



## [12] 发明专利申请公布说明书

[21] 申请号 200680019185.4

[43] 公开日 2008 年 6 月 25 日

[11] 公开号 CN 101208928A

[22] 申请日 2006.5.12

[21] 申请号 200680019185.4

[30] 优先权

[32] 2005.6.3 [33] US [31] 11/145,530

[86] 国际申请 PCT/US2006/018752 2006.5.12

[87] 国际公布 WO2006/132765 英 2006.12.14

[85] 进入国家阶段日期 2007.11.30

[71] 申请人 微软公司

地址 美国华盛顿州

[72] 发明人 R·A·弗兰科 A·P·盖加姆  
J·G·贝德沃茨  
P·T·伯德瑞特 R·K·托库米[74] 专利代理机构 上海专利商标事务所有限公司  
代理人 顾嘉运

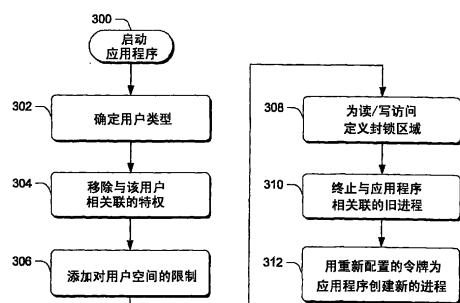
权利要求书 2 页 说明书 11 页 附图 6 页

## [54] 发明名称

运行具有低权限的因特网应用程序

## [57] 摘要

在各种实施例中，被配置成以某些方式与因特网交互的应用程序在具有简化特权级别的受限进程中被执行，其中该简化特权级别可以禁止应用程序访问计算设备的各部分(100)。例如，在一些实施例中，受限进程能禁止应用程序对系统的诸如硬盘的计算机可读介质各部分的读和写访问，其中该计算机可读介质包含管理数据和设置信息以及用户数据和设置。在这些实施例中，指定被称“封锁区域”(110)的磁盘的特别部分，并由这一受限进程中的各应用程序使用。



1. 一种计算机实现方法，包括：

提供阻断机制，所述阻断机制被配置用以阻断因特网应用程序对在其上执行所述因特网应用程序的客户端计算设备的已定义空间的访问；以及

定义至少一个所述因特网应用程序将在其中写入和读取数据的封锁区域。

2. 如权利要求 1 所述的方法，其特征在于，所述已定义空间包括所述客户端计算设备的管理空间和用户空间。

3. 如权利要求 1 所述的方法，其特征在于，所述阻断机制被配置用以按独立于用户的方式来阻断访问。

4. 如权利要求 1 所述的方法，进一步包括逻辑地将代理人机制插入到所述因特网应用程序和所述已定义空间之间，以代理对所述已定义空间的访问。

5. 如权利要求 4 所述的方法，其特征在于，所述代理人机制包括单独的代理人对象，所述单独的代理人对象中的每一个都与一个不同的已定义空间相关联。

6. 如权利要求 5 所述的方法，其特征在于，一个已定义空间包括用户空间，并且一个已定义空间包括管理空间。

7. 如权利要求 4 所述的方法，其特征在于，所述代理人机制被配置用以使得用户能够批准对关联已定义空间的访问。

8. 如权利要求 1 所述的方法，其特征在于，提供阻断机制的所述动作通过重新配置与所述因特网应用程序的用户相关联的令牌来执行。

9. 如权利要求 1 所述的方法，其特征在于，提供阻断机制的所述动作通过在与所述因特网应用程序的用户相关联的令牌上设置完整性级别来执行。

10. 如权利要求 1 所述的方法，进一步包括使用垫层把到已定义空间的所尝试的访问重定向到封锁区域。

11. 如权利要求 1 所述的方法，其特征在于，所述因特网应用程序包括 web 浏览器应用程序。

12. 如权利要求 1 所述的方法，其特征在于，还包括，作为用户和所述因特网应用程序交互的结果，启动不被阻断机制阻断的且使用与所述至少一个封锁区域隔离的封锁区域的不同的因特网应用程序。

13. 一种计算机实现方法，包括：

提供基于令牌的阻断机制，所述阻断机制被配置用以阻断因特网应用程序对在其上执行所述因特网应用程序的客户端计算设备的至少管理和用户空间的访问；

定义至少一个所述因特网应用程序将在其中写入和读取数据的封锁区域；

逻辑地把管理代理人对象插入到所述因特网应用程序和所述管理空间之间，以代理对所述管理空间的访问；以及

逻辑地把用户空间代理人对象插入到所述因特网应用程序和所述用户空间之间，以代理对所述用户空间的访问。

14. 如权利要求 13 所述的方法，其特征在于，所述代理人对象被配置用以使得用户能够批准对关联已定义空间的访问。

15. 如权利要求 14 所述的方法，其特征在于，所述管理代理人对象被配置用以提醒管理用户输入关联证书以访问所述管理空间。

16. 如权利要求 13 所述的方法，其特征在于，提供基于令牌的阻断机制的所述动作包括：

从与所述因特网应用程序的用户相关联的令牌中移除特权；以及  
在所述令牌上添加对所述用户空间的访问限制。

17. 如权利要求 16 所述的方法，其特征在于，添加限制的所述动作包括：

从所述令牌中移除用户名；以及

其中定义至少一个封锁区域的所述动作通过用组名来替代所述被移除的用户名来执行，其中所述组名将所述至少一个封锁区域指定为用于所述组名各成员的读/写访问的唯一区域。

18. 如权利要求 13 所述的方法，其特征在于，提供基于令牌的阻断机制的所述动作包括在关联令牌上设置完整性级别。

19. 如权利要求 13 所述的方法，其特征在于，所述因特网应用程序包括 web 浏览器应用程序。

20. 如权利要求 13 所述的方法，其特征在于，还包括，作为用户和所述因特网应用程序交互的结果，启动不被阻断机制阻断的且使用与所述至少一个封锁区域隔离的一个封锁区域的不同的因特网应用程序。

