РОССИЙСКАЯ ФЕДЕРАЦИЯ

ФЕДЕРАЛЬНАЯ СЛУЖБА
ПО ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ

(19) **RU** (11) **2 646 332**(13) **C2**

(51) МПК
*G06F 21/53* (2013.01)
*G06F 21/70* (2013.01)

# (12) ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ПАТЕНТУ

(52) СПК
***G06F 21/53** (2017.08); **G06F 21/70** (2017.08)*

(21)(22) Заявка: **2015150949, 20.09.2013**

(24) Дата начала отсчета срока действия патента:
**20.09.2013**

Дата регистрации:
**02.03.2018**

Приоритет(ы):
(30) Конвенционный приоритет:
**31.05.2013 US 13/906,902**

(43) Дата публикации заявки: **01.06.2017** Бюл. № **16**

(45) Опубликовано: **02.03.2018** Бюл. № **7**

(85) Дата начала рассмотрения заявки PCT на национальной фазе: **27.11.2015**

(86) Заявка PCT:
**US 2013/060753 (20.09.2013)**

(87) Публикация заявки PCT:
**WO 2014/193443 (04.12.2014)**

Адрес для переписки:
**129090, Москва, ул. Большая Спасская, д. 25, строение 3, ООО "Юридическая фирма Городисский и Партнеры"**

(72) Автор(ы):
**ДИАС-КУЭЛЬЯР, Херардо (US),**
**ГУПТА, Дхирадж Кант (US)**

(73) Патентообладатель(и):
**МАЙКРОСОФТ ТЕКНОЛОДЖИ ЛАЙСЕНСИНГ, ЭлЭлСи (US)**

(56) Список документов, цитированных в отчете о поиске: **US 2006/0242270 A1, 26.10.2006. US 2007/0088890 A1, 19.04.2007. US 2007/0079385 A1, 05.04.2007. US 2008/0005791 A1, 03.01.2008. US 2010/0082926 A1, 01.04.2010. WO 99/39254 A2, 05.08.1999. US 2006/0259675 A1, 16.11.2006. US 2006/0253859 A1, 09.11.2006. US 2004/0252325 A1, 16.12.2004. RU 2443012 C2, 20.02.2012.**

## (54) ОГРАНИЧЕННАЯ ПЛАТФОРМА ДРАЙВЕРОВ, КОТОРАЯ ЗАПУСКАЕТ ДРАЙВЕРЫ В ПЕСОЧНИЦЕ В ПОЛЬЗОВАТЕЛЬСКОМ РЕЖИМЕ

(57) Реферат:

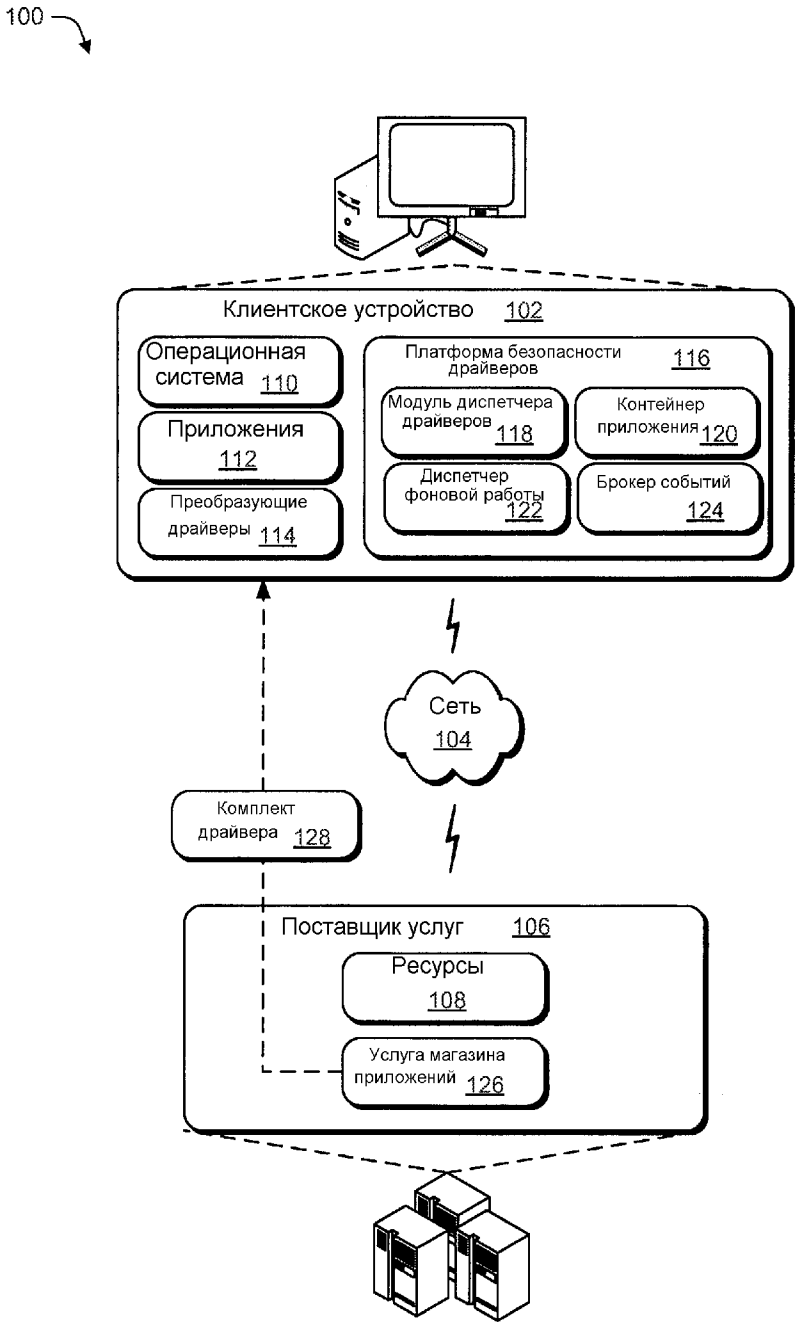
Изобретение относится к способу исполнения преобразующего драйвера, реализуемому вычислительным устройством. Технический результат заключается в повышении надежности вычислительной системы за счет обеспечения безопасности в работе преобразующих драйверов. Способ содержит этапы, на которых: получают преобразующий драйвер, содержащийся в комплекте драйвера, имеющем назначенный формат, ассоциированный с платформой безопасности драйверов; распознают назначенный формат комплекта драйвера при установке, основываясь по меньшей мере отчасти на идентификационных данных, включенных в комплект драйвера; в качестве реакции на упомянутое распознавание регистрируют преобразующий драйвер в платформе безопасности драйверов, реализуемой вычислительным устройством; создают экземпляр ограниченной среды исполнения для преобразующего драйвера посредством платформы безопасности драйверов; и исполняют



RU 2646332 C2

RU 2646332 C2

преобразующий драйвер в ограниченной среде исполнения, чтобы выполнять одну или несколько задач по указанию платформы безопасности драйверов. 3 н. и 17 з.п. ф-лы, 6 ил.




ФИГ.1

RU 2646332 C2

RU 2646332 C2

RUSSIAN FEDERATION

FEDERAL SERVICE
FOR INTELLECTUAL PROPERTY

(19) **RU** (11) **2 646 332**(13) **C2**

(51) Int. Cl.
*G06F 21/53* (2013.01)
*G06F 21/70* (2013.01)

## (12) ABSTRACT OF INVENTION

(52) CPC
*G06F 21/53 (2017.08); G06F 21/70 (2017.08)*

(21)(22) Application: **2015150949, 20.09.2013**

(24) Effective date for property rights:
**20.09.2013**

Registration date:
**02.03.2018**

Priority:
(30) Convention priority:
**31.05.2013 US 13/906,902**

(43) Application published: **01.06.2017** Bull. № **16**

(45) Date of publication: **02.03.2018** Bull. № **7**

(85) Commencement of national phase: **27.11.2015**

(86) PCT application:
**US 2013/060753 (20.09.2013)**

(87) PCT publication:
**WO 2014/193443 (04.12.2014)**

Mail address:
**129090, Moskva, ul. Bolshaya Spasskaya, d. 25, stroenie 3, OOO "Yuridicheskaya firma Gorodisskij i Partnery"**

(72) Inventor(s):
**DIAS-KUELYAR, Kherardo (US),**
**GUPTA, Dkhiradzh Kant (US)**

(73) Proprietor(s):
**MAJKROSOFT TEKNOLODZHI LAJSENSING, ElElSi (US)**

RU 2646332 C2

## (54) LIMITED DRIVER PLATFORM WHICH LAUNCHES DRIVERS IN SANDBAND IN USER REGIME

(57) Abstract:

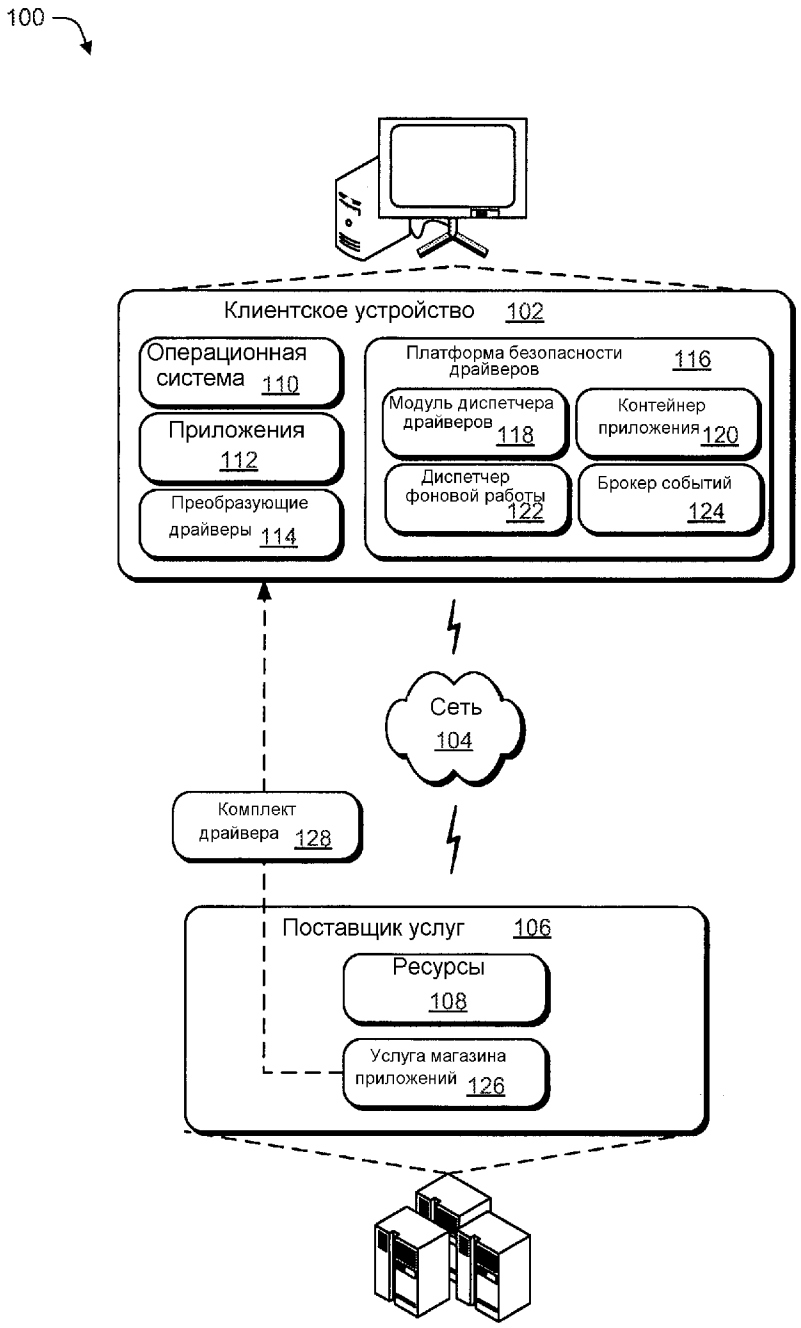
FIELD: information technology.

SUBSTANCE: method comprises the steps of: obtaining a converting driver included in the driver package having a designated format associated with the driver safety platform; recognizing the designated format of the driver package during installation, based at least in part on the identification data included in the driver package; in response to the noted recognition, a converting driver is registered in the driver safety platform implemented by the computing device; creating a copy of the limited execution environment for the converting driver through the driver safety platform; and executing the converting driver in a limited execution environment to perform one or more tasks as directed by the driver security platform.

EFFECT: increased reliability of the computer system by providing security in the work of the converting drivers.

20 cl, 6 dwg

RU 2646332 C2




ФИГ.1

RU 2646332 C2

## УРОВЕНЬ ТЕХНИКИ

[0001] В настоящее время преобразующие (transmogrifying) драйверы, которые осуществляют преобразования лежащих в основе данных из одного вида в другой, например драйверы виртуальной частной сети (VPN), могут обладать общесистемными привилегиями, высоким приоритетом и доступом к конфиденциальным данным, и поэтому могут создавать значительные угрозы безопасности. Частично из-за этих вопросов пользователи могут с большой неохотой устанавливать такие драйверы и ассоциированное программное обеспечение на свои устройства. Например, пользователи могут неохотно загружать и устанавливать подключаемый модуль VPN из интернет-магазина приложений вследствие предполагаемых угроз этого типа продукта. Соответственно могут быть затруднены торговля и распространение продуктов, которые содержат такие преобразующие драйверы.

## СУЩНОСТЬ ИЗОБРЕТЕНИЯ

[0002] В этом документе описывается ограниченная платформа преобразующих драйверов. В одной или нескольких реализациях предоставляется платформа, которая обеспечивает ограниченную среду исполнения для драйверов виртуальной частной сети (VPN) и других преобразующих драйверов. Платформу можно реализовать в виде компонента операционной системы, который предоставляет интерфейс, посредством которого драйверы можно регистрировать в платформе и вызывать для выполнения поддерживаемых платформой функций. Ограниченная среда исполнения накладывает одно или несколько ограничений на преобразующие драйверы, которые работают посредством той платформы. Например, исполнение может происходить в пользовательском режиме для каждого пользователя в отдельности и в рамках песочницы. Кроме того, платформа побуждает ассоциированные драйверы запускаться в фоновых процессах со сравнительно небольшими привилегиями. Кроме того, платформа может приостанавливать драйверы и управлять операциями драйверов путем планирования фоновых задач. Соответственно посредством платформы управляется и ограничивается воздействие преобразующих драйверов на систему.

[0003] Данное краткое изложение сущности изобретения предоставляется, чтобы представить подборку идей в упрощенном виде, которые дополнительно описываются ниже в подробном описании. Данное краткое изложение сущности изобретения не предназначено ни для определения ключевых признаков или существенных признаков заявленного изобретения, ни для использования в качестве содействия в определении объема заявленного изобретения.

## КРАТКОЕ ОПИСАНИЕ ЧЕРТЕЖЕЙ

[0004] Подробное описание приводится со ссылкой на прилагаемые фигуры. На чертежах крайняя левая цифра(цифры) в номере ссылки определяет чертеж, в котором номер ссылки появляется первый раз. Использование одних и тех же номеров ссылок в разных случаях в описании и на чертежах может указывать аналогичные или идентичные элементы.

[0005] Фиг. 1 – иллюстрация среды в соответствии с одной или несколькими реализациями методик ограниченной платформы преобразующих драйверов.

[0006] Фиг. 2 – иллюстрация примерного сценария в соответствии с одной или несколькими реализациями.

[0007] Фиг. 3 – логическая блок-схема, изображающая примерную процедуру для применения ограниченной среды исполнения для исполнения преобразующего драйвера.

[0008] Фиг. 4 – логическая блок-схема, изображающая примерную процедуру, в которой управляют работой преобразующего драйвера.

[0009] Фиг. 5 – логическая блок-схема, изображающая примерную процедуру, в которой преобразующие драйверы распространяются через интернет-магазин приложений.

[0010] Фиг. 6 изображает примерную вычислительную систему в соответствии с одним или несколькими вариантами осуществления.

## ПОДРОБНОЕ ОПИСАНИЕ

### ОБЗОР

[0011] Частично из-за вопросов безопасности с драйверами виртуальной частной сети (VPN) и другими преобразующими драйверами пользователи могут с большой неохотой устанавливать на свои устройства продукты, которые применяют преобразующие драйверы. Соответственно могут быть затруднены торговля и распространение продуктов, которые содержат такие преобразующие драйверы.

[0012] В этом документе описывается ограниченная платформа преобразующих драйверов. В одной или нескольких реализациях предоставляется платформа, которая обеспечивает ограниченную среду исполнения для драйверов виртуальной частной сети (VPN) и других преобразующих драйверов. Платформу можно реализовать в виде компонента операционной системы, который предоставляет интерфейс, посредством которого драйверы можно регистрировать в платформе и вызывать для выполнения поддерживаемых платформой функций. Ограниченная среда исполнения накладывает одно или несколько ограничений на преобразующие драйверы, которые работают посредством той платформы. Например, исполнение может происходить в пользовательском режиме для каждого пользователя в отдельности и в рамках песочницы. Кроме того, платформа побуждает ассоциированные драйверы запускаться в виде фоновых процессов со сравнительно небольшими привилегиями. Кроме того, платформа может приостанавливать драйверы и управлять операциями драйвера путем планирования фоновых задач. Соответственно посредством платформы управляется и ограничивается воздействие преобразующих драйверов на систему.

[0013] В нижеследующем обсуждении сначала описывается примерная операционная среда, которая может применять описанные в этом документе методики. Далее описываются примерные подробности и методики, которые можно реализовать в примерной среде, а также в других средах. Следовательно, выполнение методик не ограничивается примерной средой, и примерная среда не ограничивается выполнением примерных методик. Наконец, описываются примерные системы и устройства, которые могут применяться для реализации одного или нескольких вариантов осуществления.

### ПРИМЕРНАЯ ОПЕРАЦИОННАЯ СРЕДА

[0014] Фиг. 1 – иллюстрация среды 100 в примерной реализации, которая функционирует для применения описанных в этом документе методик. Проиллюстрированная среда 100 включает в себя клиентское устройство 102, которое коммуникационно соединено с поставщиком 106 услуг по сети 104. Поставщик 106 услуг может конфигурироваться для предоставления различных ресурсов 108 (например, контента и услуг) клиентскому устройству 102 и другим клиентам по сети 104. Как правило, ресурсы 108, сделанные поставщиком 106 услуг доступными, могут включать в себя любое подходящее сочетание услуг и/или контента, обычно предоставляемых по сети одним или несколькими поставщиками. Некоторые примеры услуг включают в себя, но не ограничиваются, услугу поиска, услугу электронной почты, услугу обмена мгновенными сообщениями, комплекс онлайн-приложений для продуктивной работы и услугу аутентификации для управления доступом клиентов к ресурсам. Контент может включать в себя различные сочетания текста, мультимедийных потоков, документов,

файлов приложений, фотографий, аудио/видеофайлов, анимаций, изображений, веб-страниц, веб-приложений, приложений для устройств, контента для отображения обозревателем или иным клиентским приложением и т. п.

[0015] Клиентское устройство 102 и поставщика 106 услуг можно реализовать с помощью одного или нескольких вычислительных устройств, а также они могут соответствовать одному или нескольким объектам. Вычислительное устройство может конфигурироваться различными способами. Например, вычислительное устройство может конфигурироваться в виде компьютера, который допускает осуществление связи по сети, такого как настольный компьютер, мобильная станция, развлекательное устройство, телевизионная приставка, коммуникационно соединенные с устройством отображения, беспроводной телефон, игровая приставка и так далее. Таким образом, вычислительное устройство может варьироваться от полноценных устройств со значительными ресурсами памяти и процессора (например, персональные компьютеры, игровые приставки) до ограниченного устройства с ограниченными ресурсами памяти и/или обработки (например, традиционные телевизионные приставки, наладонные игровые приставки). Более того, хотя в некоторых случаях показано одно вычислительное устройство, вычислительное устройство может соответствовать множеству разных устройств, например нескольким серверам, используемым поставщиком 106 услуг.

