Office de la Propriété Intellectuelle du Canada
Un organisme d'Industrie Canada

Canadian Intellectual Property Office
An agency of Industry Canada

CA 2419863 A1 2002/03/07

(21) 2 419 863

(12) DEMANDE DE BREVET CANADIEN
CANADIAN PATENT APPLICATION

(13) A1

(86) Date de dépôt PCT/PCT Filing Date: 2001/08/28
(87) Date publication PCT/PCT Publication Date: 2002/03/07
(85) Entrée phase nationale/National Entry: 2003/02/17
(86) N° demande PCT/PCT Application No.: US 2001/026634
(87) N° publication PCT/PCT Publication No.: 2002/019598
(30) Priorité/Priority: 2000/08/28 (09/649,838) US

(51) Cl.Int.^{7}/Int.Cl.^{7} G06F 12/14, G06F 12/16, H04L 9/00

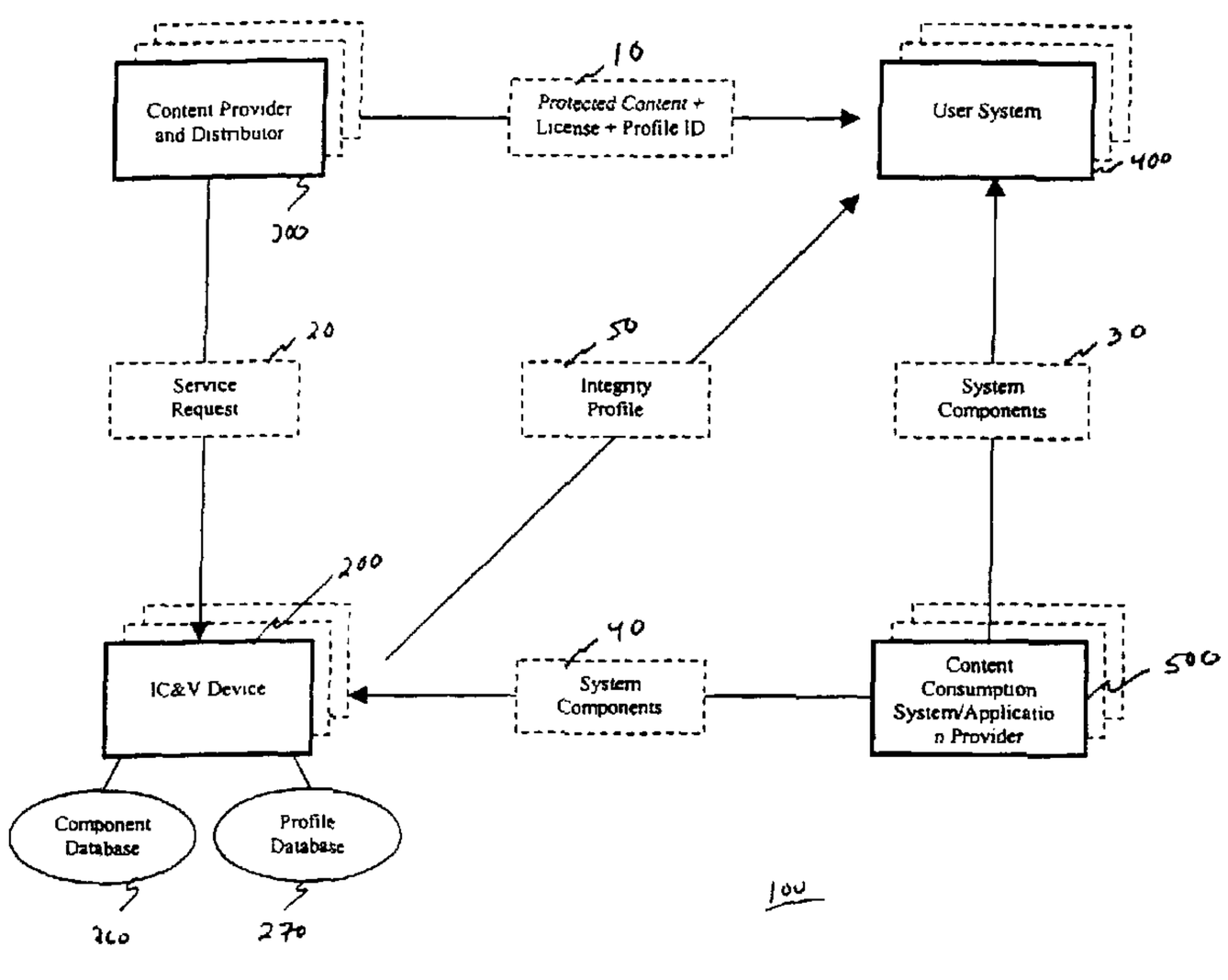
(71) Demandeur/Applicant:
CONTENTGUARD HOLDINGS, INC., US

(72) Inventeurs/Inventors:
TA, THANH, US;
WANG, XIN, US

(74) Agent: ROBIC

(54) Titre : SYSTEMES ET PROCEDES DE CERTIFICATION D'INTEGRITE ET DE VERIFICATION D'ENVIRONNEMENTS DE CONSOMMATION DE CONTENU

(54) Title: SYSTEMS AND METHODS FOR INTEGRITY CERTIFICATION AND VERIFICATION OF CONTENT CONSUMPTION ENVIRONMENTS



(57) Abrégé/Abstract:
A provider (300), provides protected content to a user, for consumption within a trusted environment. By providing integrity certification and verification services, the authenticity of the contents can be verified. The content provider (300) forwards to the user (400) a protected version (10) of the digital content which includes a license agreement and an integrity profile identification, which profile includes the applications and system components to be used in conjunction with the protected content. The content provider initiates and forwards a request (20) for the integrity profile to a device (200), which if an integrity profile does not already exist for the requested applications and/or systems components, queries a provider (500), who has supplied the system components to the user. The provider returns to the device authentication information (40) which is about the particular applications or systems components, and which allows a comparison between an application and/or system component on a user's system, and the original application or system component as distributed by the provider.

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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
7 March 2002 (07.03.2002)

PCT

(10) International Publication Number
WO 02/19598 A3

(51) International Patent Classification[7]: G06F 12/14, 12/16, H04L 9/00

(21) International Application Number: PCT/US01/26634

(22) International Filing Date: 28 August 2001 (28.08.2001)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
09/649,838 28 August 2000 (28.08.2000) US

(71) Applicant: CONTENTGUARD HOLDINGS, INC. [US/US]; 103 Foulk Road, Suite 200-M, Wilmington, DE 19803 (US).

(72) Inventors: TA, Thanh; 18694 Stratton Lane, Huntington Beach, CA 92648 (US). WANG, Xin; 3005 Shrine Place, #8, Los Angeles, CA 90007 (US).

(74) Agent: KAUFMAN, Marc, S.; Nixon Peabody LLP, Suite 800, 8180 Greensboro Drive, McLean, VA 22102 (US).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW.

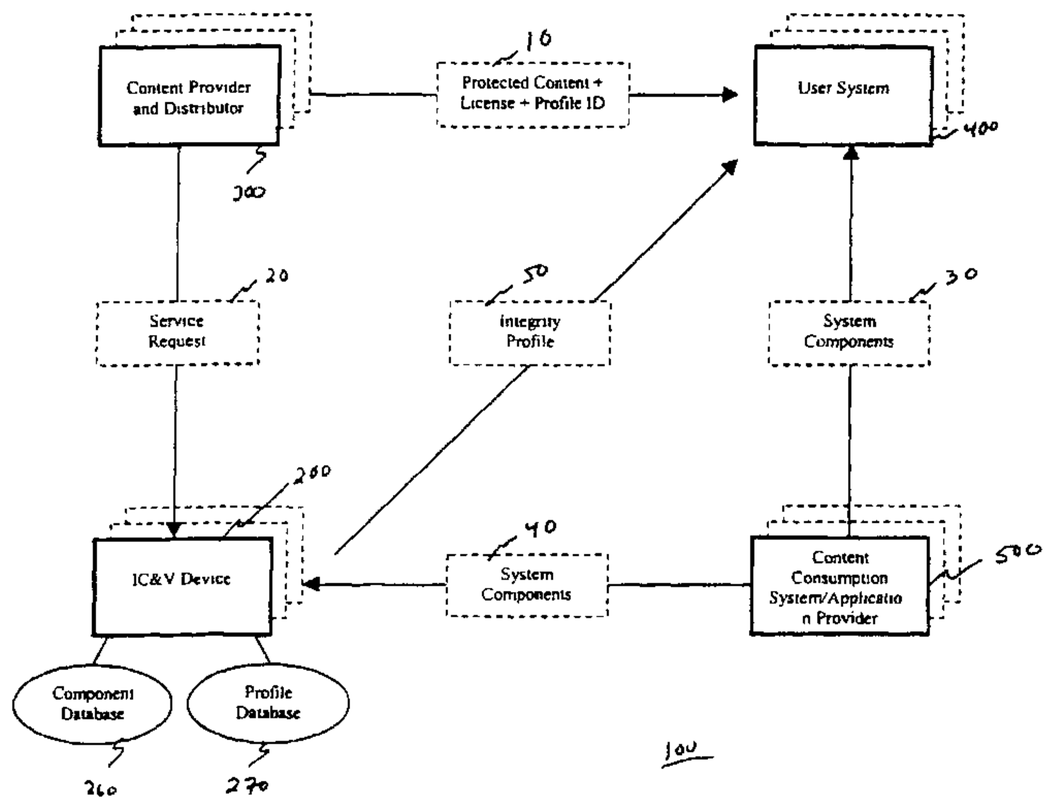
(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— *with international search report*

(88) Date of publication of the international search report:
13 June 2002

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: SYSTEMS AND METHODS FOR INTEGRITY CERTIFICATION AND VERIFICATION OF CONTENT CONSUMPTION ENVIRONMENTS



(57) Abstract: A provider (300), provides protected content to a user, for consumption within a trusted environment. By providing integrity certification and verification services, the authenticity of the contents can be verified. The content provider (300) forwards to the user (400) a protected version (10) of the digital content which includes a license agreement and an integrity profile identification, which profile includes the applications and system components to be used in conjunction with the protected content. The content provider initiates and forwards a request (20) for the integrity profile to a device (200), which if an integrity profile does not already exist for the requested applications and/or systems components, queries a provider (500), who has supplied the system components to the user. The provider returns to the device authentication information (40) which is about the particular applications or systems components, and which allows a comparison between an application and/or system component on a user's system, and the original application or system component as distributed by the provider.

