



- (51) International Patent Classification:  
*G06F 21/50* (2013.01)
- (21) International Application Number:  
PCT/US2013/048036
- (22) International Filing Date:  
27 June 2013 (27.06.2013)
- (25) Filing Language: English
- (26) Publication Language: English
- (30) Priority Data:  
13/690,401 30 November 2012 (30.11.2012) US
- (71) Applicant: INTEL CORPORATION [US/US]; 2200 Mission College Boulevard, Santa Clara, California 95054 (US).
- (72) Inventors: DEWAN, Prashant; 119 NE Atlantic Place, Hillsboro, Oregon 97124 (US). SAVAGAONKAR, Uday R.; 5507 NW 133rd Avenue, Portland, Oregon 97229 (US). DURHAM, David M.; 20431 SW Tremont Way, Beaverton, Oregon 97007 (US). SCHMITZ, Paul S.; 30852 NW Brooking Court, North Plains, Oregon 97133 (US). MARTIN, Jason; 6248 SW 153rd Avenue, Beaverton, Oregon 97007 (US). GOLDSMITH, Michael A.; 17905 Kelok, Lake Oswego, Oregon 97034 (US). SAHITA, Ravi; 5314 NW 131st Avenue, Portland, Oregon 97229 (US). MCKEEN, Francis X.; 10612 NW LeMans Court, Portland, Oregon 97229 (US). ROZAS, Carlos V.; 1534 NW Morgan Lane, Portland, Oregon 97229 (US). VEMBU, Balaji; 1144, Bozio Court, Folsom, California 95630 (US). SCOTT, Janus; 5631 Birkdale Court, Rocklin, California 95677 (US). STRONGIN, Geoffrey S.; 11575 SW Camden Lane, Beaverton, Oregon 97008 (US). KANG, Xiaozhu; 4006 Dill Terrace, Fremont, California 94538 (US). GREWAL, Karanvir S.;

2631 NE 9th Drive, Hillsboror, Oregon 97124 (US). **CHH-ABRA, Siddhartha**; 1221 NE 51st Avenue, Apt. 30, Hillsboro, Oregon 97124 (US). **TRIVEDI, Alpa T. Narendra**; 6434 NE Meadowgate Court, Hillsboro, Oregon 97124 (US).

(74) Agent: **TROP, Timothy N.**; Trop, Pruner & Hu, P.C., 1616 S. Voss Rd., Ste. 750, Houston, Texas 77057-2631 (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, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, 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, 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, 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 the identity of the inventor (Rule 4.17(i))

**Published:**

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

(54) Title: SECURE ENVIRONMENT FOR GRAPHICS PROCESSING UNITS

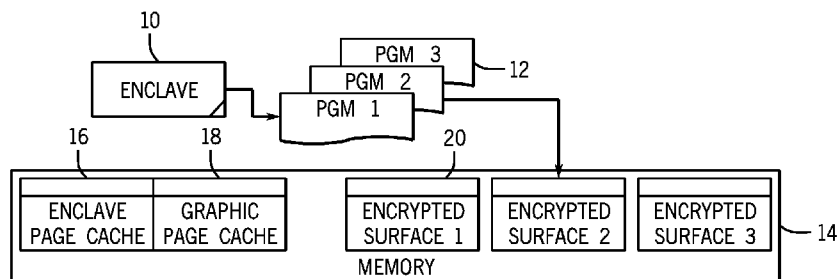


FIG. 1

(57) Abstract: In accordance with some embodiments, a protected execution environment may be defined for a graphics processing unit. This framework not only protects the workloads from malware running on the graphics processing unit but also protects those workloads from malware running on the central processing unit. In addition, the trust framework may facilitate proof of secure execution by measuring the code and data structures used to execute the workload. If a part of the trusted computing base of this framework or protected execution environment is compromised, that part can be patched remotely and the patching can be proven remotely throughout attestation in some embodiments.

WO 2014/084914 A1

## SECURE ENVIRONMENT FOR GRAPHICS PROCESSING UNITS

### Background

[0001] This relates generally to processing graphics sometimes called graphics processors or graphics processing units.

[0002] Processing graphics are increasingly being exposed to workloads which require some degree of security. Security sensitive workloads on processing graphics have relied on the operating system to provide the necessary security. However, the increasing number of malware attacks suggest that these solutions are not meeting the security requirements for a number of workloads.

[0003] Examples of workloads that may need trust computation frameworks on graphics processing units include bank transactions where a browser offloads part of a transaction to a graphics engine, antivirus engines where part of the pattern matching is offloaded to graphics engine, and medical imaging. In addition many non-security workloads need added security, such as computer aided design, and graphical and other workloads that need unhampered execution in the presence of malware. In addition, security sensitive workloads are being offloaded to processing graphics due to the power, efficiency and performance improvements achieved by graphics processors. Examples may include cryptographic functions, pattern matching primitives, and face detection algorithms as well as certain workloads for mining, oil refineries, financial calculations and other calculations involving money.

[0004] Security sensitive workloads may need a trust framework on processing graphics that not only enforces the correctness of execution of the workload but also enables strict access control of the graphics assets to only authorized entities.

### Brief Description Of The Drawings

[0005] Some embodiments are described with respect to the following figures:

Figure 1 is a pictorial diagram of one embodiment of the present invention;

Figure 2 is a diagram of the life cycle of a protected graphics module in accordance with one embodiment;

Figure 3 is a flow chart for one embodiment of the present invention;

Figure 4 is a system depiction for one embodiment; and

Figure 5 is a front elevational view of one embodiment.

#### Detailed Description

[0006] In accordance with some embodiments, a protected execution environment may be defined for processing graphics. This framework not only protects the workloads from malware running on the graphics processing unit but also protects those workloads from malware running on the central processing unit. In addition, the trust framework may facilitate proof of secure execution by measuring the code and data structures used to execute the workload. If a part of the trusted computing base of this framework or protected execution environment is compromised, that part can be patched remotely and the patching can be proven remotely throughout attestation in some embodiments.

[0007] Referring to Figure 1, protected graphics modules, resident on processing graphics, contain code, data, and states needed for correct execution of a workload in the processing graphics. Similar to non-secure graphics workloads, the protected graphics module is created by software running on the central processing unit (CPU) by special CPU instructions. However, the protected graphics module can only be executed on the graphics processing unit in some embodiments. It may have the ability to assert correctness of execution to remote parties. In one embodiment, the module relies on the Secure Enclaves infrastructure to provide this assertion. See Patent Cooperation Treaty published application number WO/2010/057065, published on 20.05.2010.

