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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
04 June 2020 (04.06.2020)

WIPO | PCT

(10) International Publication Number
WO 2020/112989 A1

(51) International Patent Classification:
H04L 29/06 (2006.01)

(21) International Application Number:
PCT/US20 19/063 592

(22) International Filing Date:
27 November 2019 (27. 11.2019)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
62/773,860 30 November 2018 (30. 11.2018) US

(71) Applicant: **JPMORGAN CHASE BANK, N.A.** [US/US]; 383 Madison Avenue, New York, New York 10179 (US).

(72) Inventors: **VUDATHU, Raghuram;** c/o JPMorgan Chase Bank, N.A., 383 Madison Avenue, New York, New York 10179 (US). **SPECTOR, Howard;** c/o JPMorgan Chase Bank, N.A., 383 Madison Avenue, New York, New York 10179 (US).

(74) Agent: **KING, Robert A.;** c/o Greenberg Traurig, LLP, 77 West Wacker Drive, Suite 3100, Intellectual Property Department, Chicago, Illinois 60601 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA,

(54) **Title:** SYSTEMS AND METHODS FOR SECURELY CALLING APIS ON AN API GATEWAY FROM APPLICATIONS NEEDING FIRST PARTY AUTHENTICATION

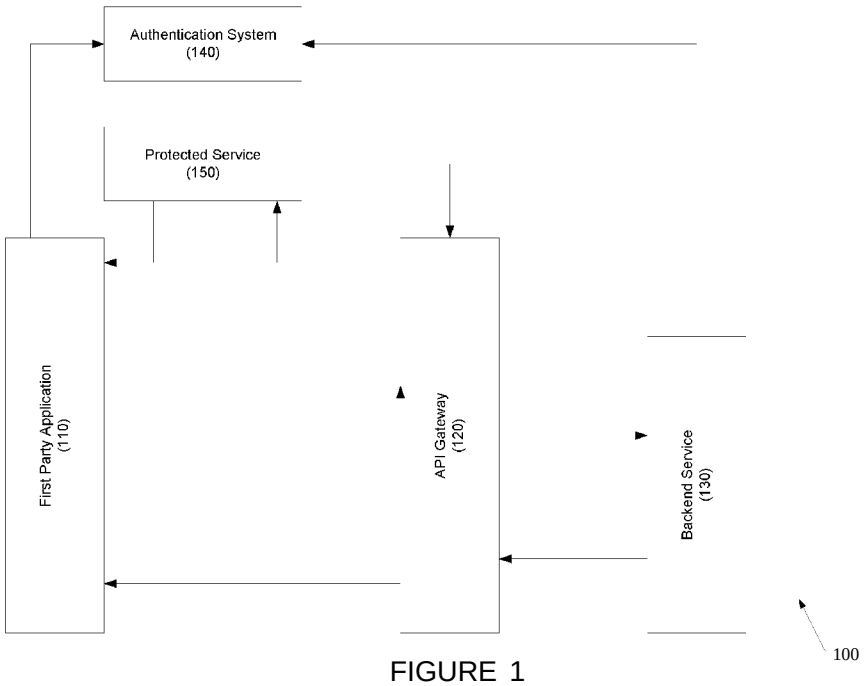

FIGURE 1

(57) Abstract: Systems and methods for securely calling APIs on an API gateway from applications that need first party authentication are disclosed. In one embodiment, a method may include: (1) receiving, from a protected service, an authentication system token/cookie identifier, a first plurality of user identifying attributes, and a request to create an oAuth access token; (2) creating an attribute string; (3) encrypting the attribute string with a private key, resulting in the oAuth access token; (4) sending the oAuth access token to the first party computer application; (5) receiving, from the first party computer application, a request to access a backend service, a second plurality of user identifying attributes, and the oAuth access token; (6) decrypting the oAuth access token; (7) validating the decrypted oAuth access token; (8) inserting the authentication system token/cookie identifier into the request to access; and (9) communicating

[Continued on next page]

WO 2020/112989 A1

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, Cl, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:

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

SYSTEMS AND METHODS FOR SECURELY CALLING APIS ON AN API GATEWAY FROM APPLICATIONS NEEDING FIRST PARTY AUTHENTICATION

RELATED APPLICATIONS

[0001] This application claims the benefit of U.S. Provisional **Patent Application Serial No.** 62/773,860, filed November 30, 2018, the disclosure of which is hereby incorporated, by reference, in its entirety.

BACKGROUND OF THE INVENTION

1. Field of the Invention

[0002] Embodiments generally relate to systems and methods for securely calling APIs on an API gateway applications that need first party authentication.

2. Description of the Related Art

[0003] Authentication systems, Single Sign On systems, and Active Directory systems provide centralized web access management systems that enable user authentication and single sign-on, policy-based authorization, identity federation, and auditing of access to Web applications and portals. The use of these systems has made it a common practice and industry standard to use tools and frameworks for first party authentication.

[0004] First party computer applications (e.g., mobile apps / web applications that are deployed in the same domain) cannot directly call Application Programmable Interfaces, or APIs (e.g., REST Services, or SOAP Services, etc.) hosted on an API Gateway due to the lack of an authentication and authorization concerns built into the API Gateway type

tools. The services hosted on API Gateway and API Gateway itself, relies on an external entity or a layer such as SiteMinder or some other equivalent tools or frameworks to authenticate the user.

