

15915 U.S. PTO
102003

Please type a plus sign (+) inside this box +

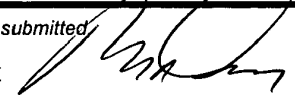
PTO/SB/16 (5-03)
Approved for use through 4/30/2003. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

PROVISIONAL APPLICATION FOR PATENT COVER SHEET
This is a request for filing a PROVISIONAL APPLICATION FOR PATENT under 37 CFR 1.53(c).

U.S. PTO
22264
60/512103
102003

INVENTOR(S)					
Given Name (first and middle [if any])		Family Name or Surname		Residence (City and either State or Foreign Country)	
Donn Dean Paul		Rochette Huffman O'Leary		Fenton, Iowa, U.S.A. Kanata, Ontario, Canada Kanata, Ontario, Canada	
<input type="checkbox"/> Additional inventors are being named on the _____ separately numbered sheets attached hereto					
TITLE OF THE INVENTION (280 characters max)					
SYSTEM FOR CONTAINERIZATION OF APPLICATION SETS					
Direct all correspondence to: CORRESPONDENCE ADDRESS					
<input checked="" type="checkbox"/> Customer Number		000293		Place Customer Number Bar Code Label here	
OR Type Customer Number here					
<input type="checkbox"/> Firm or Individual Name					
Address					
Address					
City		State		ZIP	
Country		Telephone		Fax	
ENCLOSED APPLICATION PARTS (check all that apply)					
<input checked="" type="checkbox"/> Specification		Number of Pages: 27		<input type="checkbox"/> CD(s), Number	
<input checked="" type="checkbox"/> Drawing(s)		Number of Sheets: 3		<input type="checkbox"/> Other (specify)	
<input type="checkbox"/> Application Data Sheet. See 37 CFR 1.76					
METHOD OF PAYMENT OF FILING FEES FOR THIS PROVISIONAL APPLICATION FOR PATENT (check one)					
<input checked="" type="checkbox"/> Applicant claims small entity status. See 37 CFR 1.27.		FILING FEE AMOUNT (\$)		80.00	
<input checked="" type="checkbox"/> A check or money order is enclosed to cover the filing fees		The Director is hereby authorized to charge filing fees or credit any overpayment to Deposit Account Number		80.00	
<input type="checkbox"/> Payment by credit card. Form PTO-2038 is attached.					
The invention was made by an agency of the United States Government or under a contract with an agency of the United States Government.					
<input checked="" type="checkbox"/> No.					
<input type="checkbox"/> Yes, the name of the U.S. Government agency and the Government contract number are: _____					

Respectfully submitted
SIGNATURE 
TYPED or PRINTED NAME **Ralph A. Dowell**
TELEPHONE (703) 415-2555

Date **10/20/03**
REGISTRATION NO. **26,868**
(if appropriate)
Docket Number: **14487 PRo**

USE ONLY FOR FILING A PROVISIONAL APPLICATION FOR PATENT

This collection of information is required by 37 CFR 1.51. The information is used by the public to file (and by the PTO to process) a provisional application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 8 hours to complete, including gathering, preparing, and submitting the complete provisional application to the PTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Mail Stop Provisional Application, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

Amazon Ex. 1014
IPR Petition - USP 7,519,814

Oracle Ex. 1014
Oracle v. VirtaMove, IPR2025-01001

SYSTEM FOR CONTAINERIZATION OF APPLICATION SETS

Field of the Invention

The invention relates to computer software. In particular, the invention relates to management and deployment
5 of server applications.

Background of the Invention

Computer systems are designed in such a way that application programs share common resources. It is traditionally the task of an operating system to provide a
10 mechanism to safely and effectively control access to shared resources. This is the foundation of multi-tasking systems that allow multiple disparate applications to co-exist on a single computer system.

The current state of the art creates a situation
15 where a collection of applications each designed for a distinct function must be separated with each application installed on an individual computer system. In some cases this is driven by conflict over shared resources, such as network port numbers. In other situations the separation is driven by the need to
20 securely separate data (files contained on disk-based storage) and/or applications between disparate users. In yet other situations, the separation is driven by the reality that certain applications require a specific version of operating system facilities and as such will not co-exist with
25 applications that require another version.

As a computer system architecture is applied to support specific services it inevitably requires that separate systems be deployed for each application set. This fact coupled with increased demand for support of additional
30 application sets results in a significant increase in the

number of computer systems being deployed. Such deployment makes it quite costly to manage the number of systems required to support several applications.

Summary of the Invention

5 In accordance with a first broad aspect, the invention provides a method of establishing a secure environment for executing, on a computer system, a plurality of applications that require shared resources. The method involves, for each group of a plurality of groups, associating
10 at least one respective application of the plurality of applications with the group. A respective resource allocation is associated with the group for the at least one respective application associated with the group to allow the at least one
15 respective application associated with the group to be executed on the computer system according to the respective resource allocation without conflict with the at least one respective application associated with other groups of the plurality of groups.

 In some embodiments of the invention, a group of
20 applications having one or more applications resides in a container. A container, and therefore the applications within the container, are provided with a secure environment from which to execute.

 In some embodiments of the invention, each group of
25 applications is provided with a secure storage medium.