[0008] In Secure Enclaves, a protected execution environment is built inside an application. An operating system constructs an enclave using a set of privileged instructions. Once the enclave is constructed, the application can enter and exit the enclave using a set of unprivileged instructions. An enclave executes in a physically protected area of memory called the enclave page cache. The hardware ensures that memory pages belonging to the enclave page cache can only be accessed by the enclave that owns those pages, and also ensures that malicious privileged

software cannot redirect memory accesses originating from an enclave in an unexpected fashion. Software executing inside an enclave can prove that the enclave was constructed correctly by the operating system via hardware based attestation.

[0009] The graphics page cache 18 in the memory 14 holds code and data used by the protected graphics modules (PGMs) 12. In one embodiment this section of memory is implemented in the system dynamic random access memory (DRAM) and is cryptographically protected by a memory encryption engine. In yet another embodiment, this section of memory is implemented inside the processor package as static random access memory (SRAM) or embedded DRAM (eDRAM). This section of memory may be managed by the operating system graphics driver using a set of privileged CPU instructions. It may not be accessible to either of them for purposes of read or write or execute. The modules 12 reside inside the graphics page cache 18. In one embodiment, the graphics page cache may be combined with an enclave page cache 16.

[0010] The enclave 10 is responsible for creating and initializing the modules 12. Once a module is created and initialized by the enclave 10, various hardware engines on the processing graphics can enter the module 12 using specific entry points. Additionally, after module initiation, the enclave itself can carry out portions of the secure computation, and can communicate with the module via the graphics page cache 18. In this regard, the enclave 10 is the central processing unit (CPU) 24 counterpart of the modules 12 and it can have a one-to-many relationship with the modules 12.

[0011] The enclave 10 and modules 12 are both optional embodiments from the perspective of a developer. The application developer may decide whether an application needs a level of security provided by the combination of enclave and protected graphics module 12 in some embodiments.

[0012] Thus, within the memory, may be an enclave page cache 16, graphics page cache 18 including the modules 12 and the enclave 10 and one or more encrypted surfaces 20 to hold necessary data and/or instructions.

[0013] The processing graphics 26, shown in Figure 2, includes the protected graphics module 12 while the application 28 is resident within the CPU 24. The CPU 24 may include a just in time execution engine (jitter) 30 and the enclave 32. It may also support a kernel driver 34. Shared virtual memory 36 may be shared between the central processing unit 24 and the processing graphics 26. Shared system memory 38 may store the enclave page cache 16 and graphics page cache 18.

[0014] Initially, the application 28 is launched on the CPU 24. This application is typically a ring three application launched in a process of its own. The application creates an enclave 32 and the enclave page cache 16. The enclave may contain metadata and metacode to create the protected graphics modules 12. The enclave also contains the jitter 30 for converting the module metacode from a high level language to the binary format that can be recognized by the processor and graphics hardware. The enclave also contains metadata and metacode to create the modules 12.

[0015] To create a module 12, the enclave loads metacode and metadata from the graphics workload and measures or verifies the workload using appropriate crypto operations. This is indicated by the arrows 1 and 2 in Figure 2. Then the enclave sets up the processor graphics state and the data structures needed for the execution of the module 12 in the graphics page cache 18.

[0016] The invocation of the module proceeds as follows. The enclave requests through the application, that the kernel mode graphics driver 34 execute the module by providing it to relevant metadata. This is indicated by the arrow 3 in Figure 2. The kernel mode graphics driver 34 (ring zero) takes the metadata and sets up the graphics engine state as per the metadata (arrow 4). The kernel graphics driver then points the processing graphics to the entry point of the module 12 via ring buffer 37 as indicated by the arrow 5. The kernel mode graphics driver maps the graphics page cache into the graphics address space by appropriately modifying the shared virtual memory 36 tables.

[0017] The processing graphics 26 executes the module 12 as indicated by the arrow 6. The graphics scheduler 41, within the processing graphics 26, informs the

driver 34, which then closes the access to the graphics page cache provided for execution of the module 12. When the module 12 is scheduled for execution again, the graphics driver provides the graphics page cache access to the module and the module is executed. After the module completes its execution, it writes its results 39 of the execution to the output address inside the graphics page cache as indicated by the arrow 6. As specified in the metadata during the creation of the module 12, the fronting enclave subsequently reads the output and provides it to the central processing unit code executing inside the enclave.

[0018] In accordance with one embodiment, the module 12 may have a well-defined format memory and the format is used to measure the unique components of the module and bind them to the results. The microcode measures the module at the time of execution by the enclave and provides a cryptographically signed measurement to the enclave. The enclave includes the measurement of the module and the measurement of the enclave and the combined data is used for attestation using the enclave attestation protocol.

[0019] In another embodiment, the module has an arbitrary format and is only understood by a loader inside the enclave. The enclave uses public key cryptography to verify the source of the module blob. The fact that the enclave enforces publicly verification is implicitly attested to by the enclave's measurement.

[0020] Even though the embodiment above pertains to processing graphics, these procedures can be applied to any co-processor/accelerator/device including crypto accelerators integrated on the same chip as the central processing unit.

[0021] Referring next to Figure 3, a sequence depicted there may be implemented in software, firmware and/or hardware. In software and firmware embodiments it may be implemented by computer executed instructions stored in one or more non-transitory computer readable media such as magnetic, optical or semiconductor storages.

[0022] The sequence begins at block 40 when the system starts up and both the central processing unit and the protected graphics are configured with a protected enclave page cache. Then, in block 42, the central processing unit launches a

enclave on the request of an application. The enclave may be protected from any untrusted CPU or protected graphics code by the hardware in the processor. Then, as shown in block 44, the enclave loads protected graphics module inside the page cache, translates the module code to module binary, sets up the data and then submits the module to the processing graphics for execution.

[0023] The graphics scheduler or command streamer 41, which is a trusted entity, gets a context, identifies it as protected context, sets the graphics hardware for protected context, disables the range registers for the enclave page cache for this context, reads the state from the context and starts executing the context as indicated in block 46. Then at diamond 48 a check determines whether the context leads to a fault such as page fault, misconfiguration, or access permissions. If so, the scheduler writes the intermediate state into an encrypted memory buffer, disables the range registers, and loads the next context as indicated in block 52. Then the operating system processes the fault by clearing the condition that caused the fault as indicated in block 54. The scheduler picks up the swapped out protected graphics module and schedules it back in as indicated in block 56. Then the flow returns back to the check at diamond 48.

[0024] If the context does not lead to a fault, then the context writes the result to the enclave page cache and the schedule swaps the context out, disables a range register, and cleans the hardware state and picks up the next context as indicated in block 58. The workload might decide to send its results to the display engine.

