determine whether communication between the first device and the second device is secure. For example, the second device 200 (and/or the API communication engine 220, or other resource of the second device 200) may determine whether communication between the first device and the second device is secure. The second device may whether communication between the first device and the second device is secure in a manner similar or the same as that described above in relation to the execution of API communication engine 220, and/or other resource of the second device 200.

[0065] FIG. 4 is a flowchart of an example method for execution by an example consumer device and provider device engaging in secure API communication.

[0066] In an operation 400, the first device may send an API call to the second device. For example, the first device 100 (and/or the API communication engine 120, or other resource of the first device 100) may send the first API call. The first device may send the first API call in a manner similar or the same as that described above in relation to the execution of API communication engine 120, and/or other resource of the first device 100.

[0067] In an operation 410, the first device may receive, from the second device, a first API response to the first device. For example, the first device 100 (and/or the API communication engine 120, or other resource of the second device 100) may receive the API response. The first device may receive the API response in a manner similar or the same as that described above in relation to the execution of API communication engine 220, and/or other resource of the first device 100.

[0068] In an operation 420, the first device may determine whether communication between the first device and the second device is secure. For example, the first device 100 (and/or the API communication engine 120, or other resource of the first device 100) may determine whether communication between the first device and the second device is secure. The first device may whether communication between the first device and the second device is secure in a manner similar or the same as that described above in relation to the execution of API communication engine 120, and/or other resource of the first device 100.

[0069] In some examples, the determination of whether communication between the first device and the second device is secure may be performed in various manners. FIG. 4A is a flowchart of an example method for execution by an example consumer device and provider device engaging in secure API communication.

[0070] In an operation 421, the first device may determine whether the first dynamic device access token of the API call matches the first dynamic access token of the API response. For example, the first device 100 (and/or the API communication engine 120, or other resource of the first device 100) may

determine whether the first dynamic device access token of the API call matches the first dynamic access token of the API response. The first device may determine whether the first dynamic device access token of the API call matches the first dynamic access token of the API response in a manner similar or the same as that described above in relation to the execution of API communication engine 120, and/or other resource of the first device 100.

[0071] Responsive to the tokens not matching, in an operation 422, the first device may indicate that a security breach has occurred. For example, the first device (and/or the API communication engine 120, or other resource of the second device 100) may indicate that a breach has occurred by providing an alert. The first device may indicate that a breach has occurred by providing an alert in a manner similar or the same as that described above in relation to the execution of API communication engine 120, and/or other resource of the first device 100.

[0072] Responsive to the tokens not matching, in an operation 423, the first device may require authentication of the user. For example, the first device 100 (and/or the API communication engine 120, or other resource of the first device 100) may require authentication. The first device may require authentication in a manner similar or the same as that described above in relation to the execution of API communication engine 120, and/or other resource of the first device 100.

[0073] Responsive to the tokens matching, in an operation 424, the first device may determine whether the first meter information of the API call corresponds to the second meter information of the API response. For example, the first device 100 (and/or the API communication engine 120, or other resource of the first device 100) may determine whether the first meter information corresponds to the second meter information. The first device may determine whether the first meter information corresponds to the second meter information in a manner similar or the same as that described above in relation to the execution of API communication engine 120, and/or other resource of the first device 100.

[0074] Responsive to the first meter information not corresponding to the second meter information, in an operation 425, the first device may indicate that a security breach has occurred. For example, the first device (and/or the

API communication engine 120, or other resource of the second device 100) may indicate that a breach has occurred by providing an alert. The first device may indicate that a breach has occurred by providing an alert in a manner similar or the same as that described above in relation to the execution of API communication engine 120, and/or other resource of the first device 100.

[0075] Responsive to the first meter information not corresponding to the second meter information, in an operation 426, the first device may require authentication of the user. For example, the first device 100 (and/or the API communication engine 120, or other resource of the first device 100) may require authentication. The first device may require authentication in a manner similar or the same as that described above in relation to the execution of API communication engine 120, and/or other resource of the first device 100.

[0076] Responsive to the first meter information corresponding to the second meter information, in an operation 427, communication between the first device and the second device may continue. For example, the first device 100 (and/or the API communication engine 120, or other resource of the first device 100) may continue communicating with the second device 200 (and/or the API communication engine 220 or other resource of the second device 200). The first device may communicate with the second device in a manner similar or the same as that described above in relation to the execution of API communication engine 120, API communication engine 220, another resource of the first device 200 and/or other resource of the second device 200.

[0077] FIG. 5 is a block diagram of an example device engaging in secure API communication. In the example depicted in FIG. 5, first device 500 includes a non-transitory machine-readable storage medium 520 and a processor 510.

[0078] Processor 510 may be one or more central processing units (CPUs), microprocessors, and/or other hardware devices suitable for retrieval and execution of instructions stored in machine-readable storage medium 520.

[0079] Processor 510 may fetch, decode, and execute program instructions 521, and/or other instructions to provide secure API communication, as described below. As an alternative or in addition to retrieving and executing instructions, processor 510 may include one or more electronic circuits comprising a number of electronic components for performing the functionality of one or more of program instructions 521, and/or other instructions.

[0080] In one example, the program instructions can be part of an installation package that can be executed by processor 510 to implement the functionality described herein. In this case, machine-readable storage medium 520 may be a portable medium such as a CD, DVD, or flash drive or a memory maintained by another backup storage device from which the installation package can be downloaded and installed. In another example, the program instructions may be part of an application or applications already installed on backup storage device 500.

