AUTOMATIC PROVISIONING OF IOT DEVICES

BACKGROUND

[0001]The Internet of Things ("IoT") generally refers to a system of devices capable of communicating over a network including the communications of data over the network. The devices can include everybody objects such as toasters, coffee machines, thermostat systems, washers, dryers, lamps, automobiles, and the like. The network communications can be used for device automation, data capture, providing alerts, personalization of settings, and numerous other applications.

SUMMARY OF THE DISCLOSURE

[0002]This Summary is provided to introduce a selection of concepts in a simplified form that are further described below in the Detailed Description. This Summary is not intended to identify key features or essential features of the claimed subject matter, nor is it intended to be used to limit the scope of the claimed subject matter.

[0003]Briefly stated, the disclosed technology is generally directed to IoT communications. For example, such technology is usable in provisioning IoT devices in an automatic manner with no manual steps. In one example of the technology, upon initial boot, identification information is automatically sent to a provisioning service endpoint stored in the IoT device. The identification information includes an identification (ID) of the first IoT device. Cryptographic information is received from the provisioning service. The cryptographic information is associated with an IoT hub selected from a plurality of IoT hubs based, in part, on the ID of the first IoT device. A message is automatically sent to the IoT hub in response to receiving the cryptographic information. A new configuration file and a firmware update are received from the IoT hub without requiring a user association. The new configuration file and the firmware update are automatically installed.

[0004]In some examples, in a zero-touch manner with no manual steps required, an IoT device connects to a global endpoint, is routed to the appropriate cloud-based IoT solution, and then automatically has configuration data and firmware updates applied specific to that IoT solution. Automatic configuration and firmware update occurs once a device is first connected to an IoT solution without requiring a user association with the device. The IoT device is provisioned and fully updated without human intervention. Multiple types of hardware/operating system (OS) combinations may connect to the same global endpoint, as opposed to being specific to hardware and only connecting to one backend solution. In some examples, all connections to and from the provisioning service are secured. Also, multiple

IoT solutions may be connected to by one provisioning service. Some examples are low-touch rather than zero-touch—that is, in these examples, there is minimal human action rather than no human action.

[0005]Other aspects of and applications for the disclosed technology will be appreciated upon reading and understanding the attached figures and description.

BRIEF DESCRIPTION OF THE DRAWINGS

[0006]Non-limiting and non-exhaustive examples of the present disclosure are described with reference to the following drawings. In the drawings, like reference numerals refer to like parts throughout the various figures unless otherwise specified. These drawings are not necessarily drawn to scale.

[0007]For a better understanding of the present disclosure, reference will be made to the following Detailed Description, which is to be read in association with the accompanying drawings, in which:

[0008]FIG. 1 is a block diagram illustrating one example of a suitable environment in which aspects of the technology may be employed;

[0009]FIG. 2 is a block diagram illustrating one example of a suitable computing device according to aspects of the disclosed technology;

[0010]FIG. 3 is a block diagram illustrating an example of a system for IoT communications;

[0011]FIG. 4 is a diagram illustrating an example dataflow for a process for IoT communications;

[0012]FIG. 5 is a block diagram illustrating an example of a portion of the system of FIG. 3;

[0013]FIG. 6 is a block diagram illustrating yet another example of a portion of the system of FIG. 3;

[0014]FIG. 7 is a logical flow diagram illustrating an example of a process for IoT communications;

[0015]FIG. 8 is a logical flow diagram illustrating an example of another process for IoT communications; and

[0016]FIG. 9 is a logical flow diagram illustrating an example of yet another process for IoT communications, in accordance with aspects of the present disclosure.

DETAILED DESCRIPTION

[0017]The following description provides specific details for a thorough understanding of, and enabling description for, various examples of the technology. One skilled in the art will

understand that the technology may be practiced without many of these details. In some instances, well-known structures and functions have not been shown or described in detail to avoid unnecessarily obscuring the description of examples of the technology. It is intended that the terminology used in this disclosure be interpreted in its broadest reasonable manner, even though it is being used in conjunction with a detailed description of certain examples of the technology. Although certain terms may be emphasized below, any terminology intended to be interpreted in any restricted manner will be overtly and specifically defined as such in this Detailed Description section. Throughout the specification and claims, the following terms take at least the meanings explicitly associated herein, unless the context dictates otherwise. The meanings identified below do not necessarily limit the terms, but merely provide illustrative examples for the terms. For example, each of the terms "based on" and "based upon" is not exclusive, and is equivalent to the term "based, at least in part, on", and includes the option of being based on additional factors, some of which may not be described herein. As another example, the term "via" is not exclusive, and is equivalent to the term "via, at least in part", and includes the option of being via additional factors, some of which may not be described herein. The meaning of "in" includes "in" and "on." The phrase "in one embodiment," or "in one example," as used herein does not necessarily refer to the same embodiment or example, although it may. Use of particular textual numeric designators does not imply the existence of lesser-valued numerical designators. For example, reciting "a widget selected from the group consisting of a third foo and a fourth bar" would not itself imply that there are at least three foo, nor that there are at least four bar, elements. References in the singular are made merely for clarity of reading and include plural references unless plural references are specifically excluded. The term "or" is an inclusive "or" operator unless specifically indicated otherwise. For example, the phrases "A or B" means "A, B, or A and B." As used herein, the terms "component" and "system" are intended to encompass hardware, software, or various combinations of hardware and software. Thus, for example, a system or component may be a process, a process executing on a computing device, the computing device, or a portion thereof.

