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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
6 December 2007 (06.12.2007)

PCT

(10) International Publication Number
WO 2007/139696 A2

(51) International Patent Classification: Not classified

(21) International Application Number: PCT/US2007/011637

(22) International Filing Date: 14 May 2007 (14.05.2007)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
11/436,804 18 May 2006 (18.05.2006) US

(71) Applicant (*for all designated States except US*): **MICROSOFT CORPORATION** [US/US]; One Microsoft Way, Redmond, Washington 98052-6399 (US).

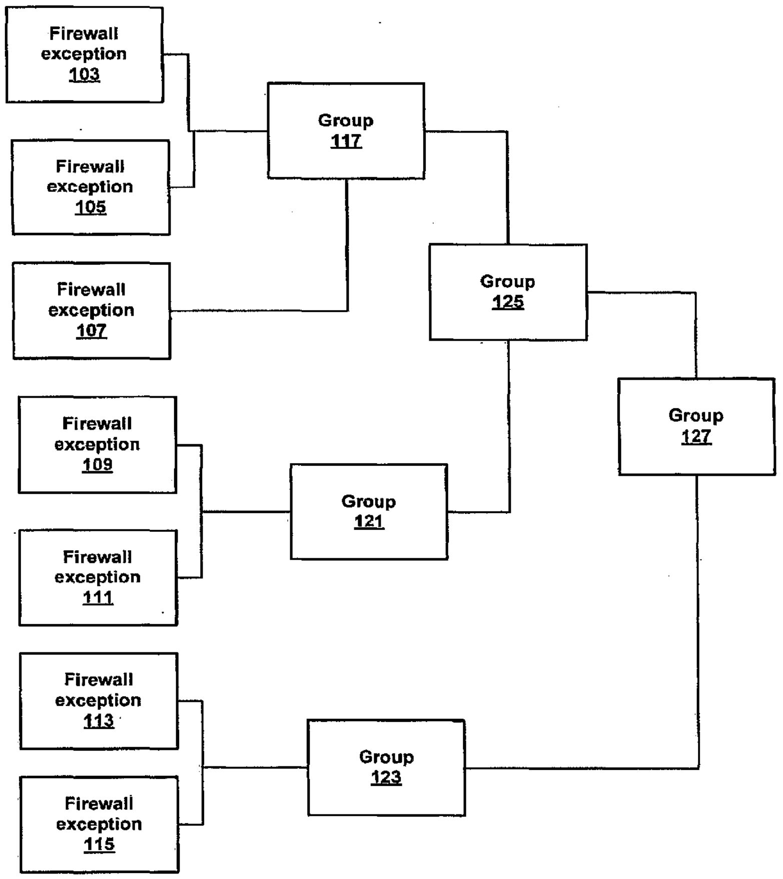
(72) Inventors: **ABZARIAN, David**; c/o Microsoft Corporation, International Patents, One Microsoft Way, Redmond, Washington 98052-6399 (US). **HORTON, Noah**; c/o Microsoft Corporation, International Patents, One Microsoft Way, Redmond, Washington 98052-6399 (US). **TAMASI, David C.**; c/o Microsoft Corporation, International Patents, One Microsoft Way, Redmond, Washington 98052-6399 (US).

(81) **Designated States** (*unless otherwise indicated, for every kind of national protection available*): AE, AG, AL, AM, 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, ME, 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, 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,

[Continued on next page]

(54) Title: EXCEPTIONS GROUPING



(57) **Abstract:** Methods of experience-based exception grouping are described. A number of exceptions are read. The exceptions are intelligently associated with one of a number of exception groups. Each exception group corresponds to a common user experience.

PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Declarations under Rule 4.17:

— *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*

— *as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii))*

Published:

— *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.

EXCEPTIONS GROUPING

TECHNICAL FIELD

[0001] Embodiments of the technology pertain to exceptions management methods.

BACKGROUND

[0002] A firewall is a piece of hardware and/or software that functions in a networked environment to control and/or prevent certain network traffic because of network security reasons. As different end-users often have different needs, a firewall usually can be customized by an end-user. In one example, an end-user may select one or more firewall exceptions to the firewall so that certain types of network traffic are allowed to pass. Specifically, in an operating system, a firewall control panel may allow an end-user to select one or more firewall exceptions (e.g., programs and/or services) to the firewall.

[0003] However, as operating systems evolve to become more sophisticated, the numbers of network facing services and features have increased as well. As a result, a firewall control panel can include a long list of firewall exceptions.

[0004] Unfortunately, a significant number of end-users may not be technically proficient to be able to understand and interact with such a long list of firewall exceptions. Not only would going through such a long list of firewall exceptions be time consuming, an end-user may not know which firewall exceptions to select.

[0005] Moreover, because different application program features associated with firewall exceptions may be interconnected functionally, an end-user that enables or disables a particular firewall exception may result in unintentionally impacting another application program feature.

SUMMARY

[0006] Technologies for experience-based exception grouping are disclosed. The technology includes reading exceptions. (e.g., accessing a list of firewall exceptions) The exceptions are intelligently associated with one or more of a number of exception groups. Each exception group corresponds to a user experience. Thus, rather than selectively choosing amongst a bewildering array of exceptions, ordinary users can intelligently toggle on/off certain exception groups.

BRIEF DESCRIPTION OF THE DRAWINGS

[0007] Figure 1 illustrates block diagrams of a system for grouping firewall exceptions in accordance.with an embodiment of the present claimed subject matter.

[0008] Figures 2A and 2B illustrate block diagrams of an embodiment of the present claimed subject matter in operation.

[0009] Figure 3 illustrates a flowchart of an experience-based method for grouping firewall exceptions upon which embodiments in accordance with the present claimed subject matter can be implemented.

[0010] Figure 4 illustrates a flowchart of a user experience-based method for collapsing firewall exceptions upon which embodiments in accordance with the present claimed subject matter can be implemented.

[0011] Figure 5 illustrates an example of a suitable computing system environment on which the claimed subject matter may be implemented.