[0081] Machine-readable storage medium 520 may be any hardware storage device for maintaining data accessible to first device 500. For example, machine-readable storage medium 520 may include one or more hard disk drives, solid state drives, tape drives, and/or any other storage devices. The storage devices may be located in first device 500 and/or in another device in communication with first device 500. For example, machine-readable storage medium 520 may be any electronic, magnetic, optical, or other physical storage device that stores executable instructions. Thus, machine-readable storage medium 520 may be, for example, Random Access Memory (RAM), an Electrically-Erasable Programmable Read-Only Memory (EEPROM), a storage drive, an optical disc, and the like. As detailed below, storage medium 520 may maintain and/or store the data and information described herein.

[0082] Machine-readable storage medium 520 may also be encoded with executable instructions for enabling execution of the functionality described herein. For example, machine-readable storage medium 520 may store API communication instructions 521, and/or other instructions that may be used to carry out the functionality of the herein disclosed present techniques.

[0083] API communication instructions 521, when executed by processor 510, may communicate with an application executing on a second device. For example, API communication instructions 521 may comprise instructions to send an API call to the second device, the API call comprising a dynamic device access token, first meter information related to first usage of the application, and a second device dynamic access token. API communication instructions 521 may also comprise instructions to receive, from the second computing device, an API response comprising a dynamic first device access token, second meter information related to second usage of the application,

and a second dynamic access token. API communication instructions 521 may further comprise instructions to determine whether the communication between the first device and the second device is secure based on a comparison of the dynamic device access token of the API call and the dynamic first device access token of the API response. In some examples, the functionality performed by the API communication instructions 521, when executed by processor 510, may be the same as or similar to functionality performed by API communication engine 120 of first device 100.

[0084] FIG. 6 is a block diagram of an example device engaging in secure API communication. In the example depicted in FIG. 6, second device 600 includes a non-transitory machine-readable storage medium 620 and a processor 610. Processor 610, similar to processor 510, may be one or more central processing units (CPUs), microprocessors, and/or other hardware devices suitable for retrieval and execution of instructions stored in machine-readable storage medium 620. Machine-readable storage medium 620 may comprise hardware, functionality, and data stored thereon that is the same as or similar to machine-readable storage medium 620. Machine-readable storage medium 620 may also be encoded with executable instructions for enabling execution of the functionality described herein. For example, machine-readable storage medium 620 may store API communication instructions 621, and/or other instructions that may be used to carry out the functionality of the herein disclosed present techniques. Machine-readable storage medium 620 may also store metering information 622. Metering information 622 may comprise the metering information stored at second device 200 as discussed above.

[0085] API communication instructions 621, when executed by processor 610, may communicate with a first device. For example, API communication instructions 621 may comprise instructions to send an API response to the first device. API communication instructions 621 may also comprise instructions to receive, from the second computing device, an API call. API communication instructions 621 may further comprise instructions to determine whether the communication between the first device and the second device is secure based on a comparison of the API call and the API response. In some examples, the functionality performed by the API communication instructions

621, when executed by processor 610, may be the same as or similar to functionality performed by API communication engine 220 of first device 200.

[0086] The foregoing disclosure describes a number of examples for providing secure API communication. The disclosed examples may include systems, devices, computer-readable storage media, and methods for providing secure API communication. For purposes of explanation, certain examples are described with reference to the components illustrated in FIGS. 1-6. The functionality of the illustrated components may overlap, however, and may be present in a fewer or greater number of elements and components. Further, all or part of the functionality of illustrated elements may co-exist or be distributed among several geographically dispersed locations. Further, the disclosed examples may be implemented in various environments and are not limited to the illustrated examples.

[0087] Further, the sequence of operations described in connection with FIGS. 1-6 are examples and are not intended to be limiting. Additional or fewer operations or combinations of operations may be used or may vary without departing from the scope of the disclosed examples. Furthermore, implementations consistent with the disclosed examples need not perform the sequence of operations in any particular order. Thus, the present disclosure merely sets forth possible examples of implementations, and many variations and modifications may be made to the described examples.

CLAIMS

We claim:

1. A first computing device to provide secure Application Programming Interface (API) communication, the first computing device comprising:

an API communication engine to communicate with an application executing on a second computing device, wherein the API communication engine to communicate with the application that includes to:

send an API call to the second device, the API call comprising a dynamic device access token and first meter information, the first meter information related to first usage of the application;

receive, from the second computing device, an API response comprising a dynamic first device access token and second meter information, the second meter information related to second usage of the application; and

determine whether the communication between the first device and the second device is secure based on a comparison of the API call and the API response.

2. The first computing device of claim 1, wherein the API communication engine to determine whether the communication is secure that includes to:

determine whether the dynamic device access token of the API call matches the dynamic first device access token of the API response;

responsive to the dynamic device access token not matching the dynamic first device access token, indicate that a security breach has occurred.

3. The first computing device of claim 2, wherein the API communication engine further to determine whether the communication is secure that includes to:

responsive to the dynamic device access token matching the dynamic first device access token, determine whether the first meter information corresponds to the second meter information;

responsive to the first meter information not corresponding to the second meter information, indicate that a security breach has occurred.

4. The first computing device of claim 1, wherein the API communication engine to:

generate a new dynamic device access token subsequent to receiving the API response;

generate a second API call by replacing the dynamic device access token with the new dynamic device access token; and

send the second API call to the second computing device.

5. The first computing device of claim 4, wherein the API communication engine to generate the new dynamic device access token using a random generator.

6. The first computing device of claim 4, wherein the API communication engine to generate the second API call further includes to:

determine updated first meter information; and

replace the first meter information with the determined updated first meter information.