[0005] The integration of any authentication framework into API Gateway is not ideal because of many architectural reasons. For example: (1) the API Gateway is an appliance, not an Active Directory / SiteMinder type authentication software/service; (2) the API Gateway is stateless and does not maintain a state or a session or a user context for the logged in user (i.e., the API Gateway can maintain the user context for a stateless user using token references); (3) the API Gateway does not have a persistence layer built for application concerns; (4) the API Gateway is mainly designed for stateless systems, to route the requests, throttle the requests, terminate the SSL, protocol transformation and content-based routing; and (5) the API Gateway works mainly with oAuth tokens.

[0006] Typically, mobile applications and websites have an authentication system that protects all the calls going to the back-end or middleware systems. The authentication system intercepts the requests and checks for a certain type of data that can only be obtained after the user logs into the authentication system. If that certain type of data is not found, the request gets forwarded to an authentication system, which challenges the user to log in to the application with proper credentials.

[0007] In this set up, one cannot leverage the API Gateway without creating the authentication layer in front of the API Gateway.

[0008] The industry standard way of integrating with an API Gateway is either with 2 legged oAuth or 3 legged oAuth.

SUMMARY OF THE INVENTION

[0009] Systems and methods for securely calling APIs on an API gateway from applications that need first party authentication are disclosed. In one embodiment, in an API gateway comprising at least one computer processor, a method for securely calling APIs on an API gateway from computer applications that need first party authentication may include: (1) receiving, from a protected service, an authentication system token or an authentication system cookie identifier from an authentication system, a first plurality of user identifying attributes, and a request to create an oAuth access token, the request originating with a first party computer application; (2) creating the oAuth access token and an attribute string comprising at least one of the first plurality of user identifying attributes and the authentication system token or the authentication system cookie identifier; (3) encrypting the attribute string with a private key, resulting in the oAuth access token; (4) sending the oAuth access token to the first party computer application; (5) receiving, from the first party computer application, a request to access a backend service, a second plurality of user identifying attributes, and the oAuth access token; (6) decrypting the oAuth access token with the private key; (7) validating the decrypted oAuth access token; (8) inserting the authentication system token or the authentication system cookie identifier into the request to access; and (9) communicating the request to access and the authentication system token or the authentication system cookie identifier to the backend service.

[0010] In one embodiment, the first plurality of user identifying attributes may include at least one of a device mac id, a device manufacturer, a device geo-location, a device operating system, a device operating system version, a device IP address, a user profile id, and a user id. In one

embodiment, the method may further include setting an expiration for the oAuth access token.

[0011] In one embodiment, the step of validating the decrypted oAuth access token may include verifying that the oAuth access token has not expired.

[0012] In one embodiment, the backend service may include a micro service, a SOA service, a REST service, a SOAP service, monolith service, a standard routine, a standard function, a lambda function, or a procedure.

[0013] In one embodiment, a plurality of the user identifying attributes may be concatenated in the attribute string in a random order, in a rotating order, etc.

[0014] In one embodiment, the step of validating the decrypted oAuth access token may include comparing the extracted values from the oAuth token to the second plurality of user identifying attributes.

[0015] In one embodiment, the backend service may call the authentication system to check if the authentication system token or the authentication system cookie identifier is valid, and the method may further include: receiving an error from the backend system in response to the authentication system token or the authentication system cookie identifier being invalid; and sending an access grant denied error to the first party computer application.

[0016] According to another embodiment, a system for securely calling APIs on an API gateway from computer applications that need first party authentication may include: a first party computer application; an authentication system; a protected service; an API gateway; and a backend

service. The authentication system may authenticate a user logging in to the first party computer application, and may create a session and returns session details to the first party computer application. The protected service may receive a request involving the backend service from the first party computer application and a first plurality of user identifying attributes, and may call the API gateway to create an oAuth access token and the first plurality of user identifying attributes. The API gateway may create the oAuth access token and an attribute string comprising at least one of the first plurality of user identifying attributes and the authentication system token or the authentication system cookie identifier; may encrypt the attribute string with a private key, resulting in the oAuth access token; may send the oAuth access token to the first party computer application; may receive, from the first party computer application, a request to access the backend service, a second plurality of user identifying attributes, and the oAuth access token; may decrypt the oAuth access token with the private key; may validate the decrypted oAuth access token; may insert the authentication system token or the authentication system cookie identifier into the request to access; and may communicate the request to access and the authentication system token or the authentication system cookie identifier to the backend service.

[0017] In one embodiment, the first plurality of user identifying attributes may include at least one of a device mac id, a device manufacturer, a device geo-location, a device operating system, a device operating system version, a device IP address, a user profile id, and a user id.

[0018] In one embodiment, the API gateway may set an expiration for the oAuth access token.

[0019] In one embodiment, the API gateway may validate the decrypted oAuth access token by verifying that the oAuth access token has not expired.

[0020] In one embodiment, the backend service may include a micro service, a SOA service, a REST service, a SOAP service, monolith service, a standard routine, a standard function, a lambda function, or a procedure.

[0021] In one embodiment, a plurality of the user identifying attributes may be concatenated in the attribute string in a random order, in a rotating order, etc.

[0022] In one embodiment, the API gateway may validate the decrypted oAuth access token by comparing the extracted values from the oAuth token to the second plurality of user identifying attributes.

