

## РАЗРАБОТКА МУЛЬТИОБЛАЧНОГО СЕРВИСА ДЛЯ МИГРАЦИИ ОБЛАЧНЫХ РЕСУРСОВ

Р. Р. Галиев<sup>1</sup> [0000-0002-5404-7088], К. А. Григорян<sup>2</sup> [0000-0001-6470-1832]

<sup>1, 2</sup>Казанский (Приволжский) федеральный университет, ул. Кремлевская,  
35, г. Казань, 420008

<sup>1</sup>galiffun@gmail.com, <sup>2</sup>karigri@yandex.ru

### ***Аннотация***

Важным фактором взрывного развития цифровой экономики последнего десятилетия стали облачные платформы и сервисы. Возможность быстрого масштабирования сервиса в совокупности со снижением инвестиционных затрат на старте проектов в рамках подходов Iaas, PaaS, SaaS дала положительные результаты и легла в основание новых бизнес-моделей и корпоративных решений.

В данной статье мы обсуждаем причины значимости мультиоблачности и исследуем подходы к интеграции облачных сервисов в мультиоблачной архитектуре. Также предложен способ решения проблемы облачной миграции — разработка системы для миграции облачных ресурсов между облачными сервисами.

***Ключевые слова:*** мультиоблачность, облачная миграция, облачные функции, бессерверные вычисления.

### **ВВЕДЕНИЕ**

С ростом числа пользователей информационных систем встает вопрос о росте производительности и масштабируемости системы. Из-за простоты и выгоды многие компании предпочитают использовать горизонтальное масштабирование, увеличивая число экземпляров приложения. Для развертывания новых экземпляров приложения необходимо больше вычислительных мощностей, которые требуют дополнительных затрат. Данную проблему помогают решить облачные сервисы, предоставляющие платные услуги по использованию вычислительных ресурсов. В последние годы их популярность возрастает – согласно исследованию, проведенному компанией Gartner, в 2021 году затраты на облачные ре-

сурсы должны были возрасти на 23% [1]. Ведущими представителями зарубежного рынка являются Amazon, Google, Microsoft и Alibaba [2]. В последние годы российский рынок облачных решений активно развивается, его крупными представителями являются Яндекс, Сбер и МТС [3].

Несмотря на все плюсы, использование облачных сервисов несет ряд недостатков и рисков для информационной системы [4]. Важными недостатками являются:

1. Зависимость от провайдера: применение только одного облачного сервиса лишает информационные системы гибкости. Так, например, некоторые поставщики могут не предоставлять определенные SaaS услуги. Другим критическим риском является полная зависимость инфраструктуры информационной системы от облачного сервиса – в случае непредвиденных ситуаций возможен рост стоимости и потеря данных [5].

2. Несоответствие законодательным актам: некоторые страны имеют особые законодательные акты, связанные с хранением данных пользователей, являющихся гражданами данной страны. Например, федеральный закон №242-ФЗ [6] ограничивает место хранения и обработки персональных данных граждан России ее территорией. Сервера Microsoft Azure не расположены на территории РФ, поэтому можно использовать данный облачный сервис через партнера поставщика облачных решений [7].

Из-за большого числа рисков и недостатков некоторые компании используют несколько публичных облаков вместо одного. Такое решение называется мультиоблачность – использование более одного облачного сервиса [4], [8]. Данный подход популярен – 61% компаний использует более одного облачного провайдера (согласно исследованию, которое было проведено компанией EMA в 2017 году [9], [10]). Более новые исследования говорят о том, что 92% компаний используют услуги более 1 поставщика, а среднее число используемых облачных сервисов составляет 2.6 [11]. Согласно другому исследованию, к 2022 году более 90% компаний будут использовать несколько облачных сервисов [12]. Плюсами использования нескольких поставщиков облачных услуг являются гибкость, надежность и доступность, а также оптимизация стоимости.

## ОБЗОР ЛИТЕРАТУРЫ

На сегодняшний день на практике применяется большое число различных способов построения и управления жизненным циклом мультиоблачных приложений. Все эти способы можно условно отнести к двум категориям на основании того, используется в них или нет оркестрация контейнеров при интеграции различных облачных сервисов. Далее описаны некоторые способы интеграции из каждой выделенной категории.

