(12) STANDARD PATENT (11) Application No. **AU 2011238676 B2**
(19) AUSTRALIAN PATENT OFFICE

(54) Title
Virtual application extension points

(51) International Patent Classification(s)
G06F 9/44 (2006.01)

(21) Application No: **2011238676** (22) Date of Filing: **2011.03.25**

(87) WIPO No: **WO11/126776**

(30) Priority Data

(31) Number	(32) Date	(33) Country
12/754,623	**2010.04.06**	**US**

(43) Publication Date: **2011.10.13**
(44) Accepted Journal Date: **2014.04.10**

(71) Applicant(s)
Microsoft Corporation

(72) Inventor(s)
Sheehan, John M.;Reierson, Kristofer H.

(74) Agent / Attorney
Davies Collison Cave, Level 15 1 Nicholson Street, MELBOURNE, VIC, 3000

(56) Related Art
US 7542988 B1 (COOK et al.) 02 June 2009

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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
13 October 2011 (13.10.2011)

PCT

(10) International Publication Number
WO 2011/126776 A3

(51) International Patent Classification:
G06F 9/44 (2006.01)

(21) International Application Number:
PCT/US2011/030053

(22) International Filing Date:
25 March 2011 (25.03.2011)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
12/754,623 6 April 2010 (06.04.2010) US

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

(72) Inventors: **SHEEHAN, John, M.**; c/o Microsoft Corporation, LCA - International Patents, One Microsoft Way, Redmond, WA 98052-6399 (US). **REIERSON, Kristofer, H.**; c/o Microsoft Corporation, LCA - International Patents, One Microsoft Way, Redmond, WA 98052-6399 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, 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, PE, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK,

[Continued on next page]

(54) Title: VIRTUAL APPLICATION EXTENSION POINTS

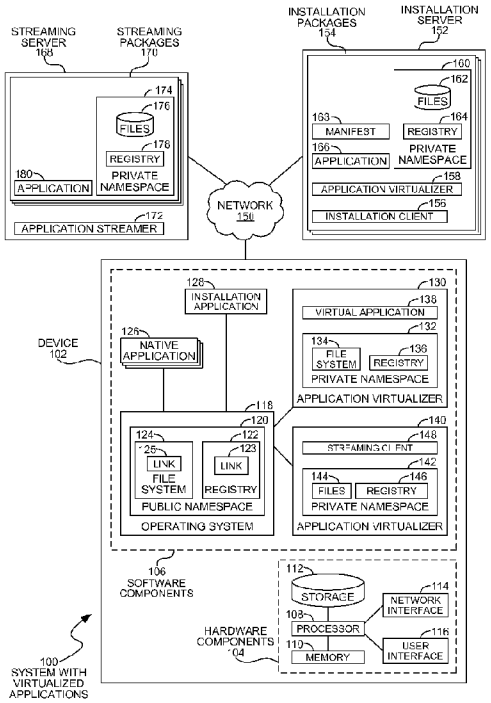

FIG. 1

(57) Abstract: A virtual application may be configured with several extension points within a host operating system. The virtual application may be configured with a private namespace in which various components, such as registry settings, dynamic linked libraries, and other components may reside. During configuration, links may be placed in the host operating system that may point to objects in the virtual application's private namespace so that the operating system and other applications may launch, control, or otherwise interact with the virtual application. The links may be located in a file system, registry, or other locations and may be available to other applications, including other virtual applications. A configuration routine may place the links into the host operating system at the time the application may be configured.

WO 2011/126776 A3

SM, 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:

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

— *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments (Rule 48.2(h))*

(88) Date of publication of the international search report:
2 February 2012

20112386676 10 Mar 2014

VIRTUAL APPLICATION EXTENSION POINTS

Background

[0001] Application virtualization is a technique that isolates an application from a host operating system or from other applications that may operate in the operating system. The application virtualization may have many benefits, such as allowing two or more otherwise incompatible applications to execute side by side on the same operating system. In many cases, configuring and managing a virtual application may be much simpler than installing a host application.

Summary

[0002] A virtual application may be configured with several extension points within a host operating system. The virtual application may be configured with a private namespace in which various components, such as registry settings, dynamic linked libraries, and other components may reside. During configuration, links may be placed in the host operating system that may point to objects in the virtual application's private namespace so that the operating system and other applications may launch, control, or otherwise interact with the virtual application. The links may be located in a file system, registry, or other locations and may be available to other applications, including other virtual applications. A configuration routing may place the links into the host operating system at the time the application may be configured.

[0002a] In one aspect there is provided a method performed by a computer processor for operating a virtual application on a device, said method comprising:

installing an application virtualizer at said device, said application virtualizer configured to execute said virtual application and to manage a private namespace storing one or more items related to the virtual application, said private namespace being private to said virtual application such that said stored items stored in said private namespace are generally accessible to said virtual application but other applications on said device are prevented from natively accessing said stored items stored in said private namespace;

formulating a link into said private namespace, said link providing an extension point for accessing a stored item, selected from among said one or more stored items, said link configured such that said other applications can interact with said virtual application in a specified manner by accessing said stored item;

making said link accessible to said other applications by storing said link in a public namespace of a host operating system at said device;

receiving a command comprising selection of said link at said application virtualizer, said link having been selected through said public namespace by an application from among said other applications; and

in response to receiving selection of said link:

identifying said stored item from said link; and

permitting said application to interact with said virtual application in said specified manner through access to said stored item.

