



- (51) International Patent Classification: *H04L 29/06* (2006.01)
- (21) International Application Number: PCT/IB2018/058692
- (22) International Filing Date: 06 November 2018 (06.11.2018)
- (25) Filing Language: English
- (26) Publication Language: English
- (30) Priority Data: 15/809,333 10 November 2017 (10.11.2017) US
- (71) Applicant: **INTERNATIONAL BUSINESS MACHINES CORPORATION** [US/US]; New Orchard Road, Armonk, New York 10504 (US).
- (71) Applicants (for MG only): **IBM UNITED KINGDOM LIMITED** [GB/GB]; PO Box 41, North Harbour, Portsmouth Hampshire PO6 3AU (GB). **IBM (CHINA) INVESTMENT COMPANY LIMITED** [CN/CN]; 25/ F, Pangu Plaza, No.27, Central North 4th Ring Road, Chaoyang District, Beijing 100101 (CN).
- (72) Inventors: **WANG, Zhennan**; IBM Corporation, 11501 Burnet Rd, Austin, Texas 78758-3400 (US). **SANCHEZ, John, Conrad**; IBM Corporation, 11501 Burnet Rd, Austin, Texas 78758-3400 (US). **RAMACHANDRAN, Puthukode**; IBM Corporation, 11501 Burnet Rd, Austin, Texas 78758-3400 (US).
- (74) Agent: **SHAW, Anita**; IBM United Kingdom Limited, Intellectual Property Law, Hursley Park, Winchester Hampshire SO21 2JN (GB).
- (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: ACCESSING GATEWAY MANAGEMENT CONSOLE

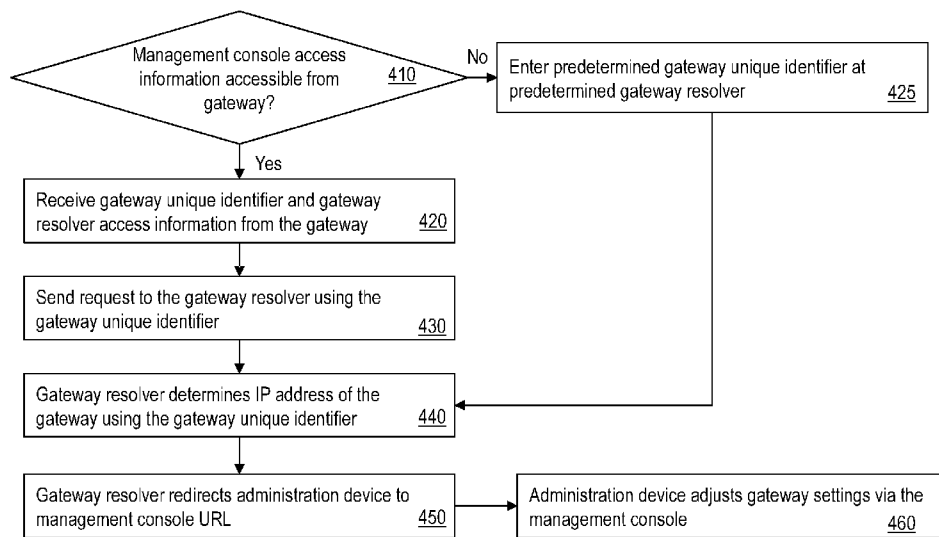
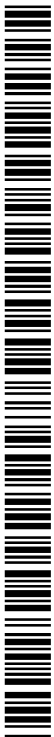


Fig. 5

(57) Abstract: Technical solutions are described for configuring a gateway in an IoT system. An example method includes providing, by an identification module, a user-interface (UI) access information to an administrative device. The method further includes receiving, by a setup module, from the administrative device, and via a management console generated based on the UI access information, setup data. The method further includes adjusting, by the setup module, one or more settings of the gateway according to the setup data.



SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN,
TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

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

Published:

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

ACCESSING GATEWAY MANAGEMENT CONSOLE

BACKGROUND

[0001] The present invention generally relates to an improvement to a computer system, particularly a gateway used in an Internet of Things (IoT) setting by facilitating accessing a management console of the gateway.

[0002] As IoT grows and numerous devices are connected to one or more IoT cloud servers, the functionality of an IoT gateway gains further importance. An IoT gateway communicates data between the cloud servers and multiple end-point devices in the IoT setting. For example, the gateway aggregates sensor data, translates between sensor protocols, processes sensor data before sending it onward and more to the server(s). With scores of protocols, connectivity models and energy profiles and the highly dispersed nature of IoT systems, gateways manage and control these complex environments. IoT gateways perform several critical functions such as device connectivity, protocol translation, data filtering and processing, security, updating, management and more. In one or more examples, IoT gateways also operate as platforms for application code that processes data and becomes an intelligent part of a device-enabled system. IoT gateways are considered to be at the intersection of edge systems - devices, controllers and sensors -- and the cloud. Thus, managing the IoT gateway itself is a critical aspect of managing the IoT setup. The management of the IoT gateway is performed using a management console.

SUMMARY

[0003] According to one or more embodiments, an example computer implemented method for configuring a gateway in an IoT system includes providing, by an identification module, a user-interface (UI) access information to an administrative device. The method further includes receiving, by a setup module, from the administrative device, and via a management console generated based on the UI access information, setup data. The method further includes adjusting, by the setup module, one or more settings of the gateway according to the setup data.

[0004] According to one or more embodiments, an internet of things (IoT) system includes a gateway device that connects one or more application devices to corresponding application server, the gateway device including a memory device and a processor coupled with the memory device. The processor provides access to a management console for setting up the gateway device, providing the access including providing a user-interface (UI) access information to an administrative device to access the management console. Providing the access further including receiving setup data from the administrative device, via the management console. Providing the access further including adjusting one or more settings of the gateway device according to the setup data.

[0005] According to one or more embodiments, an example gateway device that is inaccessible to plug input-output devices into, the gateway device including a processor provides a management console for configuring the gateway device. Providing access to the management console includes providing a user-interface (UI) access information to an administrative device. Further, providing the access includes receiving, from the administrative device, and via a management console generated based on the UI access information, setup data. Further, providing the access includes adjusting one or more settings of the gateway according to the setup data.

[0006] According to one or more embodiments, an example apparatus includes a memory, and a processor coupled with the memory, the processor operates as a gateway device for an internet of things (IoT) system. The processor performs transporting data between an application device and an application server via a router. Further, in response to powering on, the processor sends apparatus identification information to a gateway resolver using a predetermined URL for the gateway resolver, and executes a management console accessible via a URL that has a predetermined format.

[0007] According to one or more embodiments, an example computer program product including a computer readable storage medium having program instructions embodied therewith, the program instructions executable by a processing circuit to cause the processing circuit to operate as a gateway device for an internet of things (IoT) system, the processing circuit performs transporting data between an application device and an application server via a router. Further, the processing circuit, in response to the processing circuit being powered on, sending gateway identification information to a gateway resolver using a predetermined URL for the gateway resolver, the gateway identification information comprises a unique identifier assigned to the gateway device and an IP address assigned to the gateway device. Further, the processing circuit executing a management console accessible via a URL that has a predetermined format.

BRIEF DESCRIPTION OF THE DRAWINGS

[0008] The examples described throughout the present document will be better understood with reference to the following drawings and description. The components in the figures are not necessarily to scale. Moreover, in the figures, like-referenced numerals designate corresponding parts throughout the different views.

FIG. 1 depicts a cloud computing environment according to an embodiment of the present invention;

FIG. 2 depicts abstraction model layers according to an embodiment of the present invention;

FIG. 3 depicts an example system for accessing a gateway management console of a gateway in an IoT system according to one or more embodiments;

FIG. 4 illustrates an example system according to one or more embodiments;

FIG. 5 depicts a flowchart of an example method for accessing a management console of an IoT gateway according to one or more embodiments;

FIG. 6 depicts a flowchart of an example method for configuring a gateway in an IoT system, according to one or more embodiments; and

FIG. 7 depicts a flowchart of an example method for accessing a management console of a gateway in an IoT system according to one or more embodiments.

DETAILED DESCRIPTION

[0009] Various embodiments of the invention are described herein with reference to the related drawings. Alternative embodiments of the invention can be devised without departing from the scope of this invention. Various connections and positional relationships (e.g., over, below, adjacent, etc.) are set forth between elements in the following description and in the drawings. These connections and/or positional relationships, unless specified otherwise, can be direct or indirect, and the present invention is not intended to be limiting in this respect. Accordingly, a coupling of entities can refer to either a direct or an indirect coupling, and a positional relationship between entities can be a direct or indirect positional relationship. Moreover, the various tasks and process steps described herein can be incorporated into a more comprehensive procedure or process having additional steps or functionality not described in detail herein.

[0010] The following definitions and abbreviations are to be used for the interpretation of the claims and the specification. As used herein, the terms "comprises," "comprising," "includes," "including," "has," "having," "contains" or "containing," or any other variation thereof, are intended to cover a non-exclusive inclusion. For example, a composition, a mixture, process, method, article, or apparatus that comprises a list of elements is not necessarily limited to only those elements but can include other elements not expressly listed or inherent to such composition, mixture, process, method, article, or apparatus.

[0011] Additionally, the term "exemplary" is used herein to mean "serving as an example, instance or illustration." Any embodiment or design described herein as "exemplary" is not necessarily to be construed as preferred or advantageous over other embodiments or designs. The terms "at least one" and "one or more" may be understood to include any integer number greater than or equal to one, i.e. one, two, three, four, etc. The terms "a plurality" may be understood to include any integer number greater than or equal to two, i.e. two, three, four, five, etc. The term "connection" may include both an indirect "connection" and a direct "connection."

[0012] The terms "about," "substantially," "approximately," and variations thereof, are intended to include the degree of error associated with measurement of the particular quantity based upon the equipment available at the time of filing the application. For example, "about" can include a range of $\pm 8\%$ or 5% , or 2% of a given value.

[0013] For the sake of brevity, conventional techniques related to making and using aspects of the invention may or may not be described in detail herein. In particular, various aspects of computing systems and specific computer programs to implement the various technical features described herein are well known. Accordingly, in the interest of brevity, many conventional implementation details are only mentioned briefly herein or are omitted entirely without providing the well-known system and/or process details.