 In some embodiments of the invention, for each group of the plurality of groups, the method involves executing on the computer system the at least one respective applications associated with the group according to the respective resource
30 allocation associated with the group.

In some embodiments of the invention, for each group of the plurality of groups, the method involves containerizing the at least one respective application associated with the group into a respective secure application container for the group, the respective secure application container being storable in a storage medium.

In some embodiments of the invention, for each group of the plurality of groups, application files for the at least one respective application associated with the group are containerized into the respective secure application container for the group.

In some embodiments of the invention, for each group of the plurality of groups, application system calls for the at least one respective application associated with the group are containerized into the respective container for the group.

In some embodiments of the invention, for each group of the plurality of groups, the at least one respective application of the plurality of applications are associated with the group according to classes of service.

In some embodiments of the invention, for each group of the plurality of groups, the at least one respective application of the plurality of applications are associated with the group according to classes of service comprising specific application services.

In some embodiments of the invention, the method involves exporting one or more of the respective secure application containers to a remote computer system.

In some embodiments of the invention, each respective secure application container has data files for the at least one application within the respective secure application

container. The method involves making the data files within one of the respective secure application containers inaccessible to the at least one respective application within another one of the respective secure application containers.

5 In some embodiments of the invention, for each respective secure application container, the method involves providing the at least one respective application within the respective secure application container with a respective root file system.

10 In some embodiments of the invention, for each group of the plurality of groups, the method involves associating a respective IP address for the group.

 In some embodiments of the invention, for each group of the plurality of groups, the method involves associating
15 resource limits for the at least one respective application associated with the group.

 In some embodiments of the invention, for each group of the plurality of groups, the method involves associating resource limits comprising any one or more of limits on memory,
20 CPU bandwidth, Disk size and bandwidth and Network bandwidth for the at least one respective application associated with the group.

 In some embodiments of the invention, each group of the plurality of groups forms a secure application container
25 having files related to the at least one respective application associated with the group. For each secure application container, the method involves determining whether resources available on the computer system can accommodate the respective resource allocation associated with the group comprising the
30 secure application container. If the computer system can

accommodate the respective resource allocation associated with the group forming the secure application container, the secure application container is exported to the computer system.

5 In some embodiments of the invention, if one or more secure application containers of the secure application containers are already installed on the computer system, the method involves determining whether the computer system has enough resources available to accommodate the one or more secure application containers which are already installed and
10 the secure application container being exported. If there are enough resources available, the resources are distributed between the one or more secure application containers and the secure application container being exported to provide resource control.

15 In some embodiments of the invention, in determining whether resources available on the computer system can accommodate the respective resource allocation, the method involves verifying whether the computer system supports any specific hardware required by the secure application container
20 to be exported.

In some embodiments of the invention, for each group of the plurality of groups, in associating a respective resource allocation with the group, the method involves associating resource limits for the at least one respective
25 application associated with the group. The method also involves during execution on the computer system: monitoring resource usage of the at least one respective application associated with the group; intercepting system calls, made by the at least one respective application associated with the
30 group, from user mode to kernel mode; comparing the monitored resource usage of the at least one respective application

associated with the group with the resource limits; and forwarding the system calls to a kernel on the basis of the comparison between the monitored resource usage and the resource limits.

5 In some embodiments of the invention, the method involves searching for application specific files associated with applications that are installed and working on computer systems.

10 In some embodiments of the invention, the method involves extracting the application specific files from the computer systems to create application sets ready to install in secure application containers on the computer system.

15 In some embodiments of the invention, the method involves converting the application specific files into an intermediate format that allows the applications that are installed and working on computer systems to be installed in secure application containers on the computer system.

20 In some embodiments of the invention, the method involves copying the application specific files, and related data and configuration information to a file storage medium.

 In some embodiments of the invention, the copying is achieved with the use of a user interface in which application programs are displayed and selected for copying to a storage medium.

25 In some embodiments of the invention, the application specific files, and related data and configuration information are made available for installation into secure application containers on a remote computer system.

In some embodiments of the invention, the method involves installing at least one of the plurality of applications in a root file system.

In accordance with a second broad aspect, the invention provides a method of establishing a secure environment for executing, on a computer system, a plurality of applications that require shared resources. The method involves, for each secure application container of a plurality of secure application containers, containerizing at least one respective application of the plurality of applications into the secure application container. A respective resource allocation is associated with the secure application container for the at least one respective application containerized within the secure application container to allow the at least one respective application containerized within the secure application container to be executed on the computer system according to the respective resource allocation without conflict with the at least one respective application containerized within other secure application containers of the plurality of secure application containers.

In accordance with a third broad aspect, the invention provides a computer usable medium having computer readable program code means embodied therein for establishing a secure environment for executing, on a computer system, a plurality of applications that require shared resources. The computer readable code means in the article of manufacture has, for each group of a plurality of groups, computer readable code means for associating at least one respective application of the plurality of applications with the group; and computer readable code means for associating a respective resource allocation with the group for the at least one respective application associated with the group to allow the at least one

respective application associated with the group to be executed on the computer system according to the respective resource allocation without conflict with the at least one respective application associated with other groups of the plurality of groups.