**Способы интеграции, использующие оркестрацию контейнеров.** Способы интеграции, принадлежащие к данной категории, основаны на оркестрации контейнеров. В последние годы все больше компаний использует данную технологию – число контейнеров под управлением оркестратора выросло до 90%, начиная с 2018 года [13]. Популярными также являются as service услуги, связанные с оркестраторами [13].

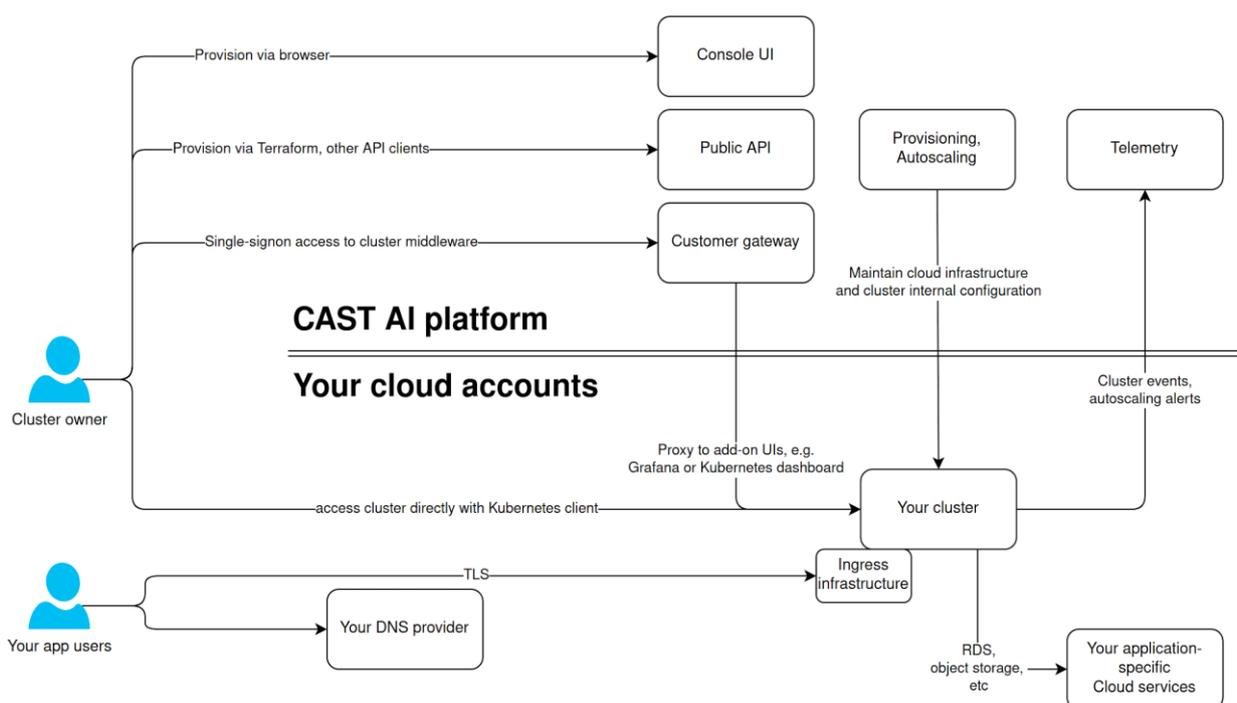


Рис. 1. Высокоуровневая архитектура cast.ai. Источник: Cluster architecture – CAST AI Documentation. URL: <https://docs.cast.ai/concepts/cluster-architecture/>

**1. Cast.ai** (рис. 1) –это сервис, запущенный в 2020 году, который позволяет оптимизировать использование облачных услуг с помощью использования технологий машинного обучения [14]. Оптимизация производится как для стоимости,

так и для мощностей кластера. Сервис позволяет как подсоединить существующий managed Kubernetes кластер (с поддержкой AWS, Azure, Google Cloud Platform) к движку CAST, так и создать новый. Сетевая связность кластера между различными облачными сервисами обеспечивается за счет создания частных сетей и VPN, позволяющих узлам общаться между собой.

