

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
29 November 2007 (29.11.2007)

PCT

(10) International Publication Number
WO 2007/136489 A2

(51) International Patent Classification: **Not classified**

(21) International Application Number:
PCT/US2007/009668

(22) International Filing Date: 18 April 2007 (18.04.2007)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
11/434,313 15 May 2006 (15.05.2006) US

(71) Applicant (for all designated States except US): **THE BOEING COMPANY** [US/US]; 100 North Riverside Plaza, Chicago, IL 60606-2016 (US).

(72) Inventors; and

(75) Inventors/Applicants (for US only): **BALZA, Richard, M.** [US/US]; 13316 SE 183rd Street, Renton, WA 98058 (US). **UNG, Kevin, Y.** [US/US]; 14830 SE 58th Street, Bellevue, WA 98006 (US). **ARNOLD, Steven, L.** [US/US]; 10807 106th Avenue NE, Kirkland, WA 98033 (US). **DONOFRIO, Thomas, E.** [US/US]; 2100 South 375th Street, Federal Way, WA 98003 (US). **MURPHY,**

Michael, J. [US/US]; 10 Tesoro, Newport Coast, CA 92657 (US). **HARTMAN, Brett** [US/US]; 6043 44th Avenue NE, Seattle, WA 98115 (US). **KWOK, Samuel, C.** [US/US]; 14901 SE 60th Street, Bellevue, WA 98006 (US).

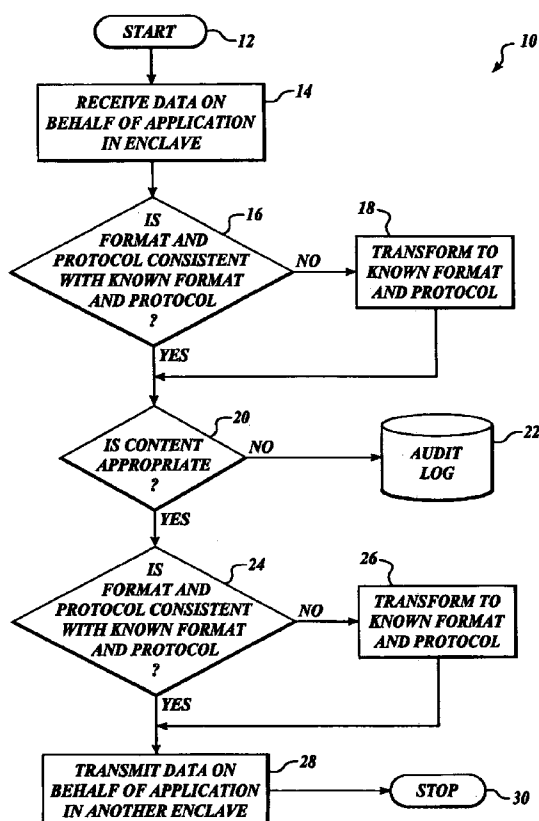
(74) Agents: **HALPERIN, Brett, L.** et al.; The Boeing Company, P.O. Box 2515, MC 110-SD54, Seal Beach, CA 90740-1515 (US).

(81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, SV, SY, 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,

[Continued on next page]

(54) Title: MULTIPLE LEVEL SECURITY ADAPTER



(57) Abstract: In exemplary embodiments, data with a format compatible with a first protocol standard is received on behalf of a first application. When the format of the data is not compatible with a second protocol standard, the format of the data is automatically transformed to a format that is compatible with the second protocol standard. The data is transmitted to a second application service using the second protocol standard. The data may be received from the second application. When the format of the data is not compatible with a third protocol standard, the format of the data is automatically transformed to a format that is compatible with the third protocol standard. The data is transmitted on behalf of a third application using the third protocol standard. The first and third applications may be in first and second protected enclaves. The second application may include a security gateway service.



GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, MT, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

— *without international search report and to be republished upon receipt of that report*

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.

5

MULTIPLE LEVEL SECURITY ADAPTER**BACKGROUND**

Several technological approaches and managerial procedures have been developed to help ensure availability, integrity, and confidentiality of information exchanged and managed by a computer system or network. One such technological approach is to provide and manage
10 protected enclaves (or security enclaves).

In a protected enclave, a community-of-interest is established in which users having a need-to-know run applications within a network that is protected by predetermined security measures. The users may be co-located but need not be co-located. The applications may include, without limitation, electronic mail (Email), instant messaging (IM), file transfer, and the
15 like. The security measures may be directed to protect content that includes national security information (such as confidential, secret, top secret, and the like); financial information; business information; personal information; and the like.

When a protected enclave is established, measures must be taken to enable appropriate sharing or exchange of data between the protected enclave and the outside world. In one
20 approach to sharing data, a secure network server (or a guard server) acts as a gateway to route and filter data exchanged between the protected enclave and the outside world, or between a first protected enclave operating at a first security level and a second protected enclave operating at a second security level that is different (either higher or lower) than the first security level.

Thus, the guard server may filter sensitive information from the data originated by an
25 application in the protected enclave. However, in order for the guard server to filter the data, the guard server must understand how to parse the data. Therefore, both the guard server and the application must be cognizant of format of the data originated by the application.

If the guard server understands how to parse the data, then the guard server can filter the data and pass on the data, if appropriate. As an example, a file transfer application in a protected
30 enclave may transfer files using an FTP protocol in its application layer. The guard server application may also include the FTP protocol in its application layer. In such a case, the guard server understands how to parse the data that the application wants to send out of the protected enclave. Therefore, the guard server filters the information and can route the data accordingly.

On the other hand, if the guard server does not understand how to parse the data, then the guard server will send the data to a suitable "data dump" and may generate an event for a security event log. As an example, an application in a protected enclave may send a web page using an HTTP protocol in its application layer. However, the guard server application may not include the HTTP protocol in its application layer. In such a case, the guard server does not understand how to parse the data that the application wants to send out of the protected enclave. Therefore, the guard server cannot filter the information and cannot route the data accordingly. Instead, the guard server will send the data to a suitable "data dump" and may generate an event for a security event log.