DETAILED DESCRIPTION OF THE DRAWINGS

[0012] Reference will now be made in detail to embodiments of the present claimed subject matter, examples of which are illustrated in the accompanying drawings. While the claimed subject matter will be described in conjunction with these embodiments, it will be understood that they are not intended to limit the claimed subject matter to these embodiments. On the contrary, the claimed subject matter is intended to cover alternatives, modifications and equivalents, which may be included within the spirit and scope of the claimed subject matter as defined by the appended claims. Furthermore, in the following detailed description of the present claimed subject matter, numerous specific details are set forth in order to provide a thorough understanding of the present claimed subject matter. However, it will be evident to one of ordinary skill in the art that the present claimed subject matter may be practiced without these specific details. In other instances, well known methods, procedures, components, and circuits have not been described in detail as not to unnecessarily obscure aspects of the claimed subject matter.

[0013] Some portions of the detailed descriptions that follow are presented in terms of procedures, logic blocks, processing, and other symbolic representations of operations on data bits within a computer memory. These descriptions and representations are the means used by those skilled in the data processing arts to most effectively convey the substance of their work to others skilled in the art. A procedure, logic block, process, etc., is here, and generally, conceived to be a self-consistent sequence of steps or instructions leading to a desired result. The steps are those requiring physical manipulations of physical quantities. Usually, though not necessarily, these quantities take the form of electrical or magnetic signals capable of being stored, transferred, combined, compared, and otherwise manipulated in a computer system. It has proven convenient at times, principally for reasons of common usage, to refer to these signals as bits, bytes, values, elements, symbols, characters, terms, numbers, or the like.

[0014] It should be borne in mind, however, that all of these and similar terms are to be associated with the appropriate physical quantities and are merely convenient labels applied to these quantities. Unless specifically stated otherwise as apparent from the following discussions, it is appreciated that throughout the present claimed subject matter, discussions utilizing terms such as "setting," "storing," "scanning," "receiving," "sending," "disregarding," "entering," or the like, refer to the action and processes of a computer system or similar electronic computing device, that manipulates and transforms data represented as physical (electronic) quantities within the computer system's registers and memories into other data similarly represented as physical quantities within the computer system memories or registers or other such information storage, transmission or display devices.

[0015] A firewall is an application program or hardware device that filters information (e.g., network traffic) coming through a network connection (e.g., Internet connection) to a computer system. In one example, if incoming packets of information is flagged by the firewall, it is not allowed through.

[0016] In general, although a computer user values the protection that a firewall provides, he or she is usually not interested in blocking all network traffic. Consequently, a firewall is usually equipped with a set of firewall exceptions that a computer user can select to allow certain types of information to pass through.

[0017] With conventional operating systems, different firewall exceptions may be presented in a list format to a computer user. However, as operating systems become more complex and the number of network facing features increase, the list of firewall exceptions expanded as well. As the average computer user does not have a high level of expertise with regards to network protocols and computer security, the management of a long list of firewall exceptions evolved into a daunting task.

[0018] For a particular network-facing application feature or function, multiple firewall exceptions may need to be enabled for the application program to operate properly. Under conventional approaches, a computer user may be required to read through a long list of firewall exceptions to enable all the related firewall exceptions. Not only is this excessively time consuming, a mistake can cause another application program to malfunction and/or render the firewall worthless as a security tool.

[0019] In contrast to conventional approaches, embodiments effectively reduce the complexity involved in firewall exceptions management and present a more efficient way for a computer user to manage network traffic. In one example, a plurality of firewall exceptions is read. The plurality of firewall exceptions are then intelligently associated with one or more of a plurality of firewall exception groups, and each firewall exception group corresponds to a common user experience (e.g., a user experience associated with a particular application program).

[0020] By grouping exceptions based on a common user experience, embodiments create a simpler and more user-friendly firewall management experience. Also, the attack surface for a user is minimized through the use of previously invisible exception types. The exceptions, in one example, are not made visible because the exceptions may be too complex for an average user to understand and/or control. Furthermore, all dependencies for a particular experience are ensured to be enabled simultaneously.

[0021] Figure 1 illustrates block diagrams of a system 100 for grouping firewall exceptions in accordance with an embodiment of the present claimed subject matter. System 100 includes firewall exception 103, firewall exception 105, firewall exception 107, firewall exception 109, firewall exception 111, firewall exception 113, firewall exception 115, Group 117, Group 121, Group 123, Group 125, and Group 127.

[0022] Although system 100 is shown and described as having certain numbers and types of elements, the present claimed subject matter is not so limited; that is, system 100 may include elements other than those shown, and may include more than one of the elements that are shown. For example, system 100 can include a greater or fewer number of firewall exceptions than the seven (firewall exception 103, firewall exception 105, firewall exception 107, firewall exception 109, firewall exception 111, firewall exception 113, and firewall exception 115) shown.

[0023] In one embodiment, the plurality of firewall exceptions (e.g., firewall exceptions 103, 105, 107, 109, 111, and 113) is read. Subsequently, the plurality of firewall exceptions are then intelligently associated with one or more of a plurality of firewall exception groups (e.g., group 117, 121, and 123). Each firewall exception group corresponds to a user experience. Also, one or more firewall exceptions Groups (e.g., group 117 and Group 121) can be associated with a higher level group (e.g., Group 125). In one embodiment, Group 125 and Group 123 are associated with Group 127. In one example, Group 127 is a single line item displayed to a user.

[0024] Further, in one example, individual firewall exceptions 103, 105, and 107 may all relate to a media player network sharing service and are associated with Group 117. In another example, firewall exceptions 109 and 111 may both relate to a printing service and are associated with Group 121. In yet another example, firewall exceptions 113 and 115 may both related to an online meeting service and are associated with Group 123. As illustrated, embodiments allow a long list of individual firewall exceptions to be intelligently organized into experience-based groups.

[0025] Figures 2A and 2B illustrate block diagrams of an embodiment of the present claimed subject matter in operation. Figure 2A includes a firewall exceptions control panel 251, firewall exception 203, firewall exception 205, firewall exception 207, firewall exception 209, firewall exception 211, firewall exception 213, and firewall exception 215. In Figure 2A, a plurality of firewall exceptions (203, 205, 207, 209, 211, 213, and 215) are accessed. Also, each of the firewall exceptions is related to a user experience (e.g., an application feature). In Figure 2B, the plurality of firewall exceptions are collapsed into user experience groups, wherein each user experience group includes firewall exceptions that are related to a common experience.