In accordance with a fourth broad aspect, the invention provides a memory for storing data for access by an application program being executed on a data processing system under a secure environment in which a plurality of applications require shared resources. The memory has a data structure stored in the memory. The data structure includes information resident in a database used by the application program and includes, for each group of a plurality of groups, a plurality of first data objects containing information associating at least one respective application of the plurality of applications with the group; and a plurality of second data objects associating a respective resource allocation with the group for the at least one respective application associated with the group to allow the at least one respective application associated with the group to be executed on the computer system according to the respective resource allocation without conflict with the at least one respective application associated with other groups of the plurality of groups.

In accordance with a fifth broad aspect, the invention provides a method for a program to interact with a user to establish a secure environment for executing, on a computer system, a plurality of applications that require shared resources. The method involves displaying the plurality of applications and, for each group of a plurality of groups, receiving a selection associating at least one respective application of the plurality of applications with the group to associate a respective resource allocation with the group for

the at least one respective application associated with the group and allow the at least one respective application associated with the group to be executed on the computer system according the respective resource allocation without conflict
5 with the at least one respective application associated with other groups of the plurality of groups.

In accordance with a sixth broad aspect, the invention provides a system adapted to perform any of the above method steps.

10 In another aspect, the invention relates to an apparatus to automate the process of extracting an application set installed on a computer system in such a way as to create an entity suitable for use as a secure software application container. The process involves parsing the files installed on
15 an existing system, extracting the application and system specific information that is associated with an application set and creating a set of files that can be used in a secure application set container.

Brief Description of the Drawings

20 Preferred embodiments of the invention will now be described with reference to the attached drawings in which:

Figure 1 is a schematic of computing platforms operable to interact with containers which are created by combining applications and system files, according to an
25 embodiment of the invention;

Figure 2 is a schematic of the computing platform of Figure 1 showing application sets being a collection of one or more software applications encompassed in a secure application container; and

Figure 3 is a flow diagram showing how application sets are created, according to another embodiment of the invention.

Detailed Description of the Preferred Embodiments

5 The following definitions are used herein:

- Computing platform: A computer system with a single instance of a fully functional operating system installed is referred to as a computing platform.
- 10 •Container: An aggregate of all files required to successfully execute a set of software applications on a computing platform is referred to as a container.
- 15 •Secure application container: An environment where each application set appears to have individual control of some critical system resources and/or where data within each application set is insulated from effects of other application sets is referred to as a secure application container.
- 20 •Consolidation. The ability to support multiple, possibly conflicting, sets of software applications on a single computing platform is referred to as consolidation.

Containers

Containers are created through the aggregation of application specific files.

25 A container contains application and data files for applications that provide a specific service. Examples of specific services include but are not limited to CRM (Customer Relation Management) tools, Accounting, and Inventory.

Referring to Figure 1, shown is a schematic of computing platforms operable to interact with containers that are created by combining applications and system files, according to an embodiment of the invention. A container in
5 Figure 1 combines un-installed application files on a storage medium or network, application files installed on a computer, and system files. The container is an aggregate of all files required to successfully execute a set of software applications on a computing platform.

10 In embodiments of the invention, containers are created by combining application files with none or more system files. The containers are output to a file storage medium and installed on the computing platforms from the file storage medium. Each container contains application and data files for
15 applications that together provide a specific service. The containers are created using, for example, Linux, Windows and Unix systems and preferably programmed in C and C++; however, the invention is not limited to these operating systems and programming languages and other operating systems and
20 programming language may be used.

An application set is a set of one or more software applications that support a specific service.

As shown in Figure 1, application files are combined with system files to create a composite (or container) of the
25 files required to support a fully functional software application.

Application specific files come from, for example, one of two sources:

- A distribution supplied by the creator of an application. These application specific files are often provided on a removable storage medium or downloaded in a packaged form.
- A computing platform in which the application specific files have been installed to support a fully functional software application and from which the application specific files are extracted.

The above two sources are used to categorize the applications as applications which have not yet been installed and applications which have already been installed and containers are created for both types of applications.

In some embodiments of the invention, system files are combined with application files as part of the aggregation process. These may include but are not limited to certain libraries, configuration files, and data sources.

Once a container is created it is placed in a file storage medium.

Secure Application Containers

In some cases, Applications from a container which contains application and data files for a specific service may need to share resources with Applications from a container which contains application and data files for another specific service. Some control over the shared resources may be required, for example, 1) for security purposes; 2) due to possible conflict arising from applications from different containers requiring a shared system resource, for example, the same port number; or 3) possible lack of resources on a system to accommodate all of the applications from different containers.

An environment where each application set appears to have individual control of critical system resources and/or where data from each application set is protected from effects of other application sets is referred to as a secure
5 application container.

A secure application container is created on computing platforms for software applications that have been containerized. A secure application container creates an environment where application sets have unique access to
10 critical system resources. Aspects of applications that may cause contention over shared resources are isolated in the secure application container.

In some embodiments of the invention, the secure application container is exported on a single instance of an
15 operating system. This is different than prior art systems in which multiple, potentially conflicting application sets, are hosted on a single computing platform but with several operating systems. Current solutions (commonly called Virtual Machine (VM) technology) enable each application set to run
20 within its own copy of an operating system. The net effect being, there are multiple instances of an operating system each running above a VM all on a single computing platform. For example, if eight application sets were hosted on a single computing platform using VM technology there would be eight
25 operating systems sitting above eight VMs on the computing platform. The VM approach does not allow an operating system to utilize H/W resources directly, thus changing in significant ways the behavior of a system. In embodiments of the invention, the use of containerization allows an operating
30 system to execute in it's native form on a computing platform.