[0016] Клиентское устройство 102 дополнительно иллюстрируется как включающее в себя операционную систему 110. Операционная система 110 конфигурируется для обобщения лежащих в основе функциональных возможностей аппаратных средств для приложений 112, которые исполняются на клиентском устройстве 102. Например, операционная система 110 может обобщать функциональные возможности обработки, запоминающего устройства, сети и/или дисплея, так что приложения 112 можно писать без знаний о том, "как" реализуются эти лежащие в основе функциональные возможности. Например, приложения 112 могут предоставлять операционной системе 110 данные для обработки и отображения с помощью устройства отображения без понимания того, как будет выполняться эта обработка.

[0017] В соответствии с описанными в этом документе методиками клиентское устройство 102 также иллюстрируется как включающее в себя преобразующие драйверы 114 и платформу 116 безопасности драйверов, которая может конфигурироваться различными способами, чтобы накладывать ограничения на работу преобразующих драйверов 114. Преобразующие драйверы 114 соответствуют различным видам модулей приложений, подключаемых модулей и/или других сторонних программ, которыми можно управлять с помощью описанной в этом документе платформы 116 безопасности драйверов. Как правило, преобразующие драйверы 114 осуществляют преобразования лежащих в основе данных из одного вида в другой. Одним примерным типом преобразующих драйверов являются драйверы виртуальной частной сети (VPN), которые преобразуют пакеты данных (например, инкапсулируют/декапсулируют) в закрытый (собственный) вид для защищенной связи VPN между VPN-клиентом и VPN-сервером. Другие примеры могут включать в себя, но не ограничиваются, драйверы сетевой интерфейсной платы для пакетирования/депакетирования данных, драйверы антивируса для распознавания и обработки вредоносного ПО, драйверы графической обработки для обработки/преобразования графики, драйверы принтера для перевода данных приложений в пригодный для печати формат для принтера, и так далее. Обсуждаемые в этом документе методики также применимы к туннелям VPN и другим туннелям Интернет-протокола (IP), конвейерам данных, транспортным потокам, брандмауэрам

и так далее.

[0018] В традиционной модели преобразующие драйверы 114 могут обладать общесистемными привилегиями, высоким приоритетом и доступом к конфиденциальным данным. Как правило, преобразующие драйверы тесно связаны с операционной системой и запускаются в режиме ядра. Таким образом, преобразующие драйверы 114 могут включать в себя и/или называться драйверами режима ядра, системными драйверами и/или драйверами класса. Поскольку преобразующие драйверы 114 обычно обладают повышенными привилегиями и полным доступом к системе в режиме ядра, эти виды драйверов создают значительную угрозу безопасности и могут становиться целями/инструментами, применяемыми злоумышленниками с противозаконными целями. Преобразующие драйверы 114, которые свободно работают с неограниченными привилегиями и доступом, также могут отрицательно влиять на ресурсы обработки и время работы от батарей. Кроме того, распространение преобразующих драйверов через магазин приложений или другой онлайн-канал может быть затруднено, по меньшей мере частично, из-за только что перечисленных вопросов.

[0019] Однако в соответствии с патентоспособными принципами, описанными в этом документе, можно создать ограниченную среду исполнения, чтобы сделать возможным ограниченное исполнение преобразующих драйверов 114 в пользовательском режиме, что частично снимает вышеупомянутые вопросы. Например, путем управления преобразующими драйверами 114 в ограниченной среде исполнения, которая обсуждается выше и ниже, создается безопасность в работе драйверов, что эффективно для повышения надежности системы, уменьшения потребления ресурсов в операциях драйверов, предотвращения эксплуатации системы посредством драйверов, обеспечения возможности установок для каждого пользователя и, соответственно, приведения распространения драйверов через Интернет-каналы к практичному варианту.

[0020] В частности, платформа 116 безопасности драйверов представляет функциональные возможности клиентского устройства 102, чтобы создать экземпляр ограниченной среды исполнения для преобразующих драйверов 114. Функциональные возможности, представленные платформой 116 безопасности драйверов, можно реализовать различными способами. В некоторых реализациях платформу 116 безопасности драйверов можно предоставить в виде компонента операционной системы, однако платформу 116 безопасности драйверов также можно реализовать в виде автономного компонента, как проиллюстрировано. Как изображено, платформа 116 безопасности драйверов может включать в себя или иным образом состоять из модуля 118 диспетчера драйверов, контейнера 120 приложения, диспетчера 122 фоновой работы и брокера 124 событий для реализации различных аспектов платформы.

[0021] Модуль 118 диспетчера драйверов представляет функциональные возможности платформы для распознавания и управления преобразующими драйверами 114, ассоциированными с платформой. Модуль 118 диспетчера драйверов может управлять исполнением преобразующих драйверов 114 в пользовательском режиме для каждого пользователя посредством контейнера 120 приложения или "песочницы", созданной для каждого драйвера. Контейнер 120 приложения задает ограниченный набор задач, которые могут выполняться для отдельных драйверов, и ограничивает операции заданным набором задач. Контейнер 120 приложения дополнительно обеспечивает изоляцию в песочнице "с двойными стенками", означающую, что доступ ограничивается как изнутри контейнера в систему, так и от внешних объектов в контейнер. Это не позволяет внешним объектам перехватывать и/или получать управление над драйверами в песочнице в контейнерах приложений.

[0022] Диспетчер 122 фоновой работы функционирует для управления контейнерами 120 приложений (и ассоциированными драйверами) с использованием фоновых процессов. Это может включать в себя, но не ограничивается, создание экземпляра фоновых процессов для контейнеров и проведение планирования задач для фоновых процессов. Фоновым процессам можно назначить маркеры небольших привилегий или сопоставимые управляющие данные, которые передают сравнительно мало привилегий и прав доступа. Другими словами, платформа 116 безопасности драйверов ассоциирует сравнительно низкий приоритет с преобразующими драйверами 114, что ограничивает доступ и/или свободу действия. Диспетчер 122 фоновой работы также может вызвать приостановку фоновых процессов для преобразующих драйверов 114, когда они не используются, что дополнительно ограничивает возможность преобразующих драйверов выполнять работу помимо конкретного указания от пользователя и/или платформы 116 безопасности драйверов. Например, можно приостанавливать преобразующий драйвер 114 кроме случаев, когда набор задач, специально заданный для того драйвера или типа драйвера, выполняется по указанию модуля 118 диспетчера драйверов.

[0023] Диспетчер 122 фоновой работы можно реализовать, как проиллюстрировано, в виде компонента платформы 116 безопасности драйверов. В качестве альтернативы диспетчер 122 фоновой работы может конфигурироваться в виде компонента операционной системы (ОС), спроектированного для управления и координации операций множества приложений 112 с использованием фоновых процессов, включая преобразующие драйверы 114. При этом подходе платформа 116 безопасности драйверов может конфигурироваться для взаимодействия с диспетчером 122 фоновой работы, чтобы воспользоваться встроенными функциональными возможностями ОС по поддержке фоновой работы приложений.

[0024] Брокер 124 событий представляет функциональные возможности для формирования событий, чтобы урегулировать взаимодействие между драйвером и системными службами для выполнения обозначенных задач, которые разрешены по отношению к преобразующему драйверу и соответствующему контейнеру 120 приложения. В частности, брокер 124 событий может создавать события, которые побуждают выполнение задач, которые планируются диспетчером 122 фоновой работы. В некоторых реализациях это включает в себя формирование подходящих вызовов точек входа в код, например программных интерфейсов приложения (API) системы или других подходящих интерфейсов. Брокер 124 событий может дополнительно конфигурироваться для создания событий, чтобы урегулировать взаимодействия с элементами интерфейса пользователя, показанными вычислительным устройством, например экраном дисплея и UI, ассоциированными с ОС, обозревателем, приложением VPN-клиента и/или другими приложениями. Преобразующий драйвер, который "помещен в песочницу" описанным способом, можно ограничить в прямом манипулировании компонентами и элементами UI. Соответственно брокер 124 событий может работать в качестве заменителя, который делает возможным взаимодействие и координацию с компонентами UI от лица преобразующих драйверов 114 без чрезмерного нарушения безопасности.

[0025] Как дополнительно проиллюстрировано на фиг. 1, поставщик 106 услуг может конфигурироваться для управления и предоставления клиентского доступа к услуге 126 магазина приложений. Услуга 126 магазина приложений представляет конкретный ресурс из ресурсов 108, который может предоставляться клиентам поставщиком 106 услуг по сети 104. Услуга 126 магазина приложений конфигурируется для разрешения пользовательского доступа к онлайн-базе данных приложений (например, к торговой

площадке приложений) для просмотра, выбора, покупки и/или загрузки приложений. Приложения от различных разработчиков могут быть доступны клиентам посредством услуги 126 магазина приложений.

[0026] В соответствии с обсуждаемыми в этом документе методиками услуга 126 магазина приложений может выбираться в качестве механизма распространения для преобразующих драйверов 114, которые управляются через платформу 116 безопасности драйверов. Это становится возможным благодаря дополнительным мерам безопасности, ограничениям и улучшениям характеристик, которые достигаются с использованием ограниченной среды(сред) исполнения, созданной для драйвера посредством платформы 116 безопасности драйверов. В одном подходе преобразующие драйверы 114 можно упаковать в комплект для включения в торговую площадку приложений в назначенном известном формате, ассоциированном с платформой 116 безопасности драйверов. Комплект 128 драйвера, содержащий соответствующий преобразующий драйвер 114 в назначенном формате, можно сделать доступным посредством услуги 126 магазина приложений и можно загружать для использования клиентским устройством 102 по сети 104, как представлено на фиг. 1. В одном подходе комплект 128 драйвера может применять закрытый формат (например, .appx или другой сопоставимый формат, специфический для магазина или для поставщика), ассоциированный с услугой 126 магазина приложений. Дополнительно или в качестве альтернативы комплект может конфигурироваться содержащим идентификатор, ключ, код, расширение файла или другие подходящие идентифицирующие данные, чтобы упростить распознавание комплекта 128 драйвера как содержащего преобразующий драйвер, который поддерживается платформой 116 безопасности драйверов.

[0027] Во время установки платформа 116 безопасности драйверов посредством модуля 118 диспетчера драйверов или иным образом может распознать комплект 128 драйвера и/или формат как ассоциируемый с платформой 116 безопасности драйверов на основе подходящих идентифицирующих данных, включенных в пакет. Соответственно в ответ на установку комплекта 128 драйвера в распознанном формате модуль 118 диспетчера драйверов может зарегистрировать соответствующий преобразующий драйвер в платформе и посредством этого привести в исполнение ограничения и дополнительную безопасность для драйвера, доступные посредством платформы 116 безопасности драйверов.

