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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
23 January 2020 (23.01.2020)

WIPO | PCT

(10) International Publication Number
WO 2020/016480 A1

(51) International Patent Classification:
G06F 8/65 (2018.01) *H04L 9/06* (2006.01)
G06F 11/30 (2006.01) *H04L 9/08* (2006.01)
G06F 21/57 (2013.01) *H04L 9/32* (2006.01)
G06F 21/64 (2013.01) *H04L 29/06* (2006.01)

(21) International Application Number:
PCT/FI2019/050529

(22) International Filing Date:
05 July 2019 (05.07.2019)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
18183586.9 16 July 2018 (16.07.2018) EP

(71) Applicant: **NOKIA TECHNOLOGIES OY** [FI/FI]; Karakaari 7, 02610 Espoo (FI).

(72) Inventors: **OLIVER, Ian Justin**; Hitåvägen 153, 01150 Söderkulla (FI). **LIMONTA MARQUEZ, Gabrial Claret**; Rantaharju 10 B 42, 02230 Espoo (FI). **ORMISKANGAS VIGMOSTAD, Borger**; Itämerenkatu 18 A 3, 00180 Helsinki (FI). **KALLIOLA, Aapo**; Pyynikintie 4-8 A 10, 00710 Helsinki (FI).

(74) Agent: **NOKIA TECHNOLOGIES OY** et al.; Ari Aarnio, IPR Department, Karakaari 7, 02610 Espoo (FI).

(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, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM,

(54) Title: ELECTRONIC DEVICE UPDATE MANAGEMENT

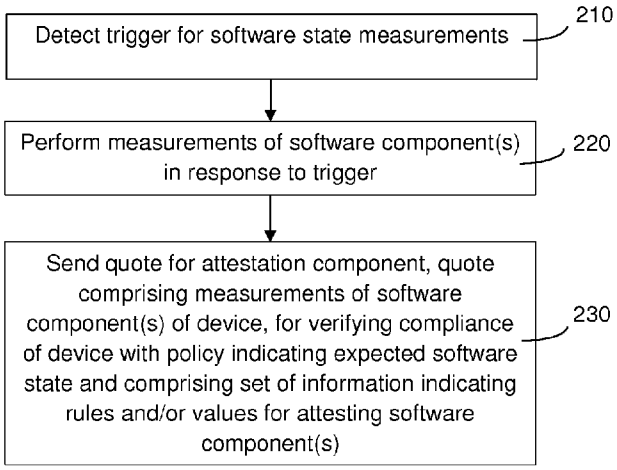

Fig. 2

(57) Abstract: According to an example aspect, there is provided a method comprising: receiving a policy for a device quoted for software state measurements, wherein the policy comprises a set of information indicating rules and/or values for attesting one or more software components, receiving a quote of the device, the quote comprising measurements of one or more software components of the device, verifying compliance of the device with the policy on the basis of the received measurements of the one or more software components, and sending a quote event for the device for a distributed attestation database.

WO 2020/016480 A1

[Continued on next page]

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))*

ELECTRONIC DEVICE UPDATE MANAGEMENT

FIELD

[0001] Various example embodiments relate to managing updating of electronic devices.

BACKGROUND

[0002] Devices are shipped with a set of firmware and software. After some time, they need to be updated by means of software updates or patches. These updates may occur regularly. They can include firmware, software and application updates. Regular updates keep the devices protected against vulnerabilities in software.

[0003] Outdated software may contain vulnerabilities that are exploitable by attackers. Therefore, it is important to keep devices up to date with the latest firmware or software which fixes those vulnerabilities. Any device in a network running an old version of software, or one specifically known to be vulnerable, is a threat to the whole system.

[0004] Requirements for device updating are changing. In cloud environments, such as the Telco Cloud, there are machines running the virtual workload, as well as different hardware connected to the cloud, such as base stations. With the increasing amount of Internet of Things (IoT) devices that can be connected to a cloud, it is important to facilitate control over the software running in these devices and the update process.

SUMMARY

[0005] The invention is defined by the features of the independent claims. Some specific embodiments are defined in the dependent claims.

[0006] According to a first aspect, there is provided a method, comprising: receiving, by an attestation component, a policy for a device quoted for software state measurements, wherein the policy comprises a set of information indicating rules and/or values for attesting one or more software components, receiving a quote of the device, the quote comprising measurements of one or more software components of the device, verifying compliance of the device with the policy on the basis of the received

measurements of the one or more software components, and sending a quote event for the device for an attestation database.

[0007] According to a second aspect, there is provided a method, comprising: detecting, by a device for attestation, a trigger for software state measurements, performing measurements of one or more software components in response to the trigger, and sending a quote to an attestation component, the quote comprising measurements of the one or more software components of the device, for verifying compliance of the device with a policy indicating expected software state and comprising a set of information indicating rules and/or values for attesting the one or more software components.

[0008] According to a third aspect, there is provided an apparatus, comprising at least one processor and at least one memory including computer program code, the at least one memory and the computer program code being configured to, with the at least one processing core, cause the apparatus at least to: receive a policy for a device quoted for software state measurements, wherein the policy comprises a set of information indicating rules and/or values for attesting one or more software components, receive a quote of the device, the quote comprising measurements of one or more software components of the device, verify compliance of the device with the policy on the basis of the received measurements of the one or more software components, and send a quote event for the device for an attestation database.

[0009] According to a fourth aspect, there is provided an apparatus comprising at least one processor and at least one memory including computer program code, the at least one memory and the computer program code being configured to, with the at least one processing core, to cause the apparatus at least to: detect a trigger for software state measurements, perform measurements of one or more software components in response to the trigger, and send a quote to an attestation component, the quote comprising measurements of the one or more software components of the device, for verifying compliance of the device with a policy indicating expected software state and comprising a set of information indicating rules and/or values for attesting the one or more software components.

[0010] According to a fifth aspect, there is provided a computer program product, a computer readable medium, or a non-transitory computer readable medium comprising

program instructions for causing an apparatus to perform at least the following: receive a policy for a device quoted for software state measurements, wherein the policy comprises a set of information indicating rules and/or values for attesting one or more software components, receive a quote of the device, the quote comprising measurements of one or more software components of the device, verify compliance of the device with the policy on the basis of the received measurements of the one or more software components, and send a quote event for the device for an attestation database.

[0011] According to a sixth aspect, there is provided a computer program product, a computer readable medium, or a non-transitory computer readable medium comprising program instructions for causing an apparatus to perform at least the following: detect a trigger for software state measurements, perform measurements of one or more software components in response to the trigger, and send a quote to an attestation component, the quote comprising measurements of the one or more software components of the device, for verifying compliance of the device with a policy indicating expected software state and comprising a set of information indicating rules and/or values for attesting the one or more software components.

[0012] An embodiment according to any one of the aspects can comprise triggering an update from a patch system to the device in response to detecting an update to be available for the device, and verifying compliance of the device with the policy on the basis of a subsequent quote after the update to the device.

[0013] An embodiment according to any one of the aspects can comprise sending an update event associated with the device for the attestation database in response to performing the update to the device.

[0014] In an embodiment according to any one of the aspects, the measurements are hashed in the quote response.

[0015] In an embodiment according to any one of the aspects, the attestation database is a distributed ledger and/or the attestation component is provided by an agent of a distributed attestation network. The agent may be a blockchain node, for example.

[0016] In an embodiment according to any one of the aspects, the agent is an event listener.

[0017] An embodiment according to any one of the aspects can comprise detecting events added in the attestation database indicative of one or more of a policy definition, a policy update, a successful quote, an unsuccessful quote, a device startup, a shutdown, a device reboot, a device update, a device attestation, an update patch release, and an update patch revocation.

[0018] An embodiment according to any one of the aspects can comprise reviewing events added in the attestation database after last successful quote in response to non-compliance of the device with the policy, and detecting a need for update and/or an update patch for the device on the basis of the review of events. The events may be detected and/or reviewed by an event listener.

[0019] An embodiment according to any one of the aspects can comprise sending a quote request to the device, and receiving the quote in a quote response from the device.

[0020] An embodiment according to any one of the aspects can comprise receiving a key for attestation, and verifying a signature in the quote on the basis of the key for attestation.

[0021] An embodiment according to any one of the aspects can comprise causing tagging of the device as not-trusted in response to at least one of failing validation of a signature received from the device, one or more values of the received measurements failing to match associated values in the policy, and conflict between additional information in the received quote and device state stored in the attestation database. Thus, an alert may be sent to a system administrator or a software update from a patch system to the device may be triggered in response to detecting non-compliance with the policy and availability of an update for the device.

[0022] In an embodiment according to any one of the aspects, the policy and the quote request indicate one or more platform configuration registers to be quoted, and the quote comprises measurements for the platform configuration registers.

[0023] In an embodiment according to any one of the aspects, information is obtained or measurements are performed also on hardware and/or configuration data of the device being attested. Information/measurements on the hardware and/or configuration data may be included in the quote and compliance verification may comprise verifying the

received information/measurements on the hardware and/or configuration against associated rule(s) and/or values defined by the policy. For example, the quote and/or policy may comprise one or more of measurements or information on firmware, boot code, device microcode, boot records, disk partitions, trusted platform module measurements, configuration strings, operating system components, and kernel(s).

BRIEF DESCRIPTION OF THE DRAWINGS

[0024] Some example embodiments will now be described with reference to the accompanying drawings.

[0025] FIGURES 1 an 2 illustrate methods in accordance with at least some embodiments;

[0026] FIGURES 3 and 4 illustrate example systems in accordance with at least some embodiments;

[0027] FIGURE 5 illustrates introduction of a device in accordance with at least some embodiments;

[0028] FIGURES 6 and 7 illustrates methods in accordance with at least some embodiments;

[0029] FIGURE 8 illustrates a signalling chart in accordance with at least some embodiments;

[0030] FIGURE 9 illustrates attestation functions in accordance with at least some embodiments;

[0031] FIGURE 10 illustrates a software update example in accordance with at least some embodiments; and

[0032] FIGURE 11 illustrates an apparatus in accordance with at least some embodiments.