[0002b] In another aspect there is provided a method comprising:

installing an application virtualizer at a device, said application virtualizer configured to execute a virtual application and to manage a private namespace storing one or more items related to the virtual application, said one or more stored items including a definition of a File Type Association, said File Type Association associating said virtual application with a specified file type, said private namespace being private to said virtual application such that said stored items stored in said private namespace are generally accessible to said virtual application but other applications on said device are prevented from natively accessing said stored items stored in said private namespace;

formulating a link into said private namespace, said link providing an extension point for accessing said File Type Association, said link defined in a manifest, said link permitting said other applications to launch said virtual application by accessing said File Type Association;

making said link accessible to said other applications by storing said link in a host operating system registry at said device, said host operating system registry located in a public namespace at said device;

receiving a request to open a file having said specified file type;

determining from said host operating system registry that said link is associated with said specified file type; and

following said first link to launch said virtual application for processing said file.

[0002c] In a further aspect there is provided a computer program product, the computer program product for implementing as method for operating a virtual application on a device, the computer program product comprising computer-executing instructions stored on one or more computer storage devices that, when executed by a processor, perform the method, including the following:

install an application virtualizer at said device, said application virtualizer configured to execute said virtual application and to manage a private namespace storing one or more items related to the virtual application, said private namespace being private to said virtual application such that said stored items stored in said private namespace are generally accessible to said virtual application but other applications on said device are prevented from natively accessing said stored items stored in said private namespace;

formulate a link into said private namespace, said link providing an extension point for accessing a stored item, selected from among said one or more stored items, said link configured such that said other applications can interact with said virtual application in a specified manner by accessing said stored item;

make said link accessible to said other applications by storing said link in a public namespace of a host operating system at said device;

receive a command comprising selection of said link at said application virtualizer, said link having been selected through said public namespace by an application from among said other applications;

in response to receiving selection of said link:

identify said stored item from said link; and

permit said application to interact with said virtual application in said specified manner through access to said stored item.

[0003] This Summary is provided to introduce a selection of concepts in a simplified form that are further described below in the Detailed Description. This Summary is not intended to identify key features or essential features of the claimed subject matter, nor is it intended to be used to limit the scope of the claimed subject matter.

Brief Description of the Drawings

[0004] In the drawings,

[0005] FIGURE 1 is a diagram illustration of an embodiment showing a system with virtualized applications.

[0006] FIGURE 2 is a flowchart illustration of an embodiment showing a method for installing a virtualized application.

[0007] FIGURE 3 is a flowchart illustration of an embodiment showing a method for responding to a link into a private namespace.

Detailed Description

[0008] A virtual application may be executed in a host operating system with pointers from the host operating system's public namespace into a private namespace used by the virtual application. The pointers may reside in the host operating system's registry and file system, for example, and may allow other applications and the operating system to access, control, and operate the virtual application.

[0009] The virtual application may be configured and made available by a installation program that may create links within the host operating system to a private namespace within an application virtualizer. The installation program may install the application virtualizer so that a user or another application may launch and interact with the virtual application.

[0010] In many embodiments, a user may experience the virtual application operating within application virtualizer as if the virtual application were executing natively in the host operating system. The virtual application may present a graphical user interface and may allow interaction in the same manner as a natively executing application, such as cut and paste functions and other similar graphical user interface elements.

[0011] The virtual application may operate within an application virtualizer and may have the private namespace and the host operating system namespace made available to the virtual application. Such a configuration may allow the virtual application to access the host operating system file system, registry, and other functions, while keeping many data sources for the virtual application within the private namespace.

[0012] Throughout this specification, like reference numbers signify the same elements throughout the description of the figures.

[0013] When elements are referred to as being "connected" or "coupled," the elements can be directly connected or coupled together or one or more intervening elements may also be present. In contrast, when elements are referred to as being "directly connected" or "directly coupled," there are no intervening elements present.

[0014] The subject matter may be embodied as devices, systems, methods, and/or computer program products. Accordingly, some or all of the subject matter may be embodied in hardware and/or in software (including firmware, resident software, micro-code, state machines, gate arrays, etc.) Furthermore, the subject matter may take the form of a computer program product on a computer-usable or computer-readable storage medium having computer-usable or computer-readable program code embodied in the medium for use by or in connection with an instruction execution system. In the context

of this document, a computer-usable or computer-readable medium may be any medium that can contain, store, communicate, propagate, or transport the program for use by or in connection with the instruction execution system, apparatus, or device.

[0015] The computer-usable or computer-readable medium may be for example, but not limited to, an electronic, magnetic, optical, electromagnetic, infrared, or semiconductor system, apparatus, device, or propagation medium. By way of example, and not limitation, computer-readable media may comprise computer storage media and communication media.

[0016] 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. 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 may be accessed by an instruction execution system. Note that the computer-usable or computer-readable medium can be paper or other suitable medium upon which the program is printed, as the program can be electronically captured via, for instance, optical scanning of the paper or other suitable medium, then compiled, interpreted, of otherwise processed in a suitable manner, if necessary, and then stored in a computer memory.

[0017] 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" can be defined as 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. Combinations of any of the above-mentioned should also be included within the scope of computer-readable media.

[0018] When the subject matter is embodied in the general context of computer-executable instructions, the embodiment may comprise program modules, executed by one or more systems, computers, or other devices. Generally, program modules include routines, programs, objects, components, data structures, and the like, that perform particular tasks

or implement particular abstract data types. Typically, the functionality of the program modules may be combined or distributed as desired in various embodiments.

[0019] Figure 1 is a diagram of an embodiment 100, showing a system with virtualized applications. Embodiment 100 is a simplified example of a system that may have virtualized applications in either a fully installed or streaming configuration.

[0020] The diagram of Figure 1 illustrates functional components of a system. In some cases, the component may be a hardware component, a software component, or a combination of hardware and software. Some of the components may be application level software, while other components may be operating system level components. In some cases, the connection of one component to another may be a close connection where two or more components are operating on a single hardware platform. In other cases, the connections may be made over network connections spanning long distances. Each embodiment may use different hardware, software, and interconnection architectures to achieve the described functions.