[0028] Дополнительные подробности касательно этих и других аспектов ограниченной платформы преобразующих драйверов обсуждаются в отношении примерного сценария использования, изображенного на фиг. 2 в целом по ссылке 200. В этом примере контейнер 120 приложения или песочница иллюстрируется как содержащий (содержащая) драйвер 202 виртуальной частной сети (VPN), хотя, как отмечается в этом документе, также предполагаются другие драйверы, подключаемые модули и код. Контейнер 120 приложения конфигурируется для разрешения ограниченного набора задач для содержащегося драйвера, который в этом случае оказывается драйвером 202 VPN. В этом сценарии VPN разрешенные задачи можно ограничить операциями подключения, отключения, инкапсуляции и/или декапсуляции, ассоциированными со связью VPN с VPN-сервером по сети. Можно ассоциировать другие типы драйверов и код с соответствующими операциями, которые разрешены платформой, для поддержки их базовых функциональных возможностей, минимизируя при этом доступ и влияние на производительность. Допустимый набор задач может ассоциироваться с каждым типом драйвера и/или по каждому отдельному драйверу. Как отмечалось, контейнер 120 приложения эффективно ограничивает драйвер конкретным набором задач,

разрешенным для драйвера посредством платформы.

[0029] Фиг. 2 также представляет распределение различных компонентов между пользовательским режимом и режимом ядра. Как правило, система обработки в вычислительном устройстве может переключаться между пользовательским режимом и режимом ядра в зависимости от типа кода, который исполняется. Как правило, настольные приложения запускаются в пользовательском режиме, а базовые компоненты операционной системы запускаются в режиме ядра. Драйверы/код, которые работают в режиме ядра, могут совместно использовать распределение общей памяти и не изолированы от других драйверов/кода или ресурсов ОС. Как отмечалось, преобразующие драйверы традиционно реализуются в виде драйверов режима ядра, которые обладают практически неограниченным доступом к ОС и возможностью вызывать общесистемные отказы. Тем не менее, в соответствии с описанными в этом документе методиками преобразующие драйверы вместо этого могут исполняться в пользовательском режиме посредством подходящей ограниченной среды исполнения.

[0030] В частности, контейнер 120 приложения и содержащийся в нем драйвер 202 VPN изображаются на фиг. 2 как исполняемые в пользовательском режиме. Драйвер 202 VPN может взаимодействовать с базовыми службами и функциональными возможностями операционной системы 110, которые предоставляются ядром 206, через интерфейс 204 драйвера. Ядро 206 в целом реализует программное обобщение лежащих в основе аппаратных средств 208, чтобы упростить работу аппаратных средств с помощью программных компонентов. Интерфейс 204 драйвера может представлять один или несколько программных интерфейсов приложения (API) или других подходящих интерфейсов, ассоциированных с операционной системой 110, через которые базовые службы и функциональные возможности, представленные ядром 206, становятся доступны коду, исполняемому в пользовательском режиме.

[0031] В конкретном примере базовые службы и функциональные возможности включают в себя по меньшей мере службы 210 VPN для упрощения связи VPN. Службы 210 VPN могут соответствовать разрешенному набору задач для драйвера 202 VPN, который может конфигурироваться для вызова служб через подходящие вызовы и обратные вызовы, поддерживаемые интерфейсом драйвера. Можно создать экземпляр интерфейса 204 драйвера в контейнере 120 приложения, как изображено, или в виде автономного компонента. Функциональные возможности, представленные интерфейсом 204 драйвера, также можно разделить между компонентом пользовательского режима и соответствующим компонентом ядра 206, который работает в режиме ядра. Взаимодействиями через интерфейс 204 драйвера можно управлять посредством платформы безопасности драйверов через модуль 118 диспетчера драйверов или иным образом.

[0032] В ходе работы диспетчер 122 фоновой работы может установить фоновый процесс 211 для контейнера 120 приложения и управлять исполнением драйвера 202 VPN с использованием фонового процесса 211. Задачи могут планироваться по указанию платформы. Как правило, это происходит по явному указанию пользователя (например, выбор пользователя или согласие пользователя) или в соответствии с профилем, созданным на основе пользовательского ввода. Профиль может задавать набор задач, доступный драйверу 202 VPN в контейнере 120 приложения, и соответствующим образом устанавливать ассоциированные привилегии и права доступа. Затем платформа принудительно применяет ограничения, указанные профилем, с использованием контейнера 120 приложения. Профиль, например, может указывать, что согласие пользователя нужно получать каждый раз, когда выполняются конкретные операции

(например, для каждого случая операции), и в этом случае может формироваться уведомление и приглашение для согласия пользователя, чтобы выборочно управлять конкретными операциями. Профиль также может указывать глобальное согласие пользователя или согласие пользователя по умолчанию, чтобы позволять некоторым избранным операциям совершаться автоматически без дополнительного указания согласия для каждого случая. При отсутствии согласия пользователя контейнер 120 приложения препятствует выполнению ограниченных задач драйвером. Другими словами, ограниченная среда исполнения конфигурируется для выборочного разрешения операций преобразующих драйверов в зависимости от согласия пользователя. Таким образом, пользователь обеспечивается окончательным контролем над областью взаимодействий, разрешенных для преобразующих драйверов 114, которые управляются через платформу.

[0033] Фиг. 2 дополнительно иллюстрирует взаимодействие контейнера 120 приложения с брокером 124 событий. В частности, брокер 124 событий может создать события 212 VPN и/или события 214 интерфейса пользователя от лица драйвера VPN, что побуждает выполнение задач, запланированных диспетчером 122 фоновой работы. Например, события 212 VPN могут включать в себя вызовы, направленные на интерфейс 204 драйвера, чтобы вызвать службы 210 VPN. Более того, брокер 124 событий может создать события 214 интерфейса пользователя, чтобы манипулировать компонентами UI для операционной системы 110, приложения 112 или другого UI. Это может включать в себя, но не ограничивается, вывод уведомлений, чтобы получить согласие пользователя для запланированных задач в подходящих обстоятельствах.

[0034] Таким образом, ограниченную среду исполнения можно реализовать для управления работой драйвера 202 VPN и/или других преобразующих драйверов, традиционно исполняемых в режиме ядра. Такие драйверы, как правило, принимают входные данные 216 в конкретном виде и обрабатывают данные для создания преобразованных данных 218 в ином виде, как представлено на фиг. 2. Например драйвер 202 VPN из фиг. 2 может инкапсулировать пакеты данных в закрытый формат для защищенной связи VPN с учрежденческим сервером. Драйвер 202 VPN также может конфигурироваться для декапсуляции пакетов данных, полученных от учрежденческого сервера в закрытом формате, в данные, потребляемые клиентским устройством 102. Другие типы драйверов конфигурируются для выполнения сопоставимых преобразований данных, связанных с задуманными функциональными возможностями драйверов. Например драйвер принтера преобразует данные для печати, драйвер сетевой интерфейсной платы работает для пакетирования и депакетирования данных, драйвер антивируса может сканировать файлы и сравнивать файлы с известными сигнатурами, и так далее.

[0035] Таким образом, можно предоставить платформу безопасности драйверов, которая обеспечивает ограниченную среду исполнения, сконфигурированную для принудительного применения различных ограничений для драйверов VPN, а также других преобразующих драйверов. Платформу можно реализовать в виде компонента операционной системы, который предоставляет интерфейс, посредством которого драйверы можно регистрировать в платформе и вызывать для выполнения задач, поддерживаемых и/или разрешенных платформой. Преобразующие драйверы могут исполняться платформой в пользовательском режиме, для каждого пользователя и в контейнере приложения "в песочнице", который ограничивает привилегии и доступ к системе у преобразующих драйверов.

[0036] Приняв во внимание предшествующее обсуждение примерной операционной

среды, рассмотрим теперь подробности касательно методик для ограниченной платформы преобразующих драйверов, описанные в отношении следующих примерных процедур.

ПРИМЕРНЫЕ ПРОЦЕДУРЫ

[0037] Этот раздел обсуждает подробности методик для ограниченной платформы преобразующих драйверов со ссылкой на примерные процедуры из фиг. 3-5. В частях нижеследующего обсуждения можно ссылаться на примерную операционную среду из фиг. 1, в которой можно реализовать различные аспекты. Аспекты каждой из описанных ниже процедур можно реализовать в аппаратных средствах, микропрограммном обеспечении или программном обеспечении, или в их сочетании. Процедуры показаны в виде набора этапов, которые задают операции, выполняемые одним или несколькими устройствами, и они не обязательно ограничены очередностями, показанными для выполнения операций с помощью соответствующих этапов. По меньшей мере в некоторой реализации процедуры могут выполняться соответствующим образом сконфигурированным вычислительным устройством, например, примерным клиентским устройством 102 из фиг. 1, которое включает в себя или использует платформу 116 безопасности драйверов или сопоставимые функциональные возможности.

[0038] Фиг. 3 – логическая блок-схема, изображающая примерную 300 процедуру для применения ограниченной среды исполнения для исполнения преобразующего драйвера. Преобразующий драйвер регистрируется в платформе безопасности драйверов (этап 302). Например модуль 118 диспетчера драйверов может конфигурироваться для распознавания драйверов, которые упакованы в комплект в назначенном формате. Комплект может включать в себя, например, подходящий идентификатор для указания платформе безопасности драйверов, что драйвер предназначен для этой платформы и должен управляться посредством платформы. Таким образом, при установке драйвера модуль 118 диспетчера драйверов обнаруживает комплект и/или идентификатор и регистрирует драйвер для управления посредством системы. Впоследствии операции драйвера управляются платформой безопасности драйверов с соответствующими ограничениями.