**2. Google Anthos** (рис. 2) – это платформа управления гибридными приложениями, развернутыми в облачных сервисах, которая поддерживает контейнеры, микросервисы и Function-as-a-Service. Архитектурно Anthos представляет собой несколько сервисов, которые позволяют мигрировать существующим приложениям в облачную инфраструктуру, а также управлять развертыванием и взаимодействием приложений в мультиоблачной среде. За счет использования Anthos пользователи Google Cloud Platform могут управлять кластерами Kubernetes, узлы которых могут быть расположены в различных публичных облаках или даже локальных датацентрах. Взаимодействие сервисов, находящихся в разных облаках, осуществляется за счет сервисной сети Istio.

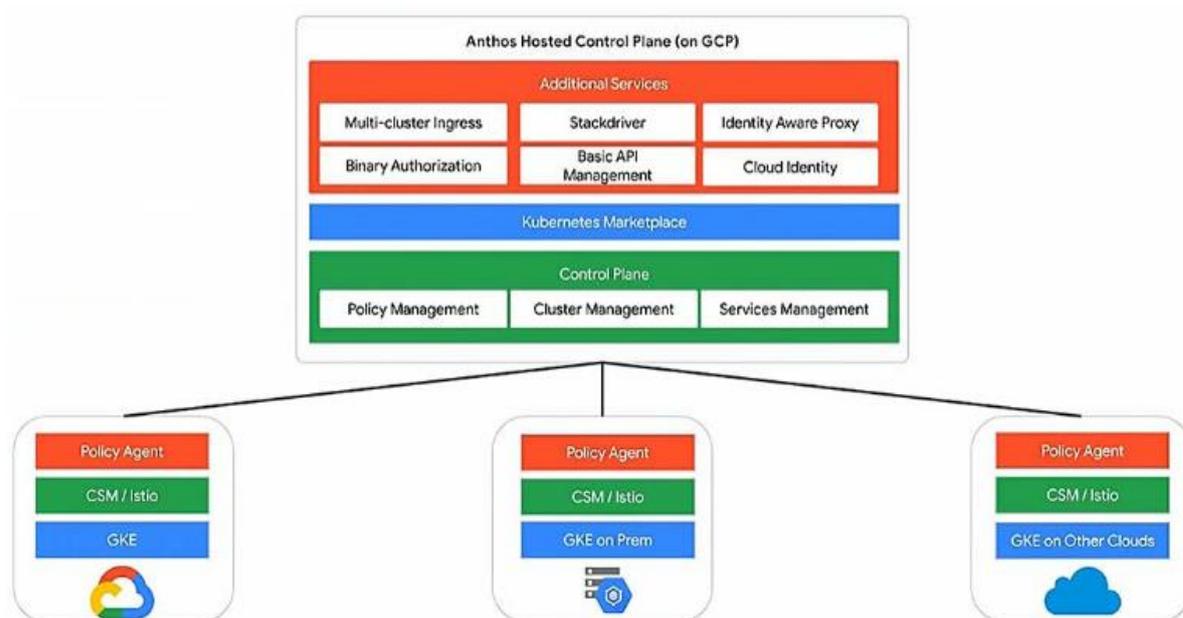


Рис. 2. Компоненты Athos. Источник: Reference Architecture for Google Cloud's Anthos with Lenovo ThinkAgile VX. URL: <https://lenovopress.com/lp1215.pdf>

**Способы интеграции, не использующие оркестрацию контейнеров.** Для работы данных способов не нужен работающий кластер оркестратора контейнеров, их представители могут использовать математические модели, сетевые конфигурации или шаблоны проектирования.

