

**FACILITATING THIRD PARTIES TO PERFORM BATCH PROCESSING OF REQUESTS REQUIRING
AUTHORIZATION FROM RESOURCE OWNERS FOR REPEAT ACCESS TO RESOURCES**

Priority Claim

[001] The instant patent application is related to and claims priority from the below co-pending applications, which are incorporated in their entirety herewith:

[002] A. India patent application entitled, “Facilitating Third Parties To Perform Batch Processing Of Requests Requiring Authorization From Resource Owners For Repeat Access To Resources”, naming the same Applicant as in the instant patent application, Serial No.: 758/CHE/2014, Filed: 18 February 2014; and

[003] B. U.S. Non Provisional Patent Application entitled “Facilitating Third Parties To Perform Batch Processing Of Requests Requiring Authorization From Resource Owners For Repeat Access To Resources”, naming the same Applicant as in the instant patent application, Serial No. 14300251, filed June 10, 2014.

Background of the Disclosure

[004] Technical Field

[005] The present disclosure relates to enterprise systems and more specifically to facilitating third parties to perform batch processing of requests requiring authorization from resource owners for repeat access to resources.

[006] Related Art

[007] A resource refers to any data item such as files, photos, user information, etc. Resources are often ‘owned’ by owners, but stored and managed by other/second parties. Ownership of a resource implies that the corresponding user (usually a human being) can control who is permitted or denied access to the resource managed by a second party. Such resources are referred to as “protected resources”.

[008] An owner can permit access of a resource to a third party by granting appropriate authorization. There is often a need to permit ‘repeat’ access to a resource, implying that a single authorization can be a basis for multiple accesses over time. OAuth protocol is one open standard, which allows for such authorization for repeat access of the resource on behalf of the owner, as is well known in the relevant arts.

[009] There are situations when batch processing of requests is desirable. Batch processing implies that several requests are accumulated (by appropriate buffering), and processed in a batch (i.e., continue to process the accumulated requests until the processing of all requests is

completed) at a later convenient time. Batch processing is generally helpful in large scale environments.

[010] Aspects of the present disclosure facilitate third parties to perform batch processing of requests requiring authorization from resource owners for repeat access to resources.

Brief Description of the Drawings

[011] Example embodiments of the present disclosure will be described with reference to the accompanying drawings briefly described below.

[012] Figure 1 is a block diagram illustrating an example environment in which several aspects of the present disclosure can be implemented.

[013] Figure 2 is a sequence diagram illustrating the manner in which authorization of protected resources is implemented in one embodiment.

[014] Figure 3 is a flow chart illustrating the manner in which third parties are facilitated to perform batch processing of requests requiring authorization from resource owners for repeat access to resources according to an aspect of the present disclosure.

[015] Figure 4A illustrates an example implementation of a server performing batch processing of requests requiring authorization from resource owners for repeat access to resources in one embodiment.

[016] Figures 4B and 4C depicts respective portions of credential data maintained at different time instances in one embodiment.

[017] Figure 5 is a block diagram illustrating the details of a digital processing system in which various aspects of the present disclosure are operative by execution of appropriate executable modules.

[018] Figure 6 is a block diagram showing a digital processing system for performing batch processing of requests in another embodiment of the present disclosure.

[019] In the drawings, like reference numbers generally indicate identical, functionally similar, and/or structurally similar elements. The drawing in which an element first appears is indicated by the leftmost digit(s) in the corresponding reference number.

Detailed Description of the Embodiments of the Disclosure

[020] 1. Overview

[021] According to an aspect of the present disclosure, a server system/third party receives multiple requests to be processed in a batch. The server system then selects a next request from the received requests, with the next request requiring a protected resource (hosted on a second party) owned by a owner/user (first party). The server system checks whether an access token is

present authorizing access of the protected resource by the server system on behalf of the owner. If the access token is not present, the server system communicates in an offline mode with the owner to receive the access token. The server system then processes the next request by accessing the protected resource using the present/received access token.

[022] Thus, third parties/server system are facilitated to perform batch processing of requests requiring authorization from resource owners for repeat access to resources.

[023] According to another aspect of the present disclosure, receiving of the access token from a owner in an offline mode is implemented by first sending a communication to the owner for asynchronous authorization of access to said first protected resource. In one embodiment, the communication sent to the owner includes a Uniform Resource Locator (URL), which when selected by the owner causes said authorization grant to be forwarded to the server system. After receiving an authorization grant for the first protected resource from the owner (in response to the owner selecting the URL), the access token is obtained by sending the received authorization grant to an authorization server.

[024] In one embodiment, the protected resource is hosted on a resource server, with the server system, the authorization server and the resource server being implemented to support OAuth protocol for authorization of protected resources.

[025] Several aspects of the present disclosure are described below with reference to examples for illustration. However, one skilled in the relevant art will recognize that the disclosure can be practiced without one or more of the specific details or with other methods, components, materials and so forth. In other instances, well-known structures, materials, or operations are not shown in detail to avoid obscuring the features of the disclosure. Furthermore, the features/aspects described can be practiced in various combinations, though only some of the combinations are described herein for conciseness.

[026] 2. Example Environment

[027] Figure 1 is a block diagram illustrating an example environment in which several aspects of the present disclosure can be implemented. The block diagram is shown containing end user systems 110A-110Z, Internet 120, intranet 130, server systems 140A-140C, SOA (service-oriented architecture) server 150, authorization server 170, and data store 180.

[028] Merely for illustration, only representative number/type of systems is shown in the Figure. Many environments often contain many more systems, both in number and type, depending on the purpose for which the environment is designed. Furthermore, it is assumed that the systems of the enterprise are implemented in accordance with OAuth standard protocol (as described in detail in RFC 6749 entitled "The OAuth 2.0 Authorization Framework"), though other similar protocols can be used in alternative embodiments. Each system/ device of Figure 1 is described

below in further detail.

[029] Intranet 130 represents a network providing connectivity between server systems 140A-140C, SOA server 150, authorization server 170 and data store 180, all provided within an enterprise (shown with dotted boundaries). Internet 120 extends the connectivity of these (and other systems of the enterprise) with external systems such as end user systems 110A-110Z. Each of intranet 140 and Internet 120 may be implemented using protocols such as Transmission Control Protocol (TCP) and/or Internet Protocol (IP), well known in the relevant arts. In general, in TCP/IP environments, an IP packet is used as a basic unit of transport, with the source address being set to the IP address assigned to the source system from which the packet originates and the destination address set to the IP address of the destination system to which the packet is to be eventually delivered.

[030] A (IP) packet is said to be directed to a destination system when the destination IP address of the packet is set to the (IP) address of the destination system, such that the packet is eventually delivered to the destination system by network 110. When the packet contains content such as port numbers, which specifies the destination application, the packet may be said to be directed to such application as well. The destination system may be required to keep the corresponding port numbers available/open, and process the packets with the corresponding destination ports. Each of Internet 120 and intranet 130 may be implemented using any combination of wire-based or wireless mediums.

[031] Data store 180 represents a non-volatile (persistent) storage facilitating storage and retrieval of a collection of data by applications executing in other systems of the enterprise such as server systems 140A-140C, SOA server 150 and authorization server 170. Data store 180 may be implemented as a database server using relational database technologies and accordingly provide storage and retrieval of data using structured queries such as SQL (Structured Query Language). Alternatively, data store 180 may be implemented as a file server providing storage and retrieval of data in the form of files organized as one or more directories, as is well known in the relevant arts.

[032] Each of end user systems 110A-110Z represents a system such as a personal computer, workstation, mobile station, mobile phones, computing tablets, etc., used by users to generate and send user requests directed to specific systems of the enterprise. The user requests may be generated using appropriate user interfaces (for example, web pages provided by applications executing in the enterprise). For example, the users may send user requests for performing various tasks to applications executing in server systems 140A-140C. Alternatively, users/owners may manage (create, remove, authorize/ deny access to) various resources of interest by sending user requests to the systems of the enterprise over Internet 120.

[033] Each of server systems 140A-140C represents a server, such as a web/application server, capable of protecting and providing access to resources in the enterprise (and as such represents the second party). Some of the protected resources such as applications, web pages provided by the applications, etc. may be hosted on the server system. Alternatively, the protected resources (e.g., personal data, media such as images or videos, etc. may be hosted on backend storage systems (not shown), with the access of such resources being controlled by the server systems. For illustration, the protected resources are shown to be hosted within the enterprise (dotted boundary). However, the features of the present disclosure may be implemented in alternative embodiments where the protected resources may be hosted on systems (not shown) external to the enterprise (with the resources being accessed via the Internet 120), as will be apparent to one skilled in the relevant arts, by reading the disclosure herein.

[034] Application server 160 represents a server, such as a web/application server, capable of executing third party applications that may wish to access some of the protected resources (hosted within the enterprise). The third party applications may require access to the protected resource for processing user requests (for example, received from end user systems 110A-110Z). In one embodiment, the third party applications are designed to operate with authorization server 170 for performing authorization of the protected resources. Upon successful authorization, the third party applications may repeatedly access the authorized protected resource and process the user requests.

[035] Authorization server 170 represents a system such as a server, which facilitates authorization of protected resources (by owners/users), and thereby enables third parties/applications to process requests requiring repeated access to resources. As noted above, authorization server 170 and other systems of the enterprise (such as server systems 140A-140C, application server 160 and data store 180) are implemented to perform the authorization of resources in accordance with OAuth protocol. The manner in which such authorization of protected resources may be performed is described below with examples.

