



(51) International Patent Classification:

H04L 29/06 (2006.01) *H04L 29/08* (2006.01)

(21) International Application Number:

PCT/US2017/069054

(22) International Filing Date:

29 December 2017 (29.12.2017)

(25) Filing Language:

English

(26) Publication Language:

English

(30) Priority Data:

62/503,783 09 May 2017 (09.05.2017) US

(72) Inventors; and

(71) Applicants: **SMITH, Ned M.** [US/US]; 375 S.W. Delta Drive, Beaverton, Oregon 97006 (US). **HELDT-SHELLER, Nathan** [US/US]; 4039 NE 75th Ave, Portland, Oregon 97213 (US).(74) Agent: **PERDOK, Monique, M.** et al.; Schwegman Lundberg & Woessner, P.A., P.O. Box 2938, Minneapolis, Minnesota 55402 (US).(81) Designated States (*unless otherwise indicated, for every kind of national protection available*): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, 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,

(54) Title: METHOD FOR IMPROVING OPERATIONAL INTEGRITY OF IOT DEVICE

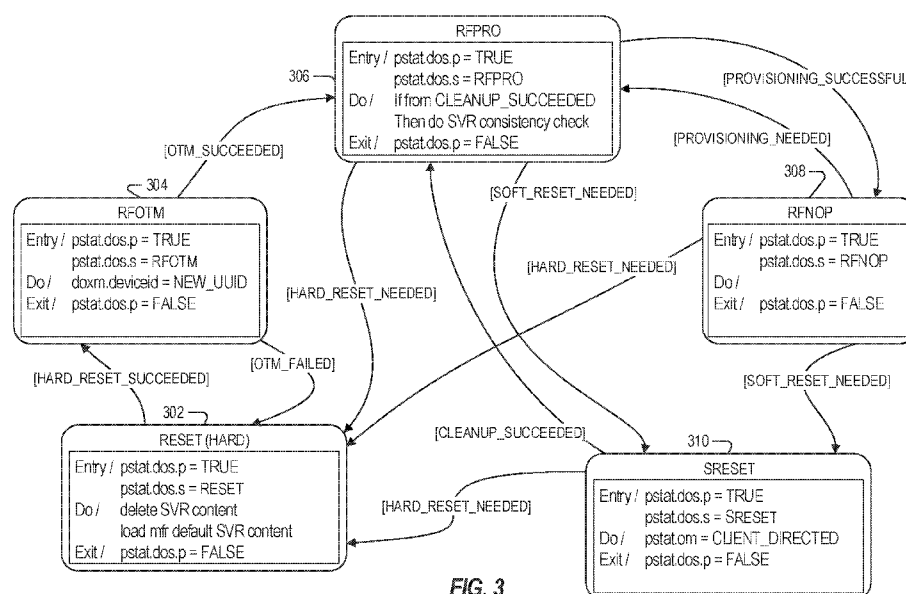


FIG. 3

(57) **Abstract:** Various systems and methods for implementing a soft reset state. A server device includes processing circuitry; and at least one storage device including instructions embodied thereon, wherein the instructions, which when executed by the processing circuitry, configure the processing circuitry to perform operations of a soft reset operation, the operations to: define a soft reset state; cause a check of a secure virtual resource (SVR) of the server device, while in the soft reset state; and transition from the soft reset state in response to an event.

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))*

METHOD FOR IMPROVING OPERATIONAL INTEGRITY OF IOT DEVICE

PRIORITY CLAIM

- 5 **[0001]** This application claims the benefit of priority to U.S. Application Serial No. 62/503,783, filed May 9, 2017, which is incorporated herein by reference in its entirety.

TECHNICAL FIELD

- 10 **[0002]** Embodiments described herein generally relate to processing techniques used with data communications and interconnected device networks, and in particular, to techniques applied within internet of things (IoT) devices and device networks.

BACKGROUND

- 15 **[0003]** IoT devices are physical or virtualized objects that may communicate on a network, and may include sensors, actuators, and other input/output components, such as to collect data or perform actions from a real world environment. For example, IoT devices may include low-powered devices that are embedded or attached to everyday things, such as buildings, vehicles,
20 packages, etc., to provide an additional level of artificial sensory perception of those things. Recently, IoT devices have become more popular and thus applications using these devices have proliferated.

- [0004]** Various standards have been proposed to more effectively interconnect and operate IoT devices and IoT network use cases. These include the
25 specialization of communication standards distributed by groups such as Institute of Electrical and Electronics Engineers (IEEE), and the specialization of application interaction architecture and configuration standards distributed by groups such as the Open Connectivity Foundation (OCF).

BRIEF DESCRIPTION OF THE DRAWINGS

- [0005] In the drawings, which are not necessarily drawn to scale, like numerals may describe similar components in different views. Like numerals having different letter suffixes may represent different instances of similar components. Some embodiments are illustrated by way of example, and not limitation, in the figures of the accompanying drawings in which:
- 5 [0006] **FIG. 1** illustrates an domain topology for respective internet-of-things (IoT) networks coupled through links to respective gateways, according to an example;
- [0007] **FIG. 2** illustrates a cloud computing network in communication with a mesh network of IoT devices operating as a fog device at the edge of the cloud computing network, according to an example;
- 10 [0008] **FIG. 3** illustrates a system state model with transition criteria, according to an embodiment;
- [0009] **FIG. 4** is a control and data flow sequencing diagram for a hard reset state, according to an embodiment;
- 15 [0010] **FIG. 5** is a control and data flow sequencing diagram for transitioning from a Ready-for-Owner-Transfer-Method (RFOTM) state to a Ready-for-Provisioning (RFPRO) state, according to an embodiment;
- [0011] **FIG. 6** is a control and data flow sequencing diagram for transitioning from a Ready-for-Provisioning (RFPRO) state to a Ready-for-Normal-Operation (RFNOP) state, according to an embodiment;
- 20 [0012] **FIG. 7** is a control and data flow sequencing diagram for transitioning from a Ready-for-Normal-Operation (RFNOP) state to a Ready-for-Provisioning (RFPRO) state, according to an embodiment;
- [0013] **FIG. 8** is a control and data flow sequencing diagram for transitioning from any state to a Soft Reset (SRESET) state, according to an embodiment;
- [0014] **FIG. 9** is a diagram illustrating the framework used for an apparatus to perform state manipulation in a hardware isolated execution environment, according to an embodiment;
- 30 [0015] **FIG. 10** is a flowchart illustrating a method for transitioning to a soft reset state, according to an embodiment;

[0016] FIG. 11 is a flowchart illustrating a method for transitioning to and operating in a soft reset state, according to an embodiment;

[0017] FIG. 12 illustrates a block diagram of a network illustrating communications among a number of IoT devices, according to an example; and

5 [0018] FIG. 13 illustrates a block diagram for an example IoT processing system architecture upon which any one or more of the techniques (e.g., operations, processes, methods, and methodologies) discussed herein may be performed, according to an example.

DETAILED DESCRIPTION

10 [0019] In the following description, methods, configurations, and related apparatuses are disclosed for the processing of security contexts in an IoT device interconnection setting through the use of a state model that is part of a device's object model.

[0020] The techniques discussed herein include a model that has five states, though operation with fewer or additional states may be applied safely. The five states include: a Reset state (also referred to as a "full reset" or "hard reset" state); a Ready-For-Onboarding (RFO) state (also referred to as a Ready-For-Ownership-Transfer-Method (RFOTM) state); a Ready-For-Provisioning (RFP) state; a Ready-For-Normal-Operation (RFNOP) state, and a Soft Reset (SRESET) state.

[0021] The Reset state is used for transitioning the device through device onboarding, which may involve limiting discoverability and where device function may be severely restricted.

25 [0022] The Ready-For-Onboarding state is used for transitioning the device into a deployment environment where trust, security and safety requirements associated with an operational environment can be assessed in the context of device's reliability, safety, security and other characteristics.

[0023] The Ready-For-Provisioning state is used for transitioning the device into an operational mode where access to application resources and normal operations is severely restricted, but where access to deployment, security and failsafe operation can be configured and adjusted.

[0024] The Ready-For-Normal-Operation state is used for transitioning to normal operations where interactions involving peer devices may be permitted.

5 [0025] The Soft Reset state is used for transitioning to a reset state where a device owner can fully re-configure and re-deploy the device without losing ownership context (that is the expected consequence of a full Reset). Soft reset allows remote and “over-the-wire” re-configuration which avoids costly manual interactions.

[0026] IoT devices often bridge IT and OT worlds. Operational integrity failure may result in greater harm than IT-only scenarios where loss of life, 10 physical harms, and property or environmental damage may result.

[0027] Many IoT applications are designed to manage state transitions where safe and unsafe state definition are part of the application design philosophy and tooling. However, these states rely on integral operation of the IoT device within a device lifecycle. Lifecycle failures may translate to operational failures.

15 [0028] The systems and methods described in this document define a device lifecycle state model that, as a feature of an IoT framework (such as OCF) may be followed by a large body of devices all expecting to interact with each other in the context of well-behaved device states.

[0029] Existing solutions make simplifying assumptions about the state of the 20 device and deployment. Typically, a vendor supplies a proprietary deployment and management infrastructure that is trusted to manage and keep track of device state. These may be thought of as centralized device state management systems. An application or operating system has limited scope when it comes to understanding device lifecycle status.

25 [0030] A challenge facing proprietary, centralized approaches is the device is not self-aware of what constitutes a safe device lifecycle state. Centrally managed lifecycle means the device may not be aware of an expected device state when the connection to the central entity is broken. Proprietary solutions also suffer when devices exist in heterogeneous deployments; each system is 30 different and devices cannot agree on whether a peer device is in a meaningful device state.

[0031] Using the five-state model described herein, IoT devices are self-aware of their device state making it possible for devices to be pro-active about

obtaining help necessary for transitioning to the next state. Devices have the appearance of being ‘smarter’ because they may facilitate recovery from failure or malicious events. In addition, IoT devices are resistant to catastrophic failure because device state ensures operationally device behaviors are contained within the device states. Special hardware may be helpful in ensuring device states are not easily overridden.

[0032] Other IoT devices and support services may introspect and assess devices according to device state awareness. This may be used by network management infrastructure to create network enclaves for devices different states (where devices autonomously assign themselves to an appropriate enclave). Automation reduces administrative cost and overhead while maintaining good operational integrity and security. Further details are provided in the following figures and associated discussion.

[0033] FIG. 1 illustrates an example domain topology for respective internet-of-things (IoT) networks coupled through links to respective gateways. The internet of things (IoT) is a concept in which a large number of computing devices are interconnected to each other and to the Internet to provide functionality and data acquisition at very low levels. Thus, as used herein, an IoT device may include a semiautonomous device performing a function, such as sensing or control, among others, in communication with other IoT devices and a wider network, such as the Internet.

[0034] Often, IoT devices are limited in memory, size, or functionality, allowing larger numbers to be deployed for a similar cost to smaller numbers of larger devices. However, an IoT device may be a smart phone, laptop, tablet, or PC, or other larger device. Further, an IoT device may be a virtual device, such as an application on a smart phone or other computing device. IoT devices may include IoT gateways, used to couple IoT devices to other IoT devices and to cloud applications, for data storage, process control, and the like.

[0035] Networks of IoT devices may include commercial and home automation devices, such as water distribution systems, electric power distribution systems, pipeline control systems, plant control systems, light switches, thermostats, locks, cameras, alarms, motion sensors, and the like. The

IoT devices may be accessible through remote computers, servers, and other systems, for example, to control systems or access data.

[0036] The future growth of the Internet and like networks may involve very large numbers of IoT devices. Accordingly, in the context of the techniques discussed herein, a number of innovations for such future networking will address the need for all these layers to grow unhindered, to discover and make accessible connected resources, and to support the ability to hide and compartmentalize connected resources. Any number of network protocols and communications standards may be used, wherein each protocol and standard is designed to address specific objectives. Further, the protocols are part of the fabric supporting human accessible services that operate regardless of location, time or space. The innovations include service delivery and associated infrastructure, such as hardware and software; security enhancements; and the provision of services based on Quality of Service (QoS) terms specified in service level and service delivery agreements. As will be understood, the use of IoT devices and networks, such as those introduced in **FIGS. 1** and **2**, present a number of new challenges in a heterogeneous network of connectivity comprising a combination of wired and wireless technologies.

[0037] **FIG. 1** specifically provides a simplified drawing of a domain topology that may be used for a number of internet-of-things (IoT) networks comprising IoT devices 104, with the IoT networks 156, 158, 160, 162, coupled through backbone links 102 to respective gateways 154. For example, a number of IoT devices 104 may communicate with a gateway 154, and with each other through the gateway 154. To simplify the drawing, not every IoT device 104, or communications link (e.g., link 116, 122, 128, or 132) is labeled. The backbone links 102 may include any number of wired or wireless technologies, including optical networks, and may be part of a local area network (LAN), a wide area network (WAN), or the Internet. Additionally, such communication links facilitate optical signal paths among both IoT devices 104 and gateways 154, including the use of MUXing/deMUXing components that facilitate interconnection of the various devices.

[0038] The network topology may include any number of types of IoT networks, such as a mesh network provided with the network 156 using

Bluetooth low energy (BLE) links 122. Other types of IoT networks that may be present include a wireless local area network (WLAN) network 158 used to communicate with IoT devices 104 through IEEE 802.11 (Wi-Fi®) links 128, a cellular network 160 used to communicate with IoT devices 104 through an
5 LTE/LTE-A (4G) or 5G cellular network, and a low-power wide area (LPWA) network 162, for example, a LPWA network compatible with the LoRaWan specification promulgated by the LoRa alliance, or a IPv6 over Low Power Wide-Area Networks (LPWAN) network compatible with a specification promulgated by the Internet Engineering Task Force (IETF). Further, the
10 respective IoT networks may communicate with an outside network provider (e.g., a tier 2 or tier 3 provider) using any number of communications links, such as an LTE cellular link, an LPWA link, or a link based on the IEEE 802.15.4 standard, such as Zigbee®. The respective IoT networks may also operate with use of a variety of network and internet application protocols such as
15 Constrained Application Protocol (CoAP). The respective IoT networks may also be integrated with coordinator devices that provide a chain of links that forms cluster tree of linked devices and networks.

[0039] Each of these IoT networks may provide opportunities for new technical features, such as those as described herein. The improved technologies
20 and networks may enable the exponential growth of devices and networks, including the use of IoT networks into as fog devices or systems. As the use of such improved technologies grows, the IoT networks may be developed for self-management, functional evolution, and collaboration, without needing direct human intervention. The improved technologies may even enable IoT networks
25 to function without centralized controlled systems. Accordingly, the improved technologies described herein may be used to automate and enhance network management and operation functions far beyond current implementations.

[0040] In an example, communications between IoT devices 104, such as over the backbone links 102, may be protected by a decentralized system for
30 authentication, authorization, and accounting (AAA). In a decentralized AAA system, distributed payment, credit, audit, authorization, and authentication systems may be implemented across interconnected heterogeneous network infrastructure. This allows systems and networks to move towards autonomous

operations. In these types of autonomous operations, machines may even contract for human resources and negotiate partnerships with other machine networks. This may allow the achievement of mutual objectives and balanced service delivery against outlined, planned service level agreements as well as

5 achieve solutions that provide metering, measurements, traceability, and trackability. The creation of new supply chain structures and methods may enable a multitude of services to be created, mined for value, and collapsed without any human involvement.

[0041] Such IoT networks may be further enhanced by the integration of

10 sensing technologies, such as sound, light, electronic traffic, facial and pattern recognition, smell, vibration, into the autonomous organizations among the IoT devices. The integration of sensory systems may allow systematic and autonomous communication and coordination of service delivery against contractual service objectives, orchestration and quality of service (QoS) based

15 swarming and fusion of resources. Some of the individual examples of network-based resource processing include the following.

