УДК 004.056.52

МОДЕЛЬ ДОСТУПА В ОБЛАЧНЫХ ИНФРАСТРУКТУРАХ

П. М. ЖУРОВ
МФТИ (ГУ), Москва, 117303, Россия

Введение

Сложно представить современную ИТ-индустрию без облачных технологий. Их популярность растёт с каждым днем [1], а вместе с ней растёт количество данных хранящихся и обрабатываемых в облаках. Задача разграничения доступа к этими данным стоит очень остро [2-6], так как одной из ключевых характеристик облака является то, что данные разных проектов, а иногда даже компаний, хранятся и обрабатываются на одних и тех же физических ресурсах. Решение этой задачи осложняется тем, что для облачных инфраструктур на текущий момент отсутствует модель, которая описывает взаимодействие субъектов и объектов в облаке при различных действиях пользователей, из-за чего практически невозможно гарантированно ограничить доступ определенных субъектов к определенным объектам. В данной работе описано возможное решение данной проблемы – модель доступа для облачных инфраструктур.

Рассматриваемые типы облачных инфраструктур

В настоящий момент принято выделять три типа облачных инфраструктур:

- Инфраструктура как сервис (Infrastructure-as-a-Service, IaaS). В IaaS пользователь имеет возможность создавать виртуальные объекты (ВМ и контейнеры), и получает полный доступ к ним (доступ суперпользователя на уровне ОС).
- Платформа как сервис (Platform-as-a-Service, PaaS). В PaaS пользователь взаимодействует с некой платформой, которая позволяет ему запускать свои приложения на серверах провайдера. Приложения разных пользователей изолируются между собой с помощью тех же виртуальных объектов (ВМ и контейнеров), однако полного доступа к ОС у пользователя уже нет – уровень этого доступа определяется платформой. Обычно это все то, что позволяет сформировать окружение приложения: библиотеки, файлы конфигурации, переменные окружения и прочее.
- Приложение как сервис (Software-as-a-Service, SaaS). В SaaS пользователь просто пользуется приложением, которое позволяет ему хранить/обрабатывать свои данные на серверах провайдера. Типичные примеры: Dropbox, iCloud, Microsoft Office Online, GSuite и т.д.

Очевидно, что SaaS не представляет большой интерес, с точки зрения моделирования доступов, так все происходит в рамках одного приложения, поэтому в дальнейшем мы будем рассматривать только IaaS и PaaS.

Ресурсы и пользователи облака

В облаке определяются два типа ресурсов: аппаратные (физические) и виртуальные. Виртуальные ресурсы – это всевозможные программно изолированные сущности (ВМ, контейнеры, виртуальные свитчи и хранилища и т.д.), которые предоставляются (напрямую или опосредованно) пользователю облачным провайдером. При создании этих сущностей, они привязываются к определенному физическому ресурсу (серверу), однако могут мигрировать при определенных условиях (в основном связанных с тем, что физическое оборудование перегружено, и необходимо распределить эту нагрузку по менее загруженным серверам). Здесь и далее, будем считать виртуальные ресурсы атомарной единицей. Доступы внутри них за рамками описываемой модели.

Облако состоит из программных компонентов – сервисов, которые управляют виртуальными ресурсами, в том числе принимают решение о том, на каких физических ресурсах будут запущены виртуальные. Существуют специальные сервисы, которые позволяют настраивать другие сервисы – сервисы администрирования. Эти сервисы неизменны и их можно только обновлять. Пользователи взаимодействуют с облаком только через сервисы и делятся на три группы по глубине этого взаимодействия: администраторы облака взаимодействуют с сервисами администрирования, тем самым имея возможность управлять другими сервисами; арендаторы взаимодействуют с сервисами, позволяющими управлять виртуальными ресурсами; рядовые пользователи взаимодействуют с виртуальными ресурсами через определенные сервисы.

Модель доступа

Перейдём непосредственно к построению модели. Для начала введём базовые определения:

Определение 4. Объект – это сущность КС, которая содержит или получает информацию (данные) и над которой субъекты выполняют операции.

Определение 5. Субъект – это сущность КС, которая инициирует выполнение действий над другими сущностями.