WO 02/19598 A3

SYSTEMS AND METHODS FOR INTEGRITY CERTIFICATION AND VERIFICATION OF CONTENT CONSUMPTION ENVIRONMENTS

Background of the Invention

Field of the Invention

[0001] This invention relates to integrity certification and verification. In particular, this invention relates to integrity certification and verification within a content consumption environment.

Description of the Related Art

[0002] One of the most important issues impeding the widespread distribution of digital documents via electronic commerce is the current lack of protection available for the intellectual property rights of content owners and providers during the distribution and use of those digital documents. Efforts to resolve this problem have been termed Intellectual Property Rights Management (IPRM), Digital Property Rights Management (DPRM), Intellectual Property Management (IPM), Digital Rights Management (DRM), Rights Management (RM) and Electronic Copyright Management (ECM).

[0003] Content providers often want their contents to be consumed by certified applications and systems that have a desired characteristic and/or behavior. The direct use of a public key infrastructure (PKI) makes it possible that application and system providers can certify their own products and content providers can verify the integrity of the applications and systems that are used to consume their contents.

Summary of the Invention

[0004] However, the direct use of the PKI creates a many-to-many relationship between the vendors and the providers. This type of relationship does not scale well, and hence, managing the relationship and conducting an efficient and real-time integrity verification is difficult, if not impossible, to achieve.

[0005] Content providers often want to have their contents consumed by certified applications and systems that have desired characteristics and behaviors. By controlling these aspects of the content consumption environment, the content provider can, for example, restrict usage, e.g., copying, printing, embedding, distribution, or the like.

[0006] For example, a content provider may want to protect content against

misuse by demanding that the system that consumes the content be of a certain level of security and rights management capability. The content provider may also want to assure that no "alien" application, e.g., a debugger, virus, interception routine, or the like, interacts with the content consumption application on the user system which may confiscate or otherwise "steal" the content or other sensitive information. For example, Attorney Docket No. 111325 000002, entitled "Document Distribution Management Method And Apparatus Using A Standard Rendering Engine And A Method And Apparatus For Controlling A Standard Rendering Engine," filed herewith and incorporated herein by reference in its entirety, allows the management of the functionality of a user system to restrict a user's access to and over a document.

[0007] In order to certify that given applications and systems have desired characteristics and behaviors, a verification of all the applications and system components needed to consume the content need be confirmed by a verification application.

[0008] This invention describes systems and methods that provide certification and verification services to content consumption environments. Within such a system, an integrity certification and verification device that provides these services is introduced between a content provider and a content consumption system/application provider. This certification device registers individual applications and/or systems from their respective providers, and certifies the integrity of these applications and/or systems to content providers according to a predetermined selection. Through the use of this service, a content provider can "trust" an integrity certification and verification device. With this trust, the provider establishes a profile of a set of applications and systems that are allowed to consume its contents, and verifies on the user system(s), according to the profile, that the user's set of applications and systems are authentic.

[0009] In particular, the systems and methods of this invention provide certification and verification services to integrity of content, e.g., a document, consumption environments. Within such a system, an integrity certification and verification device that provides these services is introduced between content providers and content consumption system and application providers who may distribute, for example, personal computers, handheld computers, PDAs, multimedia display devices, DVD players, distributed network enabled phones, and applications, such as word processors, document viewers, multimedia players, or the like. The integrity certification and verification device registers individual applications and/or systems from the content consumption system/application providers, and certifies sets of these applications and systems to content providers. By using this service, a content

provider can select, or trust, the integrity certification and verification device, establish a profile of a set of applications and systems that are allowed to consume its contents, and verify on a user system, according to the profile, that the set of applications and systems on the user system are authentic. In this manner, the extent of access to, or control over, the content requested, or submitted, by the user can be controlled.

[0010] A document, as the term is used herein, is any unit of information subject to distribution or transfer, including, but not limited to, correspondence, books, magazines, journals, newspapers, other papers, software, a plug-in, photographs and other images, audio and video clips and other multimedia presentations. A document may be embodied in printed form on paper, as digital data on a storage medium, or in any other known or later developed variety of media or software including, for example, compact discs (CD's), digital video discs (DVD), laser discs, magneto- and magneto-optic media and the like.

[0011] The systems and methods of this invention provide for integrity certification and verification services.

[0012] This invention separately provides systems and methods for integrity certification and verification services for content consumption system environments.

[0013] This invention also separately provides a system and method for determining an integrity profile.

[0014] This invention additionally provides a system and method for verifying the integrity of one or more system environments.

[0015] This invention also provides a system and method for managing integrity profiles, system and system component information.

[0016] This invention additionally provides a system and method that performs an integrity check on a user system through the use of an integrity profile.

[0017] Specifically, the content provider, such as a document publisher or distributor, initiates a request for an integrity profile. This request for the integrity profile is forwarded to an integrity certification and verification device. The integrity certification and verification device can, for example, if an integrity profile does not already exist for the requested applications and systems components, query a content consumption system/application provider, who, for example, has supplied various system components and/or applications to users. The content consumption system/application provider returns to the integrity certification and verification device authentication information about the particular applications or system components. The authentication information allows a comparison, or integrity

verification, to be made between an application or system component on a user's system, and the original application or system component as distributed by the content consumption system/application provider.

[0018] The authentication information for system applications and components are stored in a component database. The profiles for content providers are stored in a profile database. Alternatively, the content consumption system/application provider can maintain a database of authentication information that can be forwarded directly to the respective database of the integrity certification and verification device, without the need for the integrity verification and certification device determining the integrity profile. An integrity profile identification, corresponding to the determined integrity profile, is then returned to the content provider.

[0019] A content provider, such as a document distributor, provides, for example, protected content to a user. The content provider forwards to the user a protected version of the digital content that includes, for example, a license agreement and an integrity profile identification. The integrity profile identification includes, for example, the applications and system components that are allowed to be used in conjunction with the protected content, and the identification of the integrity profile for those systems/applications.

[0020] Having the authentication information from the content consumption system/application provider, the integrity certification and verification device forwards, for example, at the request of the user system, an integrity profile to the user system. With this integrity profile, an integrity verification of the user's system can be performed. If it is determined that the components/applications of the user's system are authentic, the digital content provided by the content provider can then be accessed by the user's applications and systems in accordance with, for example, the additional profile information.

[0021] However, it is to be appreciated that the request for an integrity certification need not originate with the content provider. To the contrary, the certification request can be initiated, for example, by a software application embedded in the profile identification information that is forwarded with the protected content from the content provider to the user's system.

[0022] Alternatively, the content provider may also serve as the integrity verification and certification system. In this instance, the content provider conducts the integrity certification and verification service itself by gathering the appropriate authentication information and determining an integrity profile for the content provider's own use.

[0023] Additionally, the content consumption application/system provider can also act as the integrity certification and verification device. In this instance, the content consumption application/system provider, for example, may also supply an integrity profile together with the associated application and/or system component.

[0024] These and other features and advantages of this invention are described in, or are apparent from the following detailed description of the preferred embodiments.

Brief Description of the Drawings

[0025] The preferred embodiments of the invention will be described in detail, with reference to the following figures, wherein:

[0026] Fig. 1 is a functional overview illustrating a first exemplary embodiment of the integrity certification and verification system according to this invention;

[0027] Fig. 2 is a functional block diagram illustrating a first exemplary embodiment of the integrity certification and verification system according to this invention;

[0028] Fig. 3 is a workflow diagram of an exemplary integrity certification and verification device according to this invention;

[0029] Fig. 4 illustrates an exemplary structure of an integrity profile according to this invention;

[0030] Fig. 5 illustrates an exemplary environment stack according to this invention;

[0031] Fig. 6 illustrates an exemplary environment stack according to this invention;

[0032] Fig. 7 illustrates the workflow of an exemplary stack according to this invention;

[0033] Fig. 8 illustrates an exemplary workflow of the stack according to this invention;

[0034] Fig. 9 illustrates an exemplary method of manipulating the stack according to this invention;

[0035] Fig. 10 illustrates an exemplary method of preventing dynamic tampering through the use of debugging according to this invention;

[0036] Fig. 11 is a flowchart outlining one exemplary embodiment of a method for integrity certification and verification according to this invention;

[0037] Fig. 12 is a flowchart outlining one exemplary embodiment of a method for registering applications and/or systems according to this invention;

[0038] Fig. 13 is a flowchart outlining an exemplary embodiment of a method for

determining an integrity profile according to this invention; and

[0039] Fig. 14 is a flowchart outlining an exemplary embodiment of a method for verifying the integrity of an integrity authenticator according to this invention.

Detailed Description of Preferred Embodiments

[0040] The systems and methods of this invention provide certification and verification services to determine the integrity of a content consumption environment. Within this system, an integrity certification and verification device is introduced between one or more content providers, and one or more content consumption systems and application providers. The integrity certification and verification device obtains authentication information from the content consumption application and/or system providers. This authentication information allows a content provider to trust the environment to which content will be provided. Thus, based on the authentication information received from the content consumption application and system provider, an integrity profile is established. This profile is then forwarded to the user system to confirm that the user has not altered, modified, or does not potentially interfere in a unauthorized manner with the digital content provided by the content provider.

[0041] Fig. 1 illustrates an exemplary system for performing integrity certification and verification. In particular, the integrity certification and verification system 100 includes an integrity certification and verification device 200, a content provider and/or distributor 300, a user system 400, a content consumption system/application provider 500, a component database 260 and a profile database 270.

[0042] In an exemplary operating environment, the content consumption system/application provider 500 provides applications, systems and/or software/hardware components to a user. The user system 400 allows consumption of digital content, such as documents, that are supplied by the content provider and distributor 300. In order to verify the integrity of the user system 400, the integrity certification and verification device 200 collects and registers authentication information about the individual applications, systems and/or software/hardware components from the content consumption system/application provider 500. With this authentication information, the integrity certification and verification device 200 determines and certifies an integrity profile of one or more applications, systems and/or system components based on a service request 20 from the content provider 300. This determined integrity profile 50 is then forwarded to the user system 400 so that the integrity of the user system 400 can be determined.

[0043] In operation, a content provider and distributor 300 provides digital

content, such as a document, to a user system 400. The user system 400 comprises one or more system components such as hardware components and/or various software applications. These applications and hardware/software components are usually obtained by the user from one or more content consumption system/application providers, such as a computer supplier, a software warehouse, an application provider, or the like. These applications and hardware and software components are then assembled, if not already done so, or installed, as appropriate, by the user in order to allow the user to consume content, such as documents.

