



(51) International Patent Classification:

G06F 21/53 (2013.01) G06F 21/83 (2013.01)

G06F 21/57 (2013.01) G06F 21/84 (2013.01)

(21) International Application Number:

PCT/US2020/020243

(22) International Filing Date:

27 February 2020 (27.02.2020)

(25) Filing Language:

English

(26) Publication Language:

English

(30) Priority Data:

16/297,162 08 March 2019 (08.03.2019) US

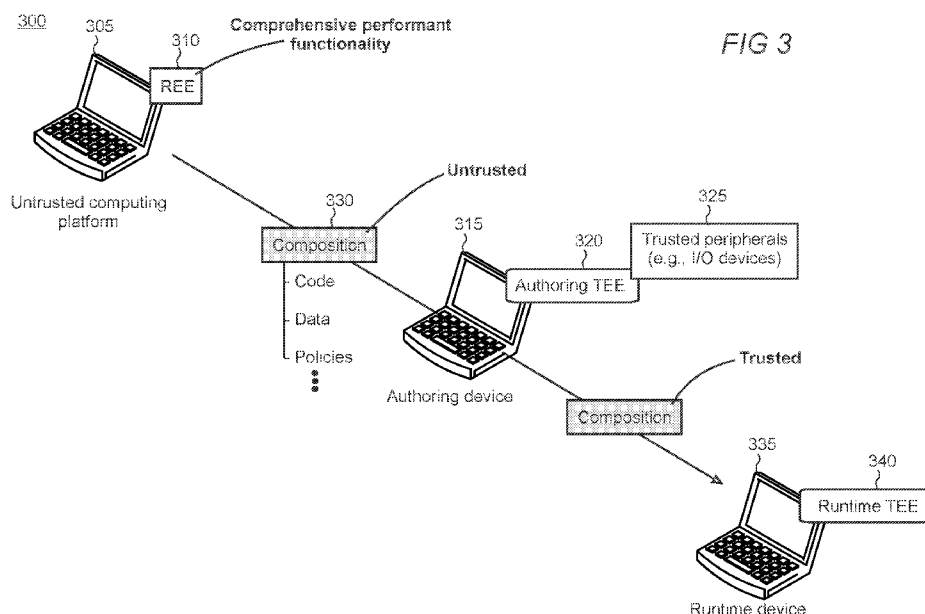
(71) Applicant: MICROSOFT TECHNOLOGY LICENSING, LLC [US/US]; One Microsoft Way, Redmond, Washington 98052-6399 (US).

(72) Inventors: THALER, David Garfield, III; MICROSOFT TECHNOLOGY LICENSING, LLC, One Microsoft Way, Redmond, Washington 98052-6399 (US). THOM, Stefan; MICROSOFT TECHNOLOGY LICENSING, LLC, One Microsoft Way, Redmond, Washington 98052-6399 (US). TELFER, Brian Clifford; MICROSOFT TECHNOLOGY LICENSING, LLC, One Microsoft Way, Redmond, Washington 98052-6399 (US).

(74) Agent: MINHAS, Sandip S. et al.; MICROSOFT TECHNOLOGY LICENSING, LLC, One Microsoft Way, Redmond, Washington 98052-6399 (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,

(54) Title: SECURE POLICY INGESTION INTO TRUSTED EXECUTION ENVIRONMENTS



(57) Abstract: Before a composition is ingested into a runtime environment at a runtime device, the composition may be verified at an authoring trusted execution environment (TEE) operating on an authoring device. A user can operate an untrusted computing platform (e.g., a personal computer, laptop computer, tablet computer, etc.) to write code, generate data, or create some other composition. Since this composition is created on an untrusted device, the authoring TEE may output the composition on a trusted peripheral device to a user for review and approval. Responsive to receiving approval at the trusted peripheral device, the authoring TEE can sign the composition with a local key and forward the composition for execution by the runtime device. The signature can be utilized by the runtime device to prove that it was reviewed and verified by an authorized user operating the authoring device.

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, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, ZA, ZM, ZW.

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

Declarations under Rule 4.17:

- *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*
- *as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii))*

Published:

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

SECURE POLICY INGESTION INTO TRUSTED EXECUTION ENVIRONMENTS

BACKGROUND

5 **[0001]** A trusted execution environment (TEE) using public/private key pairs provides a secure space for data and code execution with a higher level of security than a traditional rich execution environment that is outside the TEE (and thus considered untrusted). Code, data, policies, and the like (termed “compositions”) that are used, executed, and/or consumed in a given runtime TEE are conventionally authored in an environment that
10 includes untrusted computing platforms with peripherals such as input and display devices (e.g., mouse, keyboard, monitor, etc.).

[0002] These untrusted authoring environments can provide paths for malicious code to be incorporated into a composition without knowledge of the author. For example, an authoring tool running on an untrusted computing platform may be compromised with
15 malware that generates compositions having malicious content while appearing normal to the author with the expected appearance on a display and expected behaviors in response to inputs. The tainted composition may then be unknowingly ingested into a runtime TEE and thereby compromise its trusted functionalities.

SUMMARY

20 **[0003]** Compositions can be securely ingested into a runtime TEE by segmenting the authoring environment into two parts – an untrusted computing platform conventionally utilized by an author to create a composition, and an authoring TEE where the composition can be verified. The authoring TEE can enable the author (or another person) to verify that the composition accurately conforms with the author’s intentions and does
25 not include any unintended elements that may have been added by malware running on the untrusted computing platform. Once verified, the composition can be signed by the author in the authoring TEE using its private key and provided to the runtime TEE for secure ingestion as a trusted composition.

[0004] The verification supported in the authoring TEE uses secure I/O (input/output)
30 between a trusted processor and trusted peripherals that include a display device and an input device. Other trusted peripherals for user identification such as fingerprint and smart card readers can also be utilized in the authoring TEE to support multi-factor authentication of the author. The input device is specifically configured with limited functionality to keep its device driver compact (the display driver can also be configured

to be compact). For example, the input device may only include functionality for display navigation (e.g., “up,” “down,” “right,” “left,” etc.) and provide purposefully limited user input (e.g., “yes” and “no” or “accept” and “cancel,” etc.).

[0005] The limited user input supported in the authoring TEE ensures that only

5 verification of the integrity of the composition can be performed. No functionalities to edit, augment, or modify the composition are provided in the authoring TEE. For example, the author can scroll or page through a composition on the display using the input device to inspect and confirm integrity (the display can be configured as a large format device without negative impact on driver size in typical implementations). The composition can
10 then be signed (e.g., using the “yes” button) with a key in the authoring TEE as a trusted composition if the verification is successful, or rejected (e.g., using the “no” button) if the author believes that the composition is compromised or is otherwise untrustworthy.

[0006] The small input device and display device drivers enable the code executing inside the TEE to be kept compact which increases overall performance of the verification

15 segment in the authoring TEE while reducing the vulnerability to outside threats (i.e., a smaller execution space provides less opportunities for malicious code to hide and improves detection of such code). By purposely enabling the composition to be generated on the conventional untrusted computing platform, comprehensive performant functionality can be provided to the author including well known authoring tools plus a
20 range of peripheral devices that can be tailored to a particular composition authoring job at hand. Thus, complicated code and associated hardware for composition creation function outside the authoring TEE, while the code and hardware for the composition verification, which are relatively small, function inside. The segmented authoring solution is thereby enabled at a lower cost compared to solutions that attempt to support the entire computing
25 platform and I/O within the context of a TEE.

[0007] In cases when a composition needs to be transformed into human-consumable form from some other non-readable format, a transforming functionality can be instantiated in the authoring TEE, or distributed to another external TEE (e.g., a transforming TEE that is configured as a cloud resource). That way, the verified human-
30 consumable composition is appropriately verified when ingested into the runtime TEE. In other scenarios, the composition may be transformed from human-consumable form into machine-consumable form for execution by the runtime TEE.

[0008] This Summary is provided to introduce a selection of concepts in a simplified form that are further described below in the Detailed Description. This Summary is not

intended to identify key features or essential features of the claimed subject matter, nor is it intended to be used as an aid in determining the scope of the claimed subject matter.

Furthermore, the claimed subject matter is not limited to implementations that solve any or all disadvantages noted in any part of this disclosure. It will be appreciated that the above-described subject matter may be implemented as a computer-controlled apparatus, a
5 computer process, a computing system, or as an article of manufacture such as one or more computer-readable storage media. These and various other features will be apparent from a reading of the following Detailed Description and a review of the associated drawings.