In currently known approaches, the filtering described above is performed by the guard server. In order for the guard server to recognize a data format that was previously unrecognizable, the guard server would have to be reprogrammed to recognize the desired data format. Alternately, all of the machines running applications in the protected application would have to be reprogrammed to recognize all data formats that are recognized by the guard server. Either approach is extremely time-and-labor intensive and, therefore, is cost prohibitive.

As a result, the design of currently known guard servers is tightly coupled to the applications executed in the protected enclaves. This tight coupling can reduce flexibility of design of the guard server and of the applications; can reduce re-use of software; can increase "lock-in" and dependence on certain vendors; and can increase integration cost.

Therefore, it would be desirable for a guard server to be able to recognize data formats of applications executed in the protected enclaves without reprogramming the guard server to recognize a previously unrecognizable data format and without reprogramming all machines running applications in a protected enclave to recognize all data formats that are recognized by the guard server.

The foregoing examples of related art and limitations associated therewith are intended to be illustrative and not exclusive. Other limitations of the related art will become apparent to those of skill in the art upon a reading of the specification and a study of the drawings.

SUMMARY

The following embodiments and aspects thereof are described and illustrated in conjunction with systems and methods which are meant to be exemplary and illustrative, not

limiting in scope. In various embodiments, one or more of the problems described above in the Background have been reduced or eliminated, while other embodiments are directed to other improvements.

Exemplary embodiments provide computer executable methods, computer program
5 products, and systems for exchanging data. In exemplary embodiments, data is received on behalf of a first application using a first protocol standard. The data has a format compatible with the first protocol standard. A determination is made whether the format of the data is compatible with a second protocol standard. When the format of the data is not compatible with the second protocol standard, the format of the data is automatically transformed to a format that
10 is compatible with the second protocol standard. The data is transmitted to a second application service using the second protocol standard.

According to an aspect, the data may be received from the second application using the second protocol standard. The data has a format compatible with the second protocol standard. A determination is made whether the format of the data is compatible with a third protocol
15 standard. When the format of the data is not compatible with the third protocol standard, the format of the data is automatically transformed to a format that is compatible with the third protocol standard. The data is transmitted on behalf of a third application using the third protocol standard.

According to other aspects, the first application may be provided within a first protected
20 enclave, the second application may include a security gateway service, and the third application may be provided within a second protected enclave.

According to further aspects, the first protocol standard may define a first security level and the third protocol standard may define a second security level.

In addition to the exemplary embodiments and aspects described above, further
25 embodiments and aspects will become apparent by reference to the drawings and by study of the following detailed description.

BRIEF DESCRIPTION OF THE DRAWINGS

Exemplary embodiments are illustrated in referenced figures of the drawings. It is intended that the embodiments and figures disclosed herein are to be considered illustrative rather than restrictive.

5 FIGURE 1 is a flow chart of an exemplary method of exchanging data;

FIGURE 2 is a software block diagram of exemplary software processing blocks that implement the method of FIGURE 1;

FIGURE 3 is a software block diagram showing non-limiting details of the processing blocks of FIGURE 2; and

10 FIGURE 3 is a block diagram of an exemplary host environment that implements the method of FIGURE 1.

DETAILED DESCRIPTION

By way of overview and referring to FIGURE 1, at a block 14 data is received on behalf of a first application in a first protected enclave using a first protocol standard. The data has a
15 format compatible with the first protocol standard, and the first protocol standard may define a first security level. At a decision block 16, a determination is made whether the format of the data is compatible with a second protocol standard. When the format of the data is not compatible with the second protocol standard, at a block 18 the format of the data is automatically transformed to a format that is compatible with the second protocol standard. The
20 data is transmitted to a second application service, such as a security gateway service, using the second protocol standard. The data may be received from the second application in a second protected enclave using the second protocol standard. The data has a format compatible with the second protocol standard. At a decision block 24, a determination is made whether the format of the data is compatible with a third protocol standard. The third protocol standard may define a
25 second security level. When the format of the data is not compatible with the third protocol standard, at a block 26 the format of the data is automatically transformed to a format that is compatible with the third protocol standard. At a block 28 the data is transmitted on behalf of a third application using the third protocol standard. Details of exemplary embodiments will now be set forth below.

Exemplary Method

Still referring to FIGURE 1, an exemplary method 10 starts at a block 12. At the block 14, data is received on behalf of a first application in a first protected enclave using a first protocol standard. The protected enclave is a community-of-interest in which users having a
5 need-to-know run applications within a network that is protected by predetermined security measures. The users may be co-located but need not be co-located. The applications may include, without limitation, electronic mail (Email), instant messaging (IM), file transfer, and the like. The information exchanged within the protected enclave may include without limitation national security information, business information, financial information, personal information,
10 health care information, and the like.

The data received on behalf of the application at the block 16 has a format compatible with the first protocol standard. The data can be transferred in one of two modes: ASCII mode; and binary mode. In ASCII mode, individual letters, numbers, and characters are transferred using their ASCII character codes. When received, the data is saved in a text file in an
15 appropriate format. In binary mode, the data is sent bit for bit. When received, the data is stored by the receiving machine in its own format.

The protocol standard, sometimes referred to as a protocol stack, defines the data format and is the set of protocols for layers of interconnection, such as those set forth in the Open Systems Interconnection (OSI) model. As used herein, the term protocol standard refers to the
20 protocols implemented in software in the layers including and above the transport layer – that is, the transport layer and the application layer. While a protocol stack may also be considered to include the session layer and the presentation layer, in practice the transport layer, the session layer, the presentation layer and the application layer typically are combined into the transport layer and the application layer. As such and as used herein, the term protocol standard refers to
25 the transport layer and the application layer as used in practice – that is, that combines the transport layer, the session layer, the presentation layer and the application layer. By way of further explanation, such usage also means that the term protocol standard does not refer to the physical layer, the data link layer, or the network layer (or the layer sometimes referred to as “layer 2.5”).