[0021] Embodiment 100 is an example of a system that may execute certain applications in a virtualized manner. The virtualized manner may isolate the application from other applications and from an operating system. An application virtualizer may provide the separation layer by creating a private namespace that may be referenced by the virtualized application. The application virtualizer may allow the application to reference the private namespace first, and then if a reference is not found in the private namespace, the application may be able to reference items in a public namespace.

[0022] The private namespace may include items such as files, libraries, assemblies, registry settings, and other variables that may be used by the virtualized application for many different functions. In some cases, an item in a private namespace may launch the virtual application, and sometimes such a launch item may be received with variables or other information that may be consumed by the virtual application as it executes. In other cases, an item in a private namespace may include a value that may be queried.

[0023] The application virtualization system of embodiment 100 may include links or references in a public namespace that may point to items in a private namespace. The links may be a mechanism for different applications and direct user input to access the virtual applications. In some cases, such links may be used by one virtual application to communicate with or call another virtual application.

[0024] One use for a link may be for file type associations. Some operating systems may allow certain file types to be associated with specific applications. When a file is 'opened'

using a file browser or other mechanism, a file type association entry in a registry or other location may be queried to determine which application, if any, may be associated with the file type. In the case of a virtualized application, a link in a private namespace may be created for associating the virtual application with the file type. In many operating systems, such a link may be placed in the operating system registry.

[0025] The links may be created during an installation process for the virtual application. The installation process may install an application virtualizer and other components that may be used to execute an application, as well as configure various links in a public namespace that reference items in the private namespace for the virtual application. The public links may be accessible to other applications, and may be intercepted and processed by an application virtualizer when the links are referenced.

[0026] The application virtualizer may receive a link to a location inside a private namespace and perform some action based on the link. In some cases, the application virtualizer may launch the application. Some cases may launch the application and pass parameters to the application that may have been received along with the link. In still other cases, the application virtualizer may return a value in the private namespace based on the request for the link.

[0027] The virtualized application may be a streamed application. In a streamed application, portions of the application may be downloaded from a streaming server on request as the application is being executed. A streaming client on a client device may request portions of the application from a streaming server as those portions are referenced by the application.

[0028] Embodiment 100 illustrates a device 102 that may represent a conventional computing device that contains hardware components 104 and software components 106. The device 102 may be any type of computing device, such as a desktop computer, server computer, game console, network appliance, or other device. In some cases, the device 102 may be a portable device, such as a laptop computer, netbook computer, personal digital assistant, mobile telephone, or other device.

[0029] The hardware components 104 may include a processor 108, random access memory 110, and nonvolatile storage 112. The hardware components 104 may also include a network interface 114 and a user interface 116.

[0030] The software components 106 may include an operating system 118 that may contain a public namespace 120. The public namespace 120 may include a registry 122 and a file system 124.

[0031] The public namespace 120 may include various items tracked by an operating system and made available to applications that execute within the operating system. In many embodiments, access to the public namespace 120 may be restricted in some cases. For example, an operating system may have access restrictions that may permit or deny access to certain portions of the public namespace 120 based on credentials presented by a user, device, application, or other entity that may attempt to access the items.

[0032] The public namespace 120 is called "public" because the public namespace 120 is managed by the operating system 120 and may be made available to different applications and users, subject to access restrictions. A virtual application may have private namespace that is "private" because the private namespace is available to the application but generally may not be searched or accessed by other applications.

[0033] Native applications 126 may be those applications that execute within the public namespace 120 of the operating system 118. Native applications may be conventional, non-virtualized applications.

[0034] An installation application 128 may be used to install and configure a virtual application on the device 102. The installation application 128 may perform several different functions so that a virtual application may successfully execute on the device 102, including creating a private namespace and providing links 123 and 125 from places in the public namespace 120 into the private namespace. The links 123 and 125 are illustrated as being inside the public namespace registry 122 and file system 124, respectively, and may be an entry point into the virtual application. An example of a process that may be performed by the installation application 128 may be found in embodiment 200 presented later in this specification.

[0035] An application virtualizer 130 may be installed by the installation application 128 may be a mechanism for executing an application in a virtual manner. The application virtualizer 130 may manage a private namespace 132 that may include a private file system 134 as well as private entries for a registry 136. The virtual application 138 may be executed by the application virtualizer 130 and may access the private namespace 132.

[0036] A streaming virtualized application may also be installed by the installation application 128. A streaming version may include an application virtualizer 140 that includes a private namespace 142 that contains a private file system 144 and private entries for a registry 146. Rather than a virtual application, a streaming client 148 may fetch portions of an application from a streaming server 168 as those portions are requested by the application or the application virtualizer 140. In some embodiments, the

streaming client 148 may fetch portions of the private namespace 142, including files in the private file system 144 and private entries for the registry 146.

[0037] The private namespaces 132 and 142 may be accessible by the virtual application 138 or a virtual application supplied by the streaming client 148 as the virtual application executes. The private namespaces 132 and 142 may also be accessible through the links 123 and 125 that may point into the private namespaces 132 and 142, which may enable other applications to directly access the virtual application, its settings, and other information.

[0038] The device 102 is illustrated as being connected to a network 150 to which an installation server 152 and a streaming server 168 may be connected.

[0039] The installation server 152 may contain several installation packages 154 that may be used by the device 102 to install different virtual applications. The installation packages 154 may contain an installation client 156, which may be the executable installation application 128 that operates on the client device 102 to configure a virtual application for execution.

[0040] The installation packages 154 may include all of the components that may be used to execute an application in a virtual manner. For example, the installation package 154 may include an application virtualizer 158. The application virtualizer 158 may be installed so that it may execute natively on a client device and provide resources for a virtual application.

[0041] The installation package 154 may include a private namespace 160 that may include files 162 and entries for a registry 164. The private namespace 160 may be fully populated in that the private namespace 160 may include all of the files that represent the virtual application 166. In some cases, the private namespace 160 in the installation package 154 may include a framework or sparsely populated set of files, registry settings, and other items, and the framework may be further populated by the application 166 or the application virtualizer 158.

