

Параллельные вычислительные технологии (ПаВТ'2016)
Parallel Computational Technologies (РСТ'2016)



Разработка облачного сервиса для тестирования микросервисных приложений

Южно-Уральский государственный университет
Челябинск

Глеб Игоревич Радченко (gleb.radchenko@susu.ru)

Дмитрий Игоревич Савченко

Никита Андреевич Ашихмин

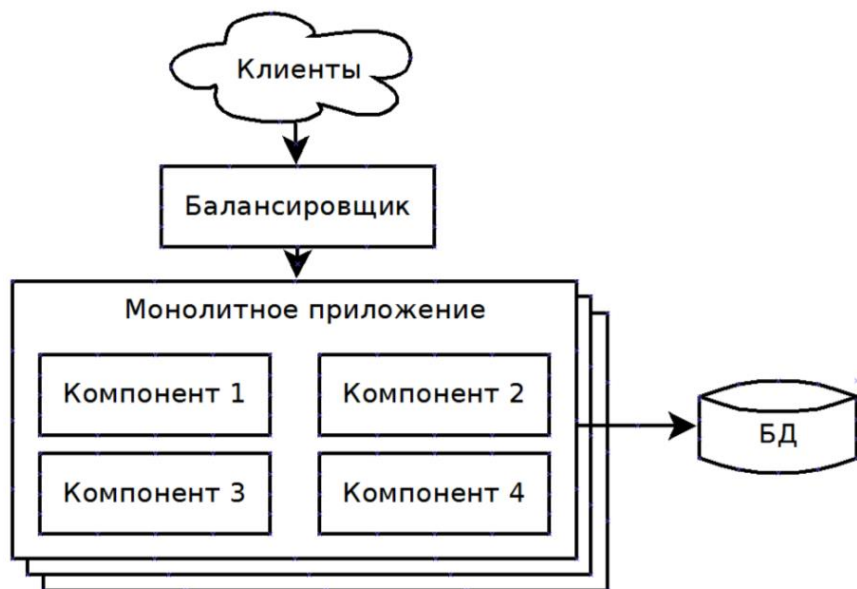


Микросервисы и все остальные

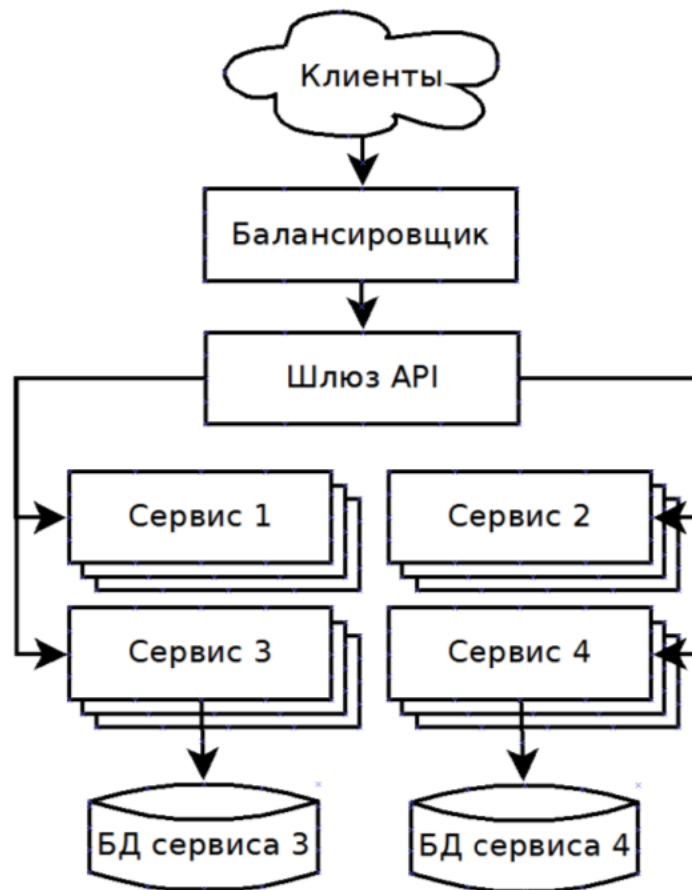
- **Микросервисная архитектура** облачных систем – это концептуальное развитие компонентно-ориентированного подхода.
- Микросервис – это независимый программный компонент, который:
 - разрабатывается, разворачивается и исполняется независимо от других компонентов системы;
 - обеспечивает прозрачный веб-доступ к своим ресурсам посредством стандартизованного интерфейса (например, REST);
 - отвечает за реализацию отдельной бизнес-задачи.
- Применяется ведущими провайдерами облачных ресурсов: Amazon, eBay, Netflix