10 DESCRIPTION OF THE DRAWINGS

[0009] FIG 1 shows an illustrative architecture with a rich execution environment (REE) and a trusted execution environment (TEE);

[0010] FIG 2 shows an illustrative hardware architecture of the rich execution environment and the trusted execution environment;

15 [0011] FIG 3 shows an illustrative environment in which a composition is verified at an authoring TEE before being executed by the runtime TEE;

[0012] FIG 4 shows an illustrative scenario where an untrusted computing platform is susceptible to malware;

[0013] FIG 5 shows an illustrative user interface which is infected with malware;

20 [0014] FIG 6 shows an illustrative environment in which trusted peripheral devices are utilized with the authoring TEE for an author to approve or deny compositions;

[0015] FIG 7 shows a taxonomy of configurations for the trusted peripheral devices;

[0016] FIG 8 shows an illustrative flowchart in which the authoring TEE approves or denies compositions;

25 [0017] FIGs 9 and 10 show illustrative types of input and output devices in communication with the authoring device;

[0018] FIG 11 shows an illustrative user interface in which a trusted peripheral device is configured to authorize or deny a composition responsive to a user's input;

[0019] FIG 12 shows an illustrative user interface in which a trusted peripheral device
30 is pre-configured with a timeout mechanism if a user fails to approve a composition; and

[0020] FIG 13 shows a flowchart of an illustrative method performed by an authoring device.

DETAILED DESCRIPTION

[0021] FIG 1 shows an illustrative architecture 100 of a computing device 105 having

a rich execution environment (REE) 115 running parallel to a trusted execution environment (TEE) 110. The REE may be subjected to additional exposure and security risks. In contrast, the TEE may operate securely in isolation from the REE to perform operations that are of a more sensitive nature. Such sensitive operations can include the handling of confidential personal information, banking data, the control of systems such as critical infrastructure, and composition verification, as discussed in further detail below.

[0022] As illustratively shown in FIG 1, the REE and the TEE include conventional computing configurations to perform various operations. In simplified form, the REE and TEE each include application, operating system, and hardware layers (collectively represented by numerals 120 and 125), although the TEE is isolated and the hardware and software therein are considered trusted. The TEE and REE can be configured according to specifications promulgated by the GlobalPlatform® standard, although other methodologies and/or standards may also be utilized. The TEE may at least satisfy minimum requirements in which the computing device has a unique security identity, any code inside the TEE is operator-authorized (e.g., such that the TEE can be enforced using hardware so that malware that is attempted to be injected into the TEE via a buffer overrun cannot be executed by the secure processor), and any data inside the TEE cannot be read by code outside the TEE.

[0023] The application layers support various applications that are executable by the operating system. The REE application layer can support applications such as a browser, messaging applications, and the like, whereas the application layer in the TEE can support trusted applications. In a system dedicated for critical infrastructure, the trusted applications can include a trusted application gateway which monitors and controls all transmissions that seek to traverse the TEE, thereby allowing or denying access. Other exemplary environments can include banking and finance in which processed transactions and associated personal customer information are executed, for example, by a trusted banking application executing in the TEE.

[0024] The operating system layers can manage system operations in which the hardware and application layers operate. In the REE, the operating system (OS) can include Windows®, whereas the TEE can run a secure OS which runs parallel to the REE OS. The OS layers can each support a TEE application programming interface (API) 130 which provides the interoperability of the two environments. That is, a client TEE API in the REE can be used to commence a session with the TEE and thereby allow the TEE to execute a trusted application.

[0025] The hardware layers support processors and memory, as depicted in FIG 2. Although the REE and TEE are illustratively shown as operating distinctly from each other, the various hardware components can be partitioned such that portions of the hardware are dedicated for only public operations, and portions of the hardware are dedicated for only trusted operations. This partitioning of the components and system provide the secure isolation offered by the TEE.

[0026] FIG 2 shows exemplary hardware of the computing device 105, which utilizes the REE 115 and TEE 110. As shown, the hardware components can be partitioned to isolate the TEE from the REE. The REE and TEE each include similar components which operate distinctly from the other environment. As illustratively shown, the various components include random access memory 205, read-only memory 210, peripherals 215 (e.g., input/output devices), processing core(s) 220, one-time password (OTP) fields 225, and crypto accelerators 230. In one exemplary embodiment, the REE can utilize external volatile memory 235 and external non-volatile memory 245, and a portion of the memories can have a protected area dedicated exclusively for the TEE, as representatively illustrated by numerals 240 and 250. The diagrams illustrated in FIGs 1 and 2 are illustrative only, and other configurations that satisfy the secure environment offered by the TEE are also possible.

[0027] FIG 3 shows an illustrative environment 300 that depicts the high-level communication path of a generated composition 330 from an untrusted computing platform 305 using an REE 310. The REE may be utilized due to the comprehensive performant functionality that is provided to a user when working on the composition, including well known authoring tools and a range of peripheral devices that can be tailored to a particular composition at hand. These components may otherwise not be available to the user when operating within a TEE. As a result, the composition, which can include code, data, policies, or other files, may be untrusted as illustratively shown in FIG 3.

[0028] The various devices can communicate over a network (not shown), which may include a personal area network, local area network, wide area network, or the Internet. The composition 330 may be transmitted to the authoring TEE 320 operating on the authoring device 315. The authoring TEE is connected with one or more trusted peripherals (e.g., I/O devices) 325 that enable exposure, review, and approval or denial of the composition 330 by a user. If the user approves the composition, then it is sent to the runtime TEE 340 operating on a runtime device 335 since the composition is verified and trusted. As discussed in further detail below, the authoring TEE signs the composition

with a private key to indicate to the runtime TEE that the composition has been reviewed and is trusted. The authoring of compositions is therefore segmented into two distinct parts, an untrusted computing platform where compositions are generated and an authoring device where the generated compositions are verified.

5 **[0029]** FIG 4 shows an exemplary diagram of the untrusted computing platform 305 that is susceptible to malware 405. In this example, the malware is configured to trick the untrusted computing platform into transmitting an operational instruction to the runtime device to bypass system limits. FIG 5 shows an illustrative user interface in which a dialog box 505 inquires whether the user would like to install an update and the user selects
10 “Yes”, as representatively shown by numeral 510. The displayed message to install an update, however, shields the actual instruction to bypass system limits at the runtime device as injected by the malware 405. Thus, the user’s approval to install the update transmitted a message destined for the runtime device to bypass its system limits beyond safety regulations, such as by increasing or decreasing temperature, revolutions per
15 minute, acceleration, etc. In the scenario in which the remote device stores private information, the malware may generate, for example, an unauthorized request for private information.

[0030] FIG 6 shows an illustrative environment in which the composition 330 from the untrusted computing platform 305 is received at the authoring TEE 320 of the
20 authoring device 315. The untrusted computing platform includes one or more applications maintained at an application layer 605 configured to operate and communicate with other devices on the network such as the authoring device 315 and runtime device 335. Transmission of the composition may be performed, for example, using SNMP (Simple Network Management Protocol), HTTP (HyperText Transfer
25 Protocol), etc.

[0031] The composition 330 is transmitted by a transport stack 610 (e.g., TCP/IP stack), where a transport stack 615 at the authoring device 315 breaks down the message for use by the TEE. A TEE API call may be utilized to facilitate the communications from the REE to the TEE. The security layer 620 is used to decrypt the transmission. For
30 example, although the security layer can depend on the protocol, SSL (Secure Sockets Layer) is one example of a security layer which is configured to decrypt the message or transmission. The composition is therefore only in decrypted form within the TEE which is inaccessible from the REE of the authoring device.

[0032] After decryption, the composition traverses through the TEE and is examined

by a message confirmer 625 which communicates with the trusted peripheral interface 635 and outputs the composition 330 for exposure, review, and approval or denial by a user using the trusted peripheral devices 640 and 645. While the trusted peripheral devices 640 and 645 are illustrated as being distinct devices, in some embodiments they may be combined devices such as a touch screen monitor. If the composition is authorized by the user, then the message is forwarded for execution by the runtime device 335. The composition may be signed by a key maintained by the trusted platform module (TPM) 650, encrypted by the security layer 655, and then assembled by the transport stack 615 for transmission to the runtime TEE 340 of the runtime device 335. If the user denies the composition (e.g., if after review, the composition appears compromised with malware), then the composition is suppressed and not forwarded for execution by the runtime TEE.

[0033] In some scenarios, the composition 330 may be in executable form for execution by the runtime environment or in some other format that is less easily verified. A transformer (e.g., compiler or decompiler depending on the scenario) 630 may transform the message into a different format (e.g., human-consumable form) for exposure on the trusted peripheral devices 640 and 645. This way the user can review the human-prepared or otherwise consumable form of the composition to further increase security and control over the executable composition. The transformer may also be used to compile the composition into machine-consumable form after it has been verified by a user. The compiler may be used when the runtime device is configured to execute machine-consumable compositions.

[0034] In other scenarios, a remote service 660, configured with a TEE 665 and transformer 630, may be utilized to transform the composition from machine-consumable form to human-consumable (e.g., human-readable) form using a decompiler. The decompiled composition is then securely transmitted back to the authoring TEE for exposure and review by the user. In other scenarios, the remote service may be configured to transform a composition from human-consumable form into machine-consumable form using a compiler. The remote service's compiler may be used after the composition has been verified and approved by the user and transmitted back to the authoring device so the runtime device can ingest the machine-consumable composition. Utilization of and interoperability with the remote service can enable and/or enhance the transformation functionality for the authoring device which may not have specific transformation capabilities.