[0026] In one example, firewall exceptions 203, 205, 207, and 209 are collapsed into firewall exception group 223. Further, firewall exceptions 211 and 213 are collapsed into firewall exception group 225. Moreover, firewall exception 215 is placed in firewall exception group 227.

[0027] Thus, embodiments provide computer users with a more efficient way of managing firewall exceptions. Embodiments automatically categorize and group individual firewall exceptions into application program feature groups. In one example, a computer user that wants to enable an application program feature can select a firewall exception group (e.g., firewall exception group 223) and all the firewall exceptions included in the group (e.g., firewall exceptions 203, 205, 207, and 209) would be automatically enabled.

[0028] Figure 3 illustrates a flowchart 300 of an experience-based method for grouping exceptions (e.g., firewall exception and/or virus scan exception) upon which embodiments in accordance with the present claimed subject matter can be implemented. Although specific steps are disclosed in flowchart 300, such steps are exemplary. That is, embodiments of the present claimed subject matter are well suited to performing various other or additional steps or variations of the steps recited in flowchart 300. It is appreciated that the steps in flowchart 300 can be performed in an order different than presented. At block 303, the process starts.

[0029] At block 305, a plurality of exceptions (e.g., firewall exception and/or virus scan exception) is read. In one embodiment, reading comprises accessing a user-defined list of exceptions (e.g., firewall exception and/or virus scan exception). In one embodiment, reading comprises accessing an application defined list of exceptions. In one embodiment, names of firewall exception groups are retrieved from an indirect reference (e.g., indirect string reference) from a resource library.

[0030] At block 307, one or more of the plurality of exceptions (e.g., firewall exception and/or virus scan exception) are intelligently associated with one of a plurality of firewall exception groups. In one embodiment, each firewall exception group corresponds to a common user experience. In one example, exceptions (e.g., firewall exception and/or virus scan exception) related to file sharing are associated with a file sharing firewall exception group. Also, in one embodiment, intelligently associating further comprises utilizing indirect string references in the plurality of exceptions (e.g., firewall exception and/or virus scan exception). Further, in one embodiment, a firewall exception group of the plurality of firewall exception groups is defined by a third party via a public application program interface.

[0031] At block 309, multiple tiers of grouping are formed. In one embodiment, the plurality of firewall exception groups is placed into higher level groups. In one example, a first level of exceptions (e.g., firewall exception and/or virus scan exception) may be grouped to form second level firewall exception groups. The second level firewall exception groups may in turn be grouped to form third level firewall exception groups. In one instance, a user may only see the third level exceptions groups but not see the second level firewall exception groups.

[0032] At block 311, an exception group manipulated (e.g., selected) automatically results in all exceptions corresponding to the selected exception group being manipulated (e.g., selected). Thus, in one example, a user unfamiliar with exceptions (e.g., firewall exception and/or virus scan exception) management can still easily pick firewall exception groups related to specific application program features or user experiences that he or she can recognize.

[0033] At block 313, a manipulation command (e.g., a user selection and/or a programmatic command) for a firewall exception group is received. In one embodiment, user selection of a first firewall exception group associated with a first user experience does not interfere with a second user experience associated with a second firewall exception group.

[0034] At block 315, one or more firewall exception dependencies associated with the firewall exception group are enabled. In one embodiment, exceptions (e.g., firewall exception and/or virus scan exception) that are not included in, but are related to the firewall exception group, may also be automatically enabled. However, embodiments are not limited to enablement of firewall exception groups. In fact, embodiments apply to general manipulation of firewall groups. In one example, disablement of one firewall exception group automatically results in disablement of other related firewall exception groups. At block 317, the process ends.

[0035] Figure 4 illustrates a flowchart 400 of a user experience-based method for collapsing exceptions upon which embodiments in accordance with the present claimed subject matter can be implemented. Although specific steps are disclosed in flowchart 400, such steps are exemplary. That is, embodiments of the present claimed subject matter are well suited to performing various other or additional steps or variations of the steps recited in flowchart 400. It is appreciated that the steps in flowchart 400 can be performed in an order different than presented. At block 403, the process starts.

[0036] At block 405, a plurality of firewall exceptions is accessed. In one embodiment, each of the plurality of firewall exceptions is related to a user experience (e.g., an application program feature). In one embodiment, a firewall exception of the plurality of exceptions controls transmission control protocol traffic. Also, in one embodiment, wherein a firewall exception of the plurality of exceptions controls user datagram protocol traffic.

[0037] At block 407, the plurality of firewall exceptions are collapsed into user experience groups (e.g., a firewall exception group related to network printing). In one embodiment, each user experience group of the user experience groups includes one or more firewall exceptions (e.g., individual firewall exceptions related to network printing) that are related to a common user experience. Also, in one embodiment, enablement of a group automatically enables all firewall exceptions included in the group.

[0038] At block 409, the user experience groups are collapsed into one or more higher level groups. In one embodiment, a user experience group of the user experience groups is defined by an individual software vendor (e.g., a third party software vendor).

[0039] At block 411, the user experience groups are displayed to an end-user. At block 413, the process ends. In one embodiment, specific exceptions corresponding to the user experience groups are hidden from view as to prevent user confusion.

[0040] Thus, embodiments set forth method for grouping complicated sets of firewall exceptions under a single user-friendly line-item. Also, embodiments allow efficient authoring of exception groups that include all firewall exceptions for feature dependencies. Furthermore, embodiments ensure that all dependencies for a particular experience are enabled simultaneously. Combining these advantages, embodiments present users with a significantly simpler firewall exception management experience.