[0042] The mesh network 156, for instance, may be enhanced by systems that perform inline data-to-information transforms. For example, self-forming chains of processing resources comprising a multi-link network may distribute the

20 transformation of raw data to information in an efficient manner, and the ability to differentiate between assets and resources and the associated management of each. Furthermore, the proper components of infrastructure and resource based trust and service indices may be inserted to improve the data integrity, quality, assurance and deliver a metric of data confidence.

25 [0043] The WLAN network 158, for instance, may use systems that perform standards conversion to provide multi-standard connectivity, enabling IoT devices 104 using different protocols to communicate. Further systems may provide seamless interconnectivity across a multi-standard infrastructure comprising visible Internet resources and hidden Internet resources.

30 [0044] Communications in the cellular network 160, for instance, may be enhanced by systems that offload data, extend communications to more remote devices, or both. The LPWA network 162 may include systems that perform non-Internet protocol (IP) to IP interconnections, addressing, and routing.

Further, each of the IoT devices 104 may include the appropriate transceiver for wide area communications with that device. Further, each IoT device 104 may include other transceivers for communications using additional protocols and frequencies. This is discussed further with respect to the communication environment and hardware of an IoT processing device depicted in **FIGS. 12** and **13**.

[0045] Finally, clusters of IoT devices may be equipped to communicate with other IoT devices as well as with a cloud network. This may allow the IoT devices to form an ad-hoc network between the devices, allowing them to function as a single device, which may be termed a fog device. This configuration is discussed further with respect to **FIG. 2**.

[0046] **FIG. 2** illustrates a cloud computing network in communication with a mesh network of IoT devices (devices 202) operating as a fog device at the edge of the cloud computing network. The mesh network of IoT devices may be termed a fog 220, operating at the edge of the cloud 200. To simplify the diagram, not every IoT device 202 is labeled.

[0047] The fog 220 may be considered to be a massively interconnected network wherein a number of IoT devices 202 are in communications with each other, for example, by radio links 222. As an example, this interconnected network may be facilitated using an interconnect specification released by the Open Connectivity Foundation™ (OCF). This standard allows devices to discover each other and establish communications for interconnects. Other interconnection protocols may also be used, including, for example, the optimized link state routing (OLSR) Protocol, the better approach to mobile ad-hoc networking (B.A.T.M.A.N.) routing protocol, or the OMA Lightweight M2M (LWM2M) protocol, among others.

[0048] Three types of IoT devices 202 are shown in this example, gateways 204, data aggregators 226, and sensors 228, although any combinations of IoT devices 202 and functionality may be used. The gateways 204 may be edge devices that provide communications between the cloud 200 and the fog 220, and may also provide the backend process function for data obtained from sensors 228, such as motion data, flow data, temperature data, and the like. The data aggregators 226 may collect data from any number of the sensors 228, and

perform the back end processing function for the analysis. The results, raw data, or both may be passed along to the cloud 200 through the gateways 204. The sensors 228 may be full IoT devices 202, for example, capable of both collecting data and processing the data. In some cases, the sensors 228 may be more
5 limited in functionality, for example, collecting the data and allowing the data aggregators 226 or gateways 204 to process the data.

[0049] Communications from any IoT device 202 may be passed along a convenient path between any of the IoT devices 202 to reach the gateways 204. In these networks, the number of interconnections provide substantial
10 redundancy, allowing communications to be maintained, even with the loss of a number of IoT devices 202. Further, the use of a mesh network may allow IoT devices 202 that are very low power or located at a distance from infrastructure to be used, as the range to connect to another IoT device 202 may be much less than the range to connect to the gateways 204.

[0050] The fog 220 provided from these IoT devices 202 may be presented to devices in the cloud 200, such as a server 206, as a single device located at the edge of the cloud 200, e.g., a fog device. In this example, the alerts coming from the fog device may be sent without being identified as coming from a specific IoT device 202 within the fog 220. In this fashion, the fog 220 may be
15 considered a distributed platform that provides computing and storage resources to perform processing or data-intensive tasks such as data analytics, data aggregation, and machine-learning, among others.

[0051] In some examples, the IoT devices 202 may be configured using an imperative programming style, e.g., with each IoT device 202 having a specific
25 function and communication partners. However, the IoT devices 202 forming the fog device may be configured in a declarative programming style, allowing the IoT devices 202 to reconfigure their operations and communications, such as to determine needed resources in response to conditions, queries, and device failures. As an example, a query from a user located at a server 206 about the
30 operations of a subset of equipment monitored by the IoT devices 202 may result in the fog 220 device selecting the IoT devices 202, such as particular sensors 228, needed to answer the query. The data from these sensors 228 may then be aggregated and analyzed by any combination of the sensors 228, data

aggregators 226, or gateways 204, before being sent on by the fog 220 device to the server 206 to answer the query. In this example, IoT devices 202 in the fog 220 may select the sensors 228 used based on the query, such as adding data from flow sensors or temperature sensors. Further, if some of the IoT devices
5 202 are not operational, other IoT devices 202 in the fog 220 device may provide analogous data, if available.

[0052] Various techniques for improving operational integrity of an IoT device in a connected device framework are described in the following sections with reference to **FIGS. 3 to 11**. Specifically, some of the device framework and
10 terminology discussed in the following examples is provided within specifications defined by the OCF family of standards (e.g., the OCF Specification 1.0 and later versions). However, it will be understood that the following techniques may be applicable to other specifications and families of standards, in addition to proprietary and non-standard implementations.

[0053] In an OCF architecture, entities in the real physical world (e.g., a temperature sensor) are represented as resources. Interactions with entities are implemented through resource representations, which use operations that adhere to Representational State Transfer (REST) architectures, e.g., RESTful interactions. As such, entities are exposed as resources, each with their unique
20 identifiers (URIs) and support interfaces that enable RESTful operations on their resources. A client initiates a RESTful operation on a server. The client is the initiator and the server is a responder. Any device may act as a client to initiate a RESTful operation or any other device acting as a server. Thus, the role of a device as a client or server, in many circumstances, may be interchangeable. Any
25 device that exposes a resource is by definition, a server. Each RESTful operation contains all of the information needed to understand the context of the operation and is supported by a set of generic operations (e.g., CREATE, RETRIEVE, UPDATE, DELETE, and NOTIFY (CRUDN)).

[0054] As discussed herein, the following techniques may be implemented in
30 connection with use of various OCF services, including DOXS (also known as DOTS, Device Owner Transfer Service). In a further example, the following techniques may be implemented in connection with an onboarding tool (OBT). In the context of an OCF implementation, an OBT is an logical entity within a

specific IoT network that establishes ownership for a specific device and helps bring the device into operational state within that network. For instance, a typical OBT may implement DOXS, AMS (Access Management Service), and CMS (Credential Management Service) functionality.

5 [0055] FIG. 3 illustrates a system state model 300 with transition criteria, according to an embodiment. The system state model 300 includes five states: a RESET state 302; a Ready-For-Onboarding (also referred to as Ready for Ownership Transfer (RFOTM)) state 304; a Ready-For-Provisioning (RFPRO) state 306; a Ready-For-Normal-Operation (RFNOP) state 308, and a Soft Reset (SRESET) state 310.

[0056] Each state 302-310 includes an “entry” procedure, a “do” procedure, and an “exit” procedure. The entry procedure is performed upon entry into the particular state. The do procedure is performed when the state is active. The exit procedure is performed when control is being transitioned to another state.

15 [0057] A device state transition controller may be used to perform some or all of the procedures in a state, control transition between states, and the like. The device state transition controller may be a trusted execution environment (TEE) or other isolated execution environment (e.g., Intel SGX, Intel TrustLite, ARM TrustZone, etc.).

20 [0058] The device includes the following resources (e.g., security virtual resources):

[0059] 1) /oic/sec/doxm (device owner transfer method) resource

[0060] 2) /oic/sec/pstat (provisioning status) resource

[0061] 3) /oic/sec/cred (credential) resource

25 [0062] 4) /oic/sec/svc (security services) resource

[0063] Upon entering a hard RESET state 302, the properties for the provisioning status “pstat” resource are set, such that the device onboarding state’s pending state value is Boolean TRUE (pstat.dos.p = TRUE) and the device onboarding state’s onboarding state indicates a hard reset (pstat.dos.s = RESET). Once in RESET state 302, the security virtual resource (SVR) content of the device is deleted and the manufacturer’s default SVR is loaded. When the reset is completed successfully, control passes to RFOTM state 304. Upon state

transition to RFOTM state 304, the device onboarding state's pending state is set to Boolean FALSE (pstat.dos.p = FALSE).

[0064] Upon entering RFOTM state 304, the properties for the provisioning status "pstat" resource are set, such that the device onboarding state's pending state value is Boolean TRUE (pstat.dos.p = TRUE) and the device onboarding state's onboarding state indicates a hard reset (pstat.dos.s = RFOTM). Once in RFOTM state 304, a new UUID is assigned to the doxm resource's deviceid property (doxm.deviceid = NEW_UUID). If the ownership transfer method (OTM) fails, then the control is transferred back to the RESET state 302 and the device onboarding state's pending state is set to Boolean FALSE (pstat.dos.p = FALSE). If the OTM succeeds, then control is transferred to RFPRO state 306 and the device onboarding state's pending state is set to Boolean FALSE (pstat.dos.p = FALSE).

[0065] Upon entering RFPRO state 306, the properties for the provisioning status "pstat" resource are set, such that the device onboarding state's pending state value is Boolean TRUE (pstat.dos.p = TRUE) and the device onboarding state's onboarding state indicates a hard reset (pstat.dos.s = RFPRO). Once in RFPRO state 306, a consistency check is performed on the SVR if the control to RFPRO state 306 was from SRESET state 310. When control is in the RFPRO state 306, if a soft reset is needed, then control transfers to SRESET state 310 for a soft reset. If a hard reset is needed, the control transfers to RESET state 302 for a hard reset. Otherwise, when provisioning succeeds, the control passes to RFNOP state 308. The device onboarding state's pending state is set to Boolean FALSE (pstat.dos.p = FALSE) upon exiting the RFPRO state 306.

[0066] Upon entering RFNOP state 308, the properties for the provisioning status "pstat" resource are set, such that the device onboarding state's pending state value is Boolean TRUE (pstat.dos.p = TRUE) and the device onboarding state's onboarding state indicates a hard reset (pstat.dos.s = RFNOP). Once in RFNOP state 308, normal operations may commence and the resource may be used by subjects according to security controls (e.g., an ACL). When control is in the RFNOP state 308, if a soft reset is needed, then control transfers to SRESET state 310 for a soft reset. If a hard reset is needed, the control transfers to RESET state 302 for a hard reset. Otherwise, the control passes to RFPRO

state 306 when the resource needs provisioning. The device onboarding state's pending state is set to Boolean FALSE (pstat.dos.p = FALSE) upon exiting the RFNOP state 308.

[0067] Upon entering SRESET state 310, the properties for the "pstat"

5 resource are set, such that the device onboarding state's pending state value is Boolean TRUE (pstat.dos.p = TRUE) and the device onboarding state's onboarding state indicates a hard reset (pstat.dos.s = SRESET). Once in SRESET state 310, the operational mode is set to be client directed (pstat.om = CLIENT_DIRECTED). If a hard reset is needed, the control transfers to RESET
10 state 302 for a hard reset. Otherwise, after the soft reset, the device is ready for provisioning and control passes to RFPRO state 306 where a consistency check is performed. The device onboarding state's pending state is set to Boolean FALSE (pstat.dos.p = FALSE) upon exiting the SRESET state 310.

[0068] FIG. 4 is a control and data flow sequencing diagram for a hard reset
15 state, according to an embodiment. OCF resources may be affected in the following ways.

[0069] Upon transition into RESET state 302:

- a. Server sets /doxm.owned = FALSE
- b. Server sets /doxm.devowneruuid = "00000000-0000-0000-0000-
20 000000000000"
- c. Server sets /doxm.deviceuuid = <a random non-repeating UUID>
- d. Server sets /doxm.oxmsel = 4 (oic.sec.oxm.self)
- e. Server sets /doxm.rowneruuid = "00000000-0000-0000-0000-
000000000000"
- 25 f. Server sets /pstat.isop = FALSE
- g. Server sets /pstat.cm = 00000001
- h. Server sets /pstat.tm = 00000010
- i. Server sets /pstat.om = <mfg_def>
- j. Server sets /pstat.rowneruuid = "00000000-0000-0000-0000-
30 000000000000"
- k. Server sets /acl.rowneruuid = "00000000-0000-0000-0000-
000000000000"

- l. Server sets /acl2.rowneruuid = "00000000-0000-0000-0000-000000000000"
 - m. Server sets /cred.rowneruuid = "00000000-0000-0000-0000-000000000000"
- 5 **[0070]** In the transition to the RFOTM state 304, all values are the same except for the following:
 - a. Server sets /pstat.dos.s = RFOTM
 - b. Server sets /pstat.cm = XXXXXX10
 - c. Server sets /pstat.tm = XXXXXX00
- 10 **[0071]** **FIG. 5** is a control and data flow sequencing diagram for transitioning from RFOTM state 304 to RFPRO state 306, according to an embodiment. OCF resources may be affected in the following ways. All values remain as they were when transitioning into RFOTM except as follows:
 - a. DOXS sets /doxm.owned = TRUE
 - 15 b. DOXS sets /doxm.devowneruuid = <DOXS_UUID>
 - c. DOXS sets /doxm.deviceuuid = <NEW_UUID> OR Server sets to a persistent UUID.
 - d. DOXS sets /doxm.oxmsel = <The selected OTM>
 - e. Server sets /pstat.cm = XXXXXX00
 - 20 f. DOXS sets /doxm.rowneruuid = <DOXS_UUID> (or other as determined by DOXS)
 - g. DOXS sets /pstat.rowneruuid = <DOXS_UUID> (or other as determined by DOXS)
 - h. DOXS sets /cred.rowneruuid = <CMS_UUID> (or other as determined by DOXS)
 - 25 i. DOXS sets /acl.rowneruuid = <AMS_UUID> (or other as determined by DOXS)
 - j. DOXS sets /acl2.rowneruuid = <AMS_UUID> (or other as determined by DOXS)
 - 30 k. DOXS sets /cred.creds = AMS and CMS credentials
 - l. DOXS sets /acl2.aces = AMS and CMS and other ACES as needed.
 - m. DOXS sets /pstat.dos.s = RFPRO

[0072] FIG. 6 is a control and data flow sequencing diagram for transitioning from RFPRO state 306 to RFNOP state 308, according to an embodiment. OCF resources may be affected in the following ways:

- a. DOXS creates and/or initializes SVRs as needed
- 5 b. CMS provisions credentials into /cred.creds as needed
- c. AMS provisions ACEs into /acl, /acl2, /amacl and /sac1 as needed
- d. DOXS, CMS or AMS sets /pstat.dos.s = RFNOP
- e. Server sets /pstat.isop = TRUE

[0073] FIG. 7 is a control and data flow sequencing diagram for transitioning from RFNOP state 308 to RFPRO state 306, according to an embodiment. OCF resources may be affected in the following ways:

- a. Device's access vertical resources permitted
- b. DOXS, AMS or CMS sets /pstat.dos.s = RFPRO
- c. Server sets /pstat.isop = FALSE

[0074] FIG. 8 is a control and data flow sequencing diagram for transitioning from any state to SRESET state 310, according to an embodiment. OCF resources may be affected in the following ways:

- a. DOXS, AMS or CMS set /pstat.dos.s = SRESET OR Server sets /pstat.dos.s = SRESET
- 20 b. Server sets /pstat.isop = FALSE
- c. Vertical resources are inaccessible
- d. DOXS creates / initializes SVRs as needed
- e. DOXS sets /pstat.dos.s = RFPRO OR RESET OR Server sets /pstat.dos.s = RESET (after DOXS connection failure)

[0075] FIG. 9 is a diagram illustrating the framework used for an apparatus to perform state manipulation in a hardware isolated execution environment, according to an embodiment. The framework 900 includes a hardware isolated execution environment layer 902 with an operating system layer 904 stacked on top of it, and an application layer 906 on top of the operating system layer 904.