EMBODIMENTS

[0033] There is now provided improved control for software updates, to one or more electronic devices, referred below to as (attested) devices or machines, by applying an

attestation procedure. The attestation procedure may be at least controlled by at least one attestation component, which may be provided by an agent of a distributed attestation network or an attestation server. The attestation component may be configured to keep track of device software state and maintain device information in a centralized or distributed attestation database. The attestation component may obtain measurements of software component(s) of device(s) it is monitoring and determines if they are trusted or not. It is to be noted that, for conciseness, the term "software" in the present disclosure also covers firmware, and the software update may be a firmware update. The software measurements refer generally to obtaining information of the software state of the attested device, in some embodiments Trusted Platform Module (TPM) measurements. The process of obtaining the software component measurements of a device is referred to as quoting. A quote may comprise or indicate results of the software component measurements, which is for simplicity herein referred to simple as the quote comprising the measurements. The quote may comprise a value or a set of values for one or more components, such as measurements, states or other (supplementary) data. According to the attestation procedure, a device may be quoted to perform measurements on software component(s) in the device and on the basis of received measurements software state of the device and a need for software update may be defined.

[0034] FIGURE 1 illustrates a method according to some embodiments. The method may be applied for software component attestation of one or more (attested) devices which in some embodiments are nodes of a distributed system, and may be carried out by the attestation component. The method may be implemented in a communications apparatus or device, such as a distributed network node or a server, e.g. a network management device, a network operator device, an IT administrator device, or a device management system (DMS) node, for example.

[0035] Block 110 comprises receiving a policy for a device quoted for software state measurements. The policy refers generally to information indicating at least expected or required state of software for a single or a set of devices. The policy may comprise a set of information indicating rules and/or values for attesting and verifying compliance of a device. The policy may define expected measurement values (associated with the rules) for a device running a certain software, such as a software version number, etc.

[0036] Block 120 comprises receiving a quote of the device, the quote comprising measurements of one or more software components of the device. The measurements reflect the software state or configuration of the machine. The quote may be received directly from the device or via one or more intermediate entities, such as an attestation database.

[0037] Block 130 comprises verifying compliance of the device with the policy on the basis of the received measurements of the software component(s). Block 140 comprises sending a quote event for the device for an attestation database. Depending on the embodiment, this may comprise adding the quote event in the database or sending to one or more entities for subsequent storage in the database, e.g. providing a blockchain transaction comprising the quote event for validation and subsequent publication in the blockchain. In some embodiments, the quote event is generated on the basis of the verification 130, to record and indicate the result of the attestation. The quote even may indicate successful or unsuccessful compliance verification. In some embodiments, a quote event comprises the quote, and may be used later to review the software state of the device.

[0038] In response to detecting an update to be available for the device, an update from a patch system to the device may be triggered. A check for an update may be carried out in response to the device failing to comply with the policy in block 130. An update event associated with the device may be caused to be added in the attestation database in response to performing the update to the device. The device may (again) be quoted and compliance verified after the update. It is to be noted that the features in blocks 110 and 120 may be carried out in different order depending on the implementation, i.e. the policy may be received for the device for which a quote has already been received or for which the quote will be requested.

[0039] FIGURE 2 illustrates a method according to some embodiments. The method may be applied for providing state information of software component(s) of one or more devices being attested, which in some embodiments are nodes of a distributed system. The method may be implemented in an apparatus or device capable of participating in the attestation procedure, such as a user device, a machine-to-machine (M2M) or an IoT device, a network or system management device, a network node, such as a base station, for example.

[0040] Block 210 comprises detecting a trigger for quote for software state measurements. The apparatus performing the method of Figure 1 may cause the trigger 200 by request the device to provide the quote 210, or the device may be triggered by another apparatus or an internal trigger.

[0041] Block 220 comprises performing measurements of one or more software components in response to the trigger. Block 230 comprises sending a quote to an attestation component, the quote comprising measurements of the software component(s) of the device, for verifying compliance of the device with the policy indicating expected software state. The quote 230 may be received in block 120 by the apparatus implementing the attestation component and verifying the compliance of the device with the policy.

[0042] In some embodiments, the attestation procedure is arranged by a centralized architecture by an attestation server. FIGURE 3 illustrates an example of such centralized system comprising the (set of) device(s) or machine(s) 300 to be attested, a software update provision system, herein referred to as a patch system or a patch management component 302, an administrator component 304, an attestation server 306, and an (attestation) database 308. FIGURE 3 illustrates how the entities or components may communicate with each other. In this architecture the attestation is carried out through the attestation server 306, which may be configured to carry out the method illustrated in FIGURE 1. The attestation server may have the responsibility of triggering the software updates on the machines as soon as it detects outdated software. The server may be configured to send a quote request to the device, and receive the quote in a quote response from the device. No event listener is necessary on this kind of architecture.

[0043] In some embodiments, the attestation procedure is provided by a distributed system. The attestation database may be a distributed ledger and the attestation procedure is performed by an agent of a distributed attestation network. The purpose of the distributed ledger is to record and enforce verifiable transactions and order of events, in the present embodiments related to the attestation procedure. A distributed ledger can be considered a general database synchronized between many parties, which, at successive time instances, comprises successive states of consensus about a particular matter, e.g., on device quote or update event. Such states may be reached by many parties, e.g., nodes in a peer-to-peer (P2P) network, following a decentralized consensus algorithm. This may

provide an agreement between the nodes without assuming trust between them. The distributed ledger enables a decentralized, auditable, and tamper-proof registry of privacy permissions for managing device updates. Attestation related information may be published to the distributed ledger from any location if the publisher is appropriately equipped for the publication, for instance has appropriate keys to sign the submission. There is no single point of failure that an attacker could attack to stop the process.

[0044] The distributed network may be a private or public distributed network, and may be built on currently available blockchain system, for example. The nodes may be physical devices connectable locally by at least one a wireless network, such as a Wi-Fi network and/or a cellular network. However, it is to be appreciated that multiple underlying networks and/or different connections may be applied to connect nodes of the distributed network. The nodes in the distributed network may comprise corporate, authority, and/or user devices, such as a server, a desktop/tablet/laptop computer, a smartphone, an IoT device, a domestic appliance, building automation device, an entertainment system device, a vehicle unit, a wearable, or any other suitable electronic device. The system may comprise an administrator or management node, a hub, relay or other kind of intermediate device for connecting a node to further networks or services, such as another distributed or centralized computing system or a cloud service. The nodes are mutually addressable in a suitable way, for example, they may be connectable to an internet protocol, IP, network.

[0045] FIGURE 4 illustrates a simplified example of components for such distributed system. The agent of the distributed attestation network may be provided as or by an event listener 400 configured to perform at least some of the attestation procedure, such as the blocks illustrated in connection with FIGURE 1. The event listener 400 may be configured to trigger actions based on the events that occur in the distributed system. For example, the event listener 400 may have the responsibility of triggering software updates when it detects on the basis of the received quote and policy that the quoted device is in an outdated state. By storing the results of attestation into a distributed database 402, such as a blockchain ledger, agents can determine what software is running on each device.

[0046] An attestation system may in some embodiments comprise both the attestation server 306 and the event listener 400. Alternatively, the system can be adapted

to work with only one of these components. In a situation where the event listener is not implemented, the attestation server can perform at least some of the features of the event listener illustrated below. Similarly, if attestation is done in a distributed manner and a centralized attestation server is not implemented, the event tracking component can carry out verification functionality instead of the attestation server.

[0047] The attestation database 308, 402 can be implemented in a centralized manner or in a distributed manner. The centralized or distributed attestation database, 308 or 402, respectively, stores the policies, and may comprise further attestation related information for the attestation system to operate properly. The information stored in the database may comprise at least some of machine identities, policy definitions, global event logs, machine event logs, attestation event logs and patch releases. Examples of events that can be detected and added to the database 308, 402 include (but are not limited to):

- Policy definition
- Policy update
- Successful quote
- Unsuccessful quote
- Device startup
- Device shutdown
- Device reboot
- Device update
- Device attestation
- Patch release
- Patch revocation

[0048] The event listener 400 can keep track of one or more of these events and trigger actions based on the events it detects.

[0049] For example, a NoSQL database, such as a MongoDB may be applied for centralized implementations. In some embodiments, for a distributed implementation, a blockchain, such as Ethereum-based implementation is applied. If a blockchain is used, we also obtain desirable further properties in the system, such as a tamper-proof record of all events that have occurred in the system. Additionally, it can increase accountability in the system, since we can verify the author of each transaction.

[0050] In both centralized and distributed attestation systems, the administrator 304 defines the policy, which may define a set of requirements or policies that the attested device 300 must follow. These policies may comprise a set of expected values for software measurements. The attestation component responsible for the verification, such as the attestation server 306 or the event listener 400, may in block 130 compare measurement values reported by the device against the policy applying for the device. If the device reports values that differ from the expected values, the device can be labelled as outdated and a software update should be triggered. Furthermore, once the software patch is applied, the device will be measured again, and the attestation system can verify that the patch was successfully applied, since the new measurements should satisfy the policy.

[0051] The system administrator 304 can communicate with the attestation server 306 to create new policies, update existing policies and/or assign policies to devices. If no attestation server is implemented, the system administrator can communicate directly with the database 308, 402, to perform these actions. Additionally, the system administrator can communicate with the patch system 302 to manage software patches for different devices. The system administrator 304 may be configured to receive security alerts from the attestation component when the attestation was unsuccessful due to a reason different than outdated software. The administrator may be configured to output an alert to a user of the respective device and/or initiate an action to remedy the problems detected by the attestation.

[0052] The patch system 302 may be configured to maintain and provide the available updates for devices. The patch system 302 may maintain information or a list of available patches as well as on mapping between target systems and released patches for that system. The patch system can be implemented in different ways, for example as part of device management system (DMS). Some examples of available implementations include but are not limited to: a traditional data store in a centralized manner or a distributed data store. In the distributed data store patches can be added e.g. to a blockchain along with metadata for each patch, and it anyone may be permitted to retrieve updates from the chain.

[0053] In some embodiments the (attested) devices 300 are provided with a Trusted Platform Module (TPM). The TPM may be configured to generate a hash that summarizes

the hardware and/or software configuration of that platform, as well as other related metadata, such as reboot count. The hash may be sent in the quote 120, 230 and compared in block 130 to an expected hash value, to verify that particular software is running on a device.

[0054] A TPM chip can store cryptographic keys, certificates and confidential data. It may store two unique keypairs: an endorsement key (EK) and an attestation key (AK). Additionally, it may comprise a set of platform configuration registers (PCRs) which store measurements, in the form of hashes, of the software components of a device. A TPM can be asked in block 210 to provide the quote for a set of PCRs, wherein the set may be defined by the policy. The quote may be a hash over the stored values of the selected PCRs. The TPM will then return the quote for the requested PCRs, a cryptographic signature of the quote (signed by the AK) and other information, including a timestamp and a reboot count.

[0055] If a device has a TPM, the policy or an entry of the policy can define a mapping between a PCR identifier and an expected value. The administrator 304 can define policies for a device to be considered in a trusted state. When attestation is done the measurements are checked against the expected PCR specific values in the policy.