[0014] The technical solutions described herein facilitate accessing an IoT gateway management, such as a user-interface (UI) console, particularly in cases where the IoT gateway is not physically accessible. For example, the IoT gateway may be a relatively small device such as a single board computer, like a RASPBERRY PI™, or any other such device. Typically, a software, such as an operating system (OS) of the gateway provides the UI, such as a web-UI to perform management tasks, such as IoT device initial configuration(s), and setting up connections between the gateway and IoT cloud server(s). To access the gateway management UI, an output device, such as a computer monitor, and an input device, such as a keyboard and/or mouse are directly plugged into the gateway, for logging into the gateway OS, and operating the management UI of the gateway.

[0015] However, because of the small size, or for any other reason, the gateway may be placed in a location that is not physically accessible for connecting other devices physically, using a physical port of the gateway or plugging a device connector into the gateway, or in any other such physical manner. Alternatively, or in addition, the gateway may not have a port for physical connections with other devices that are external to the gateway. Thus, a technical challenge exists of accessing the management console of the gateway that is physically inaccessible for plugging I/O devices. The technical solutions described herein address the technical challenge by using a Uniform Resource Locator (URL) resolver server. Thus, the technical solutions described herein facilitate accessing a management console of an IoT gateway that is physically inaccessible, thus improving the IoT systems by facilitating the use of smaller and cheaper devices as IoT gateways. The smaller devices further improve the efficiency of the IoT systems, because such devices without providing the physical pluggable ports may use lesser power. Further, in one or more examples, the technical solutions further reduce costs of the IoT systems by not requiring to use dedicated I/O devices for the IoT gateway.

[0016] The technical solutions described herein may be implemented using cloud computing in one or more examples. It is understood in advance that although this disclosure includes a detailed description on cloud computing, implementation of the teachings recited herein are not limited to a cloud computing environment. Rather, embodiments of the present invention are capable of being implemented in conjunction with any other type of computing environment now known or later developed.

[0017] Cloud computing is a model of service delivery for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g. networks, network bandwidth, servers, processing,

memory, storage, applications, virtual machines, and services) that can be rapidly provisioned and released with minimal management effort or interaction with a provider of the service. This cloud model may include at least five characteristics, at least three service models, and at least four deployment models.

[0018] Characteristics are as follows:

On-demand self-service: a cloud consumer can unilaterally provision computing capabilities, such as server time and network storage, as needed automatically without requiring human interaction with the service's provider.

Broad network access: capabilities are available over a network and accessed through standard mechanisms that promote use by heterogeneous thin or thick client platforms (e.g., mobile phones, laptops, and PDAs).

Resource pooling: the provider's computing resources are pooled to serve multiple consumers using a multi-tenant model, with different physical and virtual resources dynamically assigned and reassigned according to demand. There is a sense of location independence in that the consumer generally has no control or knowledge over the exact location of the provided resources but may be able to specify location at a higher level of abstraction (e.g., country, state, or datacenter).

Rapid elasticity: capabilities can be rapidly and elastically provisioned, in some cases automatically, to quickly scale out and rapidly released to quickly scale in. To the consumer, the capabilities available for provisioning often appear to be unlimited and can be purchased in any quantity at any time.

Measured service: cloud systems automatically control and optimize resource use by leveraging a metering capability at some level of abstraction appropriate to the type of service (e.g., storage, processing, bandwidth, and active user accounts). Resource usage can be monitored, controlled, and reported providing transparency for both the provider and consumer of the utilized service.

[0019] Service Models are as follows:

Software as a Service (SaaS): the capability provided to the consumer is to use the provider's applications running on a cloud infrastructure. The applications are accessible from various client devices through a thin client interface such as a web browser (e.g., web-based e-mail). The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, storage, or even individual application capabilities, with the possible exception of limited user-specific application configuration settings.

Platform as a Service (PaaS): the capability provided to the consumer is to deploy onto the cloud infrastructure consumer-created or acquired applications created using programming languages and tools supported by the provider. The consumer does not manage or control the underlying cloud infrastructure including networks, servers, operating systems, or storage, but has control over the deployed applications and possibly application hosting environment configurations

Infrastructure as a Service (IaaS): the capability provided to the consumer is to provision processing, storage, networks, and other fundamental computing resources where the consumer is able to deploy and run arbitrary software, which can include operating systems and applications. The consumer does not manage or control the underlying cloud infrastructure but has control over operating systems, storage, deployed applications, and possibly limited control of select networking components (e.g., host firewalls).

[0020] Deployment Models are as follows:

Private cloud: the cloud infrastructure is operated solely for an organization. It may be managed by the organization or a third party and may exist on-premises or off-premises.

Community cloud: the cloud infrastructure is shared by several organizations and supports a specific community that has shared concerns (e.g., mission, security requirements, policy, and compliance considerations). It may be managed by the organizations or a third party and may exist on-premises or off-premises.

Public cloud: the cloud infrastructure is made available to the general public or a large industry group and is owned by an organization selling cloud services.

Hybrid cloud: the cloud infrastructure is a composition of two or more clouds (private, community, or public) that remain unique entities but are bound together by standardized or proprietary technology that enables data and application portability (e.g., cloud bursting for load-balancing between clouds).

[0021] A cloud computing environment is service oriented with a focus on statelessness, low coupling, modularity, and semantic interoperability. At the heart of cloud computing is an infrastructure comprising a network of interconnected nodes.

[0022] Referring now to FIG. 1, illustrative cloud computing environment 50 is depicted. As shown, cloud computing environment 50 comprises one or more cloud computing nodes 10 with which local computing devices used by cloud consumers, such as, for example, personal digital assistant (PDA) or cellular telephone 54A, desktop computer 54B, laptop computer 54C, and/or automobile computer system 54N may communicate. Nodes 10 may communicate with one another. They may be grouped (not shown) physically or virtually, in one or more networks, such as Private, Community, Public, or Hybrid clouds as described hereinabove, or a combination thereof. This allows cloud computing environment 50 to offer infrastructure, platforms and/or software as services for which a cloud consumer does not need to maintain resources on a local computing device. It is understood that the types of computing devices 54A-N shown in FIG. 1 are intended to be illustrative only and that computing nodes 10 and cloud computing environment 50 can communicate with any type of computerized device over any type of network and/or network addressable connection (e.g., using a web browser).

[0023] Referring now to FIG. 2, a set of functional abstraction layers provided by cloud computing environment 50 (FIG. 1) is shown. It should be understood in advance that the components, layers, and functions

shown in FIG. 2 are intended to be illustrative only and embodiments of the invention are not limited thereto. As depicted, the following layers and corresponding functions are provided:

[0024] Hardware and software layer 60 includes hardware and software components. Examples of hardware components include: mainframes 61; RISC (Reduced Instruction Set Computer) architecture based servers 62; servers 63; blade servers 64; storage devices 65; and networks and networking components 66. In some embodiments, software components include network application server software 67 and database software 68.

[0025] Virtualization layer 70 provides an abstraction layer from which the following examples of virtual entities may be provided: virtual servers 71; virtual storage 72; virtual networks 73, including virtual private networks; virtual applications and operating systems 74; and virtual clients 75.

[0026] In one example, management layer 80 may provide the functions described below. Resource provisioning 81 provides dynamic procurement of computing resources and other resources that are utilized to perform tasks within the cloud computing environment. Metering and Pricing 82 provide cost tracking as resources are utilized within the cloud computing environment, and billing or invoicing for consumption of these resources. In one example, these resources may comprise application software licenses. Security provides identity verification for cloud consumers and tasks, as well as protection for data and other resources. User portal 83 provides access to the cloud computing environment for consumers and system administrators. Service level management 84 provides cloud computing resource allocation and management such that required service levels are met. Service Level Agreement (SLA) planning and fulfillment 85 provide pre-arrangement for, and procurement of, cloud computing resources for which a future requirement is anticipated in accordance with an SLA.

[0027] Workloads layer 90 provides examples of functionality for which the cloud computing environment may be utilized. Examples of workloads and functions which may be provided from this layer include: mapping and navigation 91; software development and lifecycle management 92; virtual classroom education delivery 93; data analytics processing 94; transaction processing 95; and gateway URL resolving 96.

[0028] FIG. 3 depicts an example system 100 for accessing an IoT gateway management console according to one or more embodiments of the present invention. The system includes, among other components, one or more application devices 110, a router 120, a gateway 130, an application server 140, an administration device 150, a gateway resolver 160, and a gateway identification data repository 170.

[0029] The application devices 110 and the application server 140, together, implement an IoT based system/application. The application devices 110 send/receive data to/from the application server 140 for implementing the IoT based system. The IoT system may be any system/application such as monitoring/controlling

one or more parameters by the application devices 110 using one or more analysis performed by the application server 140. The application devices 110 may be also referred to as IoT devices, and may include devices such as sensors, sensor controllers, and the like, for example, thermostats, security systems, home automation devices, production line automation sensors, warehouse automation systems, autonomous vehicle sensor systems, or any other devices that are used in an IoT based system. It should be noted that the examples of the application devices 110 listed above are just a few of various possible application devices 110 that may be used in other embodiments.

[0030] The application server 140 is a computer server that receives the data being monitored by the application devices 110, analyzes the data, and sends one or more adjustments to be made by the application devices 110. For example, in case the IoT system is a temperature adjustment system, the application server 140, based on the analysis of the data collected by the application devices 110 (for example, a thermostat), causes the application devices 110 to change the settings of an air-conditioning system to cause the temperature being monitored to increase (or decrease). In other examples, the application devices 110 may monitor different parameters and the application server causes the application devices 110 to change such different parameters.

[0031] The application devices 110 and the application server 140 communicate with each other via one or more network connectivity devices, such as the router 120 and the gateway 130, among others. The router 120 facilitates communication between the gateway 130 and the application server 140, the communication being performed using the Internet, in one or more examples. In one or more examples, the gateway 130 and the application devices 110 may be part of a private network, such as a local area network (LAN), wide area network (WAN), or the like, with the gateway 130 in communication with the router 120 to facilitate communication between the application devices 110, which are within the private network, with the application server 140, which is outside the private network. Alternatively, or in addition, the communication between the application devices 110 and the gateway 130 may be through any wireless technologies, such as LAN, WAN, BLUETOOTH™, ZIGBEE™, Z-WAVE™, etc., or through wires directly. In one or more examples, the gateway 130 and the router 120 may be part of the same device, although they are illustrated as distinct devices in FIG. 3.

