

## МОДЕЛЬ ИНТЕРОПЕРАБЕЛЬНОСТИ ОБЛАЧНЫХ ВЫЧИСЛЕНИЙ

Е. Е. Журавлёв<sup>1</sup>, С. В. Иванов<sup>2</sup>, А. Я. Олейников<sup>3</sup>

<sup>1</sup> - Физический институт им. П.Н. Лебедева РАН

<sup>2</sup> – Российский новый университет (РосНУ)

<sup>3</sup> - Институт радиотехники и электроники им. В.А. Котельникова РАН

Статья получена 6 декабря 2013 г.

**Аннотация.** На основе синтеза: эталонной архитектуры облачных вычислений, предложенной NIST; эталонной трехуровневой модели интероперабельности, приведенной в ГОСТ Р 55062-2012; и открытого интерфейса облачных вычислений, разработанного рабочей группой Open Cloud Computing Initiative международной организации Open Grid Forum, а также с использованием материалов «дорожной карты» NIST по разработке стандартов облачных вычислений, построена модель интероперабельности облачных вычислений. Показано, что предложенная модель в целом совпадает с моделью интероперабельности открытой грид-среды. Разница заключается в наличии для облачных вычислений интерфейса приложение-платформа (API).

Работа выполнена в рамках проекта РФФИ 12-07-00261-а и Программы Президиума РАН № 14.

**Ключевые слова:** облачные вычисления, архитектура облачных вычислений, стандарт, интероперабельность, интерфейс, модель интероперабельности облачных вычислений.

**Abstract.** On the basis of synthesis of: reference architecture cloud computing proposed by NIST; reference the three-tier model of interoperability given in GOST R 55062-2012; and the Open Cloud Computing Interface, developed by the working group Open Cloud Computing Initiative of the international organization Open Grid Forum, and also with use of NIST Cloud Computing Standards Roadmap the model of interoperability of

cloud computing is developed. It is shown that the proposed model in general coincides with the model of interoperability of open grid environment. The difference lies in the availability of cloud computing interface application-platform (API).

The work was performed under the project RFBR 12-07-00261 and the Program of the Presidium RAS № 14.

**Keywords:** cloud computing, cloud computing architecture, standard, interoperability, interface, interoperability model of cloud computing.

### **Введение**

В стандарте ГОСТ Р 55062-2012, разработанном ИПЭ им. В.А. Котельникова РАН, «Информационные технологии. Системы промышленной автоматизации и их интеграция. Интероперабельность. Основные положения» описывается единый подход к обеспечению интероперабельности для систем самого широкого класса [1]. Хотя настоящий стандарт предназначен в первую очередь для систем промышленной автоматизации, он имеет гораздо более широкое назначение. На его основе могут создаваться интероперабельные системы с учетом их особенностей, в том числе – для систем облачных вычислений. Одной из задач единого подхода служит введение на уровне стандарта эталонной модели интероперабельности, подобно хорошо известной 7-уровневой модели взаимосвязи открытых систем ISO 7498-1984 и эталонной модели среды открытых систем ISO/IEC 14252 -1996. Эта модель представлена на рисунке 1.

В [2] проведены исследования работ по интероперабельности облачных вычислений, представлена расширенная 6-уровневая модель интероперабельности [3]. Эта модель, как и модель, представленная на рис. 1, являются моделями высокого уровня абстракции и требуют уточнения.

Для построения модели интероперабельности облачных вычислений (см. рис 2) мы предлагаем использовать эталонную архитектуру NIST (рис. 3), эталонную модель (рис. 1), открытый интерфейс облачных вычислений, разработанный

рабочей группой Open Cloud Computing Initiative международной организации Open Grid Forum, а также документ NIST Cloud Computing Standards Roadmap [4].