[0076] The hardware isolated execution environment layer 902 may be implemented with Intel SGX, Intel ME, Intel TrustLite, ARM TrustZone, and the like. The hardware isolated execution environment 902 includes a device state transition controller 908. The operating system layer 904 includes IoT

framework drivers 910. The application layer 906 includes IoT applications and IoT framework interfaces 912.

[0077] FIG. 10 is a flowchart illustrating a method 1000 for transitioning to a soft reset state, according to an embodiment. At block 1002, a server detects that SVRs are corrupted. The pstat.dos.s value is set to "SRESET" to indicate a soft reset state (operation 1004). The server attempts to establish a secure connection with a client with DOXS. (operation 1006). If the secure connection is established, then the pstat.dos.s value is set to "RFPRO" and the server may transition to the ready-for-provisioning state and bypass a hard reset (operation 1008). If instead the SVRs are too corrupted and the secure connection cannot be established, then the pstat.dos.s value is set to "RESET" and the server transitions to a hard reset state (operation 1010).

[0078] FIG. 11 is a flowchart illustrating a method 1100 of transitioning to and operating in a soft reset state. The following operations are depicted from the perspective of an OCF device acting as a server. However, it will be understood that the techniques of flowchart 1100 may be further adapted from the perspective of a client-side device, an intermediary device, or systems involving both client, server, and intermediate entities.

[0079] The method 1100 may be implemented by a server device. At 1110, a soft reset state is defined. In an embodiment, to define the soft reset state a condition is detected and a parameter value is set based on the condition. In a further embodiment, the condition is that a secure virtual resource (SVR) of the server device is corrupted.

[0080] In an embodiment, to define the soft reset state, a parameter value of the server device is set by a client service. In a further embodiment, the client service implements an onboarding tool. In a related embodiment, the client service owns a resource provided by the server device.

[0081] At 1120, a check of a secure virtual resource (SVR) of the server device is caused to occur while in the soft reset state. In an embodiment, causing the check of the SVR includes checking the SVR at the server device during a state after the soft reset state. In a further embodiment, the state after the soft reset state is a ready for provisioning state.

[0082] In an embodiment, causing the check of the SVR includes checking the SVR at a client service during the soft reset state.

[0083] At 1130, the server device is transitioned from the soft reset state in response to an event. In an embodiment, the event is a result of a resource consistency check, and the transitioning from the soft reset state in response to the event includes transmitting resources to a client service for the resource consistency check and transitioning to a reset state or ready for provisioning state based on the resource consistency check.

[0084] In an embodiment, the event is a result of a credential check, and the transitioning from the soft reset state in response to the event includes performing the credential check on a credential of an owner of the device and transitioning to a reset state when the credential check fails. In a further embodiment, the credential check fails when the credential cannot be found or is determined to be corrupted.

[0085] In an embodiment, the method 1100 includes operating using client-directed instructions, while in the soft reset state.

[0086] In other examples, the operations and functionality described above with reference to **FIGS. 3 to 11** may be embodied by a IoT device machine in the example form of an electronic processing system, within which a set or sequence of instructions may be executed to cause the electronic processing system to perform any one of the methodologies discussed herein, according to an example embodiment. The machine may be an IoT device or an IoT gateway, including a machine embodied by aspects of a personal computer (PC), a tablet PC, a personal digital assistant (PDA), a mobile telephone or smartphone, or any machine capable of executing instructions (sequential or otherwise) that specify actions to be taken by that machine. Further, while only a single machine may be depicted and referenced in the example above, such machine shall also be taken to include any collection of machines that individually or jointly execute a set (or multiple sets) of instructions to perform any one or more of the methodologies discussed herein. Further, these and like examples to a processor-based system shall be taken to include any set of one or more machines that are controlled by or operated by a processor (e.g., a computer) to individually or jointly execute instructions to perform any one or more of the methodologies discussed herein.

[0087] FIG. 12 illustrates a drawing of a cloud computing network, or cloud 1200, in communication with a number of Internet of Things (IoT) devices. The cloud 1200 may represent the Internet, or may be a local area network (LAN), or a wide area network (WAN), such as a proprietary network for a company. The IoT devices may include any number of different types of devices, grouped in various combinations. For example, a traffic control group 1206 may include IoT devices along streets in a city. These IoT devices may include stoplights, traffic flow monitors, cameras, weather sensors, and the like. The traffic control group 1206, or other subgroups, may be in communication with the cloud 1200 through wired or wireless links 1208, such as LPWA links, optical links, and the like. Further, a wired or wireless sub-network 1212 may allow the IoT devices to communicate with each other, such as through a local area network, a wireless local area network, and the like. The IoT devices may use another device, such as a gateway 1210 or 1228 to communicate with remote locations such as the cloud 1200; the IoT devices may also use one or more servers 1230 to facilitate communication with the cloud 1200 or with the gateway 1210. For example, the one or more servers 1230 may operate as an intermediate network node to support a local edge cloud or fog implementation among a local area network. Further, the gateway 1228 that is depicted may operate in a cloud-to-gateway-to-many edge devices configuration, such as with the various IoT devices 1214, 1220, 1224 being constrained or dynamic to an assignment and use of resources in the cloud 1200.

[0088] Other example groups of IoT devices may include remote weather stations 1214, local information terminals 1216, alarm systems 1218, automated teller machines 1220, alarm panels 1222, or moving vehicles, such as emergency vehicles 1224 or other vehicles 1226, among many others. Each of these IoT devices may be in communication with other IoT devices, with servers 1204, with another IoT fog device or system (not shown, but depicted in FIG. 2), or a combination therein. The groups of IoT devices may be deployed in various residential, commercial, and industrial settings (including in both private or public environments).

[0089] As may be seen from FIG. 12, a large number of IoT devices may be communicating through the cloud 1200. This may allow different IoT devices to

request or provide information to other devices autonomously. For example, a group of IoT devices (e.g., the traffic control group 1206) may request a current weather forecast from a group of remote weather stations 1214, which may provide the forecast without human intervention. Further, an emergency vehicle 1224 may be alerted by an automated teller machine 1220 that a burglary is in progress. As the emergency vehicle 1224 proceeds towards the automated teller machine 1220, it may access the traffic control group 1206 to request clearance to the location, for example, by lights turning red to block cross traffic at an intersection in sufficient time for the emergency vehicle 1224 to have unimpeded access to the intersection.

[0090] Clusters of IoT devices, such as the remote weather stations 1214 or the traffic control group 1206, may be equipped to communicate with other IoT devices as well as with the cloud 1200. This may allow the IoT devices to form an ad-hoc network between the devices, allowing them to function as a single device, which may be termed a fog device or system (e.g., as described above with reference to **FIG. 2**).

[0091] **FIG. 13** is a block diagram of an example of components that may be present in an IoT device 1350 for implementing the techniques described herein. The IoT device 1350 may include any combinations of the components shown in the example or referenced in the disclosure above. The components may be implemented as ICs, portions thereof, discrete electronic devices, or other modules, logic, hardware, software, firmware, or a combination thereof adapted in the IoT device 1350, or as components otherwise incorporated within a chassis of a larger system. Additionally, the block diagram of **FIG. 13** is intended to depict a high-level view of components of the IoT device 1350. However, some of the components shown may be omitted, additional components may be present, and different arrangement of the components shown may occur in other implementations.

[0092] The IoT device 1350 may include a processor 1352, which may be a microprocessor, a multi-core processor, a multithreaded processor, an ultra-low voltage processor, an embedded processor, or other known processing element. The processor 1352 may be a part of a system on a chip (SoC) in which the processor 1352 and other components are formed into a single integrated circuit,

or a single package, such as the Edison™ or Galileo™ SoC boards from Intel. As an example, the processor 1352 may include an Intel® Architecture Core™ based processor, such as a Quark™, an Atom™, an i3, an i5, an i7, or an MCU-class processor, or another such processor available from Intel® Corporation, Santa Clara, CA. However, any number other processors may be used, such as available from Advanced Micro Devices, Inc. (AMD) of Sunnyvale, CA, a MIPS-based design from MIPS Technologies, Inc. of Sunnyvale, CA, an ARM-based design licensed from ARM Holdings, Ltd. or customer thereof, or their licensees or adopters. The processors may include units such as an A5-A10 processor from Apple® Inc., a Snapdragon™ processor from Qualcomm® Technologies, Inc., or an OMAP™ processor from Texas Instruments, Inc.

[0093] The processor 1352 may communicate with a system memory 1354 over an interconnect 1356 (e.g., a bus). Any number of memory devices may be used to provide for a given amount of system memory. As examples, the memory may be random access memory (RAM) in accordance with a Joint Electron Devices Engineering Council (JEDEC) design such as the DDR or mobile DDR standards (e.g., LPDDR, LPDDR2, LPDDR3, or LPDDR4). In various implementations the individual memory devices may be of any number of different package types such as single die package (SDP), dual die package (DDP) or quad die package (Q17P). These devices, in some examples, may be directly soldered onto a motherboard to provide a lower profile solution, while in other examples the devices are configured as one or more memory modules that in turn couple to the motherboard by a given connector. Any number of other memory implementations may be used, such as other types of memory modules, e.g., dual inline memory modules (DIMMs) of different varieties including but not limited to microDIMMs or MiniDIMMs.

[0094] To provide for persistent storage of information such as data, applications, operating systems and so forth, a storage 1358 may also couple to the processor 1352 via the interconnect 1356. In an example the storage 1358 may be implemented via a solid state disk drive (SSDD). Other devices that may be used for the storage 1358 include flash memory cards, such as SD cards, microSD cards, xD picture cards, and the like, and USB flash drives. In low power implementations, the storage 1358 may be on-die memory or registers

associated with the processor 1352. However, in some examples, the storage 1358 may be implemented using a micro hard disk drive (HDD). Further, any number of new technologies may be used for the storage 1358 in addition to, or instead of, the technologies described, such resistance change memories, phase change memories, holographic memories, or chemical memories, among others.

[0095] The components may communicate over the interconnect 1356. The interconnect 1356 may include any number of technologies, including industry standard architecture (ISA), extended ISA (EISA), peripheral component interconnect (PCI), peripheral component interconnect extended (PCIx), PCI express (PCIe), or any number of other technologies. The interconnect 1356 may be a proprietary bus, for example, used in a SoC based system. Other bus systems may be included, such as an I2C interface, an SPI interface, point to point interfaces, and a power bus, among others.

[0096] The interconnect 1356 may couple the processor 1352 to a mesh transceiver 1362, for communications with other mesh devices 1364. The mesh transceiver 1362 may use any number of frequencies and protocols, such as 2.4 Gigahertz (GHz) transmissions under the IEEE 802.15.4 standard, using the Bluetooth® low energy (BLE) standard, as defined by the Bluetooth® Special Interest Group, or the ZigBee® standard, among others. Any number of radios, configured for a particular wireless communication protocol, may be used for the connections to the mesh devices 1364. For example, a WLAN unit may be used to implement Wi-Fi™ communications in accordance with the Institute of Electrical and Electronics Engineers (IEEE) 802.11 standard. In addition, wireless wide area communications, e.g., according to a cellular or other wireless wide area protocol, may occur via a WWAN unit.

[0097] The mesh transceiver 1362 may communicate using multiple standards or radios for communications at different range. For example, the IoT device 1350 may communicate with close devices, e.g., within about 10 meters, using a local transceiver based on BLE, or another low power radio, to save power.

More distant mesh devices 1364, e.g., within about 50 meters, may be reached over ZigBee or other intermediate power radios. Both communications techniques may take place over a single radio at different power levels, or may

take place over separate transceivers, for example, a local transceiver using BLE and a separate mesh transceiver using ZigBee.

- [0098] A wireless network transceiver 1366 may be included to communicate with devices or services in the cloud 1300 via local or wide area network protocols. The wireless network transceiver 1366 may be a LPWA transceiver that follows the IEEE 802.15.4, or IEEE 802.15.4g standards, among others. The IoT device 1350 may communicate over a wide area using LoRaWAN™ (Long Range Wide Area Network) developed by Semtech and the LoRa Alliance. The techniques described herein are not limited to these technologies, but may be used with any number of other cloud transceivers that implement long range, low bandwidth communications, such as Sigfox, and other technologies. Further, other communications techniques, such as time-slotted channel hopping, described in the IEEE 802.15.4e specification may be used.
- [0099] Any number of other radio communications and protocols may be used in addition to the systems mentioned for the mesh transceiver 1362 and wireless network transceiver 1366, as described herein. For example, the radio transceivers 1362 and 1366 may include an LTE or other cellular transceiver that uses spread spectrum (SPA/SAS) communications for implementing high speed communications. Further, any number of other protocols may be used, such as Wi-Fi® networks for medium speed communications and provision of network communications.
- [0100] The radio transceivers 1362 and 1366 may include radios that are compatible with any number of 3GPP (Third Generation Partnership Project) specifications, notably Long Term Evolution (LTE), Long Term Evolution-Advanced (LTE-A), and Long Term Evolution-Advanced Pro (LTE-A Pro). It may be noted that radios compatible with any number of other fixed, mobile, or satellite communication technologies and standards may be selected. These may include, for example, any Cellular Wide Area radio communication technology, which may include e.g. a 5th Generation (5G) communication systems, a Global System for Mobile Communications (GSM) radio communication technology, a General Packet Radio Service (GPRS) radio communication technology, or an Enhanced Data Rates for GSM Evolution (EDGE) radio communication technology, a UMTS (Universal Mobile Telecommunications System)

communication technology, In addition to the standards listed above, any number of satellite uplink technologies may be used for the wireless network transceiver 1366, including, for example, radios compliant with standards issued by the ITU (International Telecommunication Union), or the ETSI (European Telecommunications Standards Institute), among others. The examples provided herein are thus understood as being applicable to various other communication technologies, both existing and not yet formulated.

[0101] A network interface controller (NIC) 1368 may be included to provide a wired communication to the cloud 1300 or to other devices, such as the mesh devices 1364. The wired communication may provide an Ethernet connection, or may be based on other types of networks, such as Controller Area Network (CAN), Local Interconnect Network (LIN), DeviceNet, ControlNet, Data Highway+, PROFIBUS, or PROFINET, among many others. An additional NIC 1368 may be included to allow connect to a second network, for example, a NIC 1368 providing communications to the cloud over Ethernet, and a second NIC 1368 providing communications to other devices over another type of network.

[0102] The interconnect 1356 may couple the processor 1352 to an external interface 1370 that is used to connect external devices or subsystems. The external devices may include sensors 1372, such as accelerometers, level sensors, flow sensors, optical light sensors, camera sensors, temperature sensors, a global positioning system (GPS) sensors, pressure sensors, barometric pressure sensors, and the like. The external interface 1370 further may be used to connect the IoT device 1350 to actuators 1374, such as power switches, valve actuators, an audible sound generator, a visual warning device, and the like.

[0103] In some optional examples, various input/output (I/O) devices may be present within, or connected to, the IoT device 1350. For example, a display or other output device 1384 may be included to show information, such as sensor readings or actuator position. An input device 1386, such as a touch screen or keypad may be included to accept input. An output device 1386 may include any number of forms of audio or visual display, including simple visual outputs such as binary status indicators (e.g., LEDs) and multi-character visual outputs, or more complex outputs such as display screens (e.g., LCD screens), with the

output of characters, graphics, multimedia objects, and the like being generated or produced from the operation of the IoT device 1350.

[0104] A battery 1376 may power the IoT device 1350, although in examples in which the IoT device 1350 is mounted in a fixed location, it may have a power supply coupled to an electrical grid. The battery 1376 may be a lithium ion battery, or a metal-air battery, such as a zinc-air battery, an aluminum-air battery, a lithium-air battery, and the like.