[0056] It is to be noted that the expected values in a policy are not limited to PCR values from a TPM. Policies can be defined basically for any kind of measurement that can be taken from the attested device 300. Furthermore, a policy can potentially become quite complex. We can think of simple and compound policies. A simple policy is a set of expected values, whereas a compound policy comprises a set of simple policies. Therefore, very specific policies can be generated as building blocks for more complex policies. For example: if there are two devices running the same BIOS version but different software on top, a simple policy can be defined for the BIOS and then two different policies for the other software on top. Then two (or more) compound policies may be generated, which share the BIOS policy.

[0057] Various advantages are available over known software updating systems applying centralized or distributed architectures, such as blockchain-based schemes for distributing software. By applying the presently disclosed attestation procedure, the devices do not need to detect that they need an update or periodically check if their system

is up-to-date against a verifier node. The devices do not need to traverse a blockchain to find available updates for their software, but instead an agent can determine, based on the attestation results, which devices need updates. Instead of requiring having a single network node, such as a base station to push software updates to the devices, the present procedure facilitates different policies that, mixed with the attestation results, can be used to automatically determine which device needs an update. Furthermore, obtaining attestation results after applying the patch allows to verify that the device has been updated successfully. Hence, in addition to getting the updates to the target devices, the present embodiments also verify that they have indeed been patched.

[0058] Next, the attestation procedure or protocol according to at least some embodiments is illustrated in further detail. The protocol may comprise following stages or main steps: **1)** introduction, **2)** quoting, and **3)** action. The introduction stage may be carried out once when the device starts up, and the quoting and action steps may be repeated.

[0059] FIGURE 5 illustrates **1)** introduction of a device for the attestation procedure. The device 300 first registers its identity to the attestation server 306 to be added to a set of known devices in the network. The identity of a device may comprise device's IP address, TPM's EK, TPM's AK, and/or other relevant metadata, depending on the implementation. For example, such other metadata may comprise Openstack ID, device name, kernel version and/or other system information.

[0060] The attestation server 306, upon receiving the request from the device 300 and preferably authenticating the device, may register the device into the attestation database 308. The database may store this device under a global unique ID and return this ID to the attestation server. The attestation server can now add the device to its list of known devices and return the assigned ID to the device. It is to be noted that the ID on the database will depend on the database implementation used. For example, if a Mongo database is used, this would be a BSON Object ID.

[0061] In an alternative embodiment in the distributed system, instead of applying the attestation server 306, the device 300 may register directly to the distributed database 402 by generating a transaction which is validated by at least some nodes of the distributed network. Upon validation of the transaction, the device is added to the distributed network.

If a blockchain is used, the assigned ID may be an address of a (smart) contract on the chain that comprises the device information. Note that this address is a globally unique identifier, which can serve as the unique identity of the device.

[0062] FIGURE 6 illustrates an example method for **2)** quoting of a device, such as the device 300. The method may be carried out by an apparatus implementing the method of FIGURE 1, such as an apparatus comprising the attestation server 306 or at least partly by the event listener component. This stage may comprise three steps or sub-stages: measuring 2a), validation 2b), and verification 2c). The measuring step should be carried out first, whereas validation and verification can be done in any order.

[0063] In the present examples references are made to attesting a device with a TPM. However, as already indicated, the attestation procedure may be applied for any device capable of providing measurements that can be used for verifying 130 compliance in a policy, as disclosed in connection with FIGURES 1 and 2.

[0064] For example, the quoting process can be triggered 210, 600 by at least one of: the device triggering quoting in a periodic manner or after an important event (e.g. a reboot), another device in the system can ask the device to quote itself, and the attestation server 306 requesting the device to quote itself. When the quoting process is triggered by an external party, the quoting process can be preceded by an authentication step. The attested device can request the external party to authenticate itself and vice versa. This authentication can be done by using certificates. If certificates are used, the rest of the communications for the quoting process can be secured.

[0065] Next, the sub-stages of the quoting stage **2)** are explained:

[0066] **(2a)** Measuring: the policy that the device should satisfy is obtained 602. This policy may comprise a set of PCRs and expected values. In the present example, a quote request is sent 604 to the device. This quote request may indicate the PCRs for which a quote is requested. The device returns the quote for those PCRs, a signature for the quote and other related metadata. The quote may be stored by adding a quote event to the database 308, 402.

[0067] **(2b)** Validation: the AK or another applied cryptographic key, such as a key derived from, sealed by or wrapped by the AK, is obtained 606 from the database 308, 402

and the validity of the cryptographic signature is verified 608. This may comprise verifying that the quote was signed with an AK of the device's TPM. In an alternative embodiment, the TPM of the device can store an externally provided key, which has the device's AK in the signature chain. This externally provided key can be used to sign quotes. This sub-stage is optional. If the signature is valid, the verification sub-stage 2c is entered. If the signature is not valid, the device is tagged as not trusted, and the action stage 3) is entered 618.

[0068] **(2c)** Verification: the attestation server compares 610 the quote to the expected values of the policy. If the quote matches the expected values, check(s) are performed 612, 614 based on the additional information in the quote. For example, block 612 may comprise checking if the device has been rebooted, but it will be appreciated that various other and more complex check routines may be applied.

[0069] If there is any information in the quote predefined to cause suspicion, a security alert may be generated for the administrator and the device may be tagged 616 as not trusted. For example, the device may be tagged as untrusted if the reboot count has increased and the device has no reboot events. If everything is in order on the basis of the check 612, 614, the Action step 3) is entered 618.

[0070] If the quote does not match the expected values, the device is tagged 616 as not trusted and the action stage 3) is entered 618.

[0071] FIGURE 7 illustrates an example method for the Action stage **3)** for the attestation procedure. The method may be carried out by an apparatus implementing the method of FIGURE 1, such as an apparatus comprising the attestation server 306 or the event listener component. The actions are based on the results 702, 704, 706 from the previous step 2) on the attestation protocol, the apparatus will decide which actions to take and may first check 702 if the device is tagged as trusted.

[0072] If the device is not trusted due to a validation error in block 704, an alert may be sent 708 to the administrator 304. The device is thus maintained as in a not trusted condition, which may cause that no more workload is run on that device until it is back to a trusted state. For example, the validation error from checking the TPM quote may comprise one or more of: attested part of the quote is incorrect, incorrect magic value, and incorrect firmware value.

[0073] If the device is not trusted due to the quote not matching 706 the expected values, the patch management system is contacted 710 to check 712 if there is an update

available for that device. If there is an update available, the attestation server triggers 714 the device update and the device stays in an untrusted state until it has been updated and successfully re-quoted.

[0074] If there is no update available for that device, the procedure assumes the software in the device is incorrect, so it keeps the device in an untrusted state and generates a security alert for the administrator to manually take actions. If the device is trusted, no alerts are raised and no updates are triggered.

[0075] In all cases, a quote event may be stored 716 for this device in the database. This event may indicate the result of the attestation. The actions during the stage 3) may be dependent on if the failure is over a period in time or just between two quotes.

[0076] As mentioned before, the attestation functionality can be distributed among the different devices in the network, whereby the attestation server 306 is not required. In this embodiment, a device of a distributed network can request another device of the distributed network to measure itself. The attested device may thus send the quote for the attestation component in block 230 first for publication in a distributed database, which in this embodiment may act as an intermediate storage. The attestation component may receive the quote of the device in block 120 from the distributed database. Thus, the method of FIGURE 1 and further the validation 2b) and verification 2c) of the attestation protocol may be carried out by the event listener component 400 operating as the attestation component.

[0077] In some embodiments, as already indicated, the attestation database 402 is a blockchain–based database (or blockchain ledger) and at least some of the features of the attestation procedure are performed between blockchain nodes by applying blockchain transactions.

[0078] Each blockchain node may have their own copy of the ledger which is in some embodiments permission-controlled, so participants see only appropriate transactions. Information provided in a blockchain transaction, such as the presently disclosed attestation related events, may be stored in the blockchain ledger in response to validation of the respective blockchain transaction. A node establishing the next block may be known as a miner node. A miner node may compile a set of transactions, which it

receives from the broadcasts, for the next block, and search for a proof-of-work code that covers all the transactions in the set of transactions for the next block. For example, the proof-of-work code may be a numerical value, with which the contents of the next block, that is, the set of transactions, hashes to a value that is less than a threshold. Once a miner discovers the proof-of-work, it can publish the block, which other nodes of the system will then add to the blockchain as the new most recent established block with proof of work code field. In case the miner node discovers a proof-of-work based on an incomplete set of transactions, for example if some transactions did not reach the miner node, other nodes in the network will not accept the block into the blockchain, and it will be excluded from a consensus version of the blockchain in the system.

[0079] Although discussed above in terms of proof-of-work, in some embodiments a proof-of-stake may be used instead of, or additionally to, a proof-of-work. In a proof-of-stake based system, a new block may be accepted once a predetermined fraction of resources are proven as owned by nodes ready to accept the new block version.

[0080] The event listener component 400 may be an agent implemented in a blockchain network node and is configured to traverse blockchain events and trigger actions based on the events in the system. Attestation related events, such as the events indicated above, are stored to the blockchain to make them available for the event listener 400. In particular, the event listener 400 may be configured to detect and trigger actions on the basis of events provided by the attested devices 300, the administrator component(s) 304, and the patch system(s) 302, which may be implemented in apparatuses operating as blockchain full or light nodes.

[0081] FIGURE 8 shows an example message sequence chart in which attestation results trigger an update on a device, in this example machine m. In this example the attestation is distributed, e.g. performed in a blockchain-based system, and the system comprises the event listener 400.

[0082] The messages in the flowchart are as follows:

1. The administrator defines a new policy po with a set of expected values.
2. A policy creation event is provided for the attestation database, such as the blockchain, as a global event.

3. The machine m starts up and provides its identity id to the attestation database.
4. A machine start event is provided for the attestation database.
5. A quote is triggered in the machine m (e.g. by another device requesting a quote or a periodic quote), and the machine m provides the quote for publication on the attestation database.
6. The event listener checks the published quote against the policy po in the attestation database. The quote now satisfies the policy, whereby a successful quote event is provided to the attestation database.
7. The administrator publishes a patch pa applying for machine m, and a patch release event is provided for the attestation database.
8. The administrator updates the existing policy for m with new expected values, and a policy update event is provided for the attestation database.
9. The event listener triggers the quoting process on all devices that follow the policy po. The machine m quotes itself and provides the quote to the attestation database.
10. The event listener checks the quote. This time, the quote does not match the expected values of the policy.
11. The event listener requests all the events between last successful quote and the current time.
12. On the basis of the list of events, the event listener detects that there was a software patch released for the machine m between last successful quote and the current time. The event listener provides an unsuccessful quote event to the attestation database.
13. The event listener communicates with the patch system to trigger an update for machine m.
14. The patch management system sends the update to m and the update is applied.
15. A machine update event is provided for the attestation database.
16. A machine reboot event is provided for the attestation database.
17. The machine quotes itself again because of the reboot.
18. The event listener checks the quote against the policy. The quote now satisfies the updated policy.
19. The event listener provides a successful quote event to the attestation database.