[0041] With reference to Figure 5, an exemplary system for implementing the claimed subject matter includes a computing device, such as computing device 500. In its most basic configuration, computing device 500 typically includes at least one processing unit 502 and memory 504. Depending on the exact configuration and type of computing device, memory 504 may be volatile (such as RAM), non-volatile (such as ROM, flash memory, etc.) or some combination of the two. This most basic configuration is illustrated in Figure 5 by dashed line 506. Additionally, device 500 *may* also have additional features/functionality. For example, device

500 may also include additional storage (removable and/or non-removable) including, but not limited to, magnetic or optical disks or tape. Such additional storage is illustrated in Figure 5 by removable storage 508 and non-removable storage 510. Computer storage media includes volatile and nonvolatile, removable and non-removable media implemented in any method or technology for storage of information such as computer readable instructions, data structures, program modules or other data. Memory 504, removable storage 508 and non-removable storage 510 are all examples of computer storage media. Computer storage media includes, but is not limited to, RAM, ROM, EEPROM. flash memory or other memory technology, CD-ROM, digital versatile disks (DVD) or other optical storage, magnetic cassettes, magnetic tape, magnetic disk storage or other magnetic storage devices, or any other medium which can be used to store the desired information and which can accessed by device 500. Any such computer storage media.may be part of device 500.

[0042] Device 500 may also contain communications connection(s) 512 that allow the device to communicate with other devices. Communications connection(s) 512 is an example of communication media. Communication media typically embodies computer readable instructions, data structures, program modules or other data in a modulated data signal such as a carrier wave or other transport mechanism and includes any information delivery media. The term "modulated data signal" means a signal that has one or more of its characteristics set or changed in such a manner as to encode information in the signal. By way of example, and not limitation, communication media includes wired media such as a wired network or direct-wired connection, and wireless media such as acoustic, RF, infrared and other wireless media. The term computer readable media as used herein includes both storage media and communication media.

[0043] Device 500 may also have input device(s) 514 such as keyboard, mouse, pen, voice input device, touch input device, etc. Output device(s) 516 such as a display, speakers, printer, etc. may also be included. All these devices are well know in the art and need not be discussed at length here.

[0044] To summarize, embodiments allow a computer user to manage firewall exceptions in a simple and straightforward manner. Embodiments set forth methods of experience-based firewall exception grouping. In one example, a plurality of firewall exceptions is read. The plurality of firewall exceptions are then intelligently associated with one of a plurality of firewall exception groups, and each firewall exception group corresponds to a common user experience.

[0045] Different from traditional approaches that require a user to painstakingly go through a long list of individual firewall exceptions, embodiments automatically organizes firewall exceptions into groups. Each group corresponds to a common user experience or application program feature. Thus, when the computer user desires to allow a certain feature, he or she can simply select the firewall exception group corresponding to the feature, thereby avoiding having to select all the related individual firewall exceptions. Specifically, embodiments allow all dependencies for a particular experience to be enabled simultaneous as a firewall exception group is selected. Also, embodiments allow multiple tiers of hierarchical grouping. Thus, in one example, at the first level individual firewall exceptions are organized into first level groups, and these first level groups in turn, may be organized into second level groups. Moreover, embodiments are compatible with different ways of grouping firewall exceptions.

[0046] In the foregoing specification, embodiments have been described with reference to numerous specific details that may vary from implementation to implementation. Thus, the sole and exclusive indicator of what is, and is intended by the applicants to be the claimed subject matter is the set of claims that issue from this application, in the specific form in which such claims issue, including any subsequent correction. Hence, no limitation, element, property, feature, advantage or attribute that is not expressly recited in a claim should limit the scope of such claim in any way. The specification and drawings are, accordingly, to be regarded in an illustrative rather than a restrictive sense.

CLAIMS

What is claimed is:

1. A method for grouping exceptions, said method comprising:

reading a plurality of exceptions; and

intelligently associating one or more of said plurality of exceptions with one or more of a plurality of exception groups, wherein each exception group corresponds to a common user experience;

manipulating an exception group automatically results in all exceptions corresponding to that selected exception group being manipulated.

2. The method of Claim 1, further comprises forming multiple tiers of grouping, wherein said plurality of exception groups is placed into higher level groups.

3. The method of Claim 1, wherein said plurality of exceptions are firewall exceptions.

4. The method of Claim 1 further comprising:

receiving a manipulation command for an exception group of said plurality of exception groups; and

manipulating one or more exception dependencies corresponding to said user experience.

5. The method of Claim 1, wherein manipulation of a first exception group associated with a first user experience does not interfere with a second user experience associated with a second exception group.

6. The method of Claim 1, wherein said intelligent associating further comprises utilizing indirect string references in said plurality of exceptions.

7. The method of Claim 1, wherein an exception group of said plurality of exception groups is defined by a third-party application via an application program interface.

8. The method of Claim 1, wherein said reading comprises accessing a user-defined list of exceptions.

9. The method of Claim 1, wherein said reading comprises accessing an application defined list of exceptions.

10. The method of Claim 1, wherein said plurality of exceptions are not individually visible to an end-user.

11. The method of Claim 1, wherein names of exception groups are retrieved from an indirect string reference from a resource library.

12. A computer-readable medium having computer-executable instructions for performing the steps comprising:

accessing a plurality of firewall exceptions, wherein each of said plurality of firewall exceptions is related to a user experience; and

collapsing said plurality of firewall exceptions into user experience groups, wherein each user experience group of said user experience groups includes one or more firewall exceptions that are related to a common user experience.

13. The computer-readable medium of Claim 12, wherein enablement of a group automatically enables all firewall exceptions included in said group.

14. The computer-readable medium of Claim 12, further comprises collapsing said user experience groups into one or more higher level groups.

15. The computer-readable medium of Claim 12, wherein a user experience group of said user experience groups is defined by an individual software vendor.

16. The computer-readable medium of Claim 12, further comprises displaying said user experience groups to an end-user, wherein specific exceptions corresponding to said user experience groups are hidden from view.

17. The computer-readable medium of Claim 12, further comprises adding a new interface to the public component object model firewall application program interface.

18. The computer-readable medium of Claim 12, wherein a firewall exception of said plurality of firewall exceptions controls Transmission Control Protocol traffic.

19. The computer-readable medium of Claim 12, wherein a firewall exception of said plurality of firewall exceptions controls User Datagram Protocol traffic.

20. A experience-based firewall exceptions management interface, said interface comprising:

a plurality of feature-based firewall exception groups, wherein each feature based firewall exception group includes one or more firewall exceptions associated with a feature; and

a plurality of manipulation mechanisms corresponding to each of said feature based firewall exception groups, wherein selection of an manipulation mechanism manipulates more than one firewall exception dependencies associated with said feature.