[0105] A battery monitor / charger 1378 may be included in the IoT device 1350 to track the state of charge (SoCh) of the battery 1376. The battery monitor / charger 1378 may be used to monitor other parameters of the battery 1376 to provide failure predictions, such as the state of health (SoH) and the state of function (SoF) of the battery 1376. The battery monitor / charger 1378 may include a battery monitoring integrated circuit, such as an LTC4020 or an LTC2990 from Linear Technologies, an ADT7488A from ON Semiconductor of Phoenix Arizona, or an IC from the UCD90xxx family from Texas Instruments of Dallas, TX. The battery monitor / charger 1378 may communicate the information on the battery 1376 to the processor 1352 over the interconnect 1356. The battery monitor / charger 1378 may also include an analog-to-digital (ADC) convertor that allows the processor 1352 to directly monitor the voltage of the battery 1376 or the current flow from the battery 1376. The battery parameters may be used to determine actions that the IoT device 1350 may perform, such as transmission frequency, mesh network operation, sensing frequency, and the like.

[0106] A power block 1380, or other power supply coupled to a grid, may be coupled with the battery monitor / charger 1378 to charge the battery 1376. In some examples, the power block 1380 may be replaced with a wireless power receiver to obtain the power wirelessly, for example, through a loop antenna in the IoT device 1350. A wireless battery charging circuit, such as an LTC4020 chip from Linear Technologies of Milpitas, CA, among others, may be included in the battery monitor / charger 1378. The specific charging circuits chosen depend on the size of the battery 1376, and thus, the current required. The charging may be performed using the Airfuel standard promulgated by the Airfuel Alliance, the Qi wireless charging standard promulgated by the Wireless

Power Consortium, or the Rezence charging standard, promulgated by the Alliance for Wireless Power, among others.

[0107] The storage 1358 may include instructions 1382 in the form of software, firmware, or hardware commands to implement the techniques described herein. Although such instructions 1382 are shown as code blocks included in the memory 1354 and the storage 1358, it may be understood that any of the code blocks may be replaced with hardwired circuits, for example, built into an application specific integrated circuit (ASIC).

[0108] In an example, the instructions 1382 provided via the memory 1354, the storage 1358, or the processor 1352 may be embodied as a non-transitory, machine readable medium 1360 including code to direct the processor 1352 to perform electronic operations in the IoT device 1350. The processor 1352 may access the non-transitory, machine readable medium 1360 over the interconnect 1356. For instance, the non-transitory, machine readable medium 1360 may be embodied by devices described for the storage 1358 of **FIG. 13** or may include specific storage units such as optical disks, flash drives, or any number of other hardware devices. The non-transitory, machine readable medium 1360 may include, provide, or invoke instructions 1388 to direct the processor 1352 to perform a specific sequence or flow of actions, for example, as described with respect to the flowchart(s) and block diagram(s) of operations and functionality depicted above.

[0109] In still a specific example, the instructions 1388 on the processor 1352 (separately, or in combination with the instructions 1388 of the machine readable medium 1360) may configure execution or operation of a trusted execution environment (TEE) 1390. In an example, the TEE 1390 operates as a protected area accessible to the processor 1352 for secure execution of instructions and secure access to data. Various implementations of the TEE 1390, and an accompanying secure area in the processor 1352 or the memory 1354 may be provided, for instance, through use of Intel® Software Guard Extensions (SGX) or ARM® TrustZone® hardware security extensions, Intel® Management Engine (ME), or Intel® Converged Security Manageability Engine (CSME). Other aspects of security hardening, hardware roots-of-trust, and trusted or

protected operations may be implemented in the device 1350 through the TEE 1390 and the processor 1352.

[0110] In further examples, a machine-readable medium also includes any tangible medium that is capable of storing, encoding or carrying instructions for execution by a machine and that cause the machine to perform any one or more of the methodologies of the present disclosure or that is capable of storing, encoding or carrying data structures utilized by or associated with such instructions. A “machine-readable medium” thus may include, but is not limited to, solid-state memories, and optical and magnetic media. Specific examples of machine-readable media include non-volatile memory, including but not limited to, by way of example, semiconductor memory devices (e.g., electrically programmable read-only memory (EPROM), electrically erasable programmable read-only memory (EEPROM)) and flash memory devices; magnetic disks such as internal hard disks and removable disks; magneto-optical disks; and CD-ROM and DVD-ROM disks. The instructions embodied by a machine-readable medium may further be transmitted or received over a communications network using a transmission medium via a network interface device utilizing any one of a number of transfer protocols (e.g., HTTP).

[0111] It should be understood that the functional units or capabilities described in this specification may have been referred to or labeled as components or modules, in order to more particularly emphasize their implementation independence. Such components may be embodied by any number of software or hardware forms. For example, a component or module may be implemented as a hardware circuit comprising custom very-large-scale integration (VLSI) circuits or gate arrays, off-the-shelf semiconductors such as logic chips, transistors, or other discrete components. A component or module may also be implemented in programmable hardware devices such as field programmable gate arrays, programmable array logic, programmable logic devices, or the like. Components or modules may also be implemented in software for execution by various types of processors. An identified component or module of executable code may, for instance, comprise one or more physical or logical blocks of computer instructions, which may, for instance, be organized as an object, procedure, or function. Nevertheless, the executables of an

identified component or module need not be physically located together, but may comprise disparate instructions stored in different locations which, when joined logically together, comprise the component or module and achieve the stated purpose for the component or module.

5 [0112] Indeed, a component or module of executable code may be a single instruction, or many instructions, and may even be distributed over several different code segments, among different programs, and across several memory devices or processing systems. In particular, some aspects of the described process (such as code rewriting and code analysis) may take place on a different
10 processing system (e.g., in a computer in a data center), than that in which the code is deployed (e.g., in a computer embedded in a sensor or robot). Similarly, operational data may be identified and illustrated herein within components or modules, and may be embodied in any suitable form and organized within any suitable type of data structure. The operational data may be collected as a single
15 data set, or may be distributed over different locations including over different storage devices, and may exist, at least partially, merely as electronic signals on a system or network. The components or modules may be passive or active, including agents operable to perform desired functions.

[0113] Additional examples of the presently described method, system, and
20 device embodiments include the following, non-limiting configurations. Each of the following non-limiting examples may stand on its own, or may be combined in any permutation or combination with any one or more of the other examples provided below or throughout the present disclosure.

[0114] Example 1 is a server device comprising: processing circuitry; and at
25 least one storage device including instructions embodied thereon, wherein the instructions, which when executed by the processing circuitry, configure the processing circuitry to perform operations of a soft reset operation, the operations to: define a soft reset state; cause a check of a secure virtual resource (SVR) of the server device, while in the soft reset state; and transition from the
30 soft reset state in response to an event.

[0115] In Example 2, the subject matter of Example 1 includes, wherein the instructions to define the soft reset state the server device include instructions,

which when executed by the processing circuitry, cause the processing circuitry to detect a condition and set a parameter value based on the condition.

[0116] In Example 3, the subject matter of Example 2 includes, wherein the condition is that a secure virtual resource (SVR) of the server device is
5 corrupted.

[0117] In Example 4, the subject matter of Examples 1–3 includes, wherein to define the soft reset state, a parameter value of the server device is set by a client service.

[0118] In Example 5, the subject matter of Example 4 includes, wherein the
10 client service implements an onboarding tool.

[0119] In Example 6, the subject matter of Examples 4–5 includes, wherein the client service owns a resource provided by the server device.

[0120] In Example 7, the subject matter of Examples 1–6 includes, instructions, which when executed by the processing circuitry, cause the
15 processing circuitry to operate using client-directed instructions.

[0121] In Example 8, the subject matter of Examples 1–7 includes, wherein the instructions to cause the check of the SVR of the server device include instructions, which when executed by the processing circuitry, cause the processing circuitry to check the SVR at the server device during a state after the
20 soft reset state.

[0122] In Example 9, the subject matter of Example 8 includes, wherein the state after the soft reset state is a ready for provisioning state.

[0123] In Example 10, the subject matter of Examples 1–9 includes, wherein the instructions to cause the check of the SVR of the server device include
25 instructions, which when executed by the processing circuitry, cause the processing circuitry to check the SVR of the server device at a client service during the soft reset state.

[0124] In Example 11, the subject matter of Examples 1–10 includes, wherein the event is a result of a resource consistency check, and the instructions to
30 transition from the soft reset state in response to the event include instructions, which when executed by the processing circuitry, cause the processing circuitry to: transmit resources to a client service for the resource consistency check; and

transition to a reset state or ready for provisioning state based on the resource consistency check.

[0125] In Example 12, the subject matter of Examples 1–11 includes, wherein the event is a result of a credential check, and wherein the instructions to
5 transition from the soft reset state in response to the event include instructions, which when executed by the processing circuitry, cause the processing circuitry to: perform the credential check on a credential of an owner of the server device; and transition to a reset state when the credential check fails.

[0126] In Example 13, the subject matter of Example 12 includes, wherein the
10 credential check fails when the credential cannot be found or is determined to be corrupted.

[0127] Example 14 is a method implemented by a server device, the method comprising: defining a soft reset state; causing a check of a secure virtual resource (SVR) of the server device, while in the soft reset state; and
15 transitioning from the soft reset state in response to an event.

[0128] In Example 15, the subject matter of Example 14 includes, wherein defining the soft reset state further comprises detecting a condition and set a parameter value based on the condition.

[0129] In Example 16, the subject matter of Example 15 includes, wherein the
20 condition is that a secure virtual resource (SVR) of the server device is corrupted.

[0130] In Example 17, the subject matter of Examples 14–16 includes, wherein to define the soft reset state, a parameter value of the server device is set by a client service.

[0131] In Example 18, the subject matter of Example 17 includes, wherein the
25 client service implements an onboarding tool.

[0132] In Example 19, the subject matter of Examples 17–18 includes, wherein the client service owns a resource provided by the server device.

[0133] In Example 20, the subject matter of Examples 14–19 includes,
30 operating using client-directed instructions, while in the soft reset state.

[0134] In Example 21, the subject matter of Examples 14–20 includes, wherein causing the check of the SVR includes checking the SVR at the server device during a state after the soft reset state.

[0135] In Example 22, the subject matter of Example 21 includes, wherein the state after the soft reset state is a ready for provisioning state.

[0136] In Example 23, the subject matter of Examples 14–22 includes, wherein causing the check of the SVR includes checking the SVR at a client service during the soft reset state.

[0137] In Example 24, the subject matter of Examples 14–23 includes, wherein the event is a result of a resource consistency check, and transitioning from the soft reset state in response to the event comprises: transmitting resources to a client service for the resource consistency check; and transitioning to a reset state or ready for provisioning state based on the resource consistency check.

[0138] In Example 25, the subject matter of Examples 14–24 includes, wherein the event is a result of a credential check, and transitioning from the soft reset state in response to the event comprises: performing the credential check on a credential of an owner of the server device; and transitioning to a reset state when the credential check fails.

[0139] In Example 26, the subject matter of Example 25 includes, wherein the credential check fails when the credential cannot be found or is determined to be corrupted.

[0140] Example 27 is at least one machine readable medium including instructions, which when executed by device hardware, cause the device hardware to perform or implement any of the methods of Examples 14-26.

[0141] Example 28 is an apparatus comprising means for performing any of the methods of Examples 14-26.

[0142] Example 29 is at least one storage device including instructions embodied thereon, wherein the instructions, which when executed by processing circuitry of a server device, configure the processing circuitry to perform operations of a soft reset operation of the server device, the operations to: define a soft reset state; cause a check of a secure virtual resource (SVR) of the server device, while in the soft reset state; and transition from the soft reset state in response to an event.

[0143] In Example 30, the subject matter of Example 29 includes, wherein the instructions to define the soft reset state the at least one storage device include

instructions, which when executed by the processing circuitry, cause the processing circuitry to detect a condition and set a parameter value based on the condition.

5 [0144] In Example 31, the subject matter of Example 30 includes, wherein the condition is that a secure virtual resource (SVR) of the server device is corrupted.

[0145] In Example 32, the subject matter of Examples 29–31 includes, wherein to define the soft reset state, a parameter value of the server device is set by a client service.

10 [0146] In Example 33, the subject matter of Example 32 includes, wherein the client service implements an onboarding tool.

[0147] In Example 34, the subject matter of Examples 32–33 includes, wherein the client service owns a resource provided by the server device.

15 [0148] In Example 35, the subject matter of Examples 29–34 includes, instructions, which when executed by the processing circuitry, cause the processing circuitry to operate using client-directed instructions.

[0149] In Example 36, the subject matter of Examples 29–35 includes, wherein the instructions to cause the check of the SVR of the server device include instructions, which when executed by the processing circuitry, cause the
20 processing circuitry to check the SVR at the server device during a state after the soft reset state.

[0150] In Example 37, the subject matter of Example 36 includes, wherein the state after the soft reset state is a ready for provisioning state.

25 [0151] In Example 38, the subject matter of Examples 29–37 includes, wherein the instructions to cause the check of the SVR of the server device include instructions, which when executed by the processing circuitry, cause the processing circuitry to check the SVR of the server device at a client service during the soft reset state.

30 [0152] In Example 39, the subject matter of Examples 29–38 includes, wherein the event is a result of a resource consistency check, and the instructions to transition from the soft reset state in response to the event include instructions, which when executed by the processing circuitry, cause the processing circuitry to: transmit resources to a client service for the resource consistency check; and

transition to a reset state or ready for provisioning state based on the resource consistency check.

[0153] In Example 40, the subject matter of Examples 29–39 includes, wherein the event is a result of a credential check, and wherein the instructions to transition from the soft reset state in response to the event include instructions, which when executed by the processing circuitry, cause the processing circuitry to: perform the credential check on a credential of an owner of the server device; and transition to a reset state when the credential check fails.

[0154] In Example 41, the subject matter of Example 40 includes, wherein the credential check fails when the credential cannot be found or is determined to be corrupted.

[0155] Example 42 is an apparatus, comprising: means for defining a soft reset state; means for causing a check of a secure virtual resource (SVR) of the apparatus, while in the soft reset state; and means for transitioning from the soft reset state in response to an event.

[0156] In Example 43, the subject matter of Example 42 includes, means for detecting a condition and means for setting a parameter value based on the condition.

[0157] In Example 44, the subject matter of Example 43 includes, wherein the condition is that a secure virtual resource (SVR) of the apparatus is corrupted.

[0158] In Example 45, the subject matter of Examples 42–44 includes, wherein to define the soft reset state, a parameter value of the apparatus is set by a client service.

[0159] In Example 46, the subject matter of Example 45 includes, wherein the client service implements an onboarding tool.

[0160] In Example 47, the subject matter of Examples 45–46 includes, wherein the client service owns a resource provided by the apparatus.

[0161] In Example 48, the subject matter of Examples 42–47 includes, means for operating using client-directed instructions, while in the soft reset state.

[0162] In Example 49, the subject matter of Examples 42–48 includes, wherein the means for causing the check of the SVR includes means for checking the SVR at the apparatus during a state after the soft reset state.

[0163] In Example 50, the subject matter of Example 49 includes, wherein the state after the soft reset state is a ready for provisioning state.

[0164] In Example 51, the subject matter of Examples 42–50 includes, wherein the means for causing the check of the SVR includes means for
5 checking the SVR at a client service during the soft reset state.

[0165] In Example 52, the subject matter of Examples 42–51 includes, wherein the event is a result of a resource consistency check, and the means for transitioning from the soft reset state in response to the event comprises: means for transmitting resources to a client service for the resource consistency check;
10 and means for transitioning to a reset state or ready for provisioning state based on the resource consistency check.

[0166] In Example 53, the subject matter of Examples 42–52 includes, wherein the event is a result of a credential check, and the means for transitioning from the soft reset state in response to the event comprises: means
15 for performing the credential check on a credential of an owner of the apparatus; and means for transitioning to a reset state when the credential check fails.