[0083] It is to be appreciated that at least some of the features illustrated in connection with FIGURE 8 bundling together a plurality of embodiments for conciseness. For example, one or more of the quote and/or policy related actions may be applied outside the disclosed example and other features of FIGURE 8 and modified in various ways. For example, it is to be noted that the quotes and/or events to be reviewed (11-12.) after failed verification do not have to be past and current time, but can be reviewed over any desired period of time, such as over two set times in the past. Typically the latest and preceding quote(s) are reviewed but it is to be appreciated that basically any quotes may be selected for review, e.g. for historical or forensic reasons.

[0084] The attestation process can be divided into different functionalities that can be implemented separately. FIGURE 9 shows an example of how the attestation responsibilities can be divided into independent components or functions that communicate with each other, as well as with the other components (previously described) in the system.

[0085] The illustrated functions comprise:

- Quoting: Provides a communication interface to obtain 120, 604 quotes from attested devices 300.
- Database: Provides a communication interface to the database 308, 402 and is applied to retrieve 110, 602 information for attestation.
- Monitoring: Provides a communication interface to interact with different systems and obtain information relevant for the attestation process, such as patch update information from the Patch Management System, e.g. the system 302.
- Decision maker: Makes decisions on what actions to take based on the information provided by the other components, and may perform at least some of the features in stages 2b), 2c), and 3. This component may decide whether a device is trusted or not and can trigger updates on outdated devices.
- Action: Takes instructions from the decision maker on what actions to do on the system depending on the attestation results. It can also communicate with an external security orchestration management component to perform actions in the system, to generate 708 security alerts for administrators, for example.

[0086] It is to be noted that two or more entities or components illustrated in above Figures may be implemented in a single apparatus. For example, the administrator

component 304 and the attestation server 306 and/or the event listener 400 may be implemented in a single apparatus. In the case of distributed architecture, as already indicated, a device 300 being attested may comprise the event listener 400 and at least some of the distributed database 402.

[0087] FIGURE 10 illustrates a simplified firmware update example for an attested device, showing a timeline of events from different points of view in the system. The device gets quoted twice. In the time between the first quote $Quote_1$ and the second quote $Quote_1$, the patch management system view indicates that the firmware of the device was updated. Also, the reboot event view indicates that a reboot has happened.

[0088] The attestation component, such as the attestation server 306 or the event listener 400, can be configured to compare the two quotes to verify that the changes that happened in the system are reflected in measurement changes. When comparing $quote_1$ and $quote_2$, the attestation component may detect that the reported measurements have changed. The information quoted by a TPM may include fields that can help with the verification process, such as a reboot count. The difference between the reboot count in $Quote_1$ and $Quote_2$ should equal the amount of reboot events visible for that device between $Quote_1$ and $Quote_2$. In this case, reboot count of the $Quote_2$ $quote_2$.rebootCount – reboot count of the $Quote_1$ $quote_1$.rebootCount should equal 1, since there is only one reboot event. If this is not the case, the attestation component can report this to the administrator, since it might indicate strange behaviour. Additionally, the attestation component can check that the update and reboot events happened in the correct order. Policies may be defined that indicate what should happen after an update. For example, if there is a firmware update, the device is updated and rebooted once. These policies would define rules that can be added to the attestation verification process.

[0089] In some embodiments, at least some of the above illustrated features are applied for managing software configuration of network nodes or functions, such as radio and/or core network elements. Some examples of such network nodes or functions include base stations, node Bs, core network user plane network functions (e.g. 5G User Plane Function UPF), and control plane functions (e.g. 5G Access and Mobility Management Function AMF and Session Management Function SMF).

[0090] It is to be noted that at least some of the above illustrated features may be applied in systems applying network virtualization. Hence, network functions or nodes illustrated above may be shared between two physically separate devices forming one operational entity. In general, virtual networking may involve a process of combining hardware and software network resources and network functionality into a single, software-based administrative entity, a virtual network. Network virtualization may involve platform virtualization, often combined with resource virtualization. Network virtualization may be categorized as external virtual networking which combines many networks, or parts of networks, into the server computer or the host computer. External network virtualization is targeted to optimized network sharing. Another category is internal virtual networking which provides network-like functionality to software containers on a single system. For example, instances of 5G network functions can be instantiated as virtual functions in a network function virtualization (NFV) architecture.

[0091] An electronic device comprising electronic circuitries may be an apparatus for realizing at least some embodiments. The apparatus may be or may be comprised in a computer, a laptop, a tablet computer, a cellular phone, a machine to machine (M2M) device, a wearable device, a base station, access point device, a network management or control device, a server, or any other apparatus provided with communication capability. In some embodiments, the apparatus carrying out at least some of the above-described functionalities is comprised in such a device, e.g. the apparatus may comprise circuitry, such as a chip, a chipset, a microcontroller, or a combination of circuitries for or in any one of the above-described devices.

[0092] As used in this application, the term "circuitry" may refer to one or more or all of the following:

(a) hardware-only circuit implementations (such as implementations in only analog and/or digital circuitry) and

(b) combinations of hardware circuits and software, such as (as applicable):

(i) a combination of analog and/or digital hardware circuit(s) with software/firmware and

(ii) any portions of hardware processor(s) with software (including digital signal processor(s)), software, and

memory(ies) that work together to cause an apparatus, such as a mobile phone or server, to perform various functions) and

(c) hardware circuit(s) and or processor(s), such as a microprocessor(s) or a portion of a microprocessor(s), that requires software (e.g., firmware) for operation, but the software may not be present when it is not needed for operation." This definition of circuitry applies to all uses of this term in this application, including in any claims. As a further example, as used in this application, the term circuitry also covers an implementation of merely a hardware circuit or processor (or multiple processors) or portion of a hardware circuit or processor and its (or their) accompanying software and/or firmware. The term circuitry also covers, for example and if applicable to the particular claim element, a baseband integrated circuit or processor integrated circuit for a mobile device or a similar integrated circuit in server, a cellular network device, or other computing or network device.

[0093] FIGURE 11 illustrates an example apparatus capable of supporting at least some embodiments. Illustrated is a device 100, which may comprise a communications device, in some embodiments arranged to operate as a node in a blockchain-based system. The device may be arranged to carry out the method of FIGURE 1 and the attestation server and/or event listener, or in some embodiments perform the method of FIGURE 2 and operate as the device being attested, and carry out at least some of the embodiments illustrated in connection with FIGURES 1 to 10. The device may include one or more controllers configured to carry out operations in accordance with at least some of the embodiments illustrated above.

[0094] Comprised in the device 100 is a processor 101, which may comprise, for example, a single- or multi-core processor wherein a single-core processor comprises one processing core and a multi-core processor comprises more than one processing core. The processor 101 may comprise more than one processor. The processor may comprise at least one application-specific integrated circuit, ASIC. The processor may comprise at least one field-programmable gate array, FPGA. The processor may be means for performing method steps in the device. The processor may be configured, at least in part by executing computer instructions, to perform action to perform at least some of the presently disclosed attestation related features.

[0095] The device 100 may comprise memory 102. The memory may comprise random-access memory and/or permanent memory. The memory may comprise at least

one RAM chip. The memory may comprise solid-state, magnetic, optical and/or holographic memory, for example. The memory may be at least in part accessible to the processor 101. The memory may be at least in part comprised in the processor 101. The memory 102 may be means for storing information. The memory may comprise computer program code comprising computer instructions that the processor is configured to execute. The memory, processor and computer program code may thus be configured to cause the device 100 to perform at least some of the presently disclosed features. The memory may be at least in part comprised in the processor. The memory may be at least in part external to the device 100 but accessible to the device. For example, control parameters affecting operations related to causing above-illustrated attestation related events and related information may be stored in one or more portions of the memory and used to control operation of the apparatus.

[0096] Further, the memory may comprise device-specific cryptographic information, such as secret and public key of the device 100. The transactions may be signed with a private key associated with the respective device or user. The apparatus may be provided with a public-private key pair at manufacture. The private key may be stored on the certifying device's secured memory portion, such that it cannot be tampered with and the private key cannot be stolen. Moreover, the secured portion may also hold the hashing and signing logic.

[0097] The device 100 may comprise a transmitter 103. The device may comprise a receiver 104. The transmitter and the receiver may be configured to transmit and receive, respectively, information in accordance with at least one wired or wireless, cellular or non-cellular standard. The transmitter may comprise more than one transmitter. The receiver may comprise more than one receiver. The transmitter and/or receiver may be configured to operate in accordance with global system for mobile communication, GSM, wideband code division multiple access, WCDMA, long term evolution, LTE, 5G or other cellular communications systems, wireless local area network, WLAN, and/or Ethernet standards, for example. The device 100 may comprise a short range communication, SRC, transceiver 105. The SRC transceiver may support at least one SRC technology, such as SRC, Bluetooth, Bluetooth Low Energy, or similar technologies. The cellular, other wide network technologies and short range communications may interact in the device in parallel or any other way.

[0098] The device 100 may comprise user interface, UI, 106. The UI may comprise at least one of a display, a keyboard, a touchscreen, a vibrator arranged to signal to a user by causing the device to vibrate, a speaker and a microphone. A user may be able to operate the device via the UI, for example to accept incoming telephone calls, to originate telephone calls or video calls, to browse the Internet, to manage digital files stored in the memory 102 or on a cloud accessible via the transmitter 103 and the receiver 104, or via the SRC transceiver 105, and/or to play games.

[0099] The device 100 may comprise or be arranged to accept a user identity module or other type of memory module 107. The user identity module may comprise, for example, a subscriber identity module, SIM, and/or a personal identification IC card installable in the device 100. The user identity module 107 may comprise information identifying a subscription of a user of device 100. The user identity module 107 may comprise cryptographic information usable to verify the identity of a user of device 100 and/or to facilitate encryption and decryption of communication effected via the device 100, such as private and/or public keys for creating and validating cryptographic signatures.