[0044] Thus, during the course of use of the applications and hardware/software of the user environment, the user may want to view protected content, such as a document. Thus, the user 400 would request from the content provider 300 one or more documents, such as an electronic book, a multimedia file, a presentation, a form template, or the like. Upon receiving this request, the content provider and distributor 300 could provide the requested content in protected form with a profile identification 10 to the end user 400. This profile identification 10 includes, for example, specifics as to in which applications the protected content can be viewed, and, for example, the extent to which the provided content can be manipulated within the particular software/hardware environment.

[0045] Additionally, the content provider 300 can forward a service request 20 to an integrity certification and verification device 200. The service request 20 includes, for example, a list of components and/or software applications on which the content provider 300 wishes to allow the user system 400 to consume the distributed protected content. The integrity certification and verification device 200 determines if the components and applications/software identified in the service request have corresponding authentication information stored in the component database 260 and/or the profile database 270. If the integrity certification and verification device does not have the authentication information specified in the service request 20, the integrity certification and verification device 200 can request from one or more content consumption system/application providers 500, authentication information about a particular application, system, hardware/software component, or the like. With this authentication information, the integrity certification and verification device 200 stores information pertaining to the application and system components in the component database 260. Alternatively, the integrity certification and verification device 200 can develop an integrity profile for one or more applications. With this information, which confirms the authenticity of applications, systems and system components, the integrity certification and verification device 200 forwards an

integrity profile 50 to the user system 400. This integrity profile 50 is used to confirm the authenticity of systems, system components and/or applications of the user system 400. If it is determined if the user's system components and/or applications are authentic, the protected content 10 is unprotected so that the user system 400 may view or otherwise manipulate the protected content in accordance with the integrity profile.

[0046] Fig. 2 illustrates an overview of the components of an integrity certification and verification environment 100 according to an exemplary embodiment of this invention. In particular, the integrity certification and verification environment 100 comprises one or more content providers 300, one or more user systems 400, one or more integrity certification and verification devices 200, and one or more content consumption system/application providers 500.

[0047] The content provider 300 comprises, for example, a controller 310, a memory 320, an I/O controller 330, and a content database 340. However, it is to be appreciated that the content provider 300 may also distribute content in a more traditional manner. For example, the content provider may distribute a compact disk containing content. This compact disk, for example, could be delivered through a postal service to a user. In general, any type of distribution and dissemination process will work equally well with the systems and methods of this invention.

[0048] The integrity certification and verification device 200 comprises a controller 210, a memory 220, an I/O controller 230, a digital signature device 240, a component registration device 250, a component database 260, a profile database 270, a profile creation device 280, a profile distribution device 290 and a profile verification device 295. The integrity certification and verification device 200 provides the following services: component registration service and integrity profile service. The registration service allows registration of applications, systems, and/or software/hardware components from their respective providers as authentic ones, with intended characteristics, purposes and/or behaviors.

[0049] An integrity profile service is provided to content providers to build and retrieve integrity profiles. An integrity profile is a document, which is optionally digitally signed, that contains verifiable information and a set of registered system components that are to consume the contents of protected documents. Once the integrity profile is created, the integrity profile's identification is returned to the content provider. The content provider will include the integrity profile identification and optionally a usage license with the protected documents. When the content of the protected document is consumed and there is a need to conduct a local integrity

verification of the system and environment of the user, the integrity profile can be retrieved from the integrity certification and verification device 200 to the user system.

[0050] The user system 400 comprises a controller 410, a memory 420, an I/O controller 430, a storage device 440, an integrity authentication device 450, and a profile storage device 460. However, it is to be appreciated that this exemplary user system is based on a model of a computer. However, it is to be appreciated that the components of the user system may change depending on, for example, the type of content being consumed. In general, any user system that comprises portions whose integrity can be verified will work equally well with the systems and methods of this invention.

[0051] The content consumption system/application provider 500 comprises, for example, a controller 510, a memory 520, an I/O controller 530, a registration application device 540, an application database 550, and a system database 560. However, similar to the content provider 300, the content consumption system/application provider may have several different forms depending on the type of system and/or application the content consumption system/application provider supplies. For example, if the content consumption system/application provider 500 supplies a specific hardware component, the content consumption system/application provider 500 may not maintain application and system databases. Alternatively, for example, the system/device component supplier may send, for example, on a disk, authentication information directly to the integrity certification and verification device 200.

[0052] Alternatively, the content consumption system/application provider 500 may coordinate efforts with the content provider 300 to facilitate determination of an integrity profile. In general, the content consumption system/application provider can be any entity that is capable of supplying hardware or software and authentication information about the same.

[0053] While in this exemplary embodiment the content consumption system/application provider 500 is shown comprising various system components, it is to be appreciated that the content consumption system/application provider 500 could be, for example, a computer distributor, a software developer, a software provider, a software distributor, or the like. Thus, the content consumption system/application provider 500 is capable of supplying devices and/or software that allows for the consumption of content that is provided by the content provider 300.

[0054] The various components of the integrity certification and verification

environment 100 are capable of communication therebetween, via link 5, which can be a wired or wireless link, or any other known or later-developed element(s) that is capable of supplying electronic data to and from the connected elements. For example, the link 5 can be one or more distributed networks which may in turn be connected to one or more additional integrity certification and verification environments 100, or, alternatively, multiple instances of any one or more of the content providers 300, user systems 400, content consumption system/application providers 500 and integrity certification and verification devices 200.

[0055] In an exemplary operating environment, the content consumption system/application provider 500 supplies applications, software and/or hardware to a user. These applications, software and/or hardware are used by a user to consume content, for example, viewing documents.

[0056] The content provider 300, for example, at the request of a user located at the user system 400, distributes content, such as a document, to the user system 400. In particular, a request can be received by the content provider 300 from the user system 400. This request, which is received through the I/O controller 330, is processed by the controller 310, in cooperation with memory 320 to retrieve the requested content from the content database 340. For example, the content provider 300 can be an on-line content provider, a book store, a software provider, or any other content provider that wishes to provide content, such as a document, to a user.

[0057] Upon receiving a content request from the user system 400, the content provider 300 returns to the user system the requested content as well as additional information about the protected content. This additional information can include a profile identification. Alternatively, the additional information could contain, for example, information instructing the user system to request a profile, and hence an integrity certification, before enabling of the content.

[0058] Additionally, the additional information can identify which system components and/or hardware/software can be running and/or used on the user's machine when viewing or interacting with the requested content.

[0059] Thus, one or more of the requested content, additional information and profile identification are received by the user system 400, via the I/O controller 430, and at the direction of controller 410, stored in one or more of the memory 420 and the storage device 440.

[0060] In one exemplary embodiment, the content provider 300 can initiate a service request 20, such as a request for an integrity profile, from the integrity certification and verification device 260. The integrity certification and verification

device 260, receives, via the I/O controller 230, and in cooperation with the controller 210 and memory 220 the service request from the content provider 300.

[0061] As previously discussed, the integrity certification and verification device 200 comprises a component database 260 and a profile database 270. The component database 260 stores authentication information pertaining to systems and system components that can be distributed by one or more content consumption system/application providers 500. Similarly, the profile database 270 stores verifiable information and a set of registered system components that are to consume the contents of protected documents for one or more individual content providers 300.

[0062] Thus, upon receipt of the request for an integrity profile from the content provider 300, the integrity certification and verification device 200, at the direction of the controller 210 and with the aid of memory 220, searches the component database 260 and the profile database 270 to determine if authentication information already exists that corresponds to the information in the service request.

[0063] Alternatively, the integrity certification and verification device 200 can perform an on-line verification service. The on-line verification service is provided to perform the integrity verification on-line, for example, at real time within the integrity certification and verification device 200. In order to initiate this service, a piece of software, called an integrity authenticator, is forwarded to the user system 400. The integrity authenticator allows the collection of information of local software and/or hardware components. Alternatively, the integrity authenticator can be a dedicated device, such as the integrity authentication device 450 illustrated in Fig 2. The information gathered about the local software and/or hardware components is returned along with the integrity profile identification to the integrity certification and verification device 200 so that the on-line integrity verification can be performed. The component registration device 250 examines software/hardware components from their respective providers and stores identification information in the component database 260. The information pertaining to the software/hardware component can be, for example, hashed and the hash value can be used as the authentic software/hardware identification. However, it is to be appreciated that the information to identify each software/hardware component can be any known or later-developed scheme that allows for identification of an authentic piece of hardware and/or software.

[0064] The registration of a particular software and/or hardware component is accomplished as follows. For example, the content consumption system/application provider 500 can communicate with the identification and certification verification

device 200 to request a registration service or, alternatively, the identification and certification verification device 200 can communicate with content consumption system/application provider 500 in order to secure the authentication information. In this example, the registration application device 540, in cooperation with the controller 510, the memory 520 and the I/O controller 530, searches one or more of the application database 550 and the system database 560 to secure information about the particular software and/or hardware including, for example, the provider name, a component identification, for example, a serial number, version number, build number, or the like, and alternatively, the application itself.

[0065] For example, in one particular operating scenario, instead of acquiring authentication information from a particular content consumption system/application provider 500, the integrity certification and verification device 200 could actually request, for example, a particular application, such as a software program, from the content consumption system/application provider 500. In this way, the integrity certification and verification device 200 would not need authentication information since the integrity certification and verification device 200 could secure the particular software application directly from the content consumption system/application provider 500.

[0066] The component registration device 250 verifies the information of the component, and optionally computes, for example, a hash value that can be used, for example, as the authentic software and/or hardware identification. The component registration device 250 then stores the component information and, for example, the hash value, in the component database 260.

[0067] Alternatively, instead of sending the software and/or hardware component to the registration application device 540, the content consumption system/application provider 500 can also connect to the component registration device 250 to download a small software application, such as a registration application, and have it executed locally. This registration application will examine the target software/hardware component and send information pertaining to this software/hardware component possibly along with an integrity value, such as a hash value, back to the component registration device 250 which can then store the authentication information about the component in the component database 260.

[0068] Alternatively, the profile creation device 280 builds integrity profiles for software. In particular, an integrity value, such as a hash value, of each software application can be retrieved from the component database and stored. Also included in the profile is an optional interaction relationship among the components. This

relationship is used to identify the calling and returning sequence of the components in order to prevent unintended interaction with other components. The content of the integrity profile is then, for example, digitally signed and the resulting signature is appended to the integrity profile. Each integrity profile is associated with a unique identification.