[0167] In Example 54, the subject matter of Example 53 includes, wherein the credential check fails when the credential cannot be found or is determined to be corrupted.

20 [0168] Example 55 is a device fog adapted to perform the operations of any of Examples 1 to 54.

[0169] Example 56 is a cloud service server adapted to perform the operations of a cloud service invoked by any of Examples 1 to 54.

[0170] Example 57 is an edge computing device adapted to perform the
25 operations of a client device invoked by any of Examples 1 to 54.

[0171] Example 58 is a device onboarding service server adapted to perform the operations of onboarding invoked by any of Examples 1 to 54.

[0172] Example 59 is an Open Connectivity Foundation (OCF) device, configured as a server, client, or intermediary according to an OCF specification,
30 comprising means to implement the operations of any of Examples 1 to 54.

[0173] Example 60 is an Internet of Things (IoT) network topology, the IoT network topology comprising respective communication links adapted to perform communications for the operations of any of Examples 1 to 54.

[0174] Example 61 is a network comprising respective devices and device communication mediums for performing any of the operations of Examples 1 to 54.

5 [0175] Example 62 is an apparatus comprising means for performing any of the operations of Examples 1 to 54.

[0176] Example 63 is a system to perform the operations of any of Examples 1 to 62.

[0177] Example 64 is a method to perform the operations of any of Examples 1 to 62.

10 [0178] In the above Detailed Description, various features may be grouped together to streamline the disclosure. However, the claims may not set forth every feature disclosed herein as embodiments may feature a subset of said features. Further, embodiments may include fewer features than those disclosed in a particular example. Thus, the following claims are hereby incorporated into
15 the Detailed Description, with a claim standing on its own as a separate embodiment.

CLAIMS

What is claimed is:

1. A server device comprising:
processing circuitry; and
at least one storage device including instructions embodied thereon,
5 wherein the instructions, which when executed by the processing circuitry,
configure the processing circuitry to perform operations of a soft reset operation,
the operations to:
define a soft reset state;
cause a check of a secure virtual resource (SVR) of the server
10 device, while in the soft reset state; and
transition from the soft reset state in response to an event.
2. The server device of claim 1, wherein the instructions to define the soft
reset state the server device include instructions, which when executed by the
15 processing circuitry, cause the processing circuitry to detect a condition and set a
parameter value based on the condition.
3. The server device of claim 2, wherein the condition is that a secure
virtual resource (SVR) of the server device is corrupted.
20
4. The server device of claim 1, wherein to define the soft reset state, a
parameter value of the server device is set by a client service.
5. The server device of claim 4, wherein the client service implements an
25 onboarding tool.
6. The server device of claim 4, wherein the client service owns a
resource provided by the server device.
- 30 7. The server device of claim 1, further comprising instructions, which
when executed by the processing circuitry, cause the processing circuitry to
operate using client-directed instructions.

8. The server device of claim 1, wherein the instructions to cause the check of the SVR of the server device include instructions, which when executed by the processing circuitry, cause the processing circuitry to check the SVR at
5 the server device during a state after the soft reset state.

9. The server device of claim 8, wherein the state after the soft reset state is a ready for provisioning state.

10 10. The server device of claim 1, wherein the instructions to cause the check of the SVR of the server device include instructions, which when executed by the processing circuitry, cause the processing circuitry to check the SVR of the server device at a client service during the soft reset state.

15 11. The server device of claim 1, wherein the event is a result of a resource consistency check, and the instructions to transition from the soft reset state in response to the event include instructions, which when executed by the processing circuitry, cause the processing circuitry to:

transmit resources to a client service for the resource consistency check;
20 and
transition to a reset state or ready for provisioning state based on the resource consistency check.

12. The server device of claim 1, wherein the event is a result of a
25 credential check, and wherein the instructions to transition from the soft reset state in response to the event include instructions, which when executed by the processing circuitry, cause the processing circuitry to:

perform the credential check on a credential of an owner of the server device; and
30 transition to a reset state when the credential check fails.

13. The server device of claim 12, wherein the credential check fails when the credential cannot be found or is determined to be corrupted.

14. A method implemented by a server device, the method comprising:
defining a soft reset state;
causing a check of a secure virtual resource (SVR) of the server device,
5 while in the soft reset state; and
transitioning from the soft reset state in response to an event.

15. The method of claim 14, wherein defining the soft reset state further
comprises detecting a condition and set a parameter value based on the
10 condition.

16. The method of claim 15, wherein the condition is that a secure virtual
resource (SVR) of the server device is corrupted.

15 17. The method of claim 14, wherein to define the soft reset state, a
parameter value of the server device is set by a client service.

18. The method of claim 17, wherein the client service implements an
onboarding tool, and wherein the client service owns a resource provided by the
20 server device.

19. The method of claim 14, further comprising operating using client-
directed instructions, while in the soft reset state.

25 20. The method of claim 14, wherein causing the check of the SVR
includes checking the SVR at the server device during a state after the soft reset
state.

21. The method of claim 21, wherein the state after the soft reset state is a
30 ready for provisioning state.

22. The method of claim 14, wherein causing the check of the SVR
includes checking the SVR at a client service during the soft reset state.

23. The method of claim 14, wherein the event is a result of a resource consistency check, and transitioning from the soft reset state in response to the event comprises:

- 5 transmitting resources to a client service for the resource consistency check; and
- transitioning to a reset state or ready for provisioning state based on the resource consistency check.

10 24. The method of claim 14, wherein the event is a result of a credential check, and transitioning from the soft reset state in response to the event comprises:

- performing the credential check on a credential of an owner of the server device; and
- 15 transitioning to a reset state when the credential check fails;
- wherein the credential check fails when the credential cannot be found or is determined to be corrupted.

25. At least one machine readable medium including instructions, which

20 when executed by device hardware, cause the device hardware to perform or implement any of the methods of claims 14-24.