[0023] In one embodiment, the backend service may call the authentication system to check if the authentication system token or the authentication system cookie identifier is valid, and the API gateway may receive an error from the backend system in response to the authentication system token or the authentication system cookie identifier being invalid and may send an access grant denied error to the first party computer application.

BRIEF DESCRIPTION OF THE DRAWINGS

[0024] In order to facilitate a fuller understanding of the present invention, reference is now made to the attached drawings. The drawings should not be construed as limiting the present invention but are intended only to illustrate different aspects and embodiments.

[0025] Figure 1 depicts a system for securely calling APIs on an API gateway from applications that need first party authentication, according to one embodiment; and

[0026] Figure 2 depicts a method for securely calling APIs on an API gateway from applications that need first party authentication according to one embodiment.

DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS

[0027] Embodiments are directed to systems and methods for securely calling APIs on an API gateway from that need first party authentication, web applications and any type of UI applications that will need first party authentication. Embodiments may be extended to non-UI applications including, for example, IoT type devices, sensors and other similar devices trying to send or receive data from the middleware systems or backend servers.

[0028] Referring to Figure 1, a system for securely calling APIs on an API gateway from applications that need first party authentication is disclosed according to one embodiment. System 100 may include first party computer application 100, which may be a computer program or application executed on any suitable electronic device (e.g., smart phone, tablet computer, smart watch, notebook computer, desktop computer, workstation, Internet of Things (IoT) appliance, etc.

[0029] Examples of first party computer application 110 include mobile applications, web applications, fat client applications, thin client applications, desktop applications, native applications, IoT applications, IoT sensors, etc.

[0030] First party computer application 110 may communicate with API gateway 120, authentication system 140, and protected service 150, such as a Get oAuth Access Token service. In one embodiment, API gateway 120 may provide first party computer application 110 with access to one or more APIs, and may enable developers to create, publish, maintain, monitor, and secure APIs.

[0031] In one embodiment, authentication system 140 may be any suitable authentication system, including, for example, Site Minder, Active Directory, Single Sign On using SAML, Open ID, etc., or any similar system that authenticates the user and keeps and/or maintains the same state or session of the user in some form.

[0032] API gateway 120 may communicate with backend service 130 which may be, for example, a micro service, a SOA service, a REST service, a SOAP service, monolith service, a standard routine, a standard function, a lambda function, a procedure, etc.

[0033] Referring to Figure 2, a method for securely calling APIs on an API gateway from applications that need first party authentication is disclosed according to one embodiment.

[0034] In step 205, a user may log in to a first party computer application using the user's credentials (e.g., username and password), and the first party computer application may submit the credentials to an authentication system.

[0035] In step 210, after the user is logged in successfully, the authentication system may create a session, and may return the session details to the first party computer application.

[0036] For example, once the user is authenticated, an authentication system token (e.g., authentication-system-token), an authentication system session identifier (e.g., authentication- system- session-id), or an authentication system cookie identifier (e.g., authentication-system-cookie-id) may be provided as part of the session details.

[0037] In step 215, the first party computer application may call a protected service that is protected by the authentication system. For example, if the first party computer application tries to call this protected service without actually logging in, the user will be re-directed to the authentication system, which challenges the user to log in with their login credentials (e.g., user name and password) and log the user to the application.

[0038] The protected service may be, for example, a Get oAuth Access Token service that may be deployed to a web server. An agent or other authentication plugin may be deployed that that intercepts the request coming to the protected service, and that checks if the authentication system token, the authentication system session identifier, or the authentication system cookie identifier is present and valid.

[0039] If the authentication system token, the authentication system session identifier, or the authentication system cookie identifier is missing, the request may be redirected to the authentication system.

[0040] The first party computer application may send certain one or more attributes that uniquely identify the user, such as the device mac id, the device manufacturer, the device geo-location, the device operating system (e.g., Android, iOS, Windows, Mac, Chrome, etc.), the device operating system version, a device IP address, a user profile id, a user id, etc. that

identify the user and the device to the protected service in the request, to know/audit who the token is being issued to, for security purposes.

[0041] In step 220, the protected service may call the API gateway to create an oAuth access token using these attributes and an authentication system token and/or the authentication system cookie identifier.

[0042] In step 225, the API gateway may create an oAuth access token. In one embodiment, the API gateway may also (1) set an expiration time for the oAuth access token; (2) concatenate some or all the attributes (e.g., device id, device IP, user profile id, user id, the authentication system token or the authentication system cookie identifier, device mac id, device manufacturer, device geo-location, operating system (e.g., Android, iOS. Windows, Mac, Chrome, etc.), operating system version) in, for example, a default order, a random order, a rotating order, etc. to create an attribute string; and (3) encrypt the attribute string with a private key. The encrypted attribute string is the oAuth-access-token, also known as an "oAuth access token".

[0043] **The private key may be secured and may not be shared with anyone. In one embodiment, no public key is generated.**

[0044] In step 230, the encrypted oAuth access token may be returned to the first party computer application via the protected service as a response to the backend service call.

[0045] In step 235, the first party computer application may store or cache the oAuth access token so that it is available to the entire first application, so that any process in the first party computer application may access the oAuth access token.

