

УДК 004.056

<https://doi.org/10.31854/2307-1303-2024-12-4-13-21>

EDN: TYVQYB

## Применение Open Digital Architecture для создания интегрированных цифровых экосистем

Белозеров К. В., Ганюшин Д. С., Зайдуллин Р. Р., Кисляков С. В. ✉

Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича,  
Санкт-Петербург, 193232, Российская Федерация

**Цель работы** – разработка методологии моделирования и развертывания интегрированной цифровой экосистемы на основе концепции Open Digital Architecture для модернизации телекоммуникационных систем. Исследование базируется на использовании современных технологий контейнеризации и оркестрации, включая Docker, Kubernetes и Helm, а также на применении стандартизированных API для обеспечения беспрепятственной интеграции микросервисов. **Методика исследования** предусматривает поэтапное создание и упаковку сервисов в контейнеры, их автоматизированное развертывание в кластере Kubernetes и управление конфигурацией с помощью Helm-чартов. Проведенное тестирование подтвердило корректность взаимодействия компонентов и устойчивость системы при изменении эксплуатационных условий. **Новизна** исследования заключается в комплексном подходе, объединяющем принципы Open Digital Architecture с технологиями автоматизации и оркестрации, что позволяет сократить время внедрения новых сервисов и снизить операционные затраты. **Практическая значимость** работы определяется возможностью применения разработанной модели для оптимизации управления телекоммуникационными сетями, повышения их гибкости и адаптивности к динамичным изменениям нагрузки. **Выводы исследования** указывают на эффективность предложенной методологии для создания адаптивных цифровых экосистем, что открывает перспективы для дальнейшего развития инструментов мониторинга и предиктивной аналитики в условиях современной телекоммуникационной инфраструктуры.

**Ключевые слова:** Open Digital Architecture (ODA), цифровая экосистема, контейнеризация, оркестрация, Kubernetes, Helm, микросервисы

### Актуальность

Современная эпоха характеризуется стремительным развитием цифровых технологий и бурным ростом информационных потоков, что порождает новые требования к телекоммуникационной инфраструктуре. Традиционные монолитные архитектурные решения уже не справляются с необходимостью обеспечения

#### Библиографическая ссылка на статью:

Белозеров К. В., Ганюшин Д. С., Зайдуллин Р. Р., Кисляков С. В. Применение Open Digital Architecture для создания интегрированных цифровых экосистем // Информационные технологии и телекоммуникации. 2024. Т. 12. № 4. С. 13–21. DOI: 10.31854/2307-1303-2024-12-4-13-21. EDN: TYVQYB

#### Reference for citation:

Belozerov K., Ganyushin D., Zaidullin R., Kislyakov S. Application of Open Digital Architecture for Creating Integrated Digital Ecosystems // Telecom IT. 2024. Vol. 12. Iss. 4. PP. 13–21. (in Russian). DOI: 10.31854/2307-1303-2024-12-4-13-21. EDN: TYVQYB

высокой скорости передачи данных, масштабируемости и надежности работы сетей. В условиях, когда число подключенных устройств и объем передаваемой информации постоянно увеличиваются, становится необходимым переход к более гибким и адаптивным системам управления.

В этой связи применение интегрированных цифровых экосистем, основанных на принципах Open Digital Architecture, приобретает особую актуальность. Принципы модульности и стандартизации, заложенные в данной концепции, позволяют объединить разнородные компоненты в единую систему с едиными интерфейсами, что способствует повышению эффективности эксплуатации и снижению затрат на модернизацию инфраструктуры. Современные технологии контейнеризации и оркестрации, такие как Docker, Kubernetes и Helm, обеспечивают автоматизацию процессов развертывания и управления микросервисами, что является ключевым фактором для оперативного реагирования на изменения нагрузки и условий эксплуатации.

Решение проблемы модернизации телекоммуникационных сетей посредством разработки цифровых экосистем является стратегически важным направлением, способствующим улучшению качества обслуживания, повышению устойчивости и адаптивности систем в условиях динамично меняющейся цифровой среды.