[0035] The trusted peripheral devices 640 and 645 may be configured with limited

functionality to ensure the integrity of the authoring TEE 320. FIG 7 shows a taxonomy of illustrative limited function configurations of the trusted peripheral devices associated with the authoring TEE, as illustratively shown by numeral 705. For example, the limited functionality can include limited user input 710 which can include no edit, augment, or modification functions 715, scrolling functionality to review and confirm composition's integrity 720, and confirm or deny input permissions (e.g., a button press) for user approval 725. The trusted peripheral devices may be configured to output the composition for review 730 and connections to security devices to support multi-factor authentication for the author (e.g., smart card reader, biometric scanner, PIN, etc.) 735.

[0036] FIG 8 shows an illustrative flowchart of functionality of the authoring TEE 320. Reference table 840 shows which steps are necessarily performed and which steps are optional (e.g., if applicable) in the process. In step 805, the authoring TEE receives the composition 330 from an untrusted computing platform 305. In step 810, if applicable the received composition is transformed into human-consumable form. The composition may be transformed using a decompiler so the composition can be displayed and reviewed by the user in an understandable form. This may be, for example, the form in which the composition was originally created in the REE. If the composition is received at the authoring TEE in some executable form which is not consumable for a person, then the composition may be transformed into the human-consumable form for verification. This step is optional based on its applicability, that is, some compilations may be received in human-consumable form for which transformation is unnecessary.

[0037] In step 815, the authoring TEE displays the composition using trusted peripheral devices 640 and 645. In step 820, the authoring TEE receives user input at the trusted peripheral device. In step 825, if applicable, the compilation is transformed into machine-consumable form, such as using a compiler. This may occur, for example, if the compilation is in human-consumable form and not executable by the runtime device. In some scenarios, the runtime device can utilize the human-consumable form using an interpreter, and therefore the compilation is not compiled for execution. That is, the human-consumable form exposed to the user on the trusted peripheral device is the same form utilized by the runtime device.

[0038] In step 830, responsive to user approval of the composition, the authoring TEE signs the composition with a private key and then transmits the composition to the runtime TEE 340 operating on a runtime device 335. In step 835, responsive to user denial of the composition, the authoring TEE does not sign or transmit the composition to the runtime

TEE.

[0039] FIGs 9 and 10 show respective output and input devices 645 and 640 which may be utilized as the trusted peripheral devices with the authoring TEE. Exemplary and non-exhaustive output devices can include a display, a speaker, or a printer, individually and collectively represented by numeral 905. Exemplary and non-exhaustive input devices can include a simple push-button, fingerprint scanner, retina scanner, keyboard (e.g., for an alpha-numeric password), keypad (e.g., for a numeral PIN), and an NFC (near-field communication) receiver, individually and collectively represented by numeral 1005. The NFC receiver may be pre-configured to recognize and receive authorization from a portable device, such as a user's smart card.

[0040] FIG 11 shows an illustrative I/O device (e.g., a touchscreen monitor) configured as a trusted peripheral device with the authoring TEE. In this embodiment, a policy to increase temperature by 80° F can be permitted or rejected by the user. Selection of "Yes" or "No" respectively approves or denies the displayed composition.

[0041] As another embodiment, FIG 12 shows the illustrative I/O device which only includes an instruction to use the fingerprint scanner to confirm the transmission. In this scenario, if the user does not confirm the transmission using the fingerprint scanner, then after a predetermined time-period, the message is rejected and not forwarded to the runtime TEE. The input device may be configured to transmit a signal indicating the timeout to the secure terminal, or the secure terminal may be configured with its own timeout mechanism when confirmation is not timely received.

[0042] FIG 13 shows a flowchart of an illustrative method which may be implemented in an authoring device. Unless specifically stated, methods or steps shown in the flowcharts and described in the accompanying text are not constrained to a particular order or sequence. In addition, some of the methods or steps thereof can occur or be performed concurrently and not all the methods or steps have to be performed in a given implementation depending on the requirements of such implementation and some methods or steps may be optionally utilized.

[0043] In step 1305, the authoring device receives a composition generated on a rich execution environment on an untrusted computing platform, in which the composition is generated for utilization on a runtime device. In step 1310, the authoring device uses a trusted peripheral device that supports a user interface and is associated with a trusted execution environment (TEE) on the authoring device to expose the composition on the user interface to a user for review. In step 1315, the trusted peripheral device receives user

input that approves the composition as being trusted. In step 1320, responsive to the received user input approval, the authoring device signs the composition using a private key to verify that the composition is trusted. In step 1325, the authoring device delivers the signed composition to a runtime TEE operating on the runtime device where the

5 composition is utilized.

[0044] Referring back to FIG 2, the system architecture shown therein can describe a device capable of executing the various components described herein for the secure policy ingestion into trusted execution environments. Thus, the architecture may be adapted for an untrusted computing platform 305, authoring device, runtime device, or other
10 computing device including a server computer, a desktop computer, a netbook computer, a tablet computer, and/or a laptop computer. The architecture may be utilized to execute any aspect of the components presented herein.

[0045] The architecture illustrated in FIG 2 includes processing core(s) (e.g., Central Processing Unit(s)) 220, a system memory including a RAM 205 and a ROM 210, and a
15 system bus 260 that couples the memory to the processing core(s). The components, such as the processing core(s), can be partitioned such that there is a public portion of the processing core(s) and a trusted portion of the processing core(s), or alternatively separate hardware can be dedicated to the REE or the TEE. A basic input/output system containing the basic routines that help to transfer information between elements within the
20 architecture, such as during startup, is stored in the ROM 210. The architecture also shows exemplary volatile memory external to the device as representatively shown by numerals 235 and 240. In this example, the external volatile memory includes a dedicated protected area for use with the TEE. The architecture further includes a mass storage device (representatively shown as external non-volatile memory 245 and 250) for storing
25 software code or other computer-executed code that is utilized to implement applications, the file system, and the operating system. In this example, the external non-volatile memory includes a dedicated protected area for use with the TEE. Alternatively, the mass storage device can be internal to the device.

[0046] The mass storage device is connected to the processing core(s) 220 through a
30 mass storage controller (not shown) connected to the bus 260. The mass storage device and its associated computer-readable storage media provide non-volatile storage for the architecture.

[0047] Although the description of computer-readable storage media contained herein refers to a mass storage device, such as a hard disk or CD-ROM drive, it may be

appreciated by those skilled in the art that computer-readable storage media can be any available storage media that can be accessed by the architecture.

[0048] By way of example, and not limitation, computer-readable storage media may include volatile and non-volatile, removable and non-removable media implemented in any method or technology for storage of information such as computer-readable instructions, data structures, program modules, or other data. For example, computer-readable media includes, but is not limited to, RAM, ROM, EPROM (erasable programmable read only memory), EEPROM (electrically erasable programmable read only memory), Flash memory or other solid state memory technology, CD-ROM, DVD, HD-DVD (High Definition DVD), Blu-ray, or other optical storage, magnetic cassette, magnetic tape, magnetic disk storage or other magnetic storage device, or any other medium which can be used to store the desired information and which can be accessed by the architecture. The computer-readable storage medium may consist of propagating signals and be non-transitory.

[0049] According to various embodiments, the architecture may operate in a networked environment using logical connections to remote computers through a network. The architecture may connect to the network through a network interface unit (not shown) connected to the bus 260. It may be appreciated that the network interface unit also may be utilized to connect to other types of networks and remote computer systems. The architecture also may include an input/output controller (not shown) for receiving and processing input from a number of other devices, including a keyboard, mouse, or electronic stylus (not shown in FIG 2). Similarly, the input/output controller may provide output to a display screen, a printer, or other type of output device (also not shown in FIG 2).

[0050] It may be appreciated that the software components described herein may, when loaded into the processing core(s) 220 and executed, transform the processing core(s) 220 and the overall architecture from a general-purpose computing system into a special-purpose computing system customized to facilitate the functionality presented herein. The processing core(s) 220 may be constructed from any number of transistors or other discrete circuit elements, which may individually or collectively assume any number of states. More specifically, the processing core(s) 220 may operate as a finite-state machine, in response to executable instructions contained within the software modules disclosed herein. These computer-executable instructions may transform the processing core(s) 220 by specifying how the processing core(s) 220 transitions between states,

thereby transforming the transistors or other discrete hardware elements constituting the processing core(s) 220.

[0051] Encoding the software modules presented herein also may transform the physical structure of the computer-readable storage media presented herein. The specific transformation of physical structure may depend on various factors, in different implementations of this description. Examples of such factors may include, but are not limited to, the technology used to implement the computer-readable storage media, whether the computer-readable storage media is characterized as primary or secondary storage, and the like. For example, if the computer-readable storage media is implemented as semiconductor-based memory, the software disclosed herein may be encoded on the computer-readable storage media by transforming the physical state of the semiconductor memory. For example, the software may transform the state of transistors, capacitors, or other discrete circuit elements constituting the semiconductor memory. The software also may transform the physical state of such components in order to store data thereupon.

[0052] As another example, the computer-readable storage media disclosed herein may be implemented using magnetic or optical technology. In such implementations, the software presented herein may transform the physical state of magnetic or optical media, when the software is encoded therein. These transformations may include altering the magnetic characteristics of particular locations within given magnetic media. These transformations also may include altering the physical features or characteristics of particular locations within given optical media to change the optical characteristics of those locations. Other transformations of physical media are possible without departing from the scope and spirit of the present description, with the foregoing examples provided only to facilitate this discussion.