[0025] The scheduler that schedules the protected graphics mode module may be trusted by the protected graphics module and may enjoy the same protections as the protected graphics module or it cannot read and write the protected graphics module but schedules it like a black box. The scheduler may be a software scheduler or a hardware scheduler or a combination of both. The enclave infrastructure may be one embodiment for trusted creation or execution of the protected graphics modules. The protected graphics modules can also be created in a trusted cloud environment and then executed on a client. While the preceding discussion emphasizes a graphics device, however, in other embodiments any device that has computation capabilities that can be used as an offload device for computation or for security may

benefit from the principles described herein. The modules may be provisioned with secrets after module distribution to customers.

[0026] Figure 4 illustrates an embodiment of a system 300. In embodiments, system 300 may be a media system although system 300 is not limited to this context. For example, system 300 may be incorporated into a personal computer (PC), laptop computer, ultra-laptop computer, tablet, touch pad, portable computer, handheld computer, palmtop computer, personal digital assistant (PDA), cellular telephone, combination cellular telephone/PDA, television, smart device (e.g., smart phone, smart tablet or smart television), mobile internet device (MID), messaging device, data communication device, and so forth.

[0027] In embodiments, system 300 comprises a platform 302 coupled to a display 320. Platform 302 may receive content from a content device such as content services device(s) 330 or content delivery device(s) 340 or other similar content sources. A navigation controller 350 comprising one or more navigation features may be used to interact with, for example, platform 302 and/or display 320. Each of these components is described in more detail below.

[0028] In embodiments, platform 302 may comprise any combination of a chipset 305, processor 310, memory 312, storage 314, graphics subsystem 315, applications 316 and/or radio 318. Chipset 305 may provide intercommunication among processor 310, memory 312, storage 314, graphics subsystem 315, applications 316 and/or radio 318. For example, chipset 305 may include a storage adapter (not depicted) capable of providing intercommunication with storage 314.

[0029] Processor 310 may be implemented as Complex Instruction Set Computer (CISC) or Reduced Instruction Set Computer (RISC) processors, x86 instruction set compatible processors, multi-core, or any other microprocessor or central processing unit (CPU). In embodiments, processor 310 may comprise dual-core processor(s), dual-core mobile processor(s), and so forth.



[0030] Memory 312 may be implemented as a volatile memory device such as, but not limited to, a Random Access Memory (RAM), Dynamic Random Access Memory (DRAM), or Static RAM (SRAM).

[0031] Storage 314 may be implemented as a non-volatile storage device such as, but not limited to, a magnetic disk drive, optical disk drive, tape drive, an internal storage device, an attached storage device, flash memory, battery backed-up SDRAM (synchronous DRAM), and/or a network accessible storage device. In embodiments, storage 314 may comprise technology to increase the storage performance enhanced protection for valuable digital media when multiple hard drives are included, for example.

[0032] Graphics subsystem 315 may perform processing of images such as still or video for display. Graphics subsystem 315 may be a graphics processing unit (GPU) or a visual processing unit (VPU), for example. An analog or digital interface may be used to communicatively couple graphics subsystem 315 and display 320. For example, the interface may be any of a High-Definition Multimedia Interface, DisplayPort, wireless HDMI, and/or wireless HD compliant techniques. Graphics subsystem 315 could be integrated into processor 310 or chipset 305. Graphics subsystem 315 could be a stand-alone card communicatively coupled to chipset 305.

[0033] The graphics and/or video processing techniques described herein may be implemented in various hardware architectures. For example, graphics and/or video functionality may be integrated within a chipset. Alternatively, a discrete graphics and/or video processor may be used. As still another embodiment, the graphics and/or video functions may be implemented by a general purpose processor, including a multi-core processor. In a further embodiment, the functions may be implemented in a consumer electronics device.

[0034] Radio 318 may include one or more radios capable of transmitting and receiving signals using various suitable wireless communications techniques. Such techniques may involve communications across one or more wireless networks. Exemplary wireless networks include (but are not limited to) wireless

local area networks (WLANs), wireless personal area networks (WPANs), wireless metropolitan area network (WMANs), cellular networks, and satellite networks. In communicating across such networks, radio 318 may operate in accordance with one or more applicable standards in any version.

[0035] In embodiments, display 320 may comprise any television type monitor or display. Display 320 may comprise, for example, a computer display screen, touch screen display, video monitor, television-like device, and/or a television. Display 320 may be digital and/or analog. In embodiments, display 320 may be a holographic display. Also, display 320 may be a transparent surface that may receive a visual projection. Such projections may convey various forms of information, images, and/or objects. For example, such projections may be a visual overlay for a mobile augmented reality (MAR) application. Under the control of one or more software applications 316, platform 302 may display user interface 322 on display 320.

[0036] In embodiments, content services device(s) 330 may be hosted by any national, international and/or independent service and thus accessible to platform 302 via the Internet, for example. Content services device(s) 330 may be coupled to platform 302 and/or to display 320. Platform 302 and/or content services device(s) 330 may be coupled to a network 360 to communicate (e.g., send and/or receive) media information to and from network 360. Content delivery device(s) 340 also may be coupled to platform 302 and/or to display 320.

[0037] In embodiments, content services device(s) 330 may comprise a cable television box, personal computer, network, telephone, Internet enabled devices or appliance capable of delivering digital information and/or content, and any other similar device capable of unidirectionally or bidirectionally communicating content between content providers and platform 302 and/display 320, via network 360 or directly. It will be appreciated that the content may be communicated unidirectionally and/or bidirectionally to and from any one of the components in system 300 and a content provider via network 360. Examples of content may include any media information including, for example, video, music, medical and gaming information, and so forth.

[0038] Content services device(s) 330 receives content such as cable television programming including media information, digital information, and/or other content. Examples of content providers may include any cable or satellite television or radio or Internet content providers. The provided examples are not meant to limit embodiments of the invention.

[0039] In embodiments, platform 302 may receive control signals from navigation controller 350 having one or more navigation features. The navigation features of controller 350 may be used to interact with user interface 322, for example. In embodiments, navigation controller 350 may be a pointing device that may be a computer hardware component (specifically human interface device) that allows a user to input spatial (e.g., continuous and multi-dimensional) data into a computer. Many systems such as graphical user interfaces (GUI), and televisions and monitors allow the user to control and provide data to the computer or television using physical gestures.