[036] 3. Authorization of Protected Resources

[037] Figure 2 is a sequence diagram illustrating the manner in which authorization of protected resources is implemented according to OAuth protocol in the environment of Figure 1 in one embodiment. The description is continued assuming that third party application 210 executing in application server 160 requires access to a protected resource hosted in server system 140B, the protected resource being owned by an owner/user using end user system 110A. While the description is provided in accordance with OAuth protocol for illustration, it should be appreciated the features may be implemented with other similar authorization protocols, as suited in the corresponding environments.

[038] Third party application 210 is shown sending an authorization request to end user system 110A used by the owner (event 220). The authorization request may indicate the details of the protected resource, the third party application requesting the resource, the number/duration of access required, etc. In response, the user/owner may specify the access rights (view, modify, expiry period for the access, etc.) of the third party application (in general, authorize) for the requested protected resource and thereafter send a response indicating authorization grant (event 230). The authorization grant may indicate the rights specified by the user/owner.

[039] Though not shown, it may be appreciated that the user/owner may be required to authenticate himself/ herself with a second party (such as server system 140B that is protecting the resource) prior to providing authorization. The authentication of the user (i.e., establishing the identity of the user) may be performed using an appropriate authentication technique such as using a combination of userid and password, as is well known in the arts. After successful authentication, the user may specify the desired access rights as part of authorization grant. The authentication and authorization may be performed using appropriate user interfaces provided by the second party.

[040] Third party application 210 then sends the authorization grant to authorization server 170 (event 240), which in turn generates an access token corresponding to the authorization grant and sends the generated access token as a response to the authorization grant (event 250). The access token may indicate (typically, in an encrypted form) the details of the third party application requesting access, the protected resource, and the specific access rights (including any expiry period) specified by the owner of the resource.

[041] Third party application 210 thereafter sends the access token to server system 140B, the resource server hosting the resource (event 260) and accordingly is provided access to the protected resource (event 270). It may be appreciated that after an access token is generated, the access token may be thereafter used for repeated access of the protected resource (assuming that the access token has not expired). Third party application 210 may store the access token received from authorization server in data store 180 for facilitating repeated access to the protected resource.

[042] Thus, third party software (210) is facilitated to repeatedly access a protected resource on behalf of an owner based on the authorization provided by the owner. At least in such approaches based on OAuth protocol, it may be appreciated that the generation of the access token requires the active participation of the owner, that is, the owners are required to be “live/online” for providing the necessary authorization.

[043] Several challenges are therefore presented when performing batch processing of multiple requests. As is well known in the relevant arts, processing of multiple requests (received at

different prior time instances) together in a batch (at a later time instance) is often performed in the background without any user interaction (as compared, to performing 'live/online'). Such an approach is also not feasible in large scale environment (such as the enterprise of Figure 1) containing a large number (in the magnitude of thousands) of users/owners, and where users/owners are often added dynamically during the execution of the third party software. Furthermore, the revocation/expiry of the access tokens may necessitate the regeneration of the access tokens.

[044] SOA server 150 provided according to several aspects of the present disclosure, performs batch processing of requests requiring authorization from resource owners for repeat access to resources, while overcoming some of the drawbacks noted above. As is well known, a SOA server represents a server system that processes requests by execution of appropriate business flows. Some of the activities of the executed business flows are implemented as invocations to one or more web services (protected resources) exposed by server systems 140A-140C, with the exposed web services belonging to different owners. The manner in which SOA server 150 performs batch processing of requests requiring access to such protected resources/web services is described below with examples.

[045] 4. Batch Processing of Requests

[046] Figure 3 is a flow chart illustrating the manner in which third parties are facilitated to perform batch processing of requests requiring authorization from resource owners for repeat access to resources according to an aspect of the present disclosure. The flowchart is described with respect to the systems of Figure 1, in particular, SOA server 150, merely for illustration. However, the features can be implemented in other systems and environments also without departing from the scope and spirit of various aspects of the present disclosure, as will be apparent to one skilled in the relevant arts by reading the disclosure provided herein.

[047] In addition, some of the steps may be performed in a different sequence than that depicted below, as suited to the specific environment, as will be apparent to one skilled in the relevant arts. Many of such implementations are contemplated to be covered by several aspects of the present disclosure. The flow chart begins in step 301, in which control immediately passes to step 310.

[048] In step 310, SOA server 150 receives multiple requests for processing in a batch. The requests may be received from end user systems 110A-110Z at different prior time instances spread over a substantially long duration (e.g., over a day), with the requests being buffered for processing starting at a later time instance.

[049] In step 320, SOA server 150 selects a next request to be processed, the selected request requiring a protected resource (e.g. web service) owned by a owner. The selection of the next

request may be performed in any convenient manner, based on a predetermined selection criteria. The selection criteria includes, for example, the received time instances, the priority of the requests, etc. The protected resources may be hosted and or protected by server systems 140B-140C.

[050] In step 330, SOA server 150 checks whether an access token authorizing access of the protected resource on behalf of the owner is present. The check may be performed by inspecting a credential data indicating the previously generated (and not yet expired/revoked) access tokens. The inspection operation is in accordance with the conventions of OAuth protocol, as the access token content is defined by the protocol. Control passes to step 380 if an access token for the required protected resource is present and to step 350 otherwise.

[051] In step 350, SOA server 150 sends a communication to the owner for asynchronous authorization of access to the protected resource. Any necessary authentication, etc., as described above, may be performed as a part of such asynchronous authorization.

[052] The word 'asynchronous' implies that the owner is not required to substantially immediately respond (as would be required in synchronous/online response modes) to the communication, but can respond at a later convenient time (e.g., in a few hours/days, etc.). While the communication is described as being a URL in the below described embodiments, alternative approaches can employ code snippets (e.g., applets), as will be apparent to a skilled practitioner by reading the disclosure provided herein.

[053] In step 360, SOA server 150 receives authorization grant for the protected resource, upon the owner providing the asynchronous authorization. In step 370, SOA server 150 obtains an access token by sending the authorization grant to authorization server 170, similar to events 240 and 250 of Figure 2. It should be appreciated that the steps of 350, 360 and 370 are performed in an offline mode, that is, without requiring the owner to be live/online.

[054] In step 380, SOA server 150 processes the next request by accessing the protected resource using the previously present/obtained access token (similar to events 260 and 270 of Figure 2). In step 390, SOA server 150 determines whether more requests are required to be processed in the batch. Control passes to step 320 if more requests are present (and accordingly the next requests are processed), and to step 399 otherwise. The flow chart ends in step 399.

[055] Thus, SOA server 150 performs batch processing of requests requiring authorization from resource owners for repeat access to resources. The manner in which SOA server 150 may perform batch processing of request according to Figure 3 is illustrated below with examples.

[056] 5. Illustrative Example

[057] Figures 4A-4C together illustrate the manner in which third parties are facilitated to perform batch processing of requests requiring authorization from resource owners for repeat

access to resources in one embodiment. Each of the Figures is described in detail below.

[058] Figure 4A illustrates an example implementation of a server (SOA server 150) performing batch processing of requests requiring authorization from resource owners for repeat access to resources in one embodiment. Broadly, a front-end order application (not shown) generates orders for products and stores the generated orders in a non-volatile storage (e.g. orders data 410 in data store 180). The orders are then processed in a batch by a business flow (order processor 420) executing in SOA server 150.

[059] One of the activities to be performed as part of processing an order is to periodically update, on behalf of the customer organization (who has placed the order), a personal website (e.g. Twitter[™]) with the current status of the order (e.g., “Order received”, “Order processing started”, “Invoice generated”, etc.). It is assumed that the personal website allows users to create various accounts and to post messages on to such accounts. Other users (followers) may follow the desired accounts, whereby the followers automatically receive the messages of such desired accounts.

[060] Thus, for SOA server 150 to be able to post the order status (and thus provide the requisite periodic update to followers of account of the customer organization), SOA server 150 needs to be provided authorization of the account of the customer organization. In other words, the account is viewed as a protected resource, which is to be accessed on behalf of the customer organization. As may be appreciated, OAuth described above is generally designed for providing such access to third parties. However, when requests are processed in a batch, there are several challenges, some of which are noted above.

[061] SOA server 150 addresses at least some of such challenges, as described in detail below with respect to Figures 4A and 4B.

[062] Orders data 410 (maintained in data store 180, in one embodiment) specifies the details of the requests/orders (generated and stored by the front-end order application) sought to be processed in a batch. Order processor 420 represents a business flow executing in SOA server 150, and which is enabled to perform batch processing of orders. As part of the business flow, order processor 420 is designed to invoke target (web) service 480 exposed by server system 140C to update the customer’s account on the personal website with the current status of processing the order.

[063] Accordingly, order processor 420 in response to selecting an order/request from the received orders (maintained in orders data 410), first interfaces with credential checker 430 to determine whether an access token for target server 480 is already present. Credential checker 430 inspects credential data 460 to determine whether the access token (for enabling order processor 420 to access the customer’s account on the personal website) is present. The manner

in which credential data may be maintained and used for obtaining the access token in an offline mode is described below with examples.

[064] 6. Credential Data

[065] Figures 4B and 4C depicts respective portions of credential data (460) maintained at different time instances in one embodiment. For illustration, it is assumed that the credential data is also (along with orders data 410) stored in data store 180, implemented as a database server. Accordingly, credential data 460 is shown maintained in the form of a table in a database in the database server. However, in alternative embodiments, the credential data may be maintained using other data structures (such as trees, associative arrays, etc.) and/or in any other format such as XML (Extensible Markup Language), as will be apparent to one skilled in the relevant arts by reading the disclosure herein.