For security purposes, data files within a secure application container are made inaccessible to applications from other containers. In some embodiments of the invention, this is achieved by providing the applications within secure application container with respective root file systems wherein the applications within a secure application container have a different root file system than those of applications within other secure application containers. Furthermore, data files and application files are made accessible only to users which are given permission to access applications within the secure application container. There are other security features such as: 1) providing each container with its own unique IP address that is independent of an IP address assigned to an underlying operating system on which the container is installed; 2) providing each container with a unique copy of a TCP/IP protocol stack for use by the container. This is unique copy of the protocol stack is used by the underlying operating system and application sets are not required to share resources associated the unique copy of the protocol stack with other application sets in other containers; and 3) provisioning each container with limits on consumption of hardware resources. In other words, each container is limited to an amount of CPU time, memory consumption, network bandwidth, disk storage and disk bandwidth. These limitations prevent an application in a container from doing harm to any other container or to the computing platform itself.

As discussed above, in prior art systems there may be conflicts between different applications in that the applications require the use of shared system resources. There are a number of potential resource conflicts within a computing platform. These could include, but are not limited to, conflicting uses of a common configuration file, conflicts over

firewall settings, any type of hardware device manipulation. By way of example, resource conflicts related to the use of network port numbers are described in some detail below as this is thought to be an easy to understand instance of a conflict
5 situation over shared system resources.

An example of resource contention involves two applications on the same computing platform attempting to use the same port number. In such a case a conflict will result and one application will fail. Prior art systems allow for
10 more than one IP (Internet Protocol) address to be used, through what is called aliasing, in conjunction with a single computing platform. This allows a computing platform to effectively provide more than one port for each service. For example, there can be two ports numbered 80 for web services on
15 a single computing platform as each port 80 could be bound to a unique IP address. However, such services do not guarantee that, for example, two applications running on a computing platform will always be assigned different IP addresses. As such, although there may be more than one port available to
20 provide a service, the two applications may still access the same IP address and a collision may still occur. To avoid collisions between applications from different containers, applications in different containers are given different IP (Internet Protocol) addresses. By associating respective IP
25 addresses with the secure application containers, applications within a container are always assigned the same IP address. In this way, for a particular port number, an application is always assigned the same port.

Computing platforms have limited resources to
30 accommodate different applications. For example, computing platforms have limits on their memory, CPU bandwidth, Disk size and bandwidth and Network bandwidth. In some embodiment of the

invention, provide is a capability to ensure that each container is able to use the resources that have been provisioned for it. In this manner each container is guaranteed to have sufficient resources to do work, but is not
5 allowed to consume more than its allocation at the expense of resource guarantees for the other containers.

In some embodiments of the invention, workload management capabilities are provided that enable management of software applications by removing resource and file contention normally associated with combining multiple application sets on
10 a single computing platform. Each secure application container contains information on resource requirements for applications within the container. Before a secure application container for a specific service is exported to a computing platform, the
15 resources available on the platform are verified to determine whether the computing platform can accommodate applications within the secure application container. If one or more secure application containers are already installed on the computing platform, a check is performed to determine whether the
20 computing platform has enough resources available to accommodate the secure application containers already installed and the secure application containers being exported. If there are enough resources available to accommodate the secure application container to be exported, it is exported and the
25 resources are distributed between the secure application containers. When an application is executing, its resource consumption is then monitored and controlled to assure limits on consumption are not exceeded.

As discussed in United States Provisional Application
30 entitled "USER MODE CRITICAL SYSTEM ELEMENTS AS SHARED LIBRARIES", which is incorporated herein by reference, the applications in the secure application containers execute in

user mode and a kernel portion of an operating system of a computing platform upon which the applications are installed executes in kernel mode. In conventional systems, there is a physical separation enforced by hardware between user mode and kernel mode and applications cannot run in kernel mode. System calls are used by the applications to make a transition from user mode to kernel mode. In some embodiments of the invention, resource usage of the applications within the secure application containers is monitored and system calls made by the applications are intercepted. When a system call made by an application executing in user mode is intercepted, the monitored resource usage for the application is compared to resource limits allocated to the application. If, for example, the system call requires the kernel to execute and cause the application to exceed the allocated resource limits the system call is prevented from being forwarded to the kernel, causing the application to pend, until the kernel can execute in kernel mode without the resource limits allocated to the application being exceeded. In this way control over shared resources is performed not only at an operating system level but also at a software level allowing further control over the shared resources.

The secure application containers assure separation between containers by providing each container with exclusive access to previously shared resources, such as, for example, a TCP/IP protocol stack, IP addresses and a root file system.

In some embodiments of the invention, multiple secure application containers exist above a single instance of an operating system and application sets are able to coexist inside secure application containers on the same computing platform.

With the use of the secure application containers potentially conflicting sets of applications are safely hosted on a single computing platform with a single instance of an operating system.

5 Container Extraction Tool

An extraction tool is used to discover the application specific files associated with given applications that are installed and working on a computing platform. The application specific files are extracted and used to create application sets as files on a file storage medium ready to install in secure application containers on computing platforms.