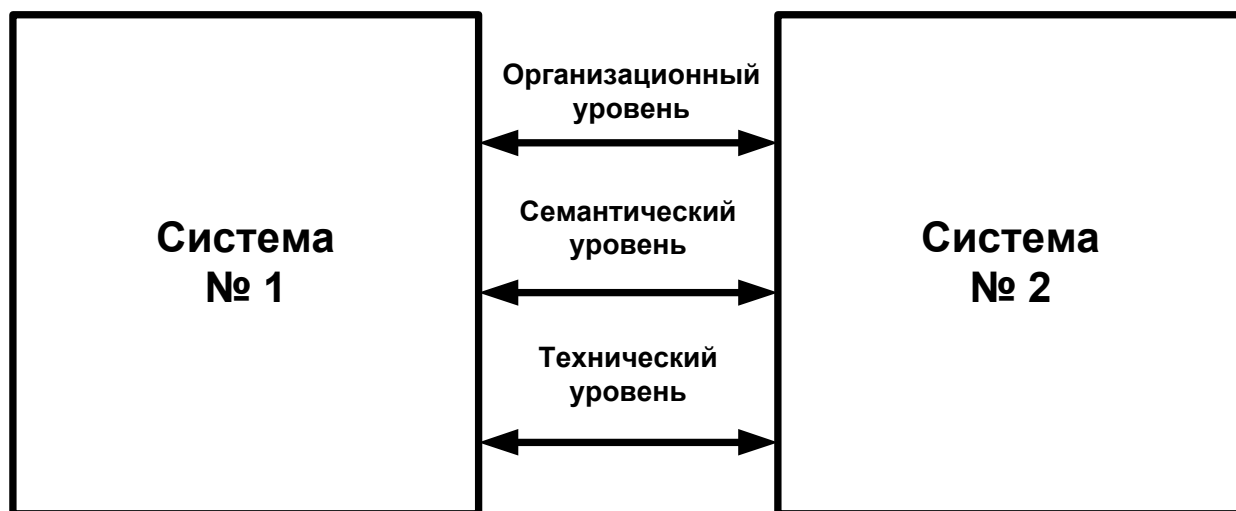


Рис. 1 Эталонная модель интероперабельности

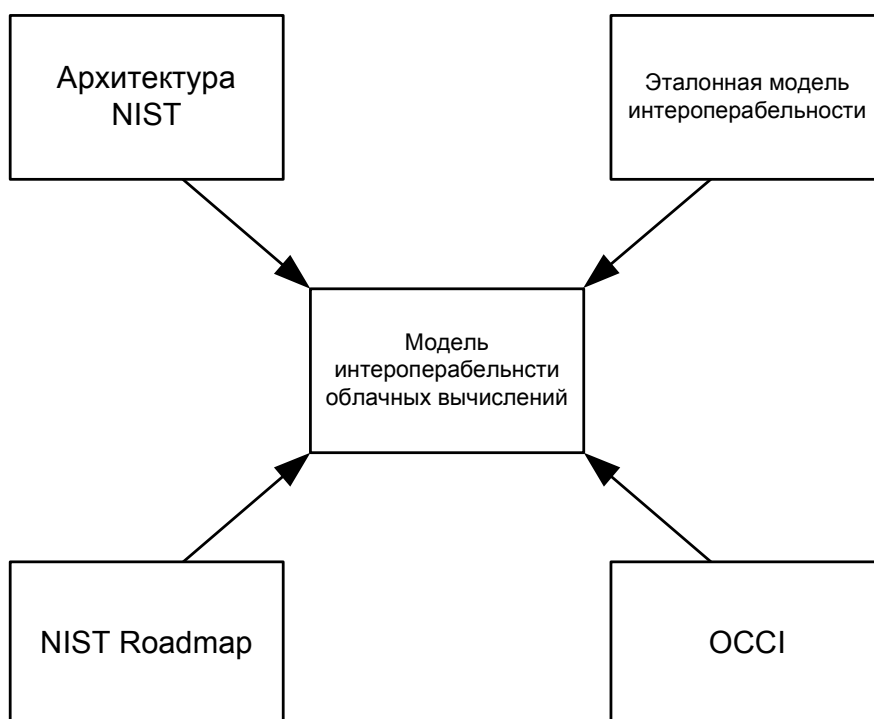


Рис. 2 Синтезированная модель.

Оговоримся, что в данной работе используются три равнозначных термина:

«пользователь», «потребитель», «клиент».

Введем несколько определений, данных в [5].

*Ресурс*: физический или виртуальный объект с ограниченной доступностью. Физические ресурсы представляют собой вычислительные, запоминающие и коммуникационные ресурсы. Виртуальные ресурсы представляют собой обычно сервисы, которые предоставляют прямо или косвенно доступ к физическим ресурсам.

*Сервис*: объект, который организует доступ к назначению ресурса или позволяет выполнить действие на ресурсе. Сервисы в свою очередь могут быть сервисами низкого уровня, которые работают в основном на физических ресурсах или сервисами высокого уровня, которые работают с виртуальными ресурсами (т.е. с другими сервисами). Сервисы проявляют своё назначение посредством интерфейса сервиса.

*Система*: набор сервисов и ресурсов, интегрированных в целое. Система может быть создана путём интеграции нескольких систем. Системы высокого уровня образуются из систем низкого уровня методом агрегации.