7. A method that provides secure Application Programming Interface (API) communication between a first device and a second device, the method comprising:

sending an API call from the first device to the second device, the API call comprising a dynamic device access token and first meter information, the first meter information related to first usage of an application executing on the second device;

receiving, by the first device and from the second computing device, an API response, the API response comprising a dynamic first device access token, second meter information, and a dynamic second device access token, the second meter information related to second usage of the application; and

determining, by the first device, whether the communication between the first device and the second device is secure based on a comparison of the API call and the API response.

8. The method of claim 7, wherein determining whether the communication is secure comprises:

determining, by the first device, whether the dynamic device access token of the API call matches the dynamic first device access token of the API response;

responsive to the dynamic device access token not matching the dynamic first device access token, indicating, by the first device, that a security breach has occurred.

9. The method of claim 8, further comprising:

responsive to the dynamic device access token matching the dynamic first device access token, determining, by the first device, whether the first meter information corresponds to the second meter information;

responsive to the first meter information not corresponding to the second meter information, indicating, by the first device, that a security breach has occurred.

10. The method of claim 8, further comprising:

responsive to determining that the security breach has occurred, performing authentication of a user of the first device, wherein performing authentication comprises:

sending, to a user device separate from the first device, user authentication information;

receiving, at the first device, authentication information from the user;

determining, by the first device, whether the received authentication information matches the user authentication information;

enabling, by the first device, communication between the first device and the second device responsive to the received authentication information matching the user authentication information.

11. The method of claim 7, further comprising:

generating, by the first device, a new dynamic device access token subsequent to receiving the API response;

generating, by the first device, a second API call by replacing the dynamic device access token with the new dynamic device access token, wherein the second API call comprises the new dynamic device access token, updated first meter information, and the dynamic second device access token; and

sending, by the first device, the second API call to the second computing device.

12. The method of claim 11, further comprising:

receiving, by the second device, the second API call;

determining, by the second device, whether the dynamic second device access token of the API call matches the dynamic second device access token of the API response;

responsive to the dynamic second device access token of the second API call not matching the dynamic second device access token of the API response, indicating, by the second device, that a security breach has occurred.

13. The method of claim 12, further comprising:

responsive to the dynamic second device access token of the second API call matching the dynamic second device access token of the API response, determining, by the second device, whether the updated first meter information corresponds to meter information stored at the second device;

responsive to the updated first meter information not corresponding to the stored meter information, indicating, by the second device, that a security breach has occurred.

14. The method of claim 11,

wherein the API call comprises a first security label, the first security label indicating a first security status of the first device, and

wherein the API response comprises a second security label, the second security label indicating a second security status of the second device.

15. A non-transitory machine-readable storage medium comprising instructions executable by a processor of a computing device to provide secure Application Programming Interface (API) communication, the machine-readable storage medium comprising instructions when executed by the processor to:

communicate with an application executing on a second computing device,

wherein the instructions to communicate with the application comprise instructions to:

send an API call to the second device, the API call comprising a dynamic device access token, first meter information related to first usage of the application, and a second device dynamic access token;

receive, from the second computing device, an API response comprising a dynamic first device access token, second meter information related to second usage of the application, and a second dynamic access token; and

determine whether the communication between the first device and the second device is secure based on a comparison of the dynamic device access token of the API call and the dynamic first device access token of the API response.