[0032] In one or more examples, the gateway 130 facilitates the communication between the application devices 110 and the application server 140 by converting the data from a format of the communication source (application devices 110/application server) 140 to a format of the communication destination (application devices 110/application server).

[0033] It should be noted that, in one or more examples, the gateway 130 may operate as a gateway for two or more IoT systems. For example, the application devices 110 may include a subset of application devices 110 that communicate with a first application server 140, for example, to monitor/control temperature. Concurrently, another subset of the application devices 110 communicates with a second application server 140, via the gateway

130, for monitoring/controlling security, for example by detecting motion. It should be noted that examples other than those described herein are possible.

[0034] Further yet, the gateway 130 facilitates setting up the one or more IoT systems being implemented using the application devices 110, for example by configuring the application devices 110. For example, configuring the application devices 110 may include setting one or more operating parameters, identifiers, login information, and the like, for the application devices 110. Alternatively, or in addition, the gateway 130 downloads and installs/updates firmware on the application devices 110, in response to command(s) received from the application server 140.

[0035] In one or more examples, the gateway 130 provides a management console for an administrator to setup the IoT system, including the configuring the application devices 110. The administrator performs management tasks, such as device initial configuration, and setting up connections between the gateway 130 and the application server 140 via the management console. The administrator also configures the gateway itself via the management console.

[0036] As described earlier, typically to access the management console, a computer monitor, a keyboard, a mouse, and other such I/O devices are directly plugged into the gateway 130. The I/O devices are then used to login to the gateway OS. In one or more examples, the management console of the gateway 130 may be web—based, and accordingly, an IP address for the gateway 130 is determined and then the management console for the gateway 130 is launched using the IP address. The management console may be accessed using a URL that is based on the IP address, for example, that includes the IP address. The management console may be launched either from the gateway 130 itself or from a remote third device, in one or more examples. However, this is not a desirable process to access the gateway's management console because under some circumstances, as described herein, the gateway 130 is physically inaccessible. Alternatively, or in addition, even if the gateway 130 is accessible, use of the I/O devices, such as a screen or keyboard add cost for the gateway's deployment. Alternatively, or in addition, the gateway 130 may not include pluggable ports, for example, to prevent outside elements to enter the gateway 130, for example in a biomedical laboratory to prevent bio-hazardous material, or any other material from entering the gateway 130, or in an industrial setting to prevent dust, moisture, fine particles, or any other material from entering the gateway 130. The technical solutions described herein address such technical challenges.

[0037] For example, as depicted in FIG. 3, the administrative device 150 facilitates accessing the management console of the gateway 130, without physically plugging into the gateway 130. In one or more examples, the network 115 may include the gateway 130 and the application devices 110. Accordingly, the administration device 150 may not be able to directly communicate with the application devices 110 to set one or

more operating parameters of the application devices 110. The administrative device 150 wirelessly determines a unique identifier associated with the gateway 130 and determines the IP address of the gateway 130 using a gateway resolver 160.

[0038] The gateway resolver 160 is a computer, such as a server. The gateway resolver 160 receives a request from the administrative device 150 to provide an IP address for the gateway 130 based on the unique identifier of the gateway 130. The gateway resolver 160 determines the IP address corresponding to the unique identifier using the gateway identification data repository 170.

[0039] The gateway identification data repository 170 includes a data structure, such as a table, that maintains a mapping between the IP address and the unique identifier of the gateway. It should be noted that the gateway identification data repository 170 includes such mapping for multiple gateways, which may be accessed to determine the IP address of one of the gateways via the request to the gateway resolver 160. It should be noted that although the gateway identification data repository 170 and the gateway resolver 160 are depicted as separate devices in FIG. 3, in one or more examples, the two may be part of a single system/device. The administrative device 150 and the gateway resolver 160, in one or more examples, communicate via a communication network, such as the Internet, where the administrative device 150 knows the access parameters, such as an IP address, URL, or other such access parameters of the gateway resolver 160.

[0040] In one or more examples, the gateway resolver 160 provides the corresponding IP address of the unique identifier of the gateway 130 only if the administrative device 150 is a trusted device. For example, the administrative device 150 may be a trusted device if credentials associated with the administrative device 150 are known or authenticated by the gateway resolver 160. For example, the credentials may include a unique identifier of the administrative device 150, such as a MAC address, a serial number, a phone number, or other such parameters. Alternatively, or in addition, the credentials may include a username, a password, a fingerprint, a face, or other biometric-type identifiers, associated with the administrative device 150.

[0041] Upon validation of such credentials, or authentication using the credentials, the gateway resolver 160 may provide the IP address of the gateway 130 to the administrative device 150. The administrative device 150 further uses the IP address to access the management console of the gateway 130, for example, by generating the URL or other access location for the management console based on the IP address. Alternatively, in one or more examples, the gateway resolver 160 determines the URL or access location for the management console of the gateway 130 and sends the access location to the administrative device 150. Alternatively, in one or more examples, the gateway resolver 160, upon determination of the IP address, requests the gateway 130 to generate and/or provide the access location of the management console and transfers the access location received from the gateway 130 to the administrative device 150.

[0042] The system 100 thus facilitates accessing the management console of the gateway 130 in a wireless manner, without using plugged into the gateway 130, the management console being used to configure the IoT setup.

[0043] FIG. 4 illustrates an example system 200 according to one or more embodiments. The system 200 may be a communication apparatus, such as a computer. For example, the system 200 may be a desktop computer, a tablet computer, a laptop computer, a phone, such as a smartphone, a server computer, or any other device that communicates via a network 265. The system 200 includes hardware, such as electronic circuitry. In one or more examples, the system 200 is any one or more of the devices depicted in FIG. 3, such as the gateway 130, the application server 140, the administrative device 150, the gateway resolver 160, or any one or more of the application devices 110.

[0044] The system 200 includes, among other components, a processor 205, memory 210 coupled to a memory controller 215, and one or more input devices 245 and/or output devices 240, such as peripheral or control devices, that are communicatively coupled via a local I/O controller 235. These devices 240 and 245 may include, for example, battery sensors, position sensors, indicator/identification lights and the like. Input devices such as a conventional keyboard 250 and mouse 255 may be coupled to the I/O controller 235. The I/O controller 235 may be, for example, one or more buses or other wired or wireless connections, as are known in the art. The I/O controller 235 may have additional elements, which are omitted for simplicity, such as controllers, buffers (caches), drivers, repeaters, and receivers, to enable communications.

[0045] The I/O devices 240, 245 may further include devices that communicate both inputs and outputs, for instance disk and tape storage, a network interface card (NIC) or modulator/demodulator (for accessing other files, devices, systems, or a network), a radio frequency (RF) or other transceiver, a telephonic interface, a bridge, a router, and the like.

[0046] The processor 205 is a hardware device for executing hardware instructions or software, particularly those stored in memory 210. The processor 205 may be a custom made or commercially available processor, a central processing unit (CPU), an auxiliary processor among several processors associated with the system 200, a semiconductor based microprocessor (in the form of a microchip or chip set), a macroprocessor, or other device for executing instructions. The processor 205 includes a cache 270, which may include, but is not limited to, an instruction cache to speed up executable instruction fetch, a data cache to speed up data fetch and store, and a translation lookaside buffer (TLB) used to speed up virtual-to-physical address translation for both executable instructions and data. The cache 270 may be organized as a hierarchy of more cache levels (L1, L2, and so on.).

[0047] The memory 210 may include one or combinations of volatile memory elements (for example, random access memory, RAM, such as DRAM, SRAM, SDRAM) and nonvolatile memory elements (for example, ROM, erasable programmable read only memory (EPROM), electronically erasable programmable read only memory (EEPROM), programmable read only memory (PROM), tape, compact disc read only memory (CD-ROM), disk, diskette, cartridge, cassette or the like). Moreover, the memory 210 may incorporate electronic, magnetic, optical, or other types of storage media. Note that the memory 210 may have a distributed architecture, where various components are situated remote from one another but may be accessed by the processor 205.

[0048] The instructions in memory 210 may include one or more separate programs, each of which comprises an ordered listing of executable instructions for implementing logical functions. In the example of Figure 2, the instructions in the memory 210 include a suitable operating system (OS) 211. The operating system 211 essentially may control the execution of other computer programs and provides scheduling, input-output control, file and data management, memory management, and communication control and related services.

[0049] Additional data, including, for example, instructions for the processor 205 or other retrievable information, may be stored in storage 220, which may be a storage device such as a hard disk drive or solid state drive. The stored instructions in memory 210 or in storage 220 may include those enabling the processor to execute one or more aspects of the systems and methods described herein.

[0050] The system 200 may further include a display controller 225 coupled to a user interface or display 230. In some embodiments, the display 230 may be an LCD screen. In other embodiments, the display 230 may include a plurality of LED status lights. In some embodiments, the system 200 may further include a network interface 260 for coupling to a network 265. The network 265 may be an IP-based network for communication between the system 200 and an external server, client and the like via a broadband connection. In an embodiment, the network 265 may be a satellite network. The network 265 transmits and receives data between the system 200 and external systems. In some embodiments, the network 265 may be a managed IP network administered by a service provider. The network 265 may be implemented in a wireless fashion, for example, using wireless protocols and technologies, such as WiFi, WiMax, satellite, or any other. The network 265 may also be a packet-switched network such as a local area network, wide area network, metropolitan area network, the Internet, or other similar type of network environment. The network 265 may be a fixed wireless network, a wireless local area network (LAN), a wireless wide area network (WAN) a personal area network (PAN), a virtual private network (VPN), intranet or other suitable network system and may include equipment for receiving and transmitting signals.

[0051] FIG. 5 depicts a flowchart of an example method for accessing a management console of an IoT gateway according to one or more embodiments. The method may be implemented using the system 100. The method includes the administration device 150 determining if management console access information for the

gateway device 130 is accessible, at 410. For example, the gateway device 130 may include a radio frequency identification (RFID) device (not shown) that outputs the access information for the gateway device 130.