**1. Частные сети и каналы связи.** Популярность мультиоблачности привела к появлению сервисов, предоставляющих услуги по созданию частных соединений 3 уровня OSI, которые обеспечивают сетевую связность между облачными сервисами, а также безопасность за счет прохода трафика через частную сеть [15]. При этом требования к безопасности для некоторых информационных систем не позволяют трафику проходить через посторонние частные сети, не являющиеся сетью облачного провайдера. В таком случае для обеспечения сетевой связности можно использовать частные каналы связи, например, AWS Direct Connect и Azure ExpressRoute. При использовании частного канала связи конфигурация маршрутизации трафика в частной сети пользователя обеспечивает связь между облачными сервисами.

**2. Melodic Cloud** (рис. 3). Melodic Cloud – это платформа для автоматизации развертывания приложений, в том числе в различных облачных сервисах. Данный программный продукт также позволяет оптимизировать работу сервиса, улучшая в первую очередь производительность и надежность, а не стоимость. Для запуска приложения необходимо описать его инфраструктуру при помощи DSL языка CAMEL, который позволяет описать процесс развертывания, а также нефункциональные требования, такие как безопасность и масштабируемость [16–18]. За счет использования языка можно реализовать инфраструктуру без оркестрации контейнеров. После описания модели строится математическая модель, которая решается при помощи программирования в ограничениях [19, 20], стохастических обучающих автоматов и методов машинного обучения. На основе решения модели вычисляется конфигурация развертывания приложения.



Рис. 3. Архитектура платформы Melodic. Источник: Complete set of documentation to start using multicloud and MELODIC. URL: <https://melodic.cloud/guidebooks/>

## МЕТОДОЛОГИЯ

В данном разделе описан предлагаемый нами подход к миграции облачных функций в мультиоблачной среде. Для исходного языка функций были выбраны два языка – Java и Python. Выбранные облачные сервисы – AWS и Yandex.Cloud.

Для получения метаданных об облачной функции необходимо сделать запрос к API облачного сервиса. В AWS для этого необходимо сделать запрос List-Functions для получения списка созданных Lambda функций. Собрав метаданные функций, нужно получить их код. Для этого потребуется запрос GetFunction, где из объекта Code в ответе от API можно получить ссылку на S3 хранилище с архивом, в котором лежит JAR архив приложения вместе с его зависимостями. Непосредственно код функции можно получить, достав данные из архива, используя имена класса и метода, указанные в свойстве Handler объекта Configuration.

Для использования облачной функции в другом сервисе необходимо адаптировать ее код к требованиям данного сервиса. Конвертация кода зависит от используемого языка программирования. Для Python необходимо просто заменить объект контекста облачной функции на другой. Для этого нужно создать адаптер контекста AWS к контексту Yandex.Cloud. Подменяется контекст за счет использования декораторов в Python. Для языка программирования Java при переносе

кода функции помимо замены класса контекста необходимо реализовать иной интерфейс. В случае AWS облачные функции реализуют функциональный интерфейс `RequestHandler<Request, Response>`, а в случае `Yandex.Cloud` функции реализуют функциональный интерфейс `YcFunction<Request, Response>`. Таким образом, для миграции функции необходимо привести функцию к другому интерфейсу, изменив как имя наследуемого интерфейса, так и имя реализуемого метода. Для работы с контекстом можно использовать шаблон проектирования Адаптер, адаптируя функции для работы с контекстом из SDK одного облачного сервиса к другому. Если же для некоторой функции контекста нет аналога, то полноценно перенести функцию невозможно. В таком случае пользователю выдается ошибка с указанием на строку, в которой найдена несовместимость. Такие же проблемы могут встретиться при использовании других SDK облачных сервисов.

После конвертации кода необходимо создать функцию в `Yandex.Cloud`, используя метод `create` ресурса `Function`. В случае успешной конвертации необходимо создать версию функции, используя полученный код.

Другим способом автоматического создания облачных функций без их поведения является конвертация существующих функций одного облачного сервиса в их описание для `Terraform`, с последующей конвертацией конфигурации в формат другого.

Вышеописанные подходы можно применить и для других языков программирования, например, `Ruby` и `C#`.