30 Examples of the protocol standards are set forth in Table 1 for exemplary applications on behalf of which data can be received (or transmitted, such as at the block 28).

TABLE 1

Application	Transport Layer	Application Layer
Email	TCP	SMTP
Instant messaging	TCP	XMPP
File transfer	TCP	FTP
Web browser	TCP	HTTP
Pub/sub	TCP	JMS, SOAP, or RDDS

The first protocol standard may define a first security level. For example, the transport and application layers may support encryption of the data. As another example, security labeling may be put in messages at the application layer.

At a decision block 16, a determination is made whether the format of the data is compatible with a second protocol standard. The second protocol standard is a protocol stack for a second application. In an exemplary embodiment, the second application is a guard service, such as that for a secure network server between a protected enclave and the outside world or between two protected enclaves. An exemplary protocol standard for a guard service generally includes TCP for use at the transport layer. Also, some guard protocol standards may require security labeling at the application layer and/or the transport layer. Given by way of non-limiting example, the determination of compatibility at the decision block 16 may be made by determining whether the first application is a type of application that uses a data format that is predetermined to be expected by the guard service.

When the format of the data received on behalf of the application at the block 14 is not compatible with the second protocol standard, at a block 18 the format of the data is automatically transformed to a format that is compatible with the second protocol standard. In an exemplary embodiment, a transformation routine may be selected from a library of transformation routines and performed as desired for a particular application. For example, the library of transformation routines may include a generic transformation function, such as an XSLT transformation engine or the like. The library of transformation routines may also include routines that have been customized for desired applications.

The following non-limiting examples illustrate scenarios in which the format of the data received on behalf of the application at the block 14 is not compatible with the second protocol standard. Given by way of non-limiting example, the first application in the protected enclave may be a web browser that is sending a web page. From Table 1, the protocol standard for the web page uses TCP at the transport layer and HTTP at the application layer. The data is transferred in ASCII mode. In this scenario, the second application is a guard service that can filter messages (because its application layer uses an SMTP protocol). But, in this scenario, the guard service does not include an HTTP protocol in its application layer. Thus, the guard service of the second application can understand how to parse information in an Email message but cannot understand how to parse information in a web page. In this scenario, at the block 18, the data format of the web page (in which the application layer uses an HTTP protocol) is transformed to a known format and known protocol that is compatible with the guard service's protocol standard (in which the application layer uses an SMTP protocol). As another example in which the format of the data received on behalf of the application at the block 14 is not compatible with the second protocol standard, a determination may be made at the block 16 that an application layer protocol is intermixed with the data received on behalf of the application. In such a case, at the block 18 the intermixed application layer protocol is automatically removed from the data.

After the format of the data has been transformed at the block 18 or, alternately, when the format of the data received on behalf of the application at the block 14 is compatible with the second protocol standard, at a decision block 20 a determination may be made whether information content of the data is appropriate for sharing outside the protected enclave. The filtering and determination may be effected by keyword matching and comparison against predetermined keywords, or by other text classification or data mining routines, if desired.

Advantageously, such filtering of the data is possible because either the format of the data received on behalf of the application at the block 14 is compatible with the second protocol standard or was transformed at the block 18 to a known format and known protocol that is compatible with the second protocol standard. In either case, the data that was received on behalf of the application at the block 14 advantageously is now compatible with the second protocol standard. When the second application is a guard service, this compatibility represents an advantage over currently known guard services that cannot understand how to parse the data and instead send the data to a suitable "data dump" and merely generate an event for an event log.

When content is determined to be inappropriate, an event is generated and sent to an audit log 22 for later analysis, as desired. Given by way of non-limiting example, content may be determined to be inappropriate when the data contains text that includes predetermined content that is not cleared for access outside the protected enclave.

When content has been determined to be appropriate for sharing outside the protected enclave, the data is shared outside the protected enclave. In another protected enclave at a decision block 24, determination is made whether the format of the data is compatible with a third protocol standard. The third protocol standard is a protocol stack for a third application. Details of the protocol standard and data format for the third application are the same as those for the first application discussed above and, as a result, need not be repeated. The third application in the second protected enclave may be, but need not be, the same application as the first application in the first protected enclave. Alternately, the third application in the second protected enclave may be a different application from the first application in the first protected enclave. The determination at the block 24 suitably is made in the same manner as the determination at the block 16.

When a determination is made at the decision block 24 that the format of the data is not compatible with the third protocol standard, at a block 26 the format of the data is automatically transformed to a format that is compatible with the third protocol standard. The transformation at the block 26 suitably is made in the same manner as the transformation at the block 18.

After the format of the data has been transformed at the block 26 or, alternately, when the format of the data is compatible with the third protocol standard, at a block 28 the data is transmitted on behalf of the third application. The method 10 ends at a block 30.

Exemplary Software

Referring now to FIGURE 2, exemplary software implements processes of the method 10 (FIGURE 1). Advantageously, the exemplary software enables data format to be transformed as desired without a need to reprogram the guard server to recognize a previously unrecognizable data format or to reprogram all machines running applications in a protected enclave to recognize all data formats that are recognized by the guard server. The exemplary software implementation will be discussed in the same order as the discussion of the method 10 (FIGURE 1).

Data from an application 40 in a protected enclave 42 is received from a server 44 on behalf of the application 40 by an application proxy 46. The application proxy 46 enables communication with the application 40 using the particular protocol standard of the application 40. The protocol standard has already been described above in the context of the method 10 (FIGURE 1).