### **Постановка задачи**

Целью данного исследования является разработка методологии моделирования и развертывания интегрированной цифровой экосистемы на основе концепции Open Digital Architecture (ODA), которая позволит оптимизировать процессы управления современными телекоммуникационными системами.

Для достижения поставленной цели в работе решаются следующие задачи:

- определение ключевых принципов и компонентов концепции ODA, способствующих созданию масштабируемых и гибких цифровых экосистем;
- разработка методологической основы для моделирования архитектуры, обеспечивающей эффективное взаимодействие микросервисов и стандартизированных API;
- описание этапов контейнеризации, оркестрации и автоматизации развертывания компонентов системы, включая использование современных инструментов, таких как Docker, Kubernetes и Helm;
- анализ преимуществ и потенциальных ограничений предложенной методологии с точки зрения обеспечения устойчивости, безопасности и масштабируемости цифровой инфраструктуры.

### **Обзор концепции Open Digital Architecture**

ODA представляет собой современный подход к формированию цифровых систем, основанный на принципах открытости, стандартизации и модульности. Данный подход формирует основу для создания гибких, масштабируемых и легко адаптируемых платформ, способных объединять разнородные элементы

информационной инфраструктуры посредством универсальных стандартов и четко определенных интерфейсов. В теоретическом плане ODA акцентирует внимание на разделении функциональных блоков, что позволяет создавать независимые модули, способные взаимодействовать друг с другом без необходимости глубокого знания внутренней реализации каждого компонента. Применение единых протоколов обмена данными и управления обеспечивает высокую степень повторного использования разработанных решений, что существенно снижает издержки и временные затраты при модернизации систем.

Концепция опирается на формализацию моделей взаимодействия, где каждая единица системы описывается с использованием универсальных терминов и стандартов. Это способствует интеграции как новых, так и уже существующих технологических решений, обеспечивая прозрачность процессов обновления и поддержки компонентов в рамках единой цифровой экосистемы. Применение данного подхода способствует формированию архитектуры, в которой процессы автоматизации развертывания, управления жизненным циклом сервисов и динамического масштабирования реализуются с высокой степенью автономии и согласованности [1–4].

### Методология и реализация цифровой экосистемы

Разрабатываемая цифровая экосистема основана на модульной архитектуре, где основное внимание уделяется разделению функциональных обязанностей между микросервисами и использованию стандартизированных API. Такая архитектурная модель обеспечивает независимость разработки отдельных компонентов, их легкую интеграцию и масштабирование, что позволяет адаптировать систему к изменяющимся требованиям. Каждый микросервис выполняет конкретную задачу, а взаимодействие между ними осуществляется посредством заранее определенных интерфейсов, что упрощает управление потоком данных и повышает надежность системы [5, 6]. Ключевым элементом реализации является внедрение кластера Kubernetes, который позволяет динамически распределять вычислительные ресурсы и управлять состоянием приложений, развернутых из контейнеров. При этом настройка и контроль над кластером осуществляются с помощью инструментов, позволяющих отслеживать его работоспособность в реальном времени. Для иллюстрации этапа развертывания кластера рекомендуется обратить внимание на результаты команды проверки состояния Minikube, представленные на рисунке 1.