Alternatively, or in addition, the access information may be available from the gateway device 130 may be accessible via near field communication protocols, such as BLUETOOTH LOW ENERGY (BLE™), ZIGBEE™, Z WAVE™, or any other such near field communication. Alternatively, or in addition, the access information may be available from the gateway device 130 via a wireless communication protocol, such as BLUETOOTH™, WIFI™, or the like. Alternatively, or in addition, the access information may be available from the gateway device 130 in the form of a code, text, image, or any other form that can be captured by the administration device 150. For example, the access information may be part of a barcode, quick response (QR) code or the like that is posted, for example as a sticker, on the gateway device 130, or in a predetermined location associated with the gateway device 130. It should be noted that above are few examples of providing the access information by the gateway device 130 in a wireless and unpluggable manner, and in other embodiments, other techniques may be used.

[0052] If the access information is accessible, the administration device 150 receives the access information from the gateway device 130, at 420. In one or more examples, the administration device 150 may capture the access information from the gateway device 130 using a camera, scanner, or other such input device. Alternatively, or in addition, the administration device 150 captures the access information from the gateway device 130 in one or more wireless manner, such as those described above.

[0053] The access information includes at least two parameters, a unique identifier (gateway ID) of the gateway device 130, and an access location of the gateway resolver 160. For example, if the access information is a QR code, the QR code is composed in a website URL format, which points to the gateway resolver 160, and the gateway ID is added as part of the URL in a way that the gateway resolver 160 can retrieve the gateway ID from the access information. For example, the gateway ID is part of the access information as a predetermined parameter value that the gateway resolver 160 is able to retrieve upon receiving a request from the administration device 150. That is because the method further includes, sending a request to the gateway resolver 160 using the access information, at 430. If the access information is not accessible from the gateway device 130, the administration device 150 sends the request to the gateway resolver 160 using a predetermined location of the gateway resolver 160, such as a URL access location, or an application programming interface (API) call or the like, at 425. The administration device 150 may provide the gateway ID as a parameter of the request even, in this case, the gateway ID being manually entered by an administrator or another user of the administration device 150.

[0054] The gateway ID may be a unique identifier of the gateway device 130, for example, a MAC address, a serial number, a unique string of digital characters, or any other unique identifier for the gateway device 130.

[0055] The gateway resolver 160, upon receiving the request from the administration device 150, determines an IP address of the gateway device 130 based on the gateway ID, at 440. The gateway resolver 160, in one or more examples, uses the mapping in the gateway identification data repository 170 to determine the IP address of the gateway device 130 using the gateway ID retrieved from the request.

[0056] The method further includes redirecting the administration device 150 to a URL of the management console of the gateway device 130 by the gateway resolver 160, at 450. In one or more examples, the gateway resolver composes the management console's URL and then redirects the administration device 150 to the URL. The URL may be composed using a predetermined URL format that the gateway device 130 uses to launch the management console, for example via a web-based UI. Alternatively, or in addition, the gateway resolver 160 requests the gateway device 130 for the URL of the management console using the IP address of the gateway device 130, and redirects the administration device 150 to the URL from the gateway device 130.

[0057] Upon being redirected, the administration device 150 uses the management console to adjust settings of the gateway device 130, at 460. The settings adjusted include those for operations of the gateway device 130 and/or the application device 110, and/or the operation of the IoT system that includes the application devices 110 and the application server 140.

[0058] In an example scenario, to access the gateway management console, an administration device 150, such as a mobile device scans a website (URL) QR code attached to the gateway device 130. The QR code contains URL of the gateway resolver 160 and the gateway's unique ID. The administration device 150 upon scanning the QR code sends a request, such as an HTTP request, to the gateway resolver 160, which retrieves the gateway unique ID from the request URL, and determines the gateway's IP address from the data repository 170 using this unique ID. The gateway resolver 160 (or the gateway device 130) composes the management console URL and redirects the administration device 150 to management console via the console URL.

[0059] FIG. 6 depicts a flowchart of an example method for configuring a gateway in an IoT system, according to one or more embodiments. The method includes starting up the gateway device 130, at 510. The gateway device 130 is connected to the network 265, either by network cable or by wireless and powered on, for example. Upon startup, the gateway device 130 determines the unique identifier assigned to the gateway device 130, at 512. For example, the unique identifier may be determined by reading the gateway ID from a predetermined location, such as a memory address, a parameter value (e.g.: serial number, MAC address), or the like. The unique identifier can be any string or the like, that is unique within all deployed gateways.

[0060] Further, the gateway device 130 executes the management console and makes it accessible using a URL that uses a predetermined URL format, at 514. For example, the URL format may be: `https://<Gateway IP`

address>/gateway-management. It should be noted that in other embodiments, the URL format may be different than the earlier example. The gateway device 130 begins typical operation, at 520, which includes transporting data between the application devices 110 and the corresponding application server 140, at 528.

[0061] In addition, the method includes, executing an IP reporting code on the gateway device 130, which sends an IP address of the gateway device 130 to the gateway resolver 160, at 526. In one or more examples, the access location of the gateway resolver 160 is predetermined and included in the gateway as a preset, such as at a predetermined memory address or as a parameter value, for example in the form of an IP address of the gateway resolver 160. The IP reporting code accesses the access location of the gateway resolver 160 and sends the IP address of the gateway device 130 along with the unique identifier of the gateway 130 to the gateway resolver 160, at 526. For example, the gateway device 130 sends a message in a format, such as {"UniqueID":"6c4008ae8786","IP":"192.168.0.28"}. It should be noted that the format used may be different in other embodiments. The gateway resolver 160 updates the mapping in the data repository 170 between the gateway identifier and the gateway IP address.

[0062] In one or more examples, the gateway device 130 reports the IP address to the gateway resolver 160 every time an IP address is assigned to the gateway device 130, at 522. For example, when the gateway device 130 is powered on and connected to the network 265, upon being assigned an IP address, the gateway device 130 reports the IP address to the gateway resolver 160. Further, if the network 265 is reset and the IP address of the gateway device 130 is changed, the updated IP address is reported to the gateway resolver 160 along with the gateway ID as described herein. Alternatively, or in addition, the gateway device 130 sends a periodic message to the gateway resolver with the gateway ID and gateway IP address, according to a predetermined timer, at 524. The gateway identification data repository 170 thus has the latest IP address of the gateway device 130 mapped to the gateway ID of the gateway device 130.

[0063] Further, the method includes configuring the gateway device 130 using the administration device 150, at 530. The configuring includes providing the management console access information to the administration device 150, at 532. As described herein, the access information is provided in a wireless manner. The access information includes at least the gateway ID and the gateway resolver access location, such as a URL, for example in the form of a QR code or the like. The administration device 150 resolves the management console URL via the gateway resolver 160 as described herein using the gateway ID.

[0064] The method further includes receiving a request from the administration device 150 to access the management console at the URL associated with the management console, at 534. As described herein, the URL uses a predetermined format and may be composed by the gateway resolver 160, or is provided by the gateway

device 130 to the gateway resolver 160, which redirects the administration device 150 to the management console URL.

[0065] The method further includes adjusting the settings of the gateway device 130 and/or the IoT settings, for example, of the application devices 110, as per instructions received via the management console, at 536.

[0066] The method facilitates the gateway device 130 to keep the gateway resolver apprised of any changes to the gateway's IP address so that a user/administrator can access the management console of the gateway device 130 in a wireless manner and configure the gateway device 130 to operate in the IoT system. It should be noted that the management console also facilitates adjusting settings of the IoT system, such as operating parameters of the application devices 110. For example, the management console may facilitate changing communication protocols, application server(s), measurements captured, the frequency of measurements, or any other such operating parameters of the application devices 110.

[0067] FIG. 7 depicts a flowchart of an example method for accessing a management console of a gateway in an IoT system according to one or more embodiments. The method includes receiving, by the gateway resolver 160, identification information from the gateway device 130, at 610. The identification information includes the gateway ID and the IP address of the gateway device 130, as described herein. The identification information may be received when the gateway device 130 starts up, experiences a network reset, and/or is assigned an updated IP address for any other reason.

[0068] The method further includes updating the data repository 170 by the gateway resolver 160, at 620. The update includes retrieving the gateway ID and the gateway IP address from the identification information received, at 622. In one or more examples, the two parameters are at the predetermined location or are predetermined portions of the identification information received. The gateway resolver 160 checks if the gateway ID already exists in the data repository 170 that maintains a mapping between gateway IDs and gateway IP addresses, at 624. If the gateway ID does not exist in the data repository 170, a new entry is created in the mapping, the entry maps the gateway ID and the gateway IP address received in the identification information, at 626. If the gateway ID already exists in an entry in the data repository 170, the entry is updated with the gateway IP address received in the identification information, at 628. The data repository 170 thus contains the latest gateway IP address.

[0069] Further, the method includes receiving, by the gateway resolver 160, from the administration device 150, a request to resolve a management access information, at 630. As described herein, the administration device 150 may send the request in response to receiving the access information from the gateway device 130. Alternatively, or in addition, the administration device 150 sends the request with manual entry for the gateway ID.

[0070] The gateway resolver 160, in one or more examples, authenticates that the request is received from a trusted administration device 150, at 640. For example, the authentication may include ensuring that the administration device 150 is one from a predetermined list of administration devices. For example, the access information received from the administration device 150 also includes a validation information for the administration device 150. For example, the validation information includes an identification of the administration devices, such as a MAC address, a serial number or the like. Alternatively, or in addition, the validation information includes a trust certification, such as a digital certificate using one or more validation techniques such as a public key certificate, or any other security technique. Alternatively, or in addition, the validation information includes a username-password combination, biometric information, or any other such data associated with the user of the administration device 150 sending the request to the gateway resolver 160.

[0071] If the administration device 150 is not a trusted device, the gateway resolver does not proceed with the method, and the administration device 150 is prevented from accessing the management console of the gateway device 130. If the administration device 150 is a trusted device, the gateway resolver 160 determines an access location, such as a URL, of the gateway management console, at 650.

[0072] Determining the gateway management console URL includes parsing the access information from the request further to retrieve the gateway ID, at 652. The gateway resolver 160 checks if the data repository 170 includes a mapping entry for the gateway ID, at 654. If an entry does not exist, the gateway resolver 160 indicates an error condition to the administration device 150, at 654. If the mapping entry for the gateway ID is retrieved from the data repository 170, the gateway resolver determines the gateway IP address from the mapping, at 656.