Software applications installed on computing platforms are extracted and converted into an intermediate format that allows the software applications to be installed in secure application containers on a remote compute platform. The process of extracting a software application installed on a fully functional computing platform is automated using a tool that executes on client systems which include, but are not limited to, Linux, Windows and Unix.

A database is used to track application specific files.

All application programs, application data and configuration information on a compute platform that are associated with specific installed software applications are identified.

The programs, data and configuration information are copied from the computing platform from which they originate and converted into an intermediate file format and stored on a

file storage medium. In one embodiment, a network storage device is used as the file storage medium. In some embodiments of the invention, the process of copying the programs, data and configuration information is achieved with the use of a user interface in which the application programs are displayed and selected for copying to a file storage medium using, for example, a drag and drop operation with a mouse, as described in United States Provisional Application entitled "Drag & Drop Application Management" which is incorporated herein by reference.

The programs, data and configuration information are then made available for installation into secure application containers on a remote computing platform.

In one embodiment of the invention, the container extraction tool is programmed in C++; however, the invention is not limited to the C++ programming language and in other embodiments of the invention other programming languages are used.

In some embodiments of the invention, containers are created from applications, at least one of which is an un-installed application. The un-installed applications is installed in a root file system.

Numerous modifications and variations of the present invention are possible in light of the above teachings. For example, embodiments of the invention have been described for implementation on computing platforms; however, the invention is not limited to implementations of computing platforms and in other embodiments of the invention there are implementations on computer systems. It is therefore to be understood that within the scope of the appended claims, the invention may be practiced otherwise than as specifically described herein.

WE CLAIM:

1. A method of establishing a secure environment for executing, on a computer system, a plurality of applications that require shared resources, the method comprising:

5 for each group of a plurality of groups:

associating at least one respective application of the plurality of applications with the group; and

10 associating a respective resource allocation with the group for the at least one respective application associated with the group to allow the at least one respective application associated with the group to be executed on the computer system according to the respective resource allocation without conflict with the at least one respective application associated with other groups of the plurality of groups.

15 2. A method of executing the applications of claim 1 comprising, for each group of the plurality of groups executing on the computer system the at least one respective application associated with the group according to the respective resource allocation associated with the group.

20 3. A method according to claim 1 or 2 comprising, for each group of the plurality of groups, containerizing the at least one respective application associated with the group into a respective secure application container for the group, the respective secure application container being storable in a
25 storage medium.

4. A method according to claim 3 wherein, for each group of the plurality of groups, the containerizing the at least one respective application associated with the group into a respective secure application container for the group comprises

containerizing application files for the at least one
respective application associated with the group into the
respective secure application container for the group.

5. A method according to claim 3 or 4 wherein, for each
5 group of the plurality of groups, the containerizing the at
least one respective application associated with the group into
a respective secure application container for the group
comprises containerizing application system calls for the at
least one respective application associated with the group into
10 the respective container for the group.

6. A method according to anyone of claims 1 to 5
wherein, for each group of the plurality of groups, the
associating at least one respective application of the
plurality of applications with the group comprises associating
15 the at least one respective application of the plurality of
applications with the group according to classes of service.

7. A method according to claim 6 wherein, for each group
of the plurality of groups, the associating at least one
respective application of the plurality of applications with
20 the group comprises associating the at least one respective
application of the plurality of applications with the group
according to classes of service comprising specific application
services.

8. A method according to claim 3 comprising exporting
25 one or more of the respective secure application containers to
a remote computer system.

9. A method according to claim 4 wherein each respective
secure application container comprises data files for the at
least one application within the respective secure application
30 container, the method comprising making the data files within

one of the respective secure application containers inaccessible to the at least one respective application within another one of the respective secure application containers.

10. A method according to claim 9 wherein the making the
5 data files within one of the respective secure application containers inaccessible to the at least one respective application within another one of the respective secure application containers comprises, for each respective secure application container providing the at least one respective
10 application within the respective secure application container with a respective root file system.

11. A method according to claim 1 or 2 wherein, for each group of the plurality of groups, the associating a respective resource allocation with the group comprises associating a
15 respective IP address for the group.

12. A method according to claim 1 or 2 wherein, for each group of the plurality of groups, the associating a respective resource allocation with the group comprises associating resource limits for the at least one respective application
20 associated with the group.

13. A method according to claim 1 or 2 wherein, for each group of the plurality of groups, the associating a respective resource allocation with the group comprises associating resource limits comprising any one or more of limits on memory,
25 CPU bandwidth, Disk size and bandwidth and Network bandwidth for the at least one respective application associated with the group.

14. A method according to claim 1 or 2 wherein each group of the plurality of groups comprises a secure application
30 container comprising files related to the at least one

• 50357-5

respective application associated with the group and, for each secure application container, the method further comprising:

determining whether resources available on the computer system can accommodate the respective resource allocation associated with the group comprising the secure application container; and

if the computer system can accommodate the respective resource allocation associated with the group comprising the secure application container, exporting the secure application container to the computer system.

15. A method according to claim 14 further comprising:

if one or more secure application containers of the secure application containers are already installed on the computer system, determining whether the computer system has enough resources available to accommodate the one or more secure application containers which are already installed and the secure application container being exported; and

if there are enough resources available, distributing the resources between the one or more secure application containers and the secure application container being exported to provide resource control.