[0069] Fig. 3 illustrates an exemplary workflow of input, output and services and operations provided by the integrity certification and verification device 200. Specifically, for the component registration service, a component identification, and optionally, meta information about the particular component, is forwarded to the component registration device 250. The component registration device 250 registers the component, for example, with intended characteristics, purposes, and behaviors in the component database. Then, the component registration device 250 returns the identification of the registered component to, for example, the content consumption system/application provider, and makes the identification available to, for example, the content provider 300.

[0070] For profile creation, the profile creation device 280 receives the identifications of registered components. The identifications of the registered components, when combined with the information about the associated components, if any, are then digitally signed and stored in the profile database. An integrity profile identification is returned to the requestor.

[0071] Similarly, the profile distribution device 290 receives an integrity profile identification. The profile database 270 is then queried to determine if an integrity profile corresponding to the integrity profile identification is available. If the integrity profile is available, the integrity profile is returned to the requestor. Otherwise, the integrity profile can be determined with the aid of the profile creation device 280.

[0072] The profile verification device 295 receives information identifying one or more components and an integrity profile identification. The profile verification device compares the component identifications, integrity profile identification and corresponding integrity profile to determine verification data. If the profiles and components and identifications match, the integrity of the system has been verified. Otherwise, the system is not the one specified in the integrity profile, or it has been altered in some way.

[0073] Fig. 4 illustrates an exemplary integrity profile. This exemplary integrity profile can be created by the profile creation device 280. To build an integrity profile for an authenticated content provider, a request for creating an integrity profile is initiated. For example, the provider can contact the integrity certification and

verification device 200 and request the creation of an integrity profile. Then, the provider sends a list of names of software and/or hardware components to the integrity certification and verification device 200. The profile creation device 280 then retrieves the identification, such as an integrity or a hash value, of each of the components from the component database 260. The profile creation device 280 then determines an integrity profile, which contains the authentication information, such as an integrity or a hash value, of each of the components, together with other information such as the integrity profile identification, version number, creation date, build date, content provider name, and for example, optionally, the interaction relationship between any of the software and/or hardware components.

[0074] The profile creation device 280 forwards the determined integrity profile to a digital signer 240, which can then sign the content of the profile. The profile creation device 280 then stores the signed profile in the profile database 270 and returns the profile identification to the content provider 300.

[0075] When creating, for example, a usage license for the content of a protected document, the content provider 300 can optionally include the integrity profile identification into the usage license. On the user system 400, the integrity profile will be used to verify all of the software/hardware components in an environment call stack. This assures that the sensitive information can only be consumed by authorized software/hardware components, or any combination thereof.

[0076] The profile distribution device 290 accepts requests for obtaining integrity profiles and retrieves them from the profile database 270 and returns the integrity profiles to the respective requestor. Similarly, the profile verification device 295 accepts requests for verifying user systems for one or more system environments. The profile verification device 295 gathers the information about the software/hardware components according to integrity profiles, verifies the information against the profiles, and returns the verification results back to the requesters.

[0077] The user system 400 comprises an integrity authentication device 450. The integrity authentication device 450, for example, runs on top of any content consumption application.

[0078] Thus, Fig. 5 illustrates an exemplary system environment stack on user device 400 for verifying system integrity. In particular, the user system environment stack comprises an integrity authenticator and one or more system components.

[0079] Fig. 6 illustrates an example of an environment stack which includes an integrity authenticator, a plug-in, a rendering application, an operating system, an operating system (OS) boot strap, and the respective hardware.

[0080] In an exemplary operating environment, the integrity authentication device 450 contains its own encryption/decryption key pair and a verification key of an identification certification and verification device. These keys are possibly hidden and/or embedded within the integrity authentication device 400 for the tamper-resistance aspects of this invention. For those applications that require the use of a user's private information or involve sensitive documents and data, the integrity authentication device 450 can use an associated integrity profile to verify all of the software/hardware components on the call stack in the user system environment.

[0081] The integrity authentication device 450 will first verify the signature of the profile using the integrity certification and verification device verification key. As illustrated in Figs. 7-9, once the signature is verified, the integrity authentication device 450 examines the current call stack and starts to authenticate each software/hardware component on the call stack using the information provided in the integrity profile. The call stack is a continuous block of memory which consists of memory images and the involved functions or procedures. The stack operates on the concept of a last-in-first-out and the stacks basic operations are the stack "push" and stack "pop." Push is used to store the images onto the stack and advance to the top of the stack to a position. Pop is used to remove the data from the stack and restore the top of the stack to a previous position.

[0082] With the call stack, the image of the currently executed function is at the top of the stack. When the currently executed function invokes or calls the next function, the memory image of the next function is pushed on the top of the call stack and the top of the call stack points to the image of the next function. Each portion of the stacked images will contain the addresses or return instruction after the called function finishes its execution.

[0083] Fig. 10 illustrates how the execution environment is protected. Specifically, to protect the Integrity Authenticator (IA), the execution of the IA is monitored by a trusted application, which is part of the IA. The monitoring process, e.g., an application, can be a debugger or a special process that can prevent the IA from being monitored by any other process or application in the system. In an environment when a process can only be debugged by only one process, then the trusted monitoring program can be implemented as a debugger. Since the monitoring program is a trusted application, the monitoring program's integrity must be in the current integrity profile. Therefore, the IA will verify the integrity of the trusted application before loading and execution. The function of the trusted monitoring application is to prevent the IA from being monitored and controlled and captured by

other processes. Another function of the trusted monitoring application is to monitor the current environment and determine if the change in environment is valid. However like the IA, the trusted monitoring application must also be protected and the IA will act as the monitor to protect the trusted monitoring application from being monitored, captured and/or controlled by other applications. This dual protection mechanism creates a closed system that will prevent other applications from monitoring the execution of the integrity authenticator.

[0084] Fig. 11 illustrates an exemplary method of operation of the integrity certification and verification device. In particular, control begins in step S100 and continues to step S110. In step S110, an integrity profile is determined. Next, in step S120, the integrity profile is certified. Then, in step S130, the integrity profile is forwarded to the user. Control then continues to step S140.

[0085] In step S140, the integrity of the user system is verified. Next, in step S150, a determination is made whether the user system is authentic. If the user system is authentic, control continues to step S160, where the user is allowed access to the selected content. Otherwise, control jumps to step S170, where the content access is denied or disabled. Control then continues to step S180, where the control sequence ends.

[0086] Fig. 12 illustrates an exemplary method of registering components/hardware and/or software according to this invention. In particular, control begins in step S200 and continues to step S210. In step S210, the registration service is initiated. Next, in step S220, the component supplier provides authentication information about particular components/hardware and/or software. Then, in step S230, information about the particular components/hardware and/or software is verified. Control then continues to step S240.

[0087] In step S240, a determination whether an integrity value should be determined. If an integrity value is to be determined, control continues to step S250, where an integrity value is determined. Otherwise, control jumps to step S260 where authentication information about the component/hardware and/or software is stored.

[0088] Next, in step S270, a determination is made whether to store an integrity value. If an integrity value is to be stored, control continues to step S280, where the integrity value is stored. Otherwise, if an integrity value is not to be stored, control jumps to step S290, where the control sequence ends.

[0089] Fig. 13 illustrates an exemplary method of determining a profile according to this invention. In particular, control begins in step S300 and continues to step S310.

In step S310, the integrity profile determination is initiated. Next, in step S320, the name, such as an identification of the component and/or hardware or software is obtained. Then, in step S330, the identification for the component/hardware or software is retrieved. Control then continues to step S340.

[0090] In step S340, the integrity profile is determined. Next, in step S350, the integrity profile is digitally signed. Then, in step S360, the digitally signed integrity profile is stored. Control then continues to step S370.

[0091] In step S370, the signed integrity profile is then forwarded to the requestor, such as the content consumption system/application provider. Control then continues to step S380 where the control sequence ends.

[0092] Fig. 14 illustrates an exemplary method of verifying the integrity of the integrity authenticator in accordance with one aspect of the present invention. Control begins in step S400 and continues to step S410. In step S410, the integrity of the integrity authenticator is verified. Next, in step S420, a determination is made whether the integrity authenticator is valid. If the integrity authenticator is valid, control continues to step S430. Otherwise control jumps to step S540.

[0093] In step S430, a tamper-resistant environment is established. Next, in step S440, the integrity profile is verified. Then, in step S450, a determination is made whether the integrity profile is valid. If the integrity profile is valid, control continues to step S460. Otherwise, control jumps to step S540.

[0094] In step S460, the integrity profile is loaded. Next, in step S470, the call stack of the current execution environment as illustrated in relation to Figure 6 is constructed. At the bottom of the call stack is a set of hardware and/or devices, with all the software components towards the top of the stack. The relationship of the components in the stack is that the lower component calls the component just above it. Once the call stack is constructed, the top of the call stack, which contains the execution image of the last executed component, is located. Thus, the execution image of each component on the stack helps identify the calling component. Then, in step S480, the identification calling component is retrieved. Control then continues to step S490.

[0095] In step S490, the integrity of the component is verified against the integrity profile. Next, in step S500, a determination is made whether the component is valid. If the component is valid, control continues to step S510. Otherwise, control jumps to step S540.

[0096] In step S510, a determination is made whether the stack is empty. If the stack is empty, control jumps to step S520. Otherwise, control jumps to step S530. In

step S520, the next component in the stack is located and this next component is set as the current stack frame. Control then returns to step S480 for verification.

[0097] In step S530, the integrity is verified and control continues to step S550, where the control sequence ends.

[0098] In step S540, the integrity check is failed and control continues to step S550 where the control sequence ends.

[0099] As illustrates in Figs. 1-2, the integrity certification and verification device is preferably implemented either on a single program general purpose computer or separate program general purpose computer. However, the integrity certification and verification device can also be implemented on a special purpose computer, a programmed microprocessor or microcontroller and peripheral integrated circuit element, an ASIC, or other integrated circuit, a digital signal processor, a hard-wired electronic or logic circuit such as a discrete element circuit, a programmable logic device such as a PLA, PLD, FPGA, PAL, or the like. In general, any device capable of implementing a finite state machine that is in turn capable of implementing the flowcharts illustrated in Figs. 11-14 can be used to implement integrity certification and verification device.