[0018]Briefly stated, the disclosed technology is generally directed to IoT communications. For example, such technology is usable in provisioning IoT devices in automatic manner with no manual steps. In one example of the technology, upon initial boot, identification information is automatically sent to a provisioning service endpoint stored in the IoT device. The identification information includes an identification (ID) of the first IoT device.

Cryptographic information is received from the provisioning service. The cryptographic information is associated with an IoT hub selected from a plurality of IoT hubs based, in part, on the ID of the first IoT device. A message is automatically sent to the IoT hub in response to receiving the cryptographic information. A new configuration file and a firmware update are received from the IoT hub without requiring a user association. The new configuration file and the firmware update are automatically installed.

[0019]Setting up a large number of IoT devices may be difficult and may involve a considerable amount of manual work. For an off-the-shelf smart device, the device may require some way of knowing which cloud back-end is one to which it should be connected. Then the device may be required be able to connect securely to that cloud back-end, which may involve customers having to individually place connection credentials on the device. The device may have been sitting in a warehouse for a year or longer, which means the onboard firmware is likely out of date and may need to be updated to ensure device security.

[0020]These problems may be faced by all IoT devices. In some other technologies, even though customers can create many device identities within the IoT hub at a time using bulk import, they still have to individually place connection credentials on the devices themselves. Some other technologies have accepted the manual work to connect devices or have required the device be enrolled in a management interface with a user association by which devices are traditionally managed. These other technologies involve substantial manual work as they involve manual steps taken between device manufacture and end use to get the device ready.

[0021]Some examples of the disclosure provide, in an automatic manner with no manual steps required, an IoT device connecting to a global endpoint, is routed to the appropriate cloud-based IoT solution, and then automatically having configuration data and firmware updates applied specific to that IoT solution. Automatic configuration and firmware update occurs once a device is first connected to an IoT solution without requiring a user association with the device. The IoT device is provisioned and fully updated without human intervention. Multiple types of hardware/operating system (OS) combinations may connect to the same global endpoint, as opposed to being specific to hardware and only connecting to one backend solution. In some examples, all connections to and from a provisioning service are secured. In some examples, the provisioning service is a globally available cloud service that acts as a global endpoint in the cloud; in other examples, the provisioning service is an endpoint within a user's subscription per service endpoint. Also, multiple IoT solutions may be connected to by one provisioning service.

Illustrative Devices/Operating Environments

[0022]FIG. 1 is a diagram of environment 100 in which aspects of the technology may be practiced. As shown, environment 100 includes computing devices 110, as well as network nodes 120, connected via network 130. Even though particular components of environment 100 are shown in FIG. 1, in other examples, environment 100 can also include additional and/or different components. For example, in certain examples, the environment 100 can also include network storage devices, maintenance managers, and/or other suitable components (not shown). Computing devices 110 shown in FIG. 1 may be in various locations, including on premise, in the cloud, or the like. For example, computer devices 110 may be on the client side, on the server side, or the like.

[0023]As shown in FIG. 1, network 130 can include one or more network nodes 120 that interconnect multiple computing devices 110, and connect computing devices 110 to external network 140, e.g., the Internet or an intranet. For example, network nodes 120 may include switches, routers, hubs, network controllers, or other network elements. In certain examples, computing devices 110 can be organized into racks, action zones, groups, sets, or other suitable divisions. For example, in the illustrated example, computing devices 110 are grouped into three host sets identified individually as first, second, and third host sets 112a-112c. In the illustrated example, each of host sets 112a-112c is operatively coupled to a corresponding network node 120a-120c, respectively, which are commonly referred to as "top-of-rack" or "TOR" network nodes. TOR network nodes 120a-120c can then be operatively coupled to additional network nodes 120 to form a computer network in a hierarchical, flat, mesh, or other suitable types of topology that allows communications between computing devices 110 and external network 140. In other examples, multiple host sets 112a-112c may share a single network node 120. Computing devices 110 may be virtually any type of general- or specific-purpose computing device. For example, these computing devices may be user devices such as desktop computers, laptop computers, tablet computers, display devices, cameras, printers, or smartphones. However, in a data center environment, these computing devices may be server devices such as application server computers, virtual computing host computers, or file server computers. Moreover, computing devices 110 may be individually configured to provide computing, storage, and/or other suitable computing services.

[0024]In some examples, one or more of the computing devices 110 is an IoT device, a device that comprises part or all of an IoT hub, a device comprising part or all of a provisioning service, or the like, as discussed in greater detail below.