[0039] Создается экземпляр ограниченной среды исполнения для преобразующего драйвера (этап 304), и преобразующий драйвер вызывается для выполнения одной или нескольких задач (этап 306). Затем преобразующий драйвер исполняется посредством ограниченной среды исполнения, чтобы выполнить задачи (этап 308). Здесь модуль 118 диспетчера драйверов может создать ограниченную среду исполнения для конкретного драйвера, когда исполняется тот драйвер. Ограниченная среда исполнения может содержать контейнер 120 приложения (например, "песочницу"), как обсуждалось ранее. Ограниченная среда исполнения накладывает различные ограничения на работу преобразующего драйвера, которые применяются во время исполнения. Например ограниченная среда исполнения изолирует соответствующий драйвер, чтобы ограничить доступ к ресурсам системы. В частности, можно ограничить доступ к назначенному набору задач, ассоциированному с функциональными возможностями драйвера. Драйвер вызывается системой для выполнения задач и может быть неспособен самостоятельно инициировать задачи. Например ограниченная среда исполнения может препятствовать самостоятельному созданию объектов драйвером. Вместо этого объекты для конкретных задач создаются платформой и передаются в ограниченную среду исполнения для использования драйвером.

[0040] Драйвер также может исполняться посредством ограниченной среды исполнения в виде фоновой задачи, и ему можно назначить маркер привилегий, который

передает ограниченные привилегии и права доступа. Фоновый процесс, ассоциированный с ограниченной средой исполнения, можно приостанавливать при отсутствии конкретного указания к выполнению работы от платформы и/или пользователя. Более того, ограниченная среда исполнения отклоняет приложения и другие компоненты, которые не ассоциируются с доступом платформы безопасности драйверов к преобразующему драйверу, чтобы предотвратить подделку и/или перехват драйвера.

[0041] Более того, драйвер исполняется в пользовательском режиме, как обсуждалось ранее, и также может быть установлен для каждого пользователя в отдельности. Соответственно разные пользователи могут устанавливать разные драйверы для использования на одном и том же клиентском устройстве и/или с одной и той же целью. Это может упростить использование разных аппаратных средств или разной версии программного обеспечения в соответствии с предпочтениями пользователя. Поскольку драйверы находятся в песочнице в пользовательском режиме, установка и удаление одного драйвера не влияет на восприятие других пользователей. Например, что касается технологии VPN, разные пользователи одного и того же устройства могли бы применять разные драйверы VPN для подключения к соответствующим VPN-серверам разных работодателей без неблагоприятного воздействия друг на друга.

[0042] Кроме того, реализованные с использованием описанных методик драйверы можно легко и полностью удалить без остаточных файлов, данных, настроек и свойств, которые могут вызывать нестабильность или непредусмотренное поведение. Это возможно потому, что ограниченная среда исполнения обеспечивает индивидуальное распределение памяти для исключительного использования соответствующим драйвером. Драйвер может не иметь доступа для записи данных или настроек куда-то в другое место. Соответственно очистка драйвера и состояния легко выполняется путем удаления данных, содержащихся в распределенной памяти, ассоциированной с драйвером.

[0043] Фиг. 4 – логическая блок-схема, изображающая примерную процедуру, в которой управляется работа преобразующего драйвера. Распознается ассоциация преобразующего драйвера с платформой безопасности драйверов (этап 402). Например, преобразующий драйвер может быть зарегистрирован в платформе 116 безопасности драйверов, как обсуждалось ранее. Работа драйвера впоследствии управляется через платформу 116 безопасности драйверов. Ассоциацию можно создать во время установки на основе драйвера, конфигурируемого в назначенном формате или комплекте пакете. Как отмечалось, формат .appx является подходящим форматом, который может использоваться для ассоциации драйверов с платформой, хотя также предполагаются другие типы файлов, имена и идентифицирующая информация.

[0044] Выявляется набор задач, разрешенный для преобразующего драйвера (этап 404). Разрешенный набор задач может основываться на идентификации драйвера как конкретного типа драйвера (например, VPN в отличие от принтера, антивируса и т. п.) и/или индивидуальной идентификации каждого конкретного драйвера (например, различие между драйверами одного типа, ассоциированными с разными производителями/поставщиками). Поэтому платформа может поддерживать и разрешать назначенный набор задач для каждого отдельного драйвера и/или на основе типа драйвера. Набор задач может быть конкретными контрактами, которые операционная система выставила и разрешает выполнить драйверам. Таким образом, при идентификации заданного драйвера платформа узнает виды задач, которые драйвер способен выполнять, разрешает те задачи и может препятствовать выполнению драйвером других задач.

[0045] В частности создается среда исполнения, которая ограничивает работу преобразующего драйвера набором задач, которые разрешены (этап 406). Например можно создать ограниченную среду исполнения ранее описанным способом. Ограниченная среда исполнения разрешает некоторые задачи или контракты, а в противном случае препятствует обращению драйвера к ресурсам операционной системы или выполнению задач, выполнение которых драйвером явно не авторизовано платформой безопасности драйверов.

[0046] Затем работа преобразующего драйвера управляется через среду исполнения, включая вызов преобразующего драйвера для выполнения набора задач, которые разрешены, и приостановку преобразующего драйвера при отсутствии вызова от платформы безопасности драйверов (этап 408). Снова преобразующий драйвер, исполняемый посредством ограниченной среды исполнения, вызывается платформой, но не способен самостоятельно инициировать задачи. Драйвер можно оставить в состоянии приостановки, когда он не вызывается интенсивно для выполнения задач. Это может происходить посредством фонового процесса 211, который возобновляют для работы, а затем возвращают в состояние приостановки после завершения той работы. Таким образом, выполняемая драйвером работа совершается по указанию платформы и/или пользователя. Как правило, драйвер не способен выполнять задачи помимо задач, назначенных платформой для драйвера. Драйвер также не способен работать, пока не вызван платформой, что предотвращает свободное потребление драйверами ресурсов системы и времени работы от батарей.

[0047] Фиг. 5 – логическая блок-схема, изображающая примерную процедуру, в которой преобразующие драйверы распространяются через интернет-магазин приложений. Задается формат комплекта приложения, который распознается платформой безопасности драйверов, для включения подключаемых модулей в интернет-магазин приложений (этап 502). Например подключаемые модули, такие как преобразующие драйверы 114 и другой код от сторонних поставщиков, могут кодироваться в комплект 128 драйвера, который применяет назначенный формат, заданный для платформы 116 безопасности драйверов. Разработчики могут использовать назначенный формат, чтобы воспользоваться платформой 116 безопасности драйверов и сигнализировать платформе, чтобы та обрабатывала их код/драйверы соответствующим образом посредством платформы. Комплект 128 драйвера может конфигурироваться для упрощения включения в торговую площадку приложений, например посредством услуги 126 магазина приложений, используя формат .appx или другой назначенный формат.

[0048] Предоставляется доступ для загрузки подключаемых модулей через интернет-магазин приложений (этап 504). Затем подключаемые модули распространяются клиентам в заданном формате, чтобы побудить клиентов регистрировать подключаемые модули в платформе безопасности драйверов и реализовать ограниченную среду исполнения, чтобы управлять работой подключаемых модулей (этап 506). Включение преобразующих драйверов 114 в виде предложений в торговую площадку приложений целесообразно отчасти из-за дополнительных мер безопасности, ограничений и улучшений характеристик, достигаемых посредством платформы 116 безопасности драйверов, что делает более вероятной ситуацию, что пользователи будут загружать и устанавливать такие драйверы из интернет-магазина. Таким образом, комплект 128 драйвера, содержащий соответствующий преобразующий драйвер 114 в назначенном формате, можно сделать доступным посредством услуги 126 магазина приложений. Комплект 128 драйвера может содержать подходящие идентифицирующие данные для

упрощения распознавания этого комплекта платформой 116 безопасности драйверов. Платформа 116 безопасности драйверов конфигурируется для распознавания комплекта 128 драйвера при установке и для выполнения операций, которые описаны в этом документе, чтобы зарегистрировать драйвер, создать ограниченную среду исполнения, принудительно применить различные ограничения, наложенные на драйвер, и так далее.

[0049] Приняв во внимание некоторые примерные процедуры, обсудим теперь примерную систему и устройство для реализации различных аспектов в соответствии с одним или несколькими вариантами осуществления.

## ПРИМЕРНАЯ СИСТЕМА И УСТРОЙСТВО

[0050] Фиг. 6 иллюстрирует примерную систему 600, которая включает в себя примерное вычислительное устройство 602, которое соответствует одной или нескольким вычислительным системам и/или устройствам, которые могут реализовывать различные, описанные в этом документе методики. Вычислительное устройство 602 может быть, например, сервером поставщика услуг, устройством, ассоциированным с клиентом (например, клиентским устройством), системой на кристалле и/или любым другим подходящим вычислительным устройством или вычислительной системой.

[0051] Примерное вычислительное устройство 602, которое проиллюстрировано, включает в себя систему 604 обработки, один или несколько машиночитаемых носителей 606 и один или несколько интерфейсов 608 I/O, которые коммуникационно соединены друг с другом. Хотя и не показано, вычислительное устройство 602 может дополнительно включать в себя системную шину или другую систему передачи данных и команд, которая соединяет друг с другом различные компоненты. Системная шина может включать в себя любую структуру или сочетание разных шинных структур, например шину памяти либо контроллер памяти, периферийную шину, универсальную последовательную шину и/или процессор либо локальную шину, которая использует любую из ряда шинных архитектур. Также предполагается ряд других примеров, например линия управления и передачи данных.

[0052] Система 604 обработки соответствует функциональным возможностям для выполнения одной или нескольких операций с использованием аппаратных средств. Соответственно система 604 обработки иллюстрируется как включающая в себя элементы 610 аппаратных средств, которые могут конфигурироваться в виде процессоров, функциональных блоков и так далее. Это может включать в себя реализацию в аппаратных средствах в виде специализированной интегральной схемы или другого логического устройства, образованного с использованием одного или нескольких полупроводников. Элементы 610 аппаратных средств не ограничиваются материалами, из которых они образованы, или используемыми в них механизмами обработки. Например процессоры могут состоять из полупроводника (полупроводников) и/или транзисторов (например, электронных интегральных схем (IC)). В таком контексте исполняемыми процессором командами могут быть исполняемые в электронном виде команды.

[0053] Машиночитаемые носители 606 иллюстрируются как включающие в себя запоминающее устройство/память 612. Запоминающее устройство/память 612 представляет объем запоминающего устройства/памяти, ассоциированный с одним или несколькими машиночитаемыми носителями. Запоминающее устройство/память 612 может включать в себя энергозависимые носители (например, оперативное запоминающее устройство (RAM)) и/или энергонезависимые носители (например, постоянное запоминающее устройство (ROM), флэш-память, оптические диски, магнитные диски и так далее). Запоминающее устройство/память 612 может включать

в себя несъемные носители (например, RAM, ROM, несъемный жесткий диск и так далее), а также съемные носители (например, флэш-память, съемный жесткий диск, оптический диск и так далее). Машиночитаемые носители 606 можно конфигурировать различными другими способами, как дополнительно описано ниже.