[0073] The method further includes obtaining the URL of the gateway device 130 using the gateway IP address from the mapping, at 658. In one or more examples, obtaining the URL includes the gateway resolver 160 composing the URL using a predetermined format for gateway device 130, as described herein. In one or more examples, different gateway devices may use different corresponding predetermined formats. Accordingly, the data repository 170, in one or more examples, may store the predetermined format for the URL creation for respective gateway devices 130 in the mapping entries. Alternatively, the gateway resolver 160 sends a request to the gateway device 130 to provide the predetermined format for the URL composition. Based on the predetermined format for the gateway device 130, the gateway resolver generates the URL for the gateway management console.

[0074] Alternatively, in one or more examples, the gateway resolver 160 sends a request to the gateway device 130 using the gateway IP address retrieved from the mapping entry, the request being for the gateway management console URL. In response, the gateway device 130 provides the URL for the gateway management.

[0075] The method further includes redirecting the administration device 150 to the gateway management console URL in response to the request for resolving the access information that was sent by the administration device 150, at 660. The administration device 150, upon being redirected to the URL, is able to provide instructions to adjust one or more operating parameters of the gateway device 130, the application devices 110, or other parameters associated with the operation of the IoT system(s) that the application devices 110 are part of.

[0076] The technical solutions described herein facilitate accessing a management console of an IoT gateway device that is physically inaccessible via ports or other pluggable hardware. Further, the technical solutions described herein facilitate reducing costs of setting up an IoT based system. The technical solutions described herein improve the IoT based system by facilitating using the gateway being accessible, or without pluggable ports being accessible, which enables using the IoT system in settings with such requirements, such as in the context of biomedical, industrial applications and the like. Further, the technical solutions improve the IoT systems by facilitating a reduction in the size of the gateway, because pluggable ports in the gateway add to the size of the gateway.

[0077] The technical solutions described herein further do not use tree structures, that may be used by name servers to resolve a computer's IP address, rather the technical solutions described herein use a gateway resolver, which is a centralized IP service server to resolve a gateway's IP address. A gateway's IP address can be either a fixed IP or a dynamic IP. According to the technical solutions described herein the gateways report their IP addresses periodically, or in response to an event such as a network reset, to the URL resolver service that the gateway resolver operates, so that the latest gateway management console URL can be retrieved. Thus, the technical solutions use an active binding between the gateway identifier and the gateway's management console URL, the binding being unique (1-to-1) and proactive (resource owner, gateway, periodically report its IP to the binding service server.)

[0078] The present technical solutions may be a system, a method, and/or a computer program product at any possible technical detail level of integration. The computer program product may include a computer readable storage medium (or media) having computer readable program instructions thereon for causing a processor to carry out aspects of the present technical solutions.

[0079] The computer readable storage medium can be a tangible device that can retain and store instructions for use by an instruction execution device. The computer readable storage medium may be, for example, but is not limited to, an electronic storage device, a magnetic storage device, an optical storage device, an electromagnetic storage device, a semiconductor storage device, or any suitable combination of the foregoing. A non-exhaustive list of more specific examples of the computer readable storage medium includes the following: a portable computer diskette, a hard disk, a random access memory (RAM), a read-only memory (ROM), an erasable

programmable read-only memory (EPROM or Flash memory), a static random access memory (SRAM), a portable compact disc read-only memory (CD-ROM), a digital versatile disk (DVD), a memory stick, a floppy disk, a mechanically encoded device such as punch-cards or raised structures in a groove having instructions recorded thereon, and any suitable combination of the foregoing. A computer readable storage medium, as used herein, is not to be construed as being transitory signals per se, such as radio waves or other freely propagating electromagnetic waves, electromagnetic waves propagating through a waveguide or other transmission media (e.g., light pulses passing through a fiber-optic cable), or electrical signals transmitted through a wire.

[0080] Computer readable program instructions described herein can be downloaded to respective computing/processing devices from a computer readable storage medium or to an external computer or external storage device via a network, for example, the Internet, a local area network, a wide area network and/or a wireless network. The network may comprise copper transmission cables, optical transmission fibers, wireless transmission, routers, firewalls, switches, gateway computers and/or edge servers. A network adapter card or network interface in each computing/processing device receives computer readable program instructions from the network and forwards the computer readable program instructions for storage in a computer readable storage medium within the respective computing/processing device.

[0081] Computer readable program instructions for carrying out operations of the present technical solutions may be assembler instructions, instruction-set-architecture (ISA) instructions, machine instructions, machine dependent instructions, microcode, firmware instructions, state-setting data, configuration data for integrated circuitry, or either source code or object code written in any combination of one or more programming languages, including an object oriented programming language such as Smalltalk, C++, or the like, and procedural programming languages, such as the "C" programming language or similar programming languages. The computer readable program instructions may execute entirely on the user's computer, partly on the user's computer, as a stand-alone software package, partly on the user's computer and partly on a remote computer or entirely on the remote computer or server. In the latter scenario, the remote computer may be connected to the user's computer through any type of network, including a local area network (LAN) or a wide area network (WAN), or the connection may be made to an external computer (for example, through the Internet using an Internet Service Provider). In some embodiments, electronic circuitry including, for example, programmable logic circuitry, field-programmable gate arrays (FPGA), or programmable logic arrays (PLA) may execute the computer readable program instructions by utilizing state information of the computer readable program instructions to personalize the electronic circuitry, in order to perform aspects of the present technical solutions.

[0082] Aspects of the present technical solutions are described herein with reference to flowchart illustrations and/or block diagrams of methods, apparatus (systems), and computer program products according to embodiments of the technical solutions. It will be understood that each block of the flowchart illustrations and/or

block diagrams, and combinations of blocks in the flowchart illustrations and/or block diagrams, can be implemented by computer readable program instructions.

[0083] These computer readable program instructions may be provided to a processor of a general purpose computer, special purpose computer, or other programmable data processing apparatus to produce a machine, such that the instructions, which execute via the processor of the computer or other programmable data processing apparatus, create means for implementing the functions/acts specified in the flowchart and/or block diagram block or blocks. These computer readable program instructions may also be stored in a computer readable storage medium that can direct a computer, a programmable data processing apparatus, and/or other devices to function in a particular manner, such that the computer readable storage medium having instructions stored therein comprises an article of manufacture including instructions which implement aspects of the function/act specified in the flowchart and/or block diagram block or blocks.

[0084] The computer readable program instructions may also be loaded onto a computer, other programmable data processing apparatus, or other device to cause a series of operational steps to be performed on the computer, other programmable apparatus or other device to produce a computer implemented process, such that the instructions which execute on the computer, other programmable apparatus, or other device implement the functions/acts specified in the flowchart and/or block diagram block or blocks.

[0085] The flowchart and block diagrams in the Figures illustrate the architecture, functionality, and operation of possible implementations of systems, methods, and computer program products according to various embodiments of the present technical solutions. In this regard, each block in the flowchart or block diagrams may represent a module, segment, or portion of instructions, which comprises one or more executable instructions for implementing the specified logical function(s). In some alternative implementations, the functions noted in the blocks may occur out of the order noted in the Figures. For example, two blocks shown in succession may, in fact, be executed substantially concurrently, or the blocks may sometimes be executed in the reverse order, depending upon the functionality involved. It will also be noted that each block of the block diagrams and/or flowchart illustration, and combinations of blocks in the block diagrams and/or flowchart illustration, can be implemented by special purpose hardware-based systems that perform the specified functions or acts or carry out combinations of special purpose hardware and computer instructions.

[0086] A second action may be said to be "in response to" a first action independent of whether the second action results directly or indirectly from the first action. The second action may occur at a substantially later time than the first action and still be in response to the first action. Similarly, the second action may be said to be in response to the first action even if intervening actions take place between the first action and the second action, and even if one or more of the intervening actions directly cause the second action to be performed. For example,

a second action may be in response to a first action if the first action sets a flag and a third action later initiates the second action whenever the flag is set.

[0087] To clarify the use of and to hereby provide notice to the public, the phrases "at least one of <A>, , ... and <N>" or "at least one of <A>, , ... <N>, or combinations thereof" or "<A>, , ... and/or <N>" are to be construed in the broadest sense, superseding any other implied definitions hereinbefore or hereinafter unless expressly asserted to the contrary, to mean one or more elements selected from the group comprising A, B, ... and N. In other words, the phrases mean any combination of one or more of the elements A, B, ... or N including any one element alone or the one element in combination with one or more of the other elements which may also include, in combination, additional elements not listed.

[0088] It will also be appreciated that any module, unit, component, server, computer, terminal or device exemplified herein that executes instructions may include or otherwise have access to computer readable media such as storage media, computer storage media, or data storage devices (removable and/or non-removable) such as, for example, magnetic disks, optical disks, or tape. Computer storage media may include volatile and non-volatile, removable and non-removable media implemented in any method or technology for storage of information, such as computer readable instructions, data structures, program modules, or other data. Such computer storage media may be part of the device or accessible or connectable thereto. Any application or module herein described may be implemented using computer readable/executable instructions that may be stored or otherwise held by such computer readable media.

[0089] The descriptions of the various embodiments of the technical features herein have been presented for purposes of illustration, but are not intended to be exhaustive or limited to the embodiments disclosed. Many modifications and variations will be apparent to those of ordinary skill in the art without departing from the scope and spirit of the described embodiments. The terminology used herein was chosen to best explain the principles of the embodiments, the practical application or technical improvement over technologies found in the marketplace, or to enable others of ordinary skill in the art to understand the embodiments disclosed herein.

CLAIMS

1. A computer implemented method for configuring a gateway in an IoT system, the method comprising:
providing, by an identification module, a user-interface (UI) access information to an administrative device;
receiving, by a setup module, from the administrative device, and via a management console generated based on the UI access information, setup data; and
adjusting, by the setup module, one or more settings of the gateway according to the setup data.
2. The method of claim 1, wherein the UI access information comprises a unique identifier of the gateway.
3. The method of claim 2, wherein the UI access information further comprises a communication address of a gateway resolver.
4. The method of claim 3, further comprising:
in response to the gateway starting up, sending, by the setup module, an IP address of the gateway to the gateway resolver.
5. The method of claim 3, further comprising:
in response to the gateway receiving an updated IP address, sending, by the setup module, the updated IP address of the gateway to the gateway resolver.
6. The method of claim 3, further comprising:
in response to receiving a request from the gateway resolver, the setup module:
generating the management console; and
sending the UI access information to the gateway resolver for accessing the management console.
7. The method of claim 1, wherein the identification module comprises a barcode associated with the gateway, the UI access information being captured by the administrative device using the barcode.
8. The method of claim 1, wherein the identification module comprises a radio frequency identification (RFID) device associated with the gateway, the UI access information being captured by the administrative device by the RFID device.
9. An internet of things (IoT) system comprising:
a gateway device configured to connect one or more application devices to corresponding application server, the gateway device comprising a memory device and a processor coupled with the memory device, the

processor configured to provide access to a management console for setting up the gateway device, providing the access comprising:

providing a user-interface (UI) access information to an administrative device to access the management console;
receiving setup data from the administrative device, via the management console; and
adjusting one or more settings of the gateway device according to the setup data.