Illustrative Computing Device

[0025]FIG. 2 is a diagram illustrating one example of computing device 200 in which aspects of the technology may be practiced. Computing device 200 may be virtually any type of general- or specific-purpose computing device. For example, computing device 200 may be a user device such as a desktop computer, a laptop computer, a tablet computer, a display device, a camera, a printer, or a smartphone. Likewise, computing device 200 may also be server device such as an application server computer, a virtual computing host computer, or a file server computer, e.g., computing device 200 may be an example of computing device 110 or network node 120 of FIG. 1. Computing device 200 may also be an IoT device that connects to a network to receive IoT services. Likewise, computer device 200 may be an example any of the devices illustrated in FIGs. 3, 5, and 6, as discussed in greater detail below. As illustrated in FIG. 2, computing device 200 includes processing circuit 210, operating memory 220, memory controller 230, data storage memory 250, input interface 260, output interface 270, and network adapter 280. Each of these afore-listed components of computing device 200 includes at least one hardware element.

[0026]Computing device 200 includes at least one processing circuit 210 configured to execute instructions, such as instructions for implementing the herein-described workloads, processes, or technology. Processing circuit 210 may include a microprocessor, a microcontroller, a graphics processor, a coprocessor, a field-programmable gate array, a programmable logic device, a signal processor, or any other circuit suitable for processing data. The aforementioned instructions, along with other data (e.g., datasets, metadata, operating system instructions, etc.), may be stored in operating memory 220 during run-time of computing device 200. Operating memory 220 may also include any of a variety of data storage devices/components, such as volatile memories, semi-volatile memories, random access memories, static memories, caches, buffers, or other media used to store run-time information. In one example, operating memory 220 does not retain information when computing device 200 is powered off. Rather, computing device 200 may be configured to transfer instructions from a non-volatile data storage component (e.g., data storage component 250) to operating memory 220 as part of a booting or other loading process.

[0027]Operating memory 220 may include 4th generation double data rate (DDR4) memory, 3rd generation double data rate (DDR3) memory, other dynamic random access memory (DRAM), High Bandwidth Memory (HBM), Hybrid Memory Cube memory, 3D-stacked memory, static random access memory (SRAM), or other memory, and such memory may comprise one or more memory circuits integrated onto a DIMM, SIMM, SODIMM, or other

packaging. Such operating memory modules or devices may be organized according to channels, ranks, and banks. For example, operating memory devices may be coupled to processing circuit 210 via memory controller 230 in channels. One example of computing device 200 may include one or two DIMMs per channel, with one or two ranks per channel. Operating memory within a rank may operate with a shared clock, and shared address and command bus. Also, an operating memory device may be organized into several banks where a bank can be thought of as an array addressed by row and column. Based on such an organization of operating memory, physical addresses within the operating memory may be referred to by a tuple of channel, rank, bank, row, and column.

[0028]Despite the above-discussion, operating memory 220 specifically does not include or encompass communications media, any communications medium, or any signals *per se*.

[0029]Memory controller 230 is configured to interface processing circuit 210 to operating memory 220. For example, memory controller 230 may be configured to interface commands, addresses, and data between operating memory 220 and processing circuit 210. Memory controller 230 may also be configured to abstract or otherwise manage certain aspects of memory management from or for processing circuit 210. Although memory controller 230 is illustrated as single memory controller separate from processing circuit 210, in other examples, multiple memory controllers may be employed, memory controller(s) may be integrated with operating memory 220, or the like. Further, memory controller(s) may be integrated into processing circuit 210. These and other variations are possible.

[0030]In computing device 200, data storage memory 250, input interface 260, output interface 270, and network adapter 280 are interfaced to processing circuit 210 by bus 240. Although, FIG. 2 illustrates bus 240 as a single passive bus, other configurations, such as a collection of buses, a collection of point to point links, an input/output controller, a bridge, other interface circuitry, or any collection thereof may also be suitably employed for interfacing data storage memory 250, input interface 260, output interface 270, or network adapter 280 to processing circuit 210.

[0031]In computing device 200, data storage memory 250 is employed for long-term non-volatile data storage. Data storage memory 250 may include any of a variety of non-volatile data storage devices/components, such as non-volatile memories, disks, disk drives, hard drives, solid-state drives, or any other media that can be used for the non-volatile storage of information. However, data storage memory 250 specifically does not include or encompass communications media, any communications medium, or any signals *per se*. In contrast to

operating memory 220, data storage memory 250 is employed by computing device 200 for non-volatile long-term data storage, instead of for run-time data storage.

[0032] Also, computing device 200 may include or be coupled to any type of processor-readable media such as processor-readable storage media (e.g., operating memory 220 and data storage memory 250) and communication media (e.g., communication signals and radio waves). While the term processor-readable storage media includes operating memory 220 and data storage memory 250, the term "processor-readable storage media," throughout the specification and the claims whether used in the singular or the plural, is defined herein so that the term "processor-readable storage media" specifically excludes and does not encompass communications media, any communications medium, or any signals *per se*. However, the term "processor-readable storage media" does encompass processor cache, Random Access Memory (RAM), register memory, and/or the like.

[0033] Computing device 200 also includes input interface 260, which may be configured to enable computing device 200 to receive input from users or from other devices. In addition, computing device 200 includes output interface 270, which may be configured to provide output from computing device 200. In one example, output interface 270 includes a frame buffer, graphics processor, graphics processor or accelerator, and is configured to render displays for presentation on a separate visual display device (such as a monitor, projector, virtual computing client computer, etc.). In another example, output interface 270 includes a visual display device and is configured to render and present displays for viewing.