[00100] The processor 101 may be furnished with a transmitter arranged to output information from the processor, via electrical leads internal to the device 100, to other devices comprised in the device. Such a transmitter may comprise a serial bus transmitter arranged to, for example, output information via at least one electrical lead to memory 102 for storage therein. Alternatively to a serial bus, the transmitter may comprise a parallel bus transmitter. Likewise the processor may comprise a receiver arranged to receive information in the processor, via electrical leads internal to the device 100, from other devices comprised in the device 100. Such a receiver may comprise a serial bus receiver arranged to, for example, receive information via at least one electrical lead from the receiver 104 for processing in the processor. Alternatively to a serial bus, the receiver may comprise a parallel bus receiver.

[00101] The device 100 may comprise further devices not illustrated in Figure 11. For example, the device may comprise at least one digital camera or other user media recording device. Some devices may comprise a back-facing camera and a front-facing camera. The device may comprise a fingerprint sensor arranged to authenticate, at least in

part, a user of the device. In some embodiments, the device lacks at least one device described above. For example, some devices may lack the SRC transceiver 105 and/or the user identity module 107.

[00102] The processor 101, the memory 102, the transmitter 103, the receiver 104, the SRC transceiver 105, the UI 106 and/or the user identity module 107 may be interconnected by electrical leads internal to the device 100 in a multitude of different ways. For example, each of the aforementioned devices may be separately connected to a master bus internal to the device, to allow for the devices to exchange information. However, as the skilled person will appreciate, this is only one example and depending on the embodiment various ways of interconnecting at least two of the aforementioned devices may be selected.

[00103] It is to be understood that the application of the disclosed embodiments are not limited to the particular structures, process steps, or materials disclosed herein, but are extended to equivalents thereof as would be recognized by those ordinarily skilled in the relevant arts. It should also be understood that terminology employed herein is used for the purpose of describing particular embodiments only and is not intended to be limiting.

[00104] References throughout this specification to one embodiment or an embodiment means that a particular feature, structure, or characteristic described in connection with the embodiment is included in at least one embodiment of the present invention. Thus, appearances of the phrases “in one embodiment” or “in an embodiment” in various places throughout this specification are not necessarily all referring to the same embodiment. The skilled person will appreciate that above-illustrated embodiments may be combined in various ways. Embodiments illustrated in connection with Figures 2 to 6 may be taken in isolation or further combined together.

[00105] As used herein, a plurality of items, structural elements, compositional elements, and/or materials may be presented in a common list for convenience. However, these lists should be construed as though each member of the list is individually identified as a separate and unique member. Thus, no individual member of such list should be construed as a de facto equivalent of any other member of the same list solely based on their presentation in a common group without indications to the contrary. In addition, various embodiments and examples may be referred to herein along with alternatives for

the various components thereof. It is understood that such embodiments, examples, and alternatives are not to be construed as de facto equivalents of one another, but are to be considered as separate and autonomous representations.

[00106] Furthermore, the described features, structures, or characteristics may be combined in any suitable manner in one or more embodiments. In the preceding description, various operational and structural specific details are provided, to provide a thorough understanding of the embodiments. One skilled in the relevant art will recognize, however, that various embodiments can be practiced without one or more of the specific details, or with other methods, components, materials, etc. In other instances, well-known structures, materials, or operations are not shown or described in detail to avoid obscuring aspects of the invention.

[00107] While the forgoing examples are illustrative of the principles of the present invention in one or more particular applications, it will be apparent to those of ordinary skill in the art that numerous modifications in form, usage and details of implementation can be made without the exercise of inventive faculty, and without departing from the principles and concepts of the invention. Accordingly, it is not intended that the invention be limited, except as by the claims set forth below.

[00108] The verbs “to comprise” and “to include” are used in this document as open limitations that neither exclude nor require the existence of also un-recited features. The features recited in depending claims are mutually freely combinable unless otherwise explicitly stated. Furthermore, it is to be understood that the use of "a" or "an", that is, a singular form, throughout this document does not exclude a plurality.

INDUSTRIAL APPLICABILITY

At least some embodiments find industrial application in communications.

ACRONYMS LIST

AK	Attestation key
AMF	Access and mobility management function

ASIC	Application-specific integrated circuit
BSON	Binary JSON
DMS	Device management system
EK	Endorsement key
FPGA	Field-programmable gate array
GSM	Global system for mobile communication
IC	Integrated circuit
IoT	Internet of things
LTE	Long term evolution
M2M	Machine to machine
NB-IoT	Narrowband IoT
NFV	Network function virtualization
PCR	Platform configuration register
PoS	Proof-of-stake
PoW	Proof-of-work
SDN	Software defined network
SMF	Session management function
SRC	Short range communication
TPM	Trusted platform module
UI	User interface
UPF	User plane function
WCDMA	Wideband code division multiple access
WLAN	Wireless local area network

CLAIMS:

1. An apparatus comprising means for performing, comprising:
   - receiving a policy for a device quoted for software state measurements, wherein the policy comprises a set of information indicating rules and/or values for attesting one or more software components,
   - receiving a quote of the device, the quote comprising measurements of one or more software components of the device,
   - verifying compliance of the device with the policy on the basis of the received measurements of the one or more software components, and
   - sending a quote event for the device for an attestation database.

2. The apparatus of claim 1, wherein the means are further configured to perform:
   - in response to detecting an update to be available for the device, triggering an update from a patch system to the device, and
   - verifying compliance of the device with the policy on the basis of a subsequent quote after the update to the device.

3. The apparatus of claim 1 or 2, wherein the attestation database is a distributed ledger and the means are further configured to perform the attestation process by an agent of a distributed attestation network.

4. The apparatus of claim 3, wherein the agent is an event listener configured to at least detect events added in the attestation database indicative of one or more of a policy definition, a policy update, a successful quote, an unsuccessful quote, a device startup, a shutdown, a device reboot, a device update, a device attestation, an update patch release, and an update patch revocation.

5. The apparatus of claim 4, wherein the event listener is configured to
   - review events added in the attestation database after last successful quote in response to non-compliance of the device with the policy, and

- detect a need for update and/or an update patch for the device on the basis of the review of events.

6. The apparatus of any one of claims 3 to 5, wherein the agent in a blockchain node is configured to:
   - send a quote request to the device, and
   - receive the quote in a quote response from the device.

7. The apparatus of any preceding claim, wherein the means are further configured to perform:
   - receiving a key for attestation, and
   - verifying a signature in the quote on the basis of the key for attestation.

8. The apparatus of any preceding claim, the means are further configured to perform:
   - causing tagging of the device as not-trusted in response to at least one of failing validation of a signature received from the device, one or more values of the received measurements failing to match associated values in the policy, and conflict between additional information in the received quote and device state stored in the attestation database, and
   - causing an alert to a system administrator, or
   - triggering a software update from a patch system to the device in response to detecting non-compliance with the policy and availability of an update for the device.

9. An apparatus comprising means for performing:
   - detecting a trigger for software state measurements,
   - performing measurements of one or more software components in response to the trigger, and
   - sending a quote to an attestation component, the quote comprising measurements of the one or more software components of the device, for verifying compliance of the device with a policy indicating expected software state and comprising a set of information indicating rules and/or values for attesting the one or more software components.

10. The apparatus of any preceding claim, wherein the policy and the quote request indicate one or more platform configuration registers to be quoted, and the quote comprises measurements for the platform configuration registers.

11. The apparatus of any preceding claim wherein the means comprises at least one processor; and at least one memory including computer program code, the at least one memory and computer program code configured to, with the at least one processor, cause the performance of the apparatus.

12. A method, comprising:
- receiving, by an attestation component, a policy for a device quoted for software state measurements, wherein the policy comprises a set of information indicating rules and/or values for attesting one or more software components,
- receiving a quote of the device, the quote comprising measurements of one or more software components of the device,
- verifying compliance of the device with the policy on the basis of the received measurements of the one or more software components, and
- sending a quote event for the device for an attestation database.

13. A method, comprising:
- detecting, by a device for attestation, a trigger for software state measurements,
- performing measurements of one or more software components in response to the trigger, and
- sending a quote to an attestation component, the quote comprising measurements of the one or more software components of the device, for verifying compliance of the device with a policy indicating expected software state and comprising a set of information indicating rules and/or values for attesting the one or more software components.

14. The method of claim 12 or 13, wherein a distributed ledger is updated after successful verification and the attestation component is provided by an agent of a distributed attestation network.

15. A computer program, comprising instructions for causing an apparatus to perform the method of any one of claims 12 to 14.

5

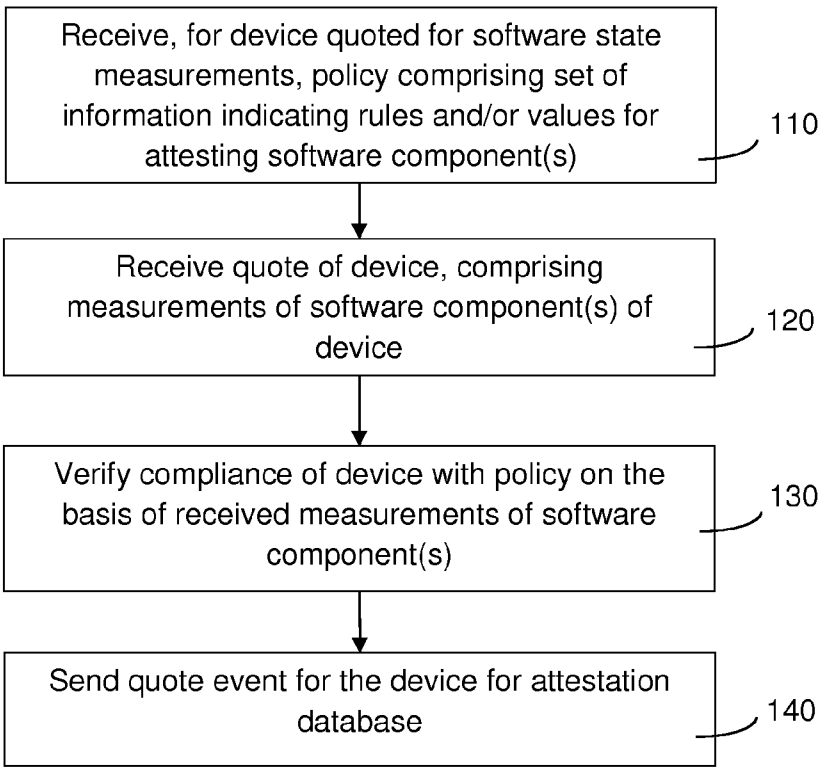
Receive, for device quoted for software state measurements, policy comprising set of information indicating rules and/or values for attesting software component(s)
110
Receive quote of device, comprising measurements of software component(s) of device
120
Verify compliance of device with policy on the basis of received measurements of software component(s)
130
Send quote event for the device for attestation database
140