[066] Referring to Figure 4B, credential data/table 460 is shown containing five columns with column "User ID" specifying a unique identifier of the user/owner, column "Access Token" specifying the access token that has been generated for the user/owner, column "Expiry Date" indicating the date until which the access token is valid, column "Request Date" indicating the date on which a communication for obtaining the access token in an offline mode was sent (to the owner/user by SOA server 150) and column "Status" indicating the current status ("Active", "Expired", "Revoked", "Pending" etc.) of the access token.

[067] Each of the rows of table 460 specifies the details of the access token received from a corresponding user/owner. As noted above, the access token specifies (in an encrypted form) the details of the third party application requesting access, the protected resource, and the specific access rights (including any expiry period) specified by the owner of the resource. An access token is deemed to be present when the status is "Active" and the corresponding value in column "Access Token" is not blank.

[068] Thus, the credential data (460) for different users/owners of protected resources is maintained by SOA server 150. During the processing of a request, credential checker 430 checks whether an access token of the customer is present in table 460.

[069] In the scenario that the access token authorizing order processor 420 to access target service 480 on behalf of the owner is present, credential checker 430 forwards the access token to order processor 420. Order processor 420 then access target service 480 using the provided access token, thereby updating the customer's account on the personal website with the current status.

[070] In the scenario that the access token is not present, credential checker 430 first stores the details of the order such as the owner/user from whom the authorization is to be obtained, the date on which the request for authorization is sent, the status as "Pending", etc., in table 460 (as

shown in row 490). Credential checker 430 then sends an indication (along with other data such as the user ID) to URL generator 440 that the owner of target service 480 is to be communicated with in an offline mode to obtain the access token. URL generator 440, in response to the indication, generates and sends to the owner (determined based on the received user ID), an URL which when selected asynchronously by the owner causes an authorization grant for the protected resource to be forwarded to callback service 450. Such an URL may be generated in any convenient manner.

[071] An example URL format that causes the authorization grant for target service 480 to be forwarded to callback service 450 is shown below:

[072] `https://domain-name/authrequest?callbackuri=<http://domain-name/CredentialCallbackService&user-id=578578>`

[073] wherein,

[074] "domain-name" is the domain of the enterprise hosting the third party application;

[075] "authrequest" is the name of the service (e.g. provided by server system 140C or authorization server 170) used to generate the authorization grant and which is invoked upon the user/owner selecting the URL;

[076] "callbackuri" is a keyword indicating that the authorization grant is to be forwarded to the URI (universal resource identifier) specified as the value ("http://domain-name/CredentialCallbackService") of the keyword;

[077] "CredentialCallbackService" is the identify of the call back service (450); and

[078] "user-id" is a keyword indicating the user/owner ID (e.g. 578578) for whom the access token is generated.

[079] The URL may then be delivered to the owner in a known way, for example, as part of an email message sent to the inbox of the owner, in the form of a short message service (SMS) message sent to the mobile device (e.g., 110A) of the owner, etc. The URL may thereafter be selected by the owner at a later time instance (asynchronously), for example (potentially after several days), when accessing the email from the inbox using end user system 110A, while viewing the SMS message in the mobile device, etc. The owner then selects the URL, performs any necessary authentication with server system 140C (hosting the protected resource), and then creates an authorization grant (indicating the specific authorizations specified by the owner) for target service 480. Due to the format of the URL, the newly created authorization grant is forwarded to callback service 450.

[080] It may be appreciated that after sending the communication to the owner, order processor 420 does not wait for the reply to the communication (as would be required in synchronous

communications), but continues with the processing of the selected order and/or other orders in the batch.

[081] Callback service 450 receives the authorization grant, and then obtains an access token by sending the authorization grant to authorization server 170, similar to events 240 and 250 of Figure 2. Callback service 450 then stores the newly obtained access token as part of credential data 460 (row 495 of Figure 4C), and then notifies order processor 420 that the access token is present. Order process 420 may thereafter interface with credential checker 430 to retrieve the newly obtained access token from credential data 460 and processes the orders requiring accesses to target service 480.

[082] Thus, the third party order processor 420 is facilitated to perform batch processing of orders/requests requiring authorization from resource owners (first party) for repeat access to resources/web services (hosted/protected by second parties such as server systems 140A-140C).

[083] It should be appreciated that the features described above can be implemented in various embodiments as a desired combination of one or more of hardware, executable modules, and firmware. The description is continued with respect to an embodiment in which various features are operative when executable modules are executed.

[084] 7. Digital Processing System

[085] Figure 5 is a block diagram illustrating the details of digital processing system 500 in which various aspects of the present disclosure are operative by execution of appropriate executable modules. Digital processing system 500 may correspond to SOA server 150.

[086] Digital processing system 500 may contain one or more processors such as a central processing unit (CPU) 510, random access memory (RAM) 520, secondary memory 530, graphics controller 560, display unit 570, network interface 580, and input interface 590. All the components except display unit 570 may communicate with each other over communication path 550, which may contain several buses as is well known in the relevant arts. The components of Figure 5 are described below in further detail.

[087] CPU 510 may execute instructions stored in RAM 520 to provide several features of the present disclosure. CPU 510 may contain multiple processing units, with each processing unit potentially being designed for a specific task. Alternatively, CPU 510 may contain only a single general-purpose processing unit.

[088] RAM 520 may receive instructions from secondary memory 530 using communication path 550. RAM 520 is shown currently containing software instructions constituting operating environment 525 and/or other user programs 526 (such as database software, etc.). In addition to operating environment 525, RAM 520 may contain other software programs such as device drivers, virtual machines, etc., which provide a (common) run time environment for execution of

other/user programs.

[089] Graphics controller 560 generates display signals (e.g., in RGB format) to display unit 570 based on data/instructions received from CPU 510. Display unit 570 contains a display screen to display the images defined by the display signals. Input interface 590 may correspond to a keyboard and a pointing device (e.g., touch-pad, mouse) and may be used to provide inputs. Network interface 580 provides connectivity to a network (e.g., using Internet Protocol), and may be used to communicate with other systems connected to the network (120 and 130 of Figure 1).

[090] Secondary memory 530 may contain hard drive 535, flash memory 536, and removable storage drive 537. Secondary memory 530 may store the data (for example, portions of the URL) and software instructions (for implementing the steps of Figure 3), which enable digital processing system 500 to provide several features in accordance with the present disclosure. The code/instructions stored in secondary memory 530 may either be copied to RAM 520 prior to execution by CPU 510 for higher execution speeds, or may be directly executed by CPU 510.

[091] Secondary memory 530 may contain hard drive 535, flash memory 536, and removable storage drive 537. Some or all of the data and instructions may be provided on removable storage unit 540, and the data and instructions may be read and provided by removable storage drive 537 to CPU 510. Removable storage unit 540 may be implemented using medium and storage format compatible with removable storage drive 537 such that removable storage drive 537 can read the data and instructions. Thus, removable storage unit 540 includes a computer readable (storage) medium having stored therein computer software and/or data. However, the computer (or machine, in general) readable medium can be in other forms (e.g., non-removable, random access, etc.).

[092] In this document, the term "computer program product" is used to generally refer to removable storage unit 540 or hard disk installed in hard drive 535. These computer program products are means for providing software to digital processing system 500. CPU 510 may retrieve the software instructions, and execute the instructions to provide various features of the present disclosure described above.

[093] The term "storage media/medium" as used herein refers to any non-transitory media that store data and/or instructions that cause a machine to operate in a specific fashion. Such storage media may comprise non-volatile media and/or volatile media. Non-volatile media includes, for example, optical disks, magnetic disks, or solid-state drives, such as storage memory 530. Volatile media includes dynamic memory, such as RAM 520. Common forms of storage media include, for example, a floppy disk, a flexible disk, hard disk, solid-state drive, magnetic tape, or any other magnetic data storage medium, a CD-ROM, any other optical data storage medium,

any physical medium with patterns of holes, a RAM, a PROM, and EPROM, a FLASH-EPROM, NVRAM, any other memory chip or cartridge.

[094] Storage media is distinct from but may be used in conjunction with transmission media. Transmission media participates in transferring information between storage media. For example, transmission media includes coaxial cables, copper wire and fiber optics, including the wires that comprise bus 550. Transmission media can also take the form of acoustic or light waves, such as those generated during radio-wave and infra-red data communications.

[095] Figure 6 is a block diagram showing digital processing system 600 for performing batch processing of requests in another embodiment of the present disclosure. The blocks of the system 600 may be implemented by hardware, software, or a combination of hardware and software to carry out the principles of the disclosure. It is understood by those skilled in the art that the blocks described in Figure 6 may be combined or separated into sub-blocks to implement the principles of the disclosure as described above. Therefore, the description herein may support any possible combination or separation or further definition of the functional blocks described herein.

[096] As shown in figure 6, the system 600 for processing requests in a server system may comprise a first receiving unit 610, a selecting unit 620, a checking unit 630, a communicating unit 640, and a processing unit 650. The first receiving unit 610 can receive a plurality of requests for processing in a batch. The selecting unit 620 can select a next request from the plurality of requests. The next request requires a first protected resource owned by a first owner. The checking unit 630 can check whether an access token is present authorizing access of the first protected resource by the server system on behalf of the first owner. The communicating unit 640 can communicate in an offline mode with the first owner to receive the access token if the access token is not present. The processing unit 650 can process the next request by accessing the first protected resource using the access token.