[0034] In the illustrated example, computing device 200 is configured to communicate with other computing devices or entities via network adapter 280. Network adapter 280 may include a wired network adapter, e.g., an Ethernet adapter, a Token Ring adapter, or a Digital Subscriber Line (DSL) adapter. Network adapter 280 may also include a wireless network adapter, for example, a Wi-Fi adapter, a Bluetooth adapter, a ZigBee adapter, a Long Term Evolution (LTE) adapter, or a 5G adapter.

[0035] Although computing device 200 is illustrated with certain components configured in a particular arrangement, these components and arrangement are merely one example of a computing device in which the technology may be employed. In other examples, data storage memory 250, input interface 260, output interface 270, or network adapter 280 may be directly coupled to processing circuit 210, or be coupled to processing circuit 210 via an input/output controller, a bridge, or other interface circuitry. Other variations of the technology are possible.

[0036]Some examples of computing device 200 include at least one storage memory (e.g. data storage memory 250), at least one operating memory (e.g., operating memory 220) and at least one processor (e.g., processing unit 210) that are respectively adapted to store and execute processor-executable code that, in response to execution, enables computing device 200 to perform actions. In some examples, computing device 200 is enabled to perform actions such as the actions in process 700, process 800, or process 900 below, or actions in a process performed by one or more of the computing devices in FIG. 3, FIG. 5, or FIG. 6 below.

Illustrative System

[0037]FIG. 3 is a block diagram illustrating an example of a system (300) for IoT communications. System 300 may include network 330, IoT hubs 351-353, IoT devices 341-343, and provisioning services 311-312, which all connect to network 330. The term "IoT device" refers to a device intended to make use of IoT services. An IoT device can include virtually any device that connects to the cloud to use IoT services, including for telemetry collection or any other purpose. Each of the provisioning services 311-312 includes one or more devices, such as a distributed system in some examples. The term "IoT hub" refers to a device, or multiple devices such as a distributed system, to which IoT devices connect on the network for IoT services after provisioning. Each of the IoT devices 341-343 and/or the devices that comprise IoT hubs 351-353 and provisioning services 311-312 may include examples of computing device 200 of FIG. 2. The term "IoT hub" is not limited to one particular type of IoT service, but refers to the device to which the IoT device communicates, after provisioning, for any type of IoT solution or IoT service. That is, the term "IoT hub," as used throughout the specification and the claims, is generic to any IoT solution. FIG. 3 and the corresponding description of FIG. 3 in the specification illustrates an example system for illustrative purposes that does not limit the scope of the disclosure.

[0038]Network 330 may include one or more computer networks, including wired and/or wireless networks, where each network may be, for example, a wireless network, local area network (LAN), a wide-area network (WAN), and/or a global network such as the Internet. On an interconnected set of LANs, including those based on differing architectures and protocols, a router acts as a link between LANs, enabling messages to be sent from one to another. Also, communication links within LANs typically include twisted wire pair or coaxial cable, while communication links between networks may utilize analog telephone lines, full or fractional dedicated digital lines including T1, T2, T3, and T4, Integrated Services Digital Networks (ISDNs), Digital Subscriber Lines (DSLs), wireless links

including satellite links, or other communications links known to those skilled in the art. Furthermore, remote computers and other related electronic devices could be remotely connected to either LANs or WANs via a modem and temporary telephone link. In essence, network 330 includes any communication method by which information may travel between IoT hubs 351-353, IoT devices 341-343, and provisioning services 311-312.

[0039]As one example, IoT devices 341-343 are devices that are intended to make use of IoT services provided by one or more IoT hubs, such as IoT hubs 351-353. Provisioning services 311-312 are devices or sets of devices (such as distributed systems) that perform actions in automatically provisioning an IoT device to an IoT hub.

[0040]System 300 may include more or less devices than illustrated in FIG. 3, which is shown by way of example only.

Illustrative Processes

[0041]For clarity, the processes described herein are described in terms of operations performed in particular sequences by particular devices or components of a system. However, it is noted that other processes are not limited to the stated sequences, devices, or components. For example, certain acts may be performed in different sequences, in parallel, omitted, or may be supplemented by additional acts or features, whether or not such sequences, parallelisms, acts, or features are described herein. Likewise, any of the technology described in this disclosure may be incorporated into the described processes or other processes, whether or not that technology is specifically described in conjunction with a process. The disclosed processes may also be performed on or by other devices, components, or systems, whether or not such devices, components, or systems are described herein. These processes may also be embodied in a variety of ways. For example, they may be embodied on an article of manufacture, e.g., as processor-readable instructions stored in a processor-readable storage medium or be performed as a computer-implemented process. As an alternate example, these processes may be encoded as processor-executable instructions and transmitted via a communications medium.

[0042]FIG. 4 is a diagram illustrating an example dataflow for a process (420) for IoT communications. FIG. 4 and the corresponding description of FIG. 4 in the specification illustrates an example process for illustrative purposes that does not limit the scope of the disclosure.