[00100] Furthermore, the disclosed method may be readily implemented in software using object or object-oriented software development techniques in environments that provide portable source code that can be used in a variety of computer or workstation hardware platforms. Alternatively, the disclosed integrity certification and verification device may be implemented partially or fully in hardware using standard logic circuits or a VLSI design. Whether software or hardware is used to implement the systems and methods in accordance with this invention is dependent on the speed and/or efficiency requirements of the system, the particular function, and particular hardware or software systems or microprocessor or microcomputer system being utilized. The integrity certification and verification devices and methods described above, however, can be readily implemented in hardware or software, using any known or later-developed systems or structures, devices, and/or software by those skilled in the applicable art without undue experimentation from the functional description provided herein, together with a general knowledge of the computer arts. Moreover, the disclosed methods may be readily implemented as software executed on a programmed general purpose computer, a special purpose computer, a microprocessor, a server or the like. In this case, the methods and systems of this invention can be implemented as a routine embedded on a personal computer or server, such as a JAVA® or CGI script as a

resource residing on a server or graphics work station as a routine embedded in a dedicated integrity certification and verification device, a web browser, a web TV interface, a PDA interface, a multimedia presentation device, or the like. The integrity certification and verification device can also be implemented by physically incorporating the systems and methods into a software and/or hardware system, such as the hardware and software systems of a graphics workstation or dedicated integrity certification and verification device.

[00101] It is, therefore, apparent that there has been provided, in accordance with the present invention, systems and methods for integrity verification. While this invention has been described in conjunction with the preferred embodiments thereof, it is evident that many alternatives, modifications and variations be apparent to those skilled in the applicable art. Accordingly, applicants intend to embrace all such alternatives, modifications, and variations that follow within the spirit and scope of this invention.

We Claim:

1. An integrity certification and verification system for content consumption environments comprising:

an integrity certification and verification device, the integrity certification and verification device storing authentication information about one or more applications, systems or system components; and

an integrity profile that is used to determine the authenticity of the one or more applications, systems or system components.

2. The system of claim 1, further comprising a component registration device that determines an integrity profile from the authentication information, the integrity profile containing at least one of verifiable information and an identification of registered applications, systems or system components.

3. The system of claim 1, further comprising a profile database that maintains an integrity profile and an identification of registered applications, systems or system components.

4. The system of claim 1, further comprising a profile verification device that verifies authenticity by comparing one or more of application, system or system component identifications, the one or more applications, systems or system components, the integrity profile and an integrity profile identification.

5. The system of claim 1, further comprising a registration application device that obtains the authentication information about the one or more applications, systems or system components from a content consumption application, system or system component provider.

6. The system of claim 1, wherein the integrity profile comprises an identification of the one or more applications, systems or system components that can be used in conjunction with distributed information.

7. The system of claim 1, further comprising a content provider that distributes information.

8. The system of claim 1, further comprising a content consumption application, system or system component provider.

9. The system of claim 1, wherein if a profile verification device determines that the one or more applications, systems or system components are not authentic, access to one or more documents is disabled.

10. The system of claim 1, further comprising a profile creation device that determines the integrity profile based on verifiable information about the one or more

applications, systems or system components and one or more authorized content consumption applications, systems or system components.

11. A method for integrity certification and verification in a content consumption environment comprising:

determining an integrity profile, the integrity profile allowing a determination of the authenticity of one or more applications, systems or system components; and

determining access rights to content based on the authenticity determination.

12. The method of claim 11, further comprising certifying the integrity profile.

13. The method of claim 11, further comprising verifying the authenticity of one or more applications, systems or system components.

14. The method of claim 11, wherein the access rights include at least one of enabling or disabling access to the content.

15. The method of claim 11, further comprising obtaining authentication information about the at least one application, system or system component.

16. The method of claim 15, further comprising digitally signing the integrity profile.

17. The method of claim 15, further comprising forwarding the digitally signed integrity profile to a content consumer.

18. The method of claim 11, further comprising verifying the integrity of an integrity authenticator.

19. The method of claim 18, further comprising establishing a tamper resistant environment.

20. The method of claim 18, further comprising verifying the integrity profile.

21. The method of claim 18, further comprising loading a valid integrity profile.

22. The method of claim 18, wherein verifying the integrity of an integrity authenticator comprises establishing that the integrity authenticator is not being at least one of monitored, controlled or recorded.

23. An information storage media that stores information for integrity certification and verification in a content consumption environment comprising:

information that determines an integrity profile, the integrity profile allowing a determination of the authenticity of one or more applications, systems or system components; and

information that determines access rights to content based on the authenticity

determination.

24. The information storage media of claim 23, further comprising information that certifies the integrity profile.

25. The information storage media of claim 23, further comprising information that verifies the authenticity of the one or more applications, systems or system components.

26. The information storage media of claim 23, wherein the access rights include at least one of enabling or disabling access to the content.

27. The information storage media of claim 23, further comprising information that obtains authentication information about the at least one application, system or system component.

28. The information storage media of claim 27, further comprising information that digitally signs the integrity profile.

29. The information storage media of claim 27, further comprising information that forwards the digitally signed integrity profile to a content consumer.

30. The information storage media of claim 23, further comprising information that verifies the integrity of an integrity authenticator.

31. The information storage media of claim 30, further comprising information that establishes a tamper resistant environment.

32. The information storage media of claim 30, further comprising information that verifies the integrity profile.

33. The information storage media of claim 30, further comprising information that loads a valid integrity profile.

34. The information storage media of claim 30, wherein verifying the integrity of an integrity authenticator comprises information that establishes that the integrity authenticator is not being at least one of monitored, controlled or recorded.

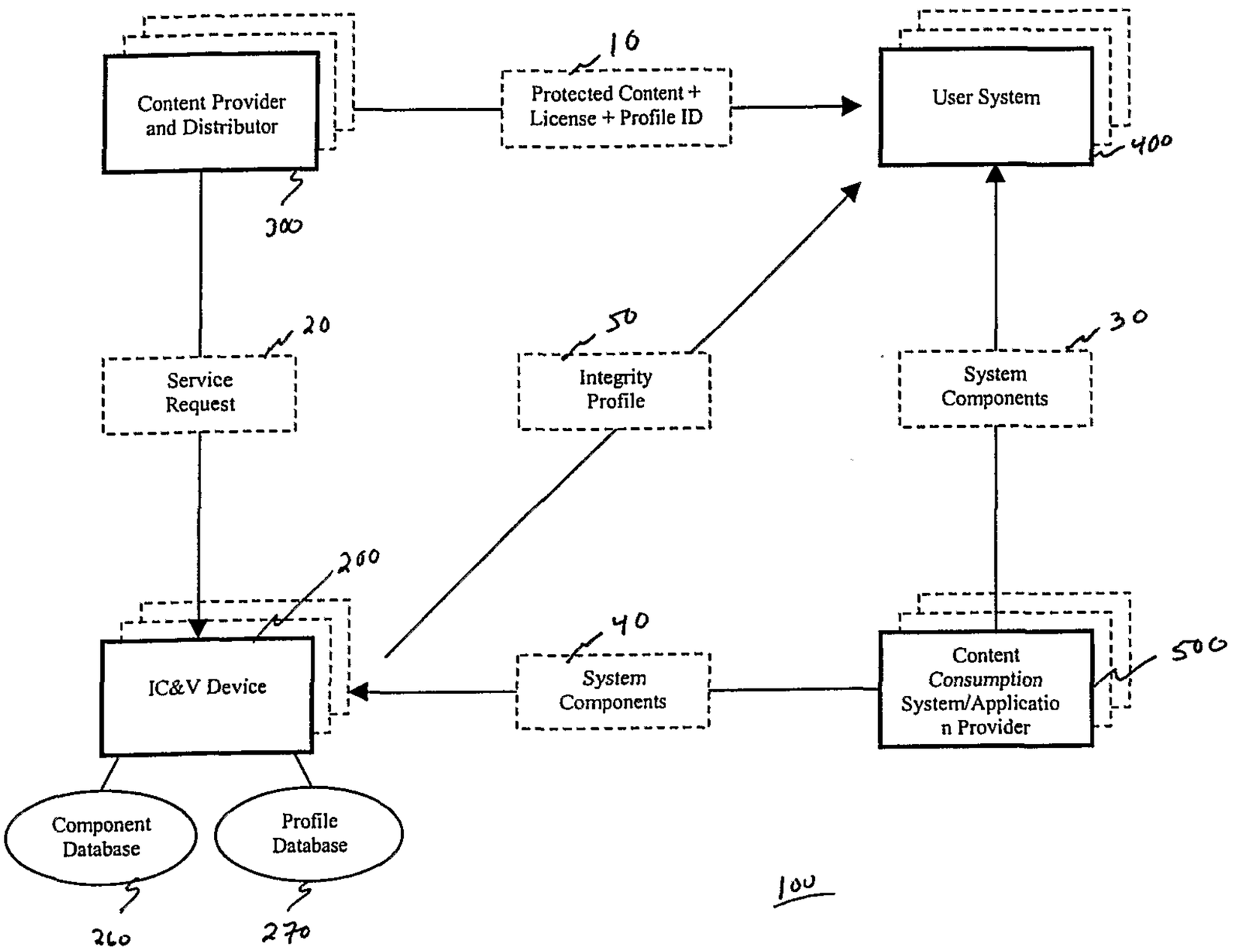

Content Provider and Distributor
300
16
Protected Content + License + Profile ID
User System
400
20
Service Request
50
Integrity Profile
30
System Components
200
IC&V Device
40
System Components
Content Consumption System/Application Provider
500
Component Database
260
Profile Database
270
100