1/13

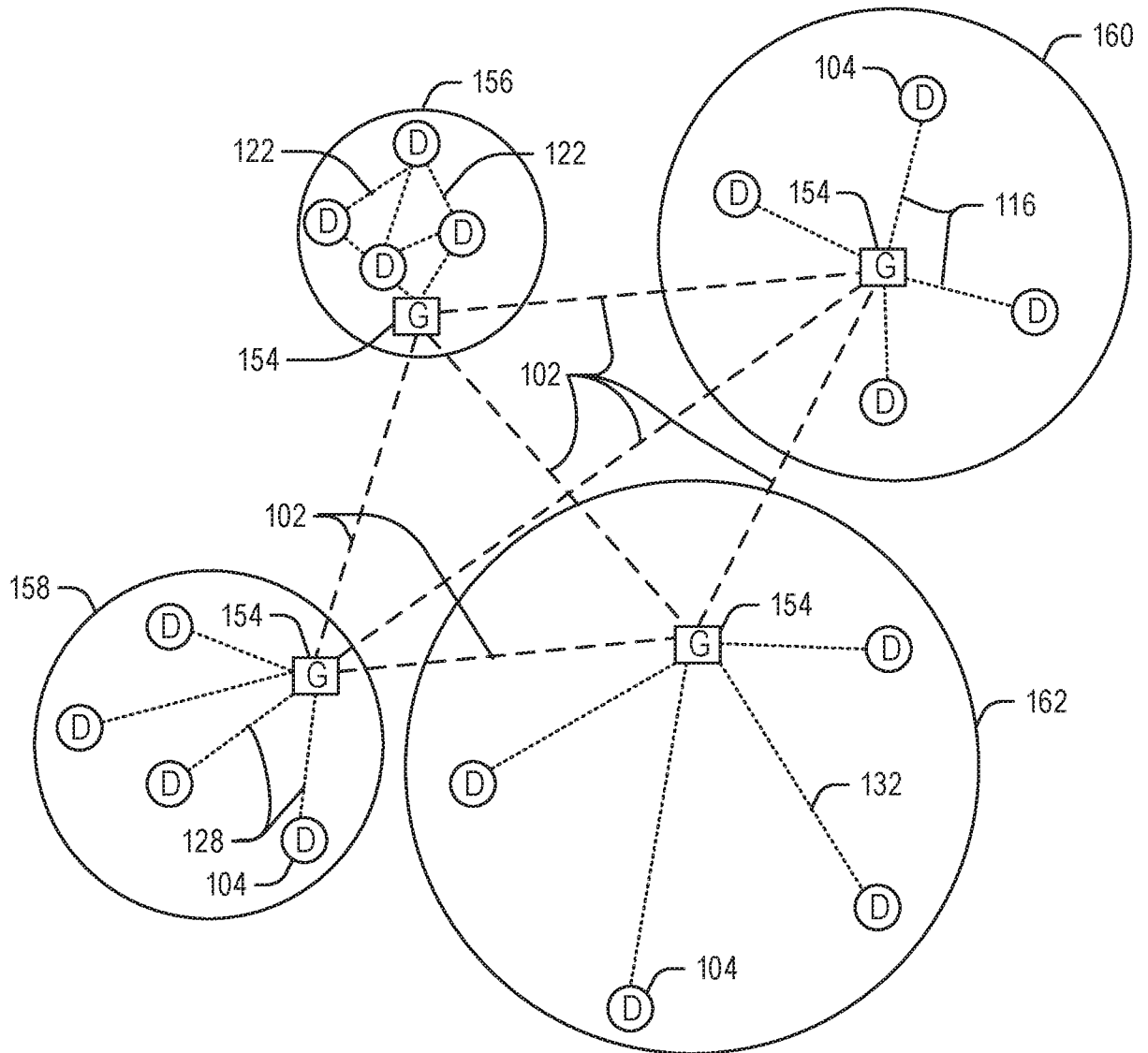


FIG. 1

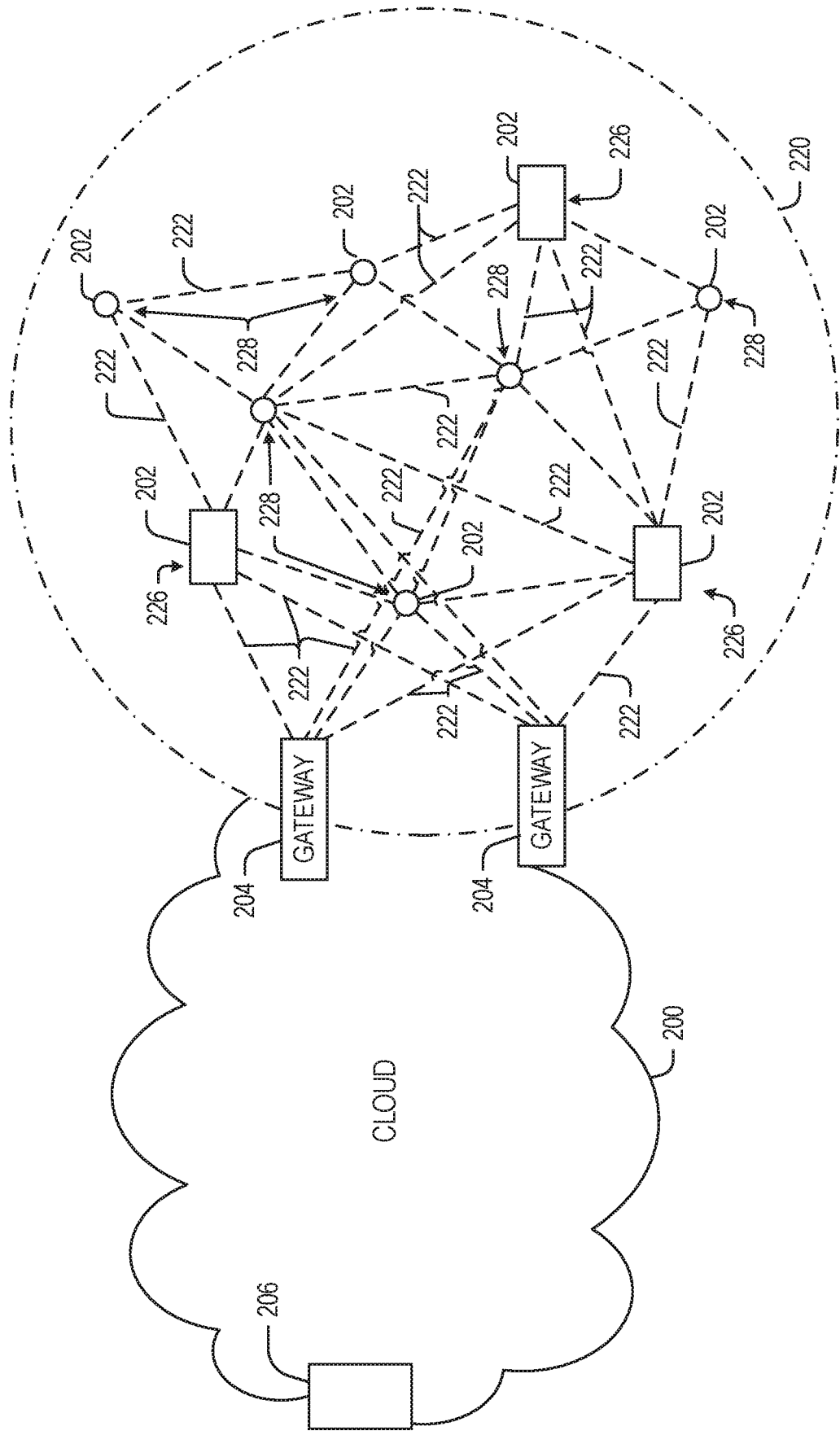


FIG. 2

3/13

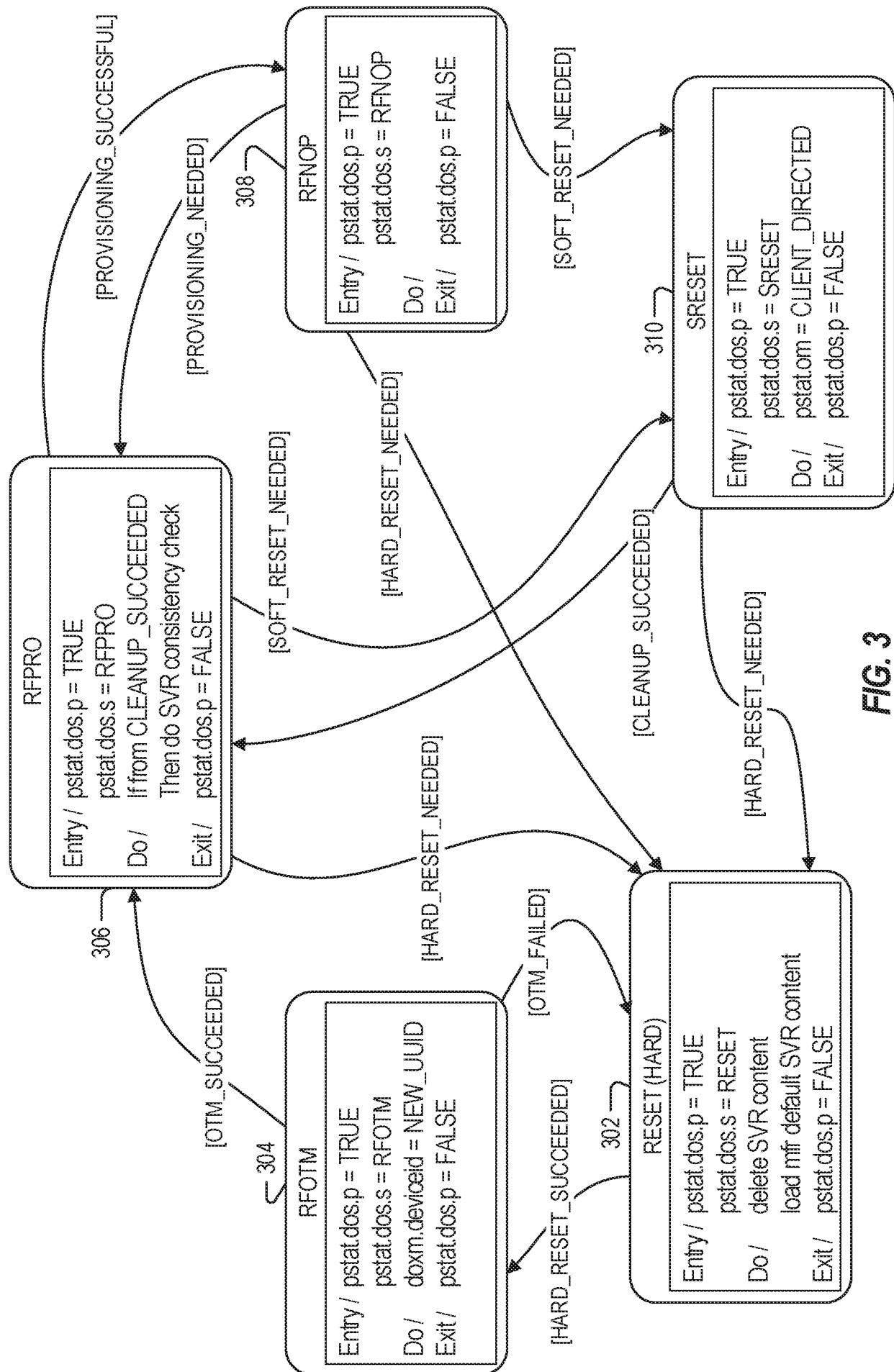


FIG. 3

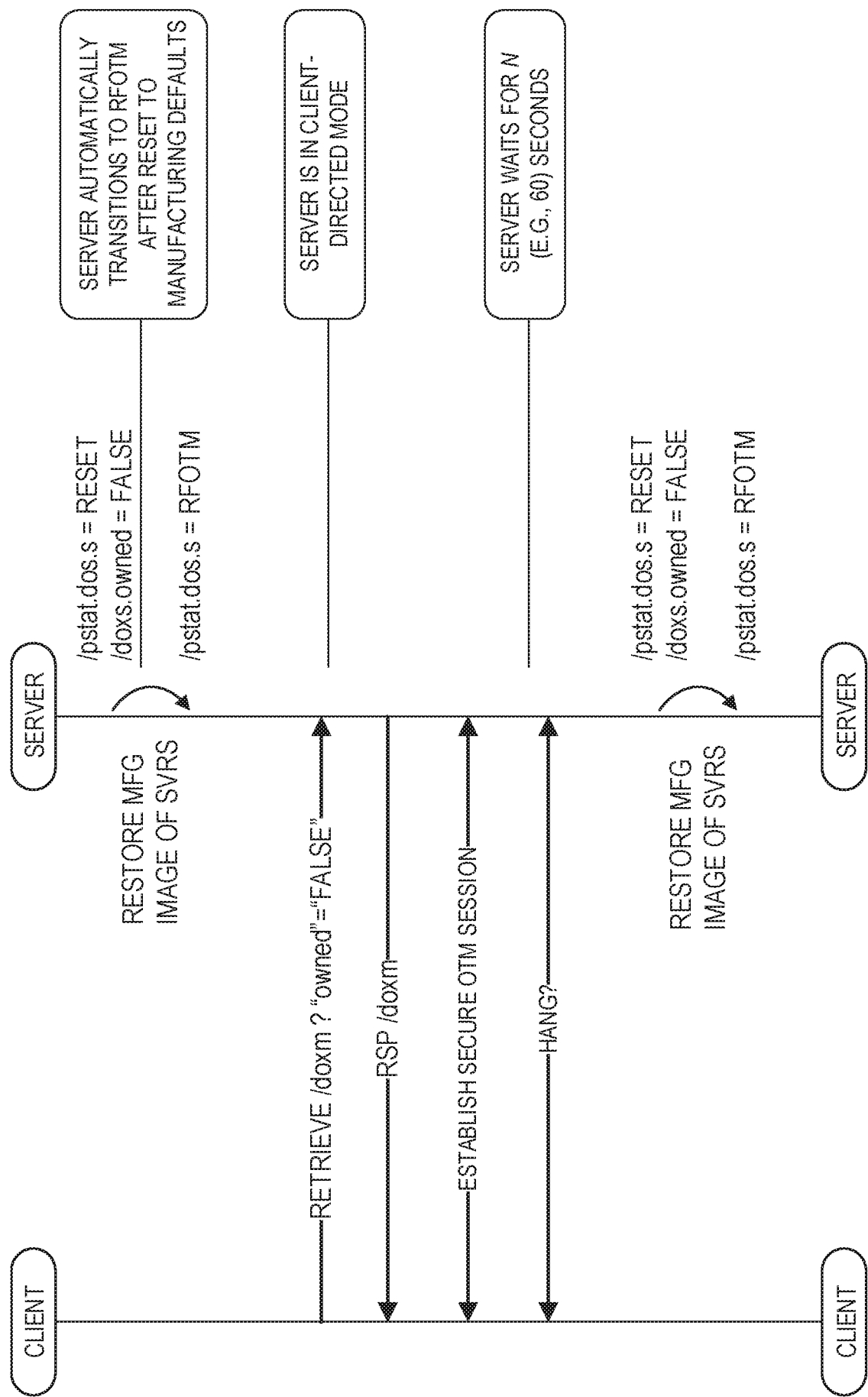


FIG. 4

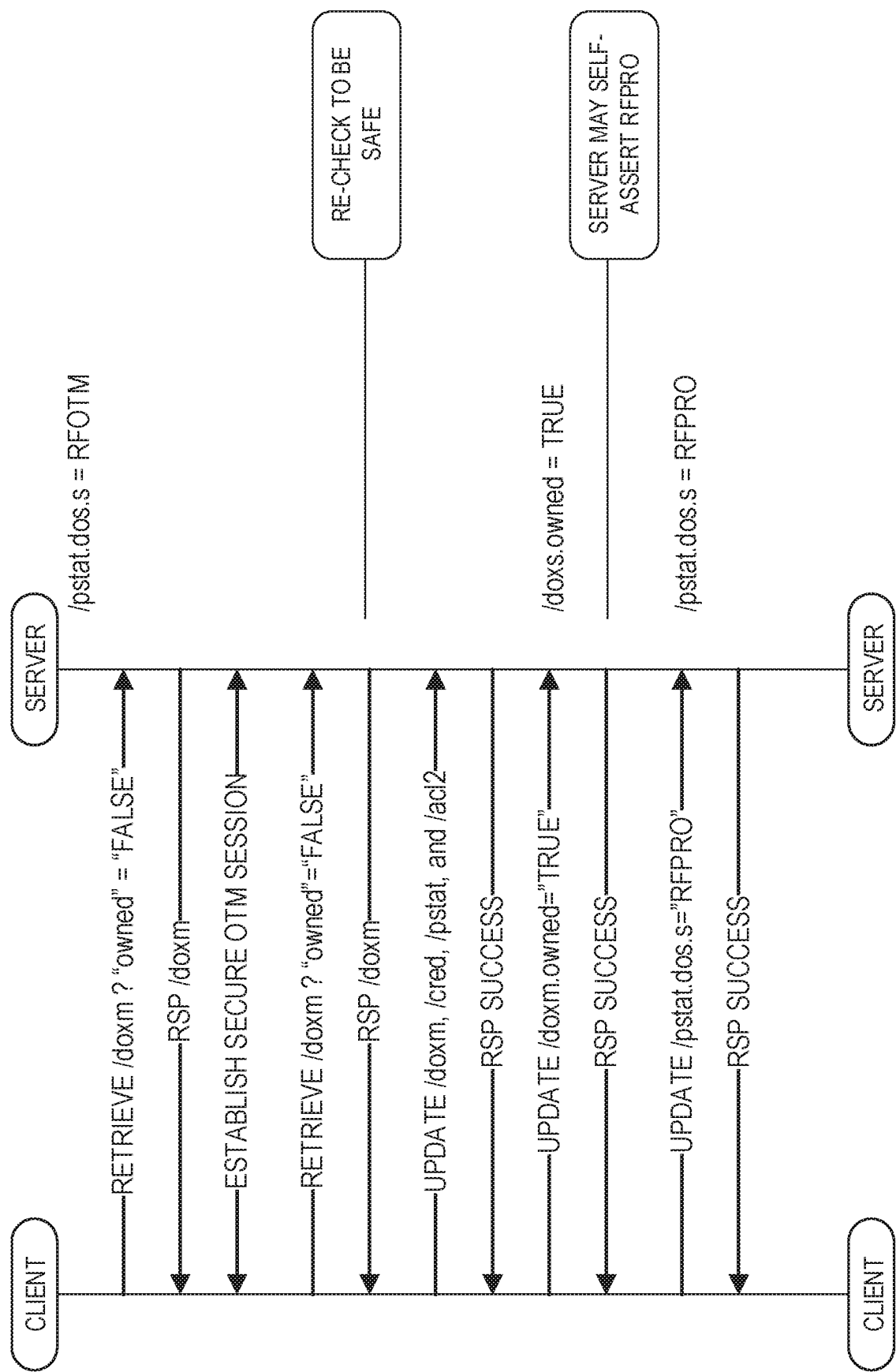


FIG. 5