## **РЕЗУЛЬТАТЫ**

В итоге изучения различных технологий и принципов интеграции был разработан и протестирован прототип системы для миграции облачных ресурсов из `AWS` в `Yandex.Cloud` и обратно (рис. 4). Работа системы состоит из нескольких этапов:

1. Выгрузка метаданных существующих облачных ресурсов в формат `Terraform`;
2. Конвертация описания ресурсов в формат другого облачного сервиса;
3. Создание облачных ресурсов при помощи `Terraform`;
4. Перенос состояния облачных ресурсов, если это возможно.

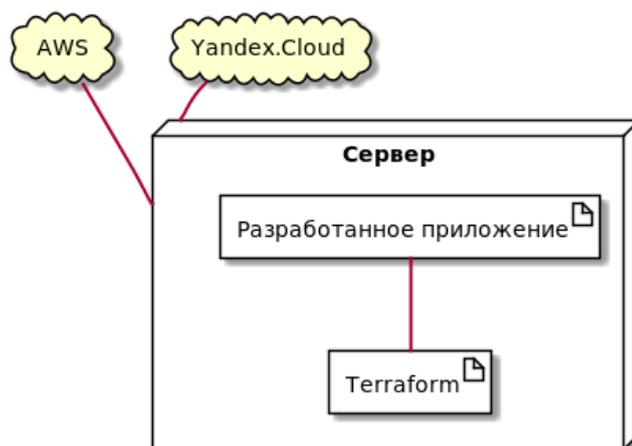


Рис. 4. Диаграмма развертывания системы

## ЗАКЛЮЧЕНИЕ

Проведенное исследование показало, что способы интеграции мультиоблачных приложений делятся на те, что используют оркестрацию контейнеров, и на те, что не используют. Установлено, что мультиоблачность набирает большую популярность.

Основным результатом стала разработка прототипа системы для миграции облачных ресурсов с AWS на Yandex.Cloud, с частичным переносом их состояния. При разработке были использованы облачные сервисы и Terraform.

В дальнейшем планируется разработка обратного переноса ресурсов с Yandex.Cloud на AWS.

## СПИСОК ЛИТЕРАТУРЫ

1. Gartner Forecasts Worldwide Public Cloud End-User Spending to Grow 23% in 2021. URL: <https://www.gartner.com/en/newsroom/press-releases/2021-04-21-gartner-forecasts-worldwide-public-cloud-end-user-spending-to-grow-23-percent-in-2021>
2. AWS vs Azure vs Google Cloud Market Share 2021: What the Latest Data Shows – ParkMyCloud. URL: <https://www.parkmycloud.com/blog/aws-vs-azure-vs-google-cloud-market-share/>
3. Облачные провайдеры в 2021 году. Лучшие в России.

URL: [https://ex-hort.ru/oblastnye\\_provajdery\\_2020\\_rossia](https://ex-hort.ru/oblastnye_provajdery_2020_rossia)

4. Hong J. et al. An Overview of multi-cloud computing // Workshops of the International Conference on Advanced Information Networking and Applications. Springer, Cham, P. 1055–1068, 2019.

5. MongoDB to terminate Russian SaaS accounts • The Register.

URL: [https://www.theregister.com/2022/03/15/mongodb\\_terminates\\_russian\\_saas/](https://www.theregister.com/2022/03/15/mongodb_terminates_russian_saas/)

6. Федеральный закон «О внесении изменений в отдельные законодательные акты Российской Федерации по вопросам применения информационных технологий в сфере охраны здоровья» от 29.07.2017 N 242-ФЗ (последняя редакция). URL: [http://www.consultant.ru/document/cons\\_doc\\_LAW\\_221184/](http://www.consultant.ru/document/cons_doc_LAW_221184/)

7. Russian Personal Data Localization Requirements – Microsoft Compliance.

URL: <https://docs.microsoft.com/en-us/compliance/regulatory/offering-russia-data-localization>

8. What is multicloud?

URL: <https://www.redhat.com/en/topics/cloud-computing/what-is-multicloud>

9. What is multi-cloud strategy? Definition from WhatIs.com.

URL: <https://searchcloudcomputing.techtarget.com/definition/multi-cloud-strategy>

10. EMA-Top10-2017-HybridCloud-DecisionGuide.pdf.

URL: <https://www.parkmycloud.com/wp-content/uploads/2017/06/EMA-Top10-2017-HybridCloud-DecisionGuide.pdf>

11. Cloud Computing Trends: 2021 State of the Cloud Report | Flexera Blog.

URL: <https://www.flexera.com/blog/cloud/cloud-computing-trends-2021-state-of-the-cloud-report/>

12. IDC Expects 2021 to Be the Year of Multi-Cloud as Global COVID-19 Pandemic Reaffirms Critical Need for Business Agility.