[0054] Интерфейс (интерфейсы) 608 ввода/вывода соответствуют функциональным возможностям для предоставления пользователю возможности вводить команды и информацию в вычислительное устройство 602, а также позволяют представлять информацию пользователю и/или другим компонентам либо устройствам с использованием различных устройств ввода/вывода. Примеры устройств ввода включают в себя клавиатуру, устройство управления курсором (например, мышь), микрофон для голосовых операций, сканер, сенсорные функциональные возможности (например, емкостные или другие датчики, которые конфигурируются для обнаружения физического касания), камеру (например, которая может применять длины волн в видимой или невидимой части спектра, например инфракрасные частоты, чтобы обнаруживать перемещение, которое не предполагает касание, а также жесты), и так далее. Примеры устройств вывода включают в себя устройство отображения (например, монитор или проектор), динамики, принтер, сетевую карту, устройство с тактильным откликом и так далее. Таким образом, вычислительное устройство 602 может конфигурироваться различными способами, как дополнительно описано ниже, чтобы поддерживать взаимодействие с пользователем.

[0055] Различные методики могут описываться в этом документе в общем контексте программного обеспечения, элементов аппаратных средств или программных модулей. Как правило, такие модули включают в себя процедуры, программы, объекты, элементы, компоненты, структуры данных и так далее, которые выполняют конкретные задачи или реализуют конкретные абстрактные типы данных. Термины "модуль", "функциональные возможности" и "компонент" при использовании в данном документе в целом представляют программное обеспечение, микропрограммное обеспечение, аппаратные средства или их сочетание. Признаки описанных в этом документе методик являются платформо-независимыми, то есть методики можно реализовать на различных промышленных вычислительных платформах, имеющих различные процессоры.

[0056] Реализация описанных модулей и методик может храниться или передаваться посредством некоторого вида машиночитаемых носителей. Машиночитаемые носители могут включать в себя ряд носителей, к которым можно обращаться с помощью вычислительного устройства 602. В качестве примера, а не ограничения, машиночитаемые носители могут включать в себя "машиночитаемые носители информации" и "средства связи".

[0057] "Машиночитаемые носители информации" относятся к носителям и/или устройствам, которые предоставляют возможность хранения информации, в отличие от простой передачи сигналов, несущих или сигналов как таковых. Таким образом, машиночитаемые носители информации не включают в себя среды переноса сигналов или сигналы как таковые. Машиночитаемые носители информации включают в себя такие аппаратные средства, как энергозависимые и энергонезависимые, съемные и несъемные носители и/или запоминающие устройства, реализованные по способу или технологии, подходящих для хранения информации, такой как машиночитаемые команды, структуры данных, программные модули, логические элементы/схемы или другие данные. Примеры машиночитаемых носителей информации могут включать в себя, но не ограничиваются, RAM, ROM, EEPROM, флэш-память или другую технологию памяти, компакт-диск, универсальные цифровые диски (DVD) или другое оптическое

запоминающее устройство, жесткие диски, магнитные кассеты, магнитную ленту, накопитель на магнитных дисках или другие магнитные запоминающие устройства, либо другое запоминающее устройство, материальные носители или изделие, которые подходят для хранения нужной информации и к которым можно обращаться с помощью компьютера.

[0058] "Средства связи" относятся к средам переноса сигналов, сконфигурированным для передачи команд в аппаратные средства вычислительного устройства 602, например по сети. Средства связи обычно могут воплощать в себе машиночитаемые команды, структуры данных, программные модули или другие данные в модулированном сигнале данных, таком как несущие, сигналы данных или другой транспортный механизм. Средства связи также включают в себя любые средства доставки информации. Термин "модулированный сигнал данных" означает сигнал, который имеет одну или несколько своих характеристик, установленных или измененных таким образом, чтобы кодировать информацию в сигнале. В качестве примера, а не ограничения, средства связи включают в себя проводные средства, такие как проводная сеть или прямое проводное соединение, и беспроводные средства, такие как акустические, радиочастотные, инфракрасные и другие беспроводные средства.

[0059] Как описывалось ранее, элементы 610 аппаратных средств и машиночитаемые носители 606 соответствуют командам, модулям, логике программируемого устройства и/или логике неизменяемого устройства, реализованным в аппаратном виде, которые могут применяться в некоторых вариантах осуществления для реализации по меньшей мере некоторых аспектов описанных в этом документе методик. Элементы аппаратных средств могут включать в себя компоненты интегральной схемы или системы на кристалле, специализированной интегральной схемы (ASIC), программируемой пользователем вентильной матрицы (FPGA), сложного программируемого логического устройства (CPLD), и другие реализации в кремнии или других аппаратных устройствах. В этом смысле элемент аппаратных средств может работать в качестве устройства обработки, которое выполняет программные задачи, заданные командами, модулями и/или логикой, воплощенными элементом аппаратных средств, а также аппаратного устройства, используемого для хранения команд для исполнения, например, описанных ранее машиночитаемых носителей.

[0060] Сочетания вышеупомянутого также могут применяться для реализации различных методик и модулей, описанных в этом документе. Соответственно программное обеспечение, аппаратные средства или программные модули, включая операционную систему 110, приложения 112, платформу 116 безопасности драйверов и другие программные модули, можно реализовать в виде одной или нескольких команд и/или логики, воплощенных в некотором виде машиночитаемых носителей информации и/или с помощью одного или нескольких элементов 610 аппаратных средств. Вычислительное устройство 602 может конфигурироваться для реализации конкретных команд и/или функций, соответствующих программному обеспечению и/или аппаратным модулям. Соответственно реализация модулей в виде модуля, который в качестве программного обеспечения исполняется вычислительным устройством 602, может по меньшей мере частично достигаться в аппаратных средствах, например, посредством использования машиночитаемых носителей и/или элементов 610 аппаратных средств в системе обработки. Команды и/или функции могут исполняться/обрабатываться с помощью одного или нескольких изделий (например, одного или нескольких вычислительных устройств 602 и/или систем 604 обработки), чтобы реализовать методики, модули и примеры, описанные в этом документе.

[0061] Как дополнительно проиллюстрировано на фиг. 6, примерная система 600 обеспечивает универсальные среды для прозрачного взаимодействия с пользователем при запуске приложений на персональном компьютере (ПК), телевизионном устройстве и/или мобильном устройстве. Службы и приложения запускаются практически аналогично во всех трех средах для общего взаимодействия с пользователем при переходе с одного устройства на другое, используя при этом приложение, играя в видеоигру, просматривая видео и так далее.

[0062] В примерной системе 600 несколько устройств взаимосвязаны посредством центрального вычислительного устройства. Центральное вычислительное устройство может быть локальным для нескольких устройств или может располагаться удаленно от нескольких устройств. В одном варианте осуществления центральное вычислительное устройство может быть облаком из одного или нескольких серверов, которые подключаются к нескольким устройствам по сети, Интернету или другой линии передачи данных.

[0063] В одном варианте осуществления эта архитектура взаимосвязей дает возможность доставки функциональных возможностей между несколькими устройствами, чтобы обеспечить пользователю нескольких устройств общее и прозрачное восприятие. Каждое из нескольких устройств может обладать разными физическими требованиями и возможностями, и центральное вычислительное устройство использует платформу, чтобы сделать возможной доставку на устройство восприятия, которое как приспособлено к этому устройству, так и является общим для всех устройств. В одном варианте осуществления создается класс целевых устройств, и восприятия приспосабливаются к родовому классу устройств. Класс устройств может задаваться физическими признаками, типами использования или другими общими характеристиками устройств.

[0064] В различных реализациях вычислительное устройство 602 может допускать ряд разных конфигураций, например для использований на компьютере 614, мобильном устройстве 616 и телевизоре 618. Каждая из этих конфигураций включает в себя устройства, которые могут обладать в целом разными конструкциями и возможностями, и таким образом, вычислительное устройство 602 может конфигурироваться в соответствии с одним или несколькими разными классами устройств. Например, вычислительное устройство 602 можно реализовать как компьютерный класс 614 устройства, который включает в себя персональный компьютер, настольный компьютер, многоэкранный компьютер, переносной компьютер, нетбук и так далее.

[0065] Вычислительное устройство 602 также можно реализовать как мобильный класс 616 устройства, который включает в себя мобильные устройства, например мобильный телефон, портативный музыкальный проигрыватель, портативное игровое устройство, планшетный компьютер, многоэкранный компьютер и так далее. Вычислительное устройство 602 также можно реализовать в виде телевизионного класса 618 устройства, который включает в себя устройства, имеющие более крупные экраны или подключенные к ним в обычных средах просмотра. Эти устройства включают в себя телевизоры, телевизионные приставки, игровые приставки и так далее.

[0066] Описанные в этом документе методики могут поддерживаться этими различными конфигурациями вычислительного устройства 602 и не ограничиваются конкретными примерами описанных в этом документе методик. Это иллюстрируется посредством включения платформы 116 безопасности драйверов в вычислительное устройство 602. Функциональные возможности платформы 116 безопасности драйверов и других модулей также можно полностью или частично реализовать посредством

использования распределенной системы, например в "облаке" 620 посредством платформы 622, как описано ниже.

[0067] Облако 620 включает в себя и/или соответствует платформе 622 для ресурсов 624. Платформа 622 обобщает лежащие в основе функциональные возможности аппаратных средств (например, серверов) и программные ресурсы облака 620. Ресурсы 624 могут включать в себя приложения и/или данные, которые могут использоваться, пока компьютерная обработка исполняется на серверах, которые удалены от вычислительного устройства 602. Ресурсы 624 также могут включать в себя услуги, предоставляемые через Интернет и/или по сети абонента, например сотовой сети или сети Wi-Fi.