[0043]In some examples, IoT device 441 and provisioning service 411 have the following starting point. First, IoT device 441 stores the endpoint it is to connect with in order to be automatically provisioned. For instance, the endpoint uniform resource indicator (URI) may

be installed in IoT device 441 the factory. In some examples, on initial power-up and initial boot-up, IoT device 441 is cryptographically guaranteed to connect to provisioning service 411 only. Also, IoT device 441 stores identity information about itself as well as optional metadata, which may include geolocation, in some examples. Throughout the specification and the claims, the terms “initial boot” and “initial boot-up” refer to the very first time that the IoT device is booted up after manufacture of the IoT device is complete.

[0044] Further, provisioning service 411 may have some method to verify the identity of IoT device 441. The source used to verify the identity of IoT device 411 may provide provisioning service 411 with additional metadata. Provisioning service device 411 may also contain a rule engine used to route an IoT device's provisioning request to the correct IoT hub. For example, one rule may be for all IoT devices within a certain geographic region to be provisioned to an IoT solution located in a certain region. Provisioning service device 411 may be configured with information about how to connect IoT devices to one or more IoT hubs each corresponding to a separate IoT solution.

[0045] In the illustrated example, upon IoT device 441 first being powered on by a customer, step 421 occurs. At step 421, an identification message may be communicated from IoT device 441 to provisioning service 411.

[0046]In some examples, IoT device 441 is manufactured with the URI of provisioning service 411. In some of these examples, step 421 happens upon IoT device 441 first being powered on. Upon IoT device 441 first being powered on, IoT device 441 may send the identification message to provisioning service 411 via the URI of provisioning service 441.

[0047] The identification information includes information that is usable for verifying that IoT device 441 is a valid device to receive IoT services, and may also include information for determining which IoT solution is appropriate for IoT device 441, such as geographical information.

[0048]As shown, step 422 occurs next. At step 422, the provisioning service determines whether or not IoT device 441 is valid. The validity determination is made in different ways in different examples, which will be discussed in greater detail below. If the provisioning service determines that IoT device 441 is not valid, the process ends.

[0049]If, instead, provisioning service 411 determines that IoT device 441 is valid, step 423 occurs. At step 423, provisioning service 411 selects an IoT hub from a plurality of IoT hubs. In some examples, the selection of the IoT hub is based on routing rules. In some examples, geographical location may be a factor in the selection of the IoT hub. For example, the closest appropriate IoT hub may be selected in some examples. Another factor in the selection of the IoT hub may be dependent on which IoT solution is appropriate based on factors relevant to the IoT device and determined by the IoT device manufacturer. For

example, all smart building IoT devices from a manufacturer may use a particular IoT solution and therefore select the corresponding IoT hub, while smart toasters from that manufacturer may go to a different IoT solution and therefore select the corresponding IoT hub.

[0050]In this example, step 424 occurs next. At step 424, a request to register IoT device 441 may be communicated from provisioning service 411 to the selected IoT hub (IoT hub 451). In some examples, the request to register IoT device 441 includes connection information associated with IoT device 441. Next, step 425 occurs. At step 425, IoT hub 451 may register IoT device 441 in a device registry in IoT hub 451. In some examples, IoT hub 451 registers IoT device 441 in both a device identity registry and a device management registry. In other examples, the device identity registry and the device management registry are combined, and IoT hub 451 registers IoT device 441 in one registry rather than two.

[0051]Accordingly, in some examples, the device registry is at least one of a device identity registry, a device management registry, a device identity registry and a device management registry, or a combined device registry. As part of the registration at step 425, in some examples, IoT hub 451 creates a separate ID for IoT device 441. By creating a separate ID for IoT device 441, IoT hub has an ID for IoT device 441 that maps to IoT device 441 so that IoT hub 451 can properly communicate with IoT device 441.

[0052]In this example, step 426 occurs next. At step 426, cryptographic information about IoT device 441 is communicated from IoT hub 451 to provisioning service 411. As part of this communication, in some examples, IoT hub 451 queues up commands for IoT device 441, or queue up commands to be sent for IoT device 441 to complete subsequently. The cryptographic information may also include credentials, the hostname of the selected IoT hub 451, connectivity information required for IoT device 441 to connect with IoT hub 451, and the like. In this example, step 427 occurs next. At step 427, the cryptographic information about IoT device 441 is communicated from provisioning service 411 to IoT device 441.

[0053]In some examples, step 428 occurs next. At step 428, a message is automatically communicated from IoT device 441 to IoT hub 451 in response to IoT device 441 receiving the cryptographic information. In some examples, step 429 occurs next. At step 429, an update is automatically communicated. That is, in some examples, in response to receiving the first message, a new configuration file and a new firmware update are automatically communicated from IoT hub 451 to IoT device 441 without requiring user association. In some examples, step 431 occurs next. At step 431, the update is installed. That is, in some

examples, the new configuration file and the firmware update are automatically installed in IoT device 441. Throughout the specification and the claims, the term “automatically” refers to performing an action automatically without an intervening manual step.