[0046] In step 240, when the first party computer application is ready to call a backend service that is hosted on the API gateway that is not protected by any traditional authentication system, the first party computer application may send the request or payload details along with, for example, the device id, device IP, user profile id, user id, device mac id, device manufacturer, device geo-location, operating system (e.g., Android, iOS. Windows, Mac, Chrome, etc.), operating system version), operating system version, and the oAuth access token to the API gateway in plain text.

[0047] In step 245, when the API gateway receives the request, it may: (1) decrypt the oAuth access token with the private key; (2) check the validity of the oAuth access token - check the expiration timestamp of the oAuth access token (if the oAuth access token is expired, the API gateway will send an access grant denied error to the first party computer application); (3) unbundle the string and extract all the attributes (e.g., the device id, device IP, user profile id, user id, authentication system cookie id, device mac id, device manufacturer, device geo-location, operating system (e.g., Android, iOS. Windows, Mac, Chrome, etc.), operating system version, etc.); (4) compare the extracted values to the values received from the first party computer application in plain text at step 240.

[0048] If any of the values mismatch during the comparison, the API gateway will stop further processing and send an access grant denied error to first party computer application.

[0049] If all the values match and the oAuth access token is valid, then the API routes the request to the backend service.

[0050] In step 250, before routing the request or payload to the backend service, the API gateway may inject the authentication system

cookie id or authentication system session id it extracted from the oAuth access token into the request or payload.

[0051] In step 255, the backend service may process the request or payload. For example, the request or payload may involve persisting data, calling other subsystems or downstream systems, making asynchronous calls, etc., and may create a response to send to the first party computer application.

[0052] In step 260, the backend service may call the authentication system to check if the authentication system cookie id or the authentication system session id is still valid (i.e., the user is still logged in and the user session is still active from the authentication system perspective). Steps 255 and 260 may be performed in parallel.

[0053] In step 265, the authentication system may provide a response to the backend service indication whether the authentication system cookie id or the authentication system session id is valid.

[0054] If, in step 270, the authentication system responds that the authentication system cookie id or the authentication system session id is valid, in step 275, the backend service may send the response back to the first party computer application.

[0055] If the authentication system responds that the authentication system cookie id or the authentication system session id is invalid (e.g. the user already logged out, or if the user session timed our, or if the browser session is closed for some other reason), in step 280, the backend system may send an error to the API gateway, and in step 285, the API gateway may, in turn, send an access grant denied error to the first party computer application.

[0056] Although several embodiments have been disclosed, it should be recognized that these embodiments are not exclusive to each other, and certain elements or features from one embodiment may be used with another.

[0057] Hereinafter, general aspects of implementation of the systems and methods of the invention will be described.

[0058] The system of the invention or portions of the system of the invention may be in the form of a "processing machine," such as a general-purpose computer, for example. As used herein, the term "processing machine" is to be understood to include at least one processor that uses at least one memory. The at least one memory stores a set of instructions. The instructions may be either permanently or temporarily stored in the memory or memories of the processing machine. The processor executes the instructions that are stored in the memory or memories in order to process data. The set of instructions may include various instructions that perform a particular task or tasks, such as those tasks described above. Such a set of instructions for performing a particular task may be characterized as a program, software program, or simply software.

[0059] In one embodiment, the processing machine may be a specialized processor.

[0060] As noted above, the processing machine executes the instructions that are stored in the memory or memories to process data. This processing of data may be in response to commands by a user or users of the processing machine, in response to previous processing, in response to a request by another processing machine and/or any other input, for example.

[0061] As noted above, the processing machine used to implement the invention may be a general-purpose computer. However, the processing

machine described above may also utilize any of a wide variety of other technologies including a special purpose computer, a computer system including, for example, a microcomputer, mini-computer or mainframe, a programmed microprocessor, a micro-controller, a peripheral integrated circuit element, a CSIC (Customer Specific Integrated Circuit) or ASIC (Application Specific Integrated Circuit) or other integrated circuit, a logic circuit, a digital signal processor, a programmable logic device such as a FPGA, PLD, PLA or PAL, or any other device or arrangement of devices that is capable of implementing the steps of the processes of the invention.

[0062] The processing machine used to implement the invention may utilize a suitable operating system. Thus, embodiments of the invention may include a processing machine running the iOS operating system, the OS X operating system, the Android operating system, the Microsoft Windows™ operating systems, the Unix operating system, the Linux operating system, the Xenix operating system, the IBM AIX™ operating system, the Hewlett-Packard UX™ operating system, the Novell Netware™ operating system, the Sun Microsystems Solaris™ operating system, the OS/2™ operating system, the BeOS™ operating system, the Macintosh operating system, the Apache operating system, an OpenStep™ operating system or another operating system or platform.

[0063] It is appreciated that in order to practice the method of the invention as described above, it is not necessary that the processors and/or the memories of the processing machine be physically located in the same geographical place. That is, each of the processors and the memories used by the processing machine may be located in geographically distinct locations and connected so as to communicate in any suitable manner. Additionally, it is appreciated that each of the processor and/or the memory

may be composed of different physical pieces of equipment. Accordingly, it is not necessary that the processor be one single piece of equipment in one location and that the memory be another single piece of equipment in another location. That is, it is contemplated that the processor may be two pieces of equipment in two different physical locations. The two distinct pieces of equipment may be connected in any suitable manner. Additionally, the memory may include two or more portions of memory in two or more physical locations.