Очень часто взаимодействие с объектами происходит не напрямую, а опосредованно: например, программа запрашивает у другой программы выполнение определенных действий. В таком случае происходит порождение новых субъектов, путем воздействия субъекта на объект. Отсюда введём предположение:

Предположение 1 Новые субъекты могут появляться в результате воздействия другого субъекта на ассоциированный с первым субъектом объект. $CreateSubject(S, O) \rightarrow S'$

Будем считать, что это предположение справедливо для любого вида КС, в том числе и ОИ. Тогда, для того чтобы описать модель доступа в ОИ необходимо описать объекты ОИ, сценарии использования ОИ, субъекты ОИ и, наконец, доступы субъектов к объектам.

Описание объектов и субъектов

Начнем с объектов, как было сказано выше, их можно разделить на три группы:

1. Сервисы
2. Виртуальные ресурсы
3. Аппаратные ресурсы

Определим пользователей ОИ – их тоже можно разделить на три группы:

1. Администраторы
2. Арендаторы
3. Рядовые пользователи

В приведенном выше описании ещё нет процесса создания объектов, поэтому на текущий момент, оно подходит только для стационарных систем (т.е. систем не меняющих своё состояния). Однако, одно из основных свойств ОИ – это постоянное изменение состояния системы за счёт непрерывного создания новых сущностей и удаления старых. Отсюда возникает необходимость описать то, как в модели доступов ОИ происходит создание объектов.

Для этого предлагается зафиксировать несколько дополнительных предположений.

Предположение 2 Облачная инфраструктура ограничена определенным количеством аппаратных ресурсов.

Действительно, у каждого облака есть ограничение по количеству доступных физических ресурсов. Да, зачастую у провайдера облачных услуг есть возможность установить новую стойку с серверами в центр обработки данных (ЦОД) или даже

построить новый ЦОД, однако это происходит не «по запросу… вне зависимости от времени суток…».

Отсюда следуют вывод, что не смотря на то, что облако является динамически изменяющейся системой, процесс создания/удаления происходит только с определенным типом объектов – виртуальными ресурсами. При этом, важно также учитывать то, что это изменение происходит только при взаимодействии сервисов с аппаратными ресурсами.

Предположение 3 В ОИ могут создаваться только объекты типа виртуальный ресурс, и создание таких объектов происходит только при взаимодействии сервисов с аппаратными ресурсами. $CreateObject(S_s, O_{HR}) \rightarrow O_{VR}$

Сценарии взаимодействия пользователей с ОИ

Когда пользователь взаимодействует с определенными объектами ОИ, они становятся субъектами – назовем это *активацией субъекта*. Субъекты, в свою очередь могут получать доступ к объектам (чтение, запись и запись в конец), активировать их, создавая новые субъекты (предположение 2), а некоторые из них могут создавать новые объекты воздействуя на аппаратные ресурсы (предположение 3).

Сценарий взаимодействия пользователя с субъектом зависит от того к какой группе принадлежит пользователь. Так, администраторы взаимодействуют со специальным сервисом, который позволяет управлять другими сервисами. Во время такого взаимодействия этот сервис (назовем его сервисом администрирования) становится субъектом и получает доступ к аппаратным ресурсам, на которых расположены другие сервисы, что позволяет ему изменять их конфигурацию (рис. 1). Например, администратор, используя VMware vSphere может настроить определенным образом гипервизор ESXi. Здесь vSphere выступает в роли сервиса администрирования, а гипервизор в качестве сервиса управления вычислительными виртуальными ресурсами.