Контейнеризация занимает центральное место в разработке современных цифровых систем. В рамках исследования реализован процесс создания Docker-образов, что позволяет упаковать приложение вместе со всеми его зависимостями в единый переносимый контейнер. После завершения сборки образы проверяются командой, выводящей список локально сохраненных контейнеров. Интеграция этих контейнеров с Kubernetes обеспечивает автоматическую балансировку нагрузки, высокую доступность сервисов и упрощает процесс масштабирования [5, 6]. Пример вывода команды для проверки образов приведен на рисунке 2, что демонстрирует корректность сборки и готовность контейнеров к дальнейшему развертыванию.

```
root@ganyushin-PC:/home/ganyushin# sudo minikube start --force -- driver=docker
minikube v1.35.0 на Ubuntu 22.04
minikube skips various validations when --force is supplied; this may lead to unexpected behavior
Automatically selected the docker driver. Other choices: virtualbox, none, ssh
The "docker" driver should not be used with root privileges. If you wish to continue as root, use --force.
If you are running minikube within a VM, consider using --driver=none:
https://minikube.sigs.k8s.io/docs/reference/drivers/none/
Using Docker driver with root privileges
Starting "minikube" primary control-plane node in "minikube" cluster
Pulling base image v0.0.46 ...
Скачивается Kubernetes v1.32.0 ...
> preloaded-images-k8s-v18-v1...: 333.57 MiB / 333.57 MiB 100.00% 4.71 Mi
> gcr.io/k8s-minikube/kicbase...: 500.31 MiB / 500.31 MiB 100.00% 5.22 Mi
Creating docker container (CPUs=2, Memory=3900MB) ...
Подготавливается Kubernetes v1.32.0 на Docker 27.4.1 ...
  ■ Generating certificates and keys ...
  ■ Booting up control plane ...
  ■ Configuring RBAC rules ...
Configuring bridge CNI (Container Networking Interface) ...
Компоненты Kubernetes проверяются ...
  ■ Используется образ gcr.io/k8s-minikube/storage-provisioner:v5
Включенные дополнения: storage-provisioner, default-storageclass
Готово! kubectl настроен для использования кластера "minikube" и "default" пространства имён по умолчанию
root@ganyushin-PC:/home/ganyushin# minikube status
minikube
type: Control Plane
host: Running
kubelet: Running
apiserver: Running
kubeconfig: Configured
```

Рис. 1. Состояние Minikube

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
lventos/productcatalogapi	0.1	dc7f71b83be0	41 minutes ago	127MB
lventos/productcatalogapi	latest	dc7f71b83be0	41 minutes ago	127MB

Рис. 2. Вывод команды Docker-образов

В целях упрощения процессов конфигурации и управления жизненным циклом компонентов экосистемы используется Helm – инструмент, позволяющий создать унифицированные пакеты для развертывания приложений в Kubernetes. Применение Helm-чартов обеспечивает стандартизацию конфигурационных файлов, ускоряет обновление компонентов и позволяет вести учет версий разрабатываемых решений. В основу чартов входит файл Chart.yaml, содержащий ключевые метаданные, такие как название, версия и описание пакета (рисунки 3). Примеры конфигурационных файлов демонстрируют единообразный подход к управлению компонентами, что способствует снижению временных затрат на обновление и оптимизацию системы [5, 6].

```
apiVersion: v2
name: productcatlog
description: A Helm chart for Kubernetes
type: application
version: 0.1.0
appVersion: "1.16.0"
```

Рис. 3. Пример содержимого файла Chart.yaml

Эффективность и надежность разрабатываемой цифровой экосистемы во многом зависят от корректной конфигурации и согласованности всех ее компонентов. Для достижения этой цели применяется многоэтапное тестирование, включающее в себя статическую и динамическую валидацию. На начальной стадии особое внимание уделяется статическим проверкам, которые помогают выявить синтаксические ошибки, неточности в конфигурационных файлах и несоответствия внутренним стандартам еще до запуска системы в рабочем режиме. В ходе статического тестирования анализируются файлы Yet Another Markup

Language (YAML-файлы), определяющие структуру микросервисов, их зависимости, настройки сетевых взаимодействий и другие параметры, влияющие на общее функционирование экосистемы. Автоматизированные скрипты, входящие в состав инструментария тестирования, проверяют целостность схемы, правильность заполнения метаданных и соответствие объявленных ресурсов установленным спецификациям. Если на данном этапе обнаруживаются ошибки, система сигнализирует об этом, что дает возможность оперативно внести исправления, не дожидаясь более ресурсозатратной динамической проверки.