[0054]As a result of process 420, IoT device 441 is provisioned, is fully able to communicate with IoT services, is up-to-date, and is fully updated with the latest firmware and the latest configuration file, even though IoT device 441 may have been sitting in a warehouse for months or years, all of which is accomplished in an automatic manner without any manual steps, and without human intervention.

[0055]After provisioning is complete, in some examples, communications between IoT device 441 and IoT hub 451 may occur directly and in a normal fashion, and provisioning service 411 are not again involved in communications between IoT device 441 and IoT hub 451, unless, in some examples, IoT device 441 needs to be re-provisioned.

[0056]In some examples, IoT device 441 retains cryptographic memory of provisioning service 411 and can be redirected to provisioning service 411 during the lifetime of IoT device 441 in order to re-provision IoT device 411. In some examples, certain events may cause IoT device 441 to initiate re-provisioning, such as IoT device 441 being resold, a change in geographical regions, or the like.

[0057]In some examples, re-provisioning of an IoT device may be performed as follows. First, a determination is made as to which new IoT hub the IoT device should be attached to (in base data). The term "base data" as used herein refers to data and/or metadata about the IoT device to be used when provisioning the IoT device, such as, for example, location information such as geolocation, device identity information, other suitable metadata, and/or the like. Next, the IoT device is provisioned in the new IoT hub. Then, the new connection information is returned. The IoT device is then deleted from the registry of the old IoT hub.

[0058]In some examples, as a security measure, provisioning service 411 may be restricted from directly connecting to a device without first being contacted by that device. In other examples, provisioning service 411 can directly connect to IoT device 441 without being contacted by IoT device 441, and the security is ensured in some other manner.

[0059JFIG. 5 is a block diagram illustrating an example of a portion (501) of system 300 of FIG. 3. Portion 501 includes IoT device 541, provisioning service 511, and IoT hubs 551 and 552. As shown, provisioning service 511 includes routing rules 591 and enrollment list 592. Also, IoT hub 551 includes device registry 593 and IoT hub 552 includes device registry 594.

[0060]In some examples, IoT device 541 is manufactured with the URI of provisioning service 511 installed therein.

[0061]In some examples, IoT device 541 is also manufactured with identification information for IoT device 541 installed therein. In some of these examples, IoT device 541 is also manufactured with other manufacturer set data. The identification information may include the device identity (ID), the manufacturer set data, and, in some examples, base data including other information that may be relevant in terms of selecting an IoT solution, such as geographical data. In some examples, the device ID is known by the manufacturer of the device.

[0062]In some examples, the manufacturer makes a list of device ideas for IoT devices eligible to use the IoT services available to provisioning service 511 via an uploaded file, or the like, so that the device ID of each IoT device can be validated.

[0063]At step 5-1, IoT device 541 contacts the provisioning service endpoint (of provisioning service 511) set at the factory. The device ID and, optionally, other manufacturer-set data are passed as part of the call.

[0064]Next, at step 5-2, provisioning service 511 ascertains the validity of IoT device 541 by validating the device ID and, optionally, other manufacturer-set data against the uploaded base data. In some examples, provisioning service 511 also looks up IoT device 541 in the source of base data to find out metadata/hub data about IoT device 541 if such data is present.

[0065]The validation at step 5-2 may be performed in different ways in different examples. In some examples, enrollment list 592 may include all devices built by a manufacturer that uses one or more IoT solutions associated with the provisioning service that have been programmed with provisioning service 511 as the endpoint to use on first boot up. In other examples, enrollment list 592 may include only devices sold, not all devices built, that use provisioning service 511 as the endpoint for provisioning. In some examples, provisioning service 511 verifies the identity by determining whether or not the provided device ID is a device ID included in enrollment list 592. In some examples, other steps are necessary to confirm the device identity. For example, other data provided by IoT device 541 may also be used in the verification.

[0066]Next, at step 5-3, provisioning service 511 runs routing rules over the data from IoT device 541 as well as data from the base data source to find the right IoT hub to register IoT device 541 with. Provisioning service device 511 registers IoT device 541 with the selected IoT hub's (551) identity registry and device management (DM) registry. In some examples, instead of two separate registries, there is one device registry to which IoT device 541 is registered.

[0067]Next, at step 5-4, IoT hub 551 returns cryptographic information about IoT device 541 to provisioning service 511.

[0068]Next, at step 5-5, provisioning service 511 returns the cryptographic information to IoT device 541.

[0069]IoT device 541 can now send data directly to IoT hub 551 at step 5-6.

[0070]Next, at step 5-7, the metadata of IoT device 541 syncs with the metadata stored in the IoT hub's (541) device management (DM) registry via DM syncing.

[0071]FIG. 6 is a block diagram illustrating an example of a portion (601) of system 300 of FIG. 3. Portion 601 includes IoT device 641, provisioning service 611, source of base data 672, and IoT hubs 651 and 652. As shown, provisioning service 611 includes routing rules 691. Also, IoT hub 651 includes device registry 693 and IoT hub 652 includes device registry 694.

[0072]In some examples, a private key (from the private/public key pair) is stored in IoT device 641 in secure storage on IoT device 641 (via Trusted Platform Module or other similar technology). In the example illustrated in FIG. 6, this is the root of trust of IoT device 641.