[097] In accordance with an embodiment of the disclosure, the communicating unit may comprise: a sending unit 641 for sending a communication to the first owner for asynchronous authorization of access to the first protected resource; a second receiving unit 642 for receiving an authorization grant for the first protected resource; and an obtaining unit 643 for obtaining the access token by sending the authorization grant to an authorization server.

[098] In accordance with an embodiment of the disclosure, the communication can be an email message or a short message service (SMS) message.

[099] In accordance with an embodiment of the disclosure, the communication may include a Uniform Resource Locator (URL), which when selected asynchronously by the first owner causes the authorization grant to be forwarded to the server system. The second receiving unit

642 may receive the authorization grant in response to the first owner selecting the URL and authorizes access to the first protected resource.

[0100] In accordance with an embodiment of the disclosure, the system may further comprise a maintaining unit 660 for maintaining credential data indicating the access tokens generated for owners including the first owner. The checking may comprise inspecting the credential data to determine whether the access token is present for the first owner.

[0101] In accordance with an embodiment of the disclosure, the first protected resource can be hosted on a resource server. The server system, the authorization server and the resource server may be implemented to support OAuth protocol for authorization of protected resources.

[0102] In accordance with an embodiment of the disclosure, the server system can be a Service Oriented Architecture (SOA) server. The plurality of requests may correspond to a plurality of orders such that the next request is a next order. The protected resource can be an account in a personal website. The first owner can be an administrator who authorizes access to the account. The processing unit may comprise an updating unit 651 for updating the account with a current status of processing of the next order.

[0103] Reference throughout this specification to “one embodiment”, “an embodiment”, or similar language means that a particular feature, structure, or characteristic described in connection with the embodiment is included in at least one embodiment of the present disclosure. Thus, appearances of the phrases “in one embodiment”, “in an embodiment” and similar language throughout this specification may, but do not necessarily, all refer to the same embodiment.

[0104] Furthermore, the described features, structures, or characteristics of the disclosure may be combined in any suitable manner in one or more embodiments. In the above description, numerous specific details are provided such as examples of programming, software modules, user selections, network transactions, database queries, database structures, hardware modules, hardware circuits, hardware chips, etc., to provide a thorough understanding of embodiments of the disclosure.

[0105] 8. Conclusion

[0106] While various embodiments of the present disclosure have been described above, it should be understood that they have been presented by way of example only, and not limitation. Thus, the breadth and scope of the present disclosure should not be limited by any of the above-described exemplary embodiments, but should be defined only in accordance with the following claims and their equivalents.

[0107] It should be understood that the figures and/or screen shots illustrated in the attachments highlighting the functionality and advantages of the present disclosure are presented for example

purposes only. The present disclosure is sufficiently flexible and configurable, such that it may be utilized in ways other than that shown in the accompanying figures.

[0108] Further, the purpose of the following Abstract is to enable the Patent Office and the public generally, and especially the scientists, engineers and practitioners in the art who are not familiar with patent or legal terms or phraseology, to determine quickly from a cursory inspection the nature and essence of the technical disclosure of the application. The Abstract is not intended to be limiting as to the scope of the present disclosure in any way.

What is Claimed Is:

1. A method of processing requests in a server system, said method comprising:
receiving a plurality of requests for processing in a batch;
selecting a next request from said plurality of requests, said next request requiring a first protected resource owned by a first owner;
checking whether an access token is present authorizing access of said first protected resource by said server system on behalf of said first owner;
if said access token is not present, communicating in an offline mode with said first owner to receive said access token; and
processing said next request by accessing said first protected resource using said access token.
2. The method of claim 1, wherein said communicating in said offline mode comprises:
sending a communication to said first owner for asynchronous authorization of access to said first protected resource;
receiving an authorization grant for said first protected resource; and
obtaining said access token by sending said authorization grant to an authorization server.
3. The method of claim 2, wherein said communication is an email message or a short message service (SMS) message.
4. The method of claim 2 or claim 3, wherein said communication includes a Uniform Resource Locator (URL), which when selected asynchronously by said first owner causes said authorization grant to be forwarded to said server system,
wherein said receiving receives said authorization grant in response to said first owner selecting said URL and authorising access to said first protected resource.
5. The method of any of the preceding claims, further comprising maintaining credential data indicating the access tokens generated for owners including said first owner,
wherein said checking comprises inspecting said credential data to determine whether said access token is present for said first owner.
6. The method of any of the preceding claims, wherein said first protected resource is hosted on a resource server, wherein said server system, said authorization server and said

resource server are implemented to support OAuth protocol for authorization of protected resources.

7. The method of any of the preceeding claims, wherein said server system is a Service Oriented Architecture (SOA) server, wherein said plurality of requests correspond to a plurality of orders such that said next request is a next order,

wherein said protected resource is an account in a personal website, wherein said first owner is an administrator who authorizes access to said account,

wherein said processing of said next order comprises updating said account with a current status of processing of said next order.

8. A non-transitory machine readable medium storing one or more sequences of instructions for enabling a server system to process requests, wherein execution of said one or more instructions by one or more processors contained in said server system enables said server system to perform the actions of:

receiving a plurality of requests for processing in a batch;

selecting a next request from said plurality of requests, said next request requiring a first protected resource owned by a first owner;

checking whether an access token is present authorizing access of said first protected resource by said server system on behalf of said first owner;

if said access token is not present, communicating in an offline mode with said first owner to receive said access token; and

processing said next request by accessing said first protected resource using said access token.

9. The machine readable medium of claim 8, wherein said communicating in said offline mode comprises one or more instructions for:

sending a communication to said first owner for asynchronous authorization of access to said first protected resource;

receiving an authorization grant for said first protected resource; and

obtaining said access token by sending said authorization grant to an authorization server.

10. The machine readable medium of claim 9, wherein said communication is an email message or a short message service (SMS) message.

11. The machine readable medium of claim 9 or claim 10, wherein said communication includes a Uniform Resource Locator (URL), which when selected asynchronously by said first owner causes said authorization grant to be forwarded to said server system,

wherein said receiving receives said authorization grant in response to said first owner selecting said URL and authorising access to said first protected resource.

12. The machine readable medium of any of claims 9 to 11, further comprising one or more instructions for maintaining credential data indicating the access tokens generated for owners including said first owner,

wherein said checking comprises one or more instructions inspecting said credential data to determine whether said access token is present for said first owner.

13. The machine readable medium of any of claims 9 to 12, wherein said first protected resource is hosted on a resource server, wherein said server system, said authorization server and said resource server are implemented to support OAuth protocol for authorization of protected resources.

14. A digital processing system comprising:

a processor;

a random access memory (RAM);

a machine readable medium to store one or more instructions, which when retrieved into said RAM and executed by said processor causes said digital processing system to process request, said digital processing system performing the actions of:

receiving a plurality of requests for processing in a batch;

selecting a next request from said plurality of requests, said next request requiring a first protected resource owned by a first owner;

checking whether an access token is present authorizing access of said first protected resource by said server system on behalf of said first owner;

if said access token is not present, communicating in an offline mode with said first owner to receive said access token; and

processing said next request by accessing said first protected resource using said access token.

15. The digital processing system of claim 14, wherein for said communicating in said offline mode, said digital processing system performs the actions of:

sending a communication to said first owner for asynchronous authorization of access to said first protected resource;

receiving an authorization grant for said first protected resource; and

obtaining said access token by sending said authorization grant to an authorization server.

16. The digital processing system of claim 15, wherein said communication is an email message or a short message service (SMS) message.

17. The digital processing system of claim 15 or claim 16, wherein said communication includes a Uniform Resource Locator (URL), which when selected asynchronously by said first owner causes said authorization grant to be forwarded to said server system,

wherein said digital processing system receives said authorization grant in response to said first owner selecting said URL and authorising access to said first protected resource.

18. The digital processing system of any of claims 14 to 17, further performing the actions of maintaining credential data indicating the access tokens generated for owners including said first owner,

wherein for said checking, said digital processing system performs the actions of inspecting said credential data to determine whether said access token is present for said first owner.

19. The digital processing system of any of claims 14 to 18, wherein said first protected resource is hosted on a resource server, wherein said server system, said authorization server and said resource server are implemented to support OAuth protocol for authorization of protected resources.

20. The digital processing system of any of claims 14 to 19, wherein said digital processing system is a Service Oriented Architecture (SOA) server, wherein said plurality of requests correspond to a plurality of orders such that said next request is a next order,

wherein said protected resource is an account in a personal website, wherein said first owner is an administrator who authorizes access to said account,

wherein said processing of said next order comprises updating said account with a current status of processing of said next order.

21. A system for processing requests in a server system, said system comprising:

a first receiving unit configured for receiving a plurality of requests for processing in a batch;

a selecting unit configured for selecting a next request from said plurality of requests, said next request requiring a first protected resource owned by a first owner;

a checking unit configured for checking whether an access token is present authorizing access of said first protected resource by said server system on behalf of said first owner;

a communicating unit configured for communicating in an offline mode with said first owner to receive said access token if said access token is not present; and

a processing unit configured for processing said next request by accessing said first protected resource using said access token.

22. The system of claim 21, wherein said communicating unit comprises:

a sending unit configured for sending a communication to said first owner for asynchronous authorization of access to said first protected resource;

a second receiving unit configured for receiving an authorization grant for said first protected resource; and

an obtaining unit configured for obtaining said access token by sending said authorization grant to an authorization server.