## 运行具有低权限的因特网应用程序

### 技术领域

本发明涉及运行具有低权限的因特网应用程序。

### 背景

许多不同类型的应用程序能够与因特网交互并从因特网获取数据或其他信息。例如，一些应用程序能允许用户下载特定内容，如网页、文件等。与这种交互相关联的各种危险也随着与因特网交互的能力而出现。

例如，通过在应用程序和因特网之间发生的各种互动，通常所说的恶意软件（malware）或间谍软件（spyware）可以被下载到用户系统上并可以有害地影响系统性能，并且也许更重要的是，可以未经允许地安装恶意软件。例如，缓冲区溢出（buffer overrun）和其他安全漏洞能允许恶意软件恶意地进入到用户系统上。

关于影响系统性能，考虑下列各项。在一些例子中，恶意软件可以尝试或可以实际上改变与一特定应用程序或总体上与用户系统关联的安全设置，从而使得恶意篡改更有可能发生。

针对这些和其他安全考虑的背景，软件开发者仍旧经常期望给用户提供一种安全、丰富的体验。

### 概述

在各种实施例中，被配置成以某些方式与因特网交互的应用程序在具有简化特权级别（reduced privilege level）的受限进程（restricted process）中被执行，其中该简化特权级别可以禁止应用程序访问计算设备的各部分。例如，在一些实施例中，受限进程能禁止应用程序对系统的诸如硬盘的计算机可读介质的部分的读和写访问，其中该计算机可读介质包含管理数据和设置信息以及用户数据和设置。在这些实施例中，指定被称为“封锁区域（containment zone）”的磁盘的特别部分，并由这一受限进程中的应用程序使用。

---

在其他实施例中，利用一个代理人机制（broker mechanism）并在逻辑上将其插入到应用程序和计算系统的受限部分或封锁区域之间。该代理人机制代理对这些受限部分的访问，并确保用户知晓并且能够批准该应用程序对计算系统的这些受限部分的访问。

在其他实施例中，一个垫层机制（shim mechanism）被用来通常为第三方扩展而把访问重定向到封锁区域。

在其他实施例中，受限进程中的应用程序执行会导致另一应用程序被启动，该另一应用程序在功能上类似于受限的应用程序，然而受到较少限制以便于已经被认为是可信赖的或至少像所期望的那样安全的特定上下文中的用户体验。

### 附图简述

图 1 是依照一个实施例的系统的方框图。图 2 是依照一个实施例的系统的方框图。

图 3 是描述依照一个实施例的一种方法中的步骤的流程图。

图 4 是依照一个实施例的系统的方框图。

图 5 是依照一个实施例的系统的方框图。

图 6 是依照一个实施例的客户端计算设备的方框图。

### 详细描述

### 纵览

在下面所描述的实施例中，被配置成以某些方式与因特网交互的应用程序在具有简化特权级别的受限进程中被执行，其中该简化特权级别可以禁止应用程序访问计算设备的部分。例如，在一些实施例中，受限进程能禁止应用程序对系统的诸如硬盘的计算机可读介质的部分的读和写访问，其中该计算机可读介质包含管理数据和设置信息以及用户数据和设置。在这些实施例中，指定被称“封锁区域”的磁盘的特别部分，并由这一受限进程中的应用程序使用。

在其他实施例中，利用一个代理人机制并在逻辑上将其插入到应用程序和计算系统的受限部分或封锁区域之间。代理人机制代理对这些受限部分的访问，并确保用户知晓并且能够批准该应用程序对计算系统的这些受限部分的访问。

在其他实施例中，一个垫层机制被用来通常为第三方扩展而把访问重定向到封锁区域。

在其他实施例中，受限进程的应用程序执行会导致另一应用程序被启动，该另一应用程序在功能上类似于受限的应用程序，然而受到较少限制以便于已经被认为是可信赖的或至少像所期望的那样安全的特定上下文中的用户体验。

可以结合与因特网交互的任何类型应用程序使用在这一文档中所描述的技术。熟练技术人员将会理解，有很多各种各样的这些类型的应用程序。然而，为提供一个实际的上下文以理解发明实施例，利用以 web 浏览器应用程序形式的应用程序。然而可以理解，技术可以与其他类型的应用程序一起使用而不会偏离所要求主题的精神和范围。作为例子而非限制，这些其他类型的应用程序包括即时消息客户端、对等客户端、RSS 阅读器、电子邮件客户端、字处理客户端等。

### 限制因特网应用程序以及使用代理人

图 1 例示依照一个实施例的系统 100 的高级别视图。在这一实例中，系统 100 包括可以与因特网交互的 web 浏览器 102 的形式的因特网应用程序。系统 100 也包括包含不同部分或“空间”的诸如硬盘指令的计算机可读介质 104，其中该不同部分或“空间”包含不同类型的信息、设置数据等。

在这一实例中，一个部分或空间是包括通常是可由系统管理员访问并操作的信息和数据的管理空间 106。这一类型的信息和数据可以包括通常被包含在操作系统文件夹、计算机系统文件夹、永久文件文件夹等中的信息和数据。这一空间通常需要管理员带有适当证书和特权以使其内容能被访问和操作。

另一部分或空间是包括用户信息和数据的用户空间 108。这一类型的信息和数据可以包括通常被包含在诸如“我的文档”、“我的音乐”、“桌面”之类的用户可访问文件夹中的信息和数据。这一空间通常与较少的特权关联，以使得访问可被准许。

依照一个实施例，计算机可读介质 104 包括一个或多个封锁区域 110。封锁区域是至少在一些实施例中可由浏览器 102 直接写入的唯一区域。为了促进这一功能性，提供一个墙 (wall) 或阻断 (blocking) 机制 112，该机制阻止浏览器 102 直接地写入管理空间 106 或用户空间。在至少一些实施例中，封锁区域允许对在各会话之间要被保存到它们不能够污染机器上任何其他应用程序的位置上的受限应用程序进行设置。封锁区域可以包括一些注册表位置和文件