[0053] In light of the above, it may be appreciated that many types of physical transformations take place in the architecture in order to store and execute the software components presented herein. It also may be appreciated that the architecture may include other types of computing devices, including handheld computers, embedded computer systems, smartphones, and PDAs known to those skilled in the art. It is also contemplated that the architecture may not include all of the components shown in FIG 2, may include other components that are not explicitly shown in FIG 2, or may utilize an architecture completely different from that shown in FIG 2.

[0054] Various exemplary embodiments of the present secure policy ingestion into trusted execution environments are now presented by way of illustration and not as an

exhaustive list of all embodiments. An example includes a method to verify compositions within a trusted execution environment (TEE) of an authoring device, comprising: receiving, at the authoring device, a composition generated on a rich execution environment (REE) on an untrusted computing platform, the composition being generated for utilization on a runtime device; using a trusted peripheral device that supports a user interface and is associated with the TEE on the authoring device to expose the composition on the user interface to a user for review; at the trusted peripheral device, receiving user input approving the composition as being trusted; responsive to the received user input approval, signing the composition using a private key to verify that the composition is trusted; and delivering the signed composition to a runtime TEE operating on the runtime device where the composition is utilized.

[0055] In another example, the composition is any one of code, data, or policies. In another example, the method further comprises transforming the composition into human-consumable form and exposing the composition in human-consumable form on the trusted peripheral device. In another example, the method further comprises: forwarding the composition to a remote service, wherein the remote service is configured with a decompiler to transform the composition into human-consumable form; and receiving the transformed composition from the remote service for exposure on the trusted peripheral device. In another example, the method further comprises transforming the composition into machine-consumable form for delivery to and ingestion by the runtime device. In another example, the method further comprises: forwarding the composition to a remote service, wherein the remote service is configured with a compiler to transform the composition into machine-consumable form; and receiving the transformed composition from the remote service for delivery to and ingestion by the runtime device. In another example, the trusted peripheral devices include a security device unique to an authorized user to approve the composition, and the security device includes a fingerprint scanner, retina scanner, keypad for a personal identification number (PIN), near-field communication (NFC) receiver, or a keyboard for an alpha-numeric password. In another example, the trusted peripheral devices are configured with limited functionality of outputting the composition and receiving approval or denial input from a user, wherein limited functionality includes prohibiting editing, augmenting, or modification of compositions. In another example, the private key is associated with a trusted platform module (TPM) of the TEE, and the private key is unique to the TPM.

[0056] A further example includes an authoring device, comprising: one or more

processors; and a hardware-based memory device comprising instructions which, when executed by the one or more processors operating in a trusted execution environment (TEE) of the authoring device, cause the authoring device to: receive, at the authoring device, a composition generated on a rich execution environment (REE) on an untrusted computing platform, the composition being generated for utilization on a runtime device; use a trusted peripheral device that supports a user interface and is associated with the TEE on the authoring device to expose the composition on the user interface to a user for review; at the trusted peripheral device, receive user input approving the composition as being trusted; responsive to the received user input approval, sign the composition using a private key to verify that the composition is trusted; and deliver the signed composition to a runtime TEE operating on the runtime device where the composition is utilized.

[0057] In another example, the composition is any one of code, data, or policies. In another example, the executed instructions further cause the authoring device to transform the composition into human-consumable form and expose the composition in human-consumable form on the trusted peripheral device. In another example, the executed instructions further cause the authoring device to: forward the composition to a remote service, wherein the remote service is configured with a decompiler to transform the composition into human-consumable form; and receive the transformed composition from the remote service for exposure on the trusted peripheral device. In another example, the executed instructions further cause the authoring device to transform the composition into machine-consumable form for delivery to and execution by the runtime device.

[0058] A further example includes one or more hardware-based non-transitory computer-readable memory devices storing instructions which, when executed by one or more processors disposed in an authoring device, cause the authoring device to: receive, at the authoring device, a composition generated on a rich execution environment (REE) on an untrusted computing platform, the composition being generated for utilization on a runtime device; use a trusted peripheral device that supports a user interface and is associated with the TEE on the authoring device to expose the composition on the user interface to a user for review; at the trusted peripheral device, receive user input approving the composition as being trusted; responsive to the received user input approval, sign the composition using a private key to verify that the composition is trusted; and deliver the signed composition to a runtime TEE operating on the runtime device where the composition is utilized.

[0059] In another example, the composition is any one of code, data, or policies. In

another example, the executed instructions further cause the authoring device to transform the composition into human-consumable form and expose the composition in human-consumable form on the trusted peripheral device. In another example, the executed instructions further cause the authoring device to: forward the composition to a remote
5 service, wherein the remote service is configured with a decompiler to transform the composition into human-consumable form; and receive the transformed composition from the remote service for exposure on the trusted peripheral device. In another example, the trusted peripheral devices are configured with limited functionality of outputting the composition and receiving approval or denial input from a user. In another example, the
10 limited functionality includes prohibiting editing, augmenting, or modification of compositions.

[0060] The subject matter described above is provided by way of illustration only and is not to be construed as limiting. Various modifications and changes may be made to the subject matter described herein without following the example embodiments and
15 applications illustrated and described, and without departing from the true spirit and scope of the present invention, which is set forth in the following claims.

CLAIMS

1. A method to verify compositions in a trusted execution environment (TEE) on an authoring device, comprising:
 - receiving, at the authoring device, a composition generated on a rich execution environment (REE) on an untrusted computing platform, the composition being generated for utilization on a runtime device;
 - using a trusted peripheral device that supports a user interface and is associated with the TEE on the authoring device to expose the composition on the user interface to a user for review;
 - at the trusted peripheral device, receiving user input approving the composition as being trusted;
 - responsive to the received user input approval, signing the composition using a private key to verify that the composition is trusted; and
 - delivering the signed composition to a runtime TEE operating on the runtime device where the composition is utilized.
2. The method of claim 1, in which the composition is any one of code, data, or policies.
3. The method of claim 1, further comprising transforming the composition into human-consumable form and exposing the composition in human-consumable form on the trusted peripheral device.
4. The method of claim 1, further comprising:
 - forwarding the composition to a remote service, wherein the remote service is configured with a decompiler to transform the composition into human-consumable form; and
 - receiving the transformed composition from the remote service for exposure on the trusted peripheral device.
5. The method of claim 1, further comprising transforming the composition into machine-consumable form for delivery to and ingestion by the runtime device.
6. The method of claim 1, further comprising:
 - forwarding the composition to a remote service, wherein the remote service is configured with a compiler to transform the composition into machine-consumable form; and
 - receiving the transformed composition from the remote service for delivery to and ingestion by the runtime device.

7. The method of claim 1, in which the trusted peripheral device includes a security device unique to an authorized user to approve the composition, and the security device includes a fingerprint scanner, retina scanner, keypad for a personal identification number (PIN), near-field communication (NFC) receiver, or a keyboard for an alpha-numeric password.
8. The method of claim 1, in which the trusted peripheral device is configured with limited functionality of outputting the composition and receiving approval or denial input from a user, wherein limited functionality includes prohibiting editing, augmenting, or modification of compositions.
9. The method of claim 1, in which the private key is associated with a trusted platform module (TPM) of the TEE, and the private key is unique to the TPM.
10. An authoring device, comprising:
 - one or more processors; and
 - a hardware-based memory device comprising instructions which, when executed by the one or more processors operating in a trusted execution environment (TEE) of the authoring device, cause the authoring device to:
 - receive, at the authoring device, a composition generated on a rich execution environment (REE) on an untrusted computing platform, the composition being generated for utilization on a runtime device;
 - use a trusted peripheral device that supports a user interface and is associated with the TEE on the authoring device to expose the composition on the user interface to a user for review;
 - at the trusted peripheral device, receive user input approving the composition as being trusted;
 - responsive to the received user input approval, sign the composition (330) using a private key to verify that the composition is trusted; and
 - deliver the signed composition to a runtime TEE operating on the runtime device where the composition is utilized.
11. The authoring device of claim 10, in which the composition is any one of code, data, or policies.
12. The authoring device of claim 10, in which the executed instructions further cause the authoring device to transform the composition into human-consumable form and expose the composition in human-consumable form on the trusted peripheral device.
13. The authoring device of claim 10, in which the executed instructions further cause

the authoring device to:

forward the composition to a remote service, wherein the remote service is configured with a decompiler to transform the composition into human-consumable form; and

receive the transformed composition from the remote service for exposure on the trusted peripheral device.

14. The authoring device of claim 10, in which the executed instructions further cause the authoring device to transform the composition into machine-consumable form for delivery to and execution by the runtime device.

15. The authoring device of claim 10, in which the executed instructions further cause the authoring device to:

forward the composition to a remote service, wherein the remote service is configured with a compiler to transform the composition into machine-consumable form; and

receive the transformed composition from the remote service for delivery to and ingestion by the runtime device.