23. The system of claim 22, wherein said communication is an email message or a short message service (SMS) message.

24. The system of claim 23, wherein said communication includes a Uniform Resource Locator (URL), which when selected asynchronously by said first owner causes said authorization grant to be forwarded to said server system,

wherein said second receiving unit is configured for receiving said authorization grant in response to said first owner selecting said URL and authorizing access to said first protected resource.

25. The system of claim 24, further comprising a maintaining unit configured for maintaining credential data indicating the access tokens generated for owners including said first owner,

wherein said checking comprises inspecting said credential data to determine whether said access token is present for said first owner.

26. The system of claim 25, wherein said first protected resource is hosted on a resource server, wherein said server system, said authorization server and said resource server are implemented to support OAuth protocol for authorization of protected resources.

27. The system of claim 25, wherein said server system is a Service Oriented Architecture (SOA) server, wherein said plurality of requests correspond to a plurality of orders such that said next request is a next order,

wherein said protected resource is an account in a personal website, wherein said first owner is an administrator who authorizes access to said account,

wherein said processing unit comprises an updating unit configured for updating said account with a current status of processing of said next order.

AMENDED CLAIMS

received by the International Bureau on 22.07.2015

What is Claimed Is:

1. A method of processing requests in a server system, said method comprising:
receiving a plurality of requests from one or more users at different time instances spread
over a substantially long duration;
5 buffering said plurality of requests for later processing in a batch;
wherein said processing in said batch comprises:
selecting a next request from said plurality of requests buffered for
processing, said next request requiring a first protected resource owned by a first
owner;
10 checking whether an access token is present authorizing access of said
first protected resource by said server system on behalf of said first owner;
if said access token is not present, communicating in an offline mode with
said first owner to receive said access token, wherein said communicating
comprises sending a communication at a first time instance and receiving said
15 access token at a second time instance, wherein operation in said offline mode
entails processing of other requests of said plurality of requests between said first
time instance and said second time instance; and
processing said next request by accessing said first protected resource
using said access token.
20
2. The method of claim 1, wherein said
communication is sent at said first time instance to said first owner for asynchronous
authorization of access to said first protected resource, said communicating in said offline mode
further comprises:
25 receiving an authorization grant for said first protected resource; and
obtaining said access token by sending said authorization grant to an authorization server,
wherein said asynchronization authorization implies that said first owner is not required
to be online at said first time instance.
- 30 3. The method of claim 2, wherein said communication is an email message or a short
message service (SMS) message.
4. The method of claim 2 or claim 3, wherein said communication includes a Uniform
Resource Locator (URL), which when selected asynchronously by said first owner causes said
35 authorization grant to be forwarded to said server system,

wherein said receiving receives said authorization grant in response to said first owner selecting said URL and authorising access to said first protected resource.

5 5. The method of any of the preceeding claims, further comprising maintaining credential data indicating the access tokens generated for owners including said first owner,
wherein said checking comprises inspecting said credential data to determine whether said access token is present for said first owner.

10 6. The method of any of the preceeding claims, wherein said first protected resource is hosted on a resource server, wherein said server system, said authorization server and said resource server are implemented to support OAuth protocol for authorization of protected resources.

15 7. The method of any of the preceeding claims, wherein said server system is a Service Oriented Architecture (SOA) server, wherein said plurality of requests correspond to a plurality of orders such that said next request is a next order,
wherein said protected resource is an account in a personal website, wherein said first owner is an administrator who authorizes access to said account,
wherein said processing of said next order comprises updating said account with a current
20 status of processing of said next order.

25 8. A non-transitory machine readable medium storing one or more sequences of instructions for enabling a server system to process requests, wherein execution of said one or more instructions by one or more processors contained in said server system enables said server system to perform the actions of:

receiving a plurality of requests from one or more users at different time instances spread over a substantially long duration;

buffering said plurality of requests for later processing in a batch;

wherein said processing in said batch comprises:

30 selecting a next request from said plurality of requests buffered for processing, said next request requiring a first protected resource owned by a first owner;

checking whether an access token is present authorizing access of said first protected resource by said server system on behalf of said first owner;

if said access token is not present, communicating in an offline mode with said first owner to receive said access token, wherein said communicating comprises sending a communication at a first time instance and receiving said access token at a second time instance, wherein operation in said offline mode entails processing of other requests of said plurality of requests between said first time instance and said second time instance; and

processing said next request by accessing said first protected resource using said access token.

9. The machine readable medium of claim 8, wherein said communication is sent at said first time instance to said first owner for asynchronous authorization of access to said first protected resource, said communicating in said offline mode further comprises one or more instructions for:

receiving an authorization grant for said first protected resource; and
obtaining said access token by sending said authorization grant to an authorization server, wherein said asynchronization authorization implies that said first owner is not required to be online at said first time instance.

10. The machine readable medium of claim 9, wherein said communication is an email message or a short message service (SMS) message.

11. The machine readable medium of claim 9 or claim 10, wherein said communication includes a Uniform Resource Locator (URL), which when selected asynchronously by said first owner causes said authorization grant to be forwarded to said server system,

wherein said receiving receives said authorization grant in response to said first owner selecting said URL and authorising access to said first protected resource.

12. The machine readable medium of any of claims 9 to 11, further comprising one or more instructions for maintaining credential data indicating the access tokens generated for owners including said first owner,

wherein said checking comprises one or more instructions inspecting said credential data to determine whether said access token is present for said first owner.

13. The machine readable medium of any of claims 9 to 12, wherein said first protected resource is hosted on a resource server, wherein said server system, said authorization server and

said resource server are implemented to support OAuth protocol for authorization of protected resources.

14. A digital processing system comprising:

5

a processor;

a random access memory (RAM);

a machine readable medium to store one or more instructions, which when retrieved into said RAM and executed by said processor causes said digital processing system to process request, said digital processing system performing the actions of:

10

receiving a plurality of requests from one or more users at different time instances spread over a substantially long duration;

buffering said plurality of requests for later processing in a batch;

wherein said processing in said batch comprises:

15

selecting a next request from said plurality of requests buffered for processing, said next request requiring a first protected resource owned by a first owner;

checking whether an access token is present authorizing access of said first protected resource by said server system on behalf of said first owner;

20

if said access token is not present, communicating in an offline mode with said first owner to receive said access token, wherein said communicating comprises sending a communication at a first time instance and receiving said access token at a second time instance, wherein operation in said offline mode entails processing of other requests of said plurality of requests between said first time instance and said second time instance; and

25

processing said next request by accessing said first protected resource using said access token.

15. The digital processing system of claim 14, wherein for said communicating in said offline mode, said digital processing system sends at said first time instance, said communication to said first owner for asynchronous authorization of access to said first protected resource, said digital processing system further performing the actions of:

30

receiving an authorization grant for said first protected resource; and

obtaining said access token by sending said authorization grant to an authorization server,

wherein said asynchronization authorization implies that said first owner is not required

35

to be online at said first time instance.

16. The digital processing system of claim 15, wherein said communication is an email message or a short message service (SMS) message.

5 17. The digital processing system of claim 15 or claim 16, wherein said communication includes a Uniform Resource Locator (URL), which when selected asynchronously by said first owner causes said authorization grant to be forwarded to said server system,

wherein said digital processing system receives said authorization grant in response to said first owner selecting said URL and authorising access to said first protected resource.

10 18. The digital processing system of any of claims 14 to 17, further performing the actions of maintaining credential data indicating the access tokens generated for owners including said first owner,

wherein for said checking, said digital processing system performs the actions of
15 inspecting said credential data to determine whether said access token is present for said first owner.

19. The digital processing system of any of claims 14 to 18, wherein said first protected resource is hosted on a resource server, wherein said server system, said authorization server and
20 said resource server are implemented to support OAuth protocol for authorization of protected resources.

20. The digital processing system of any of claims 14 to 19, wherein said digital processing system is a Service Oriented Architecture (SOA) server, wherein said plurality of
25 requests correspond to a plurality of orders such that said next request is a next order,

wherein said protected resource is an account in a personal website, wherein said first owner is an administrator who authorizes access to said account,

wherein said processing of said next order comprises updating said account with a current status of processing of said next order.

30 21. A system for processing requests in a server system, said system comprising:
a first receiving unit configured for:

receiving a plurality of requests from one or more users at different time instances spread over a substantially long duration; and

35 buffering said plurality of requests for later processing in a batch;

a selecting unit configured for selecting a next request from said plurality of requests buffered for processing, said next request requiring a first protected resource owned by a first owner;

5 a checking unit configured for checking whether an access token is present authorizing access of said first protected resource by said server system on behalf of said first owner;

a communicating unit configured for communicating in an offline mode with said first owner to receive said access token if said access token is not present, wherein said communicating unit sends a communication at a first time instance and receives said access token at a second time instance, wherein operation in said offline mode entails processing of
10 other requests of said plurality of requests between said first time instance and said second time instance; and

a processing unit configured for processing said next request by accessing said first protected resource using said access token.

15 22. The system of claim 21, wherein said communicating unit comprises:

a sending unit configured for sending at said first time instance, said communication to said first owner for asynchronous authorization of access to said first protected resource;

a second receiving unit configured for receiving an authorization grant for said first protected resource; and

20 an obtaining unit configured for obtaining said access token by sending said authorization grant to an authorization server,

wherein said asynchronization authorization implies that said first owner is not required to be online at said first time instance.

25 23. The system of claim 22, wherein said communication is an email message or a short message service (SMS) message.