[0073]Also, in some examples, IoT device 641 stores an X509 certificate containing the URI of provisioning service 611. In some examples, the X509 certificate also contains the device ID of IoT device 641 and other device metadata for IoT device 641.

[0074]The signature of the X509 certificate is accomplished using the private key in a secure process. In some examples, the X509 certificate is generated at provisioning time, and in other examples, the X509 certificate is generated at manufacture time. The public key (from the private/public key pair) is made available to the provisioning service to validate the certificate signature.

[0075]At step 6-1, IoT device 641 contacts the endpoint of provisioning service 611, where the endpoint is set at the factory. The end point is extracted from the X509 certificate, and the signature along with the X509 certificate are passed as part of the call.

[0076]At step 6-2, provisioning service 611 ascertains the validating of the X509 certificate by calculating the signature using the public key and comparing with the supplied signature. Provisioning service device 611 also looks up the IoT device 641 in the source of base data to find out metadata/hub data about the IoT device 641.

[0077]At step 6-3, provisioning service 611 runs routing rules over the data from IoT device 641 as well as data from the base data source to find the right IoT hub to register IoT device 641 with.

[0078] At step 6-4, provisioning service 611 registers IoT device 641 with the device identity registry and device management (DM) registry of IoT hub, or instead to a single device registry in some examples.

[0079] At step 6-5, IoT hub 651 returns cryptographic information about IoT device 641 to provisioning service 611.

[0080] At step 6-6, provisioning service 611 returns the cryptographic information to IoT device 641. All subsequent calls are between IoT device 641 and IoT hub 651. IoT device 641 can now start sending data directly to IoT hub 651.

[0081] At step 6-7, IoT device 641 can now send data to IoT hub 651.

[0082] At step 6-8, the metadata of IoT device 641 syncs with the metadata stored in the DM registry of IoT hub 651 via DM syncing.

[0083] FIG. 7 is a logical flow diagram illustrating an example of a process (760) for IoT communications. After a start block, the process proceeds to block 761, where, upon initial boot, identification information is automatically sent to a provisioning service endpoint stored in the IoT device. The identification information includes an identification (ID) of the first IoT device. The process then moves to block 762. At block 762, cryptographic information is received from the provisioning service. The cryptographic information is associated with an IoT hub selected from a plurality of IoT hubs based, in part, on the ID of the first IoT device.

[0084] The process then advances to block 763. At block 763, a message is automatically sent to the IoT hub in response to receiving the cryptographic information. The process the proceeds to block 764. At block 764, a new configuration file and a firmware update are received from the IoT hub without requiring a user association. The process the moves to block 765. At block 765, the new configuration file and the firmware update are automatically installed. The process then advances to a return block, where other processing is resumed.

[0085] FIG. 8 is a logical flow diagram illustrating an example of a process (880) for IoT communications. After a start block, the process proceeds to block 881. At block 881, an identification message is received from a first IoT device. The identification message includes information associated with identification of a first IoT device. The process then moves to block 882. At block 882, at least one process is employed to validate the first IoT device. In some examples, validating of the first IoT device includes checking the device ID against an enrollment list. In some examples, validating the first IoT device includes validating a certificate in the identification information.

[0086]The process then moves to block 883. At block 883, based at least in part on ID of the first IoT device, an IoT hub is selected from a plurality of IoT hubs. That is, a determination of an IoT hub from a plurality of IoT hubs to be associated with the first IoT device is made based at least in part on the ID of the first IoT device. The process then advances to block 884. At block 884, a request to register the IoT device with the selected IoT hub is automatically sent to the selected IoT hub. The process then proceeds to block 885. At block 885, connection information is received from the selected IoT hub. The connection information includes configuration information. The process then moves to block 886. At block 886, upon receiving the connection information, the configuration information is automatically sent to the first IoT device without requiring a user association. The process then proceeds to a return block, where other processing is resumed.

[0087]FIG. 9 is a logical flow diagram illustrating an example of a process (990) for IoT communications. After a start block, the process proceeds to block 991. At block 991, a device registry is created. The process then moves to block 992. At block 992, a request is received to register a first IoT device from a provisioning service. In some examples, the request to register the first IoT device is based on network communications between the provisioning service and the IoT hub, such that a hostname of the IoT hub is a first hostname, a hostname of the provisioning service is a second hostname, and such that the second hostname is different from the first hostname. The process then advances to block 993. At block 993, the first IoT device may be added to the device registry. The process then proceeds to block 994. At block 994, cryptographic information associated with the first IoT device is transmitted.

[0088]The process then moves to block 995. At block 995, a first message is received from the IoT device in response to transmitting the cryptographic information. The process then advances to block 996. At block 996, an update is automatically sent. That is, in some examples, a new configuration file and a new firmware update is automatically sent to the first IoT device in response to receiving the first message. The process then moves to a return block, where other processing is resumed.