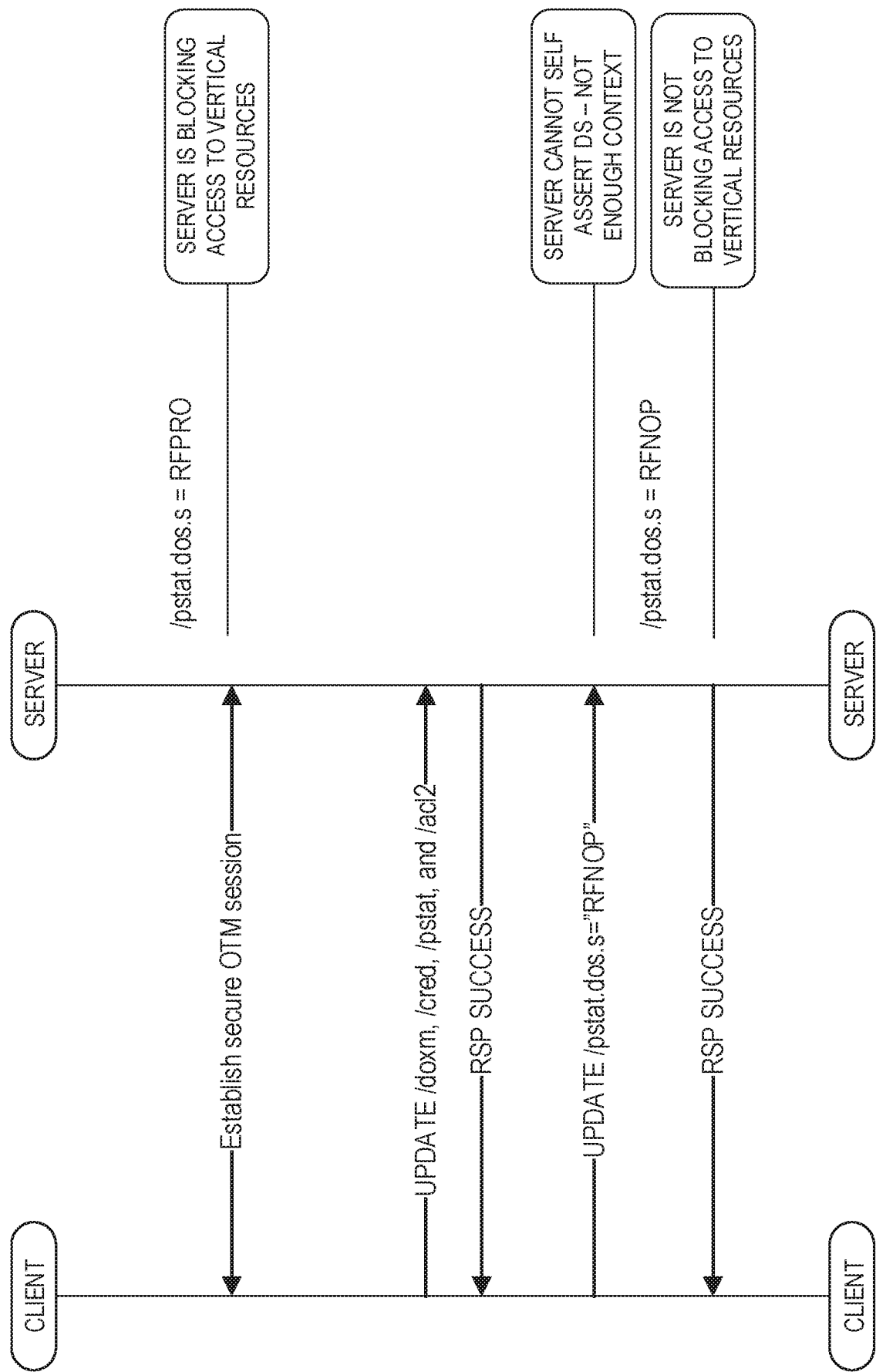


FIG. 6

7/13

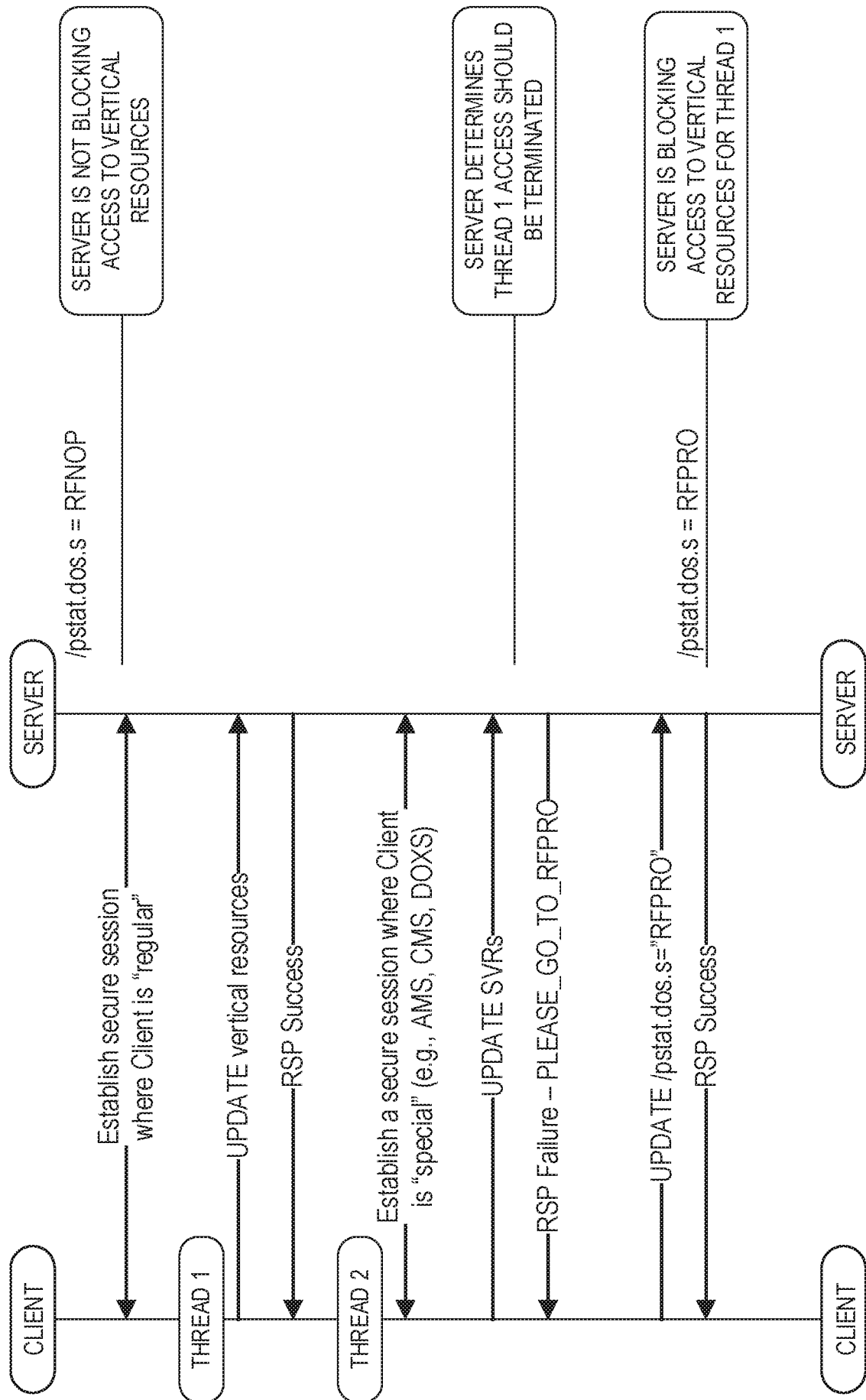


FIG. 7

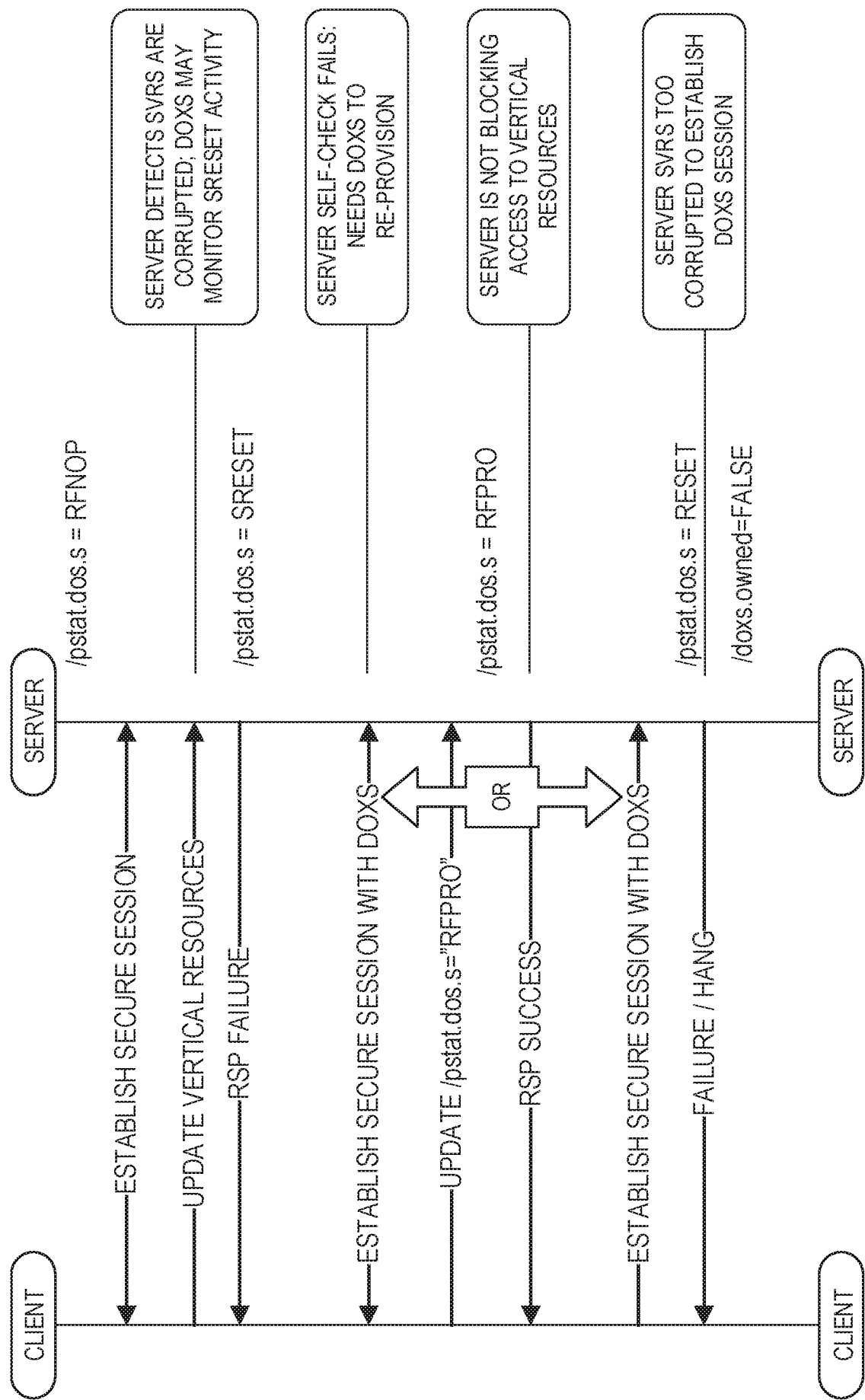
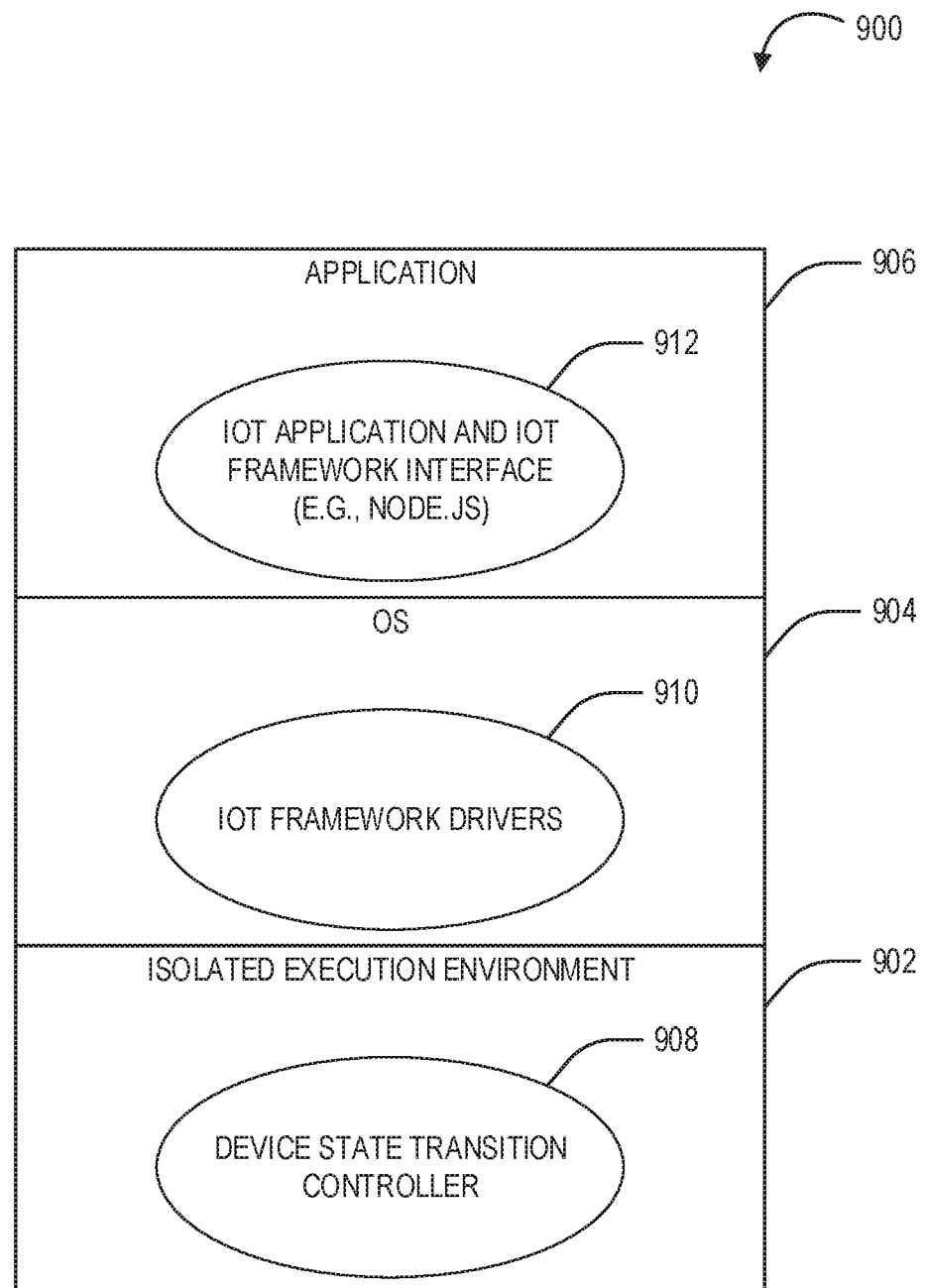


FIG. 8

9/13

**FIG. 9**

10/13

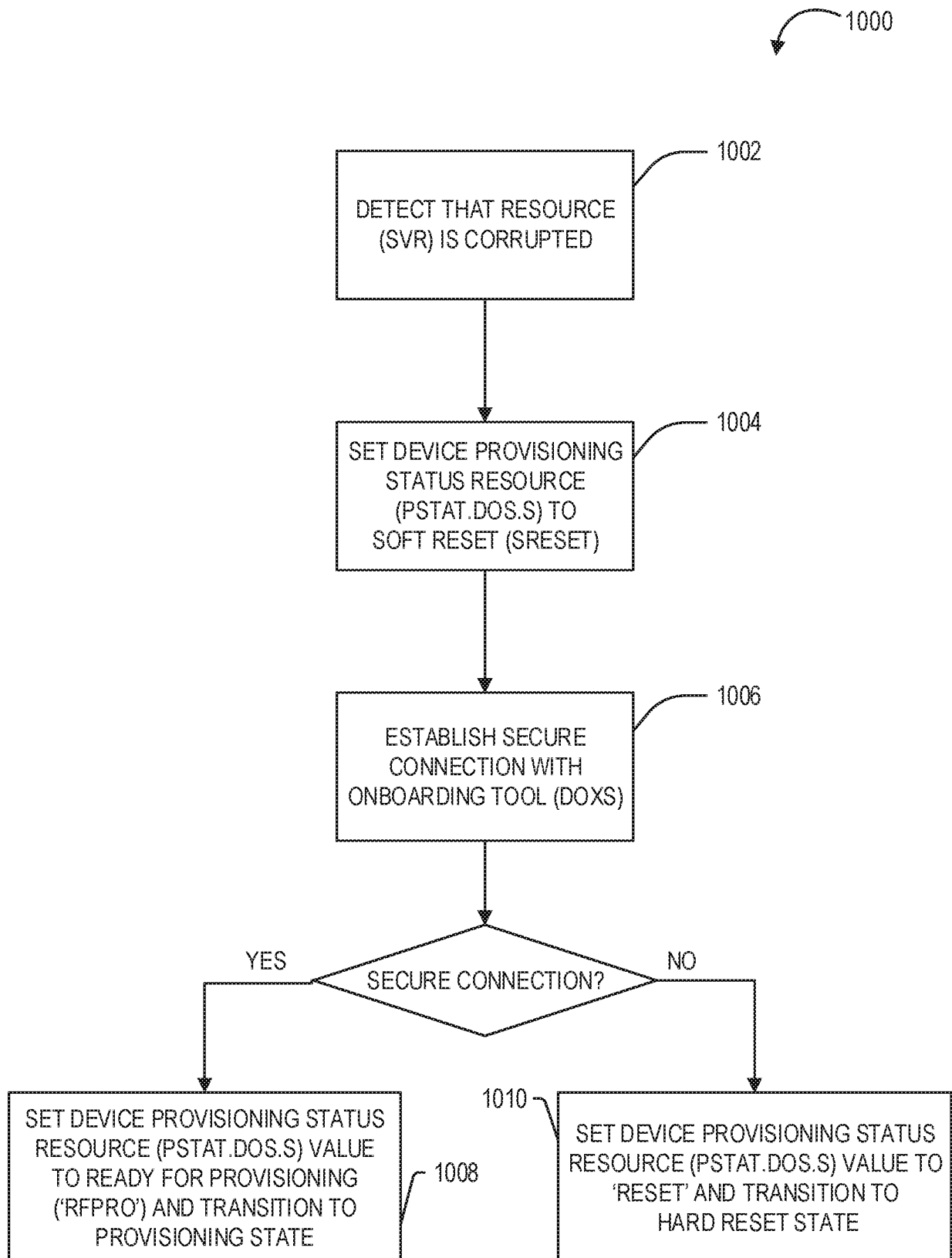
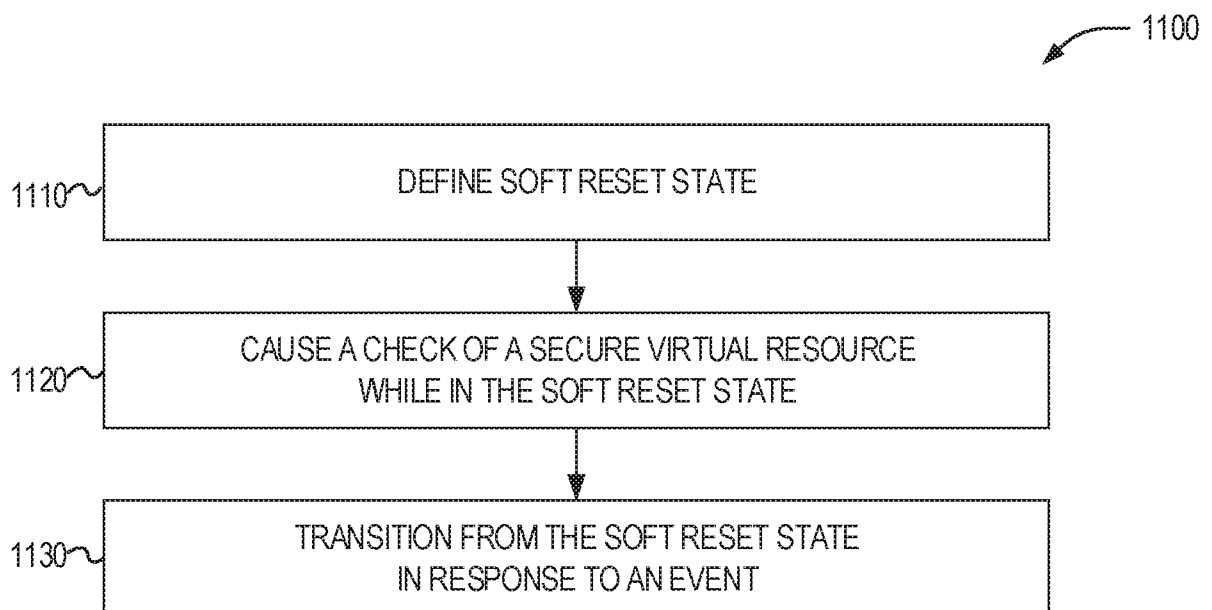