24. The system of claim 23, wherein said communication includes a Uniform Resource Locator (URL), which when selected asynchronously by said first owner causes said
30 authorization grant to be forwarded to said server system,

wherein said second receiving unit is configured for receiving said authorization grant in response to said first owner selecting said URL and authorizing access to said first protected resource.

35 25. The system of claim 24, further comprising a maintaining unit configured for

maintaining credential data indicating the access tokens generated for owners including said first owner,

wherein said checking comprises inspecting said credential data to determine whether said access token is present for said first owner.

5

26. The system of claim 25, wherein said first protected resource is hosted on a resource server, wherein said server system, said authorization server and said resource server are implemented to support OAuth protocol for authorization of protected resources.

10

27. The system of claim 25, wherein said server system is a Service Oriented Architecture (SOA) server, wherein said plurality of requests correspond to a plurality of orders such that said next request is a next order,

wherein said protected resource is an account in a personal website, wherein said first owner is an administrator who authorizes access to said account,

15

wherein said processing unit comprises an updating unit configured for updating said account with a current status of processing of said next order.

STATEMENT UNDER ARTICLE 19(1)

Sir:

This statement clarifies that presented claims 1-27 recite an inventive step over document D1 (US Patent Application with Publication Number US 2013/139241 A1 (LEEDER MICHAEL [CA]) dated 30 May 2013), and other documents cited in the International Search Report mailed on 27-May-2015 (hereafter "ISR").

Remarks are provided in comparison with the general approach of D1 and also the specific disclosure associated with Figure 8 of D1.

For example, claim 1 has been amended to clarify that 'users' from whom requests are received, are different from 'owners' of protected resources. In sharp contrast, document D1 relates to enabling a user to access an application hosted in a web-based network based on user data/access token with the user her/himself in a telecommunication network. Thus, document D1 does not teach or reasonably suggest the claimed user and owner, as two separate and distinct entities.

In addition, claim 1 has been further amendment to define the terms 'processing in a batch' and 'communicating in an offline mode'. In particular, the claim has been
5 amended to recite that the requests received over a long duration are buffered for later processing in a batch. In addition, in the duration of obtaining the access token for a resource required for processing a request, other requests are processed in view of the claimed offline mode of operation. As may be readily appreciated, the offline mode is operable in the context of batch processing, as would be apparent to a skilled person.

10

Document D1 does not teach or reasonably suggest such communicating in offline mode in the context of batch processing. The remaining documents relied upon in the ISR do not cure that deficiency.

15 For one or more of the above reasons, the invention of claim 1 involves an inventive step over the references in the ISR.