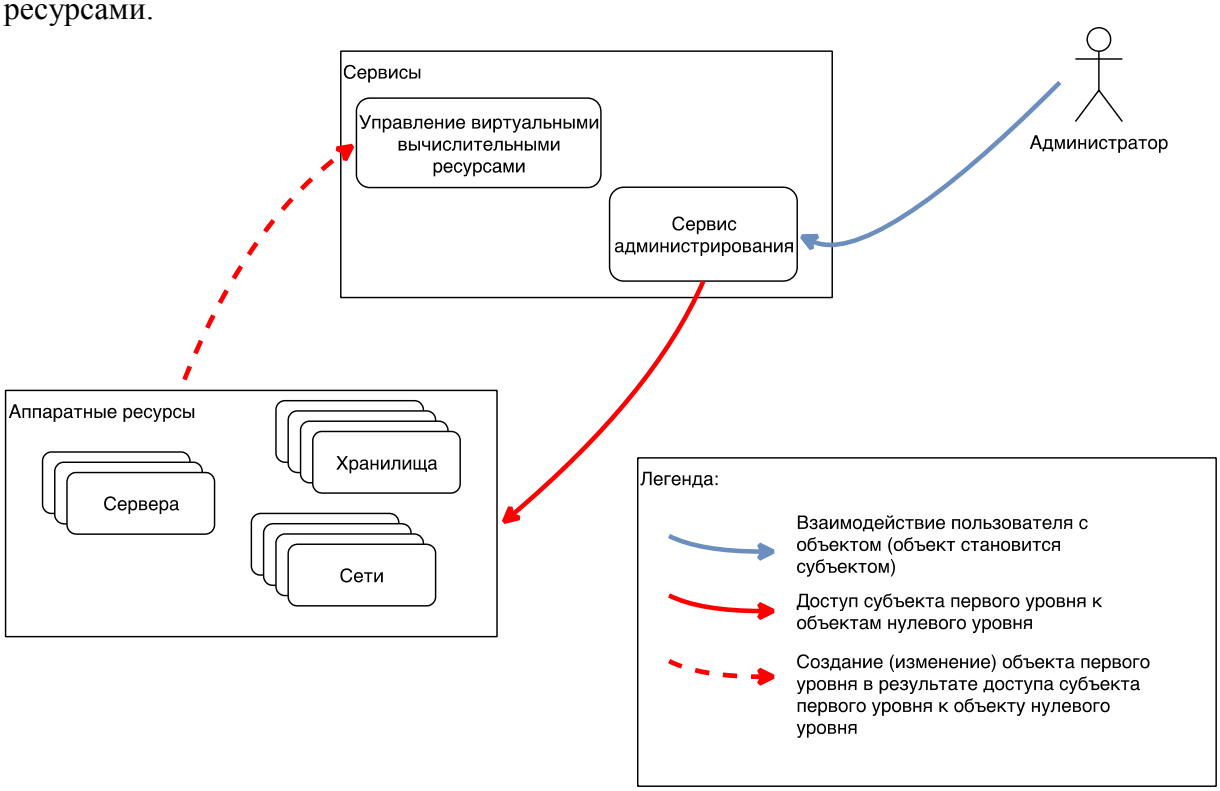

Рисунок 1. Пример взаимодействия администратора с ОИ.

Арендаторы в свою очередь взаимодействуют с различными сервисами для создания виртуальных ресурсов. Важное отличие облака от классических виртуальных инфраструктур в том, что то, на каком конкретно объекте аппаратного уровня запустится

виртуальный ресурс, решает не пользователь, а специальный сервис, который называется планировщиком. Этот процесс представлен на рисунке 2. В качестве примера из реально использующихся реализаций можно привести OpenStack, где арендатор посылает API запрос к сервису Nova для создания новой ВМ, к сервису Neutron для создания новой виртуальной сети, к сервису Cinder для создания нового виртуального блочного хранилища и т.д.