FIG. 1

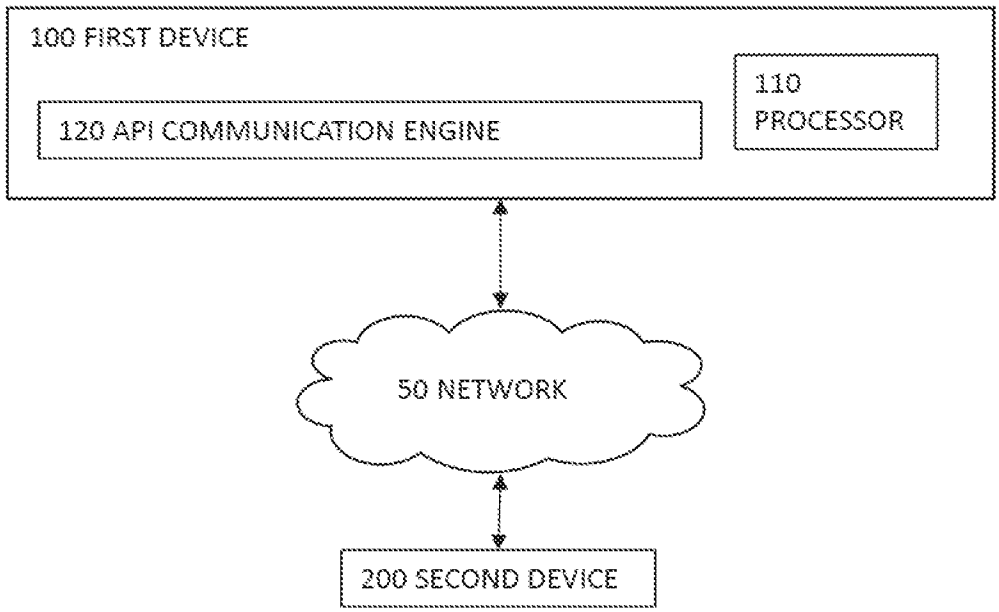

100 FIRST DEVICE
120 API COMMUNICATION ENGINE
110 PROCESSOR
50 NETWORK
200 SECOND DEVICE

FIG. 2

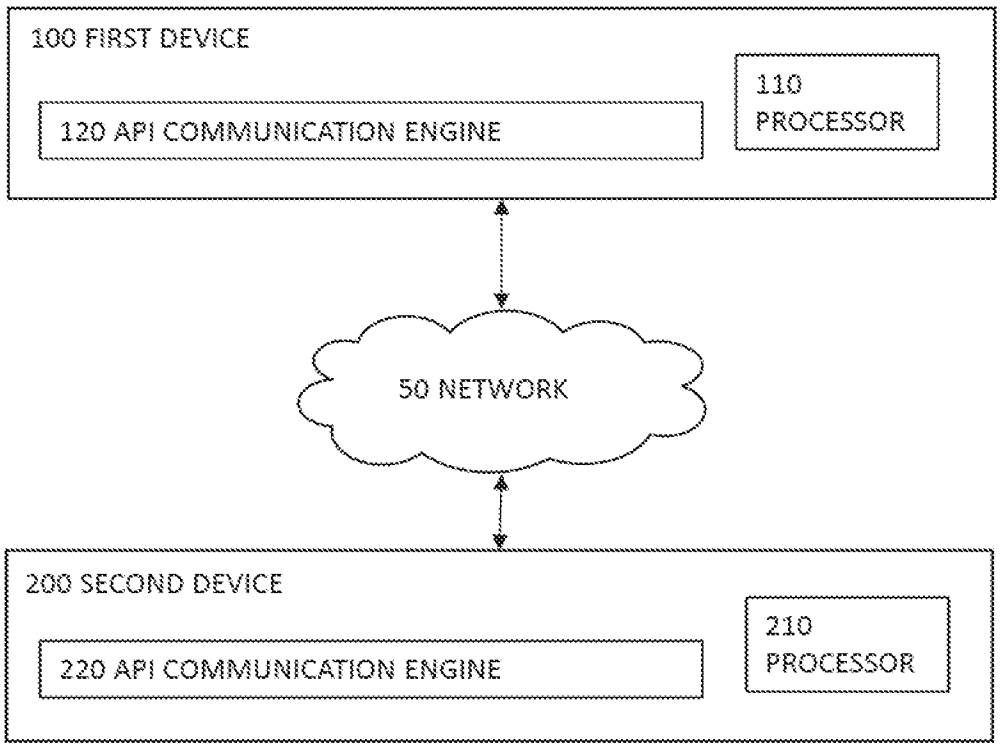

100 FIRST DEVICE
120 API COMMUNICATION ENGINE
110 PROCESSOR
50 NETWORK
200 SECOND DEVICE
220 API COMMUNICATION ENGINE
210 PROCESSOR