Fig. 1

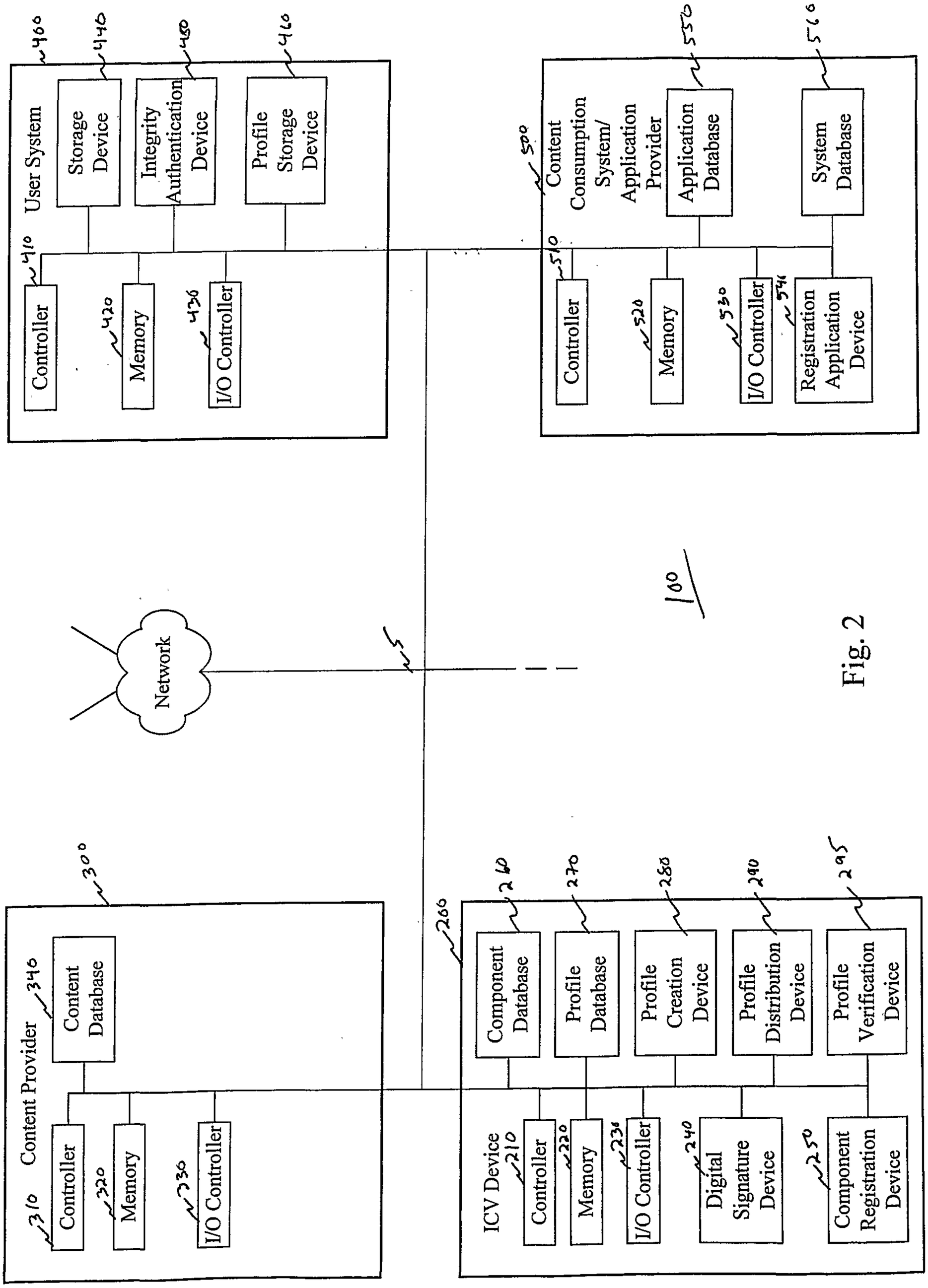

User System
460
Storage Device
440
Integrity Authentication Device
450
Profile Storage Device
460
410
Controller
420
Memory
430
I/O Controller
Content Consumption System/ Application Provider
500
Application Database
550
System Database
560
510
Controller
520
Memory
530
I/O Controller
540
Registration Application Device
Network
100
Content Provider
300
Content Database
340
310
Controller
320
Memory
330
I/O Controller
ICV Device
200
Component Database
210
Profile Database
270
Profile Creation Device
250
Profile Distribution Device
290
Profile Verification Device
295
210
Controller
220
Memory
230
I/O Controller
240
Digital Signature Device
250
Component Registration Device

Fig. 2

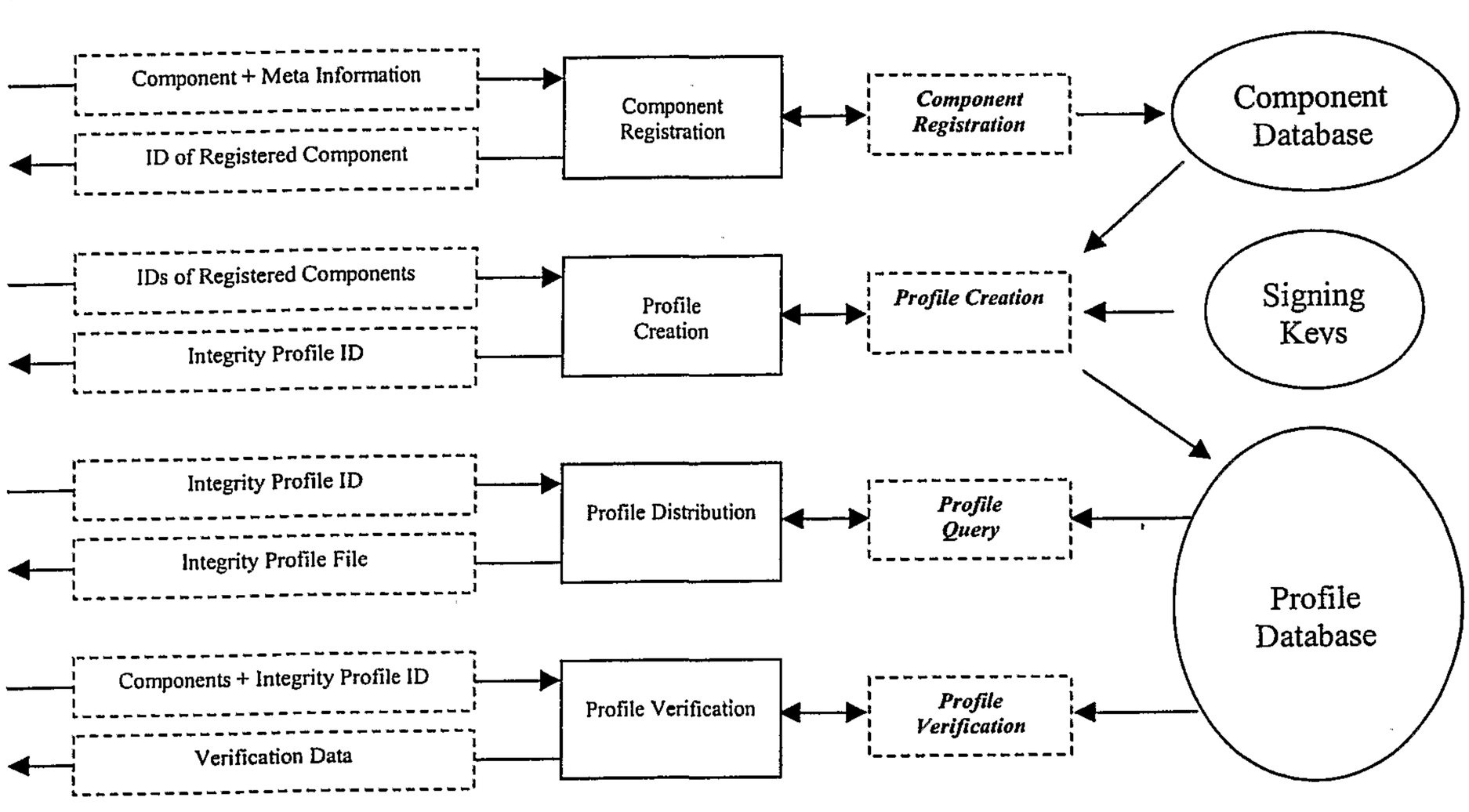

Input/Output
Services
Operations
Component + Meta Information
ID of Registered Component
Component Registration
Component Registration
Component Database
IDs of Registered Components
Integrity Profile ID
Profile Creation
Profile Creation
Signing Keys
Integrity Profile ID
Integrity Profile File
Profile Distribution
Profile Query
Profile Database
Components + Integrity Profile ID
Verification Data
Profile Verification
Profile Verification

Fig. 3

Figure 4. A structure of the integrity profile.

Integrity profile identification
Version number of integrity profile
Creation date
Creator
Content Provider Name and ID
A list of integrity values (e.g., hash values) of the components
(optional) Interaction relationship among the components
Digital signature of integrity profile

Figure 5. A general end-user system environment stack

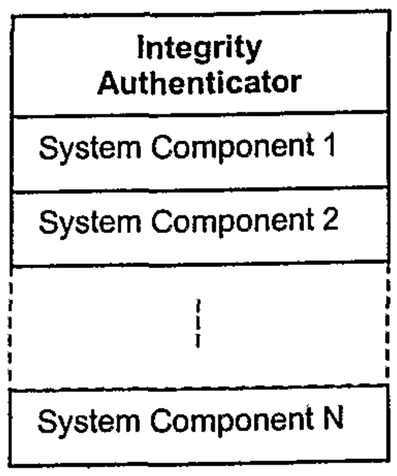


Figure 6. An example of an end-user system environment stack

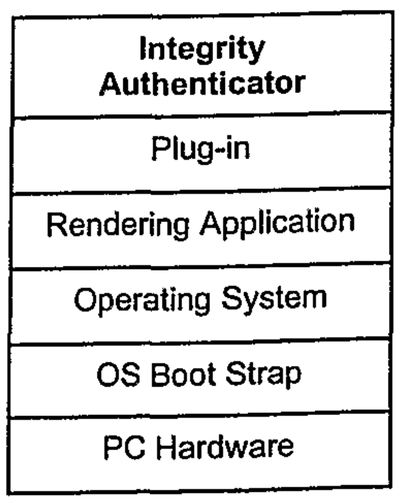


Figure 7.

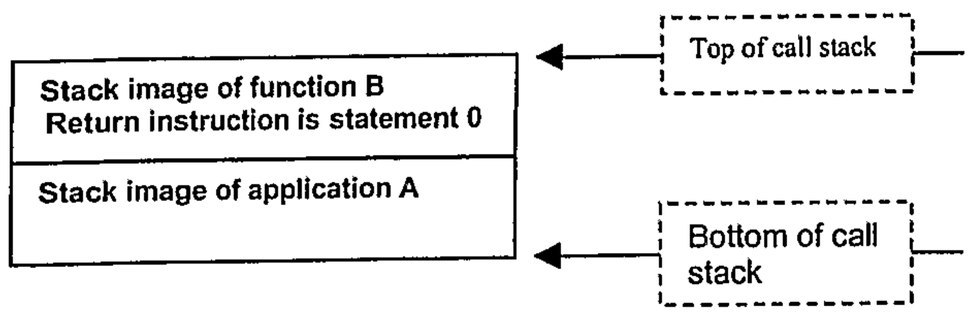


Figure 8.