На рисунке 4 продемонстрирован результат выполнения одного из тестовых сценариев, направленного на оценку корректности ключевых параметров микросервисов и их конфигураций. Положительный исход проверки подтверждает, что все объекты успешно удовлетворяют требованиям по безопасности, структуре и производительности. Данные результаты позволяют судить о готовности экосистемы к последующим этапам тестирования, где моделируется реальная эксплуатационная нагрузка и анализируется способность системы адаптироваться к изменяющимся условиям. В совокупности подобные проверки формируют основу для дальнейшей оптимизации архитектуры, своевременного обнаружения потенциальных уязвимостей и обеспечения устойчивой работы цифровой платформы [7–10].

```
> component-ctk@0.0.1 L1-static E:\Dev\git-tmforum-oda\oda-component-ctk
> node L1-staticValidationRunner.js ".\v1alpha3.component.yaml"

*****
Open Digital Architecture - Component Test Kit CTK Level 1 Static Tests
*****

Step 0: Basic file tests for component .\v1alpha3.component.yaml
  ✓ File naming convention
  ✓ Valid YAML document(s)

Step 1: Check ODA-Component Metadata for component .\v1alpha3.component.yaml
  ✓ Contains document of kind: component
  ✓ Component apiVersion "oda.tmforum.org/v1alpha3" is within supported versions
  ✓ Component has metadata
  ✓ Metadata has name and labels
  ✓ Component has spec
  ✓ Spec has type, version, description, maintainers, owners
  ✓ Spec has coreFunction with exposedAPIs and dependentAPIs
  ✓ Swagger file of exposedAPIs and dependentAPIs is accessible (773ms)
  ✓ Spec has management
  ✓ Spec has security
  ✓ Security has partyrole
  ✓ Security has controllerRole

Step 2: Check any standard kubernetes resources are labelled for component .\v1alpha3.component.yaml
  ✓ Resource 0 is labelled
  ✓ Resource 1 is labelled
  ✓ Resource 2 is labelled
  ✓ Resource 3 is labelled
  ✓ Resource 4 is labelled
  ✓ Resource 5 is labelled
  ✓ Resource 6 is labelled
  ✓ Resource 7 is labelled
  ✓ Resource 8 is labelled
```

Рис. 4. Результат выполнения статического тестирования

Дополнительно на этапе статической валидации рассматривается правильность версий и зависимостей используемых библиотек, что способствует повышению безопасности и предотвращает конфликты в работе микросервисов. Подобный подход, основанный на поэтапном выявлении и устранении несоответствий, обеспечивает высокую надежность развернутой экосистемы и создает

благоприятные условия для быстрого внедрения новых сервисов, расширения функционала и последующего масштабирования в соответствии с потребностями пользователей.

### Результаты и обсуждение

В результате реализации цифровой экосистемы на базе ODA были получены показатели, свидетельствующие о высокой стабильности и масштабируемости разработанного решения. Проведенные тесты показали, что система способна обрабатывать высокие нагрузки при сохранении стабильного времени отклика, которое в среднем не превышало 150 мс для критически важных микросервисов. При моделировании пиковых нагрузок наблюдалось равномерное распределение трафика между контейнерами, что подтверждает корректную работу механизмов оркестрации в Kubernetes и демонстрирует способность системы к динамическому горизонтальному масштабированию. Дополнительно, автоматизация процессов развертывания посредством Helm позволила минимизировать количество ошибок при обновлении конфигураций, обеспечив непрерывность работы и высокую отказоустойчивость системы [7–10].

Практическая значимость разработанной модели заключается в ее потенциале для модернизации телекоммуникационных систем. Использование принципов открытой цифровой архитектуры позволяет операторам сетей интегрировать новые сервисы без существенных затрат времени и ресурсов, что особенно актуально в условиях стремительного роста объемов данных и числа подключаемых устройств. Применение унифицированных стандартов и API облегчает взаимодействие между компонентами, повышая совместимость и ускоряя процессы внедрения инновационных технологий. Кроме того, гибкость системы позволяет адаптировать ее конфигурацию в реальном времени, что существенно улучшает управляемость и оперативное реагирование на изменения рыночных условий. В свете полученных результатов рекомендуется дальнейшее развитие модели путем оптимизации алгоритмов оркестрации, расширения функциональности системы мониторинга и внедрения предиктивных аналитических инструментов для повышения устойчивости и эффективности работы цифровой экосистемы в долгосрочной перспективе [7–10].