[0042] The installation package 154 may include a manifest 163. The manifest 163 may include links that are placed within a public namespace and may point to the private namespace 160. The manifest 163 may include additional items that may be processed by an installation client 156 during installation.

[0043] The installation server 152 illustrates one example of how a virtual application may be prepared prior to installation and execution on a client device 102. The installation server 152 may contain many different packages 154 that may be downloaded and

installed on various client devices. In some embodiments, an installation package 154 may be stored on a Digital Versatile Disk (DVD) or other storage medium and read by a client device 102 to install the application.

[0044] The installation packages 154 may be created for full installations as well as streaming application. In a full installation, the installation package 154 may contain all of the files, including executable and data files, as well as any other resource to enable the virtual application to execute. A full installation may allow a client device 102 to fully execute the application without any additional resources, such as a streaming server 168 for example.

[0045] A streaming application may retrieve a majority of the executable and data files for a virtual application from a streaming server 168. A streaming application may be installed merely by installing a application virtualizer 158 and a streaming client. The streaming client may communicate with the streaming server 168 to retrieve portions of the application as requested. In some cases, a streaming application may be stored in a local cache, which may be re-used when the application is executed again.

[0046] In a streaming embodiment, each time the application is launched, a communication may occur between the client device 102 and the streaming server 168 to identify and retrieve the latest version of the application. The streaming server 168 may transmit a small portion of the application to the client device 102 so that the application may begin execution, and then transmit additional portions as those portions are requested by the application. A streaming client may monitor which portions are being requested and retrieve those portions from the streaming server 168.

[0047] The streaming server 168 may include several streaming packages 170, each of which may represent a different application. An application streamer 172 may communicate with a streaming client to download the portions of the streaming packages 170 that may be requested by a streaming client.

[0048] Each streaming package 170 may include a private namespace 174 that may include files 176 and settings for a registry 178. The files 176 may include all or a portion of the executable files for the virtualized application 180.

[0049] Figure 2 is a flowchart illustration of an embodiment 200 showing a method for installing a virtual application. Embodiment 200 is an example of a method that may be performed by an installation application that may execute natively on an operating system to install and configure components so that a virtual application may be executed.

Embodiment 200 is an example of a process that may be performed by an installation application, such as the installation application 128 or 156 of embodiment 100.

[0050] Other embodiments may use different sequencing, additional or fewer steps, and different nomenclature or terminology to accomplish similar functions. In some embodiments, various operations or set of operations may be performed in parallel with other operations, either in a synchronous or asynchronous manner. The steps selected here were chosen to illustrate some principles of operations in a simplified form.

[0051] Embodiment 200 illustrates one mechanism by which a virtual application may be configured to be operated on a client device. The method of embodiment 200 may install an executable program that may operate natively to isolate the application, and then populate a private namespace for that application. The method may also include links from a public namespace into locations in the private namespace.

[0052] An installation package may be received in block 202, and an installation application may begin executing in block 204. The operations of the installation application may be reflected in the remaining portion of embodiment 200.

[0053] The installation application may search for a previously installed application virtualizer in block 206. If an application virtualizer has not been installed in block 208, an application virtualizer is installed in block 210.

[0054] A new instance of a private namespace may be created in block 212. In the case of a previously installed application virtualizer, a new instance of a private namespace may allow one application virtualizer to execute two or more virtual applications, each with its own private namespace. In some embodiments, a second instance of an application virtualizer may be executing to support a second virtual application, while in other embodiments, a single instance of an application virtualizer may be capable of executing two or more virtualized applications separately yet simultaneously.

[0055] The links in the manifest may be processed in block 214. The manifest may be a listing, such as a text file or XML file, that contains items to be processed by an installation application. For each link in block 214, a link may be installed in a public namespace and may link to a location in the private namespace in block 216. The link may be a pointer, reference, location, or other identifier that may be accessed from the public namespace and allow access into the private namespace.

[0056] One use scenario may be to register the virtual application as an application that may be associated with a specific file type. Many operating systems may associate files with specific filename extensions to specific applications that may open and process the

files. For example, a registry setting may be defined that associates each file with a filename ending in ".docx" as being associated with a specific word processing application. When the word processing application is a virtual application, the link in the registry may point to the virtual application's private namespace and may cause the virtualized word processing application to execute.

[0057] Some links established in block 216 may cause the virtual application to be launched. Some such links may allow parameters or other information to be passed to the application so that the application may process the information. In some instances, the virtual application may return a value or other information in response to the call.

[0058] In some cases, the links established in block 216 may point to configuration information that may be stored in a file or setting in the private namespace. In such cases, the links may be traversed to examine the file or setting and to return a value or other information that may be stored in the location.

[0059] After installing each link in block 214, the application may be configured.

[0060] If the application is a streaming application in block 218, a streaming client may be installed in block 220 and configured in block 222. The streaming client may communicate with a streaming server to fetch portions of the application as the application requests the portion. The configuration in block 222 may include installing an address for the streaming server and configuring other parameters that may be used by a streaming client. After configuration in block 222, the installation may be complete in block 228.

[0061] If the application is not a streaming client in block 218, all of the items in the private namespace may be installed in block 224, along with the application in block 226. The operations of blocks 224 and 226 may represent a full installation of the virtual application. The application may be completely installed so that the application may execute on a device without accessing other devices. After installation in block 226, the installation may be complete in block 228.

[0062] Figure 3 is a flowchart illustration of an embodiment 300 showing a method for responding to a link into a private namespace. Embodiment 300 is a simplified example of some operations that may be performed by an application virtualizer when a link from a public namespace is followed into a private namespace.