文件夹。在 web 浏览器应用程序的上下文中，封锁区域 110 可以包括被用来改进网页加载时间和缓存其他类型的数据的临时因特网文件文件夹。

因此，在这一实施例中，具体地定义一个或多个封锁区域并将其指定为诸如 web 浏览器应用程序的因特网应用程序可以访问的那些计算设备的部分。这不同于基于可能尝试这类访问的特定用户而简单地拒绝对磁盘各部分的访问并允许对其他各部分的访问的方式。相反，在本发明类型的方式中，限制是以应用程序为中心的，而不是必然以用户为中心的。即是说，本发明方式可以被认为是独立于用户的。这一方式有助于确保只有少数（例如，最少）所需要的位置被暴露在封锁区域中，并且这一方式还有助于确保其他应用程序不将设置存储在该封锁区域中。另外，这种以应用程序为中心的方式能使得管理和用户空间两者都不能被该应用程序访问。

因此，在这一点上，墙或阻断机制 112 在逻辑上被插入到浏览器 102 和诸如管理和用户空间之类的某些预定义的空间之间，以阻止浏览器直接地访问这类空间。然而，在一些例子中，希望允许应用程序访问管理或用户空间。例如，是系统管理员的用户可能希望合法地操作一些系统设置。另外，常规用户可能希望将一张照片保存到“我的文档”文件夹。

在这一实施例中，利用代理人机制并将其逻辑地插入到应用程序（在这一情况中是浏览器 102）和计算系统的受限部分或封锁区域之间。代理人机制代理对这些受限部分的访问，并确保用户知晓并且能够批准该应用程序对计算系统的这些受限部分的访问。

作为例子，考虑图 2，其中使用来自图 1 实施例的类似数字。其中，以代理人对象 200、202 的形式提供一个代理人机制。在这一实例中，代理人对象 200 是一个管理空间代理人对象，它代理对管理空间 106 的访问。另一方面，代理人对象 202 是一个用户空间代理人对象，它代理对用户空间的访问。该代理人机制能够使用任何合适类型的对象以任何合适方式实现。在一种实现中，每个代理人对象都被实现为 DCOM 本地服务器对象（DCOM local server object）。另外，代理人对象在与浏览器 102 分隔开的进程中运行，从而提供防止以浏览器 102 为目标的恶意代码的攻击的某种程度的保护。另外，在至少一种实现中，代理人对象是基于任务的，并且它们的生存期是由它们将要完成的任务所定义的。

在这一实例中，当诸如浏览器 102 的应用程序希望访问诸如管理或用户空

间的特定受限空间时，该应用程序调用该关联代理人对象，该代理人对象随后则检查该应用程序的请求。代理人对象可以出于多个理由检查该请求，其中包括确保它是构造良好的请求，或者检查该应用程序正在下载的文件上的电子签名。一旦请求已检查，代理人对象采取措施来代理对受限空间的访问。

在一些实施例中，这可以包括提示用户确定该用户是否希望以在该要求中所表示的方式来访问该空间。例如，如果用户正在尝试把一张照片保存到他们的“我的文档”文件夹，代理人对象可以通过一个适当的对话框简单地询问用户这是否是该用户的意图。如果得到证实，那么代理人对象可以允许并促进该访问。备选地或附加地，如果用户是管理员并且正在尝试写入管理空间，则代理人对象可以请求管理员输入他们的证书。以这一方式，就能维护对受限空间的访问。在这些例子中，代理人对象执行写入或修改受限空间，以便将该进程从正在发起调用的应用程序提取出来。

因此，墙或阻断机制 112 和代理人机制 200、202 一起工作以阻断对磁盘各受限部分的访问，但却不禁止适当场合中对那些部分的访问。

已经探讨了墙或阻断机制以及代理人机制的概念，接下来的讨论只提供如何可以实现阻断机制的一个实例（连同一个备选实例）。应该认识并理解，能够以其他方式实现阻断机制和代理人机制而不会偏离所要求主题的精神和范围。

### 阻断机制——实现示例

在以下讨论中，在把低权限强加给因特网应用程序的令牌化系统的上下文中描述一个阻断机制。低权限的强加反过来引起该应用程序对诸如管理和用户空间之类的客户端系统的特定部分的访问的限制。在第一实施例中，对不必然被构建成固有地允许这一类型的以应用者为中心的功能性的令牌进行处理，并将其重新配置以实现这一功能性。在第二实施例中，令牌通过所谓的“完整性级别”被构建，以允许上述以应用程序为中心的功能性。

### 第一实施例——重新配置令牌

在许多系统中，当用户运行或执行应用程序时，该应用程序在用户的上下文中执行。这意味着用户通常拥有约束该应用程序的执行的诸如用户名和用户特权之类的用户数据。更具体地，用户名和特权能由令牌表示且在令牌的上下文中表示。因此，当用户执行应用程序时，该应用程序经由令牌就可知晓并继承诸如用户特权之类的用户上下文的各方面。因此，如果用户是系统管理员，

那么关联令牌将会把该用户标识为系统管理员，而且该应用程序会继承该系统管理员特权，而该系统管理员特权反过来允许该应用程序写入上述管理空间。

图 3 是描述依照一个实施例的令牌处理方法的各步骤的流程图。该方法能够以任何合适的硬件、软件、固件或它们的组合来实现。在一个实施例中，该方法的各方面由一个别合适地配置的应用程序实现，诸如由图 1 和 2 中的浏览器应用程序 102 实现。