FIG. 3A

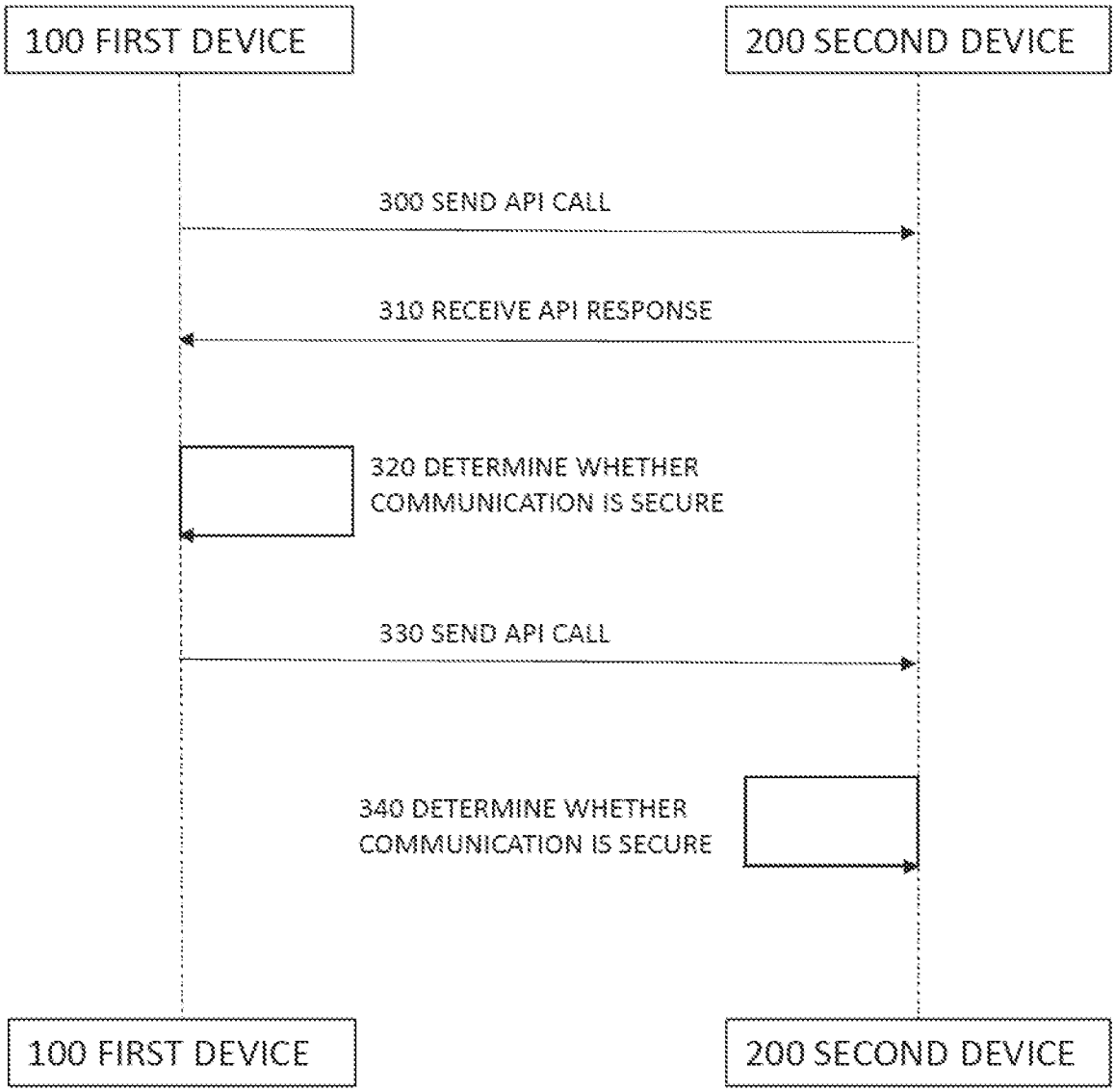

100 FIRST DEVICE
200 SECOND DEVICE
300 SEND API CALL
310 RECEIVE API RESPONSE
320 DETERMINE WHETHER COMMUNICATION IS SECURE
330 SEND API CALL
340 DETERMINE WHETHER COMMUNICATION IS SECURE
100 FIRST DEVICE
200 SECOND DEVICE

FIG. 3B

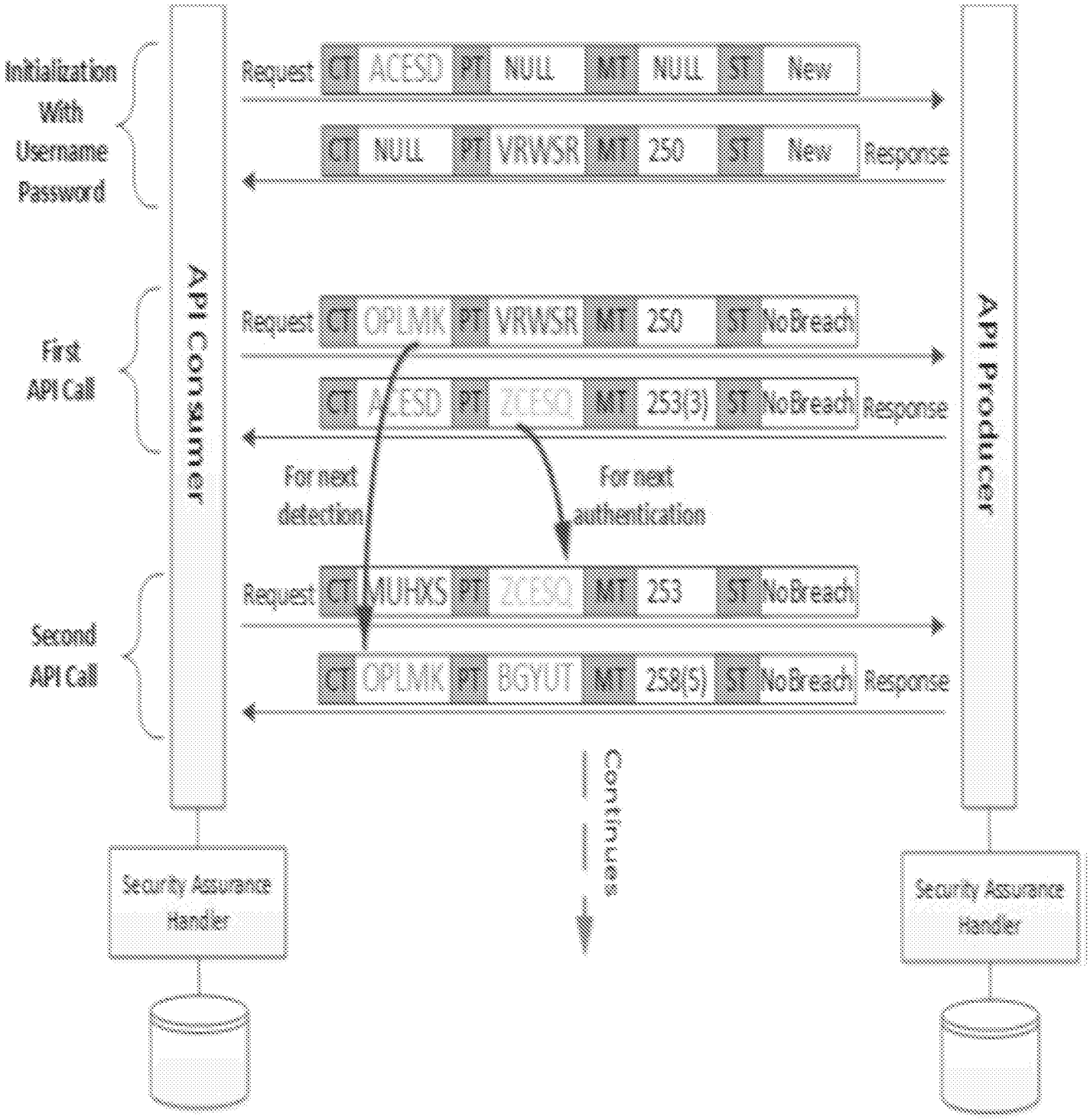

Initialization With Username Password
First API Call
Second API Call
API Consumer
API Producer
Request
CT ACESD PT NULL MT NULL ST New
CT NULL PT VRWSR MT 250 ST New
Response
Request
CT OPLMK PT VRWSR MT 250 ST NoBreach
CT ACESD PT ZCESO MT 253(3) ST NoBreach
Response
For next detection
For next authentication
Request
CT MUHXS PT ZCESO MT 253 ST NoBreach
CT OPLMK PT BGYUT MT 258(5) ST NoBreach
Response
Continues
Security Assurance Handler
Security Assurance Handler