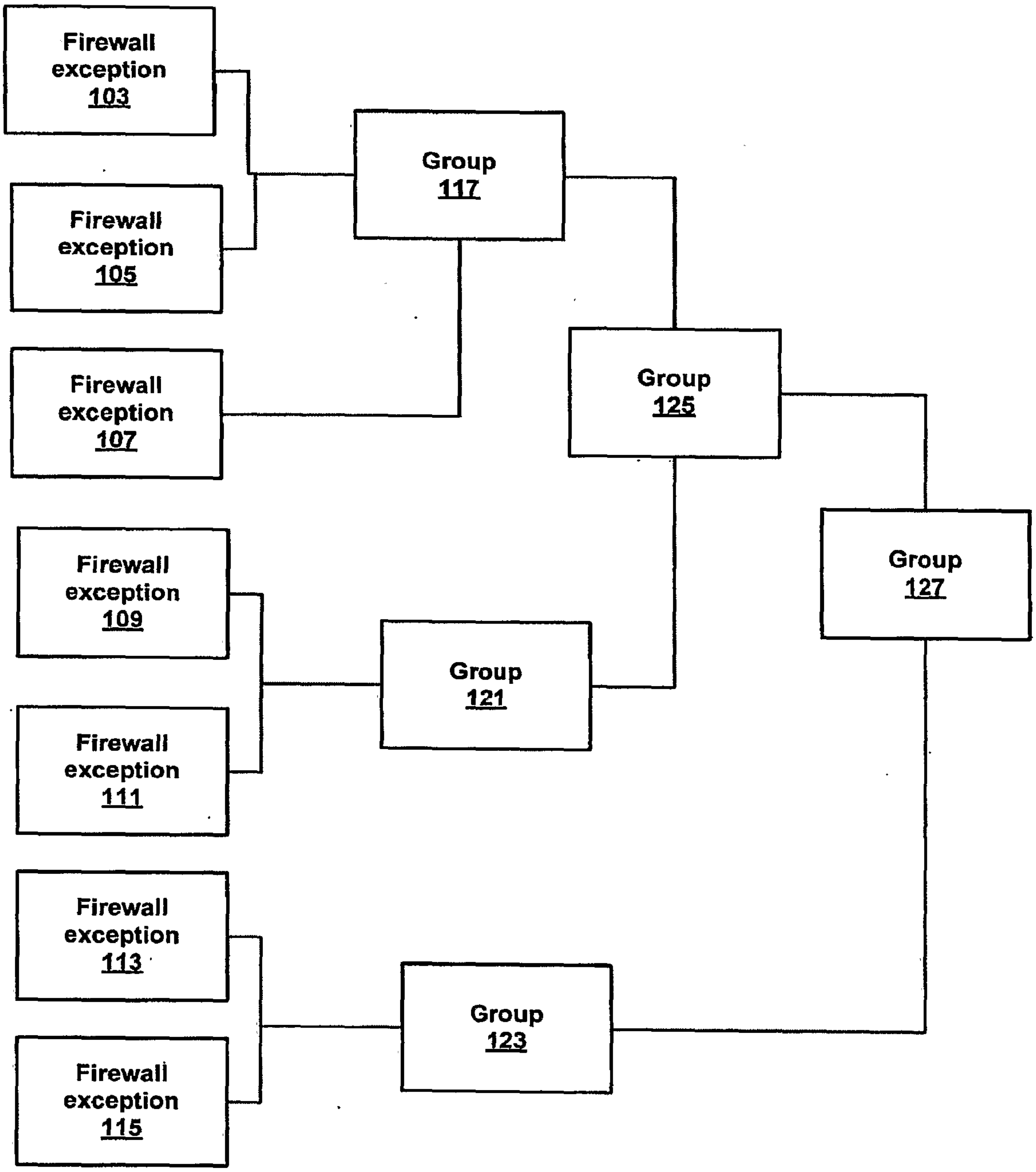

Firewall exception 103
Firewall exception 105
Firewall exception 107
Firewall exception 109
Firewall exception 111
Firewall exception 113
Firewall exception 115
Group 117
Group 125
Group 121
Group 127
Group 123