10. The system of claim 9, wherein the UI access information comprises a unique identifier of the gateway device, and a communication address of a gateway resolver, the gateway resolver maintaining a mapping between the unique identifier of the gateway device and an IP address of the gateway device.

11. The system of claim 10, the processor further configured to send the IP address of the gateway device to the gateway resolver upon startup of the gateway device.

12. The system of claim 10, the processor further configured to send the IP address of the gateway device to the gateway resolver in response to the IP address of the gateway device being updated.

13. The system of claim 10, the processor further configured to, in response to receiving a request from the gateway resolver send a URL to access the management console to the gateway resolver for redirecting the administrative device to access the management console.

14. The system of claim 9, wherein the gateway device comprises a barcode associated with the gateway device, the UI access information being captured by the administrative device using the barcode.

15. The system of claim 9, wherein the gateway device comprises a radio frequency identification (RFID) device associated with the gateway device, the UI access information being captured by the administrative device by the RFID device.

16. A gateway device that is inaccessible to plug input-output devices into, the gateway device comprising a processor configured to provide a management console for configuring the gateway device, the providing comprising:

providing, by the processor, a user-interface (UI) access information to an administrative device;
receiving, by the processor, from the administrative device, and via a management console generated based on the UI access information, setup data; and
adjusting, by the processor, one or more settings of the gateway according to the setup data.

17. The gateway device of claim 16, wherein the UI access information comprises a unique identifier of the gateway device, and a communication address of a gateway resolver, the gateway resolver maintaining a mapping between the unique identifier of the gateway device and an IP address of the gateway device.
18. The gateway device of claim 17, the processor further configured to:
send the IP address of the gateway device to the gateway resolver upon startup of the gateway device;
and
send the IP address of the gateway device to the gateway resolver in response to the IP address of the gateway device being updated.
19. The gateway device of claim 17, the processor further configured to, in response to receiving a request from the gateway resolver send a URL to access the management console to the gateway resolver for redirecting the administrative device to access the management console.
20. The gateway device of claim 17, wherein the gateway device comprises at least one of:
a barcode associated with the gateway device, the UI access information being captured by the administrative device using the barcode; and
a radio frequency identification (RFID) device associated with the gateway device, the UI access information being captured by the administrative device by the RFID device.
21. An apparatus comprising:
a memory; and
a processor coupled with the memory, the processor configured to operate as a gateway device for an internet of things (IoT) system, the processor is configured to perform:
transporting data between an application device and an application server via a router; and
in response to powering on:
sending apparatus identification information to a gateway resolver using a predetermined URL for the gateway resolver; and
executing a management console accessible via a URL that has a predetermined format.
22. The apparatus of claim 21, wherein the apparatus identification information comprises a unique identifier of the apparatus and an IP address of the apparatus.
23. The apparatus of claim 21, wherein the processor is further configured to perform sending the gateway identification information to the gate resolver in response to expiration of a periodic timer.

24. A computer program product comprising a computer readable storage medium having program instructions embodied therewith, the program instructions executable by a processing circuit to cause the processing circuit to operate as a gateway device for an internet of things (IoT) system, the processing circuit is configured to perform:

transporting data between an application device and an application server via a router; and

in response to the processing circuit being powered on:

sending gateway identification information to a gateway resolver using a predetermined URL for the gateway resolver, the gateway identification information comprises a unique identifier assigned to the gateway device and an IP address assigned to the gateway device; and

executing a management console accessible via a URL that has a predetermined format.

25. The computer program product of claim 24, wherein the processing circuit is further configured to perform sending the gateway identification information to the gate resolver in response to expiration of a periodic timer.