步骤 300 启动一个应用程序，而该应用程序在本示例中是一个诸如以上示出并描述的浏览器之类的 web 浏览器。当用户启动该应用程序时，与该用户相关联的令牌变得可以由能如上所述从中继承用户特权的该应用程序使用。

步骤 302 确定用户类型。可以存在不同类型的用户，如管理用户、高级用户（power user）、备份操作员等。步骤 304 移除与用户类型关联的特权。在所例示的实施例中，该步骤通过有效地操作令牌的数据以移除指示与令牌关联从而移除与用户类型关联的任何特权的标志而得以实现。这一步骤本质上把一个块创建到计算设备的管理空间，诸如图 1 和 2 中的管理空间 106。

步骤 306 添加对用户空间的限制。在所例示并描述实施例中，这通过有效地操纵令牌的数据以把用户名从该令牌中移除而得以实现。通过把用户名从令牌中移除，与特定用户相关联的特权也被移除。

然后，步骤 308 为读/写访问定义一个或多个封锁区域。在这一特定实例中，这一步骤通过用一个特别定义的用户组名称（例如“IEUsersGroup”）来替代所移除的用户名而得以实现。现在，对于一个或多个封锁区域，这些区域是被指定为用于该已特别定义的组名各成员的读/写访问的唯一区域。

因此，在这一点上，任何管理特权已被移除，从而有效阻断管理空间。同样地，用户特权已被移除，从而阻断对用户空间的访问。然而，通过将用户名改变为一特别组名并将该特别组名与一个或多个封锁区域相关联，该应用程序的读/写访问现在就被限制仅为上述的一个或多个封锁区域。

更具体地，如上所述继续行进，步骤 310 终止与被启动的应用程序相关联的旧进程，并且步骤 312 用重新配置的令牌为应用程序创建新的进程。

使用这一已被重新配置的令牌，该应用程序将会无法直接地访问管理空间或用户空间。相反，该应用程序将只能够直接地写入封锁区域，并且由于不受（例如，代理人机制的）进一步干预，该应用程序将无法使得数据被写入用户或管理空间。

## 第二实施例——使用完整性级别

在另一实施例中，通过所谓的“完整性级别”利用并且构建令牌，从而允许上述以应用程序为中心的功能性。即是说，通过被称为强制完整性控件（Mandatory Integrity Control）的进程，与用户相关联的令牌具有不同的完整性级别，诸如可以被设置为“高”、“中”和“低”。同样地，客户端设备上的计算资源具有相关联的完整性级别，并且为了访问资源，该资源必须具有与用户完整性级别相同或更低的完整性级别。

因此，例如，通过将管理和用户空间的完整性级别分别建立为“高”和“中”，将用户的完整性建立为“低”，就能够有效阻断对管理和用户空间的访问。然而，将封锁区域指定为具有“低”级别的完整性就能允许用户通过用户正在使用的任何应用程序来访问该封锁区域。

### 使用垫层

在至少一些实施例中，诸如在图 4 的垫层 400 的垫层机制被用来通常为第三方扩展而把访问重定向到封锁区域。更具体地，在浏览器应用程序的上下文中，可以提供许多不同的第三方扩展，并且这些第三方扩展结合该浏览器或者在该浏览器内运行。例如，Google 工具栏是被设计用于在浏览器内运行的扩展的一个实例。

特定扩展通常要求对文件系统或注册表的部分的写入访问，以便正确地工作。例如，Google 工具栏可能希望保存特定用户的喜好搜索的列表。然而，若不访问用户空间，这一类型的写入就会被墙或阻断机制 112 阻断。

依照一个实施例，当应用程序 102 或关联的第三方组件尝试写入受限空间时，垫层 400 被配置用以俘获和重定向该调用并将数据写入封锁区域。该应用程序对被重定向到该封锁区域的数据的随后调用由该垫层处理，并从封锁区域检索适当的数据。因此，特定扩展或应用程序想要写入到管理或用户空间的数据被重定向到适当的封锁区域。

这允许第三方扩展继续工作而无须要求重写任何第三方代码。在工作中，第三方扩展相信它正在把数据写入用户或管理空间。然而，通过该垫层机制，这样的数据就被写入封锁区域并从中读取。

### 启动不受限的应用程序

如上所述，在其他实施例中，受限进程中的应用程序执行会导致另一应用程序被启动，该另一应用程序在功能上类似于受限的应用程序，然而受到较少

限制以便于已经被认为是可信赖的或至少像所期望的那样安全的特定上下文中的用户体验。

作为一个更实际的实例，在浏览器上下文中考虑下列各项。假定一个公司用户可以通过他们的客户端计算设备访问因特网和公司内联网。也假定公司内联网是安全的和可信赖的实体。进一步假定用户的计算设备正在执行需要高度兼容性以保持正确运行的若干不同商业应用程序。在像这样和其他的上下文中，当在公司内联网的上下文中执行时，希望能够允许应用程序以不受限方式工作——即，以不受阻断机制 112 限制的方式工作。

作为例子，结合下列各项考虑图 5。存在应用程序可以尝试在其中运行的特定上下文，而且这些上下文属于已经被定义为是可信赖的或在其他方面带有已经被定义为“安全”的安全性级别的特定区域。在浏览器的示例中，用户可以尝试导航到一个公司内联网或其他安全区域。在这种情况下，受限浏览器 102 调用代理人机制，而该代理人机制基于应用程序正做出的调用就可实例化不受限浏览器 500，其中该用户可以用该不受限浏览器 500 在他们已经导航至的特定区域中操作。在这一示例中，令牌被创建并被配置成包括与用户相关联的特权（如管理特权，高级用户特权等）以及与用户相关联的用户名，从而向用户提供对用户空间适当部分的访问。