### Выводы

В результате проведенного исследования подтверждено, что применение концепции ODA позволяет организовать интегрированную цифровую экосистему, характеризующуюся модульностью, гибкостью и высокой совместимостью компонентов. Представленный подход, включающий последовательные этапы контейнеризации, оркестрации и автоматизации развертывания, обеспечивает систематизацию процессов разработки и упрощает интеграцию микросервисов посредством стандартизированных API. Анализ проведенных тестирований показал, что разработанная архитектура соответствует поставленным целям, демонстрируя устойчивость и корректность функционирования компонентов системы при изменяющихся условиях эксплуатации.

Перспективы дальнейшего развития интегрированных цифровых экосистем, основанных на принципах ODA, связаны с оптимизацией алгоритмов оркестрации и совершенствованием механизмов автоматизации развертывания. В частности, целесообразно исследовать возможности внедрения расширенных систем мониторинга для более детального контроля за работой микросервисов, а также разработку предиктивных моделей, позволяющих своевременно обнаруживать и устранять потенциальные сбои. Дополнительное внимание может быть уделено интеграции современных инструментов обеспечения информационной безопасности [7–10], что позволит повысить надежность и устойчивость телекоммуникационных систем в условиях динамичного развития отрасли.

### Литература

1. Овениязов А., Богачева Д. Ю., Кочеткова И. А. Об открытой цифровой архитектуре и интерфейсе TM Forum Open APIs для автоматизации бизнеспроцессов в компаниях // Информационно-телекоммуникационные технологии и математическое моделирование высокотехнологичных систем: материалы Всероссийской конференции с международным участием (Москва, 18–22 апреля 2022 г.). М.: Российский университет дружбы народов, 2022. С. 51–54. EDN: ZTUENO
2. Гольдштейн А. Б., Кисляков С. В., Феноменов М. А. Открытая цифровая архитектура. Движение к 5G // Вестник связи. 2023. № 7. С. 12–15. EDN: VZKZFU
3. Гольдштейн А. Б., Кисляков С. В. Концепция открытой цифровой архитектуры: эволюция или революция? // Вестник связи. 2022. № 6. С. 21–25. EDN: GVAFPF
4. Гольдштейн А. Б., Кисляков С. В., Феноменов М. А. Модель WFM на основе Open Digital Architecture // Вестник связи. 2024. № 6. С. 1–8. EDN: PYBZCQ
5. Головашов С., Агатий И. Обеспечение информационной безопасности и контроля в Kubernetes // Системный администратор. 2023. № 10(251). С. 30–33. EDN: JXCSGF
6. Фунг В. К., Богатырев В. А. Экспериментальное исследование производительности кластера с контейнерной виртуализацией // Известия высших учебных заведений. Приборостроение. 2024. Т. 67. № 8. С. 647–656. DOI: 10.17586/0021-3454-2024-67-8-647-656. EDN: NAQTBV
7. Ефаров И. Б., Макаров Д. А. Инфраструктурный подход к реализации отказоустойчивости в микросервисной архитектуре // Безопасность-2023: материалы Всероссийской научно-практической конференции (Чита, 03–05 мая 2023 г.). Чита: Забайкальский государственный университет, 2023. С. 115–121. EDN: SAEERT
8. Овакянян Л. К. Архитектура микросервисов как современный инструмент коммуникаций в экосистеме // Экономика, управление и политика: проблемы теории практики и взаимодействия: материалы ежегодной Всероссийской студенческой научно-практической конференции (Тверь, 22–26 апреля 2024 г.). Тверь: Тверской государственный университет, 2024. С. 241–246. EDN: RBQMYV

9. Чэнь Ю. Современные инструменты и платформы для разработки микросервисной архитектуры // Актуальные вопросы общества, науки и образования: сборник статей XVII Международной научно-практической конференции (Пенза, 20 ноября 2024 г.). Пенза: Наука и Просвещение (ИП Гуляев Г.Ю.), 2024. С. 70–72. EDN: RKRIJG

10. Резенов Г. В. Архитектура микросервисов и ее реализация с помощью технологии контейнеризации // Вестник Российского нового университета. Серия: Сложные системы: модели, анализ и управление. 2020. № 3. С. 131–138. DOI: 10.25586/RNU.V9187.20.03.P.131. EDN: NRGMXS

Статья поступила 22 ноября 2024 г.  
Одобрена после рецензирования 10 декабря 2024 г.  
Принята к публикации 25 декабря 2024 г.