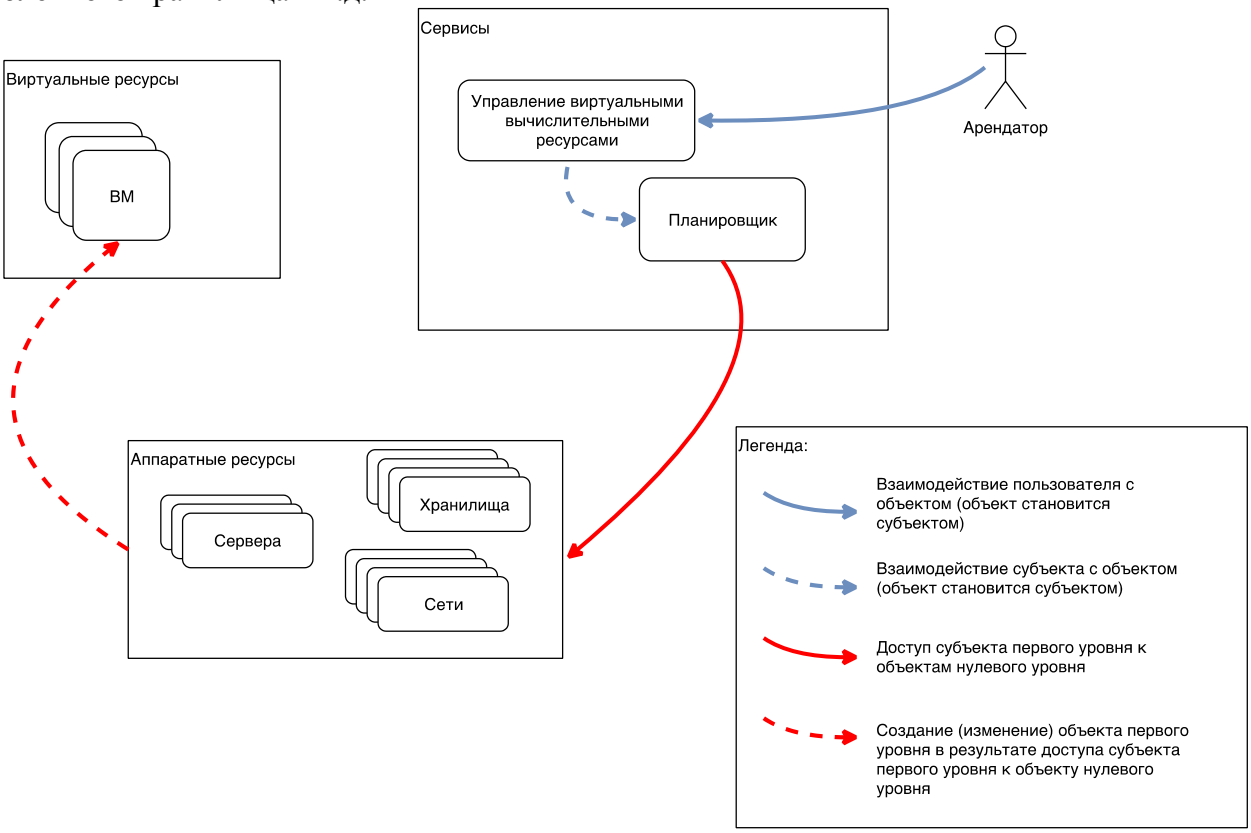

Рисунок 2. Пример взаимодействия арендатора с ОИ.

Наконец рядовые пользователи взаимодействуют напрямую с виртуальными ресурсами, получая доступ к данным, которые хранятся в этих ресурсах. (рис. 3)