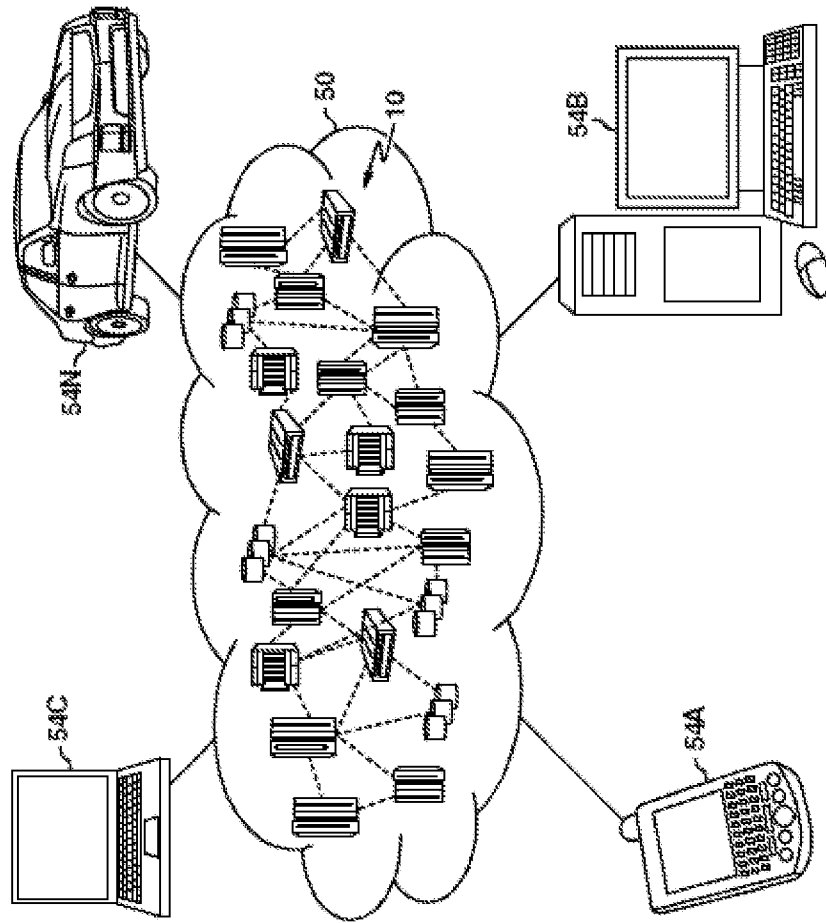


Fig. 1

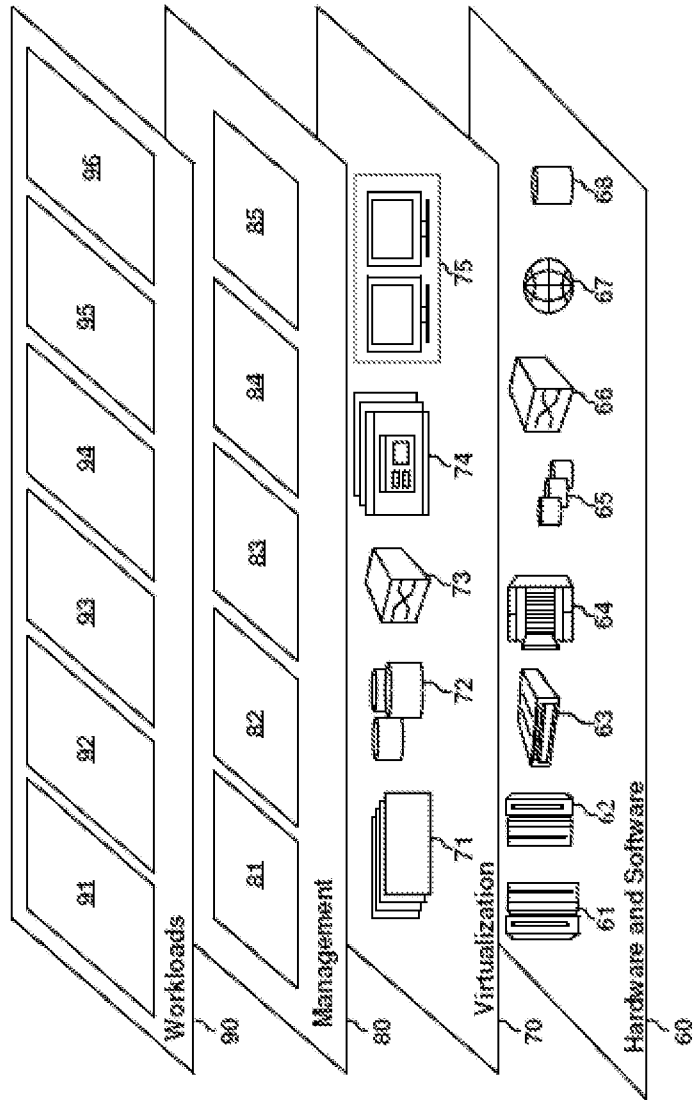


Fig. 2

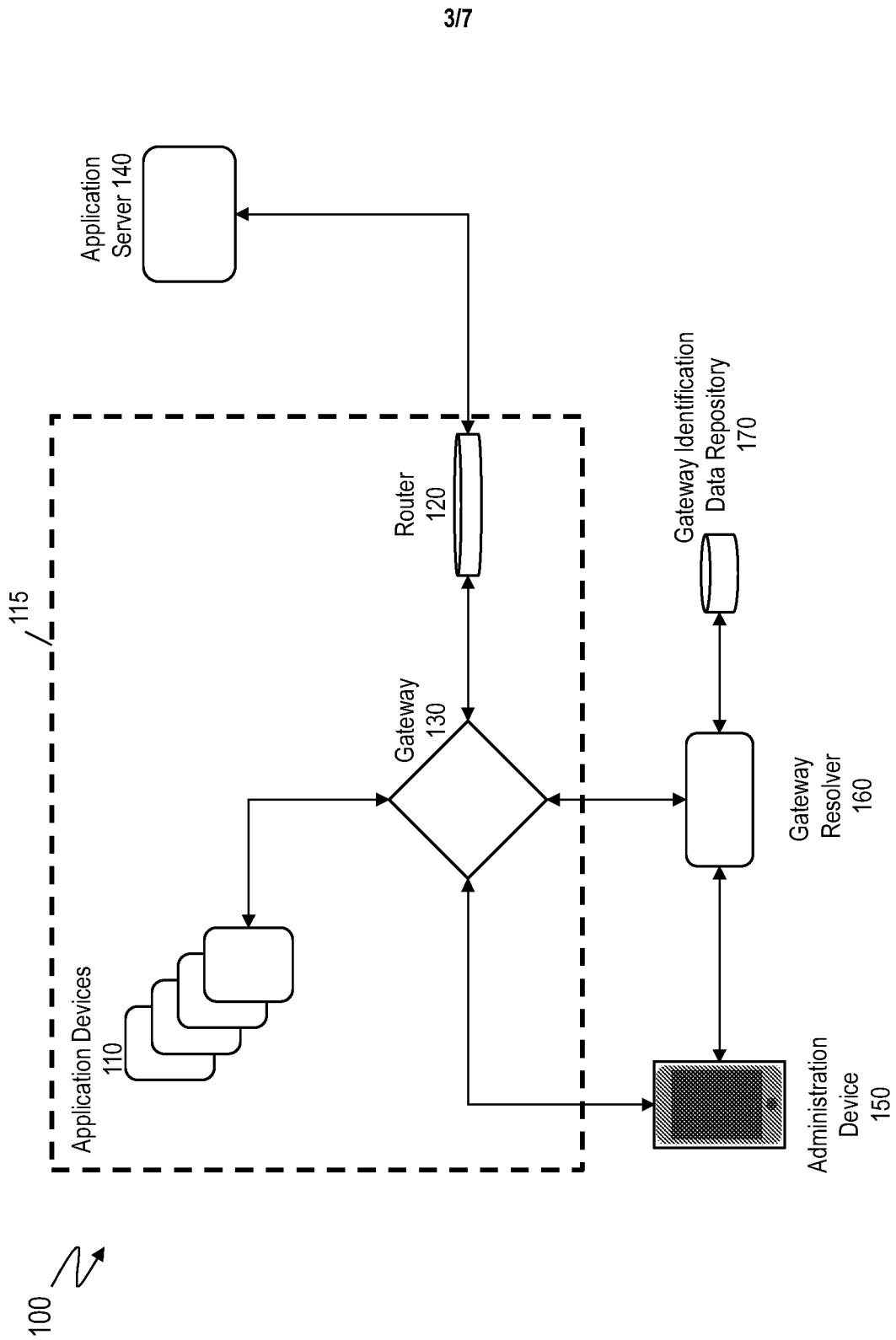


Fig. 3

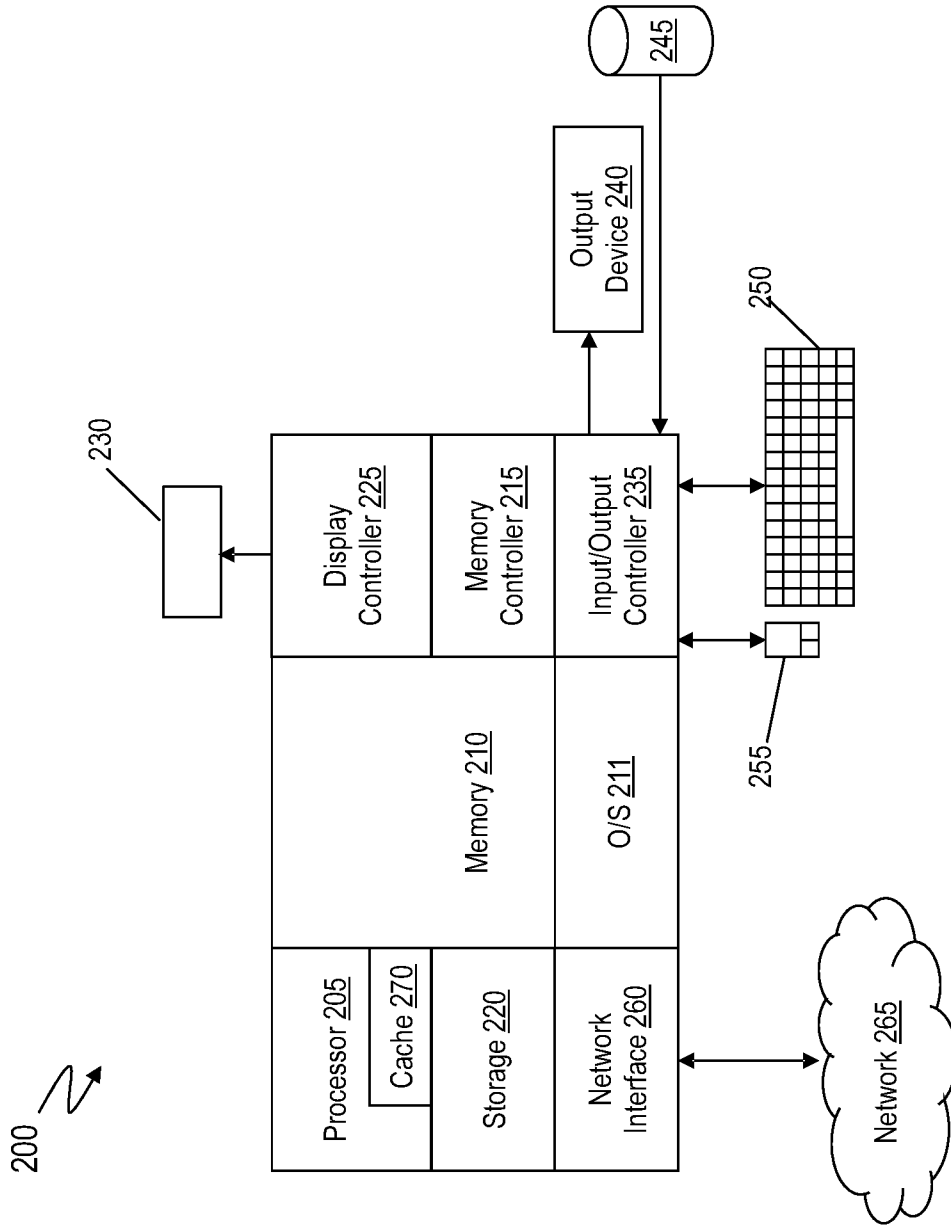


Fig. 4

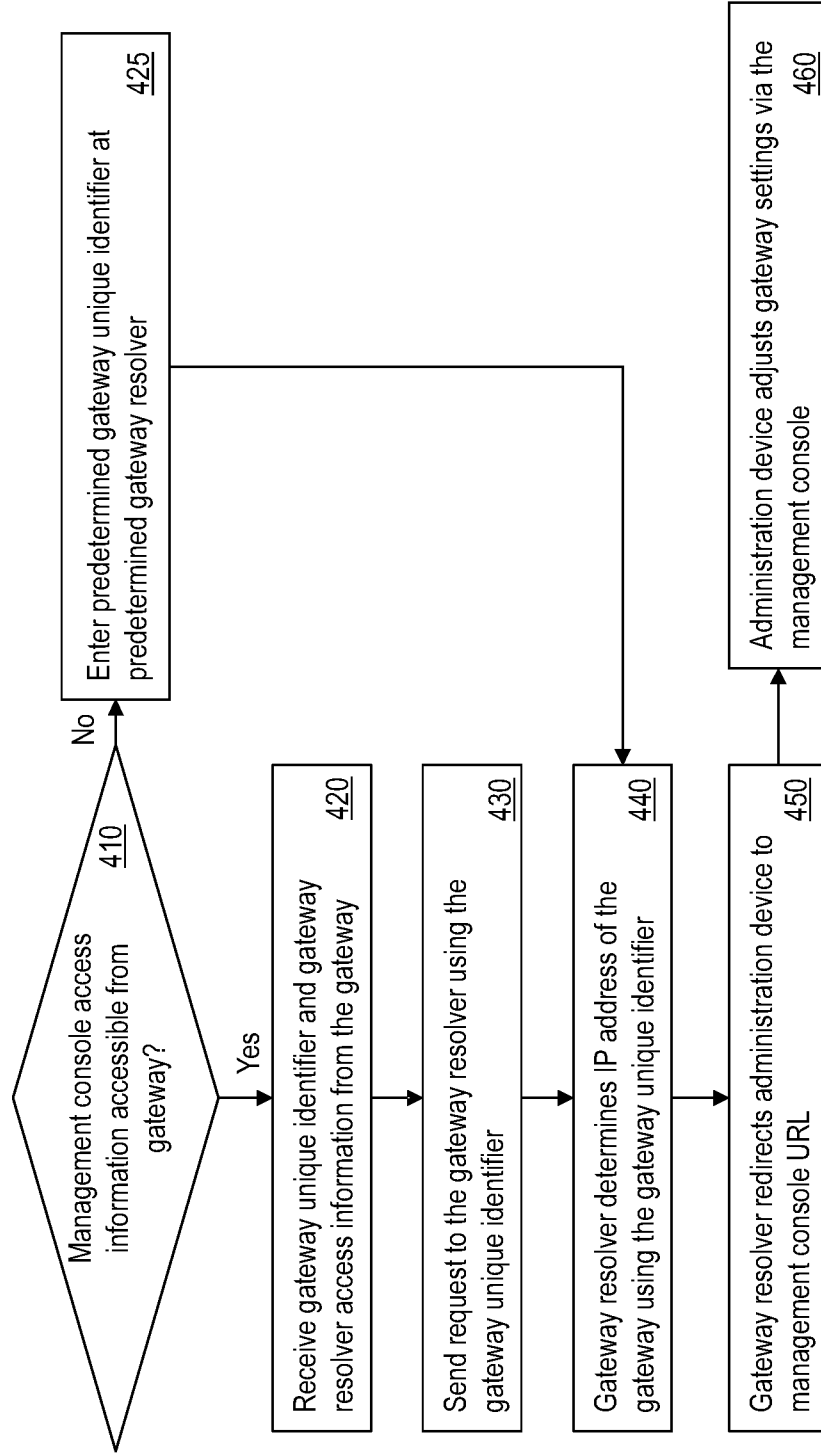


Fig. 5

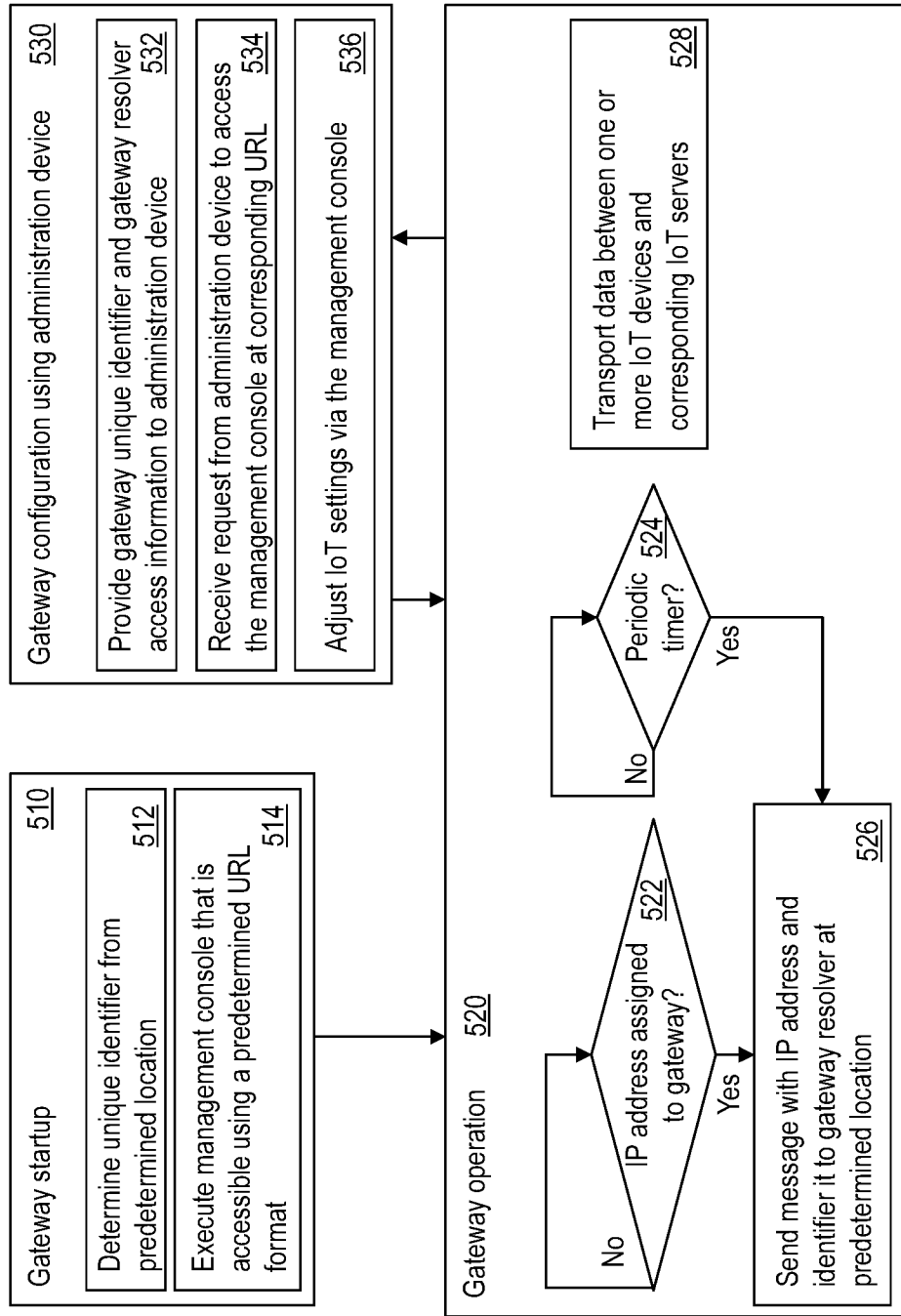


Fig. 6

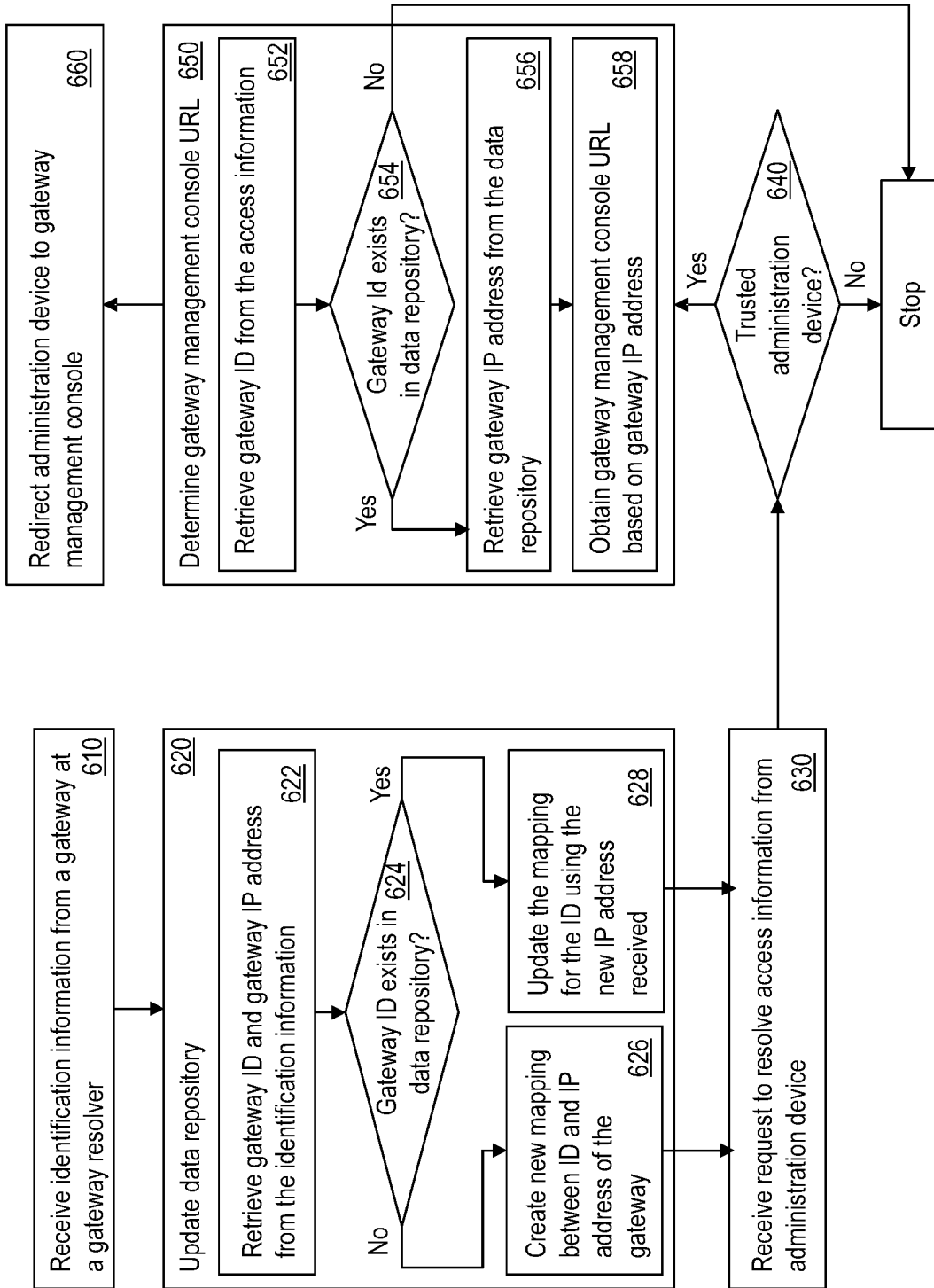


Fig. 7

INTERNATIONAL SEARCH REPORT

International application No.

PCT/IB2018/058692

A. CLASSIFICATION OF SUBJECT MATTER H04L 29/06(2006.01)i According to International Patent Classification (IPC) or to both national classification and IPC		
B. FIELDS SEARCHED Minimum documentation searched (classification system followed by classification symbols) H04L; G05B 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) CNABS; VEN; CPRSABS; CNKI; CNTXT: GATEWAY? RESOLVER REDIRECT+ DNS DOMAIN NAME SERVER IDENTIFI+ INTELLIGENT SMART HOME HOUSEHOLD LOT INTERNET THINGS UI GUI USER INTERFACE? CONSOLE? MANAGE+ CONFIGUR+ SET+ ADJUST+ CONTROL+ IP ADDRESS URL URI RFID RADIO FREQUENCY BARCODE		
C. DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	CN 102281251 A (ZTE CORP) 14 December 2011 (2011-12-14) description, paragraphs [0030]-[0167] and figures 1-8	1-25
A	CN 105763658 A (CHINA TELECOM CORP LTD) 13 July 2016 (2016-07-13) the whole document	1-25
A	CN 105163414 A (SHANGHAI INESA GROUP CO LTD) 16 December 2015 (2015-12-16) the whole document	1-25