另外，在这一实施例中，以各自在受限和不受限浏览器 102 和 500 之间维持隔离的方式来定义封锁区域。具更具体地，回想提供了一种受限浏览器 102 和其他组件可对其进行读写的按临时因特网文件文件夹形式的封锁区域。然而在本实施例中，如果不受限浏览器 500 使用这一封锁区域来写入临时因特网文件，则会存在这样的机会，即受限浏览器可以访问这一数据或使用这一封锁区域重叠来尝试恶意地获得对它应该不能访问的计算设备部分的访问。

因此，为解决这一情况和其他情况，定义不同的封锁区域，这些不同封锁区域之一与受限浏览器 102 相关联，其他的封锁区域与不受限浏览器 500 相关联并与该受限浏览器隔离开来。在所例示的实例中，封锁区域 110a 与浏览器 102 关联，并只能由浏览器 102 使用。而且，封锁区域 110b 与不受限浏览器 500 关联，并只能由浏览器 500 使用。两个浏览器都不能对其他的关联封锁区域进行读写。由此可观察到墙 112 向下扩展并阻断从受限浏览器 102 到封锁区域 110b 的访问。

在令牌被处理并被重新配置的以上实现中，封锁区域 110a 被指定为只可

以由令牌中所标识的组从中读取并向其写入。因此，在这一令牌的上下文中执行的应用程序不能够访问封锁区域 110b。

### 示例性的使用场景

下列使用场景提供在 web 浏览器的上下文中如何利用上述发明各实施例的一些附加例子。

首先考虑一个实例，其中发明各实施例可以被用来保护用户。假定用户 Abby 访问一个利用浏览器中的缓冲区溢出来安装一个控件的网站。在这里，Abby 导航到使用浏览器中的缓冲区溢出漏洞来将本地代码注入到进程空间中的页面。本地代码将一个动态链接库（DLL）下载到她的机器上的文件夹中，并尝试通过在注册表中创建条目来注册为由该浏览器加载的 ActiveX 控件。然而这里因为浏览器不被允许写入到注册表，该操作失败。然后 Abby 接收到一个通知，继续安全地浏览。

作为另一实例，假定用户 Abby 访问一个使用她已经安装的控件来尝试盖写系统的网站。在这里，Abby 导航到包含已经安装的 ActiveX 控件的页面。该控件尝试盖写她的系统文件夹中的一个 DLL。然而在这里，该操作被拒绝，Abby 接收到一个告知她该页面尝试执行一个特许操作的通知。然后她继续安全地浏览。

现在考虑一个其中发明各实施例能被用来维持 Abby 的系统的兼容性的实例。这里，假定 Abby 从一个网站升级她的视频驱动程序。Abby 导航到该网站并点击对 driver.exe 文件的链接。该文件被下载，并且可执行的安装代理人（也就是代理人机制）提示 Abby 以确认她信赖该可执行（文件）并希望安装之。如果经 Abby 批准，安装成功地完成，Abby 继续安全地浏览。

现在假定 Abby 访问她的收藏夹网站。已经添加一个新的菜单控件，因此浏览器需要安装该控件。提示 Abby，询问她是否信赖该控件并授权该安装。如果被批准，安装该控件，Abby 继续导航至该网站并安全地浏览。

### 示例性的计算系统

图 6 示出具有可以被用来实现上述一个或多个实施例的组件的示例性计算机系统。

计算机系统 630 包括一个或多个处理器或处理单元 632、系统存储器 634 和将包括系统存储器 634 在内的各种系统组件耦合到处理器 632 的总线 636。总线 636 代表多种类型的总线结构中的任何一个或多种，包括存储器总线或存

储器控制器、外围总线、加速图形端口，以及处理器或使用各种总线体系结构中的任何一种的局部总线。系统存储器 634 包括只读存储器 (ROM) 638 和随机存取存储器 (RAM) 640。基本输入/输出系统 (BIOS) 642 存储在 ROM 638 中，它包含比如在启动过程中帮助在计算机 630 内的元件之间传输信息的基本例程。

计算机 630 进一步包括用于从硬盘 (未示出) 读取和向其写入的硬盘驱动器 644、用于从可移动磁盘 648 读取和向其中写入的磁盘驱动器 646、用于从诸如 CD ROM 或其他光学介质的可移动光盘 652 读取和向其写入的光盘驱动器 650。硬盘驱动器 644、磁盘驱动器 646 和光盘驱动器 650 由 SCSI 接口 654 或其他适当的接口连接到总线 636。驱动器及其相关的计算机可读介质为计算机 630 提供计算机可读指令、数据结构、程序模块和其他数据的非易失性存储。尽管在此描述的示例性环境使用硬盘、可移动磁盘 648 和可移动光盘 652，但本领域中的技术人员应该明白，可以存储计算机可访问数据的其他类型计算机可读介质，如盒式磁带、闪存卡、数字视频盘、随机存取存储器 (RAM)、只读存储器 (ROM) 等等，也可以用于该示例性操作环境。

一些程序模块可以被存储在硬盘 644、磁盘 648、光盘 652、ROM 638 或 RAM 640 中，包括操作系统 658、一个或多个应用程序 660、其他程序模块 662 和程序数据 664。用户可以通过诸如键盘 666 的输入设备和定位设备 668 向计算机 630 输入命令和信息。其他输入设备 (未示出) 可以包括话筒、操纵杆、游戏垫、圆盘式卫星天线、扫描仪等等。这些和其他输入设备通过一个被耦合到总线 636 的接口 670 被连接到处理单元 632。监视器 672 或其他类型的显示设备也经由诸如视频适配器 674 的接口被连接到总线 636。除了监视器之外，个人计算机通常包括其他的外围输出设备 (未示出)，如扬声器和打印机。

计算机 630 通常运行在使用到诸如远程计算设备 676 的一个或多个远程计算机的逻辑连接的网络化的环境中。远程计算机 676 可以是另一个人计算机、服务器、路由器、网络 PC、对等设备或其他公共网络节点，并且一般包括与计算机 630 相关的许多或所有上述元件，尽管图 6 中仅例示了存储器存储设备 678。在图 6 中所描述的逻辑连接包括局域网 (LAN) 680 和广域网 (WAN) 682。这种网络环境常见于办公室、企业范围的计算机网络、企业内联网和因特网。