[0040] Movements of the navigation features of controller 350 may be echoed on a display (e.g., display 320) by movements of a pointer, cursor, focus ring, or other visual indicators displayed on the display. For example, under the control of software applications 316, the navigation features located on navigation controller 350 may be mapped to virtual navigation features displayed on user interface 322, for example. In embodiments, controller 350 may not be a separate component but integrated into platform 302 and/or display 320. Embodiments, however, are not limited to the elements or in the context shown or described herein.

[0041] In embodiments, drivers (not shown) may comprise technology to enable users to instantly turn on and off platform 302 like a television with the touch of a button after initial boot-up, when enabled, for example. Program logic may allow platform 302 to stream content to media adaptors or other content services device(s) 330 or content delivery device(s) 340 when the platform is turned "off." In addition, chip set 305 may comprise hardware and/or software support for 5.1 surround sound audio and/or high definition 7.1 surround sound audio, for example. Drivers may include a graphics driver for integrated graphics platforms. In

embodiments, the graphics driver may comprise a peripheral component interconnect (PCI) Express graphics card.

[0042] In various embodiments, any one or more of the components shown in system 300 may be integrated. For example, platform 302 and content services device(s) 330 may be integrated, or platform 302 and content delivery device(s) 340 may be integrated, or platform 302, content services device(s) 330, and content delivery device(s) 340 may be integrated, for example. In various embodiments, platform 302 and display 320 may be an integrated unit. Display 320 and content service device(s) 330 may be integrated, or display 320 and content delivery device(s) 340 may be integrated, for example. These examples are not meant to limit the invention.

[0043] In various embodiments, system 300 may be implemented as a wireless system, a wired system, or a combination of both. When implemented as a wireless system, system 300 may include components and interfaces suitable for communicating over a wireless shared media, such as one or more antennas, transmitters, receivers, transceivers, amplifiers, filters, control logic, and so forth. An example of wireless shared media may include portions of a wireless spectrum, such as the RF spectrum and so forth. When implemented as a wired system, system 300 may include components and interfaces suitable for communicating over wired communications media, such as input/output (I/O) adapters, physical connectors to connect the I/O adapter with a corresponding wired communications medium, a network interface card (NIC), disc controller, video controller, audio controller, and so forth. Examples of wired communications media may include a wire, cable, metal leads, printed circuit board (PCB), backplane, switch fabric, semiconductor material, twisted-pair wire, co-axial cable, fiber optics, and so forth.

[0044] Platform 302 may establish one or more logical or physical channels to communicate information. The information may include media information and control information. Media information may refer to any data representing content meant for a user. Examples of content may include, for example, data from a voice conversation, videoconference, streaming video, electronic mail ("email") message, voice mail message, alphanumeric symbols, graphics, image, video, text and so

forth. Data from a voice conversation may be, for example, speech information, silence periods, background noise, comfort noise, tones and so forth. Control information may refer to any data representing commands, instructions or control words meant for an automated system. For example, control information may be used to route media information through a system, or instruct a node to process the media information in a predetermined manner. The embodiments, however, are not limited to the elements or in the context shown or described in Figure 4.

[0045] As described above, system 300 may be embodied in varying physical styles or form factors. Figure 5 illustrates embodiments of a small form factor device 400 in which system 300 may be embodied. In embodiments, for example, device 400 may be implemented as a mobile computing device having wireless capabilities. A mobile computing device may refer to any device having a processing system and a mobile power source or supply, such as one or more batteries, for example.

[0046] As described above, examples of a mobile computing device may include a personal computer (PC), laptop computer, ultra-laptop computer, tablet, touch pad, portable computer, handheld computer, palmtop computer, personal digital assistant (PDA), cellular telephone, combination cellular telephone/PDA, television, smart device (e.g., smart phone, smart tablet or smart television), mobile internet device (MID), messaging device, data communication device, and so forth.

[0047] Examples of a mobile computing device also may include computers that are arranged to be worn by a person, such as a wrist computer, finger computer, ring computer, eyeglass computer, belt-clip computer, arm-band computer, shoe computers, clothing computers, and other wearable computers. In embodiments, for example, a mobile computing device may be implemented as a smart phone capable of executing computer applications, as well as voice communications and/or data communications. Although some embodiments may be described with a mobile computing device implemented as a smart phone by way of example, it may be appreciated that other embodiments may be implemented using other wireless mobile computing devices as well. The embodiments are not limited in this context.

[0048] The processor 310 may communicate with a camera 322 and a global positioning system sensor 320, in some embodiments. A memory 312, coupled to the processor 310, may store computer readable instructions for implementing the sequences shown in Figures 1 and 2 in software and/or firmware embodiments. Particularly the sequences may be implemented by one or more non-transitory storage devices storing computer implemented instructions.

[0049] As shown in Figure 5, device 400 may comprise a housing 402, a display 404, an input/output (I/O) device 406, and an antenna 408. Device 400 also may comprise navigation features 412. Display 404 may comprise any suitable display unit for displaying information appropriate for a mobile computing device. I/O device 406 may comprise any suitable I/O device for entering information into a mobile computing device. Examples for I/O device 406 may include an alphanumeric keyboard, a numeric keypad, a touch pad, input keys, buttons, switches, rocker switches, microphones, speakers, voice recognition device and software, and so forth. Information also may be entered into device 400 by way of microphone. Such information may be digitized by a voice recognition device. The embodiments are not limited in this context.

[0050] Various embodiments may be implemented using hardware elements, software elements, or a combination of both. Examples of hardware elements may include processors, microprocessors, circuits, circuit elements (e.g., transistors, resistors, capacitors, inductors, and so forth), integrated circuits, application specific integrated circuits (ASIC), programmable logic devices (PLD), digital signal processors (DSP), field programmable gate array (FPGA), logic gates, registers, semiconductor device, chips, microchips, chip sets, and so forth. Examples of software may include software components, programs, applications, computer programs, application programs, system programs, machine programs, operating system software, middleware, firmware, software modules, routines, subroutines, functions, methods, procedures, software interfaces, application program interfaces (API), instruction sets, computing code, computer code, code segments, computer code segments, words, values, symbols, or any combination thereof. Determining whether an embodiment is implemented using hardware elements and/or software

elements may vary in accordance with any number of factors, such as desired computational rate, power levels, heat tolerances, processing cycle budget, input data rates, output data rates, memory resources, data bus speeds and other design or performance constraints.

[0051] One or more aspects of at least one embodiment may be implemented by representative instructions stored on a machine-readable medium which represents various logic within the processor, which when read by a machine causes the machine to fabricate logic to perform the techniques described herein. Such representations, known as "IP cores" may be stored on a tangible, machine readable medium and supplied to various customers or manufacturing facilities to load into the fabrication machines that actually make the logic or processor.