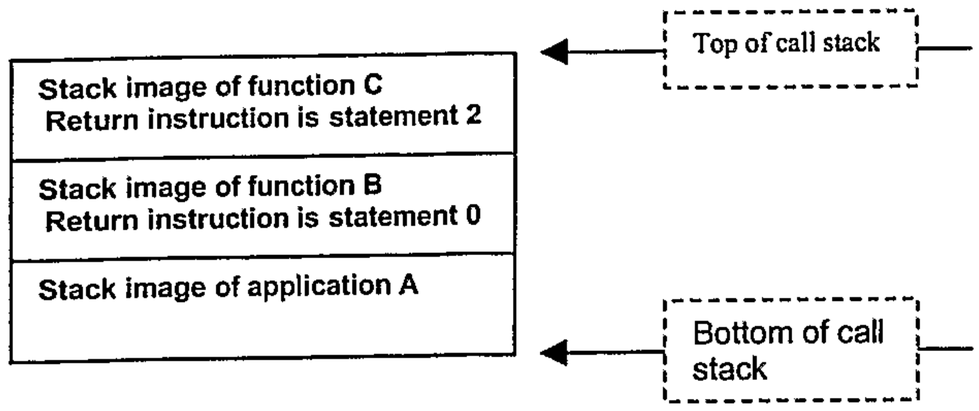


Figure 9.

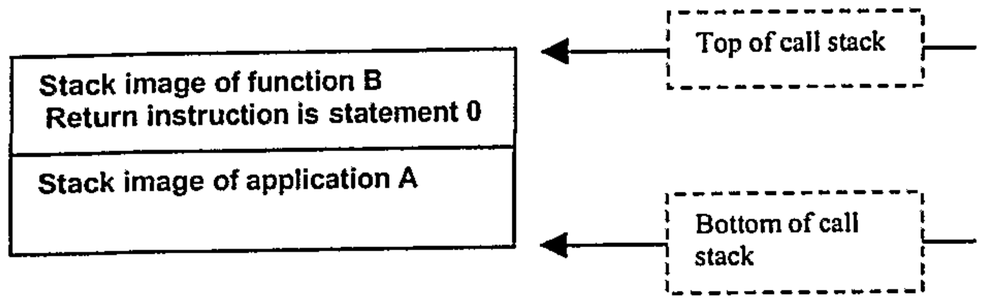
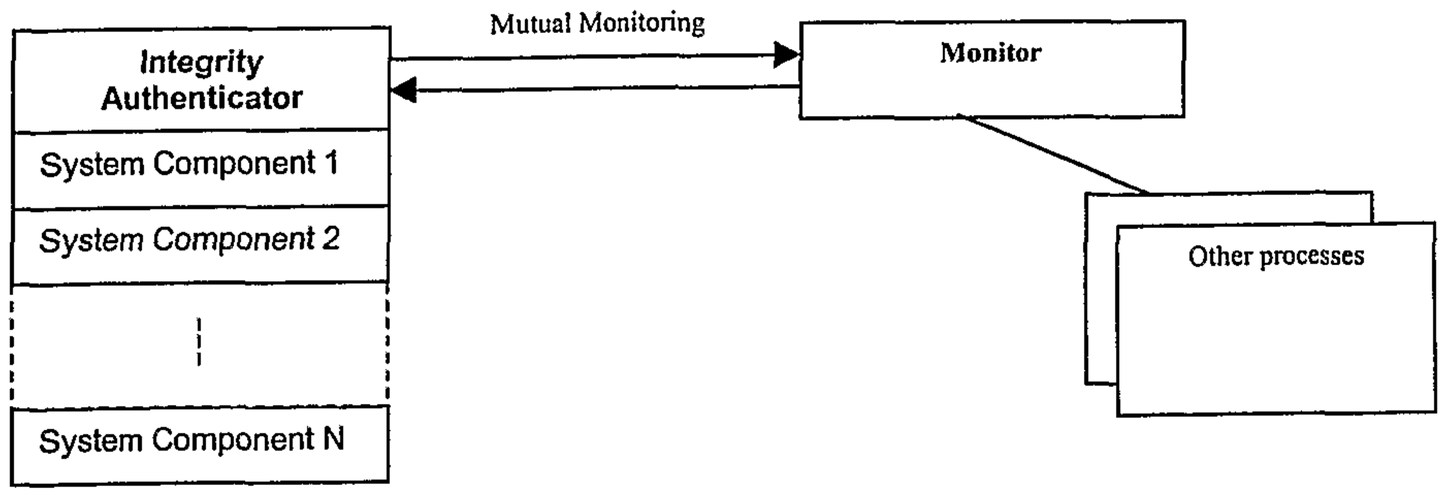


Figure 10. Protection of the execution environment by preventing dynamic tampering using monitoring, such as debugging



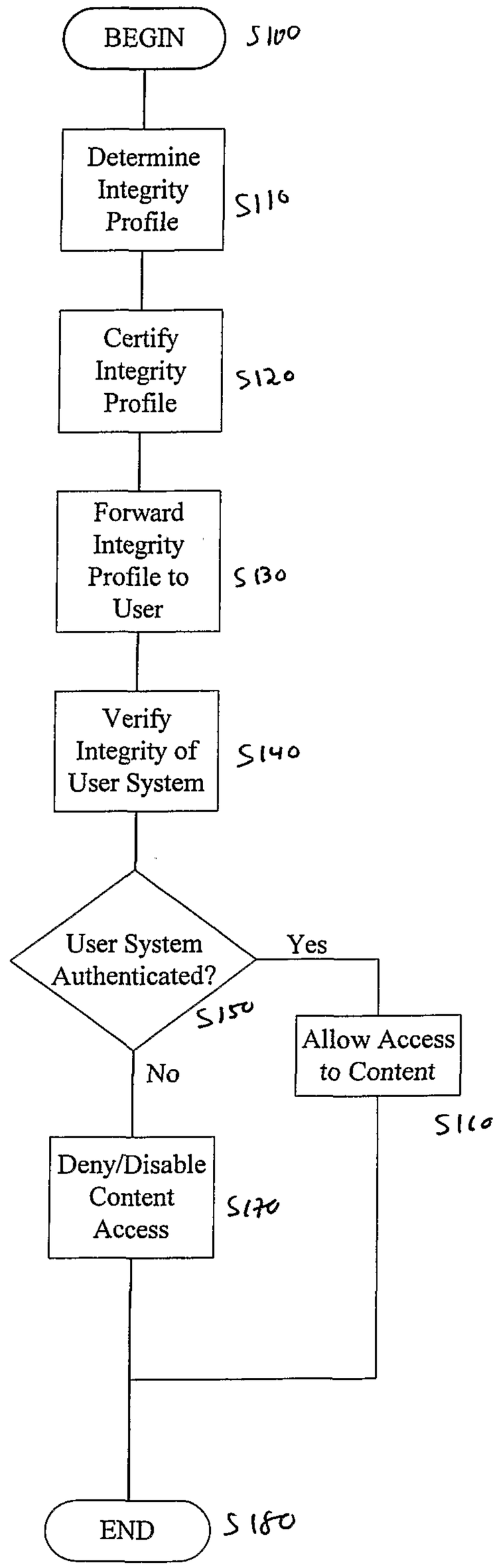

BEGIN
S100
Determine Integrity Profile
S110
Certify Integrity Profile
S120
Forward Integrity Profile to User
S130
Verify Integrity of User System
S140
User System Authenticated?
S150
Yes
No
Allow Access to Content
S160
Deny/Disable Content Access
S170
END
S180

Fig. 11

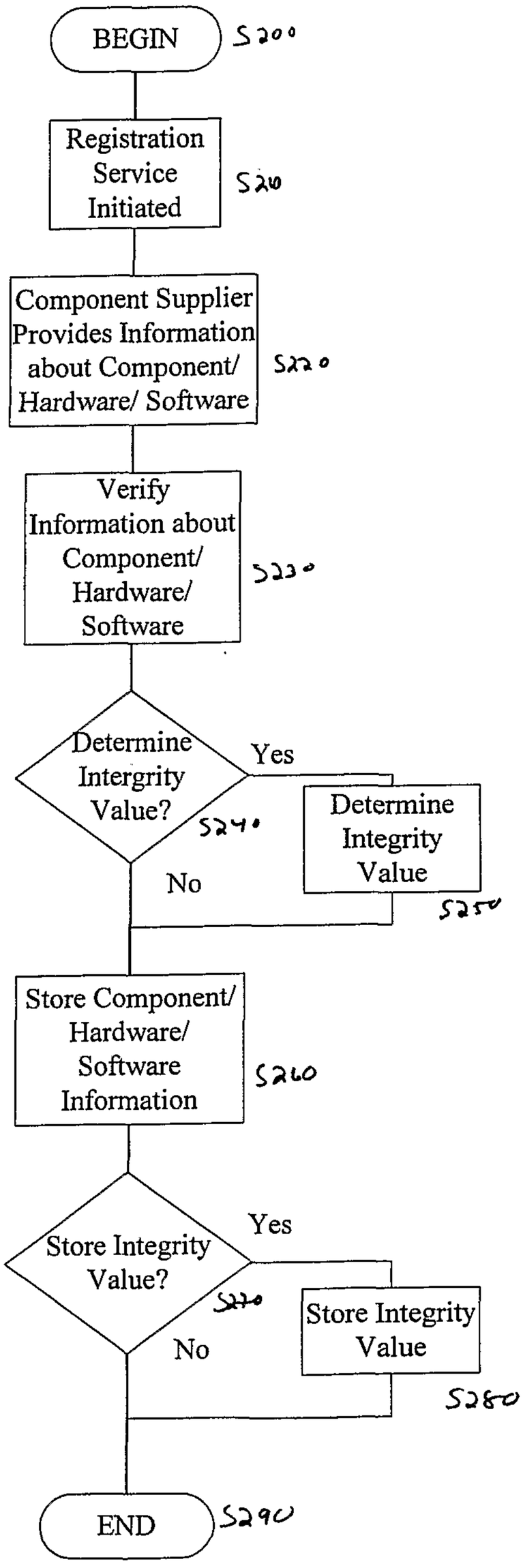

BEGIN
S200
Registration Service Initiated
S210
Component Supplier Provides Information about Component/ Hardware/ Software
S220
Verify Information about Component/ Hardware/ Software
S230
Determine Intergrity Value?
S240
Yes
No
Determine Integrity Value
S250
Store Component/ Hardware/ Software Information
S260
Store Integrity Value?
S270
Yes
No
Store Integrity Value
S280
END
S290