A transformation analyzer 48 analyzes the data received by the application proxy 46 on behalf of the application 40 and makes a determination whether the format of the data is compatible with a protocol standard for a guard service 50. In an exemplary embodiment, logic is provided by an operator. The logic decides which particular application data is to be transformed. For example, the logic may identify the type of the application 40. The logic may also identify the type of data that came from the application 40. Given by way of non-limiting example, the type of data may be identified as binary data or ASCII data, such as without limitation HTML or XML. The transformation analyzer 48 may compare the type of the application 40 against a list of predetermined application types that send data to the guard service that the guard service 50 is expecting. The transformation analyzer 48 may also compare the type of the data that came from the application 40 against a list of predetermined data types that the guard service 50 is expecting. Given by way of a non-limiting, illustrative example, the guard service 50 may be expecting ASCII SMTP data from an Email application. In this case, a list of predetermined applications includes Email applications but not web browsers, and a list of predetermined data types includes ASCII SMTP data but not ASCII HTML data. However, the application 40 may be a web browser that sends ASCII HTML data. The transformation analyzer 48 identifies the application 40 as a web browser and may identify the data type as ASCII HTML. The transformation analyzer compares the identified web browser application and/or the identified ASCII HTML data against the predetermined list that does not include web

browsers or ASCII HTML data. The transformation analyzer thereby determines that, in this non-limiting example, the data is not compatible with the guard service 50 and is to be transformed. The analysis performed by the transformation analyzer 48 implements the process of the decision block 16 (FIGURE 1).

5 When the format of the data received by the application proxy 46 on behalf of the application 42 is not compatible with the protocol standard of the guard service 50, a transformation service 52 automatically transforms the format of the data to a format that is compatible with the protocol standard of the guard service 50. The transformation performed by the transformation service 50 implements the process of the block 18 (FIGURE 1). Given by
10 way of non-limiting example, the transformation service 52 may be implemented as an XSLT transformation engine or the like.

 After the format of the data has been transformed by the transformation service 52 or, alternately, when the format of the data received on behalf of the application 40 by the application proxy 46 is compatible with the protocol standard of the guard service 50, a guard
15 proxy 54 enables communication with the guard service 50 using the protocol standard of the guard service 50 (such as by using TCP at the transport layer).

 Advantageously, the application proxy 46, the transformation analyzer 48, the transformation engine 52, and the guard proxy 54 may be provided as a software suite 56. Advantageously, the software suite 56 resides in the protected enclave 42 – but not in a guard
20 server (not shown) that provides the guard service. Providing the software suite 56 outside the guard server advantageously enables data format to be transformed as desired without a need to reprogram the guard server to recognize a previously unrecognizable data format or to reprogram all machines running applications in the protected enclave 42 to recognize all data formats that are recognized by the guard server.

25 Referring additionally to FIGURE 3, additional details will be set forth regarding the software suite 56. The application proxy 46 may include several application proxies – one for each application resident in the protected enclave 42. Given by way of non-limiting example, the application proxy may include a pub/sub proxy 46A, an instant messaging proxy 46B, and a file transfer proxy 46C. The pub/sub proxy 46A enables communication with a pub/sub
30 application 40A. The instant messaging proxy 46B enables communication with an instant messaging application 40B. The file transfer proxy 46C enables communication with a file

transfer application 40C. As an additional application(s) 40 is added to the protected enclave 42, the application proxy 46 is simply re-programmed to add a corresponding application proxy(ies) to enable communication with the added application(s). Further, if for some reason a change is made to the protocol standard of any of the applications 40 in the protected enclave, the application proxy 46 simply is reprogrammed. Thus, there is no need to reprogram all machines running applications in the protected enclave 42 to recognize all data formats that are recognized by the guard server.

Further, if the guard server is replaced or if a change is made to the protocol standard of the guard service, then the guard proxy 54 simply is reprogrammed. Advantageously, in such a case there is no need to reprogram the guard server.

Referring back to FIGURE 2, the data is transmitted to the guard service 50 for ultimate reception by an application 40' in a protected enclave 42'. The guard service 50 receives the data and filters the data to make a determination whether information content of the data is appropriate for sharing outside the protected enclave 42. Thus, the guard service implements processing of the decision block 20 (FIGURE 1).

In the protected enclave 42', a software suite 56' includes a guard proxy 54', a transformation analyzer 48', a transformation service 52', and an application proxy 46'. The guard proxy 54' receives the data from the guard service 50. The guard proxy 54', the transformation analyzer 48', the transformation service 52', and the application proxy 46' perform the functions described above for the guard proxy 54, the transformation analyzer 48, the transformation service 52, and the application proxy 46 in a mirror-image fashion. Thus, the data that is received by the guard proxy 54' is processed and is ultimately transmitted on behalf of the application 40' via a server 44'.

Exemplary Network Environment

Referring now to FIGURE 4, the software suites 56 and 56' (FIGURE 2) reside in storage (not shown) and are executed by processors (not shown) in computers 60 and 60', such as routers, in the protected enclaves 42 and 42'. The software suites 56 and 56' can each be executed by processors provided in one respective computer 60 and 60' or, if desired, execution of the software suites 56 and 56' can be distributed among more computers. Computers are

extremely well known in the art and, therefore, a discussion of their construction and operation is not necessary.

The protected enclave 42 can include users running applications on a variety of computing machines. The protected enclave 42 may represent a community of interest of users with a need to know at a certain level of security. The users may be co-located but need not be co-located. The computing machines of the protected enclave 42 may communicate with each other and the computer 60 directly (not shown) or via a network 60, such as without limitation the Internet. The protected enclave 42 can include any computing machine as desired, such as without limitation a computer 64, a pen computer 66, a hand-held computer 68, a laptop computer 70, and the like. The protected enclave 42 can also include computing machines within a mobile platform 72, such as without limitation an airplane, whose connectivity is provided via a satellite 74 and a ground station 76 (such as the mobile connectivity provided by Connexion by Boeing). Also, the satellite 74 may provide connectivity to a computer 78 via a ground station 80.