### Информация об авторах

*Белозеров Клим Владимирович* – студент группы ИКТС-33м Санкт-Петербургского государственного университета телекоммуникаций им. проф. М. А. Бонч-Бруевича. E-mail: belozerov@sut.ru

*Ганюшин Даниил Сергеевич* – студент группы ИКТС-33м Санкт-Петербургского государственного университета телекоммуникаций им. проф. М. А. Бонч-Бруевича. E-mail: ganyushin@sut.ru

*Зайдуллин Рустем Рамилевич* – студент группы ИКТС-33м Санкт-Петербургского государственного университета телекоммуникаций им. проф. М. А. Бонч-Бруевича. E-mail: zaidullin@sut.ru

*Кисляков Сергей Викторович* – кандидат технических наук, доцент кафедры Инфокоммуникационных систем Санкт-Петербургского государственного университета телекоммуникаций им. проф. М. А. Бонч-Бруевича. E-mail: kislyakov@sut.ru

<https://doi.org/10.31854/2307-1303-2024-12-4-13-21>  
EDN: TYVQYB

## Application of Open Digital Architecture for Creating Integrated Digital Ecosystems

K. Belozеров, D. Ganyushin, R. Zaidullin, S. Kislyakov✉

The Bonch-Bruевич Saint Petersburg State University of Telecommunications,  
St. Petersburg, 193232, Russian Federation

*The aim of this work is to develop a methodology for modeling and deploying an integrated digital ecosystem based on the concept of Open Digital Architecture (ODA) to modernize telecommunication systems. The research is based on the use of modern containerization and orchestration technologies, including Docker, Kubernetes, and Helm, as well as on the application of standardized APIs to ensure seamless integration of microservices. The methodology involves the step-by-step creation and packaging of services into containers, their automated deployment in a Kubernetes cluster, and configuration management using Helm charts. The conducted testing confirmed the correctness of component interactions and the system's resilience under varying operating conditions. The novelty of the research lies in the comprehensive approach that combines the principles of ODA with automation and orchestration technologies, thereby reducing the time required for the deployment of new services and lowering operational costs. The practical significance of this work is determined by the potential application of the developed model for optimizing telecommunication network management, enhancing their flexibility, and adapting to dynamic load changes. The research conclusions indicate the effectiveness of the proposed methodology for creating adaptive digital ecosystems, which opens up prospects for further development of monitoring tools and predictive analytics in the context of modern telecommunication infrastructure.*

**Key words:** Open Digital Architecture (ODA), digital ecosystem, containerization, orchestration, Kubernetes, Helm, microservices

### Information about Authors

*Klim Belozеров* – 2nd year master program student (The Bonch-Bruевич Saint Petersburg State University of Telecommunications). E-mail: [belozеров@sut.ru](mailto:belozеров@sut.ru)

*Daniil Ganyushin* – 2nd year master program student (The Bonch-Bruевич Saint Petersburg State University of Telecommunications). E-mail: [ganyushin@sut.ru](mailto:ganyushin@sut.ru)

*Rustem Zaydullin* – 2nd year master program student (The Bonch-Bruевич Saint Petersburg State University of Telecommunications). E-mail: [zaidullin@sut.ru](mailto:zaidullin@sut.ru)

*Sergey Kislyakov* – Ph. D. of Engineering Sciences, Associate Professor at the Department of Infocommunication Systems (The Bonch-Bruевич Saint Petersburg State University of Telecommunications). E-mail: [kislyakov@sut.ru](mailto:kislyakov@sut.ru)