A	US 2016036764 A1 (CONVIDA WIRELESS LLC) 04 February 2016 (2016-02-04) the whole document	1-25
A	CN 103327083 A (UNIV CHANGZHOU) 25 September 2013 (2013-09-25) the whole document	1-25
<input type="checkbox"/> Further documents are listed in the continuation of Box C. <input checked="" type="checkbox"/> 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 23 January 2019		Date of mailing of the international search report 30 January 2019
Name and mailing address of the ISA/CN National Intellectual Property Administration, PRC 6, Xitucheng Rd., Jimen Bridge, Haidian District, Beijing 100088 China		Authorized officer HU, Yan
Facsimile No. (86-10)62019451		Telephone No. 62089101

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.

PCT/IB2018/058692

Patent document cited in search report			Publication date (day/month/year)	Patent family member(s)			Publication date (day/month/year)
CN	102281251	A	14 December 2011	CN	102281251	B	17 December 2014
				WO	2011153737	A1	15 December 2011

CN	105763658	A	13 July 2016	None			

CN	105163414	A	16 December 2015	None			

US	2016036764	A1	04 February 2016	WO	2016019287	A3	24 March 2016
				EP	3175602	A2	07 June 2017
				WO	2016019287	A2	04 February 2016
				KR	20170039262	A	10 April 2017
				JP	2017525298	A	31 August 2017
				CN	106797409	A	31 May 2017
				IN	IN201717003749	A	04 April 2017

CN	103327083	A	25 September 2013	None			