The protected enclave 42' may include similar components as the protected enclave 42. Each component of the protected enclave 42' that is similar to a component of the protected enclave 42 is denoted with a same reference number that is appended with a prime (''). The protected enclave 42' may represent a community of interest of users with a need to know at a level of security that is different (that is, either higher or lower) than the security level of the protected enclave 42.

A guard server 82 provides the guard service 50 (FIGURE 2) and is directly connected to the computers 60 and 60'. Advantageously, as discussed above, the software suites 56 and 56' do not reside on the guard server 82. Instead, as discussed above, the software suites 56 and 56' reside on the computers 60 and 60', respectively, or are distributed among more computers (that are not the guard server 82) in their respective protected enclaves. Therefore, execution of the software suites 56 and 56' on computers within the protected enclaves 42 and 42' (instead of on the guard server 82) enables data format to be transformed as desired without a need to reprogram the guard server 82 to recognize a previously unrecognizable data format or to reprogram all machines running applications in the protected enclaves 42 and 42' to recognize all data formats that are recognized by the guard server 82.

While a number of exemplary embodiments and aspects have been illustrated and discussed above, those of skill in the art will recognize certain modifications, permutations, additions, and sub-combinations thereof. It is therefore intended that the following appended claims and claims hereafter introduced are interpreted to include all such modifications, permutations, additions, and sub-combinations as are within their true spirit and scope.

5

CLAIMS

What is claimed is:

5 1. A computer executable method of exchanging data, the method comprising:

receiving data on behalf of a first application using a first protocol standard, the data having a format compatible with the first protocol standard;

10 determining whether the format of the data is compatible with a second protocol standard;

when the format of the data is not compatible with the second protocol standard, automatically transforming the format of the data to a format that is compatible with the second protocol standard; and

15 transmitting the data to a second application service using the second protocol standard.

2. The method of Claim 1, further comprising:

receiving the data from the second application using the second protocol standard, the data having a format compatible with the second protocol standard;

20 determining whether the format of the data is compatible with a third protocol standard;

when the format of the data is not compatible with the third protocol standard, automatically transforming the format of the data to a format that is compatible with the third protocol standard; and

25 transmitting the data on behalf of a third application using the third protocol standard.

3. The method of Claim 2, wherein the first application is provided within a first protected enclave.

4. The method of Claim 3, wherein the second application includes a security gateway service.

5 5. The method of Claim 4, wherein the third application is provided within a second protected enclave.

6. The method of Claim 5, wherein:

the first protocol standard defines a first security level; and

the third protocol standard defines a second security level.

10 7. The method of Claim 1, further comprising:

determining whether an application layer protocol is intermixed with the data received on behalf of the first application; and

15 when an application layer protocol is intermixed with the data received on behalf of the first application, automatically removing the intermixed application layer protocol from the data.

8. The method of Claim 2, wherein the first and second and third protocol standards each include a respective transport layer protocol and a respective application layer protocol.

9. The method of Claim 1, wherein determining includes:

20 identifying a type of the first application; and

comparing the type of the first application against a list of predetermined application types.

10. The method of Claim 9, wherein determining further includes:

identifying a type of data received on behalf of the first application; and

comparing the type of the data received on behalf of the first application against a list of predetermined data types.

11. The method of Claim 1, wherein transforming is performed by an XSLT transformation engine.

5 12. A computer readable program product comprising:

first computer readable program code means for receiving data on behalf of a first application using a first protocol standard, the data having a format compatible with the first protocol standard;

10 second computer readable program code means for determining whether the format of the data is compatible with a second protocol standard;

third computer readable program code means for, when the format of the data is not compatible with the second protocol standard, automatically transforming the format of the data to a format that is compatible with the second protocol standard; and

15 fourth computer readable program code means for transmitting the data to a second application using the second protocol standard.

13. The computer readable program product of Claim 12, further comprising:

20 fifth computer readable program code means for receiving the data from the second application using the second protocol standard, the data having a format compatible with the second protocol standard

sixth computer readable program code means for determining whether the format of the data is compatible with a third protocol standard;

25 seventh computer readable program code means for, when the format of the data is not compatible with the third protocol standard, automatically transforming the format of the data to a format that is compatible with the third protocol standard; and

eight computer readable program code means for transmitting the data on behalf of a third application using the third protocol standard.

14. The computer readable program product of Claim 13, wherein the first computer readable program code means includes an application proxy.

15. The computer readable program product of Claim 14, wherein the second computer readable program code means includes a security gateway proxy.

16. The computer readable program product of Claim 12, further comprising:

ninth computer readable program code means for determining whether an application layer protocol is intermixed with the data received on behalf of the first application; and

tenth computer readable program code means for, when an application layer protocol is intermixed with the data received on behalf of the first application, automatically removing the intermixed application layer protocol from the data.

17. The computer readable program product of Claim 12, wherein the third computer readable program code means includes an XSLT transformation engine.

18. The computer readable program product of Claim 12, wherein the second computer readable program code means includes:

eleventh computer readable program code means for identifying a type of the first application; and

twelfth computer readable program code means for comparing the type of the first application against a list of predetermined application types.

19. The computer readable program product of Claim 18, wherein the second computer readable program code means further includes:

thirteenth computer readable program code means for identifying a type of data received on behalf of the first application; and

5 fourteenth computer readable program code means for comparing the type of the data received on behalf of the first application against a list of predetermined data types.

20. A system for exchanging data, the system comprising:

10 a first computer processing component configured to receive data on behalf of a first application using a first protocol standard, the data having a format compatible with the first protocol standard;

a second computer processing component configured to determine whether the format of the data is compatible with a second protocol standard;

15 a third computer processing component configured to, when the format of the data is not compatible with the second protocol standard, automatically transform the format of the data to a format that is compatible with the second protocol standard; and

20 a fourth computer processing component configured to transmit the data to a second application service using the second protocol standard.