URL: <https://www.idc.com/getdoc.jsp?containerId=prMETA46165020>

13. 11 Facts About Real-World Container Use.

URL: <https://www.datadoghq.com/container-report-2020/>

14. Cast.ai nabs \$7.7M seed to remove barriers between public clouds – TechCrunch. URL: <https://techcrunch.com/2020/11/25/cast-ai-nabs-7-7m-seed-to-remove-barriers-between-public-clouds/>

15. *Yeganeh B., Durairajan R.* A First Comparative Characterization of Multi-cloud Connectivity in Today's Internet // Lecture notes in computer science. 2020.
16. CAMEL. URL: <http://camel-dsl.org/>
17. *Kritikos K. et al.* Towards the modelling of hybrid cloud applications // 2019 IEEE 12th International Conference on Cloud Computing (CLOUD). IEEE, 2019. P. 291–295.
18. *Achilleos A.P. et al.* The cloud application modelling and execution language // Journal of Cloud Computing. 2019. V. 8, No. 1. P. 1–25.
19. *Horn G. et al.* MELODIC: selection and integration of open source to build an autonomic cross-cloud deployment platform // International Conference on Objects, Components, Models and Patterns. Springer, Cham, 2019. P. 364–377.
20. *Horn G., Skrzypek P.* MELODIC: utility based cross cloud deployment optimisation // 2018 32nd International Conference on Advanced Information Networking and Applications Workshops (WAINA). IEEE, 2018. P. 360–367.

---

## DEVELOPMENT OF A MULTICLOUD SERVICE FOR CLOUD RESOURCE MIGRATION

R. R. Galiev<sup>1</sup> [0000-0002-5404-7088], K. A. Grigoryan<sup>2</sup> [0000-0001-6470-1832]

<sup>1,2</sup>Kazan (Volga Region) Federal University, 35 Kremlevskaya str., Kazan, 420008

<sup>1</sup>galiffun@gmail.com, <sup>2</sup>karigri@yandex.ru

### **Abstract**

Cloud platforms and services have become an important factor in the explosive development of the digital economy in the last decade. The ability to quickly scale the service, coupled with a reduction in investment costs at the start of projects within the framework of the IaaS, PaaS, SaaS approaches, gave positive results and formed the basis of new business models and corporate solutions.

In this article, we discuss the reasons for the importance of multicloud and explore approaches to integrating cloud services in a multicloud architecture. The article

also proposes a way to solve the problem of cloud migration - developing a system for migrating cloud resources between cloud services.