FIG. 3B

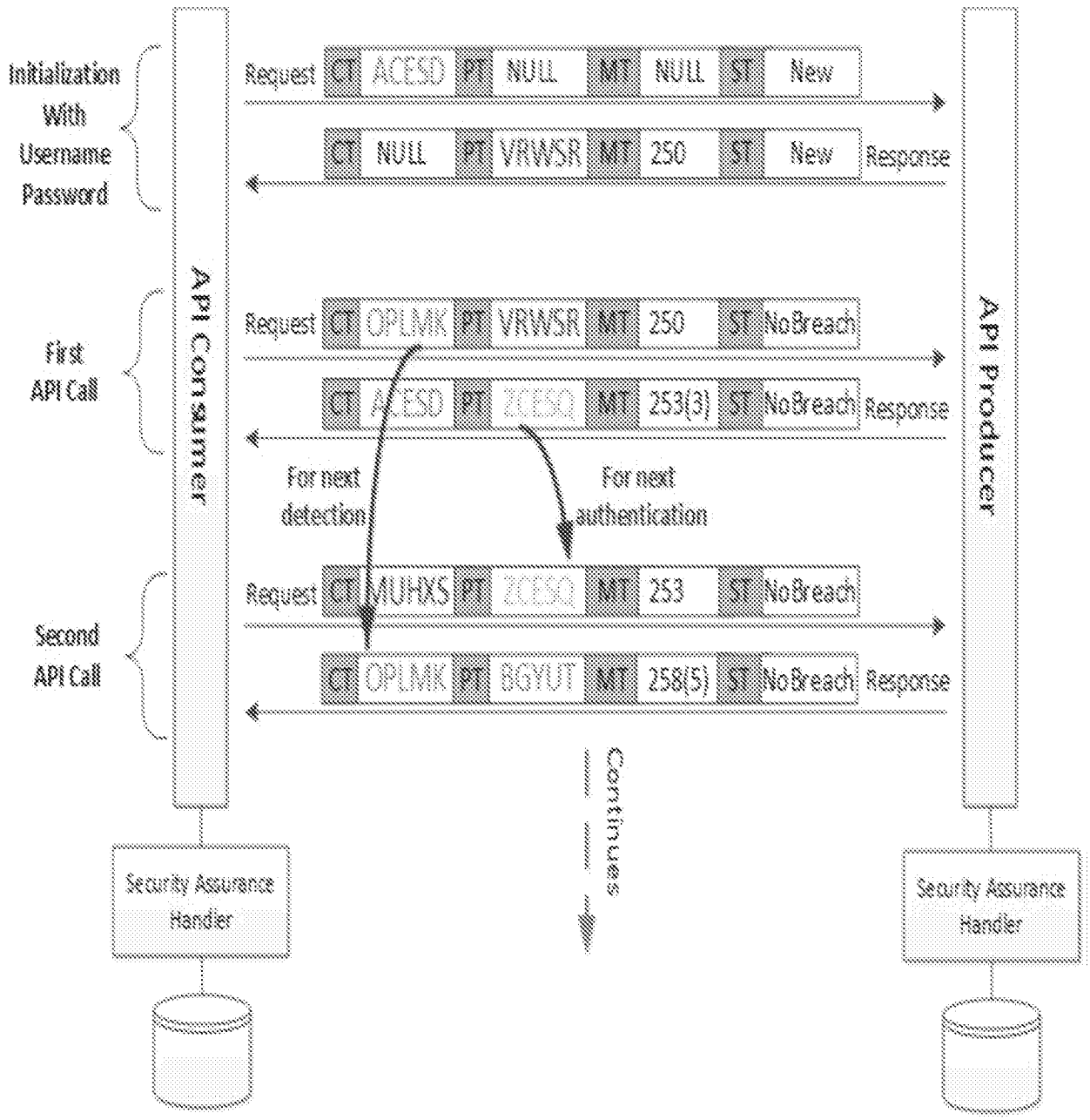

Initialization With Username Password
First API Call
Second API Call
API Consumer
API Producer
Request
CT ACESO PT NULL MT NULL ST New
CT NULL PT VRWSR MT 250 ST New
Response
Request
CT OPLMK PT VRWSR MT 250 ST NoBreach
CT ACESO PT ZCESQ MT 253(3) ST NoBreach
Response
For next detection
For next authentication
Request
CT MUHXS PT ZCESQ MT 253 ST NoBreach
CT OPLMK PT BGYUT MT 258(5) ST NoBreach
Response
Continues
Security Assurance Handler
Security Assurance Handler