[0063] Other embodiments may use different sequencing, additional or fewer steps, and different nomenclature or terminology to accomplish similar functions. In some embodiments, various operations or set of operations may be performed in parallel with

other operations, either in a synchronous or asynchronous manner. The steps selected here were chosen to illustrate some principles of operations in a simplified form.

[0064] Embodiment 300 is an example of some of the processes that may be performed by an application virtualizer. An application virtualizer may monitor references into the private namespace from links that may be available in a public namespace. The application virtualizer may intercept the links from the public namespace and enable connections into the private namespace.

[0065] In block 302, a link may be referenced in the public namespace. The link may be referenced by an application making a query to the registry to determine a value for a key, for example. In another example, a user may click on a shortcut in a start menu that references a link in the public namespace to an executable file in the private namespace.

[0066] In still another example, a user may use a command shell to connect the output of one application to a virtual application by a pipeline command. The virtual application may be referenced using a name in the public namespace that may link to another location in the private namespace.

[0067] A reference may be received by an application virtualizer in block 304 that may link into the private namespace. If the request for the link may be handled by reading the value in the private namespace in block 306 and not launching the application, the value may be retrieved in block 308 and returned to the requester in block 310.

[0068] If the request can be fulfilled by launching the virtual application in block 306, the application may be launched in the virtual environment in block 312. If a parameter, value, or other information was included in the reference received in block 304, the information may be passed to the application in block 314.

[0069] Some references may cause the application to launch and may receive a value. For example, a file type association may be used by an operating system to receive a selection from a user for a specific file type to open, then the operating system may look up the file type association in the public namespace. The file type association may link into the private namespace, at which time the application virtualizer may launch the application in block 312 and pass the filename to the application in block 314. The application may then open the file and begin operation.

[0070] In such an example, a user may be presented with a user interface generated by the virtual application, and the user may begin interacting with the application. In such an example, no response from the application may be expected in block 316 and the virtual application may execute with the values received.

[0071] In another example, a virtual application may be referenced in a command shell with a pipeline command. A pipeline command may direct the output of one application to the input of another application. Sometimes, two, three, or more applications may be pipelined together. In such an example, the virtual application may receive input from one application and respond with output that may be consumed by another application. In such an example, a response may be expected in block 316 and the application virtualizer may respond with the response value in block 320.

[0072] The foregoing description of the subject matter has been presented for purposes of illustration and description. It is not intended to be exhaustive or to limit the subject matter to the precise form disclosed, and other modifications and variations may be possible in light of the above teachings. The embodiment was chosen and described in order to best explain the principles of the invention and its practical application to thereby enable others skilled in the art to best utilize the invention in various embodiments and various modifications as are suited to the particular use contemplated. It is intended that the appended claims be construed to include other alternative embodiments except insofar as limited by the prior art.

[0073] Throughout this specification and the claims which follow, unless the context requires otherwise, the word "comprise", and variations such as "comprises" or "comprising", will be understood to imply the inclusion of a stated integer or step or group of integers or steps but not the exclusion of any other integer or step or group of integers or steps.

[0074] The reference in this specification to any prior publication (or information derived from it), or to any matter which is known, is not, and should not be taken as, an acknowledgement or admission or any form of suggestion that that prior publication (or information derived from it) or known matter forms part of the common general knowledge in the field of endeavour to which this specification relates.

2011238676 10 Mar 2014

CLAIMS

1. A method performed by a computer processor for operating a virtual application on a device, said method comprising:

installing an application virtualizer at said device, said application virtualizer configured to execute said virtual application and to manage a private namespace storing one or more items related to the virtual application, said private namespace being private to said virtual application such that said stored items stored in said private namespace are generally accessible to said virtual application but other applications on said device are prevented from natively accessing said stored items stored in said private namespace;

formulating a link into said private namespace, said link providing an extension point for accessing a stored item, selected from among said one or more stored items, said link configured such that said other applications can interact with said virtual application in a specified manner by accessing said stored item;

making said link accessible to said other applications by storing said link in a public namespace of a host operating system at said device;

receiving a command comprising selection of said link at said application virtualizer, said link having been selected through said public namespace by an application from among said other applications; and

in response to receiving selection of said link:

identifying said stored item from said link; and

permitting said application to interact with said virtual application in said specified manner through access to said stored item.

2. The method of claim 1 further comprising:

formulating a second link into said private namespace, said second link providing an extension point for accessing a second stored item, selected from among said one or more stored items, said second link configured such that said other applications can interact with said virtual application in a specified manner by accessing said second stored item;

making said second link accessible to said other applications by storing said second link in said public namespace;

receiving a second command comprising selection of said second link at said

application virtualizer, said second link having been selected through said public namespace by a second application from among said other applications;

in response to receiving selection of said second link:

identifying said second stored item from said second link; and

permitting said second application to interact with said virtual application in said second specified manner through access to said second stored item.

3. The method of claim 2, said second link being located in a host file system.

4. The method of claim 2, said second link being located in a host registry.

5. The method of any one of claims 1 to 4, said private namespace comprising executable code for said virtual application.

6. The method of claim 5, said private namespace comprising registry settings for said virtual application.

7. The method of any one of claims 1 to 6, said application virtualizer and said first link being comprised in an installation package.

8. The method of claim 7, said installation package comprising an installation program that:

reads a manifest comprising said first link; and

creates said first link in said host operating system.

9. The method of any one of claims 1 to 8, wherein receiving a command comprising selection of said link comprises receiving said command from a second virtual application executing within a second application virtualizer at the device.