[0052] The foregoing description of one or more implementations provides illustration and description, but is not intended to be exhaustive or to limit the scope of the invention to the precise form disclosed. Modifications and variations are possible in light of the above teachings or may be acquired from practice of various implementations of the invention.

[0053] The graphics processing techniques described herein may be implemented in various hardware architectures. For example, graphics functionality may be integrated within a chipset. Alternatively, a discrete processing graphics may be used. As still another embodiment, the graphics functions may be implemented by a general purpose processor, including a multicore processor.

[0054] References throughout this specification to "one embodiment" or "an embodiment" mean that a particular feature, structure, or characteristic described in connection with the embodiment is included in at least one implementation encompassed within the present invention. Thus, appearances of the phrase "one embodiment" or "in an embodiment" are not necessarily referring to the same embodiment. Furthermore, the particular features, structures, or characteristics may be instituted in other suitable forms other than the particular embodiment illustrated and all such forms may be encompassed within the claims of the present application.

[0055] While the present invention has been described with respect to a limited number of embodiments, those skilled in the art will appreciate numerous modifications and variations therefrom. It is intended that the appended claims cover all such modifications and variations as fall within the true spirit and scope of this present invention.



What is claimed is:

- 1 1. A method comprising:  
2 creating a trusted framework for another processing unit on a central  
3 processing unit by providing an enclave on the central processing unit to build a  
4 protected module on the another processing unit.
  
- 1 2. The method of claim 1 including enabling the enclave to communicate with  
2 the module via a cache in memory shared between said central processing unit and  
3 said another processing unit.
  
- 1 3. The method of claim 2 including storing code and data used by said module in  
2 said cache.
  
- 1 4. The method of claim 3 including creating said cache in memory inaccessible  
2 by an operating system on said central processing unit.
  
- 1 5. The method of claim 1 including executing an application on said central  
2 processing unit to launch said enclave.
  
- 1 6. The method of claim 5 including using said enclave to convert module code to  
2 binary code.
  
- 1 7. The method of claim 1 including verifying a workload in said enclave and  
2 loading code and data from said workload to create said module.
  
- 1 8. The method of claim 1 including causing an untrusted kernel driver to execute  
2 enclave supplied code to invoke said module.
  
- 1 9. The method of claim 1 including causing the module to write execution results  
2 to said enclave.

1 10. The method of claim 1 including creating a framework for another processing  
2 unit including processing graphics.

1 11. The method of claim 1 including asserting correctness of the module to a  
2 remote party.

1 12. The method of claim 1 including provisioning the module with secrets after  
2 distribution.

1 13. The method of claim 1 including protecting the enclave from an untrusted  
2 central processing unit.

1 14. One or more non-transitory computer readable media storing instructions  
2 executed by a central processing unit to perform a sequence comprising:  
3 creating a trusted framework for another processing unit on the central  
4 processing unit by providing an enclave on the central processing unit to build a  
5 protected module on the another processing unit.

1 15. The media of claim 14 further storing instructions to perform a sequence  
2 including enabling the enclave to communicate with the module via a cache in  
3 memory shared between said central processing unit and said another processing  
4 unit.

1 16. The media of claim 15 further storing instructions to perform a sequence  
2 including storing code and data used by said module in said cache.

1 17. The media of claim 16 further storing instructions to perform a sequence  
2 including creating said cache in memory inaccessible by an operating system on  
3 said central processing unit.

1 18. The media of claim 14 further storing instructions to perform a sequence  
2 including verifying a workload in said enclave and loading code and data from said  
3 workload to create said module.

1 19. The media of claim 14 further storing instructions to perform a sequence  
2 including causing an untrusted kernel driver to execute enclave supplied code to  
3 invoke said module.

1 20. The media of claim 14 further storing instructions to perform a sequence  
2 including causing the module to write execution results to said enclave.

1 21. The media of claim 14 further storing instructions to perform a sequence  
2 including creating a framework for another processing unit including processing  
3 graphics.

1 22. The media of claim 14 further storing instructions to perform a sequence  
2 including asserting correctness of the module to a remote party.

1 23. The media of claim 14 further storing instructions to perform a sequence  
2 including provisioning the module with secrets after distribution.

1 24. The media of claim 14 further storing instructions to perform a sequence  
2 including protecting the enclave from an untrusted central processing unit.

1 25. A computer comprising:  
2 another processing unit; and  
3 a central processing unit coupled said another processing unit to provide an  
4 enclave to build a protected module on the another processing unit using a trusted  
5 framework on the another processing unit.

1 26. The computer of claim 25 wherein said another processing unit includes a  
2 processing graphics.

1 27. The computer of claim 25 including a memory with a cache, said memory  
2 shared between the units, said enclave to communicate with the module via the  
3 cache.

1 28. The computer of claim 25 including an operating system.

1 29. The computer of claim 25 including a battery.

1 30. The computer of claim 25 including firmware and a module to update said  
2 firmware.