FIG. 4

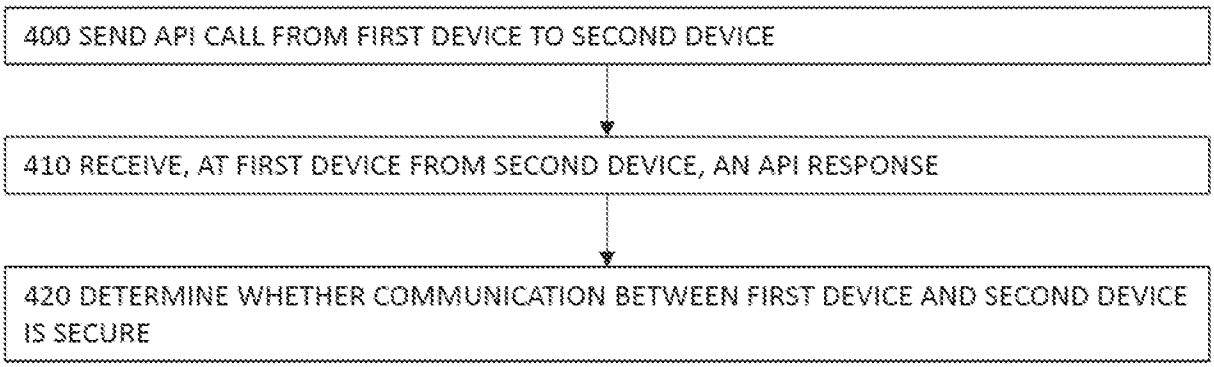

400 SEND API CALL FROM FIRST DEVICE TO SECOND DEVICE
410 RECEIVE, AT FIRST DEVICE FROM SECOND DEVICE, AN API RESPONSE
420 DETERMINE WHETHER COMMUNICATION BETWEEN FIRST DEVICE AND SECOND DEVICE IS SECURE

FIG. 4A

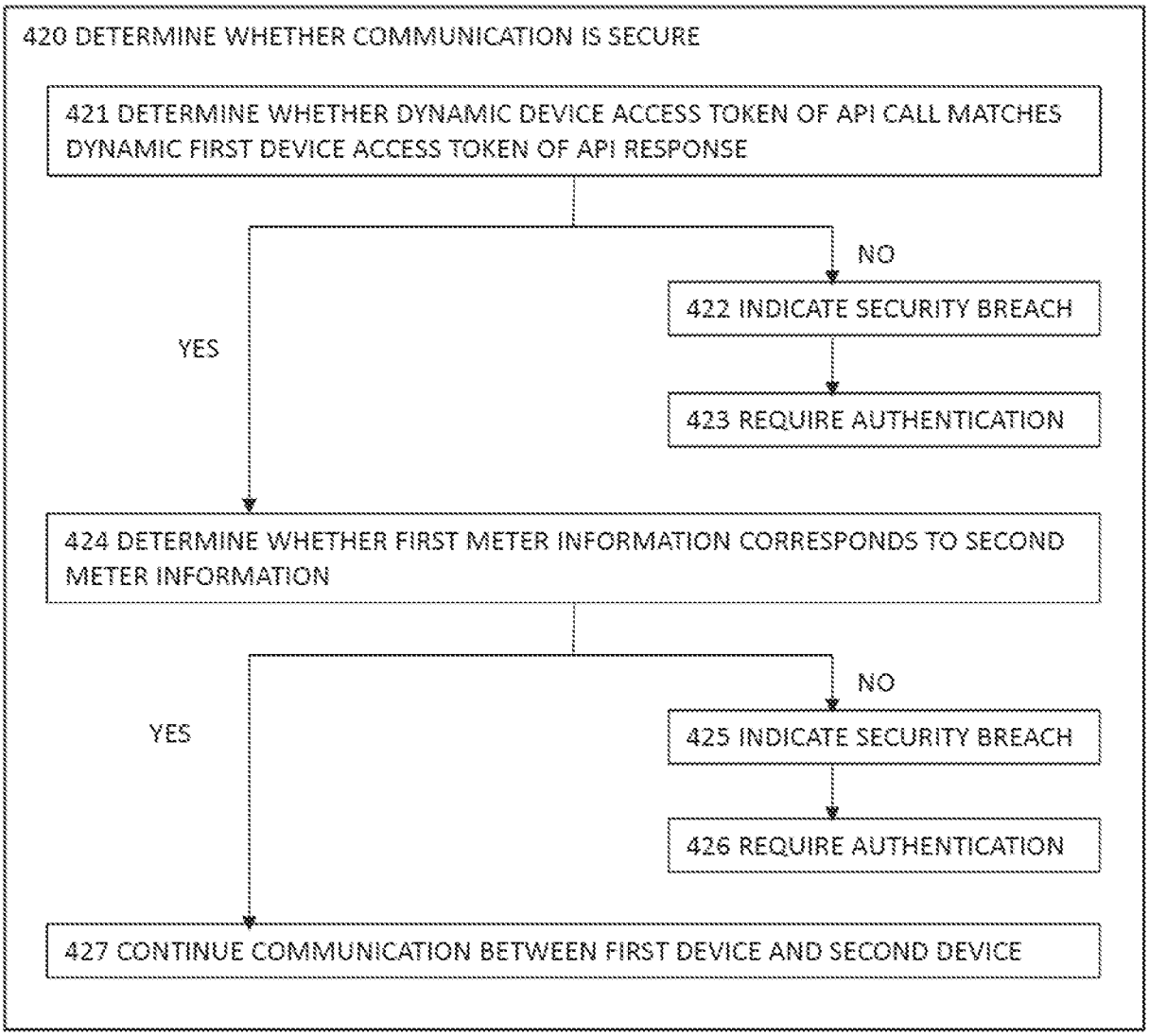

420 DETERMINE WHETHER COMMUNICATION IS SECURE
421 DETERMINE WHETHER DYNAMIC DEVICE ACCESS TOKEN OF API CALL MATCHES DYNAMIC FIRST DEVICE ACCESS TOKEN OF API RESPONSE
NO
422 INDICATE SECURITY BREACH
YES
423 REQUIRE AUTHENTICATION
424 DETERMINE WHETHER FIRST METER INFORMATION CORRESPONDS TO SECOND METER INFORMATION
NO
YES
425 INDICATE SECURITY BREACH
426 REQUIRE AUTHENTICATION
427 CONTINUE COMMUNICATION BETWEEN FIRST DEVICE AND SECOND DEVICE