[0064] To explain further, processing, as described above, is performed by various components and various memories. However, it is appreciated that the processing performed by two distinct components as described above may, in accordance with a further embodiment of the invention, be performed by a single component. Further, the processing performed by one distinct component as described above may be performed by two distinct components. In a similar manner, the memory storage performed by two distinct memory portions as described above may, in accordance with a further embodiment of the invention, be performed by a single memory portion. Further, the memory storage performed by one distinct memory portion as described above may be performed by two memory portions.

[0065] Further, various technologies may be used to provide communication between the various processors and/or memories, as well as to allow the processors and/or the memories of the invention to communicate with any other entity; i.e., so as to obtain further instructions or to access and use remote memory stores, for example. Such technologies used to provide such communication might include a network, the Internet, Intranet, Extranet, LAN, an Ethernet, wireless communication via cell tower or satellite, or any client server system that provides communication, for

example. Such communications technologies may use any suitable protocol such as TCP/IP, UDP, or OSI, for example.

[0066] As described above, a set of instructions may be used in the processing of the invention. The set of instructions may be in the form of a program or software. The software may be in the form of system software or application software, for example. The software might also be in the form of a collection of separate programs, a program module within a larger program, or a portion of a program module, for example. The software used might also include modular programming in the form of object oriented programming. The software tells the processing machine what to do with the data being processed.

[0067] Further, it is appreciated that the instructions or set of instructions used in the implementation and operation of the invention may be in a suitable form such that the processing machine may read the instructions. For example, the instructions that form a program may be in the form of a suitable programming language, which is converted to machine language or object code to allow the processor or processors to read the instructions. That is, written lines of programming code or source code, in a particular programming language, are converted to machine language using a compiler, assembler or interpreter. The machine language is binary coded machine instructions that are specific to a particular type of processing machine, i.e., to a particular type of computer, for example. The computer understands the machine language.

[0068] Any suitable programming language may be used in accordance with the various embodiments of the invention. Illustratively, the programming language used may include assembly language, Ada, APL,

Basic, C, C++, COBOL, dBase, Forth, Fortran, Java, Modula-2, Pascal, Prolog, REXX, Visual Basic, and/or JavaScript, for example. Further, it is not necessary that a single type of instruction or single programming language be utilized in conjunction with the operation of the system and method of the invention. Rather, any number of different programming languages may be utilized as is necessary and/or desirable.

[0069] Also, the instructions and/or data used in the practice of the invention may utilize any compression or encryption technique or algorithm, as may be desired. An encryption module might be used to encrypt data. Further, files or other data may be decrypted using a suitable decryption module, for example.

[0070] As described above, the invention may illustratively be embodied in the form of a processing machine, including a computer or computer system, for example, that includes at least one memory. It is to be appreciated that the set of instructions, i.e., the software for example, that enables the computer operating system to perform the operations described above may be contained on any of a wide variety of media or medium, as desired. Further, the data that is processed by the set of instructions might also be contained on any of a wide variety of media or medium. That is, the particular medium, i.e., the memory in the processing machine, utilized to hold the set of instructions and/or the data used in the invention may take on any of a variety of physical forms or transmissions, for example. Illustratively, the medium may be in the form of paper, paper transparencies, a compact disk, a DVD, an integrated circuit, a hard disk, a floppy disk, an optical disk, a magnetic tape, a RAM, a ROM, a PROM, an EPROM, a wire, a cable, a fiber, a communications channel, a satellite transmission, a memory card, a SIM card, or other remote transmission, as well as any other

medium or source of data that may be read by the processors of the invention.

[0071] Further, the memory or memories used in the processing machine that implements the invention may be in any of a wide variety of forms to allow the memory to hold instructions, data, or other information, as is desired. Thus, the memory might be in the form of a database to hold data. The database might use any desired arrangement of files such as a flat file arrangement or a relational database arrangement, for example.

[0072] In the system and method of the invention, a variety of "user interfaces" may be utilized to allow a user to interface with the processing machine or machines that are used to implement the invention. As used herein, a user interface includes any hardware, software, or combination of hardware and software used by the processing machine that allows a user to interact with the processing machine. A user interface may be in the form of a dialogue screen for example. A user interface may also include any of a mouse, touch screen, keyboard, keypad, voice reader, voice recognizer, dialogue screen, menu box, list, checkbox, toggle switch, a pushbutton or any other device that allows a user to receive information regarding the operation of the processing machine as it processes a set of instructions and/or provides the processing machine with information. Accordingly, the user interface is any device that provides communication between a user and a processing machine. The information provided by the user to the processing machine through the user interface may be in the form of a command, a selection of data, or some other input, for example.

[0073] As discussed above, a user interface is utilized by the processing machine that performs a set of instructions such that the

processing machine processes data for a user. The user interface is typically used by the processing machine for interacting with a user either to convey information or receive information from the user. However, it should be appreciated that in accordance with some embodiments of the system and method of the invention, it is not necessary that a human user actually interact with a user interface used by the processing machine of the invention. Rather, it is also contemplated that the user interface of the invention might interact, i.e., convey and receive information, with another processing machine, rather than a human user. Accordingly, the other processing machine might be characterized as a user. Further, it is contemplated that a user interface utilized in the system and method of the invention may interact partially with another processing machine or processing machines, while also interacting partially with a human user.