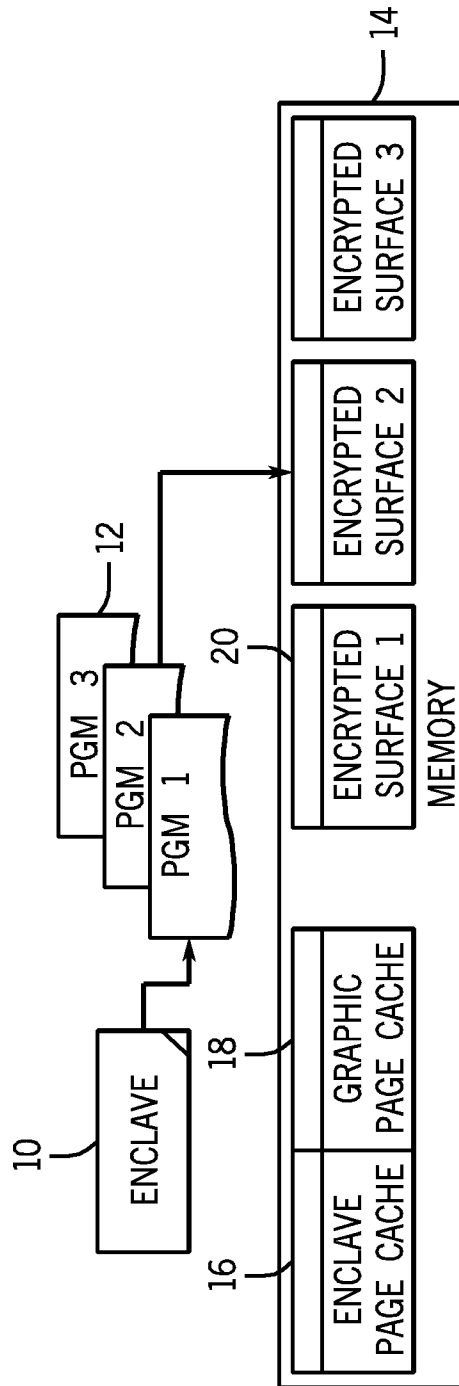


FIG. 1

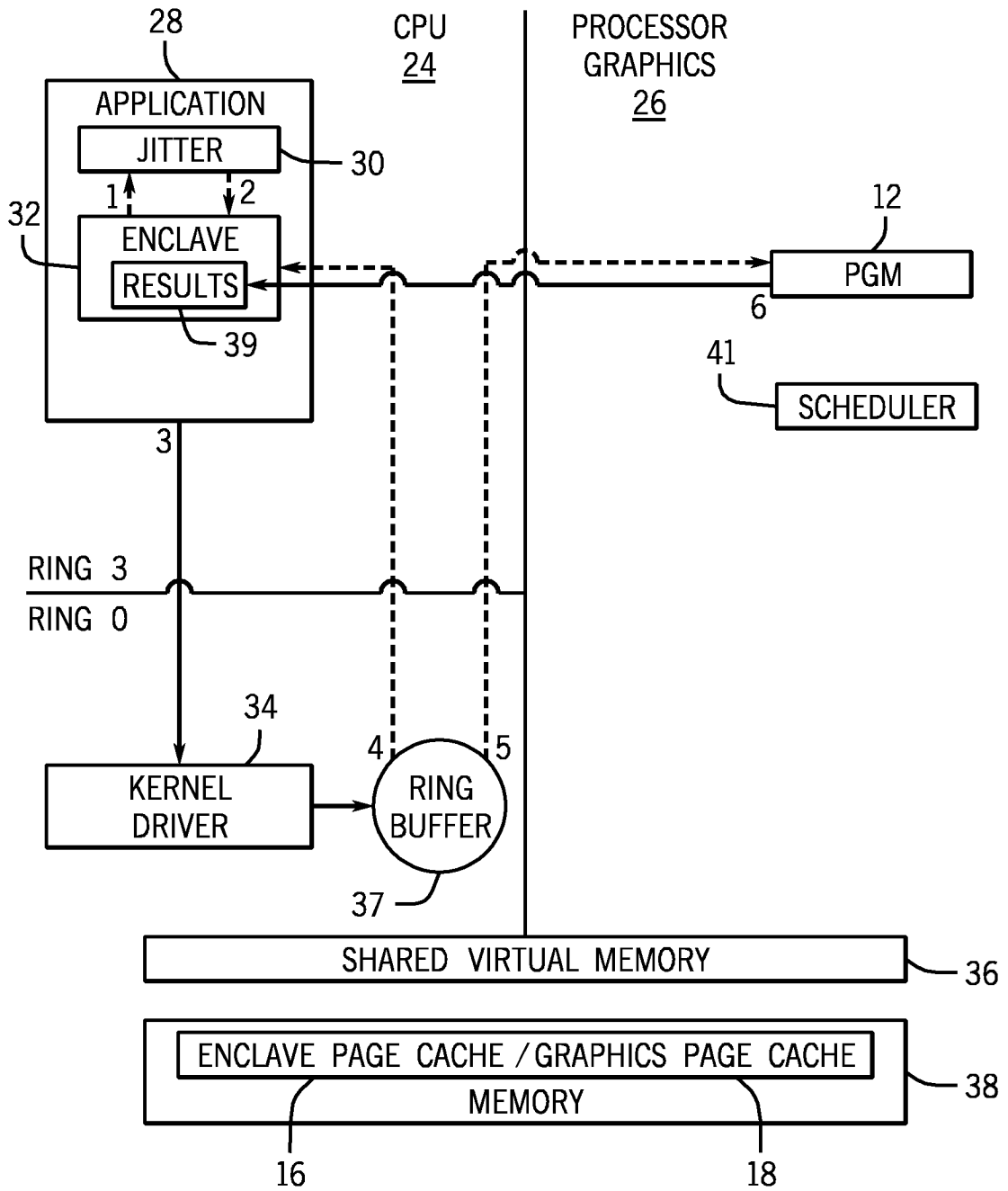


FIG. 2

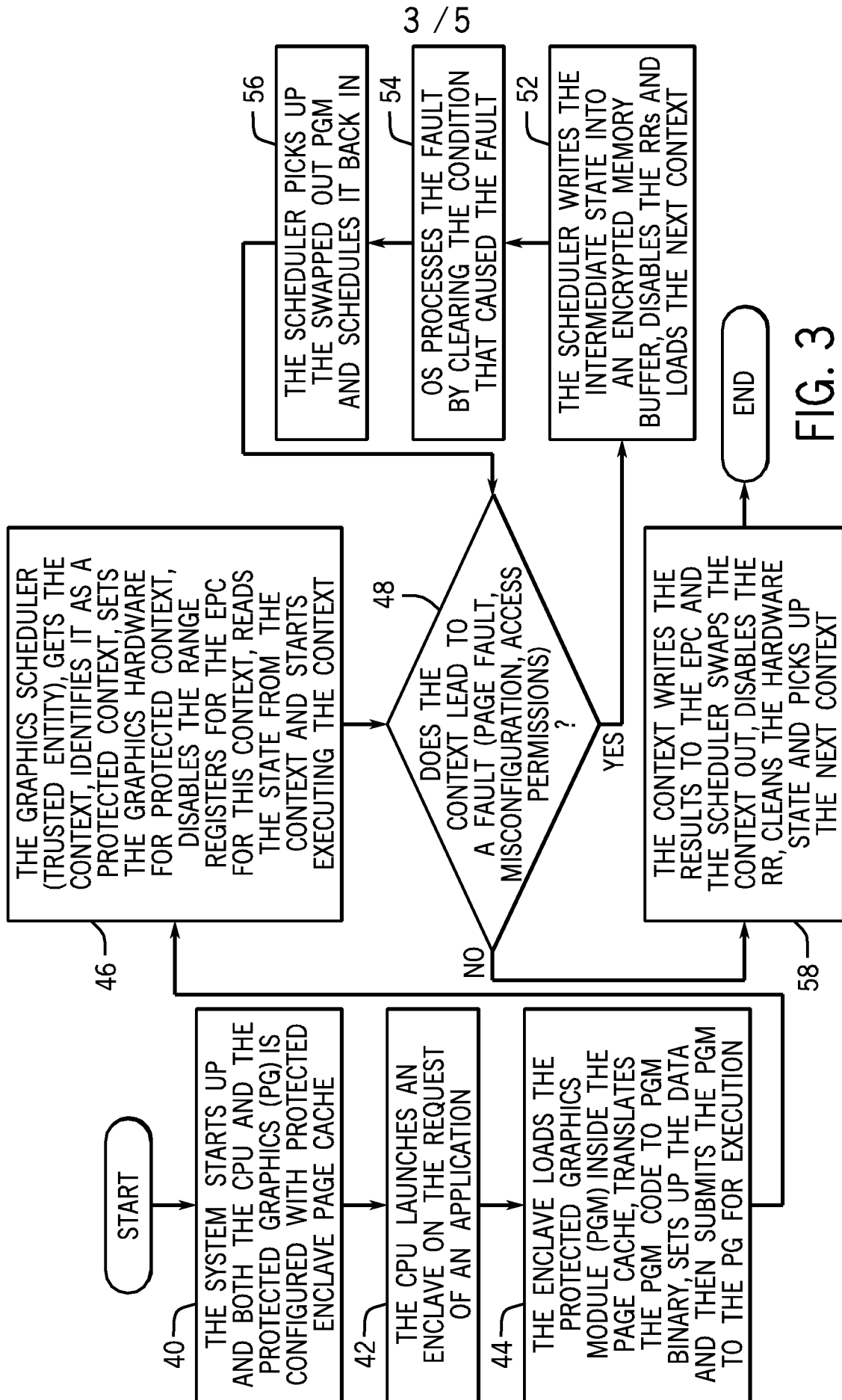


FIG. 3

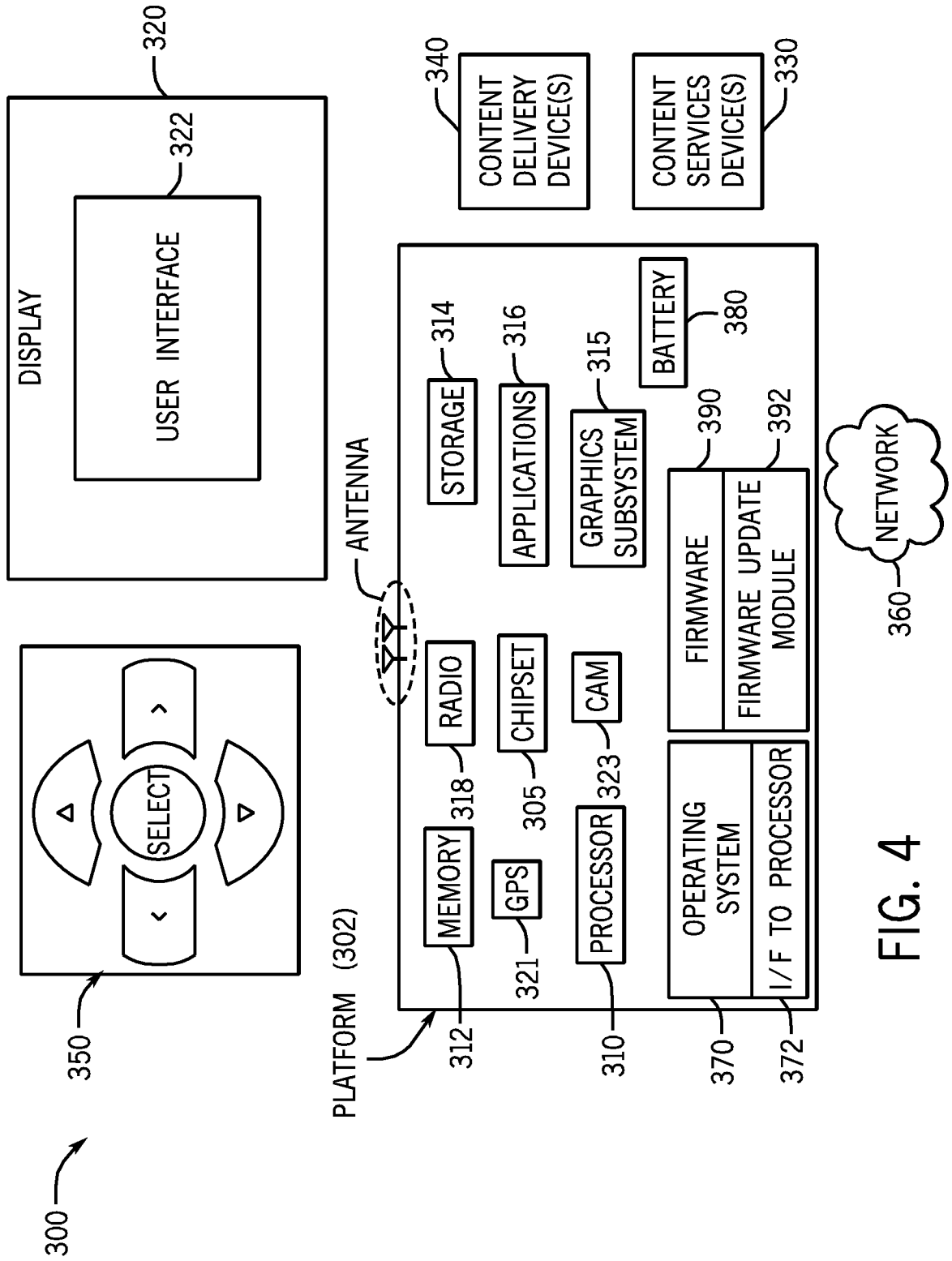


FIG. 4



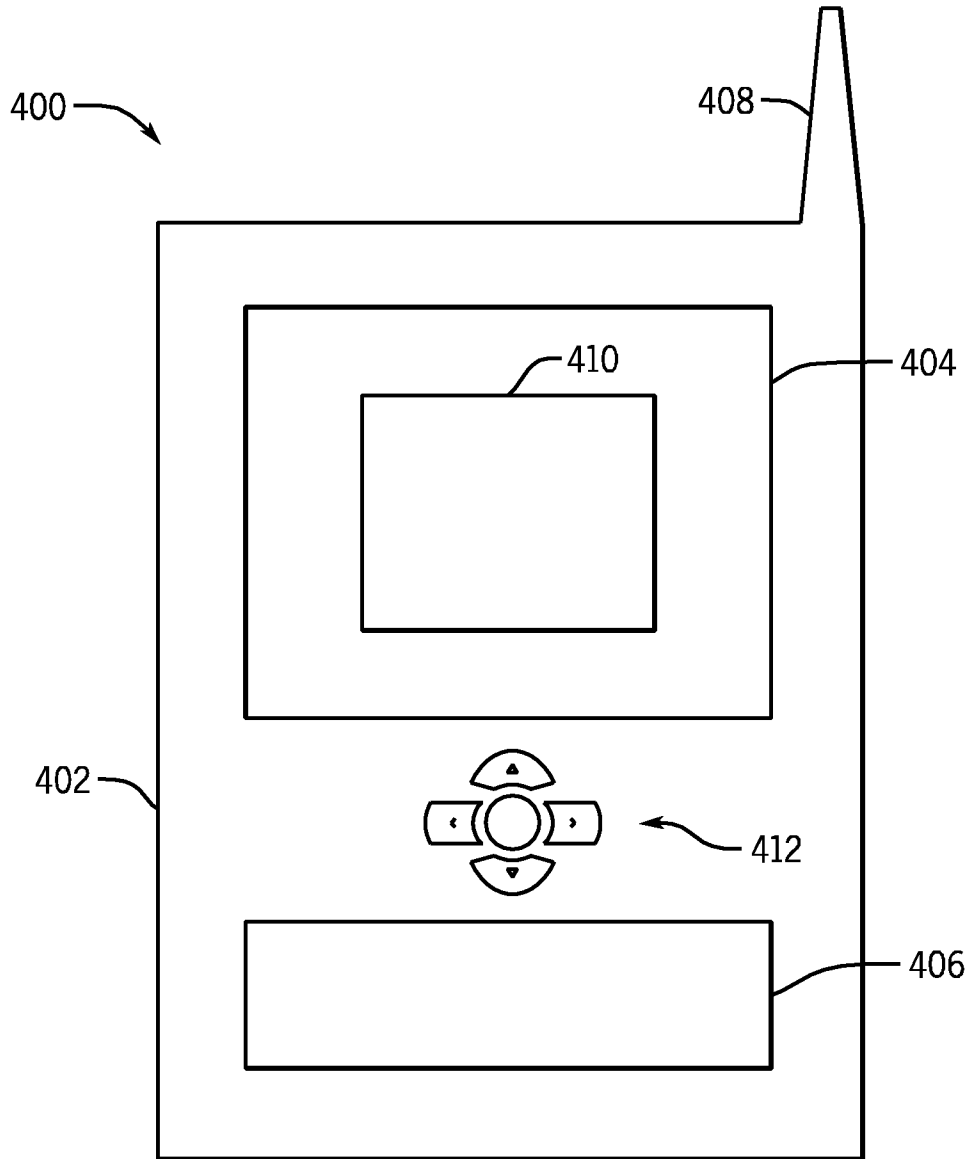


FIG. 5

**A. CLASSIFICATION OF SUBJECT MATTER****G06F 21/50(2013.01)i**

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

**B. FIELDS SEARCHED**

Minimum documentation searched (classification system followed by classification symbols)

G06F 21/50; G06F 21/24; H04L 9/30; H04L 29/06; G06F 21/00; G06F 21/22

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

Korean utility models and applications for utility models

Japanese utility models and applications for utility models

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

eKOMPASS(KIPO internal) &amp; Keywords: central processing unit, another processing unit, graphic processing unit, graphics processing, secure, trusted, protected module, enclave, cache

**C. DOCUMENTS CONSIDERED TO BE RELEVANT**

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y A	US 2012-0163589 A1 (SIMON P. JOHNSON et al.) 28 June 2012 See paragraphs [0001]-[0010], [0018]-[0032], [0036]-[0045], [0049]-[0062], [0083]-[0085]; claim 1; and figures 2-3.	1, 5, 10, 14, 21, 25-26 , 28-30  2-4, 6-9, 11-13 , 15-20, 22-24, 27