FIG. 1

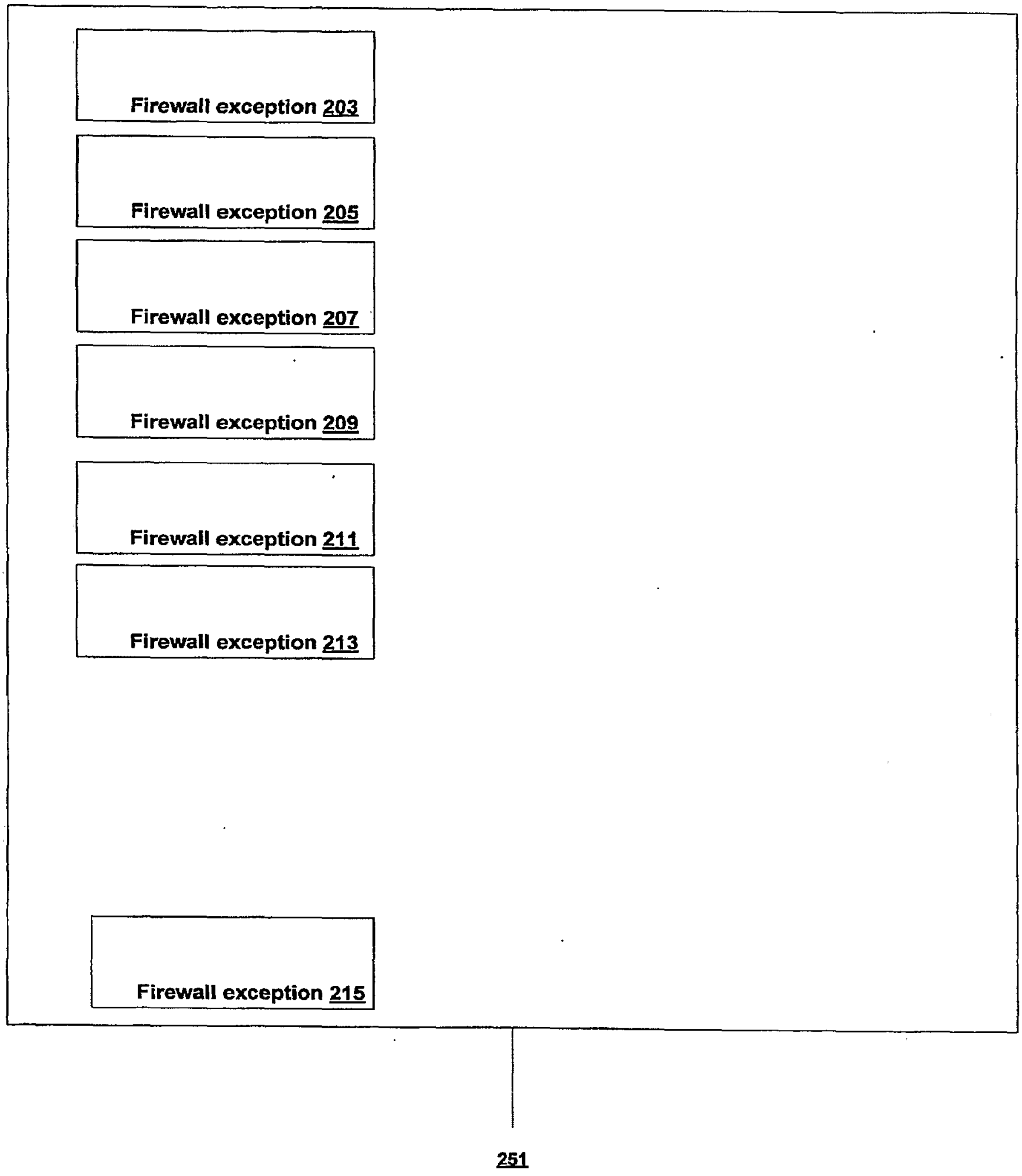

Firewall exception 203
Firewall exception 205
Firewall exception 207
Firewall exception 209
Firewall exception 211
Firewall exception 213
Firewall exception 215
251

FIG. 2A

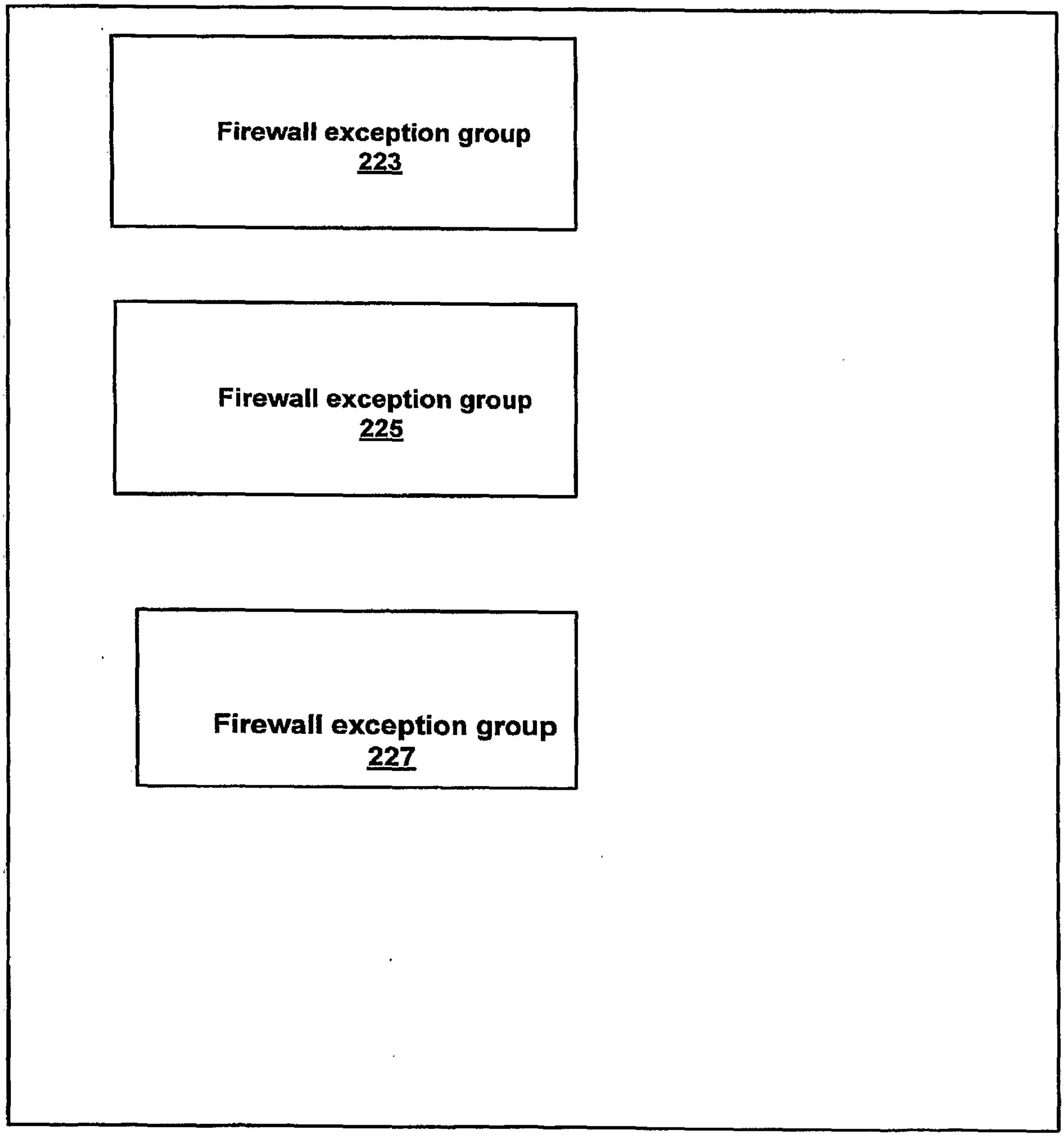

Firewall exception group
223
Firewall exception group
225
Firewall exception group
227
251