[0074] It will be readily understood by those persons skilled in the art that the present invention is susceptible to broad utility and application. Many embodiments and adaptations of the present invention other than those herein described, as well as many variations, modifications and equivalent arrangements, will be apparent from or reasonably suggested by the present invention and foregoing description thereof, without departing from the substance or scope of the invention.

Accordingly, while the present invention has been described here in detail in relation to its exemplary embodiments, it is to be understood that this disclosure is only illustrative and exemplary of the present invention and is made to provide an enabling disclosure of the invention. Accordingly, the foregoing disclosure is not intended to be construed or to limit the present invention or otherwise to exclude any other such embodiments, adaptations, variations, modifications or equivalent arrangements.

CLAIMS

What is claimed is:

1. A method for securely calling APIs on an API gateway from computer applications that need first party authentication, comprising:

in an API gateway comprising at least one computer processor:

receiving, from a protected service, an authentication system token or an authentication system cookie identifier from an authentication system, a first plurality of user identifying attributes, and a request to create an oAuth access token, the request originating with a first party computer application;

creating an attribute string comprising at least one of the first plurality of user identifying attributes and the authentication system token or the authentication system cookie identifier;

encrypting the attribute string with a private key, resulting in the oAuth access token;

sending the oAuth access token to the first party computer application;

receiving, from the first party computer application, a request to access a backend service, a second plurality of user identifying attributes, and the oAuth access token;

decrypting the oAuth access token with the private key;

validating the decrypted oAuth access token;

inserting the authentication system token or the authentication system cookie identifier into the request to access; and

communicating the request to access and the authentication system token or the authentication system cookie identifier to the backend service.

2. The method of claim 1, wherein the first plurality of user identifying attributes comprise at least one of a device mac id, a device manufacturer, a device geo-location, a device operating system, a device operating system version, a device IP address, a user profile id, and a user id.

3. The method of claim 1, further comprising:
setting an expiration for the oAuth access token.

4. The method of claim 3, wherein the step of validating the decrypted oAuth access token comprises verifying that the oAuth access token has not expired.

5. The method of claim 1, wherein the backend service comprises a micro service, a SOA service, a REST service, a SOAP service, monolith service, a standard routine, a standard function, a lambda function, or a procedure.

6. The method of claim 1, wherein a plurality of the user identifying attributes are concatenated in the attribute string in a random order.

7. The method of claim 1, wherein a plurality of the user identifying attributes are concatenated in the attribute string in a rotating order.

8. The method of claim 1, wherein the step of validating the decrypted oAuth access token comprises comparing the extracted values from the oAuth token to the second plurality of user identifying attributes.

9. The method of claim 1, wherein the backend service calls the authentication system to check if the authentication system token or the authentication system cookie identifier is valid, and further comprising:

receiving an error from the backend system in response to the authentication system token or the authentication system cookie identifier being invalid; and

sending an access grant denied error to the first party computer application.

10. A system for securely calling APIs on an API gateway from computer applications that need first party authentication, comprising:

a first party computer application;

an authentication system;

a protected service;

an API gateway; and

a backend service;

wherein:

the authentication system authenticates a user logging in to the first party computer application;

the authentication system creates a session and returns session details to the first party computer application;

the protected service receives a request involving the backend service from the first party computer application and a first plurality of user identifying attributes;

the protected service calls the API gateway to create an oAuth access token and the first plurality of user identifying attributes;

the API gateway creates an attribute string comprising at least one of the first plurality of user identifying attributes and the authentication system token or the authentication system cookie identifier;

the API gateway encrypts the attribute string with a private key, resulting in the oAuth access token;

the API gateway sends the oAuth access token to the first party computer application;

the API gateway receives, from the first party computer application, a request to access the backend service, a second plurality of user identifying attributes, and the oAuth access token;

the API gateway decrypts the oAuth access token with the private key;

the API gateway validates the decrypted oAuth access token;

the API gateway inserts the authentication system token or the authentication system cookie identifier into the request to access; and

the API gateway communicates the request to access and the authentication system token or the authentication system cookie identifier to the backend service.

11. The system of claim 10, wherein the first plurality of user identifying attributes comprise at least one of a device mac id, a device manufacturer, a device geo-location, a device operating system, a device operating system version, a device IP address, a user profile id, and a user id.

12. The system of claim 10, wherein the API gateway sets an expiration for the oAuth access token.

13. The system of claim 12, wherein the API gateway validates the decrypted oAuth access token by verifying that the oAuth access token has not expired.

14. The system of claim 12, wherein the backend service comprises a micro service, a SOA service, a REST service, a SOAP service, monolith service, a standard routine, a standard function, a lambda function, or a procedure.

15. The system of claim 10, wherein a plurality of the user identifying attributes are concatenated in the attribute string in a random order.

16. The system of claim 10, wherein a plurality of the user identifying attributes are concatenated in the attribute string in a rotating order.

17. The system of claim 10, wherein the API gateway validates the decrypted oAuth access token by comparing the extracted values from the oAuth token to the second plurality of user identifying attributes.

18. The system of claim 10, wherein:

the backend service calls the authentication system to check if the authentication system token or the authentication system cookie identifier is valid;

the API gateway receives an error from the backend system in response to the authentication system token or the authentication system cookie identifier being invalid; and

the API gateway sends an access grant denied error to the first party computer application.