Fig. 1

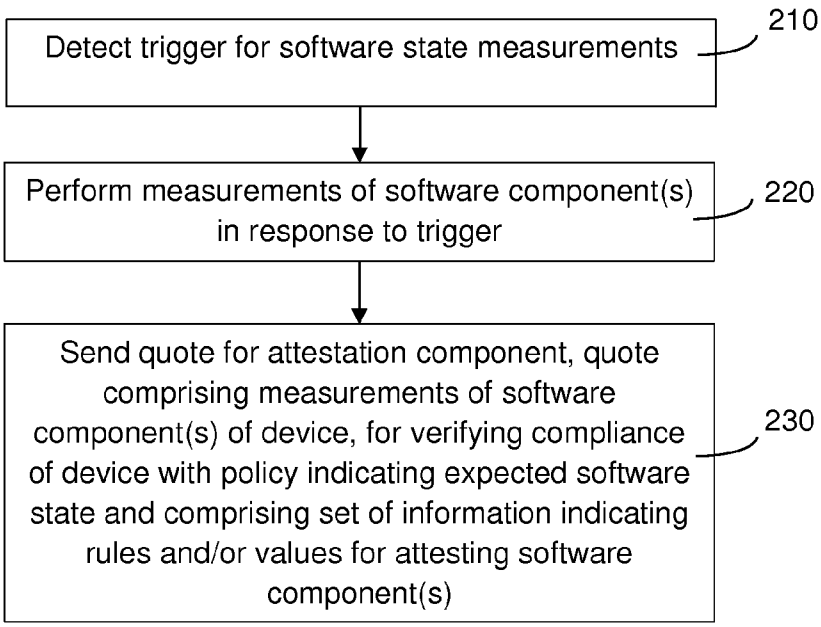
Detect trigger for software state measurements
210
Perform measurements of software component(s) in response to trigger
220
Send quote for attestation component, quote comprising measurements of software component(s) of device, for verifying compliance of device with policy indicating expected software state and comprising set of information indicating rules and/or values for attesting software component(s)
230

Fig. 2

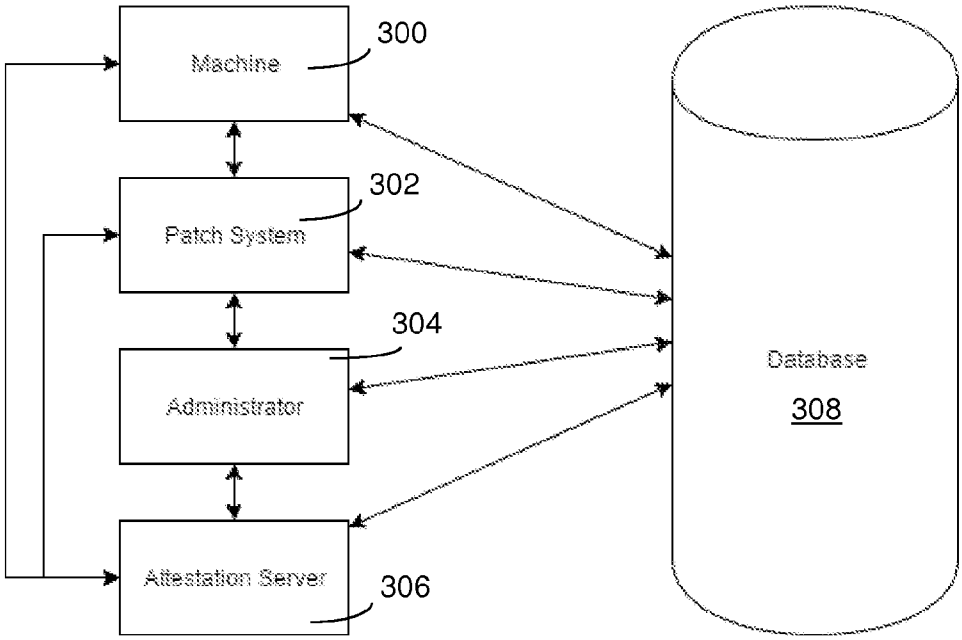
Machine
300
Patch System
302
Administrator
304
Attestation Server
306
Database
308

Fig. 3

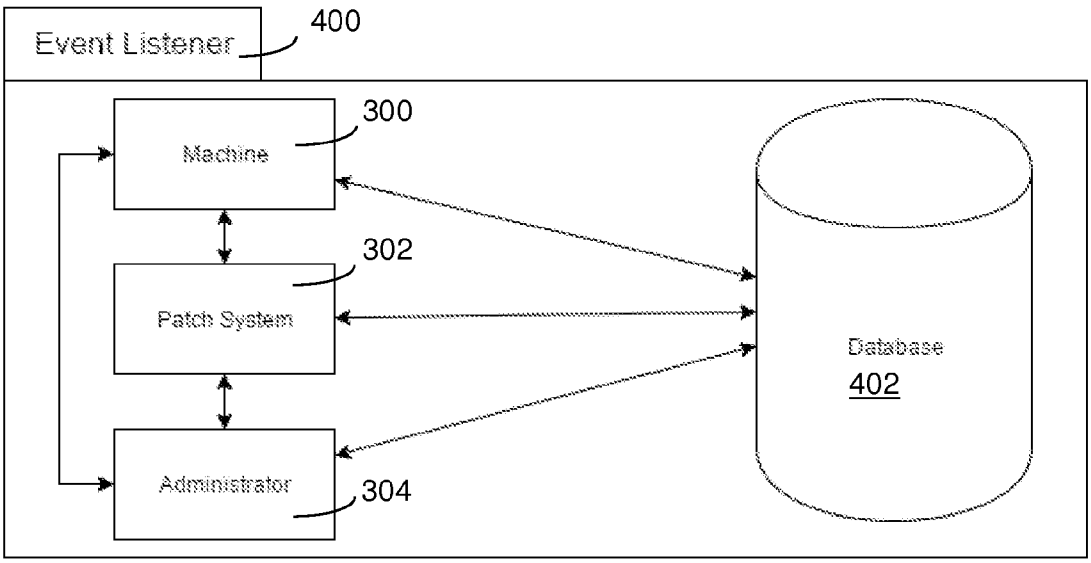
Event Listener
400
Machine
300
Patch System
302
Administrator
304
Database
402

Fig. 4

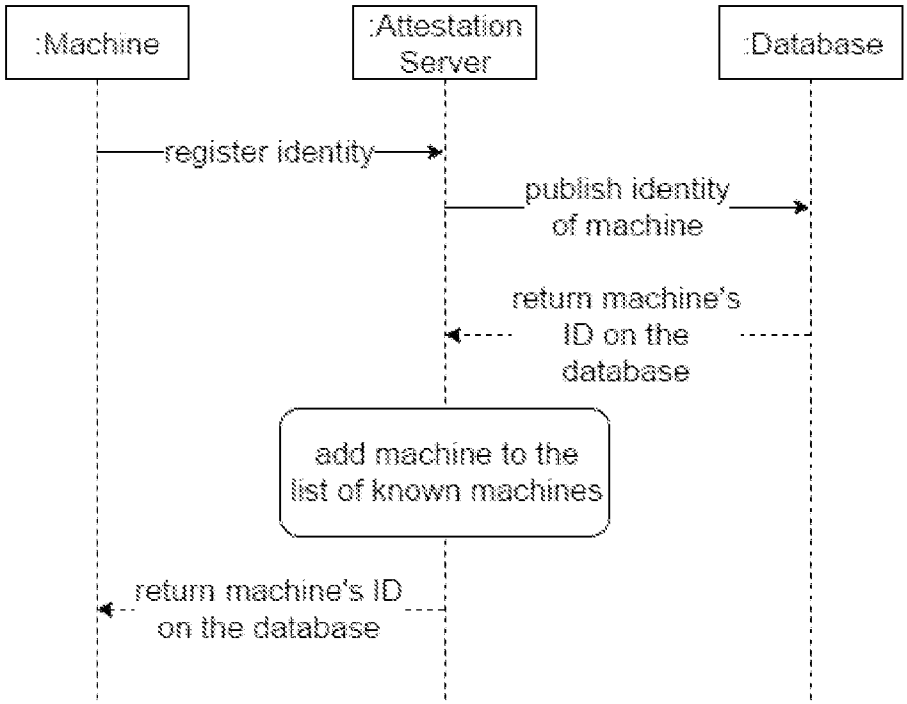
:Machine
:Attestation Server
:Database
register identity
publish identity of machine
return machine's ID on the database
add machine to the list of known machines
return machine's ID on the database