当用于 LAN 网络环境时，计算机 630 通过网络接口或适配器 684 连接到

局域网 680。当用于 WAN 网络环境时，计算机 630 通常包括调制解调器 686 或其他用于在诸如因特网的广域网 682 上建立通信的装置。可以内置或外置的调制解调器 686 经由串行端口接口 656 连接到总线 636。在网络化环境中，所述与个人计算机 630 相关的程序模块或其部分可以被存储在远程存储器设备内。应该明白，所示出的网络连接是示例性的，并且可以使用在计算机之间建立通信链路的其他方式。

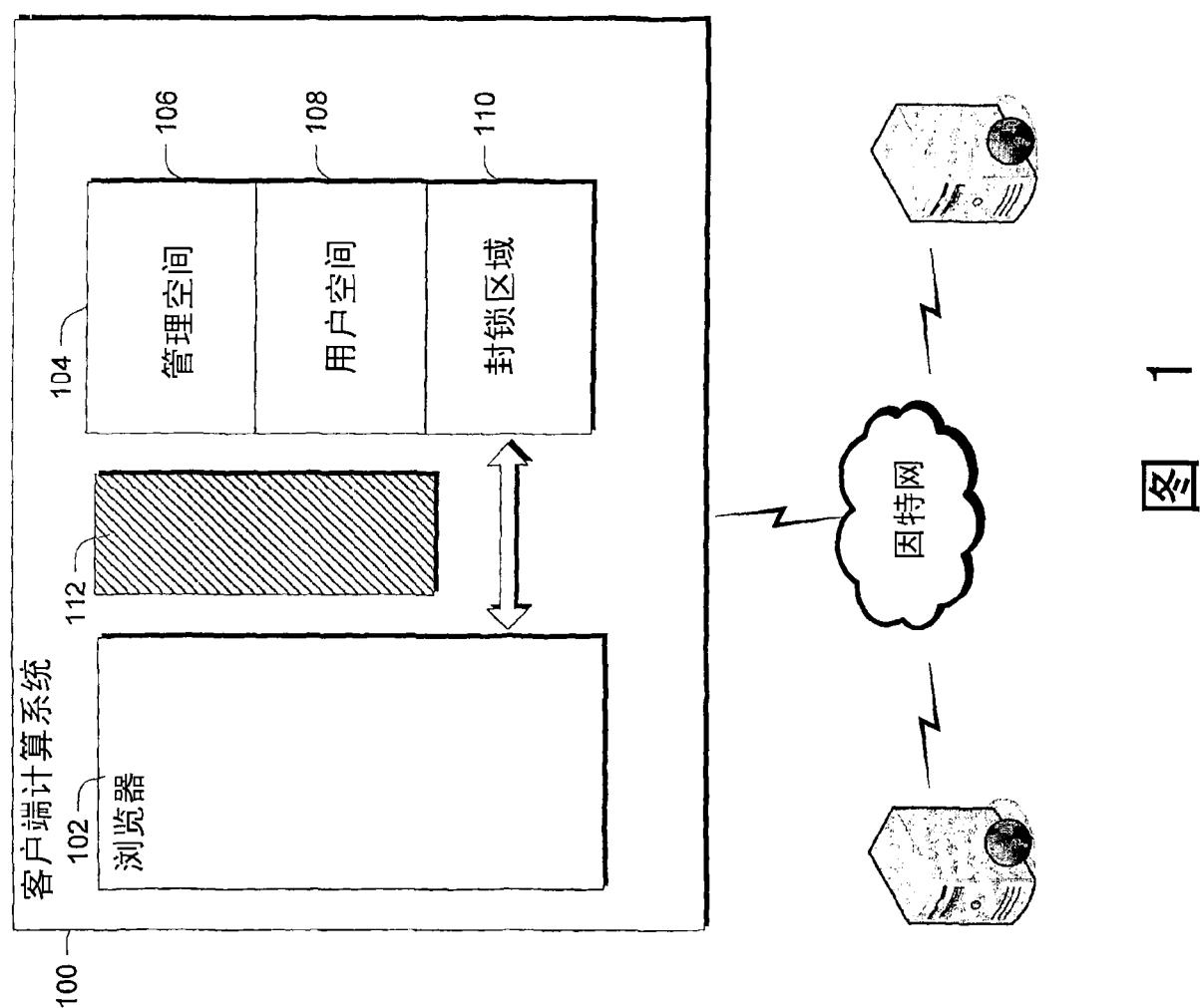
一般地，计算机 630 的数据处理器通过在不同时间存储在计算机上的各种计算机可读存储介质中的指令来编程。程序和操作系统通常是分布式的，例如，分布在软盘或 CD-ROM 上。从那里，它们被安装或加载到计算机的辅助存储器中。在执行时，它们被至少部分地加载到计算机的主要电子存储器中。当这类介质包含用于与微处理器或其他数据处理器一起实现下面描述的步骤的指令或程序时，此处描述的本发明包括这些和其他各种类型的计算机可读存储介质。当根据下面描述的方法和技术进行编程时，本发明也包括计算机本身。