FIG 1
100

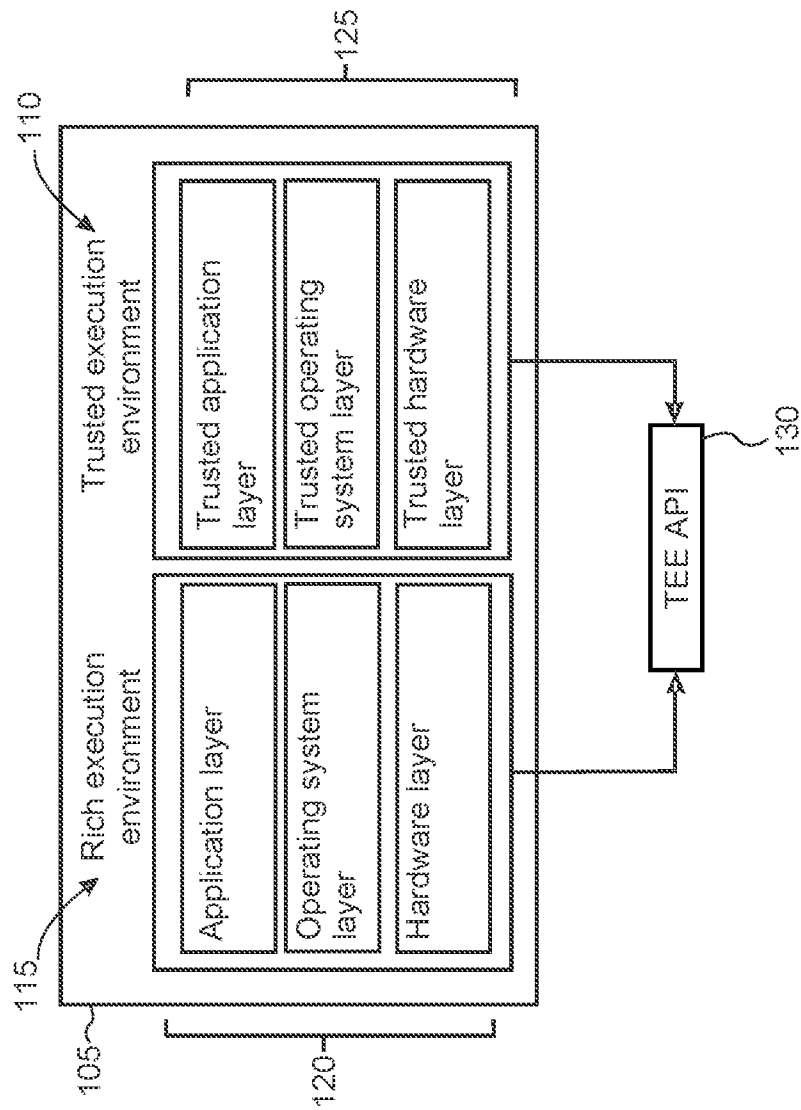


FIG 2

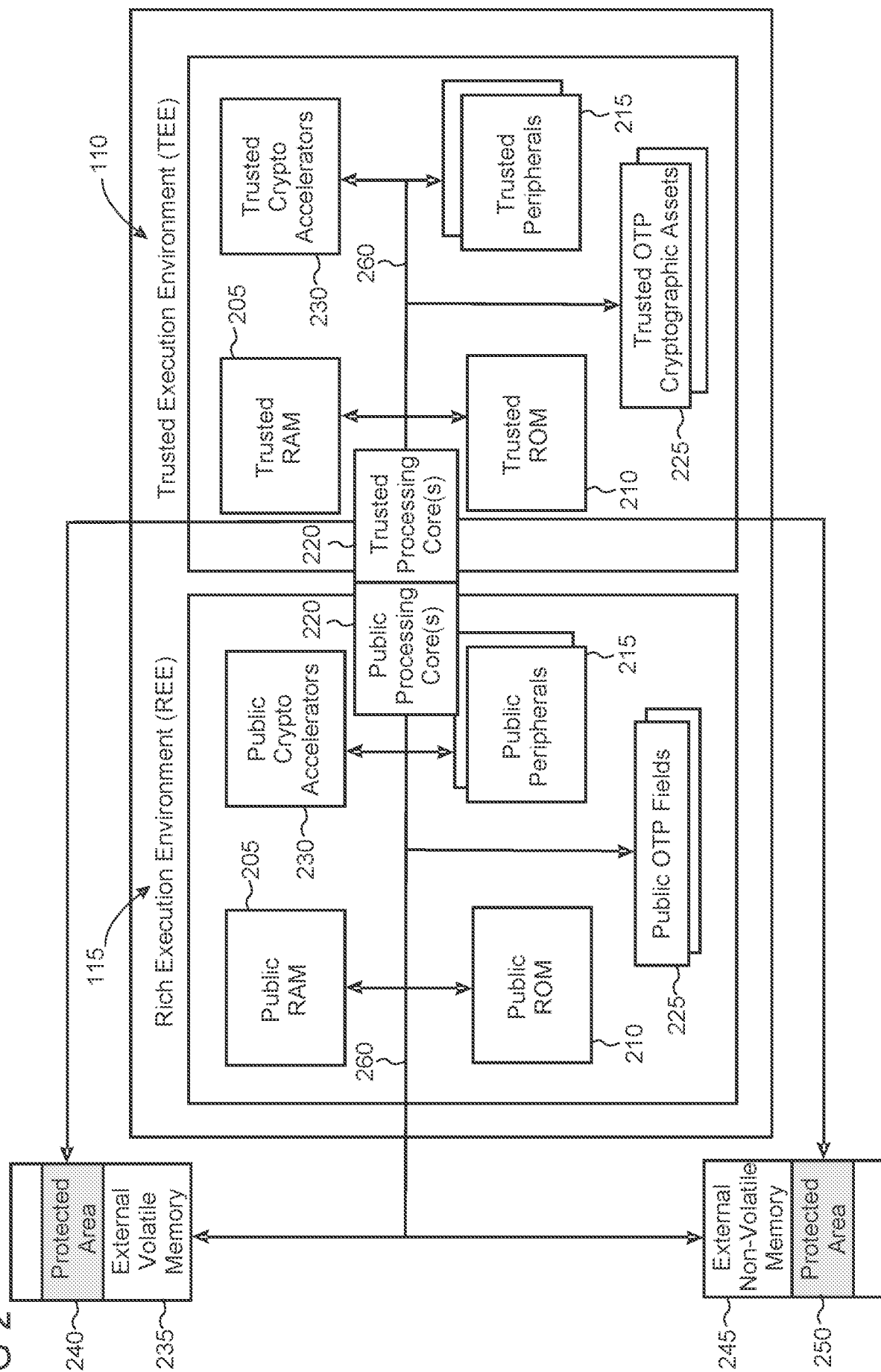


FIG 3

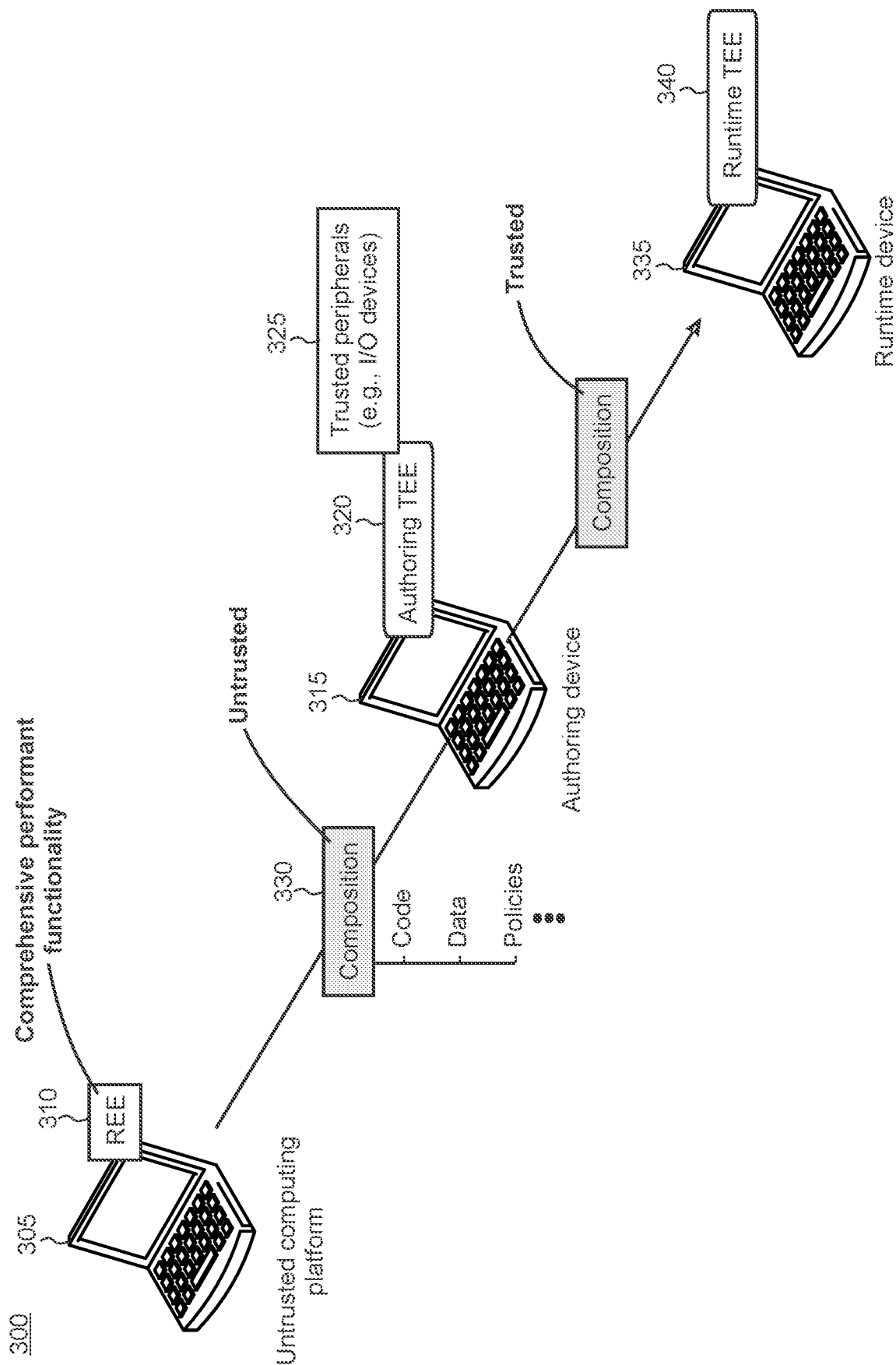


FIG 4