16. A method according to claim 15 wherein the determining whether resources available on the computer system can accommodate the respective resource allocation comprises verifying whether the computer system supports any specific hardware required by the secure application container to be exported.

17. A method according to claim 2 wherein, for each group of the plurality of groups, the associating a respective

• 50357-5

resource allocation with the group comprises associating resource limits for the at least one respective application associated with the group, the method further comprising during execution on the computer system:

5 monitoring resource usage of the at least one respective application associated with the group;

 intercepting system calls, made by the at least one respective application associated with the group, from user mode to kernel mode;

10 comparing the monitored resource usage of the at least one respective application associated with the group with the resource limits; and

 forwarding the system calls to a kernel on the basis of the comparison between the monitored resource usage and the
15 resource limits.

18. A method according to claim 1 further comprising searching for application specific files associated with applications that are installed and working on computer systems.

20 19. A method according to claim 18 further comprising extracting the application specific files from the computer systems to create application sets ready to install in secure application containers on the computer system.

20. A method according to claim 19 further comprising
25 converting the application specific files into an intermediate format that allows the applications that are installed and working on computer systems to be installed in secure application containers on the computer system.

21. A method according to claim 20 further comprising copying the application specific files, and related data and configuration information to a file storage medium.

22. A method according to claim 21 wherein the copying is
5 achieved with the use of a user interface in which application programs are displayed and selected for copying to a storage medium.

23. A method according to claim 21 or 22 wherein the
10 application specific files, and related data and configuration information are made available for installation into secure application containers on a remote computer system.

24. A method according to claim 1 further comprising installing at least one of the plurality of applications in a root file system.

15 25. A method of establishing a secure environment for executing, on a computer system, a plurality of applications that require shared resources, the method comprising:

for each secure application container of a plurality of secure application containers:

20 containerizing at least one respective application of the plurality of applications into the secure application container; and

associating a respective resource allocation with the secure application container for the at least one respective
25 application containerized within the secure application container to allow the at least one respective application containerized within the secure application container to be executed on the computer system according to the respective resource allocation without conflict with the at least one

respective application containerized within other secure application containers of the plurality of secure application containers.

26. An article of manufacture comprising:

5 a computer usable medium having computer readable program code means embodied therein for establishing a secure environment for executing, on a computer system, a plurality of applications that require shared resources, the computer readable code means in said article of manufacture comprising:

10 for each group of a plurality of groups:

computer readable code means for associating at least one respective application of the plurality of applications with the group; and

15 computer readable code means for associating a respective resource allocation with the group for the at least one respective application associated with the group to allow the at least one respective application associated with the group to be executed on the computer system according to the respective resource allocation without conflict with the at
20 least one respective application associated with other groups of the plurality of groups.

27. A memory for storing data for access by an application program being executed on a data processing system under a secure environment in which a plurality of applications
25 require shared resources, the memory comprising:

a data structure stored in said memory, the data structure including information resident in a database used by said application program and including:

for each group of a plurality of groups:

a plurality of first data objects containing information associating at least one respective application of the plurality of applications with the group; and

5 a plurality of second data objects associating a respective resource allocation with the group for the at least one respective application associated with the group to allow the at least one respective application associated with the group to be executed on the computer system according to the
10 respective resource allocation without conflict with the at least one respective application associated with other groups of the plurality of groups.