FIG. 5

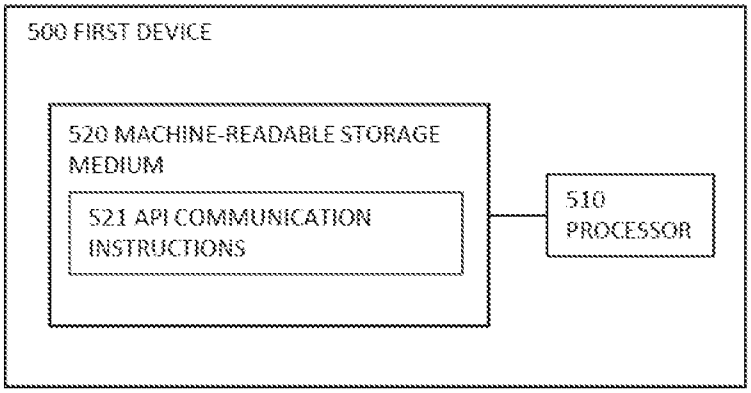
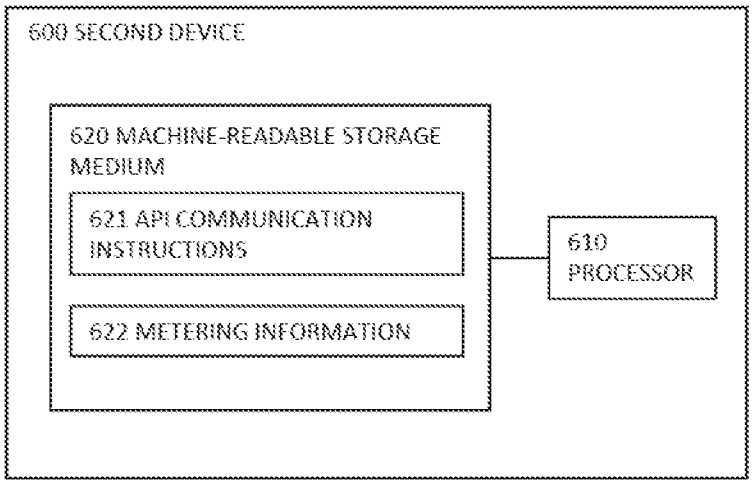

500 FIRST DEVICE
520 MACHINE-READABLE STORAGE MEDIUM
521 API COMMUNICATION INSTRUCTIONS
510 PROCESSOR

FIG. 6


600 SECOND DEVICE
620 MACHINE-READABLE STORAGE MEDIUM
621 API COMMUNICATION INSTRUCTIONS
622 METERING INFORMATION
610 PROCESSOR

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2015/013864

A. CLASSIFICATION OF SUBJECT MATTER

G06F 9/54(2006.01)i, G06F 21/00(2006.01)i

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)
G06F 9/54; G06F 15/16; G06F 17/60; G06F 21/53; H04L 9/00; G06F 15/18; G06F 21/00

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
Korean utility models and applications for utility models
Japanese utility models and applications for utility models

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
eKOMPASS(KIPO internal) & Keywords: API, call, response, secure, communication

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 7016965 B2 (MARK R. GAMBINO) 21 March 2006 See abstract; claims 1, 5; and figures 1, 2.	1,7,15
A		2-6,8-14
A	US 2010-0031308 A1 (ATM SHAFIQUL KHALID) 04 February 2010 See abstract; claims 1, 2, 7; and figure 1.	1-15
A	US 2011-0191593 A1 (PETER CHENG) 04 August 2011 See abstract; and claims 1-3, 16-18.	1-15
A	US 2005-0289072 A1 (VINAY SABHARWAL) 29 December 2005 See abstract; claim 14; and figure 1.	1-15
A	US 2014-0304771 A1 (KRISTOFER HELLICK REIERSON et al.) 09 October 2014 See abstract; claims 1-27; and figures 1, 2.	1-15

☐ 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
30 July 2015 (30.07.2015)	**30 July 2015 (30.07.2015)**

Name and mailing address of the ISA/KR
International Application Division
Korean Intellectual Property Office
189 Cheongsa-ro, Seo-gu, Daejeon Metropolitan City, 302-701,
Republic of Korea
Facsimile No. +82-42-472-7140

Authorized officer
LEE, Seok Hyung
Telephone No. +82-42-481-5983

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.
PCT/US2015/013864

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 7016965 B2	21/03/2006	US 2003-093573 A1	15/05/2003
US 2010-0031308 A1	04/02/2010	US 8286219 B2	09/10/2012
US 2011-0191593 A1	04/08/2011	CN 102576391 A	11/07/2012
		CN 102576391 B	04/03/2015
		EP 2488982 A1	22/08/2012
		EP 2488982 A4	29/05/2013
		JP 2013-507671 A	04/03/2013
		JP 5596159 B2	24/09/2014
		US 8205096 B2	19/06/2012
		WO 2011-044710 A1	21/04/2011
US 2005-0289072 A1	29/12/2005	None	
US 2014-0304771 A1	09/10/2014	CN 102609279 A	25/07/2012
		EP 2659421 A2	06/11/2013
		EP 2659421 A4	29/04/2015
		TW 201227502 A	01/07/2012
		US 2012-167121 A1	28/06/2012
		US 2014-250494 A1	04/09/2014
		US 8789138 B2	22/07/2014
		WO 2012-092111 A2	05/07/2012
		WO 2012-092111 A3	17/01/2013