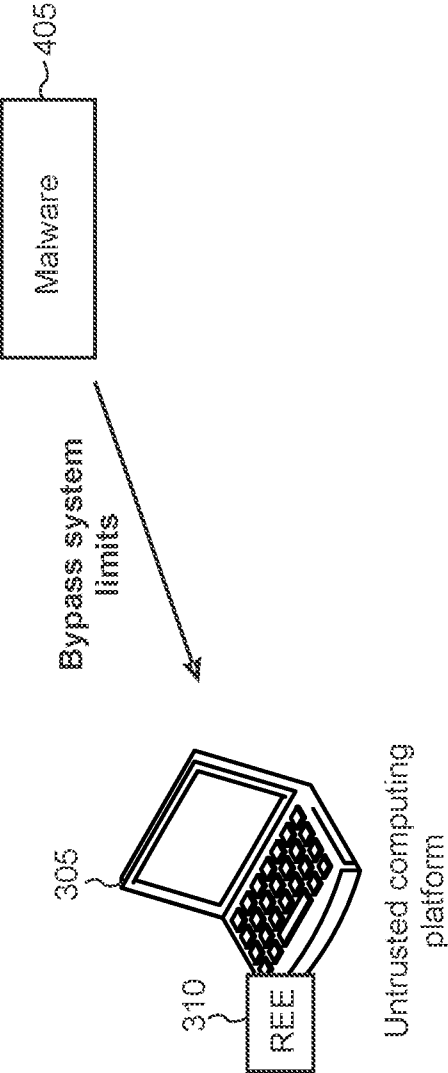


FIG 5

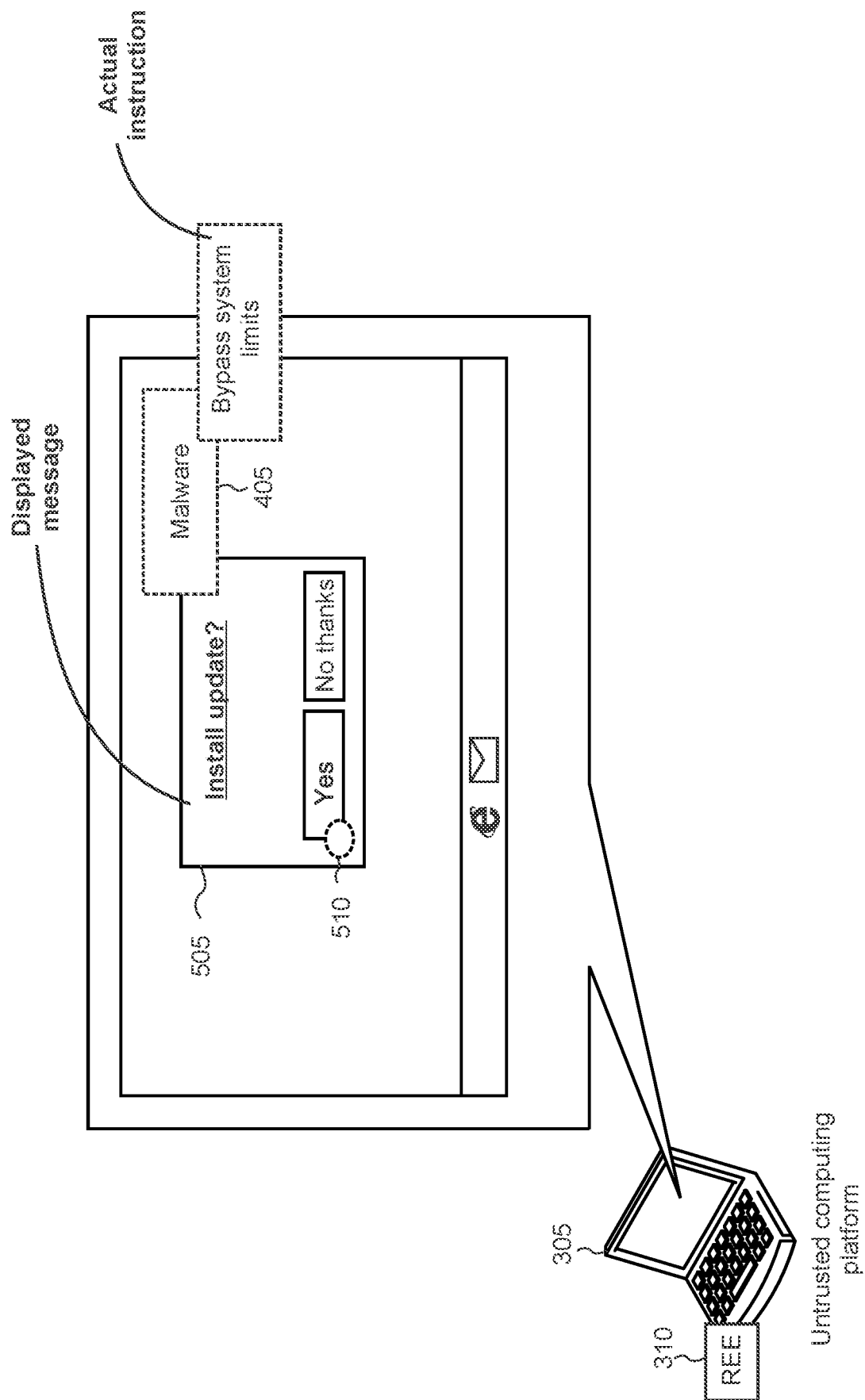


FIG 6

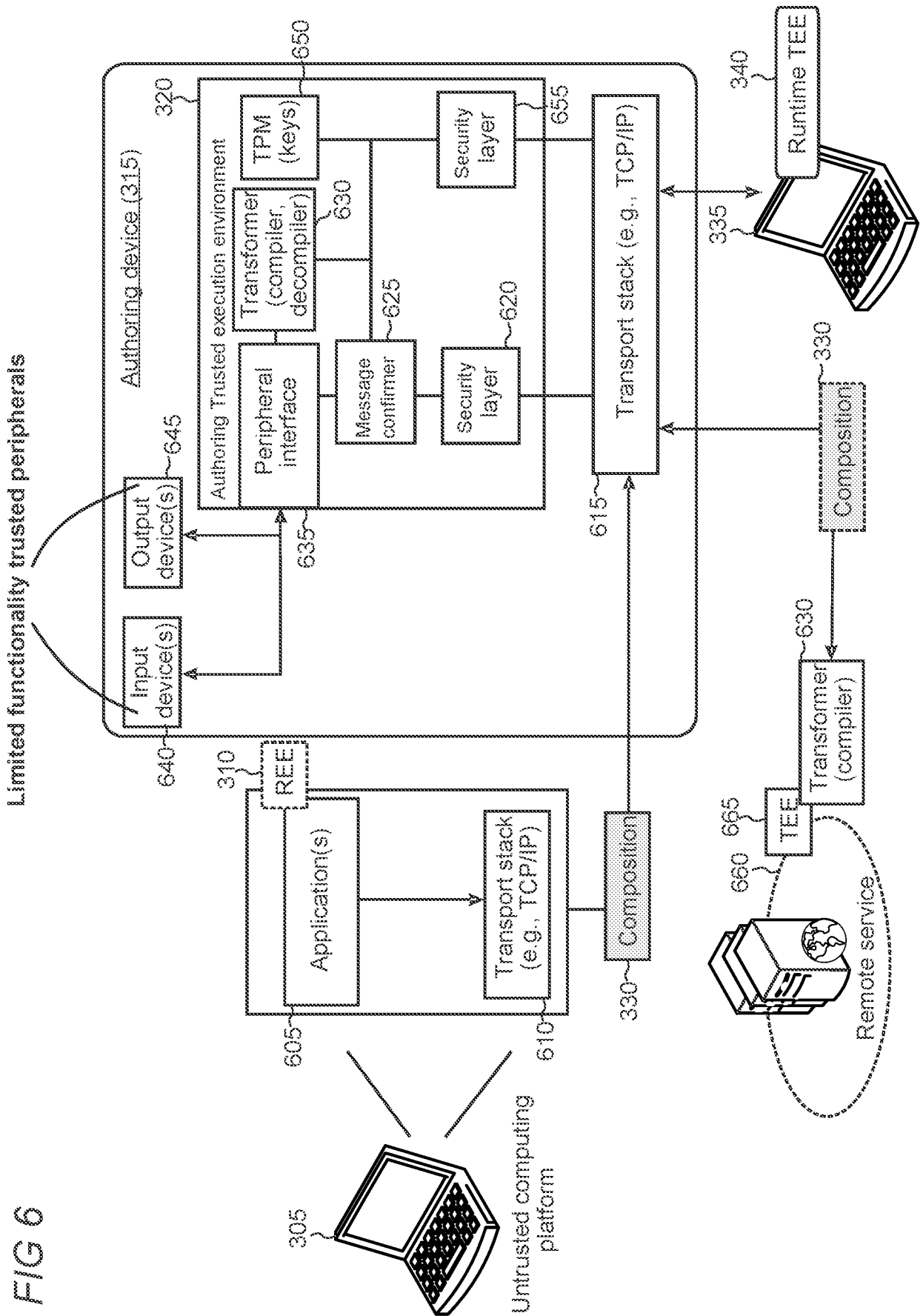
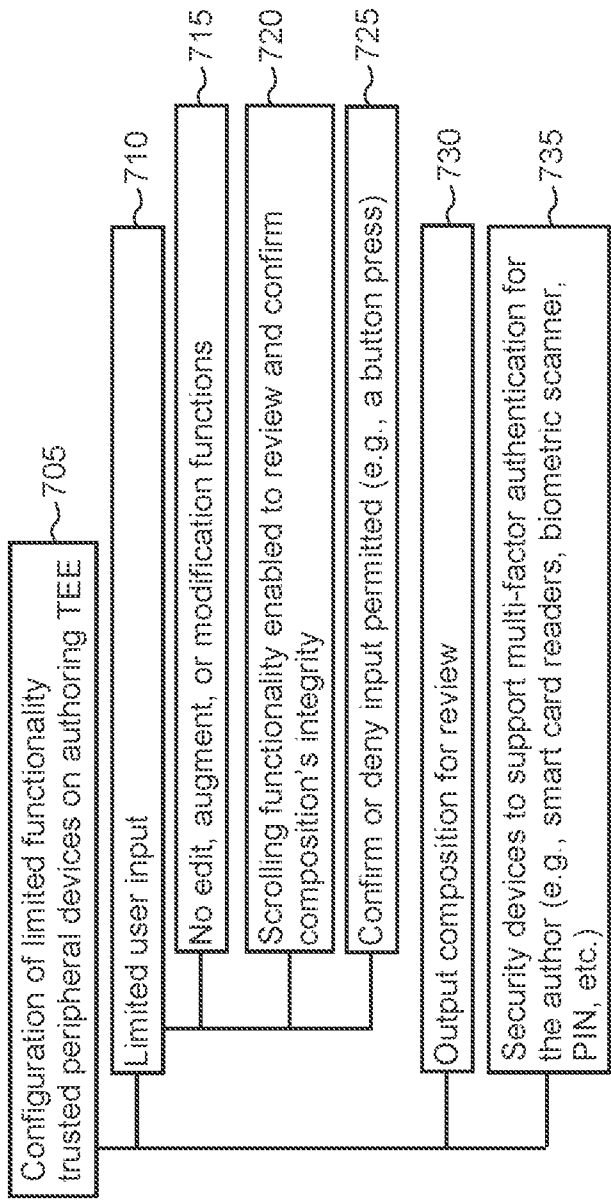