FIG. 10

11/13

**FIG. 11**

12/13

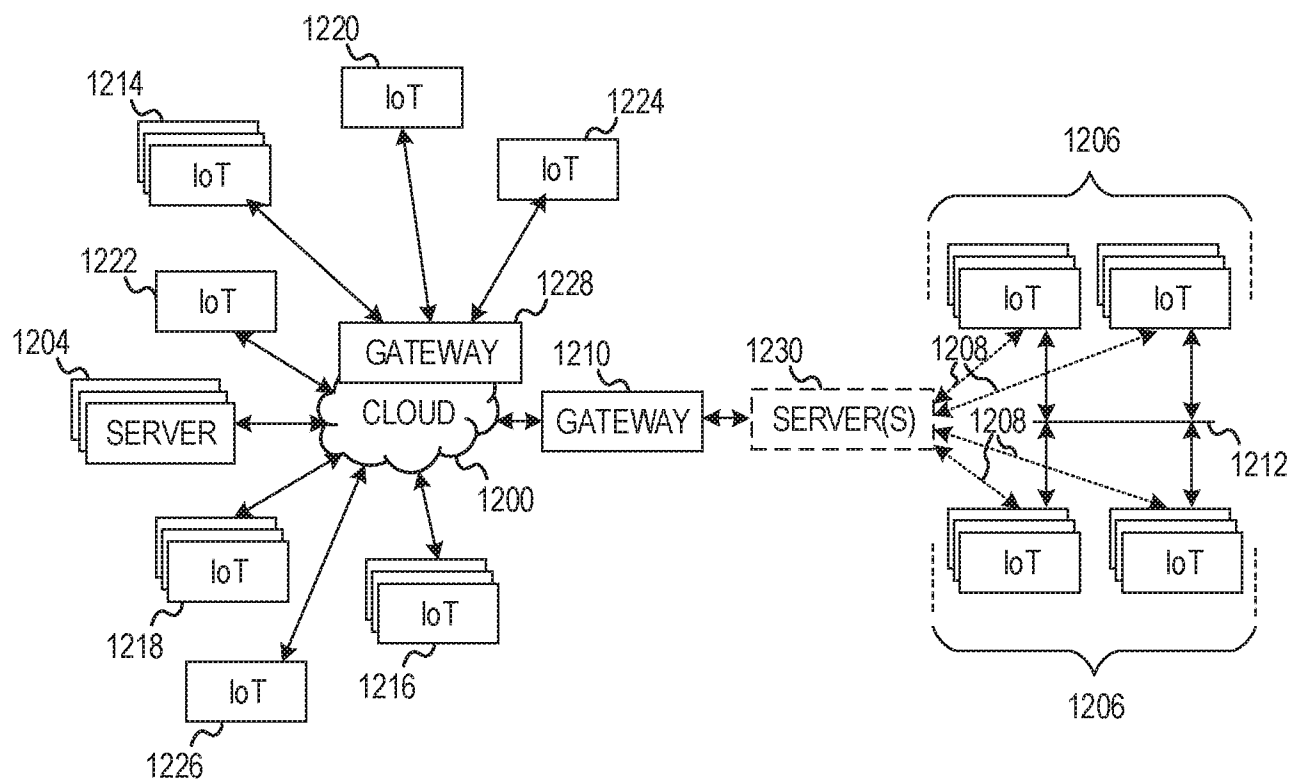


FIG. 12

13/13

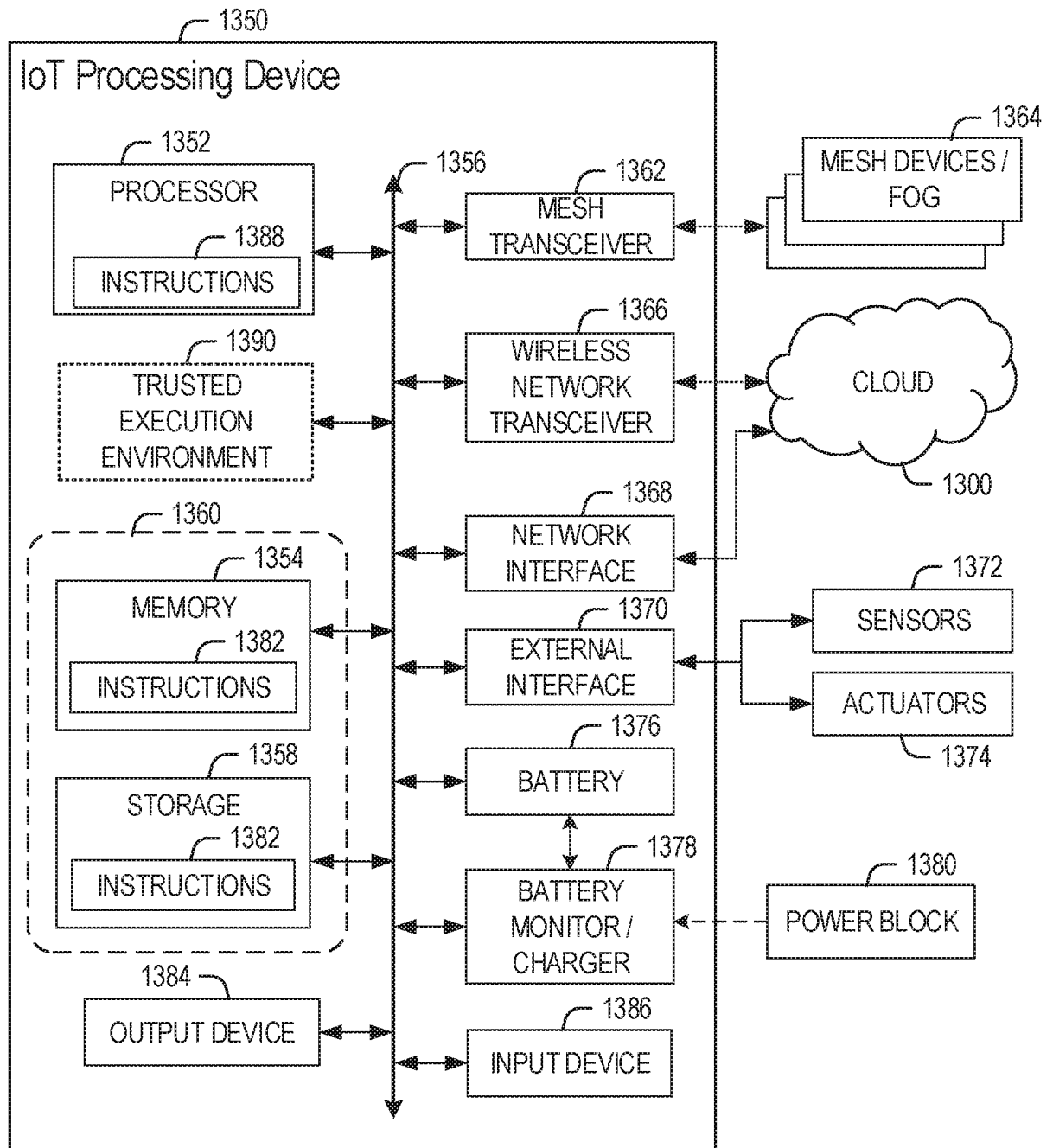


FIG. 13

A. CLASSIFICATION OF SUBJECT MATTER**H04L 29/06(2006.01)i, H04L 29/08(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 29/06; H04L 9/08; G06F 11/07; H04W 48/04; G06F 9/44; H04L 9/32; G06F 9/06; H04W 24/08; H04L 29/08

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Korean utility models and applications for utility models

Japanese utility models and applications for utility models

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

eKOMPASS(KIPO internal) & Keywords: state transition, soft reset state, ready for provisioning state, hard reset state, secure virtual resource (SVR), onboarding tool

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 2015-0087291 A1 (JASPER TECHNOLOGIES, INC.) 26 March 2015 See paragraphs [0003], [0009], [0030], [0036], [0040], [0044], [0083], [0108]; claim 12; and figures 1A, 2.	1-25
Y	US 2014-0189339 A1 (TELEFONAKTIEBOLAGET L M ERICSSON (PUBL)) 03 July 2014 See paragraphs [0021], [0062], [0101]; claim 8; and figure 9.	1-25
Y	OIC SECURITY SPECIFICATION V1.1.0. [online]. Open Connectivity Foundation (OCF). 31 December 2016. Retrieved from the Internet: <URL: https://openconnectivity.org/specs/OIC_Security_Specification_v1.1.0.pdf >. page 10, line number 260 - page 102, line number 2356; and tables 7, 38.	3-13, 16-24
A	US 2013-0046981 A1 (PAUL D. DUCHARME et al.) 21 February 2013 See paragraphs [0015], [0034]-[0035], [0038]; and figures 1, 3.	1-25
A	US 2012-0042215 A1 (SHUNJI MURAYAMA et al.) 16 February 2012 See paragraphs [0007], [0043], [0051], [0071]; and figure 10.	1-25



Further documents are listed in the continuation of Box C.



See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search

25 April 2018 (25.04.2018)

Date of mailing of the international search report

25 April 2018 (25.04.2018)

Name and mailing address of the ISA/KR

International Application Division

Korean Intellectual Property Office

189 Cheongsa-ro, Seo-gu, Daejeon, 35208, Republic of Korea

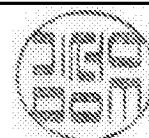


Facsimile No. +82-42-481-8578

Authorized officer

KIM, Seong Woo

Telephone No. +82-42-481-3348



INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/US2017/069054

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2015-0087291 A1	26/03/2015	CA 2840314 A1	27/12/2012
		CN 104106256 A	15/10/2014
		CN 104106256 B	26/10/2016
		CN 107105385 A	29/08/2017
		EP 1875618 A2	09/01/2008
		EP 1875618 A4	25/01/2012
		EP 2522121 A1	14/11/2012
		EP 2522121 A4	17/09/2014
		EP 2724522 A1	30/04/2014
		EP 2724522 A4	24/12/2014
		EP 2724522 B1	02/11/2016
		EP 2759120 A1	30/07/2014
		EP 2759120 A4	22/04/2015
		EP 2759120 B1	16/03/2016
		EP 2763441 A1	06/08/2014
		EP 2763441 B1	06/09/2017
		EP 2829047 A1	28/01/2015
		EP 2829047 A4	09/09/2015
		EP 2829047 B1	21/12/2016
		EP 2936867 A1	28/10/2015
		EP 2936867 A4	30/12/2015
		EP 2939458 A1	04/11/2015
		EP 2939458 A4	02/11/2016
		EP 2939458 B1	21/02/2018
		EP 3029967 A1	08/06/2016
		EP 3029967 B1	14/02/2018
		EP 3253092 A1	06/12/2017
		ES 2574421 T3	17/06/2016
		JP 2013-516907 A	13/05/2013
		JP 2014-529383 A	06/11/2014
		JP 2015-213347 A	26/11/2015
		JP 2015-505190 A	16/02/2015
		JP 5769730 B2	26/08/2015
		JP 6023802 B2	09/11/2016
		JP 6062494 B2	18/01/2017
		JP 6195573 B2	13/09/2017
		US 2006-0246949 A1	02/11/2006
		US 2010-0204667 A1	12/08/2010
		US 2011-0164511 A1	07/07/2011
		US 2012-0142314 A1	07/06/2012
		US 2012-0231785 A1	13/09/2012
		US 2012-0238265 A1	20/09/2012
		US 2012-0282891 A1	08/11/2012
		US 2012-0327787 A1	27/12/2012
		US 2012-0327813 A1	27/12/2012
		US 2012-0331421 A1	27/12/2012
		US 2013-0017830 A1	17/01/2013
		US 2013-0065575 A1	14/03/2013
		US 2013-0148532 A1	13/06/2013

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/US2017/069054

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
		US 2013-0176940 A1	11/07/2013
		US 2013-0182554 A1	18/07/2013
		US 2013-0217361 A1	22/08/2013
		US 2013-0273911 A1	17/10/2013
		US 2013-0331080 A1	12/12/2013
		US 2014-0011478 A1	09/01/2014
		US 2014-0024361 A1	23/01/2014
		US 2014-0120912 A1	01/05/2014
		US 2014-0179263 A1	26/06/2014
		US 2014-0199961 A1	17/07/2014
		US 2014-0199962 A1	17/07/2014
		US 2014-0242943 A1	28/08/2014
		US 2014-0242951 A1	28/08/2014
		US 2014-0242986 A1	28/08/2014
		US 2014-0273945 A1	18/09/2014
		US 2014-0315514 A1	23/10/2014
		US 2014-0357221 A1	04/12/2014
		US 2014-0357222 A1	04/12/2014
		US 2014-0378120 A1	25/12/2014
		US 2015-0024708 A1	22/01/2015
		US 2015-0071054 A1	12/03/2015
		US 2015-0072682 A1	12/03/2015
		US 2015-0087291 A1	26/03/2015
		US 2015-0092568 A1	02/04/2015
		US 2015-0133077 A1	14/05/2015
		US 2015-0163366 A1	11/06/2015
		US 2015-0163661 A1	11/06/2015
		US 2015-0244676 A1	27/08/2015
		US 2015-0256684 A1	10/09/2015
		US 2015-0312745 A1	29/10/2015
		US 2015-0358843 A1	10/12/2015
		US 2015-0373206 A1	24/12/2015
		US 2016-0036779 A1	04/02/2016
		US 2016-0037378 A1	04/02/2016
		US 2016-0050555 A1	18/02/2016
		US 2016-0100047 A1	07/04/2016
		US 2016-0105792 A1	14/04/2016
		US 2016-0134990 A1	12/05/2016
		US 2016-0191715 A1	30/06/2016
		US 2016-0205546 A1	14/07/2016
		US 2016-0255192 A1	01/09/2016
		US 2016-0261970 A1	08/09/2016
		US 2016-0323731 A1	03/11/2016
		US 2016-0323766 A1	03/11/2016
		US 2017-0026823 A1	26/01/2017
		US 8275357 B1	25/09/2012
		US 8325614 B2	04/12/2012
		US 8346214 B2	01/01/2013
		US 8391161 B1	05/03/2013
		US 8478238 B2	02/07/2013

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/US2017/069054

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
		US 8498615 B2	30/07/2013
		US 8531972 B2	10/09/2013
		US 8565101 B2	22/10/2013
		US 8626164 B2	07/01/2014
		US 8634407 B2	21/01/2014
		US 8725140 B2	13/05/2014
		US 8730820 B2	20/05/2014
		US 8730823 B2	20/05/2014
		US 8745184 B1	03/06/2014
		US 8767630 B1	01/07/2014
		US 8818331 B2	26/08/2014
		US 8837370 B2	16/09/2014
		US 8867575 B2	21/10/2014
		US 8868042 B2	21/10/2014
		US 8897146 B2	25/11/2014
		US 8897776 B2	25/11/2014
		US 8917611 B2	23/12/2014
		US 8937910 B2	20/01/2015
		US 8942181 B2	27/01/2015
		US 8958773 B2	17/02/2015
		US 8965332 B2	24/02/2015
		US 9084088 B2	14/07/2015
		US 9094538 B2	28/07/2015
		US 9100851 B2	04/08/2015
		US 9106768 B2	11/08/2015
		US 9119131 B2	25/08/2015
		US 9161248 B2	13/10/2015
		US 9166950 B2	20/10/2015
		US 9167471 B2	20/10/2015
		US 9179295 B2	03/11/2015
		US 9220025 B2	22/12/2015
		US 9226151 B2	29/12/2015
		US 9247415 B2	26/01/2016
		US 9288337 B2	15/03/2016
		US 9307397 B2	05/04/2016
		US 9338581 B2	10/05/2016
		US 9338636 B2	10/05/2016
		US 9398169 B2	19/07/2016
		US 9398172 B2	19/07/2016
		US 9462453 B2	04/10/2016
		US 9565552 B2	07/02/2017
		US 9699646 B2	04/07/2017
		US 9756014 B2	05/09/2017
		WO 2006-118742 A2	09/11/2006
		WO 2006-118742 A3	28/12/2006
		WO 2011-084945 A1	14/07/2011
		WO 2012-177665 A1	27/12/2012
		WO 2013-085852 A1	13/06/2013
		WO 2013-142615 A1	26/09/2013
		WO 2014-062384 A1	24/04/2014

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/US2017/069054

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2014-0189339 A1	03/07/2014	WO 2014-105995 A1	03/07/2014
		WO 2014-151711 A1	25/09/2014
		CN 103430185 A	04/12/2013
		CN 103430185 B	26/10/2016
		EP 2689368 A1	29/01/2014
		US 2012-0246641 A1	27/09/2012
		US 8713567 B2	29/04/2014
		US 9189247 B2	17/11/2015
		WO 2012-128681 A1	27/09/2012
		ZA 201304000 B	27/08/2014
US 2013-0046981 A1	21/02/2013	US 9203617 B2	01/12/2015
US 2012-0042215 A1	16/02/2012	JP 2012-523024 A	27/09/2012
		JP 5427245 B2	26/02/2014
		US 2012-0042215 A1	16/02/2012
		US 8782469 B2	15/07/2014
		WO 2011-027382 A1	10/03/2011