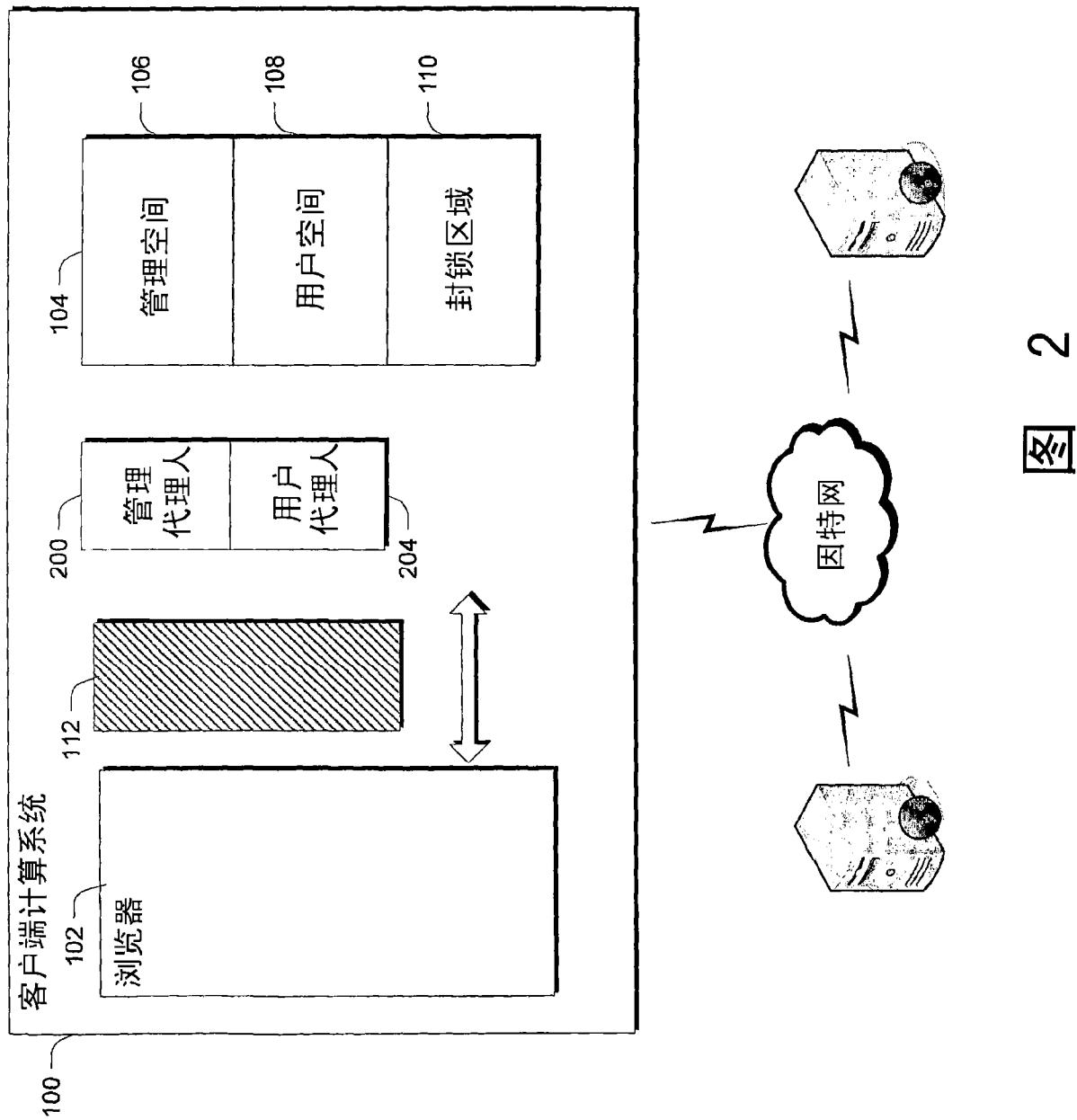
尽管认识到应用程序和诸如操作系统的其他可执行程序组件在不同的时刻驻留于计算机的不同存储组件中并由计算机的数据处理器执行，但为了进行例示，此类程序和组件被例示成离散的块。