Y A	US 2010-0031342 A1 (ROBIN O. VOGSLAND) 04 February 2010 See paragraphs [0002]-[0014], [0030]-[0034], [0037]-[0039], [0042]; and claim 1.  KR 10-2012-0099472 A (INTEL CORPORATION) 10 September 2012 See paragraphs [0001]-[0002], [0004]-[0010], [0016]-[0058], [0073]-[0074], [0123]-[0129], [0147]-[0156], [0180]-[0181], [0191]-[0197]; claims 1, 15; and figure 1, 5.	1, 5, 10, 14, 21, 25-26 , 28-30  1-30
A	US 2012-0159184 A1 (SIMON P. JOHNSON et al.) 21 June 2012 See paragraphs [0001]-[0002], [0032]-[0034], [0037], [0046]-[0053], [0060]-[0065], [0112], [0151]-[0153], [0160]-[0165]; claims 1, 11; and figure 1.	1-30
A	EP 2302861 A1 (INTEL CORPORATION) 30 March 2011 See paragraphs [0001]-[0004], [0012]-[0014], [0016]-[0019], [0023]; and claim 1.	1-30



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 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

"&amp;" document member of the same patent family


Date of the actual completion of the international search

08 November 2013 (08.11.2013)

Date of mailing of the international search report