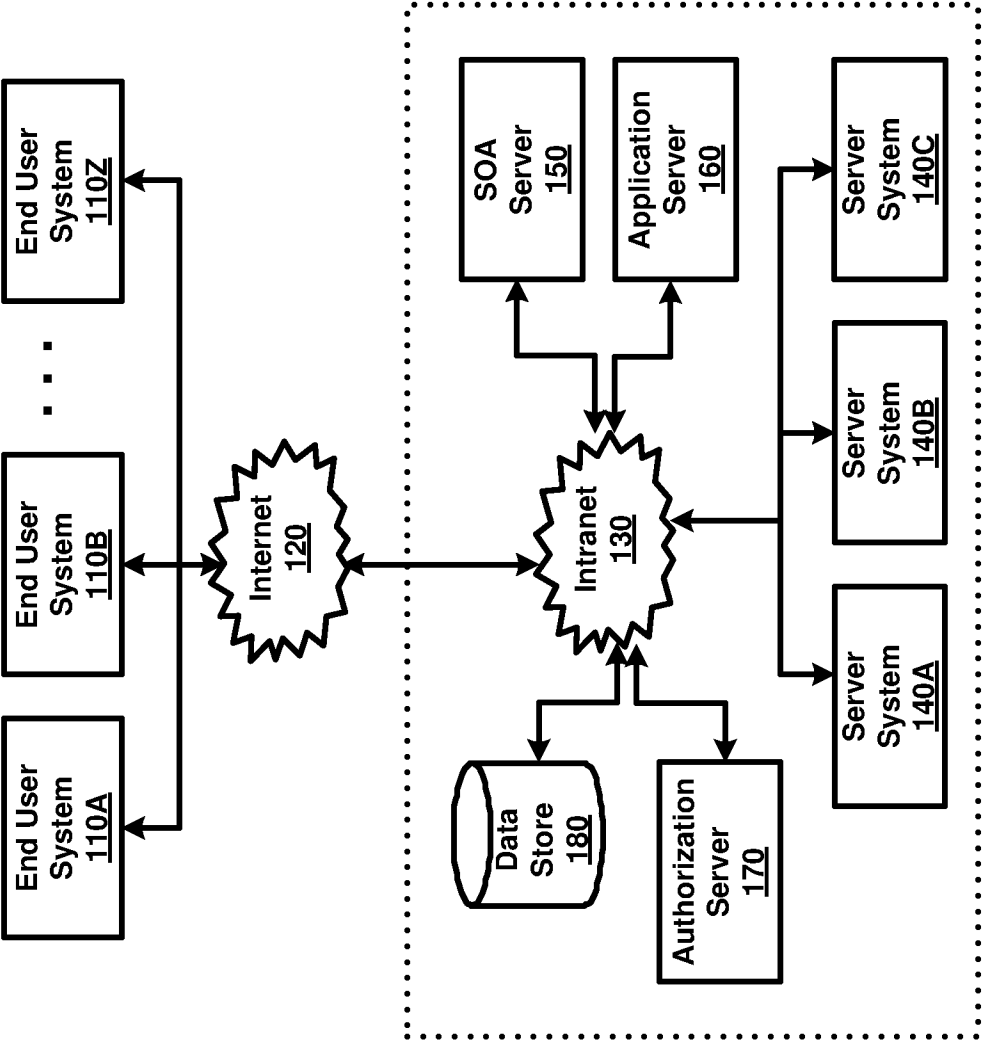


FIG. 1

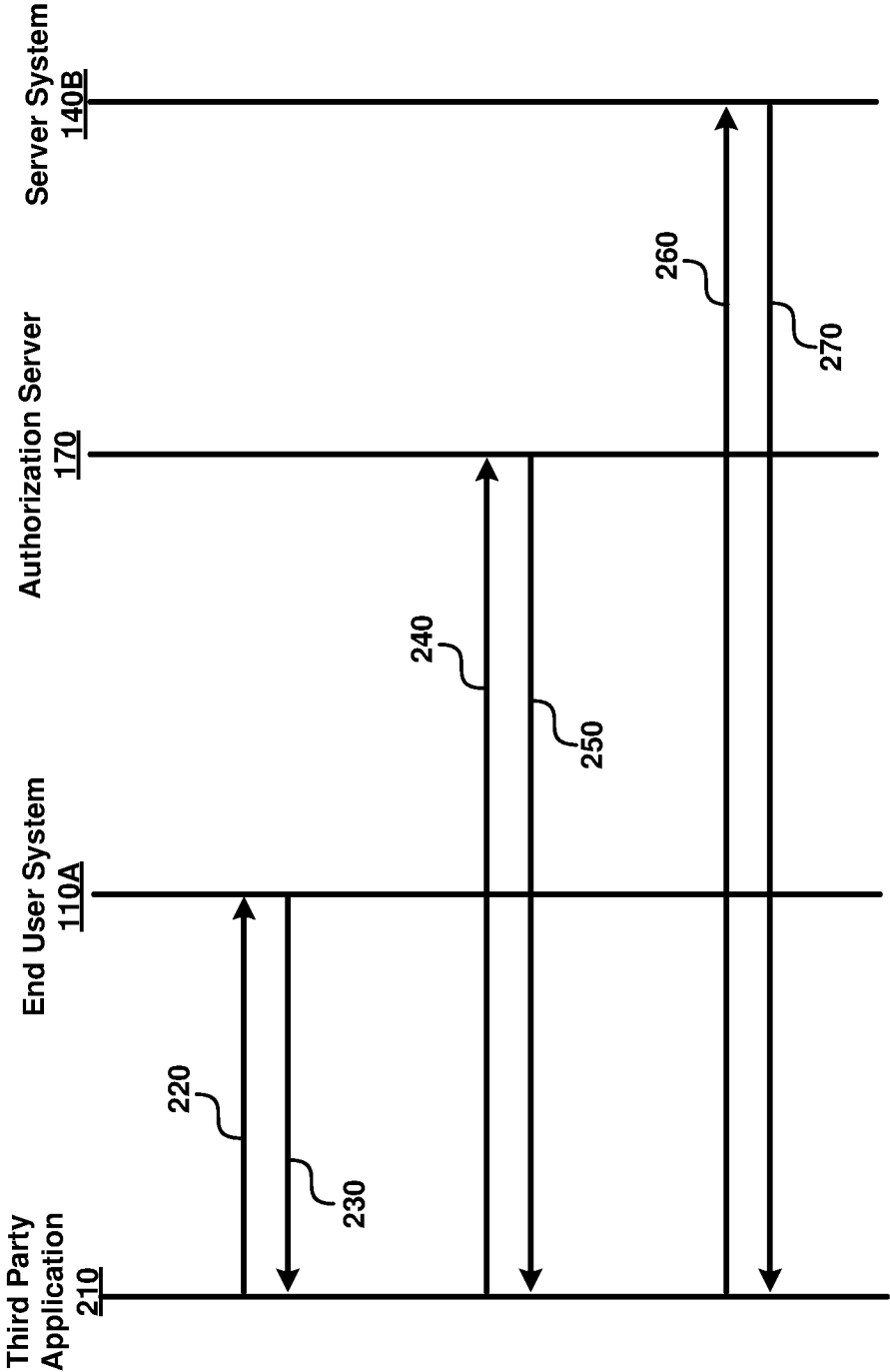
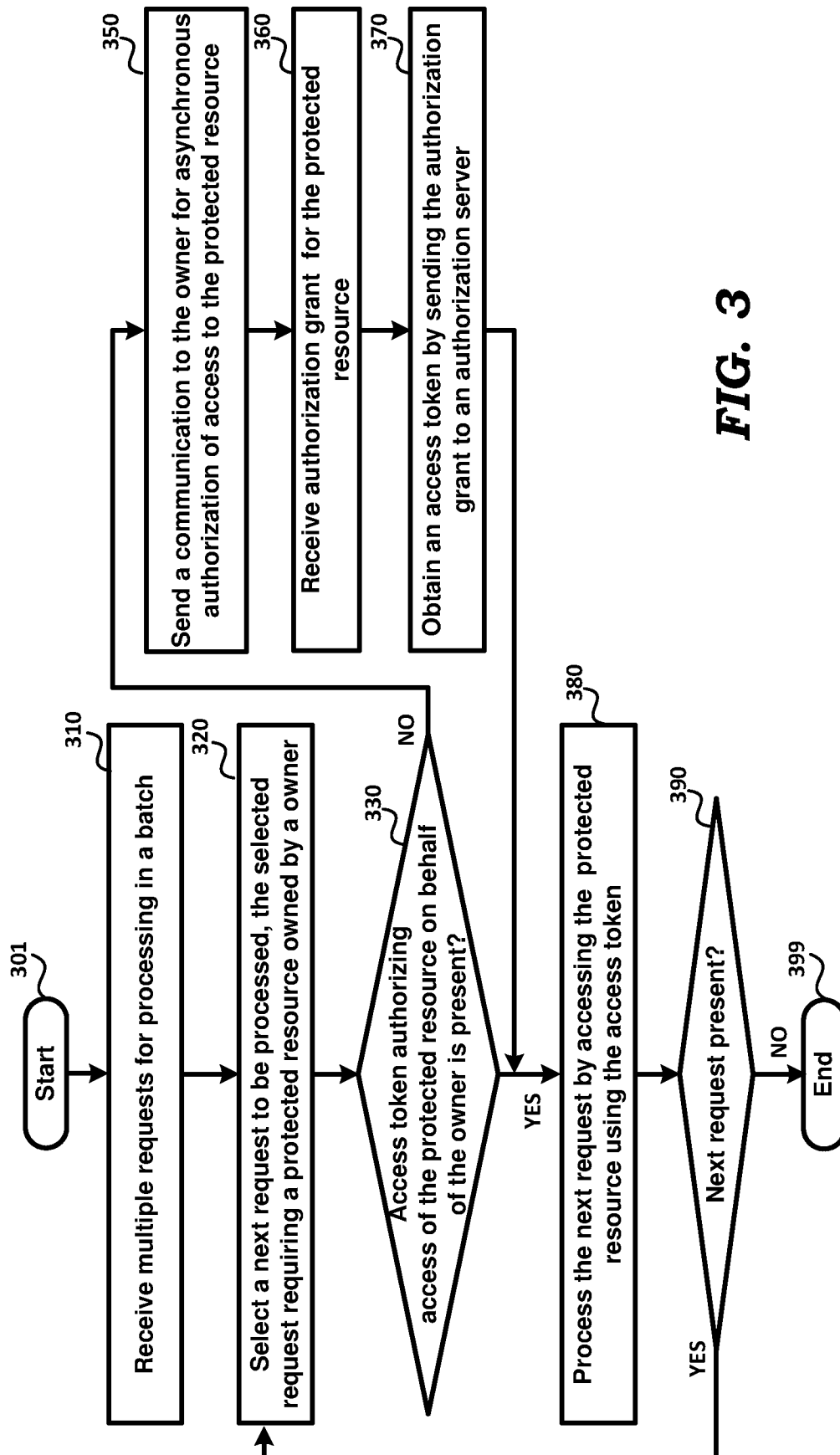


FIG. 2

**FIG. 3**

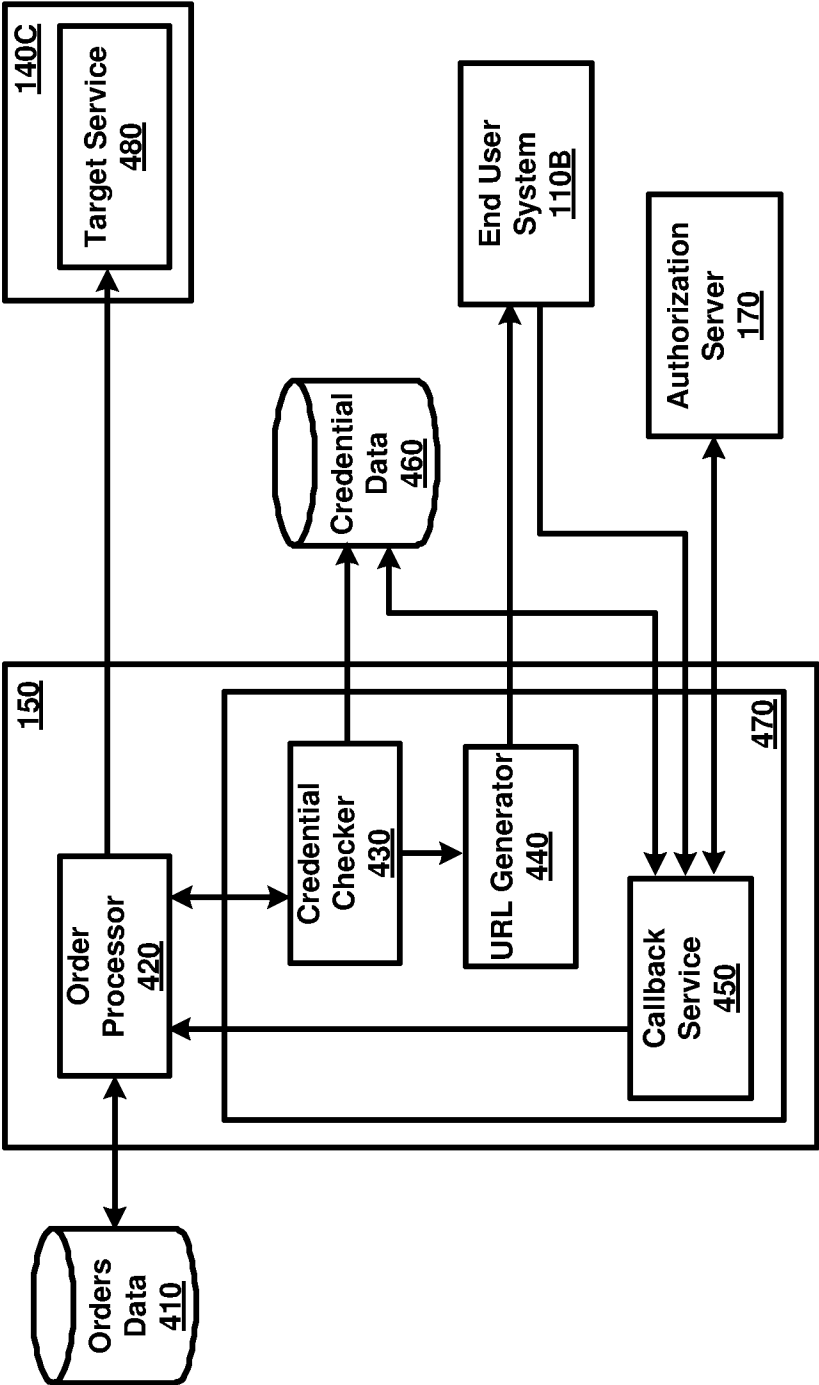


FIG. 4A

460

User ID	Access Token	Expiry Date	Request Date	Status
11212	ECEBDFB6EAAF009C975AD91A69AC5B26	12-Jan-15	12-Jan-14	Active
278452	B3FCCDFEF9277B097E0D0933D41DF7F3	30-Oct-14	28-Oct-13	Active
360008	4A41350F38146AE ECB83E737E6A744D1	15-Sep-13	14-Sep-12	Expired
715544	89F7BCEBDDAAC131162357C47B1254DC	20-Sep-14	17-Sep-13	Active
578578			10-Jun-14	Pending

465

490

FIG. 4B

460

User ID	Access Token	Expiry Date	Request Date	Status
11212	ECEBDFB6EAAF009C975AD91A69AC5B26	12-Jan-15	12-Jan-14	Active
278452	B3FCCDFEF9277B097E0D0933D41DF7F3	30-Oct-14	30-Oct-13	Active
360008	4A41350F38146AE ECB83E737E6A744D1	15-Sep-13	15-Sep-12	Expired
715544	89F7BCEBDDAAC131162357C47B1254DC	20-Sep-14	17-Sep-13	Active
578578	7A9426E1E4A433282D02968CC7FF4779	12-Jun-15	10-Jun-14	Active