FIG. 2B

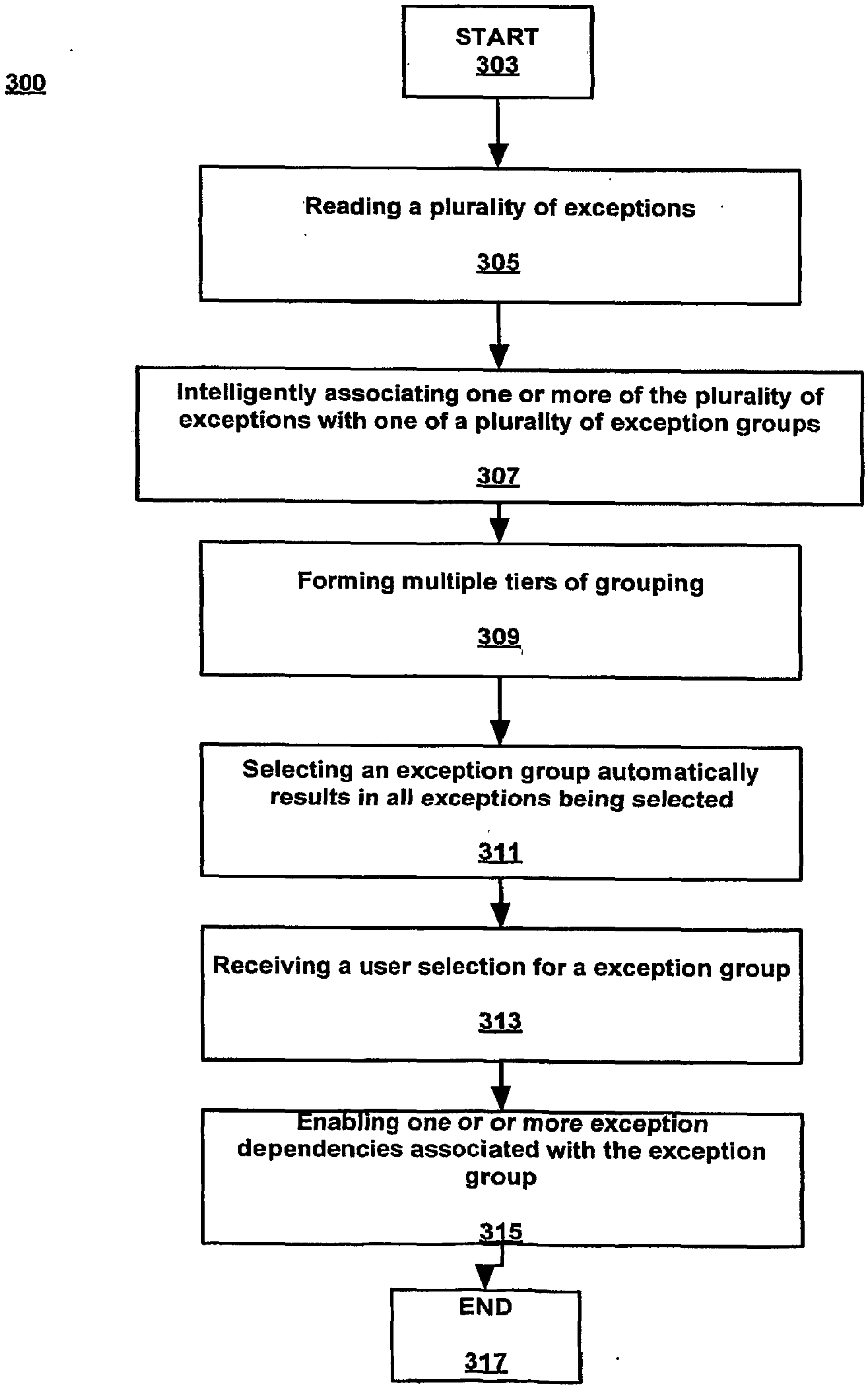

300
START
303
Reading a plurality of exceptions
305
Intelligently associating one or more of the plurality of exceptions with one of a plurality of exception groups
307
Forming multiple tiers of grouping
309
Selecting an exception group automatically results in all exceptions being selected
311
Receiving a user selection for a exception group
313
Enabling one or or more exception dependencies associated with the exception group
315
END
317