10. The method of any one of claims 1 to 9, said virtual application being streamed into said application virtualizer.

11. A method comprising:

installing an application virtualizer at a device, said application virtualizer configured to execute a virtual application and to manage a private namespace storing one or more items related to the virtual application, said one or more stored items including a definition of a File Type Association, said File Type Association associating said virtual application with a specified file type, said private namespace being private to said virtual application such that said stored items stored in said private namespace are generally accessible to said virtual application but other applications on said device are prevented from natively accessing said stored items stored in said private namespace;

formulating a link into said private namespace, said link providing an extension point for accessing said File Type Association, said link defined in a manifest, said link permitting said other applications to launch said virtual application by accessing said File Type Association;

making said link accessible to said other applications by storing said link in a host operating system registry at said device, said host operating system registry located in a public namespace at said device;

receiving a request to open a file having said specified file type;

determining from said host operating system registry that said link is associated with said specified file type; and

following said first link to launch said virtual application for processing said file.

12. The method of claim 11 further comprising:

formulating a second link into said private namespace, said second link providing an extension point for accessing a second stored item, selected from among said one or more stored items, said second link configured such that said other applications can interact with said virtual application in a specified manner by accessing said second stored item;

making said second link accessible to said other applications by storing said second link in said public namespace;

receiving a second command comprising selection of said second link at said application virtualizer, said second link having been selected through said public namespace by a second application from among said other applications; and

in response to receiving selection of said second link:

identifying said second stored item from said second link; and

permitting said second application to interact with said virtual application in said second specified manner through access to said second stored item.

13. The method of claim 12, said second request being created by a second virtual application.

14. The method of claim 13, said virtual application being streamed into said application virtualizer.

15. A computer program product, the computer program product for implementing as method for operating a virtual application on a device, the computer program product comprising computer-executing instructions stored on one or more computer storage devices that, when executed by a processor, perform the method, including the following:

install an application virtualizer at said device, said application virtualizer configured to execute said virtual application and to manage a private namespace storing one or more items related to the virtual application, said private namespace being private to said virtual application such that said stored items stored in said private namespace are generally accessible to said virtual application but other applications on said device are prevented from natively accessing said stored items stored in said private namespace;

formulate a link into said private namespace, said link providing an extension point for accessing a stored item, selected from among said one or more stored items, said link configured such that said other applications can interact with said virtual application in a specified manner by accessing said stored item;

make said link accessible to said other applications by storing said link in a public namespace of a host operating system at said device;

receive a command comprising selection of said link at said application virtualizer, said link having been selected through said public namespace by an application from among said other applications;

in response to receiving selection of said link:

identify said stored item from said link; and

permit said application to interact with said virtual application in said specified manner through access to said stored item.

16. The computer program product of claim 15 further comprising computer-executable instructions that, when executed, perform the following:

formulate a second link into said private namespace, said second link providing an extension point for accessing a second stored item, selected from among said one or more stored items, said second link configured such that said other applications can interact with said virtual application in a specified manner by accessing said second stored item;

make said second link accessible to said other applications by storing said second link in said public namespace;

receive a second comment comprising selection of said second link at said application virtualizer, said second link having been selected through said public namespace by a second application from among said other applications;

in response to receiving selection of said second link:

identify said second stored item from said second link; and

permit said second application to interact with said virtual application in said second specified manner through access to said second stored item.

17. The computer program product of claim 16, wherein the application and the second application are the same application.

18. The computer program product of claim 15, wherein the application is a second virtualized application, the second virtualized application related to a second one or more stored items stored in a second private namespace, the second private namespace differing from the first private namespace.

19. The computer program product of claim 15, wherein computer-executable instructions that when executed perform formulating a link into said private namespace comprise computer-executable instructions that when executed perform formulating a link into said private namespace that permits one of: passing parameters to said virtual application and traversing configuration information for said virtual application.

20. The computer program product of claim 15, computer-executable instructions that when executed perform formulating a link into said private namespace comprise computer-executable instructions that when executed perform formulating a link from one of: a file

system in said public namespace or a registry in said public namespace into said private namespace.

21. A method performed by a computer processor for operating a virtual application on a device substantially as hereinbefore described with reference to the accompanying drawings.