FIG 7



7/10

FIG 8

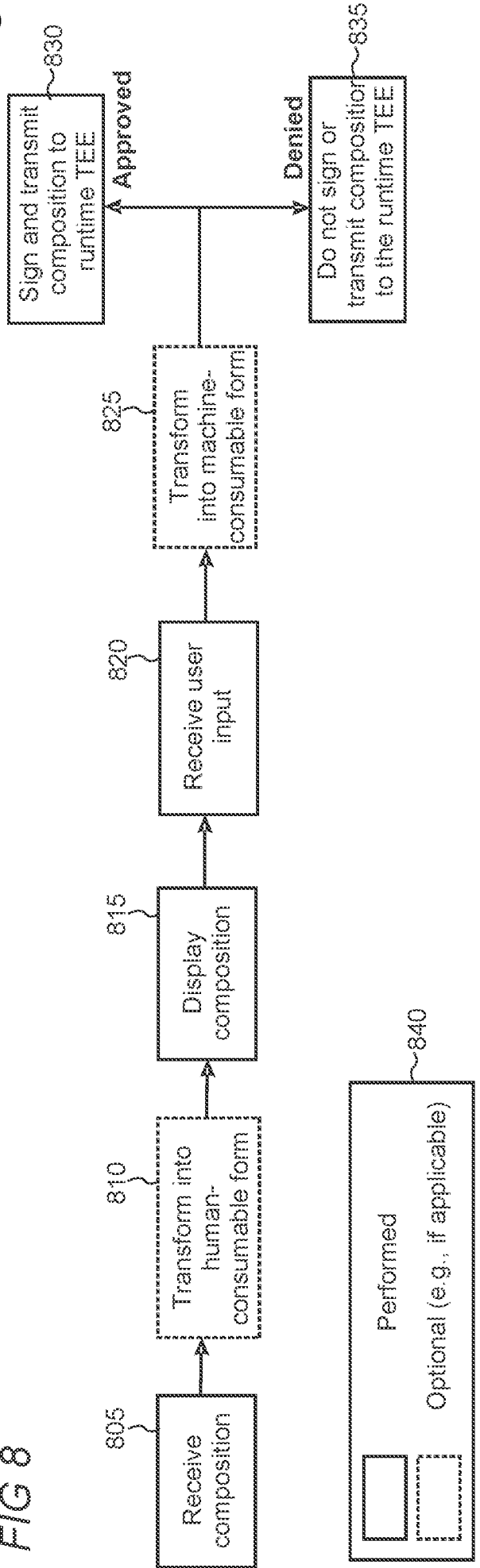


FIG 9

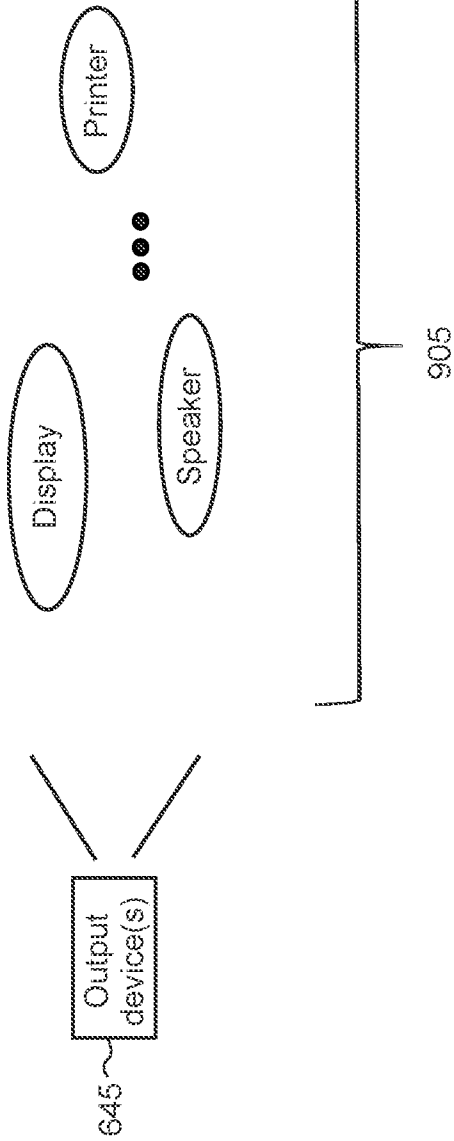


FIG 10

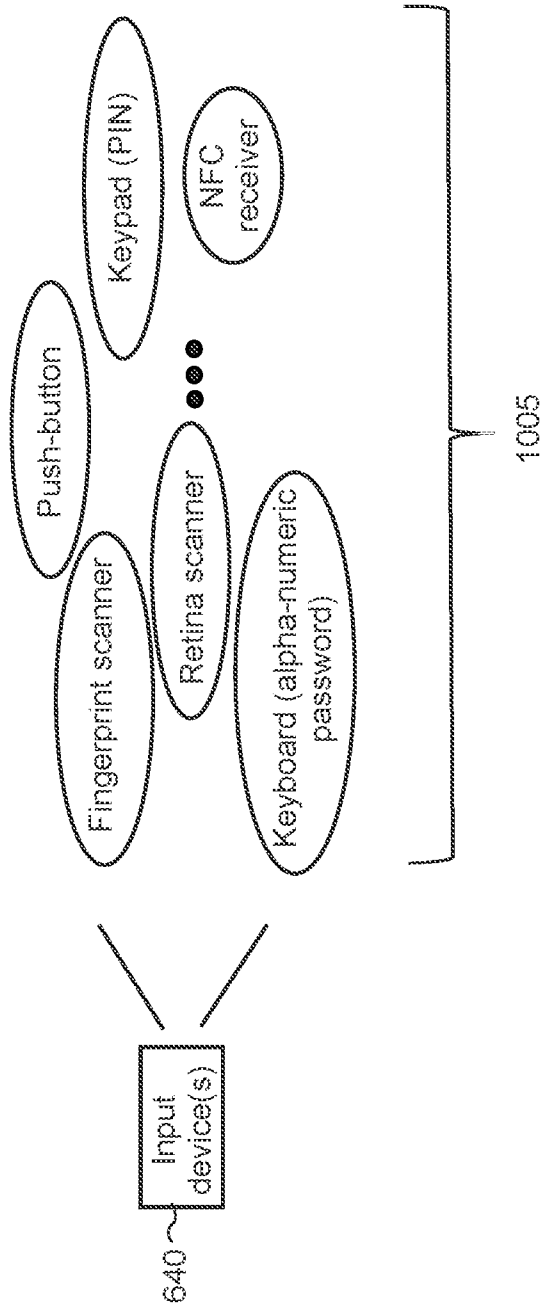


FIG 11

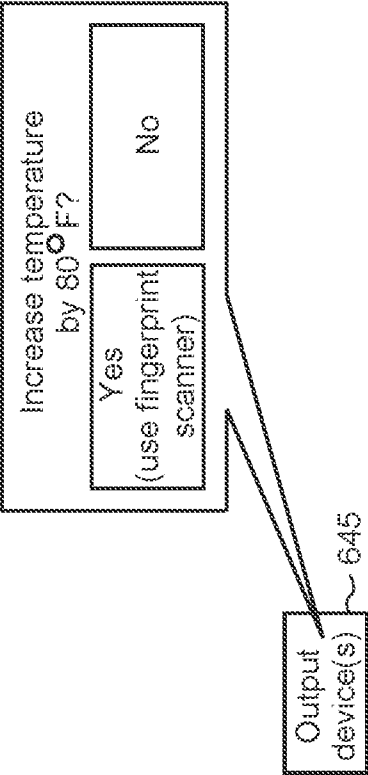
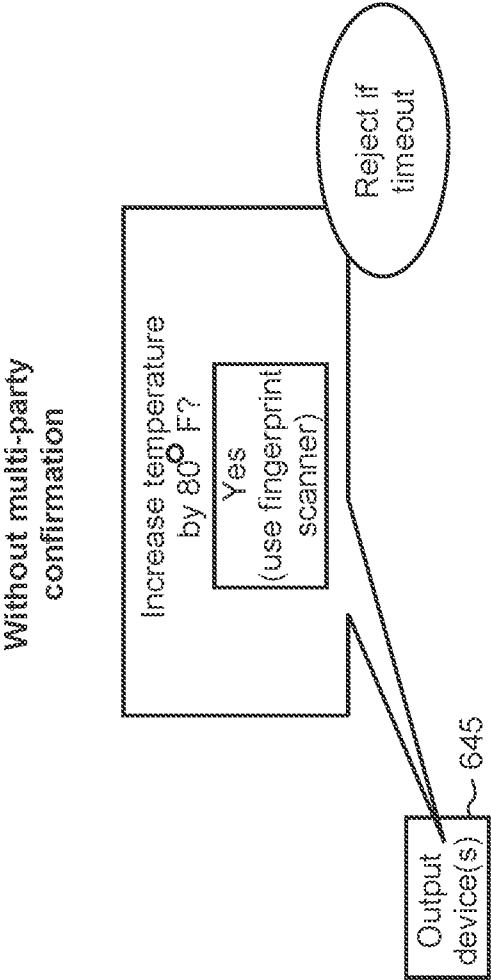
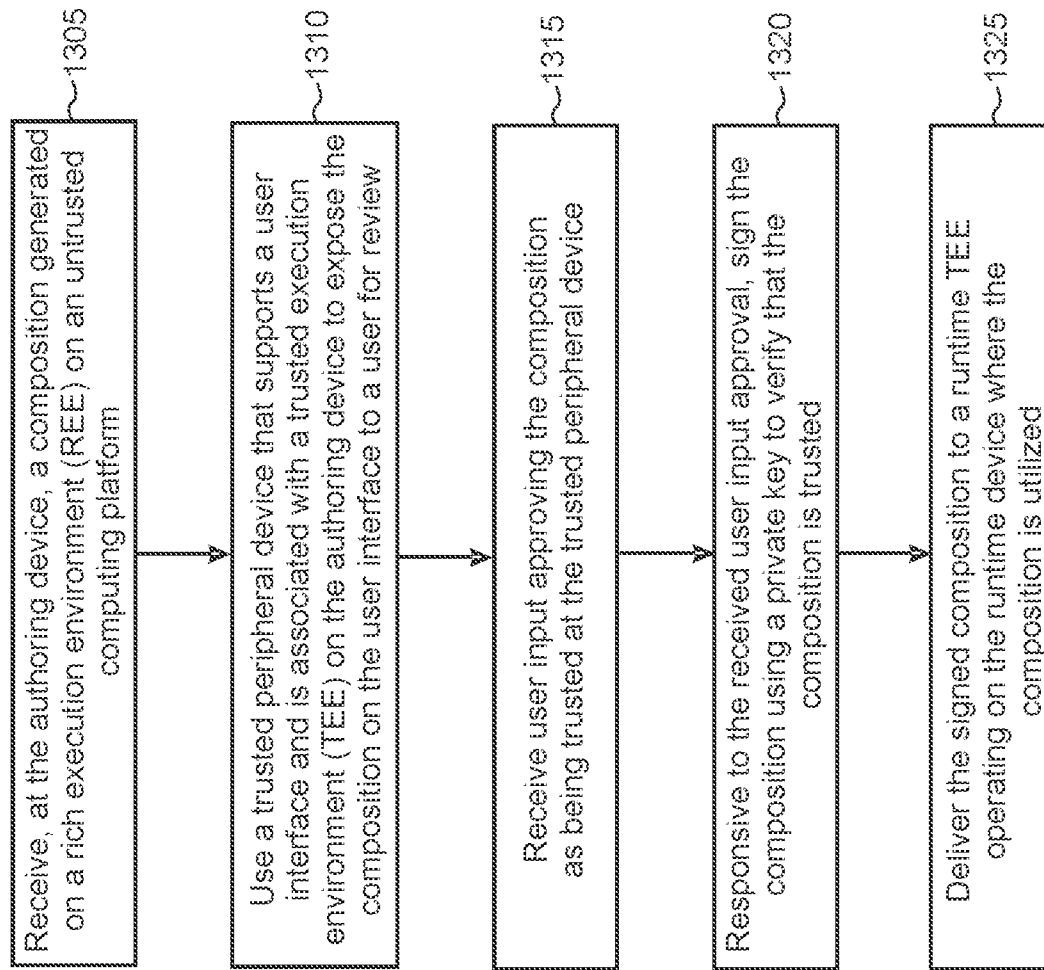


FIG 12



10/10

FIG 13



INTERNATIONAL SEARCH REPORT

International application No
PCT/US2020/020243

A. CLASSIFICATION OF SUBJECT MATTER

INV. G06F21/53 G06F21/57 G06F21/83 G06F21/84
ADD.

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
G06F

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

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

EPO-Internal, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	J. G. Beekman ET AL: "Improving Cloud Security using Secure Enclaves", UC Berkeley Electronic Theses and Dissertations, 1 January 2016 (2016-01-01), XP055509465, Retrieved from the Internet: URL:https://jbeekman.nl/publications/phd-dissertation.pdf [retrieved on 2018-09-25]	1-6,9-15
Y	section 5.5	7,8
Y	US 2015/310427 A1 (YI DONG KEUN [KR] ET AL) 29 October 2015 (2015-10-29) paragraphs [0041] - [0087]; figures 2,3a,3b,9a,9b	7,8
	----- -/-	



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

28 May 2020

Date of mailing of the international search report

08/06/2020

Name and mailing address of the ISA/

European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040,
Fax: (+31-70) 340-3016

Authorized officer

Rey, Salvador

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2020/020243

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	THOMAS NEUBAUER ET AL: "A Roadmap for Personal Identity Management", SYSTEMS (ICONS), 2010 FIFTH INTERNATIONAL CONFERENCE ON, IEEE, PISCATAWAY, NJ, USA, 11 April 2010 (2010-04-11), pages 134-139, XP031676115, ISBN: 978-1-4244-6231-5 page 4, left-hand column -----	7,8

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No

PCT/US2020/020243

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2015310427 A1	29-10-2015	CN 105046488 A	11-11-2015
		GB 2527189 A	16-12-2015
		US 2015310427 A1	29-10-2015