Conclusion

[0089]While the above Detailed Description describes certain examples of the technology, and describes the best mode contemplated, no matter how detailed the above appears in text, the technology can be practiced in many ways. Details may vary in implementation, while still being encompassed by the technology described herein. As noted above, particular terminology used when describing certain features or aspects of the technology should not

be taken to imply that the terminology is being redefined herein to be restricted to any specific characteristics, features, or aspects with which that terminology is associated. In general, the terms used in the following claims should not be construed to limit the technology to the specific examples disclosed herein, unless the Detailed Description explicitly defines such terms. Accordingly, the actual scope of the technology encompasses not only the disclosed examples, but also all equivalent ways of practicing or implementing the technology.

We claim :

1. An apparatus for Internet of Things (IoT) communications, comprising:

a first IoT device, including:

a memory adapted to store run-time data for the first IoT device; and

at least one processor that is adapted to execute processor-executable code that, in response to execution, enables the first IoT device to perform actions, including:

responsive to an initial boot, automatically sending identification information to a provisioning service endpoint stored in the first IoT device, wherein the identification information includes an identification (ID) of the first IoT device;

receiving cryptographic information from the provisioning service endpoint, wherein the cryptographic information is associated with an IoT hub selected from a plurality of IoT hubs based, in part, on the ID of the first IoT device;

automatically sending a message to the IoT hub in response to receiving the cryptographic information;

receiving a new configuration file and a firmware update from the IoT hub without requiring a user association; and

automatically installing the new configuration file and the firmware update.

2. The apparatus of claim 1, wherein the identification information further includes geographical information associated with the first IoT device, and wherein the IoT hub is selected from the plurality of IoT hubs further based upon the geographical information associated with the first IoT device.

3. The apparatus of claim 1, wherein the cryptographic information includes connectivity information associated with the first IoT device connecting to the IoT hub.

4. The apparatus of claim 1, wherein the cryptographic information includes credentials, and wherein the cryptographic information further includes a hostname associated with the IoT hub.

5. The apparatus of claim 1, wherein the actions further include, via communication with the IoT hub, syncing metadata stored in the first IoT device with metadata stored in the IoT hub via device management syncing.

6. The apparatus of claim 1, wherein the identification information includes a certificate that includes the ID of the first IoT device.

7. The apparatus of claim 6, wherein the certificate further includes a Uniform Resource Indicator of the provisioning service endpoint.

8. A method for Internet of Things (IoT) communications, comprising:

responsive to an initial boot, automatically sending identification information to a provisioning service end-point stored in a first IoT device, wherein the identification information includes an identification (ID) of the first IoT device;

receiving cryptographic information from the provisioning service endpoint, wherein the cryptographic information is associated with an IoT hub selected from a plurality of IoT hubs based, in part, on the ID of the first IoT device;

automatically sending a message to the IoT hub in response to receiving the cryptographic information;

receiving a new configuration file and a firmware update from the IoT hub without requiring a user association; and

automatically installing the new configuration file and the firmware update.

9. The method of claim 8, wherein the identification information further includes geographical information associated with the first IoT device, and wherein the IoT hub is selected from the plurality of IoT hubs further based upon the geographical information associated with the first IoT device.

10. The method of claim 8, wherein the cryptographic information includes connectivity information associated with the first IoT device connecting to the IoT hub.

11. The apparatus of claim 8, wherein the cryptographic information includes credentials, and wherein the cryptographic information further includes a hostname associated with the IoT hub.

12. The method of claim 8, further comprising :

via communications with the IoT hub, syncing metadata stored in the first IoT device with metadata stored in the IoT hub via device management syncing

13. The method of claim 8, wherein the identification information includes a certificate that includes the ID of the first IoT device.

14. The method of claim 13, wherein the certificate further includes a Uniform Resource Indicator of the provisioning service endpoint.

15. A processor-readable storage medium having instructions stored therein that, when executed by a first IoT device, cause the first IoT device to perform operations, the operations comprising :

responsive to an initial boot, automatically sending identification information to a provisioning service end-point stored in a first IoT device, wherein the identification information includes an identification (ID) of the first IoT device;

receiving cryptographic information from the provisioning service endpoint, wherein the cryptographic information is associated with an IoT hub selected from a plurality of IoT hubs based, in part, on the ID of the first IoT device;

automatically sending a message to the IoT hub in response to receiving the cryptographic information;

receiving a new configuration file and a firmware update from the IoT hub without requiring a user association; and

automatically installing the new configuration file and the firmware update.

16. The processor-readable storage medium of claim 15, wherein the identification information further includes geographical information associated with the first IoT device , and wherein the IoT hub is selected from the plurality of IoT hubs further based upon the geographical information associated with the first IoT device.

17. The processor-readable storage medium of claim 15, wherein the cryptographic information includes connectivity information associated with the first IoT device connecting to the IoT hub.

18. The processor-readable storage medium of claim 15, wherein the cryptographic information includes credentials, and wherein the cryptographic information further includes a hostname associated with the IoT hub.

19. The processor-readable storage medium of claim 15, wherein the operations further include :

via communication with the IoT hub, syncing metadata stored in the first IoT device with metadata stored in the IoT hub via device management syncing.

20. The processor-readable storage medium of claim 15, wherein the identification information includes a certificate that includes the ID of the first IoT device.