*Семантики (систем)*: набор возможностей и функций, доступных в системе. Семантика в системе вместе взятых систем низкого уровня может быть больше (более мощная), чем семантика отдельных ее (нижний уровень) составляющих систем. Семантики ресурса состоят из набора основных возможностей спецификации к ресурсу и способностей управлять этими возможностями (предоставления, доступность QoS, безопасности и т.д.)

*Интерфейс системы*: набор интерфейсов, который дает возможность приложению (и другой системе высокого уровня) получать доступ к возможностям системы. API предоставляют программный доступ к этим интерфейсам. Прикладные среды обеспечивают абстракции пользовательского уровня к API, и, следовательно, доступ к сервису интерфейсов. Системные интерфейсы часто используют только часть всех семантик системы.

*Виртуализация*: дополнительный слой между реальными системами и

приложениями, который переводит параллельный доступ к реальным системам в казалось бы, эксклюзивный доступ к виртуальной системе. Интерфейс виртуализации часто скрывает детали и различия реальных компонентов системы.

*Приложение:* объект с использованием системы, например с помощью API или прикладной среды (см. ниже).

*Порталы и научные входы:* прикладная среда высокого уровня, которая ориентирована в направлении облегчения конечного применения и служит интерфейсами доступа. Эти интерфейсы лежат в основе написания задания, написания исходной программы, развёртывания и управления приложением, как предполагаемым, так и реальным в данной системе. Прикладная среда может предоставлять дополнительные часто специфичные для приложения семантики, которые не могли быть предусмотрены в базовой системе.

## **1. Архитектура облачных вычислений**

Эталонная архитектура облачных вычислений NIST является, по сути, обобщённой концептуальной моделью высокого уровня и служит эффективным инструментом при формировании требований, структур и действий облачных вычислений. Архитектура определяет состав участников, деятельность и функции, которые могут быть реализованы в процессе разработки архитектур облачных вычислений и устанавливает взаимоотношения между участниками облачных вычислений. Она содержит некоторые примеры и описания, являющиеся основой при обсуждении требований и стандартов для облачных вычислений. На рисунке 3 показаны сервисы, с точки зрения задачи данной работы представляет интерес элемент «переносимость/интероперабельность».

Однако, вычисления являются только одной из областей применения «облаков». Второй областью является хранение информации. И, наконец, третьей областью является гибридное использование облаков: для сбора, хранения данных и в качестве средства обработки данных. Мы остановимся на проблеме

интероперабельности облаков с целью раскрыть содержание Рис.3.

Как видно, архитектура облачных вычислений NIST содержит пять главных действующих субъектов: облачный потребитель, облачный провайдер, облачный аудитор, облачный брокер, облачный оператор связи.



Рис. 3 Эталонная архитектура облачных вычислений NIST [6].

Рассмотрим, какие сервисы доступны потребителям IaaS, PaaS и SaaS.

### 1.1. Сервисы, доступные пользователю

На рисунке 4 показаны примеры сервисов для потребителей SaaS, PaaS и IaaS. Как следует из рисунков 3 и 4, все проблемы обеспечения потребителей требуемыми сервисами решает провайдер (поставщик). Следовательно, вопросы интероперабельности тоже решает провайдер.