22. A method substantially as hereinbefore described with reference to the accompanying drawings.

23. A computer program product substantially as hereinbefore described with reference to the accompanying drawings.

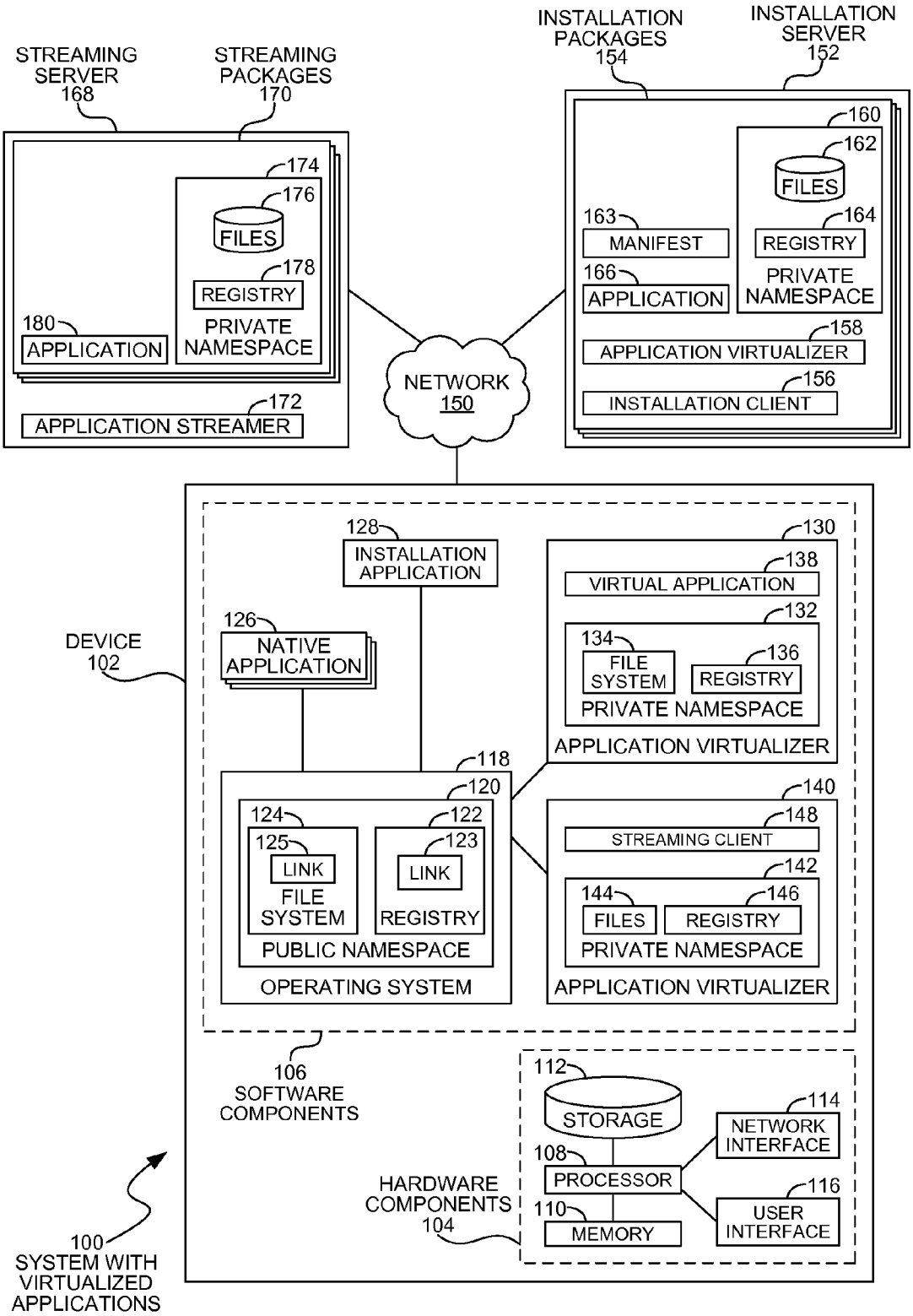
STREAMING SERVER 168
STREAMING PACKAGES 170
174
176
FILES
178
REGISTRY
180
APPLICATION
PRIVATE NAMESPACE
172
APPLICATION STREAMER
NETWORK 150
INSTALLATION PACKAGES 154
INSTALLATION SERVER 152
160
162
FILES
163
MANIFEST
164
REGISTRY
166
APPLICATION
PRIVATE NAMESPACE
158
APPLICATION VIRTUALIZER
156
INSTALLATION CLIENT
DEVICE 102
128
INSTALLATION APPLICATION
126
NATIVE APPLICATION
130
138
VIRTUAL APPLICATION
132
134
FILE SYSTEM
136
REGISTRY
PRIVATE NAMESPACE
APPLICATION VIRTUALIZER
118
120
124
125
LINK
FILE SYSTEM
122
123
LINK
REGISTRY
PUBLIC NAMESPACE
OPERATING SYSTEM
140
148
STREAMING CLIENT
142
144
FILES
146
REGISTRY
PRIVATE NAMESPACE
APPLICATION VIRTUALIZER
106
SOFTWARE COMPONENTS
112
STORAGE
114
NETWORK INTERFACE
108
PROCESSOR
116
USER INTERFACE
110
MEMORY
HARDWARE COMPONENTS 104
100
SYSTEM WITH VIRTUALIZED APPLICATIONS