495

FIG. 4C

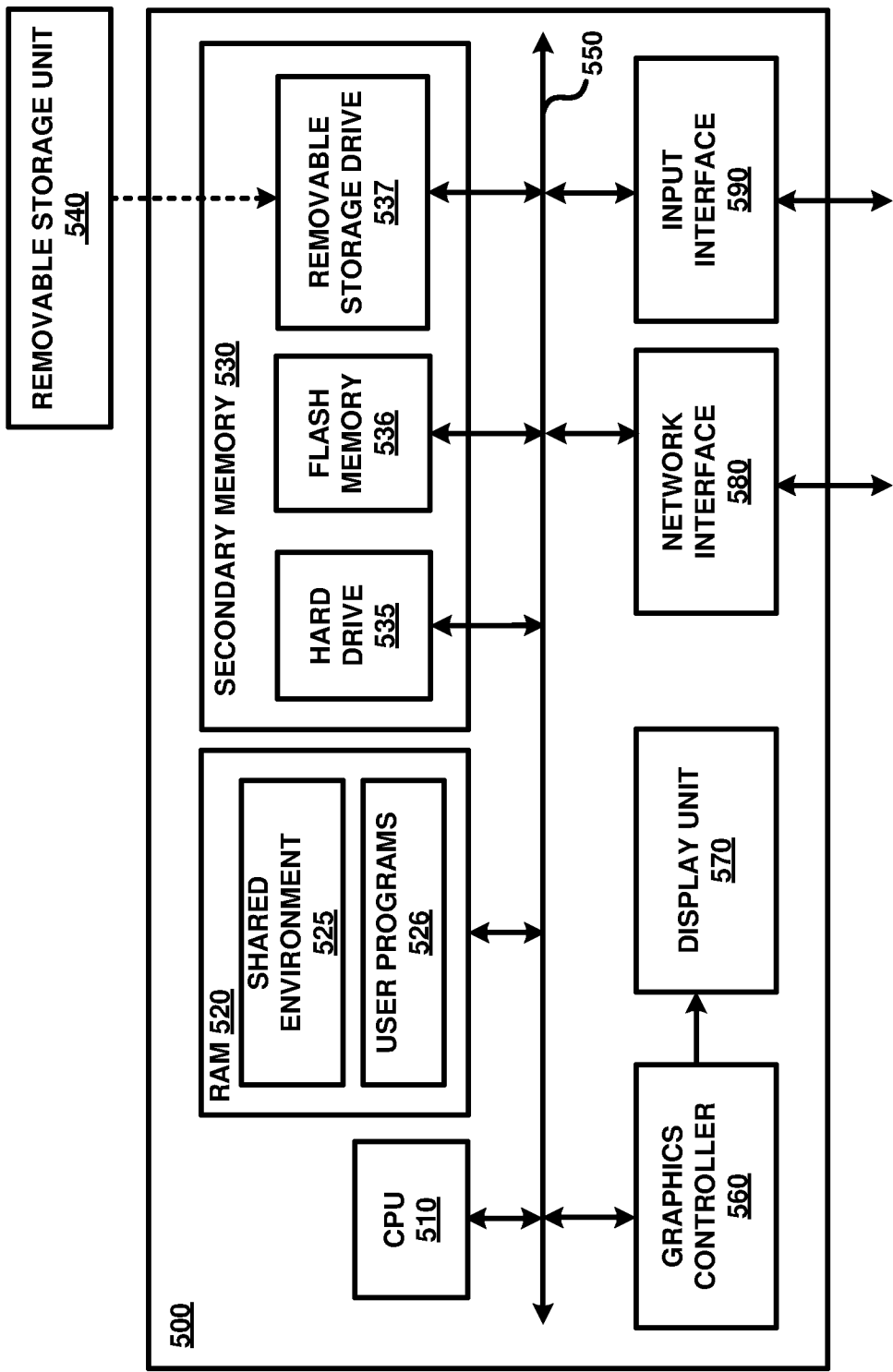


FIG. 5

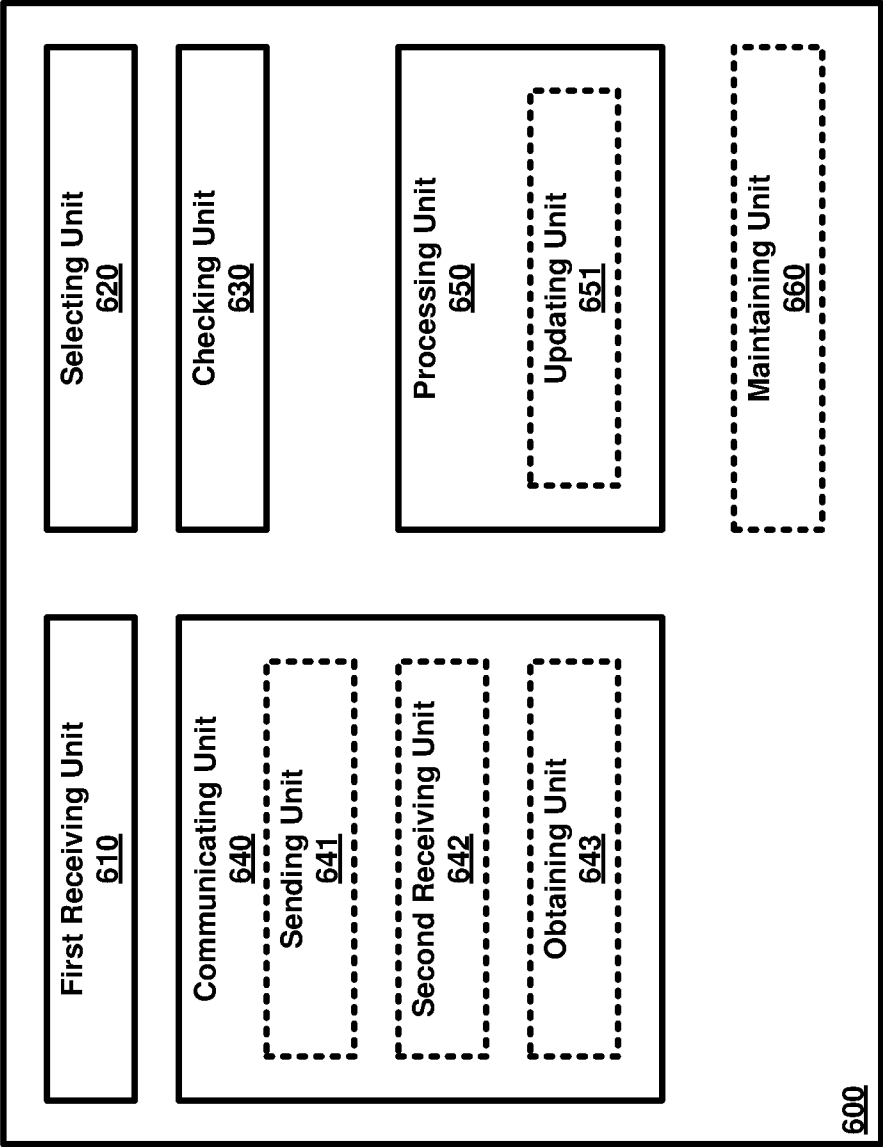


FIG. 6

INTERNATIONAL SEARCH REPORT

International application No

PCT/IB2015/050845

A. CLASSIFICATION OF SUBJECT MATTER
INV. H04L29/06
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 G06F

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, WPI Data, INSPEC, COMPENDEX, IBM-TDB

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2013/139241 A1 (LEEDER MICHAEL [CA]) 30 May 2013 (2013-05-30) abstract paragraph [0019] - paragraph [0028] paragraph [0029] - paragraph [0039] figures 1, 2A, 2B	1-27
X	US 2013/019295 A1 (PARK SUNG-JIN [KR] ET AL) 17 January 2013 (2013-01-17) abstract paragraph [0045] - paragraph [0046] paragraph [0062] - paragraph [0071] figures 3A, 5	1-27



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

20 May 2015

Date of mailing of the international search report

27/05/2015

Name and mailing address of the ISA/

European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040,
Fax: (+31-70) 340-3016

Authorized officer

Poppe, Fabrice

INTERNATIONAL SEARCH REPORT

International application No

PCT/IB2015/050845

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X A	JP 2013 246655 A (CANON KK) 9 December 2013 (2013-12-09) the whole document	1,8,14, 21 2-7, 9-13, 15-20, 22-27
X,P A,P	& US 2014/230020 A1 (MOGAKI SHUNSUKE [JP]) 14 August 2014 (2014-08-14) abstract paragraph [0057] - paragraph [0059] paragraph [0067] - paragraph [0068] paragraph [0088] - paragraph [0115] figures 1, 5, 10A, 10B -----	1,8,14, 21 2-7, 9-13, 15-20, 22-27
A	LODDERSTEDT T ET AL: "OAuth 2.0 Threat Model and Security Considerations", RFC 6819, INTERNET ENGINEERING TASK FORCE, IETF, 7 January 2013 (2013-01-07), XP015086521, [retrieved on 2013-01-07] 2. Overview 2.3. Architectural Assumptions 3. Security Features 3.2. Access Token 4. Threat Model 4.3. Token Endpoint -----	1-27
A	HARDT D ET AL: "The OAuth 2.0 Authorization Framework", RFC 6749, INTERNET ENGINEERING TASK FORCE, IETF, 13 October 2012 (2012-10-13), XP015086448, [retrieved on 2012-10-13] the whole document -----	1-27

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No

PCT/IB2015/050845

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2013139241 A1	30-05-2013	NONE	
US 2013019295 A1	17-01-2013	KR 20130007797 A US 2013019295 A1	21-01-2013 17-01-2013
JP 2013246655 A	09-12-2013	CN 104350501 A JP 2013246655 A KR 20150013855 A US 2014230020 A1 WO 2013175901 A1	11-02-2015 09-12-2013 05-02-2015 14-08-2014 28-11-2013