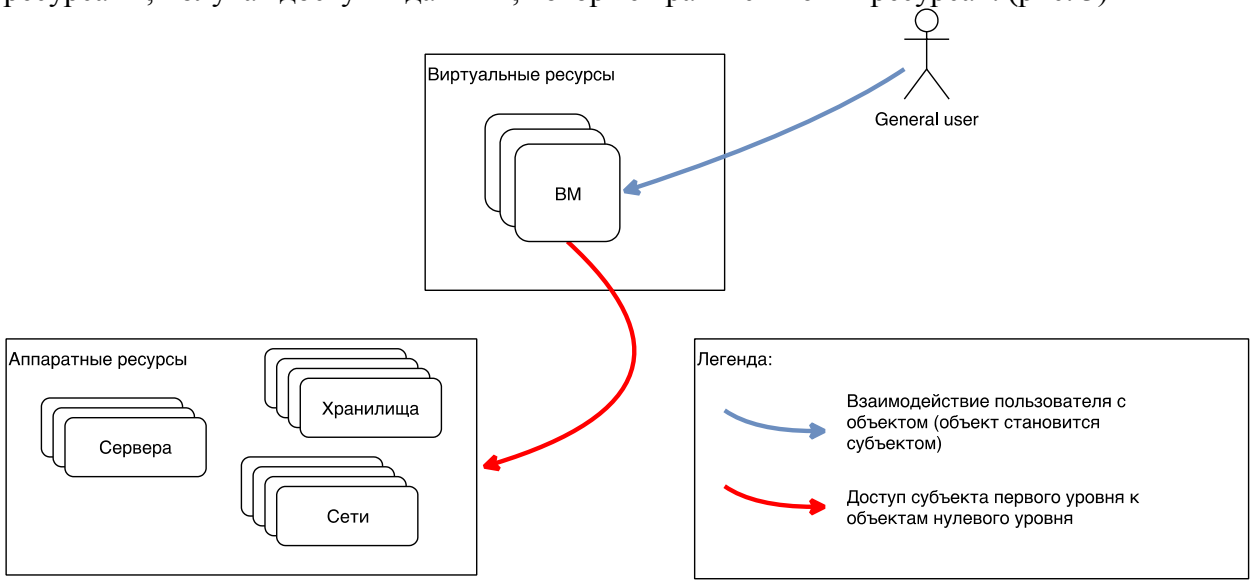

Рисунок 3. Пример взаимодействия рядовых пользователей с ОИ.

Формальное описание модели доступа в ОИ.

Теперь, когда мы определили все основные понятия, воспользуемся классическим подходом [7] для формального описания модели доступа, состоящим в том, что моделируемая КС (в нашем случае ОИ) представляется абстрактной системой, каждое состояние которой представляется графом доступов. Переходы этой системы из состояния в состояние происходят при создании новых субъектов и объектов согласно предположениям 2 и 3.

Введем следующие обозначения:

$O = O_S \cup O_{VR} \cup O_{HR}$ – множество всех объектов в данный момент времени, где O_S – множество сервисов, O_{VR} – множество виртуальных ресурсов, O_{HR} – множество аппаратных ресурсов и $O_S \cap O_{VR} = \emptyset,\ O_{VR} \cap O_{HR} = \emptyset, O_S \cap O_{HR} = \emptyset$

$S = S_S \cup S_{VR}$ – множество текущих субъектов, где $S_S \subseteq O_S$ – множество субъектов-сервисов, $S_{VR} \subseteq O_{VR}$ – множество субъектов-виртуальных ресурсов.

$U = U_A \cup \mathrm{U}_T \cup U_{GU}$ – множество пользователей, где U_A – множество администраторов, U_T – множество арендаторов, U_{GU} – множество рядовых пользователей и $U_T \subseteq U_{GU},\ U_{GU} \cap U_A = \emptyset.$

$\mathrm{A} = \{byUser, bySubject\}$ – множество видов активации субъекта, где $byUser$ – активация субъекта пользователем, а $bySubject$ – другим субъектом.

$R_a = \{read_a, write_a, append_a\}$ – множество видов доступа, где $read_a$ – доступ на чтение из сущности; $write_a$ – доступ на запись в сущность; $append_a$ – доступ на запись в конец слова, содержащегося в сущности.

Определение 8. Определим $\forall u \in U: userSubjects(u) = (u, S, A)$ – ориентированное дерево с корневой вершиной u, которое показывает все активированные пользователем субъекты в данный момент времени.

Определение 9. Определим $G = (U, O, \{A \cup R_a\})$ – конечный помеченный ориентированный граф, без петель, обладающий следующими характеристиками:

1. Элементы множеств U, O являются вершинами графа
2. Элементы множества $\{A \cup R_a\}$ – ребра графа.
3. Граф содержит в себе все деревья $userSubjects$
4. Вершины, являющиеся элементами множества $O \backslash S$ (т.е. объекты, не являющиеся субъектами) соединены с вершинами, являющимися элементами множества S ребрами, являющимися элементами множества R_a.

Назовем $G = (U, O, \{A \cup R_a\})$ графом доступа. Обозначим также множество E – множество всех ребер графа G. При этом в самом графе доступов будем использовать следующие обозначения:

- вершины из множества U (соответствующие пользователям) в графе доступов будут обозначатся «⊕»
- вершины из множества S (соответствующие субъектам) в графе доступов будут обозначатся «•»
- вершины из множества $O \backslash S$ (соответствующие объектам, не являющимися субъектами) в графе доступов будут обозначатся «⊗»
- каждое ребро графа доступов будет помечено одним из элементов множества $\{A \cup R_a\}$
- ребра, помеченные элементами множества A будут обозначаться синей стрелкой, а помеченные элементами множества R_a – красной.