[0068] Платформа 622 может обобщать ресурсы и функции для соединения вычислительного устройства 602 с другими вычислительными устройствами. Платформа 622 также может служить для обобщения масштабирования ресурсов, чтобы предоставлять соответствующий уровень масштаба возникшей потребности в ресурсах 624, которые реализуются посредством платформы 622. Соответственно в варианте осуществления с взаимосвязанными устройствами реализация описанных в этом документе функциональных возможностей может распределяться по всей системе 600. Например функциональные возможности можно частично реализовать на вычислительном устройстве 602, а также посредством платформы 622, которая обобщает функциональные возможности облака 620.

ЗАКЛЮЧЕНИЕ

[0069] Несмотря на то, что изобретение описано на языке, характерном для структурных признаков и/или методологических действий, необходимо понимать, что изобретение, определенное в прилагаемой формуле изобретения, необязательно ограничивается описанными характерными признаками или действиями. Скорее, характерные признаки и действия раскрываются в виде примерных форм реализации заявленного изобретения.

## (57) Формула изобретения

1. Способ исполнения преобразующего драйвера, реализуемый вычислительным устройством, при этом способ содержит этапы, на которых:

получают преобразующий драйвер, содержащийся в комплекте драйвера, имеющем назначенный формат, ассоциированный с платформой безопасности драйверов;

распознают назначенный формат комплекта драйвера при установке, основываясь, по меньшей мере отчасти, на идентификационных данных, включенных в комплект драйвера;

в качестве реакции на упомянутое распознавание, регистрируют преобразующий драйвер в платформе безопасности драйверов, реализуемой вычислительным устройством;

создают экземпляр ограниченной среды исполнения для преобразующего драйвера посредством платформы безопасности драйверов; и

исполняют преобразующий драйвер в ограниченной среде исполнения, чтобы выполнять одну или несколько задач по указанию платформы безопасности драйверов.

2. Способ по п. 1, в котором преобразующий драйвер исполняется в пользовательском режиме.

3. Способ по п. 1, в котором преобразующий драйвер исполняется для каждого конкретного пользователя.

4. Способ по п. 1, в котором ограниченная среда исполнения создает контейнер

приложения, сконфигурированный для изолирования преобразующего драйвера, чтобы ограничить доступ преобразующего драйвера к ресурсам системы.

5. Способ по п. 4, в котором контейнер приложения дополнительно конфигурируется для отказа в доступе к преобразующему драйверу компонентам, которые не ассоциированы с платформой безопасности драйверов.

6. Способ по п. 1, в котором ограниченная среда исполнения содержит фоновый процесс для преобразующего драйвера, для которого планирование задач осуществляется платформой безопасности драйверов.

7. Способ по п. 1, в котором ограниченная среда исполнения конфигурируется для приостановки преобразующего драйвера при отсутствии явного вызова преобразующего драйвера платформой безопасности драйверов для выполнения задач.

8. Способ по п. 1, в котором ограниченная среда исполнения конфигурируется для выборочного разрешения операций преобразующего драйвера в зависимости от согласия пользователя.

9. Способ по п. 1, в котором упомянутое получение преобразующего драйвера выполняется по сети из магазина приложений у поставщика услуг.

10. Способ по п. 1, в котором преобразующий драйвер сконфигурирован преобразовывать данные из одной формы в другую.

11. Способ по п. 1, в котором платформа безопасности драйверов реализована как компонент операционной системы для вычислительного устройства.

12. Способ по п. 1, в котором преобразующий драйвер выполнен в виде драйвера виртуальной частной сети (VPN), сконфигурированного для инкапсуляции и декапсуляции пакета для связи VPN по сети.

13. Машиночитаемый носитель информации, на котором сохранены инструкции, которыми при их исполнении одним или более компонентами вычислительного устройства реализуется платформа безопасности драйверов, выполненная с возможностью осуществления операций, в соответствии с которыми:

получают преобразующий драйвер, содержащийся в комплекте драйвера, имеющем назначенный формат, ассоциированный с платформой безопасности драйверов;

распознают, что преобразующий драйвер ассоциирован с платформой безопасности драйверов, основываясь по меньшей мере отчасти на идентификационных данных, включенных в комплект драйвера;

в качестве реакции на упомянутое распознавание удостоверяют набор задач, разрешенных для преобразующего драйвера через платформу безопасности драйверов;

создают среду исполнения, которая ограничивает работу преобразующего драйвера упомянутым набором задач, которые разрешены через платформу безопасности драйверов; и

управляют работой преобразующего драйвера через среду исполнения, которая ограничивает работу преобразующего драйвера упомянутым набором задач.

14. Машиночитаемый носитель информации по п. 13, при этом преобразующий драйвер представляет собой драйвер виртуальной частной сети (VPN), и упомянутый набор задач, которые разрешены через платформу безопасности драйверов для драйвера VPN, ограничен задачами подключения, отключения, инкапсуляции и декапсуляции, ассоциированными со связью VPN с VPN-сервером по сети.

15. Машиночитаемый носитель информации по п. 13, при этом упомянутое управление работой преобразующего драйвера включает в себя:

вызов преобразующего драйвера для выполнения упомянутого набора задач, которые разрешены;

приостановку преобразующего драйвера при отсутствии указания со стороны упомянутой платформы выполнять задачи.

16. Машиночитаемый носитель информации по п. 13, при этом упомянутое управление работой преобразующего драйвера включает в себя планирование упомянутого набора задач в качестве фоновых задач, которые регулируются платформой безопасности драйверов.

17. Машиночитаемый носитель информации по п. 13, при этом среда исполнения содержит контейнер приложений "в песочнице", который исполняется в пользовательском режиме для каждого конкретного пользователя и которому назначен маркер малых привилегий, сконфигурированный для предотвращения доступа к системе, за исключением того, что относится к упомянутому набору задач, которые разрешены.

18. Вычислительная система, выполненная с возможностью исполнения драйверов виртуальной частной сети (VPN), при этом система содержит:

один или более компонентов обработки данных;

один или более машиночитаемых носителей информации, на которых сохранены инструкции, которыми при их исполнении одним или более компонентами обработки данных реализуется платформа безопасности драйверов, которая ограничивает работу по меньшей мере одного драйвера VPN, включающая в себя:

модуль диспетчера драйверов, чтобы:

получать упомянутый драйвер VPN по сети из магазина приложений у поставщика услуг, такой драйвер VPN содержится в комплекте драйвера, имеющем назначенный формат, ассоциированный с платформой безопасности драйверов,

распознавать назначенный формат комплекта драйвера при установке, основываясь по меньшей мере отчасти на идентификационных данных, включенных в комплект драйвера, включающих в себя идентификатор, код или расширение файла, и

в качестве реакции на упомянутое распознавание создавать контейнер приложений для помещения в него упомянутого драйвера VPN, который исполняется в пользовательском режиме для каждого конкретного пользователя и которому назначен маркер малых привилегий, сконфигурированный для предотвращения доступа к системе, за исключением того, что относится к заданному набору задач, которые явно разрешены для упомянутого драйвера VPN платформой безопасности драйверов;

модуль диспетчера фоновой работы, чтобы:

создавать экземпляр фонового процесса для контейнера приложений, в который помещен упомянутый драйвер VPN,

осуществлять планирование упомянутого заданного набора задач через этот фоновый процесс, и

приостанавливать данный фоновый процесс кроме случаев, когда этот заданный набор задач выполняется по указанию со стороны модуля диспетчера драйверов; и

брокер событий для формирования событий, чтобы урегулировать взаимодействие между упомянутым драйвером VPN и системными службами для выполнения задач, запланированных через модуль диспетчера фоновой работы.

19. Вычислительная система по п. 18, при этом события, формируемые брокером событий, включают в себя события пользовательского интерфейса (UI) для урегулирования взаимодействий с компонентами UI, предоставляемыми через вычислительную систему.

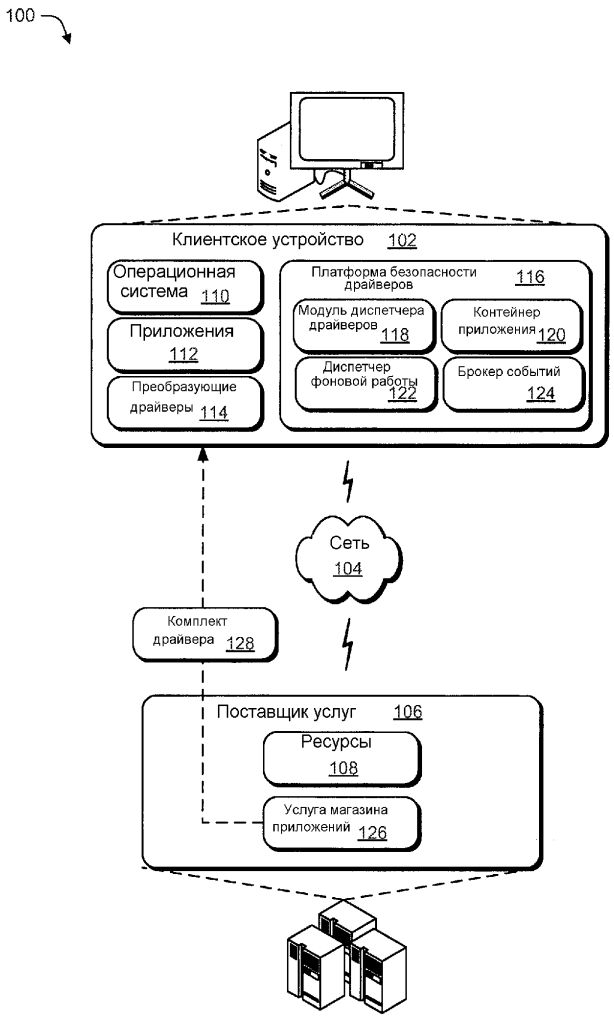
20. Вычислительная система по п. 18, при этом упомянутый заданный набор задач, которые явно разрешены для драйвера VPN платформой безопасности драйверов, включает в себя одну или более из задач подключения, отключения, инкапсуляции и

декапсуляции, ассоциированных со связью VPN с VPN-сервером по сети.