**08 November 2013 (08.11.2013)**

Name and mailing address of the ISA/KR


 Korean Intellectual Property Office  
 189 Cheongsa-ro, Seo-gu, Daejeon Metropolitan City,  
 302-701, Republic of Korea

Facsimile No. +82-42-472-7140

Authorized officer

BYUN, Sung Cheal

Telephone No. +82-42-481-8262



**INTERNATIONAL SEARCH REPORT**

Information on patent family members

International application No.

**PCT/US2013/048036**

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2012-0163589 A1	28/06/2012	US 2013-232345 A1 WO 2012-087562 A2 WO 2012-087562 A3	05/09/2013 28/06/2012 16/08/2012
US 2010-0031342 A1	04/02/2010	None	
KR 10-2012-0099472 A	10/09/2012	CN 102473224 A DE 112009005466 T5 GB 2481563 A JP 2012-530961 A US 2013-0159726 A1 US 2013-0198853 A1 WO 2011-078855 A1 WO 2011-078855 A9	23/05/2012 31/10/2012 28/12/2011 06/12/2012 20/06/2013 01/08/2013 30/06/2011 09/09/2011
US 2012-0159184 A1	21/06/2012	US 2013-232344 A1 WO 2012-082410 A2 WO 2012-082410 A3	05/09/2013 21/06/2012 16/08/2012
EP 2302861 A1	30/03/2011	CN 102034039 A JP 2011-070665 A JP 2013-054786 A JP 5166499 B2 US 2011-0069835 A1 US 2013-0182837 A1 US 8363831 B2	27/04/2011 07/04/2011 21/03/2013 21/03/2013 24/03/2011 18/07/2013 29/01/2013