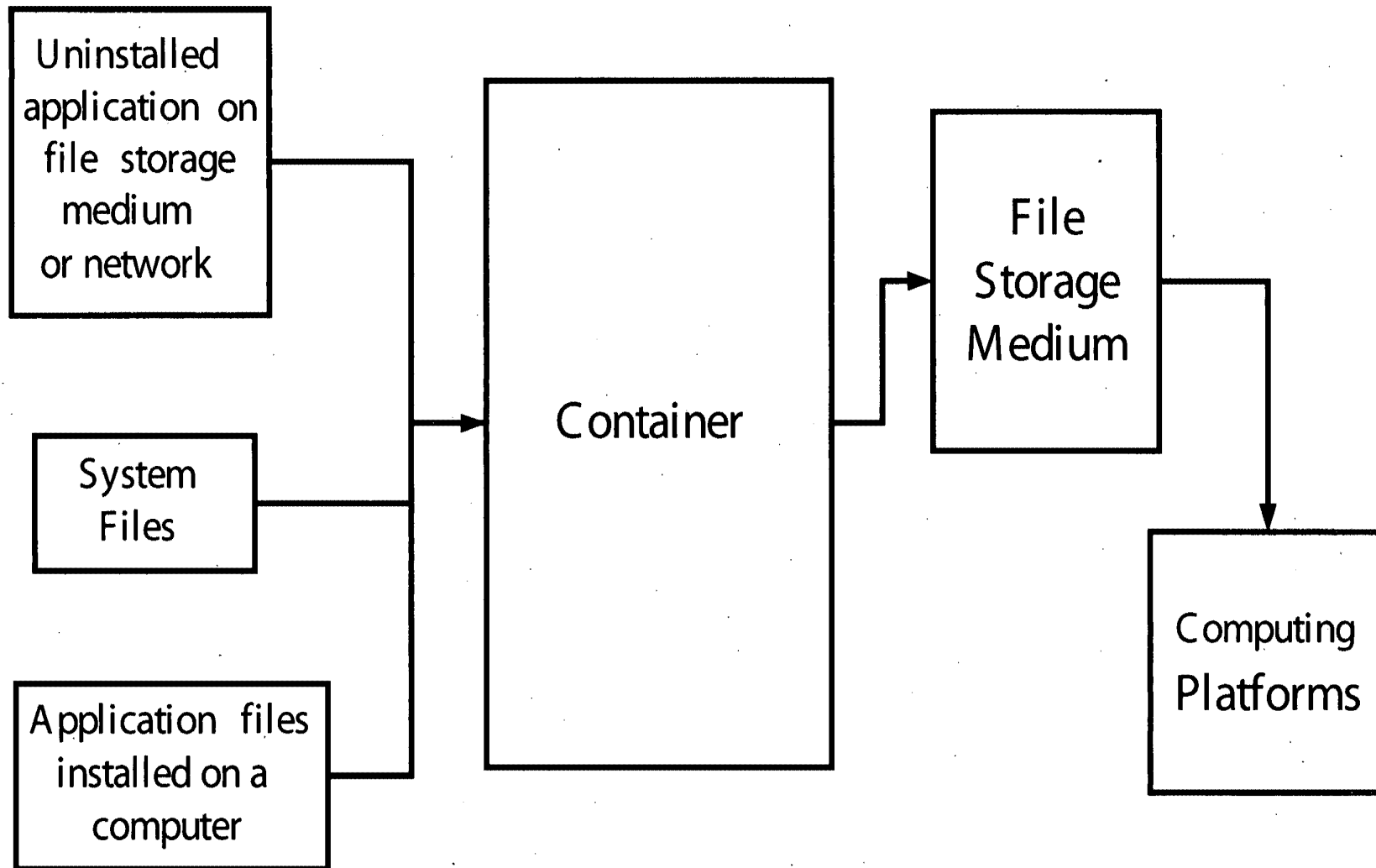
28. A method for a program to interact with a user to establish a secure environment for executing, on a computer
15 system, a plurality of applications that require shared resources, the method comprising:

displaying the plurality of applications;

for each group of a plurality of groups;

receiving a selection associating at least one
20 respective application of the plurality of applications with the group to associate a respective resource allocation with the group for the at least one respective application associated with the group and allow the at least one respective application associated with the group to be executed on the
25 computer system according the respective resource allocation without conflict with the at least one respective application associated with other groups of the plurality of groups.

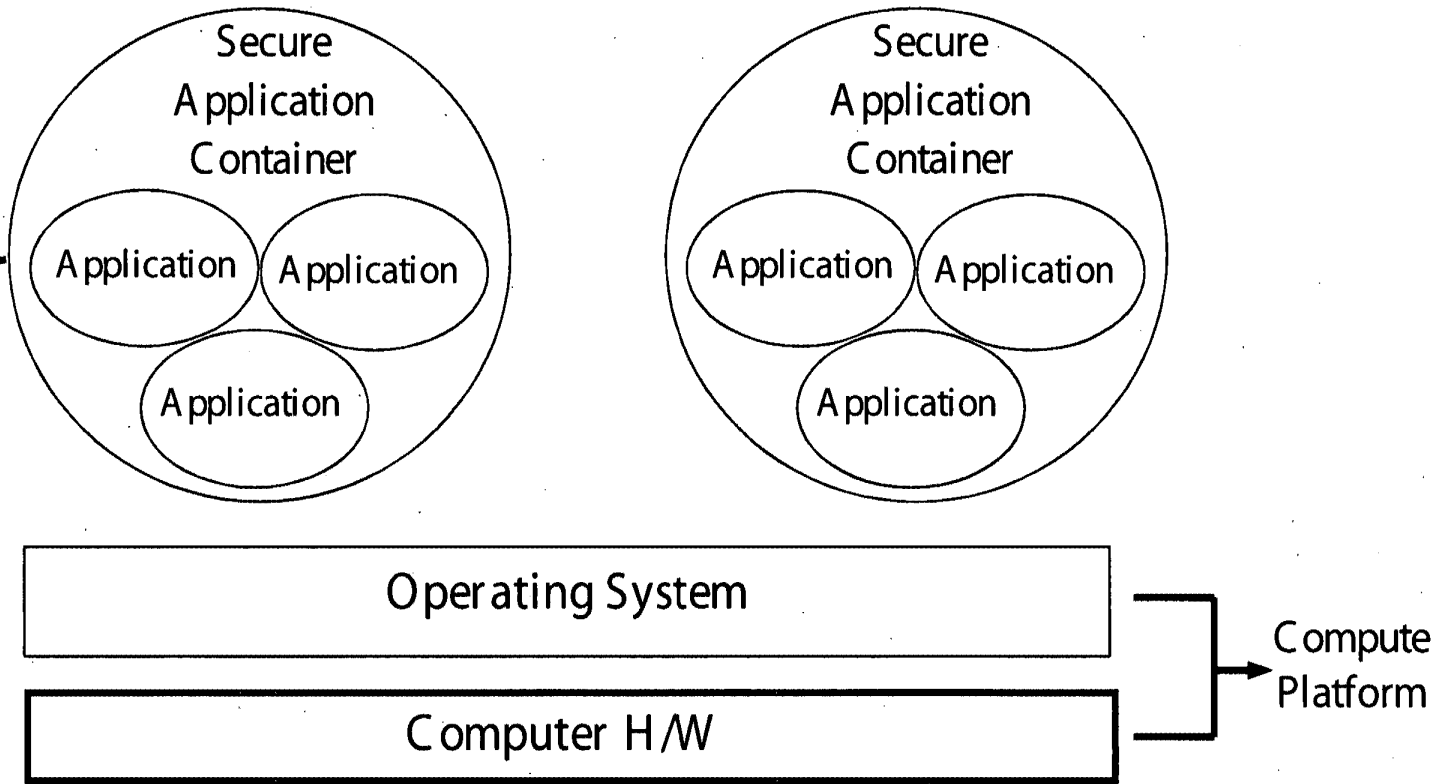
29. A system adapted to perform the method steps of any one of claims 1 to 25 and 28.