21. The system of Claim 20, further comprising:

a fifth computer processing component configured to receive the data from the second application using the second protocol standard, the data having a format compatible with the second protocol standard;

25 a sixth computer processing component configured to determine whether the format of the data is compatible with a third protocol standard;

a seventh computer processing component configured to, when the format of the data is not compatible with the third protocol standard, automatically transform the format of the data to a format that is compatible with the third protocol standard; and

5 an eighth computer processing component configured to transmit the data on behalf of a third application using the third protocol standard.

22. The system of Claim 20, wherein the first, second, third, and fourth computer processing components are provided in one computer processor.

10 23. The system of Claim 20, wherein the first, second, third, and fourth computer processing components are distributed among more than one computer processor.

24. The system of Claim 21, wherein the first, second, third, and fourth computer processing components are provided within a first protected enclave.

15 25. The system of Claim 24, wherein the second application includes a security gateway service implemented on a secure network server.

26. The system of Claim 25, wherein the fifth, sixth, seventh, and eighth computer processing components are provided within a first protected enclave.

20 27. The system of Claim 20, further comprising:

a ninth computer processing component configured to determine whether an application layer protocol is intermixed with the data received on behalf of the first application; and

25 a tenth computer processing component configured to, when an application layer protocol is intermixed with the data received on behalf of the first application, automatically remove the intermixed application layer protocol from the data.

28. The system of Claim 20, wherein the second computer processing component includes:

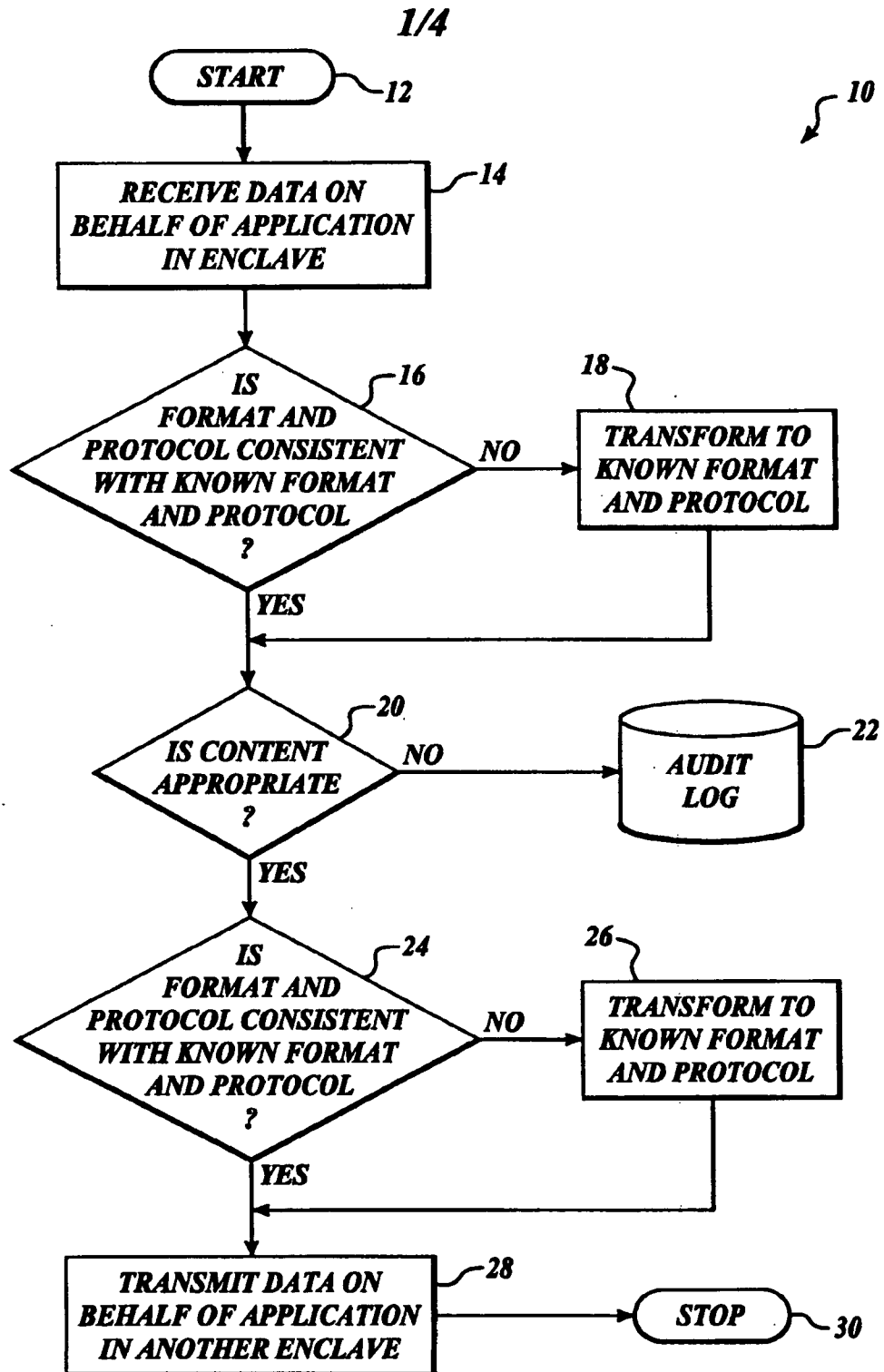
an eleventh computer processing component configured to identify a type of the first application; and