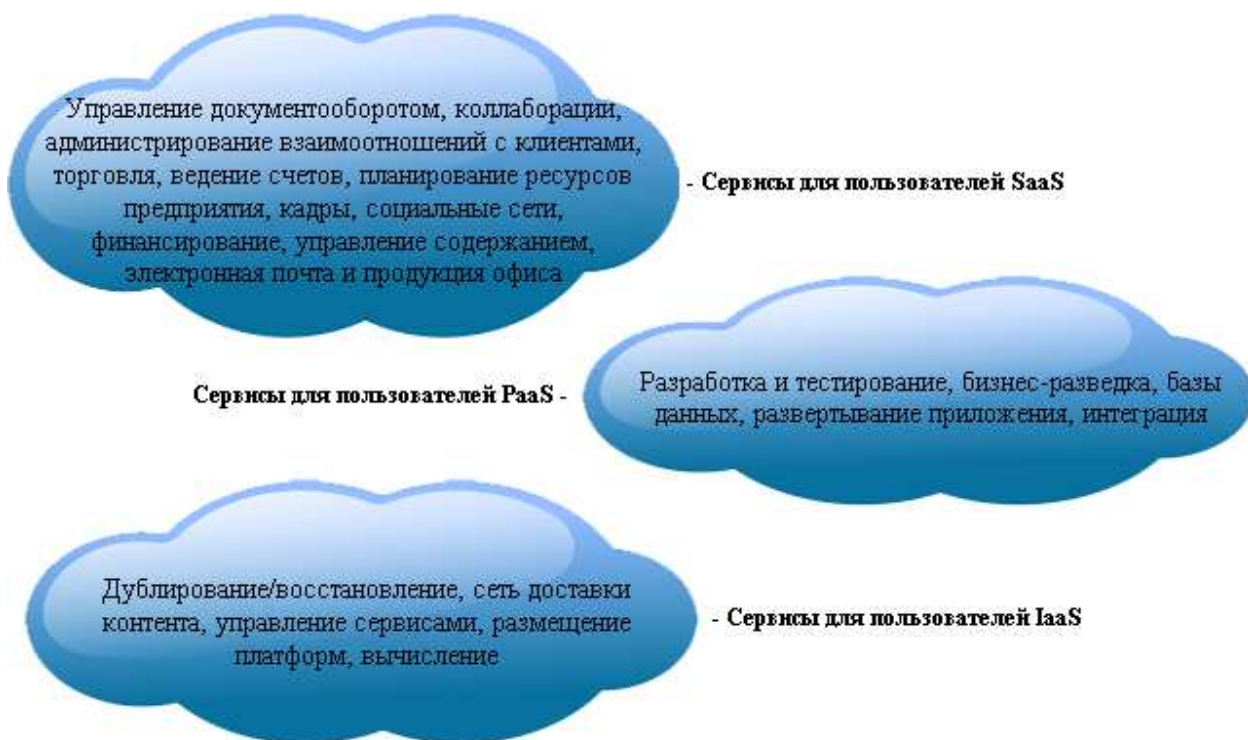


Рис. 4 Примеры сервисов для пользователей облачных вычислений [6].

### 1.2. Функции провайдера

На рисунке 5 показаны основные функции провайдера: развёртывание сервиса, оркестровка сервиса, облачный сервис-менеджмент, безопасность и конфиденциальность.

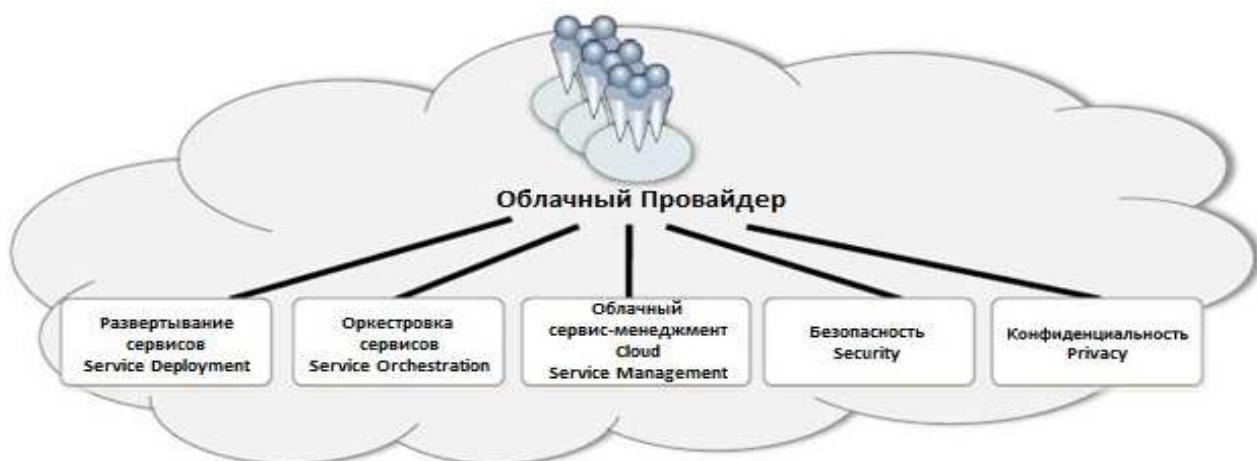


Рис.5 Основные функции провайдера облака [6].

*Развертывание сервиса* состоит в сборке ресурсов.

Оркестровка сервиса показана на рисунке 6.



Рис. 6 Сервис оркестровки провайдера облака [6].



Рис. 7 Сервис управления облаком провайдера облака [6].