**Keywords:** *multicloud, cloud migration, cloud functions, serverless.*

## REFERENCES

1. Gartner Forecasts Worldwide Public Cloud End-User Spending to Grow 23% in 2021. URL: <https://www.gartner.com/en/newsroom/press-releases/2021-04-21-gartner-forecasts-worldwide-public-cloud-end-user-spending-to-grow-23-percent-in-2021>
2. AWS vs Azure vs Google Cloud Market Share 2021: What the Latest Data Shows – ParkMyCloud.  
URL: <https://www.parkmycloud.com/blog/aws-vs-azure-vs-google-cloud-market-share/>
3. Oblachnye provajdery v 2021 godu. Luchshie v Rossii.  
URL: [https://ex-hort.ru/oblachnye\\_provajdery\\_2020\\_rossia](https://ex-hort.ru/oblachnye_provajdery_2020_rossia)
4. *Hong J. et al.* An Overview of multi-cloud computing // Workshops of the International Conference on Advanced Information Networking and Applications. Springer, Cham, P. 1055–1068, 2019.
5. MongoDB to terminate Russian SaaS accounts • The Register.  
URL: [https://www.theregister.com/2022/03/15/mongodb\\_terminates\\_russian\\_saas/](https://www.theregister.com/2022/03/15/mongodb_terminates_russian_saas/)
6. Federal'nyj zakon "O vnesenii izmenenij v otдел'nye zakonodatel'nye akty Rossijskoj Federacii po voprosam primeneniya informacionnyh tekhnologij v sfere ohrany zdorov'ya" ot 29.07.2017 No. 242-FZ (poslednyaya redakciya). URL: [http://www.consultant.ru/document/cons\\_doc\\_LAW\\_221184/](http://www.consultant.ru/document/cons_doc_LAW_221184/)
7. Russian Personal Data Localization Requirements – Microsoft Compliance.  
URL: <https://docs.microsoft.com/en-us/compliance/regulatory/offering-russia-data-localization>
8. What is multicloud?  
URL: <https://www.redhat.com/en/topics/cloud-computing/what-is-multicloud>
9. What is multi-cloud strategy? Definition from WhatIs.com.  
URL: <https://searchcloudcomputing.techtarget.com/definition/multi-cloud-strategy>
10. EMA-Top10-2017-HybridCloud-DecisionGuide.pdf.

URL: <https://www.parkmycloud.com/wp-content/uploads/2017/06/EMA-Top10-2017-HybridCloud-DecisionGuide.pdf>

11. Cloud Computing Trends: 2021 State of the Cloud Report | Flexera Blog.

URL: <https://www.flexera.com/blog/cloud/cloud-computing-trends-2021-state-of-the-cloud-report/>

12. IDC Expects 2021 to Be the Year of Multi-Cloud as Global COVID-19 Pandemic Reaffirms Critical Need for Business Agility.

URL: <https://www.idc.com/getdoc.jsp?containerId=prMETA46165020>

13. 11 Facts About Real-World Container Use.

URL: <https://www.datadoghq.com/container-report-2020/>

14. Cast.ai nabs \$7.7M seed to remove barriers between public clouds – TechCrunch. URL: <https://techcrunch.com/2020/11/25/cast-ai-nabs-7-7m-seed-to-remove-barriers-between-public-clouds/>

15. *Yeganeh B., Durairajan R.* A First Comparative Characterization of Multi-cloud Connectivity in Today's Internet // Lecture notes in computer science. 2020.

16. CAMEL. URL: <http://camel-dsl.org/>

17. *Kritikos K. et al.* Towards the modelling of hybrid cloud applications // 2019 IEEE 12th International Conference on Cloud Computing (CLOUD). IEEE, 2019. P. 291–295.

18. *Achilleos A.P. et al.* The cloud application modelling and execution language // Journal of Cloud Computing. 2019. V. 8, No. 1. P. 1–25.

19. *Horn G. et al.* MELODIC: selection and integration of open source to build an autonomic cross-cloud deployment platform // International Conference on Objects, Components, Models and Patterns. Springer, Cham, 2019. P. 364–377.

20. *Horn G., Skrzypek P.* MELODIC: utility based cross cloud deployment optimisation // 2018 32nd International Conference on Advanced Information Networking and Applications Workshops (WAINA). IEEE, 2018. P. 360–367.

## СВЕДЕНИЯ ОБ АВТОРАХ



**ГАЛИЕВ Рустем Рамилевич** – магистрант, Казанский (Приволжский) федеральный университет, г. Казань.

**Rustem Ramilevich GALIEV** – Master’s student, Kazan (Volga region) Federal University, Kazan.

email: galiffun@gmail.com

ORCID: 0000-0002-5404-7088.



**ГРИГОРЯН Карен Альбертович** – кандидат экономических наук, доцент, Казанский (Приволжский) федеральный университет, г. Казань.

**Karen Albertovich GRIGORIAN** – Candidate of Economics, Associate Professor, Kazan (Volga region) Federal University, Kazan.

email: karigri@yandex.ru

ORCID: 0000-0001-6470-1832

*Материал поступил в редакцию 28 февраля 2022 года*