Fig. 5

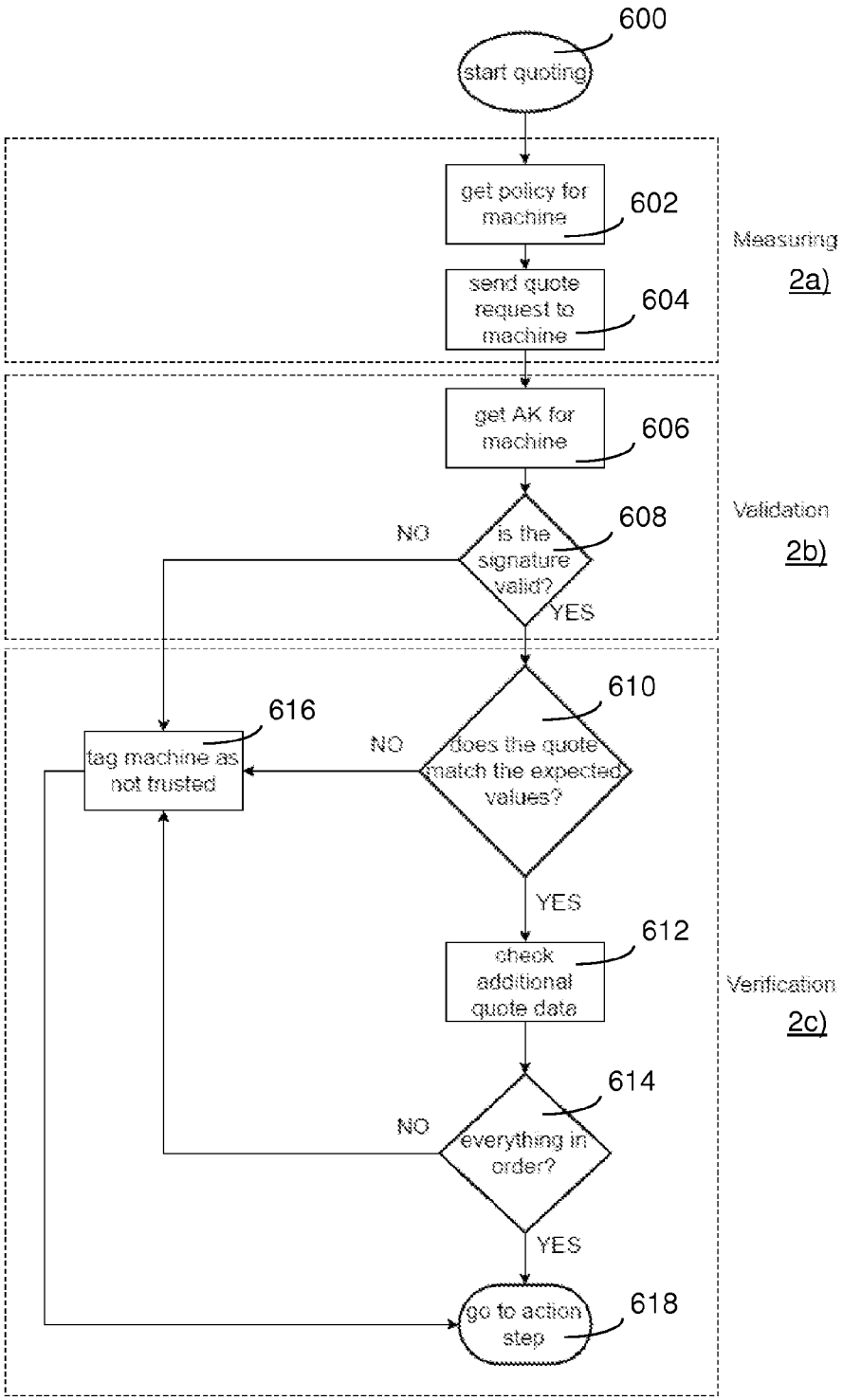
600
start quoting
get policy for machine
602
send quote request to machine
604
Measuring
2a)
get AK for machine
606
is the signature valid?
608
NO
YES
Validation
2b)
does the quote match the expected values?
610
NO
tag machine as not trusted
616
YES
check additional quote data
612
everything in order?
614
NO
YES
go to action step
618
Verification
2c)

Fig. 6

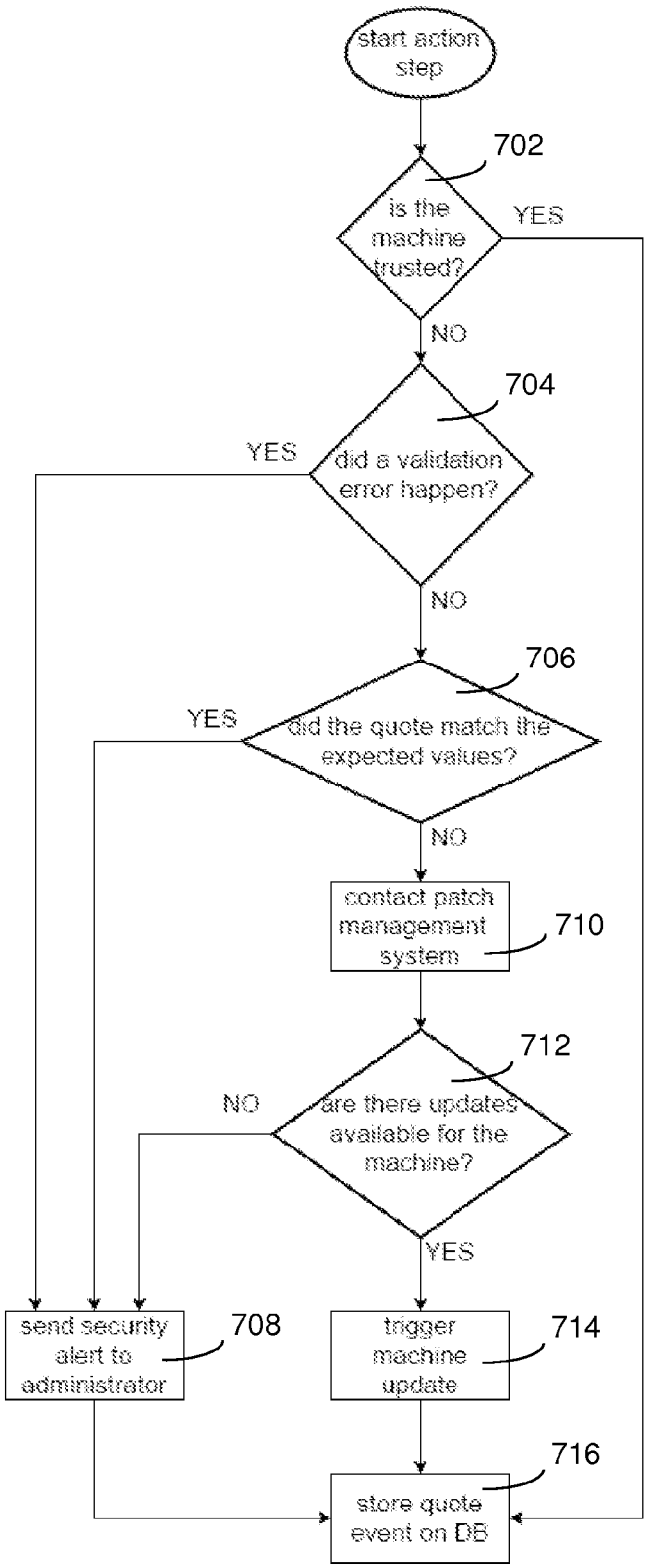
start action step
702
is the machine trusted?
YES
NO
704
did a validation error happen?
YES
NO
706
did the quote match the expected values?
YES
NO
contact patch management system
710
712
are there updates available for the machine?
NO
YES
send security alert to administrator
708
trigger machine update
714
716
store quote event on DB

Fig. 7

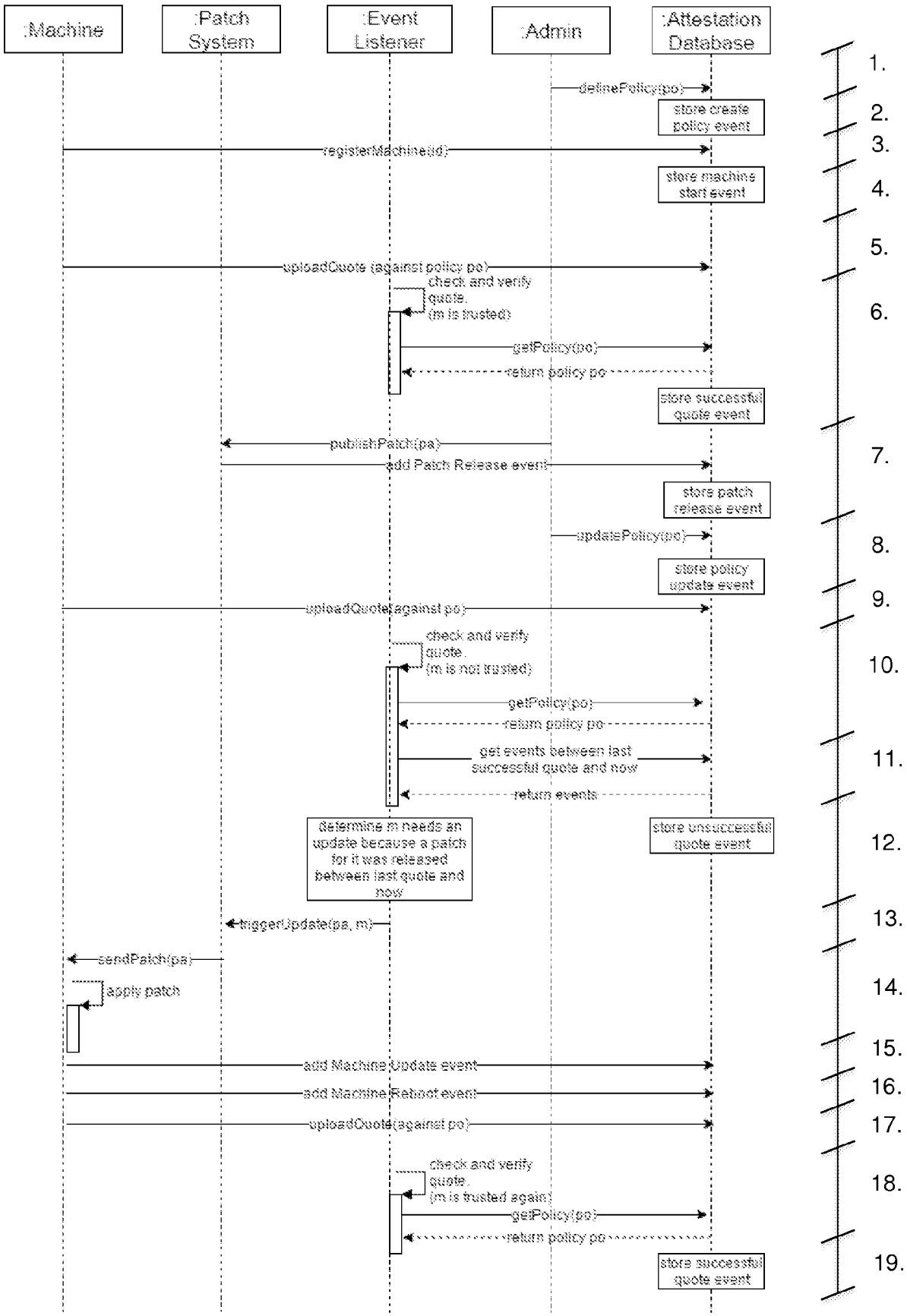
:Machine
:Patch System
:Event Listener
:Admin
:Attestation Database
definePolicy(po)
store create policy event
registerMachine(id)
store machine start event
uploadQuote (against policy po)
check and verify quote. (m is trusted)
getPolicy(po)
return policy po
store successful quote event
publishPatch(pa)
add Patch Release event
store patch release event
updatePolicy(po)
store policy update event
uploadQuote(against po)
check and verify quote. (m is not trusted)
getPolicy(po)
return policy po
get events between last successful quote and now
return events
determine m needs an update because a patch for it was released between last quote and now
store unsuccessful quote event
triggerUpdate(pa, m)
sendPatch(pa)
apply patch
add Machine Update event
add Machine Reboot event
uploadQuote(against po)
check and verify quote. (m is trusted again)
getPolicy(po)
return policy po
store successful quote event
1.
2.
3.
4.
5.
6.
7.
8.
9.
10.
11.
12.
13.
14.
15.
16.
17.
18.
19.