Облачный сервис-менеджмент включает все сервис-ориентированные функ-



ции, которые необходимы для администрирования и действия затребованных сервисов или которые могут оказаться затребованными клиентами облака. Как показано на Рис. 7, сервис управления облака включает описание с точки зрения поддержки бизнеса, сборки и конфигурирования и с точки зрения удовлетворения требований переносимости и интероперабельности. Это подчёркивает ответственность провайдера за обеспечение интероперабельности между потребителем облачных сервисов и собственно провайдером.

## 2. Интерфейсы облаков

Интерфейсы, предоставленные потребителям облака, могут быть разделены на два главных типа, в которых интероперабельность задаётся отдельно для каждого типа. Как показано на рисунке 8, каждый вид облачной модели предоставления услуг содержит интерфейс к каждому типу [4].



Рис. 8 Структура сервиса облака.

Функциональный интерфейс (ФИ), который представляется к (или самой) части (частью) облака, выполняет главную функцию сервиса облака. В этом его отличие от интерфейса, который применяется для управления использованием сервиса облака (API).

### 2.1. IaaS-интерфейс

В случае IaaS (см. рис.9), ФИ представляет собой виртуализованные централь-

ный процессор (ЦП), память и пространство ввода/вывода, обычно применяемые операционной системой, и стек ПО, работающий с этой операционной системой.

Пользователь облака использует интерфейс администратора, чтобы управлять самому сервисом облака при старте, остановке и манипулировании образом виртуальной машины и подключенными к ней ресурсами.



Рис. 9 IaaS-интерфейс.

API IaaS является кандидатом на то, чтобы стать стандартным средством обеспечения интероперабельности. Примером стандарта интерфейса управления ресурсами IaaS является интерфейс, предложенный рабочей группой Open Cloud Computing Interface (OCCI) из международной организации Open Grid Forum. Стандарт интерфейса администрирования данных – Cloud Data Management Interface (CDMI) является примером интерфейса, как для администрирования хранения, так и функциональным интерфейсом хранилища.

## 2.2. PaaS-интерфейс

Для PaaS, как показано на рис. 10 [7], мы снова видим необходимость в разделении интерфейсов на два типа.



Рис. 10 PaaS-интерфейс.

ФИ PaaS представляет собой среду выполнения программ с использованием библиотек и компонентов, необходимых для приложения. API PaaS, однако, может быть очень похож на интерфейс управления IaaS модели. Интероперабельность интерфейсов управления самообслуживания может быть достигнута отдельно от интероперабельности функциональных интерфейсов PaaS.

### **2.3. SaaS-интерфейс**

В таком случае, когда SaaS-приложение заказывается с помощью Web-программы (рис. 11), возникает множество стандартов, которые используются для достижения интероперабельности между (что существенно) Web-сервером и программой пользователя, например, IP (v4, v6), TCP, HTTP, SSL/TLS, HTML, XML, REST, Atom, AtomPub, RSS и JavaScript/JSON. Ни один из этих стандартов не специфицирован под облако, и, тем не менее, эти стандарты применяются в Web-браузерах, положенных в основу API.

В случае, когда SaaS-приложение заказано другой системой в качестве сервиса, облака или чем-то подобным, существуют различные стандарты, как на содержание данных, так и на интерфейсы. Самыми важными для интероперабельности являются канонические форматы данных, обычно сегодня представляемыми стандартами XML.

Существуют различные стандарты, отвечающие различным прикладным областям (т.е. OAGi BODs для бизнес-документов или ODF и OOXML для офисных документов). Также важным является комплект стандартов интероперабельности на интерфейсы, упаковку и транспортировку данных, примером которых служат SOAP, WS-\* и ebXML.



Рис. 11 SaaS-интерфейс.

### 3. Общий вид модели

По нашим представлениям не имеется точных определений облаков и грид-систем, и не определено соотношение между этими понятиями, несмотря на значительное число публикаций [8,9,10]. Однако, на наш взгляд, фундаментальным различием между облаками и гридами, является поддержка различных интерфейсов – синтаксически простых, семантически ограниченных и высокого уровня.

Как и грид-системам, системам облачных вычислений присущи следующие свойства: поддержка гетерогенной среды, разделение ресурсов, оптимизация, контроль качества обслуживания, выполнение задач, работа с данными, безопасность, упрощенное администрирование, масштабируемость, работоспособность, использование и расширяемость [11].