1/3

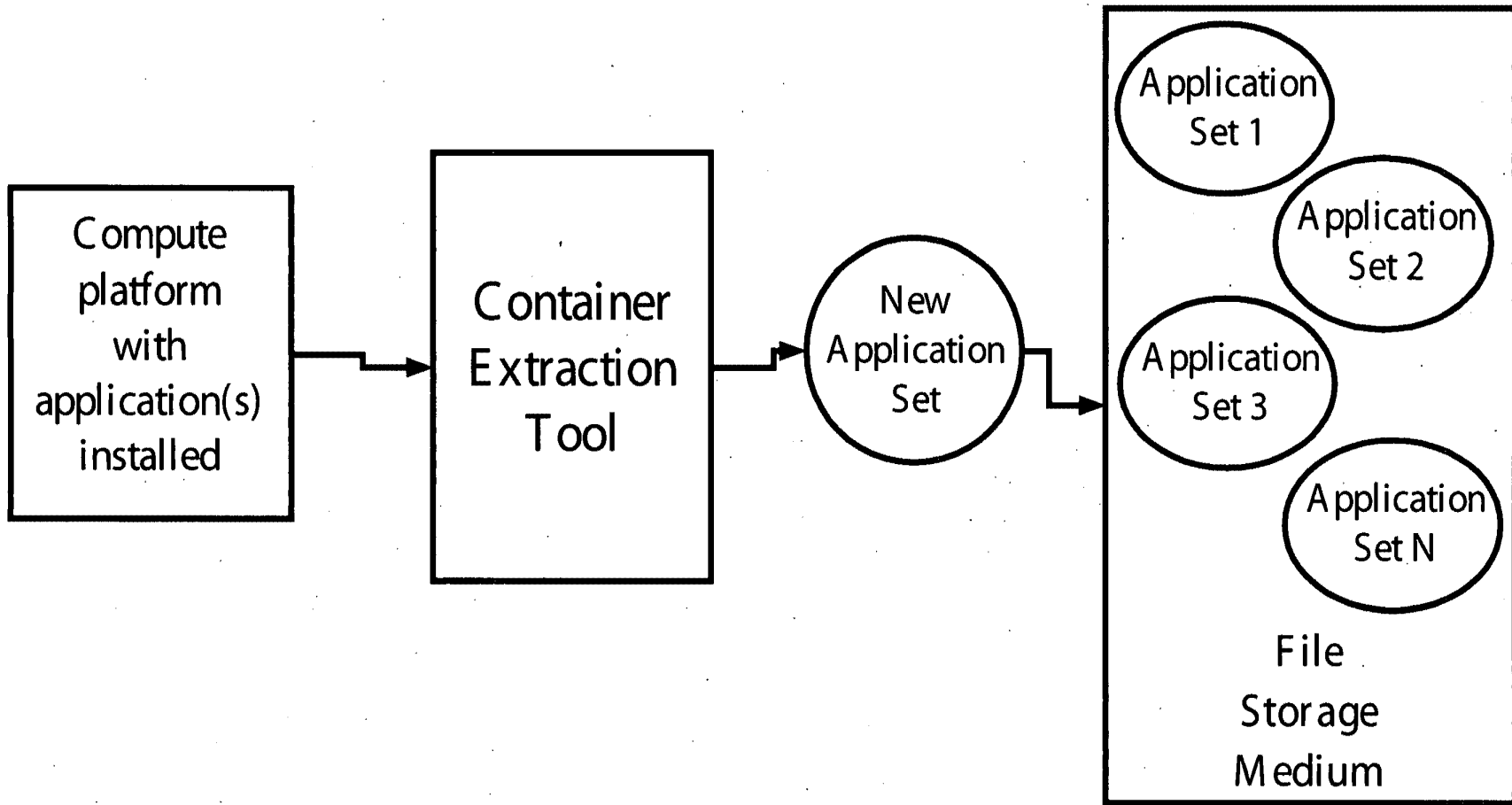
Figure 1.

Application Set



2/3

Figure 2.



3/3

Figure 3.

PATENT APPLICATION SERIAL NO. _____

U.S. DEPARTMENT OF COMMERCE
PATENT AND TRADEMARK OFFICE
FEE RECORD SHEET

10/23/2003 MAHMED1 00000038 60512103

01 FC:2005

80.00 DP

PTO-1556
(5/87)