Монолитная архитектура



Микросервисная архитектура



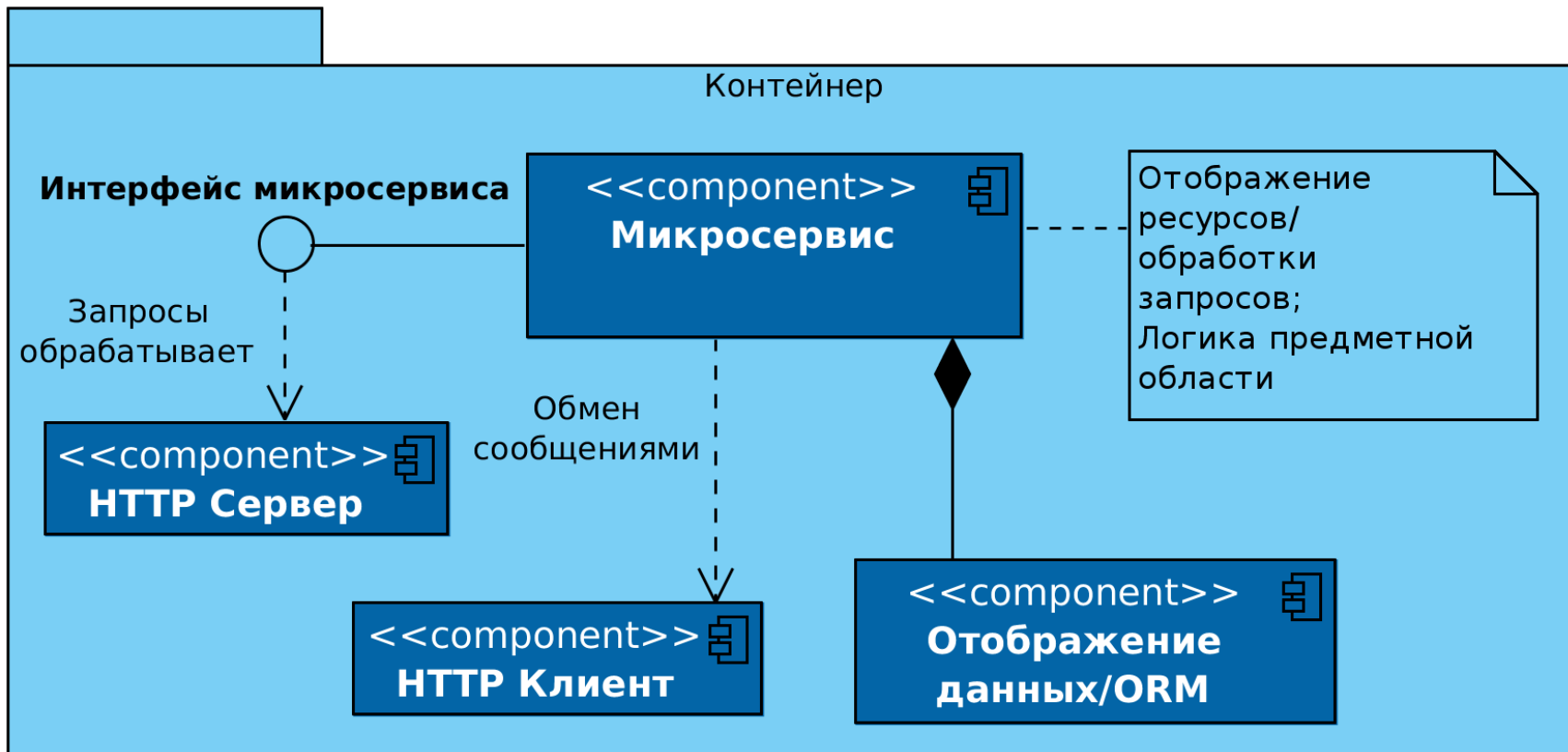


Достоинства микросервисного подхода

- *Четкая граница ответственности:* каждый компонент отвечает за реализацию своей части бизнес-логики (включая хранение и обработку данных);
- *Автономность* – микросервисы могут быть разработаны, развернуты, запущены, остановлены и др. независимо друг от друга;
- *Открытый стандартизованный интерфейс* обеспечивает прозрачное использование сторонних микросервисов (как в виде исходного кода, отдельных контейнеров, так и в виде независимых сервисов);