Возвращаясь к рисункам 1 и 3, можем констатировать, что для облачных вычислений проблемы интероперабельности решаются на всех уровнях поставщиком сервиса с помощью нижеследующих сервисов, которые формируются на основе требований потребителей (Use Case) и реализуют свойства, описанные выше [12,13]:

- 1) Сервисы инфраструктуры. Сервисы этой группы обеспечивают внутреннюю целостность облака. Они решают задачи внутренней безопасности, создания и управления пространством имен ресурсов, поддержки системы описания состояния ресурса.
- 2) Сервисы управления выполнением. Сервисы управления выполнением обеспечивают обнаружение в облаке подходящего для предъявленного

пользователем задания ресурса, выполняют выделение выбранного ресурса, подготавливают его к запуску приложения, обеспечивают инициализацию процесса выполнения и управление им.

- 3) Сервисы управления данными. Эти сервисы используются для обеспечения перемещения массивов данных между отдельными ресурсами, обработки запросов, выполнения обновления данных, а также объединения данных, размещенных на географически разделенных ресурсах.
- 4) Сервисы управления ресурсами. Эта группа объединяет в себе сервисы, которые используются для управления: а) собственно физическими и логическими ресурсами; б) ресурсами, уже объединенными в облако, с точки зрения доступа к этим ресурсам пользователя; в) инфраструктурой облака.
- 5) Сервисы безопасности. Сервисы этой группы используются для обеспечения безопасности в рамках виртуальной организации – поставщика услуг.
- 6) Сервисы самоуправления. Данные сервисы реализуют задачу поддержания целостности облака при минимизации затрат на восстановление системы после сбоев, проведение диагностики системы в целом.
- 7) Информационные сервисы. Основная задача сервисов этой группы сводится к реализации механизма передачи информации о свойствах ресурса (со стороны поставщика услуг) и о свойствах задания (со стороны потребителя).

Open Grid Forum принял предложение группы Open Cloud Computing Initiative, оформленное в виде документов. На рисунке 12 изображено место ОССИ в конфигурации провайдера [7,14,15].



Рис. 12 Место OCCI в конфигурации провайдера. [7]

Следует заметить, что интерфейс OCCI разработан с целью использовать облачный сервис IaaS.

Потребителями сервиса могут быть как конечные пользователи, так и системы. OCCI подходит для обоих случаев. Главная особенность – то, что OCCI может использоваться в качестве API для всех видов облаков.

Открытый интерфейс облачных вычислений (OCCI) является RESTful протоколом и API для всех видов управленческих задач. OCCI первоначально был инициирован для создания удаленного API управления IaaS на основе модели услуг, дающий возможность разработки инструментов интероперабельности для общих задач, включая развертывание, «вегетативное» масштабирование и мониторинг. Эта версия Открытого интерфейса облачных вычислений пригодна для работы со многими моделями: кроме IaaS включают PaaS и SaaS. Текущая спецификация состоит из трех документов. Эта спецификация описывает версию 1.1 OCCI. Будущие версии OCCI могут включать дополнительные преобразования и расширения спецификаций. Документы этого комплекта спецификаций OCCI

следующие:

- OCCI Core – описывает формальное определение базовой модели OCCI [3].
- OCCI HTTP Rendering – содержит способы взаимодействия с моделью OCCI, используя RESTful OCCI API [14]. В документе определено, как OCCI Core Model может осуществлять связи и формировать последовательности, используя HTTP протокол.
- OCCI Infrastructure – содержит определение расширения инфраструктуры OCCI для домена IaaS [15]. Документ определяет дополнительные типы ресурсов, их атрибуты и действия, которые могут быть приняты для каждого типа ресурса.

Одним из применений облачных вычислений в качестве IaaS не исключается взаимодействие с грид. В таком случае мы видим, что для взаимодействия с грид применим разработанный в ИРЭ РАН им. В.А. Котельникова стандарт ГОСТ Р 55022-2012 «Информационная технология. Спецификация языка описания представления задач (JSDL). Версия 1.0»

Исходя из описанных выше свойств облака, наборов сервисов, необходимых для их реализации и рис. 12, составлена модель интероперабельности для облаков [11] (Рис.13). Как видно, модели интероперабельности для облаков и для грид-систем в целом похожи. На этом рисунке по оси абсцисс перечислены группы сервисов, обеспечивающих реализацию свойств облаков. По оси ординат представлены три уровня технический, семантический и организационный интероперабельности сервисов. По оставшемуся направлению перечислены свойства интерфейса обеспечения интероперабельности в облачной среде. Выше изображен элемент «Интерфейс-API», способный управлять сервисами.