FIG. 1

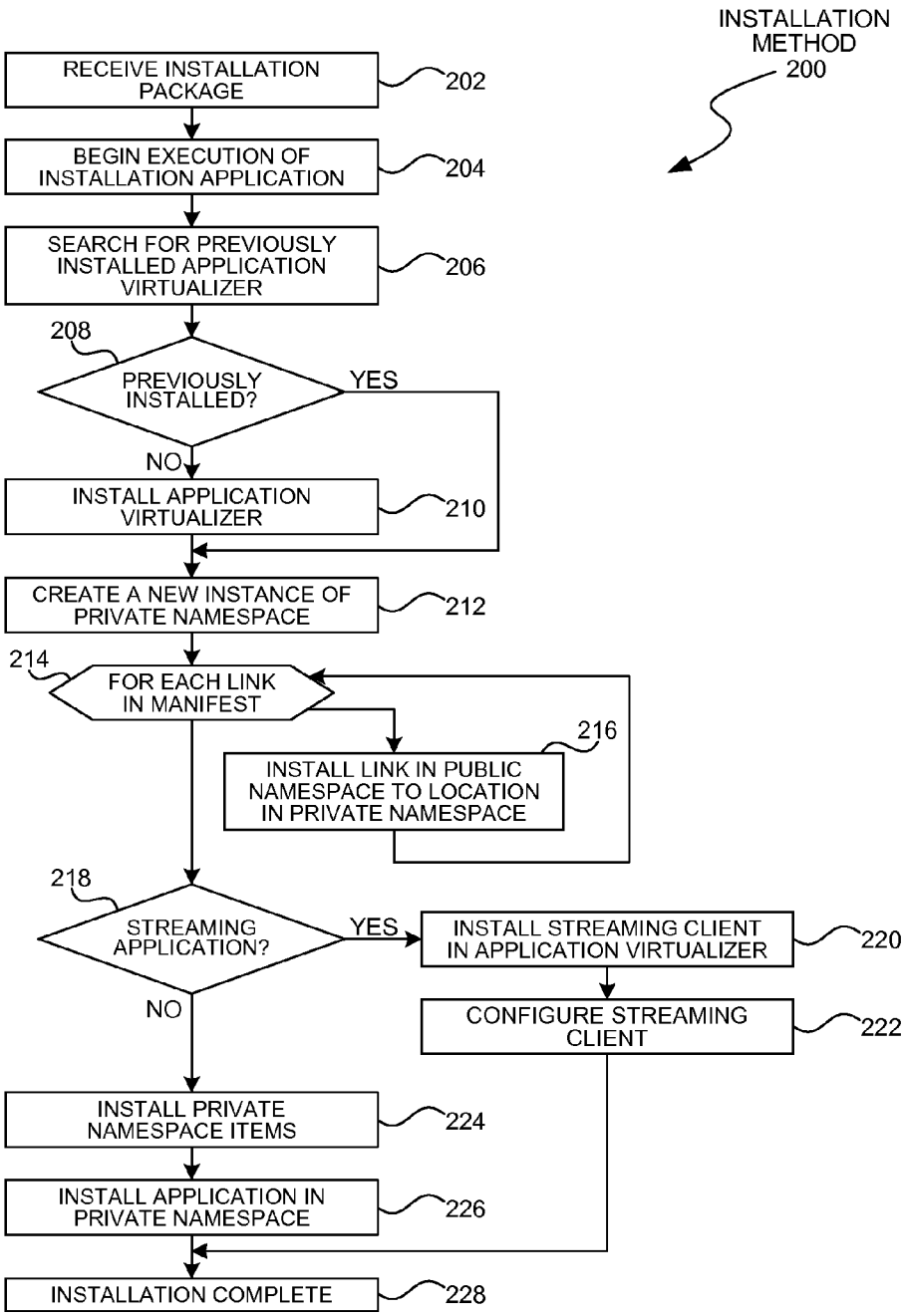
INSTALLATION METHOD
200
RECEIVE INSTALLATION PACKAGE
202
BEGIN EXECUTION OF INSTALLATION APPLICATION
204
SEARCH FOR PREVIOUSLY INSTALLED APPLICATION VIRTUALIZER
206
208
PREVIOUSLY INSTALLED?
YES
NO
INSTALL APPLICATION VIRTUALIZER
210
CREATE A NEW INSTANCE OF PRIVATE NAMESPACE
212
214
FOR EACH LINK IN MANIFEST
216
INSTALL LINK IN PUBLIC NAMESPACE TO LOCATION IN PRIVATE NAMESPACE
218
STREAMING APPLICATION?
YES
INSTALL STREAMING CLIENT IN APPLICATION VIRTUALIZER
220
CONFIGURE STREAMING CLIENT
222
NO
INSTALL PRIVATE NAMESPACE ITEMS
224
INSTALL APPLICATION IN PRIVATE NAMESPACE
226
INSTALLATION COMPLETE
228

FIG. 2

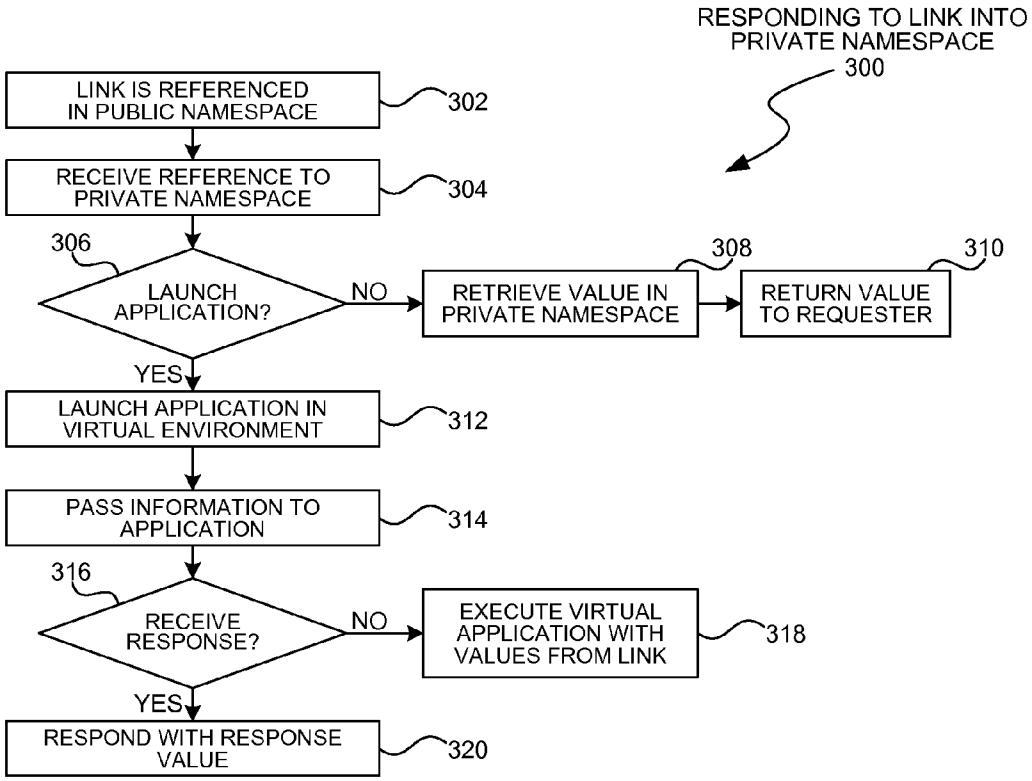
RESPONDING TO LINK INTO PRIVATE NAMESPACE
300
LINK IS REFERENCED IN PUBLIC NAMESPACE
302
RECEIVE REFERENCE TO PRIVATE NAMESPACE
304
306
LAUNCH APPLICATION?
NO
308
RETRIEVE VALUE IN PRIVATE NAMESPACE
310
RETURN VALUE TO REQUESTER
YES
LAUNCH APPLICATION IN VIRTUAL ENVIRONMENT
312
PASS INFORMATION TO APPLICATION
314
316
RECEIVE RESPONSE?
NO
EXECUTE VIRTUAL APPLICATION WITH VALUES FROM LINK
318
YES
RESPOND WITH RESPONSE VALUE
320

FIG. 3