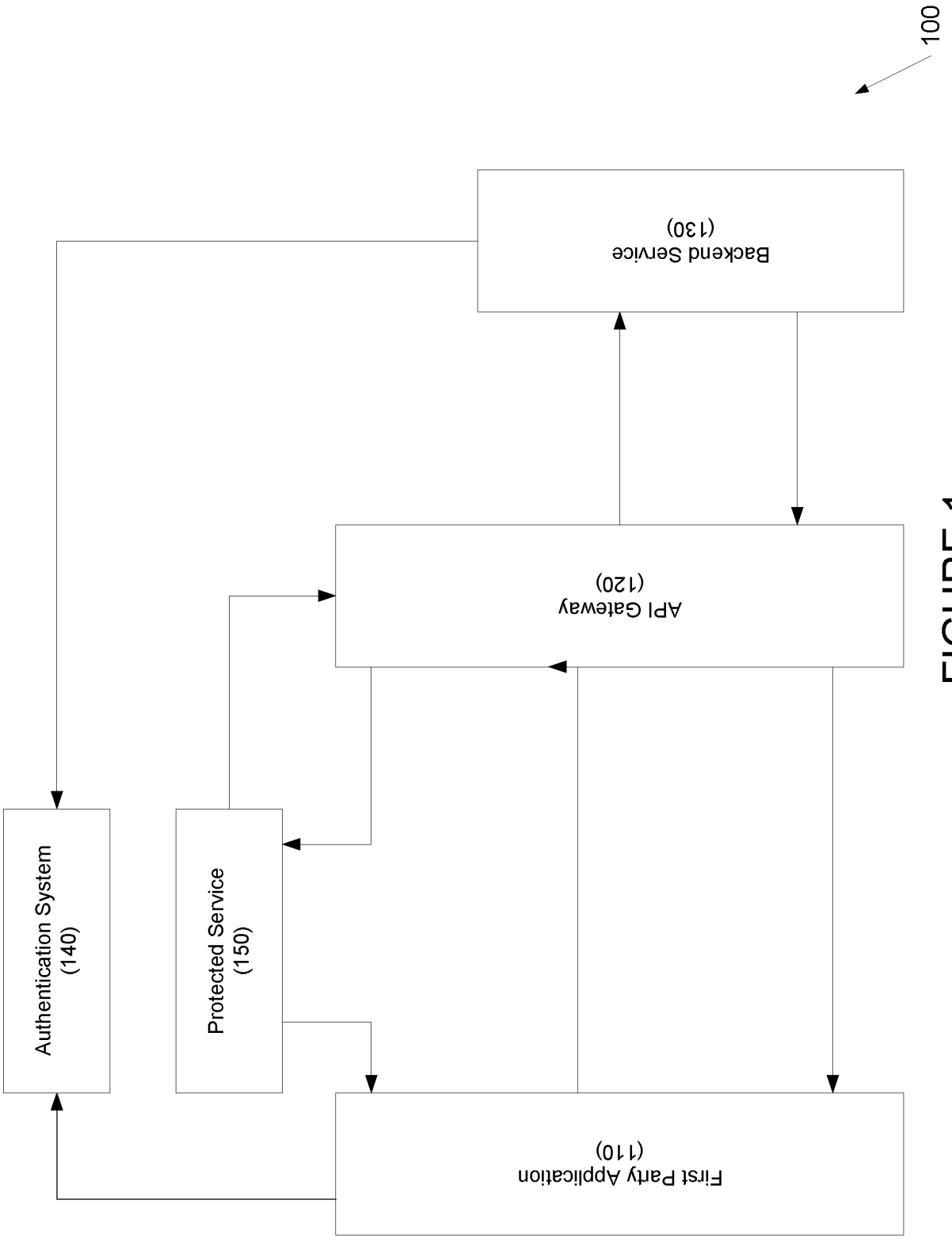
100
Backend Service
(130)
API Gateway
(120)
Authentication System
(140)
Protected Service
(150)
First Party Application
(110)