Fig. 8

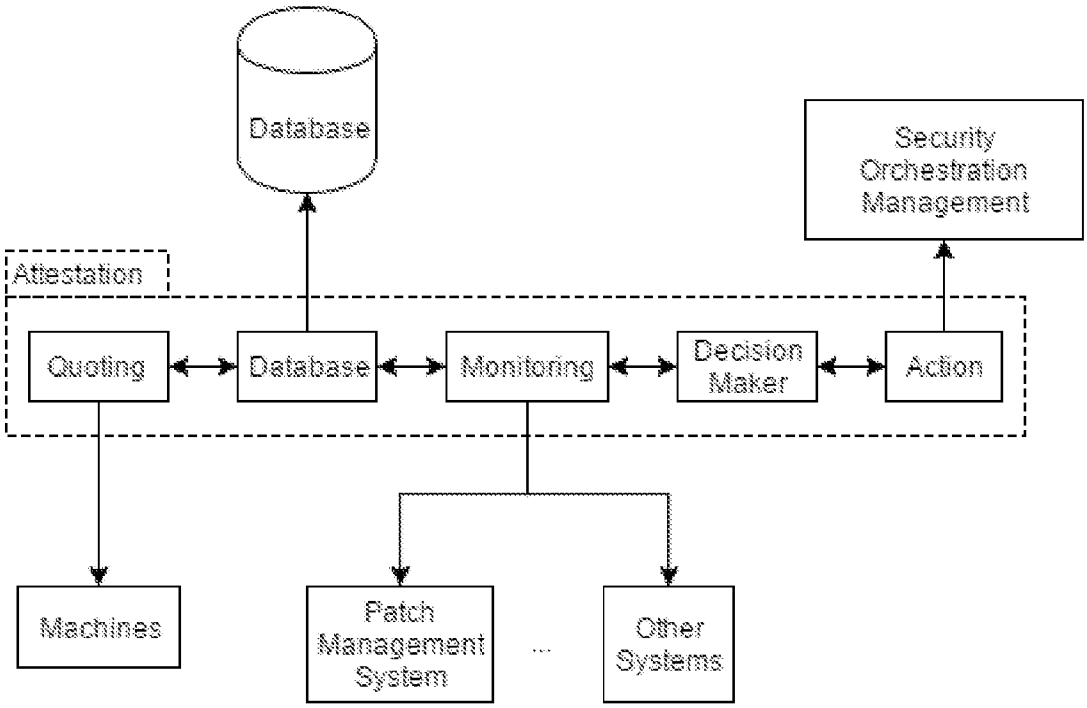
Database
Security Orchestration Management
Attestation
Quoting
Database
Monitoring
Decision Maker
Action
Machines
Patch Management System
...
Other Systems

Fig. 9

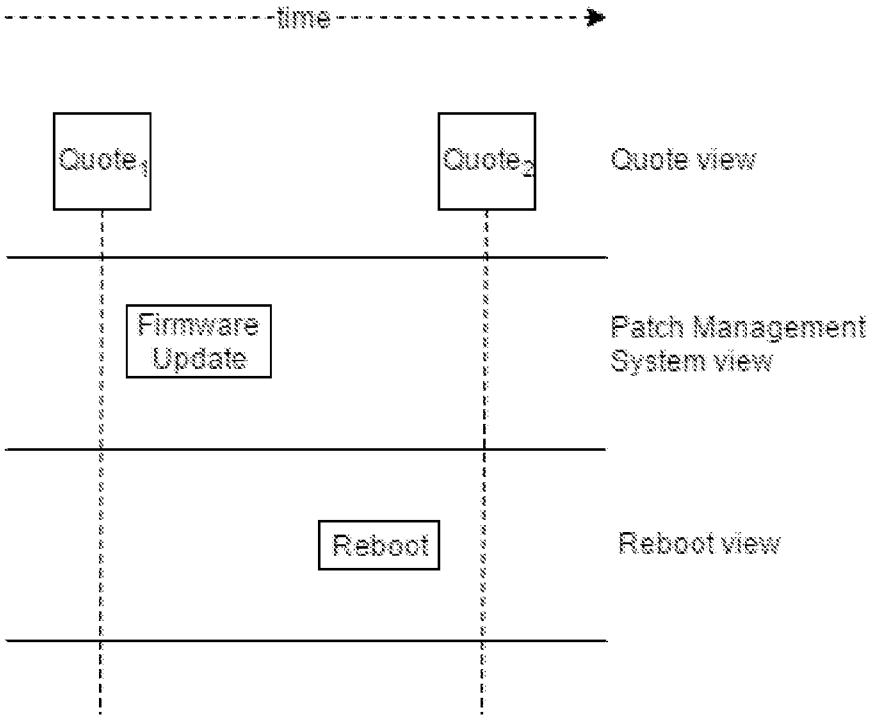
time
Quote1
Quote2
Quote view
Firmware Update
Patch Management System view
Reboot
Reboot view

Fig. 10

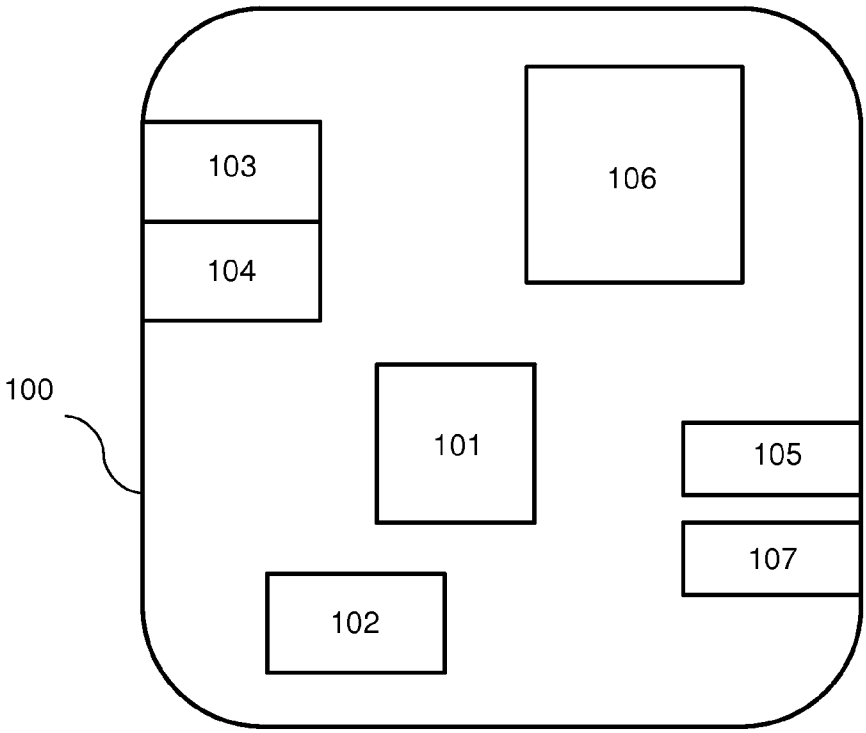

Fig. 11

INTERNATIONAL SEARCH REPORT

International application No.
PCT/FI2019/050529

A. **CLASSIFICATION OF SUBJECT MATTER**

See extra sheet

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)

IPC: G06F, H04L

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

FI, SE, NO, DK

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

EPODOC, EPO-Internal full-text databases, Full-text translation databases from Asian languages, WPIAP, XPAIP, XPESP, XPI3E, XPIEE, XPIOP, XPIPCOM, XPMISC, XPOAC, XPRD, XPTK, COMPDX, INSPEC, TDB

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	WO 2012098478 A1 (IBM [US]) 26 July 2012 (26.07.2012) Fig. 6; page 2, lines 4-8; page 2, lines 16-20; page 3, lines 26-31; page 9, line 8 – page 11, line 12; claims 21-22	1-2, 7-13, 15
Y	ibid.	3-6, 14
X	US 2017235946 A1 (POTLAPALLY NACHIKETH RAO [US] et al.) 17 August 2017 (17.08.2017) Figs. 1-2; paragraphs [0022]-[0023], [0031]-[0032], [0048], [0059]; claim 2	1-2, 7-13, 15
Y	ibid.	3-6, 14

☒ 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	Date of mailing of the international search report
17 October 2019 (17.10.2019)	22 October 2019 (22.10.2019)

Name and mailing address of the ISA/FI
Finnish Patent and Registration Office
FI-00091 PRH, FINLAND
Facsimile No. +358 29 509 5328

Authorized officer
Vesa-Matti Louekoski
Telephone No. +358 29 509 5000

INTERNATIONAL SEARCH REPORT

International application No.
PCT/FI2019/050529

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 2018088928 A1 (SMITH NED M [US] et al.) 29 March 2018 (29.03.2018) paragraphs [0001], [0013]	3-6, 14

INTERNATIONAL SEARCH REPORT
Information on Patent Family Members

International application No.
PCT/FI2019/050529

Patent document cited in search report	Publication date	Patent family members(s)	Publication date
WO 2012098478 A1	26/07/2012	CN 103329093 A	25/09/2013
		CN 103329093 B	12/09/2017
		DE 112012000512 T5	24/10/2013
		GB 201313795 D0	18/09/2013
		GB 2501433 A	23/10/2013
		GB 2501433 B	04/06/2014
		JP 2014503101 A	06/02/2014
		JP 5932837 B2	08/06/2016
		KR 20130114672 A	17/10/2013
		US 2014026124 A1	23/01/2014
		US 9317276 B2	19/04/2016
		US 2016162285 A1	09/06/2016
		US 10007510 B2	26/06/2018
		US 2016162396 A1	09/06/2016
		US 10108413 B2	23/10/2018
		US 2018246709 A1	30/08/2018
US 2017235946 A1	17/08/2017	US 10409985 B2	10/09/2019
		US 9147086 B1	29/09/2015
		US 2016070929 A1	10/03/2016
		US 9576155 B2	21/02/2017
US 2018088928 A1	29/03/2018	US 10185550 B2	22/01/2019
		CN 110169036 A	23/08/2019
		EP 3520368 A1	07/08/2019
		US 2019146778 A1	16/05/2019
		WO 2018064154 A1	05/04/2018

INTERNATIONAL SEARCH REPORT

International application No.
PCT/FI2019/050529

CLASSIFICATION OF SUBJECT MATTER

IPC
G06F 8/65 (2018.01)
G06F 11/30 (2006.01)
G06F 21/57 (2013.01)
G06F 21/64 (2013.01)
H04L 9/06 (2006.01)
H04L 9/08 (2006.01)
H04L 9/32 (2006.01)
H04L 29/06 (2006.01)