Fig. 12

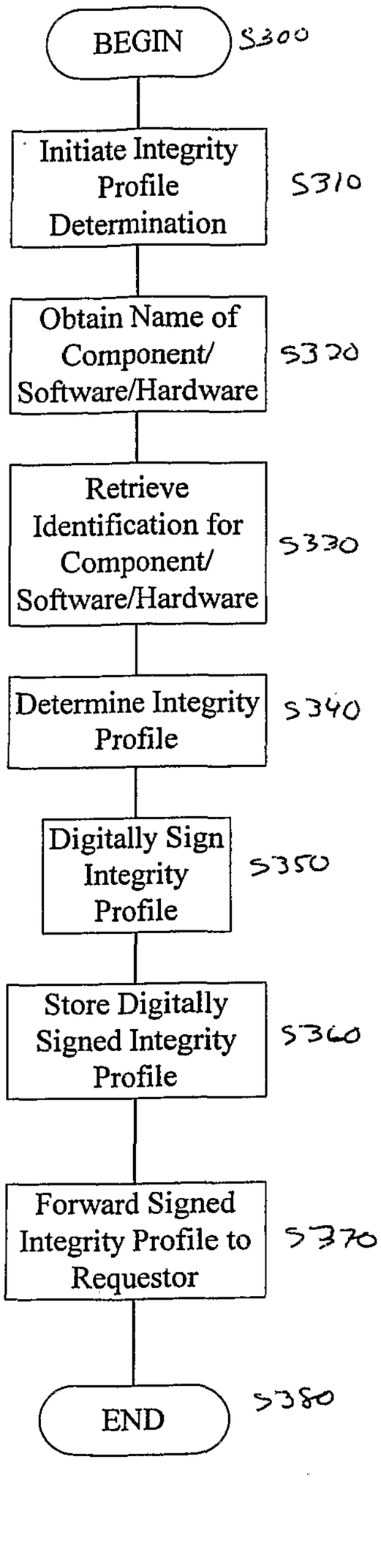

BEGIN
S300
Initiate Integrity Profile Determination
S310
Obtain Name of Component/ Software/Hardware
S320
Retrieve Identification for Component/ Software/Hardware
S330
Determine Integrity Profile
S340
Digitally Sign Integrity Profile
S350
Store Digitally Signed Integrity Profile
S360
Forward Signed Integrity Profile to Requestor
S370
END
S380

Fig. 13

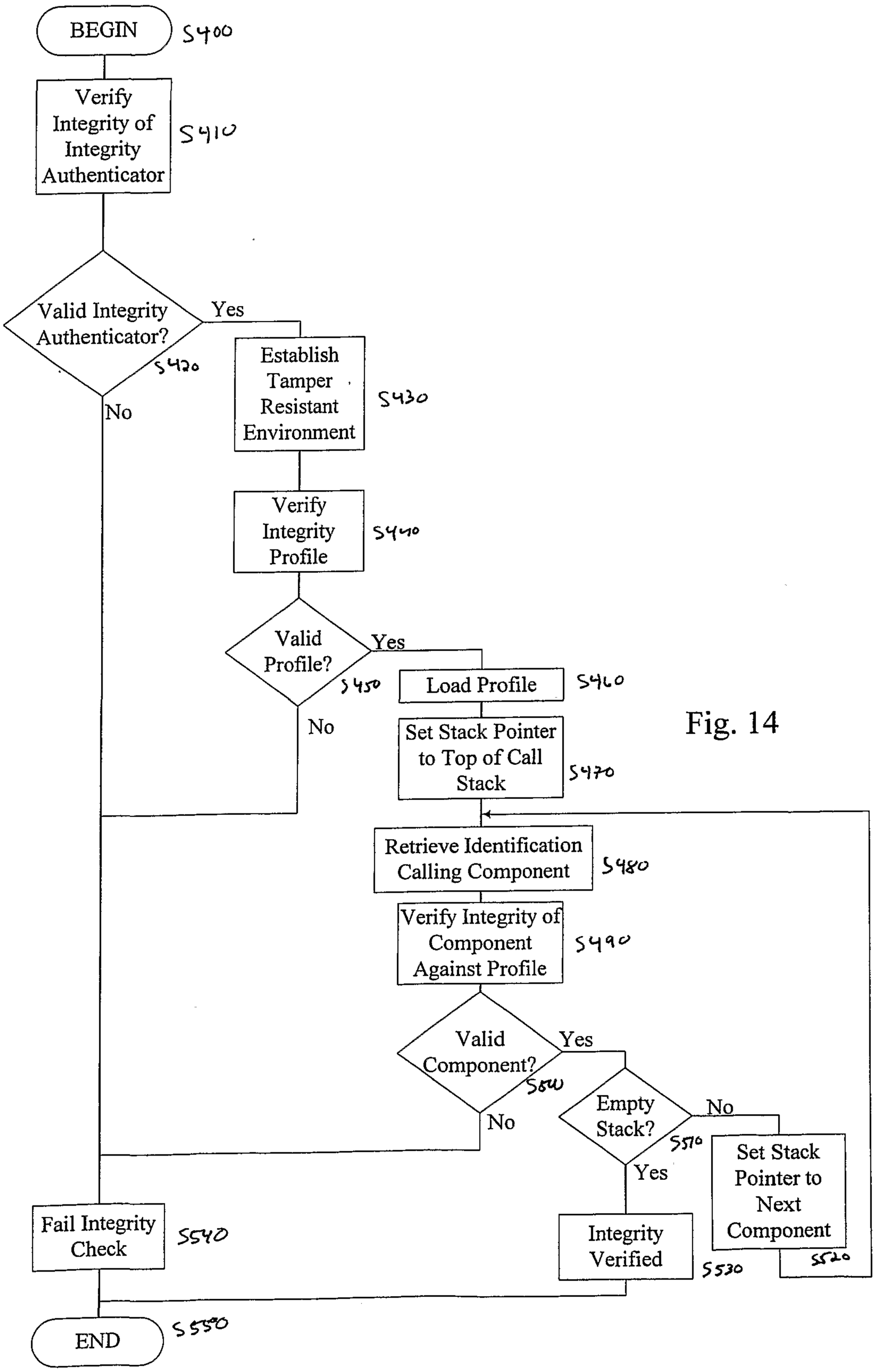
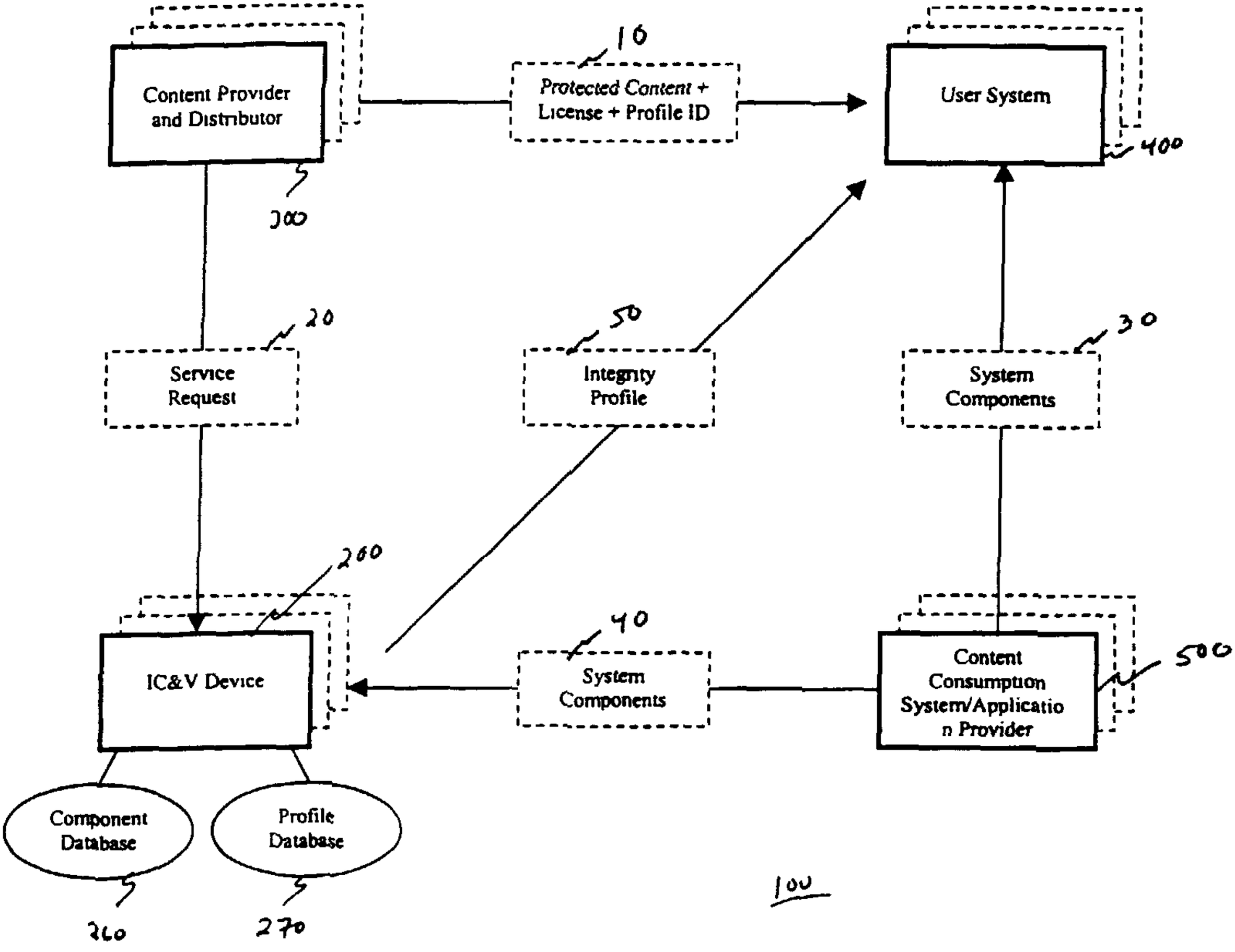

BEGIN
S400
Verify Integrity of Integrity Authenticator
S410
Valid Integrity Authenticator?
S420
Yes
No
Establish Tamper Resistant Environment
S430
Verify Integrity Profile
S440
Valid Profile?
S450
Yes
No
Load Profile
S460
Set Stack Pointer to Top of Call Stack
S470
Retrieve Identification Calling Component
S480
Verify Integrity of Component Against Profile
S490
Valid Component?
S500
Yes
No
Empty Stack?
S510
No
Yes
Set Stack Pointer to Next Component
S520
Integrity Verified
S530
Fail Integrity Check
S540
END
S550

Fig. 14


Content Provider and Distributor
300
16
Protected Content + License + Profile ID
User System
400
20
Service Request
50
Integrity Profile
30
System Components
260
IC&V Device
40
System Components
Content Consumption System/Application Provider
500
Component Database
Profile Database
260
270
100