1

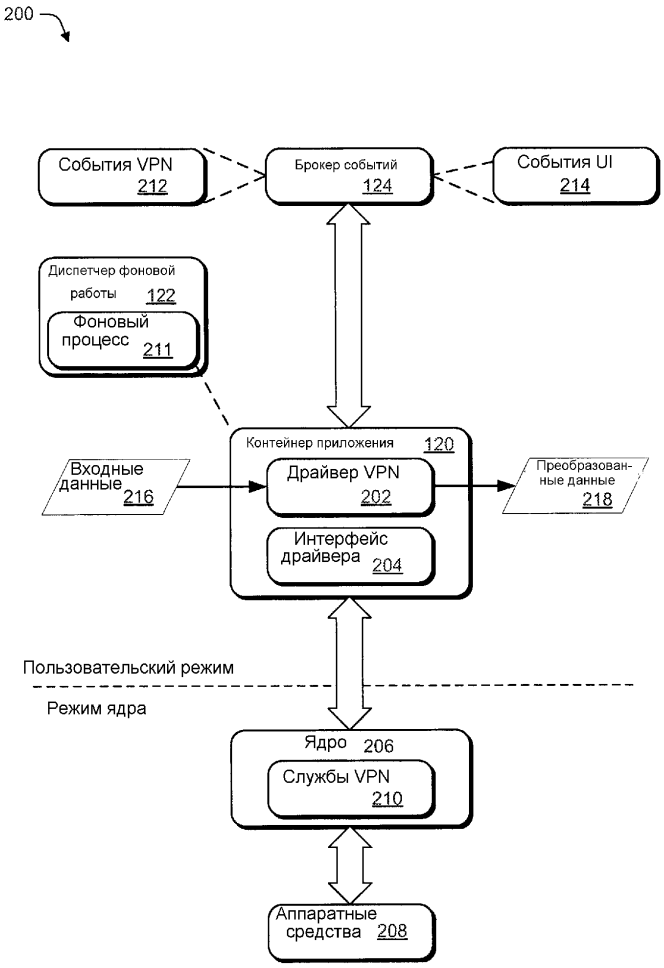





100
Клиентское устройство 102
Операционная система 110
Приложения 112
Преобразующие драйверы 114
Платформа безопасности драйверов 116
Модуль диспетчера драйверов 118
Контейнер приложения 120
Диспетчер фоновой работы 122
Брокер событий 124
Сеть 104
Комплект драйвера 128
Поставщик услуг 106
Ресурсы 108
Услуга магазина приложений 126

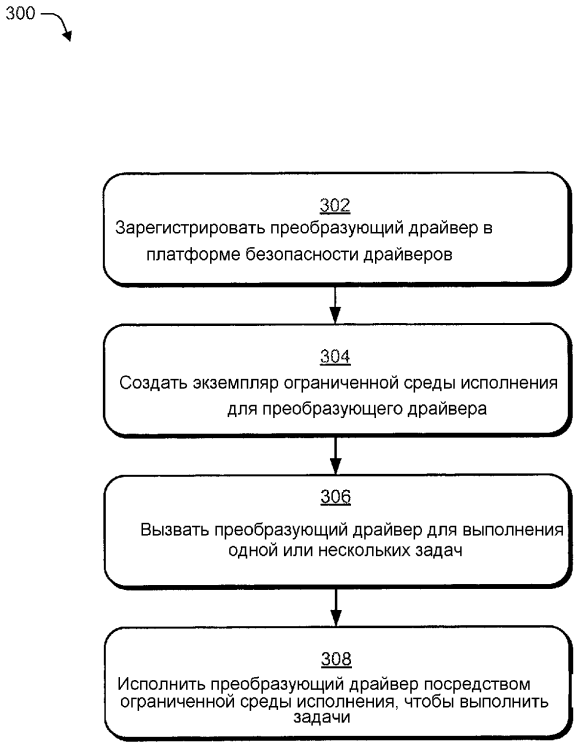
ФИГ.1

2


200
События VPN 212
Брокер событий 124
События UI 214
Диспетчер фоновой работы 122
Фоновый процесс 211
Контейнер приложения 120
Входные данные 216
Драйвер VPN 202
Преобразованные данные 218
Интерфейс драйвера 204
Пользовательский режим
Режим ядра
Ядро 206
Службы VPN 210
Аппаратные средства 208

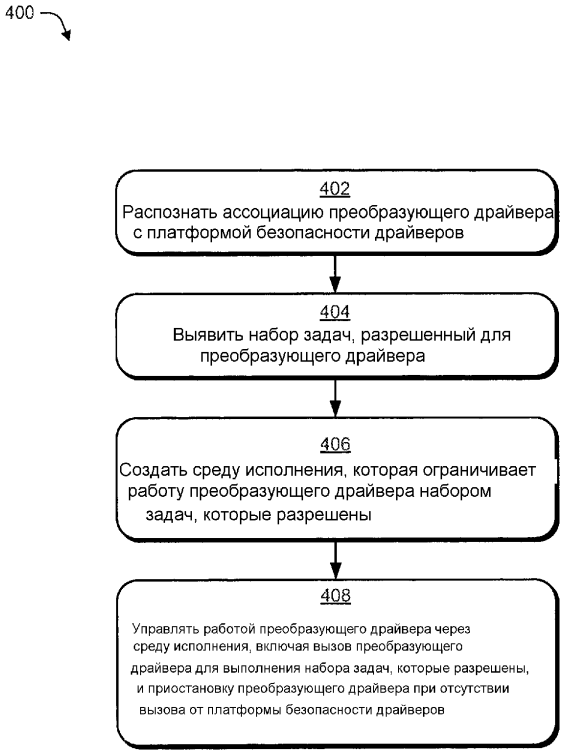
ФИГ.2

3

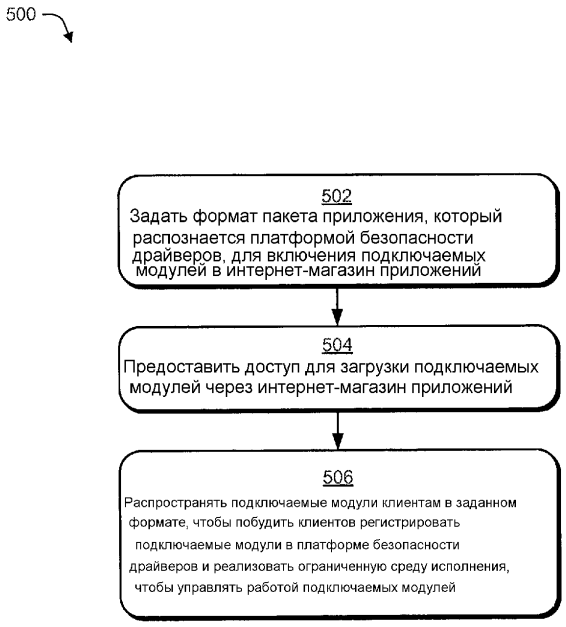

300
302
Зарегистрировать преобразующий драйвер в платформе безопасности драйверов
304
Создать экземпляр ограниченной среды исполнения для преобразующего драйвера
306
Вызвать преобразующий драйвер для выполнения одной или нескольких задач
308
Исполнить преобразующий драйвер посредством ограниченной среды исполнения, чтобы выполнить задачи

ФИГ.3

4

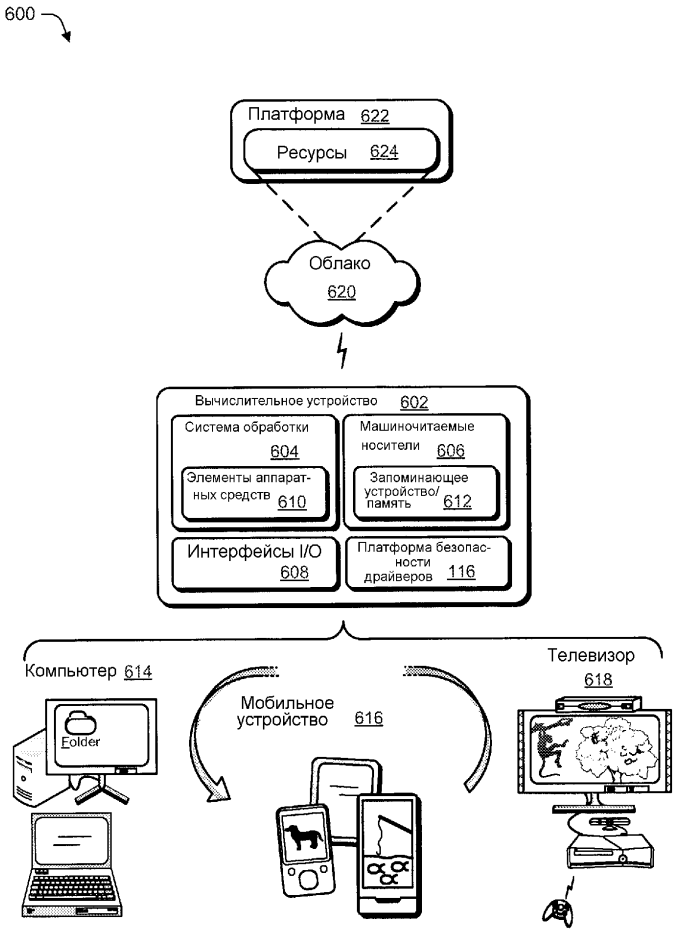

400
402
Распознать ассоциацию преобразующего драйвера с платформой безопасности драйверов
404
Выявить набор задач, разрешенный для преобразующего драйвера
406
Создать среду исполнения, которая ограничивает работу преобразующего драйвера набором задач, которые разрешены
408
Управлять работой преобразующего драйвера через среду исполнения, включая вызов преобразующего драйвера для выполнения набора задач, которые разрешены, и приостановку преобразующего драйвера при отсутствии вызова от платформы безопасности драйверов

ФИГ.4


500
502
Задать формат пакета приложения, который распознается платформой безопасности драйверов, для включения подключаемых модулей в интернет-магазин приложений
504
Предоставить доступ для загрузки подключаемых модулей через интернет-магазин приложений
506
Распространять подключаемые модули клиентам в заданном формате, чтобы побудить клиентов регистрировать подключаемые модули в платформе безопасности драйверов и реализовать ограниченную среду исполнения, чтобы управлять работой подключаемых модулей

ФИГ.5

6


600
Платформа 622
Ресурсы 624
Облако 620
Вычислительное устройство 602
Система обработки 604
Элементы аппаратных средств 610
Машиночитаемые носители 606
Запоминающее устройство/ память 612
Интерфейсы I/O 608
Платформа безопасности драйверов 116
Компьютер 614
Folder
Мобильное устройство 616
Телевизор 618

ФИГ.6