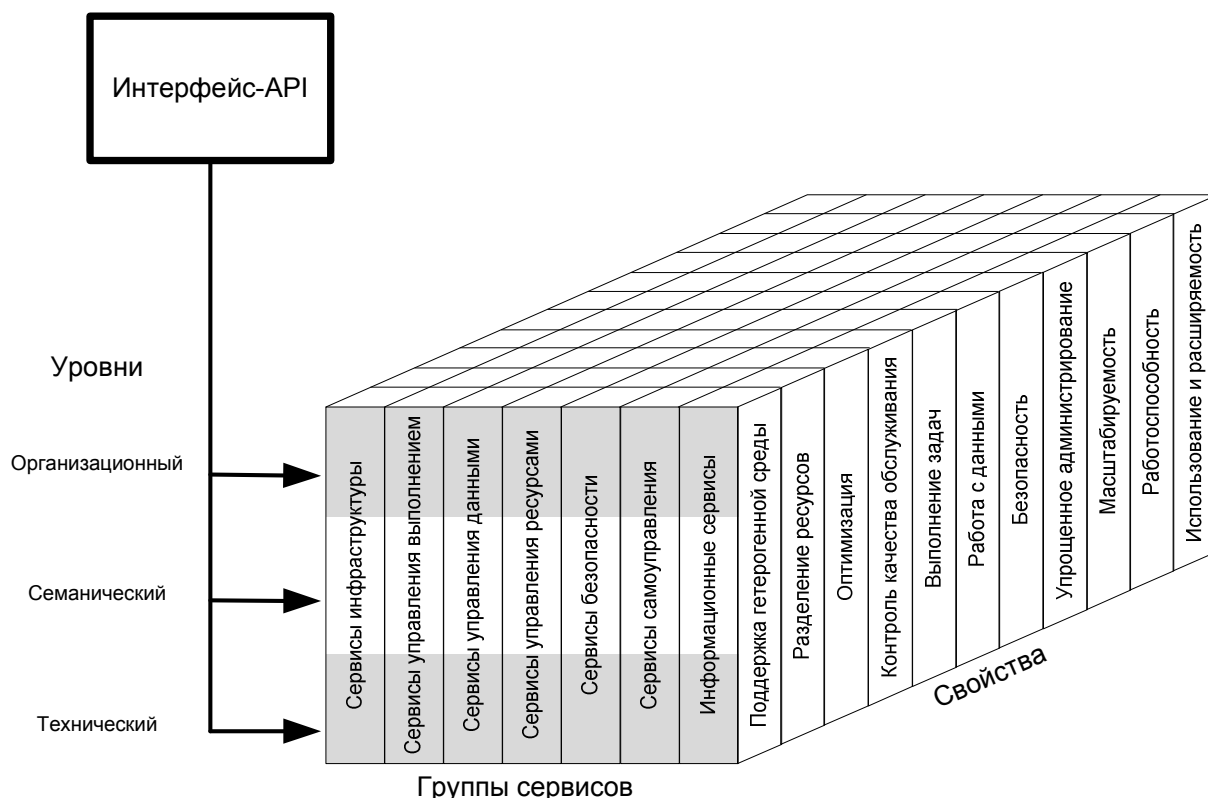


Рис.13 Модель обеспечения интероперабельности облачных вычислений.

#### 4. Заключение

С целью построения модели интероперабельности облачных вычислений рассмотрены модели сервисов. Показано, что модель интероперабельности облачных вычислений по общему виду совпадает с моделью интероперабельности для грид-систем, но может иметь дополнительные свойства. В отличие от модели интероперабельности грид-систем, в модели для облаков присутствует элемент, названный нами «Интерфейс-API», цель которого обеспечить интероперабельность при облачных вычислениях. В основу данного элемента может лечь разработанный рабочей группой OCCI международного общества Open Grid Forum стандартный интероперабельный интерфейс.