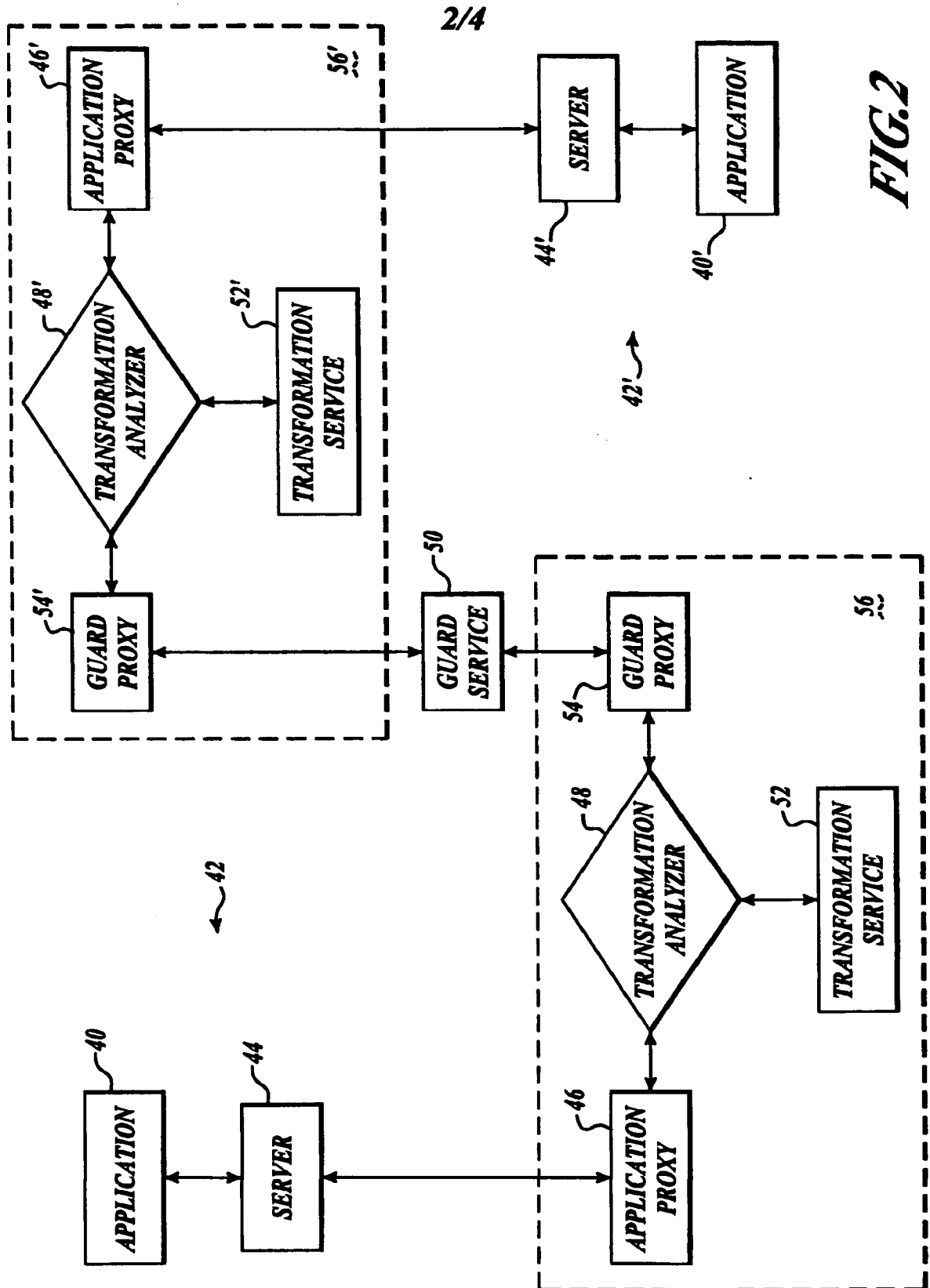
5 a twelfth computer processing component configured to compare the type of the first application against a list of predetermined application types.

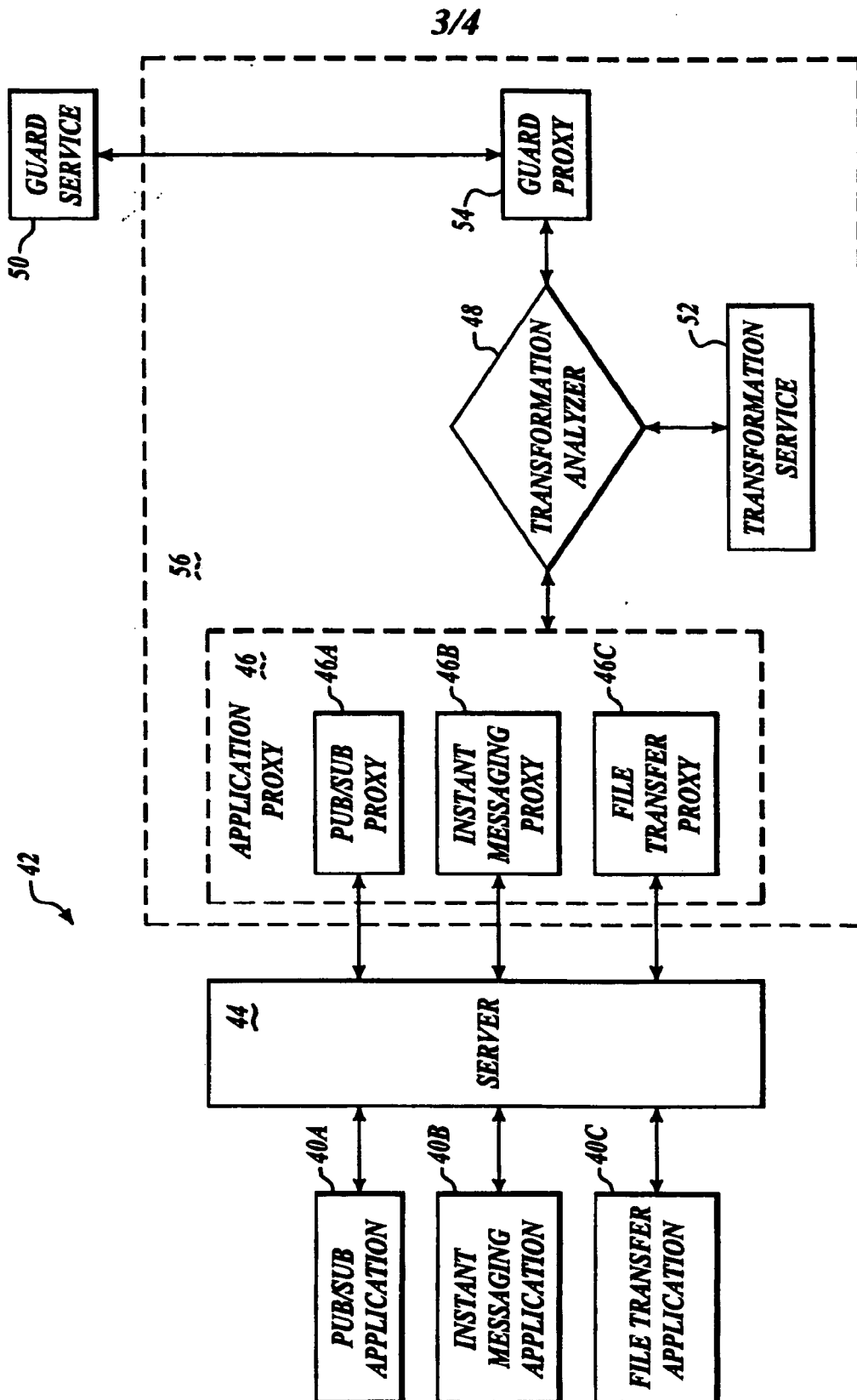
29. The system of Claim 28, wherein the second computer processing component further includes:

10 a thirteenth computer processing component configured to identify a type of data received on behalf of the first application; and

a fourteenth computer processing component configured to compare the type of the data received on behalf of the first application against a list of predetermined data types.

**FIG. 1**



**FIG. 3**

4/4

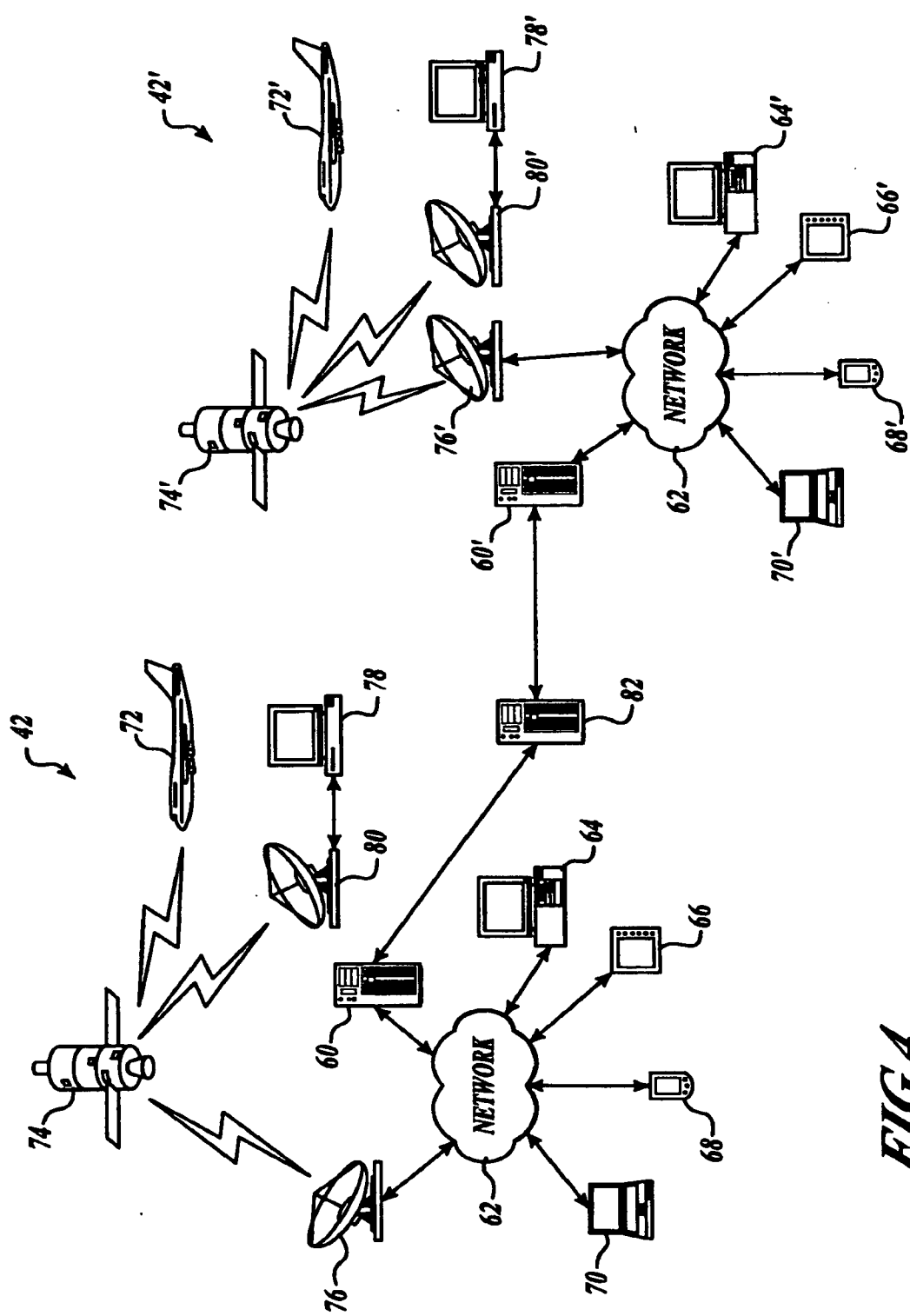


FIG. 4