FIG. 3

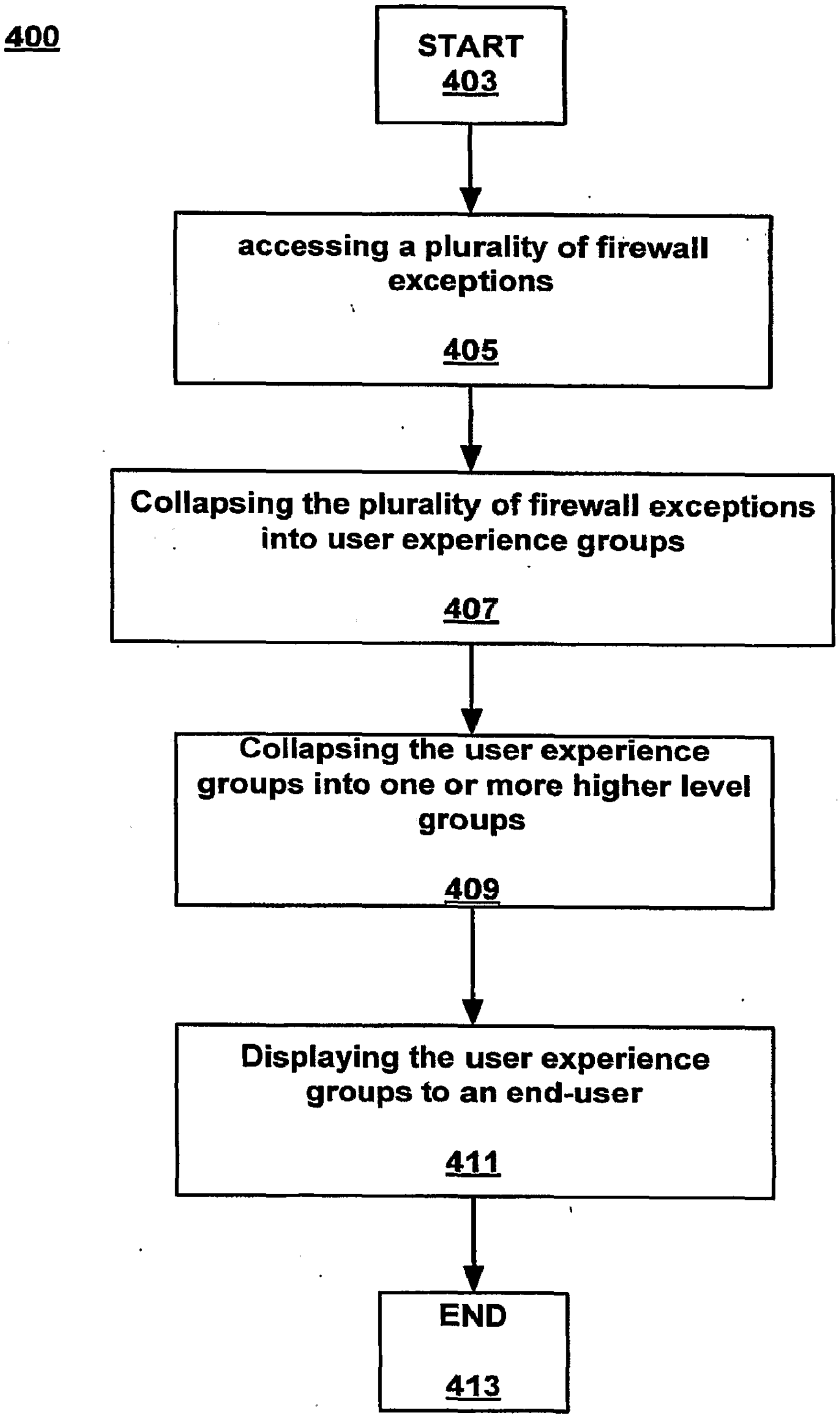

400
START
403
accessing a plurality of firewall exceptions
405
Collapsing the plurality of firewall exceptions into user experience groups
407
Collapsing the user experience groups into one or more higher level groups
409
Displaying the user experience groups to an end-user
411
END
413

FIG. 4

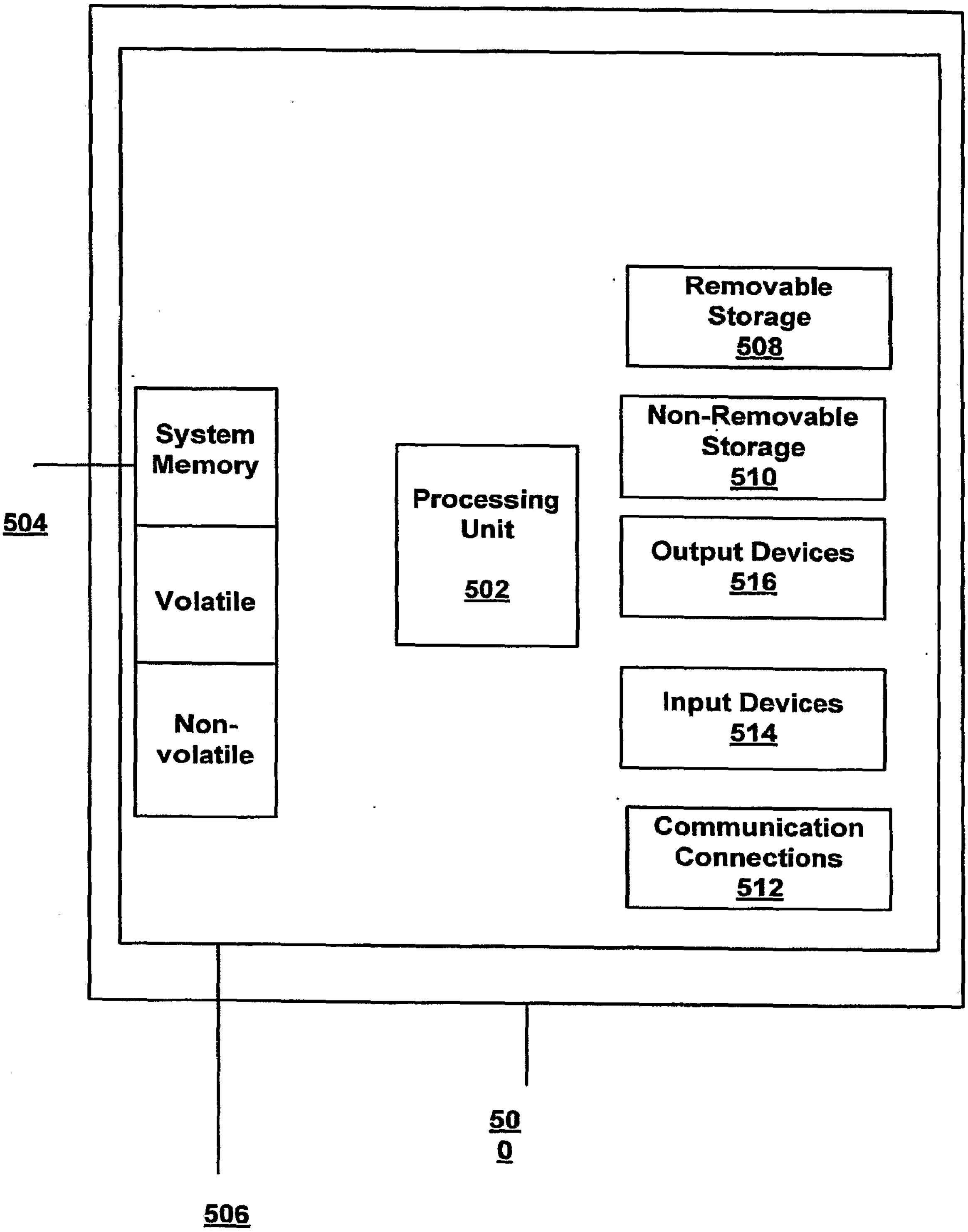

Removable Storage 508
Non-Removable Storage 510
System Memory
Processing Unit 502
Output Devices 516
504
Volatile
Input Devices 514
Non-volatile
Communication Connections 512
500
506

FIG. 5