## Литература

1. ГОСТ Р 55062-2012 Системы промышленной автоматизации и их интеграция. Интероперабельность. Основные положения [Электронный ресурс] // Центр открытых систем ИРЭ РАН. Создание и внедрение профилей на основе технологии открытых систем: [сайт]. [2012]. URL: [http://opensys.info/files/data\\_20130514161145.pdf](http://opensys.info/files/data_20130514161145.pdf) (дата обращения: 19.06.2013).
2. Журавлев Е.Е., Иванов С.В., Каменщиков А.А., Олейников А.Я., Разинкин Е.И., Рубан К.А. Интероперабельность в облачных вычислениях [Электронный ресурс] // Журнал радиоэлектроники: [сайт]. [2013]. URL: <http://jre.cplire.ru/koi/sep13/4/text.html> (дата обращения: 19.11.2013).
3. Wendt M. A Standards-based Approach To Cloud Interoperability [Электронный ресурс] // OW2 Consortium <http://ow2.org/>: [сайт]. [2012]. URL: [http://ow2.org/view/Events/OW2\\_Berlin\\_Day\\_2012](http://ow2.org/view/Events/OW2_Berlin_Day_2012) (дата обращения: 27.06.2013).
4. NIST Cloud Computing Standards Roadmap [Электронный ресурс] // National Institute of Standards and Technology <http://www.nist.gov/index.html>: [сайт]. [2011]. URL: [http://www.nist.gov/customcf/get\\_pdf.cfm?pub\\_id=909024](http://www.nist.gov/customcf/get_pdf.cfm?pub_id=909024) (дата обращения: 27.05.2013).
5. Using Clouds to Provide Grids Higher-Levels of Abstraction and Explicit Support for Usage Modes. GFD-I.150 [Электронный ресурс] // Open Grid Forum <http://ogf.org/>: [сайт]. [2009]. URL: [www.ogf.org/documents/GFD.150.pdf](http://www.ogf.org/documents/GFD.150.pdf) (дата обращения: 19.11.2013).
6. NIST Cloud Computing Reference Architecture [Электронный ресурс] // National Institute of Standards and Technology <http://www.nist.gov/index.html>: [сайт]. [2011]. URL: [http://www.nist.gov/customcf/get\\_pdf.cfm?pub\\_id=909505](http://www.nist.gov/customcf/get_pdf.cfm?pub_id=909505) (дата обращения: 27.05.2013).
7. Open Cloud Computing Interface - Core. GFD-P-R.183 [Электронный ресурс] // Open Grid Forum <http://ogf.org/>: [сайт]. [2011]. URL: <http://ogf.org/documents/GFD.183.pdf> (дата обращения: 19.11.2013).
8. Grid and Cloud Operations Interoperability – An overview | ZENODO [Электронный ресурс] // ZENODO <http://zenodo.org/>: [сайт]. [2013]. URL: <http://zenodo.org/record/6674?ln=en#.UcyGEJwlGJs> (дата обращения: 27.05.2013).
9. Begin M.E., Burne C., Mary Q., Grey F., O'Neill N., Pearce S. Grids and Clouds: a Comparison [Электронный ресурс] // International Symposium on Grid Computing

- 2009 <http://event.twgrid.org/isgc2009/>: [сайт]. [2009]. URL: <http://event.twgrid.org/isgc2009/program.htm> (дата обращения: 27.06.2013).
10. Ian F., Yong Z., Ioan R., Shiyong L. Cloud Computing and Grid Computing 360-Degree Compared [Электронный ресурс] // Microsoft Academic Search <http://academic.research.microsoft.com/>: [сайт]. [2008]. URL: <http://academic.research.microsoft.com/Publication/50721241> (дата обращения: 27.06.2013).
11. Журавлев Е.Е., Корниенко В.Н., Олейников А.Я., Широбокова Т.Д. Модель открытой Грид-системы [Электронный ресурс] // Журнал радиоэлектроники <http://jre.cplire.ru>: [сайт]. [2012]. URL: <http://jre.cplire.ru/koi/dec12/3/text.html> (дата обращения: 19.11.2013).
12. Open Cloud Computing Interface - Use cases and requirements for a Cloud API. GFD-I.162 [Электронный ресурс] // Open Grid Forum <http://ogf.org>: [сайт]. [2010]. URL: <http://www.ogf.org/documents/GFD.162.pdf> (дата обращения: 19.11.2013).
13. Журавлев Е.Е., Корниенко В.Н., Олейников А.Я. Исследование особенностей проблемы интероперабельности в GRID технологии и технологии облачных вычислений // Исследование особенностей проблемы интероперабельности в GRID технологии и технологии облачных вычислений. Дубна. 2012. С. 312-320.
14. Open Cloud Computing Interface - Infrastructure. GFD-P-R.184 [Электронный ресурс] // Open Grid Forum <http://ogf.org>: [сайт]. [2011]. URL: <http://ogf.org/documents/GFD.184.pdf> (дата обращения: 19.11.2013).
15. Open Cloud Computing Interface - RESTful HTTP Rendering. GFD-P-R.185 [Электронный ресурс] // Open Grid Forum <http://ogf.org>: [сайт]. [2011]. URL: <http://ogf.org/documents/GFD.185.pdf> (дата обращения: 19.11.2013).
16. Enslow P.H.J., "What is a "Distributed" Data Processing," *Computer*, Vol. 11, No. 1, Jan 1978. pp. 13-21.