На рис. 4 изображен пример графа доступа. Для удобства пунктиром объединены различные подмножества объектов.

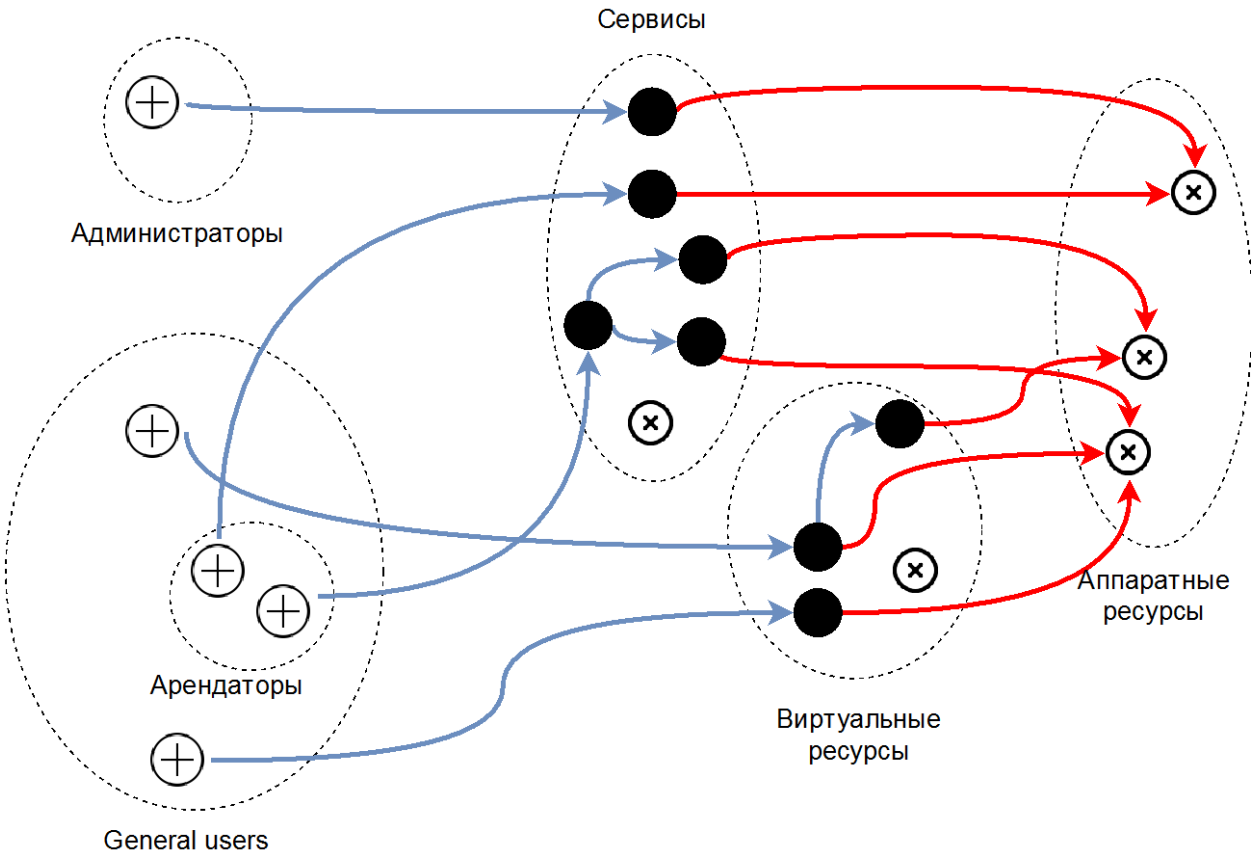

Рисунок 4 Пример графа доступов

Описание функций перехода модели из состояния в состояние.

Опишем теперь, как изменяются состояния графа доступов. Согласно предположениям 1–4 введем следующие функции перехода:

$CreateSubject(u/s',\ o) \to s$ – функция активации объекта пользователем или субъектом (таб. 1), где $u \in U,\ o \in O,\ s, s' \in S$.