FIGURE 1

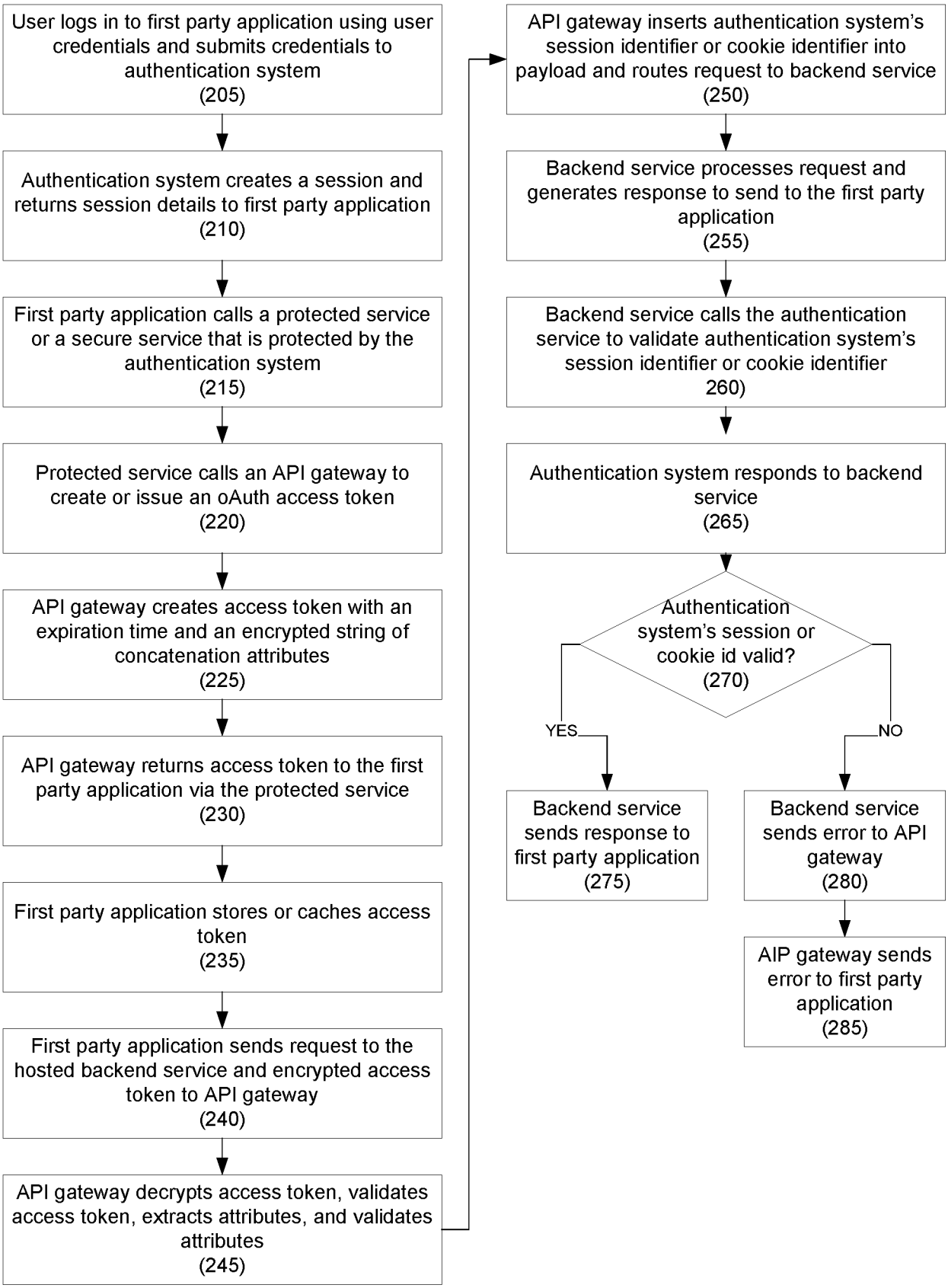
User logs in to first party application using user credentials and submits credentials to authentication system (205)
Authentication system creates a session and returns session details to first party application (210)
First party application calls a protected service or a secure service that is protected by the authentication system (215)
Protected service calls an API gateway to create or issue an oAuth access token (220)
API gateway creates access token with an expiration time and an encrypted string of concatenation attributes (225)
API gateway returns access token to the first party application via the protected service (230)
First party application stores or caches access token (235)
First party application sends request to the hosted backend service and encrypted access token to API gateway (240)
API gateway decrypts access token, validates access token, extracts attributes, and validates attributes (245)
API gateway inserts authentication system's session identifier or cookie identifier into payload and routes request to backend service (250)
Backend service processes request and generates response to send to the first party application (255)
Backend service calls the authentication service to validate authentication system's session identifier or cookie identifier 260)
Authentication system responds to backend service (265)
Authentication system's session or cookie id valid? (270)
YES
NO
Backend service sends response to first party application (275)
Backend service sends error to API gateway (280)
AIP gateway sends error to first party application (285)

FIGURE 2

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2019/063592

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

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

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	Oracle: "Oracle API Gateway OAuth User Guide", , 1 July 2015 (2015-07-01), XP055663910, Retrieved from the Internet: URL:https://docs.oracle.com/cd/E65459_01/user.1112/E65456_01.pdf [retrieved on 2020-01-31] Sections 2 and 3	1-18
A	US 2017/331832 A1 (LANDER VADIM [US] ET AL) 16 November 2017 (2017-11-16) paragraph [0136] - paragraph [0144]	1-18
A	US 2018/007035 A1 (ZHANG JENNY QIAN [US] ET AL) 4 January 2018 (2018-01-04) paragraph [0033] - paragraph [0045] claims 1-3	1-18

-/--

[X] Further documents are listed in the continuation of Box C.

[X] 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
31 January 2020	21/02/2020

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

1

Form PCT/ISA/210 (second sheet) (April 2005)

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2019/063592

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2017/099148 A1 (OCHMANSKI STEVEN R [US] ET AL) 6 April 2017 (2017-04-06) paragraph [0029] - paragraph [0051]	1-18
A	US 2017/111336 A1 (DAVIS CHARLES A [US] ET AL) 20 April 2017 (2017-04-20) paragraph [0055] - paragraph [0066]	1-18

1

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No
PCT/US2019/063592

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2017331832 A1	16-11-2017	NONE	
US 2018007035 A1	04-01-2018	NONE	
US 2017099148 A1	06-04-2017	NONE	
US 2017111336 A1	20-04-2017	NONE	