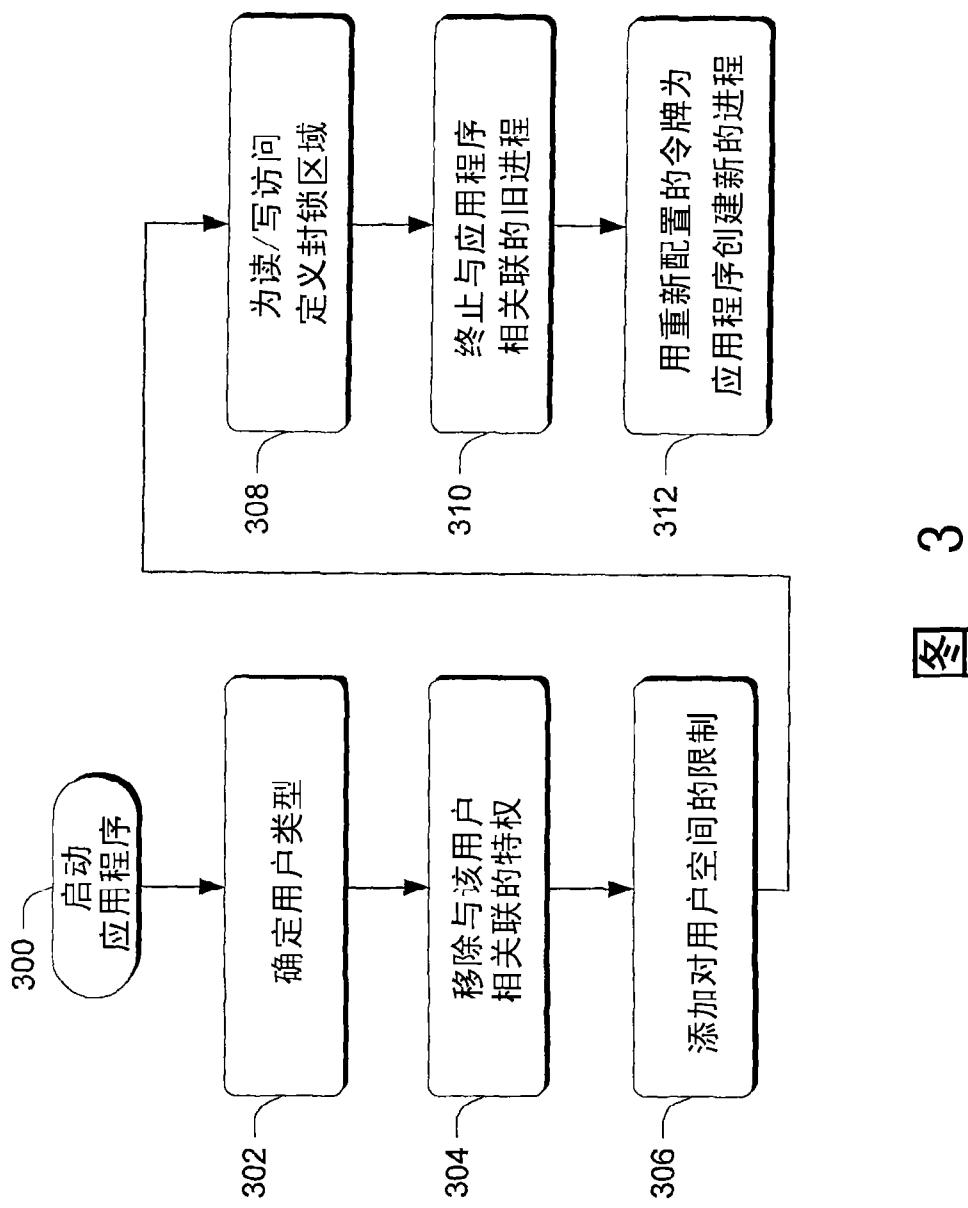
## 结论

上面所描述的实施例能减少与可以访问因特网的应用程序相关联的安全风险，同时仍向用户提供安全、丰富的体验。

尽管已经用结构特征和/或方法步骤的特有语言对本发明进行了描述，但应该明白，在所附权利要求书中定义的本发明，并不一定限于所描述的特定功能或步骤。相反，具体的特征和动作只是作为实现所要求的发明的示例性形式而揭示的。







三

四

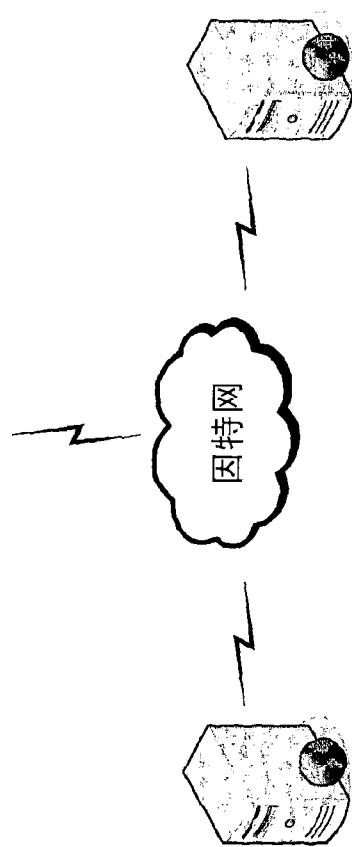
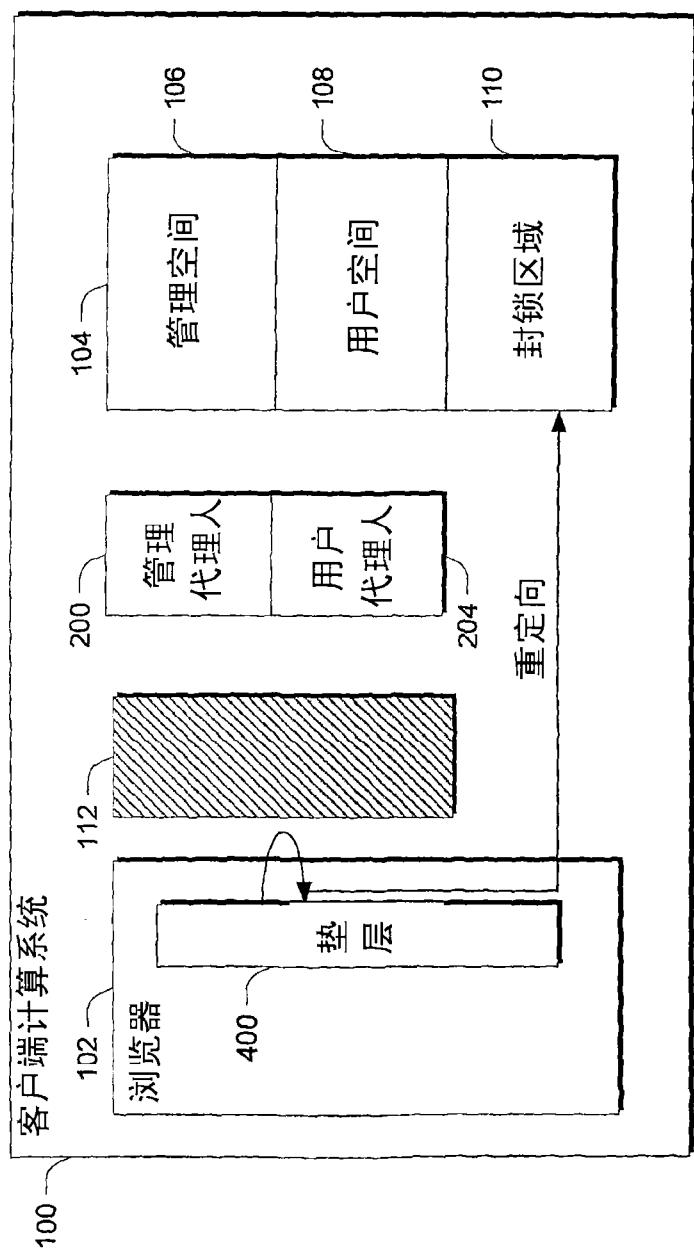


图 4