Таблица 1. Состояния системы в результате применения функции активации объекта пользователем.

Начальное состояние	Конечное состояние
$U,\ S,\ O,\ E$	$U, S \cup \{s\}, E \cup \{(u/s', s, byUser/bySubject), (s, o', r)\},$ $r \in R_a$

В итоговом графе доступов появится либо ребро $byUser$, указывающее на то, что субъект был активирован пользователем, либо ребро $bySubject$, указывающее на то, что субъект был активирован другим субъектом. При этом активированный субъект получает доступ к одному или нескольким объектам системы.

$CreateObject(s, o) \to o'$ – функция создания объекта (таб. 2), где $o \in O_{HR}, o' \in O_{VR}, s \in S_s$.

Таблица 2. Состояния системы в результате применения функции создания объекта.

Начальное состояние	Конечное состояние
$U,\ S,\ O,\ E$	$U, S, O \cup \{o'\}, E$

$DeactivateSubject(u/s', s) \to o$ – функция деактивации субъекта пользователем или субъектом (таб. 3), где $u \in U, o \in O,\ s, s' \in S$.

Таблица 3. Состояния системы в результате применения функции деактивации субъекта пользователем.

Начальное состояние	Конечное состояние
$U,\ S,\ O,\ E$	$U,\ \ E \setminus \{(u/s', s, byUser/bySubject)\}$

$DeleteObject(s, o)$ – функция удаления объекта (таб. 4), где $o \in O_{VR},\ s \in S_S$.

Таблица 4. Состояния системы в результате применения функции удаления объекта.

Начальное состояние	Конечное состояние
$U,\ S,\ O,\ E$	$U, S, O \setminus \{o\}, E$

Возможные применения описанной модели доступа

Помимо того, что модель доступа решает проблему описанную во вступлении (а именно, позволяет предсказывать изменение состояния облака при взаимодействии с ней пользователей), она также может иметь несколько практических применений. В первую очередь, она позволяет применять классические модели разграничения доступа (такие как модель Белла-ЛаПудулы, модель take-grant и т.д.) в ОИ. Во-вторых, дополнив эту модель концептом информационных потоков, можно реализовать продвинутую систему аудита событий, которая позволит быстрее расследовать причины возникновения инцидентов в ОИ.

Заключение

В статье описана модель доступа для облачных инфраструктур, которая решает сразу несколько проблем связанных с обеспечением безопасности в ОИ. В дальнейшем планируется добавить в эту модель концепт информационных потоков, а также использовать эту модель для разработки централизованной системы управления доступа в облаке [8].

Список литературы

1. The 2020 state of it: The annual report on it budgets and tech trends [Электронный ресурс]. – Режим доступа: https://www.spiceworks.com/marketing/state-of-it/report/
2. Subashini S., Kavitha V. A Survey on Security Issues in Service Delivery Models of Cloud Computing // Journal of Network and Computer Applications. – 2011. vol. 34. no.1. – pp. 1-11.
3. Sumitra B. Pethuru C.R., Misbahuddin M. A Survey of Cloud Authentication Attacks and Solution Approaches // International Journal of Innovative Research in Computer and Communication Engineering. – 2014. vol. 2. Issue 10. – pp. 6245-6253.
4. Cai, F., Zhu, N., He, J. et al. Survey of access control models and technologies for cloud computing // Cluster Computing 22, 6111–6122 (2019)
5. Charanya R., Aramudhan M. Survey on access control issues in cloud computing // 2016 International Conference on Emerging Trends in Engineering, Technology and Science (ICETETS), Pudukkottai, 2016, pp. 1-4
6. Кодолов П. А. Проблемы безопасности облачных вычислений // Наука, техника и образование. 2016. №4 (22)
7. Девянин П.М. Модели безопасности компьютерных систем: учебное пособие для студентов высших учебных заведений. – М: Издательский дом “Академия”. – 2005. – 144 с.

8. Журов П. М. Централизованная система разграничения доступа в облаке. – Сборник тезисов КЗИ 2019