Структура микросервиса

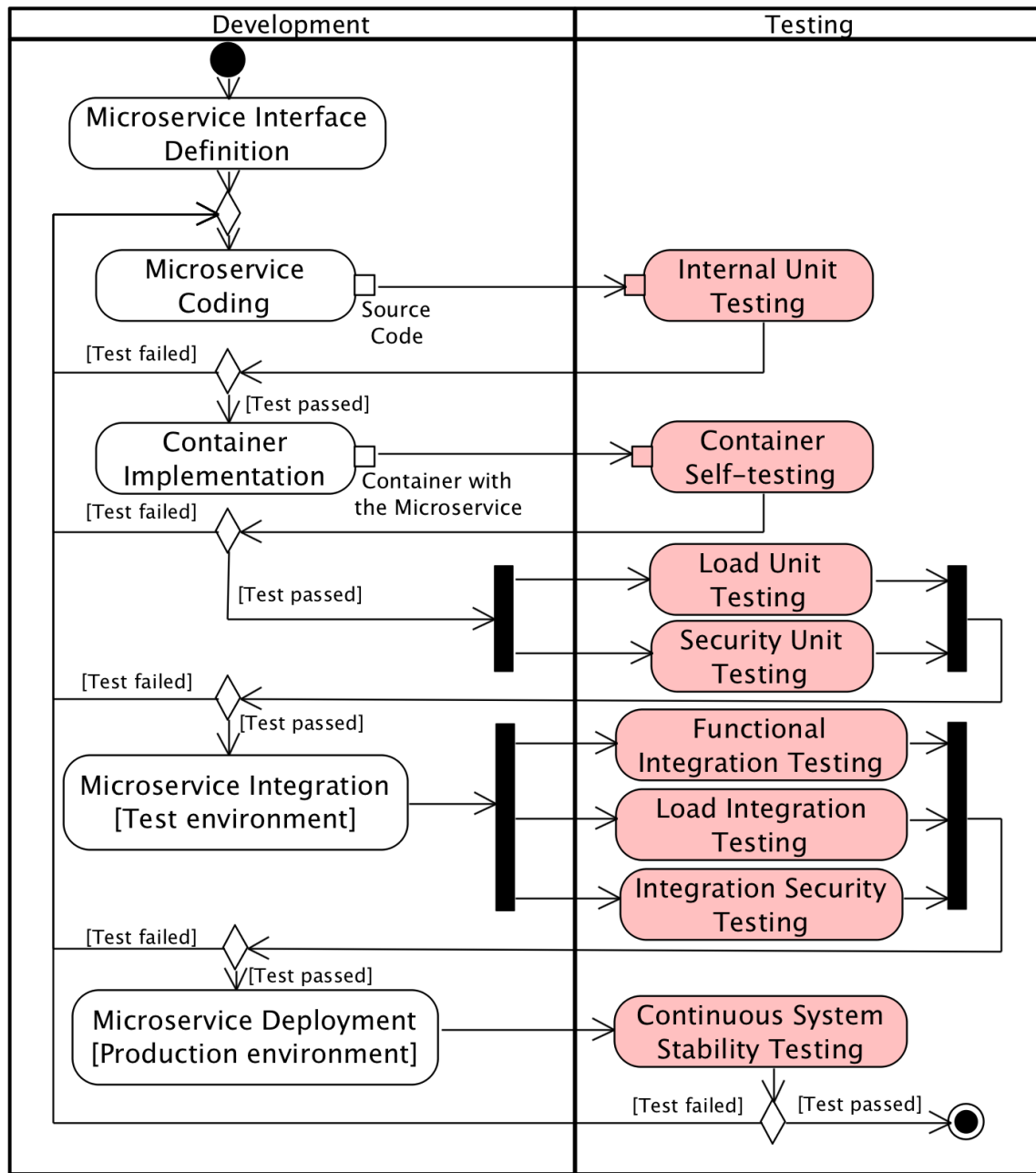


Особенности и ограничения микросервисов



- Для имплементации микросервисного подхода необходима вычислительная среда, обеспечивающая автоматическое **развертывание, масштабирование и взаимодействие** микросервисов.
- Разработка микросервисных систем невозможен без **непрерывной интеграции** разработанных и/или модифицированных микросервисов в существующую инфраструктуру.
- Что влечет необходимость в постоянном **всестороннем тестировании** как отдельных микросервисов, так и совокупности системы в целом.





Методы тестирования микросервисных приложений



- компонентное тестирование классов, входящих в микросервис
- самотестирование микросервисов
- нагрузочное компонентное тестирование
- компонентное тестирование безопасности
- функциональное интеграционное тестирование
- нагрузочное интеграционное тестирование
- интеграционное тестирование безопасности



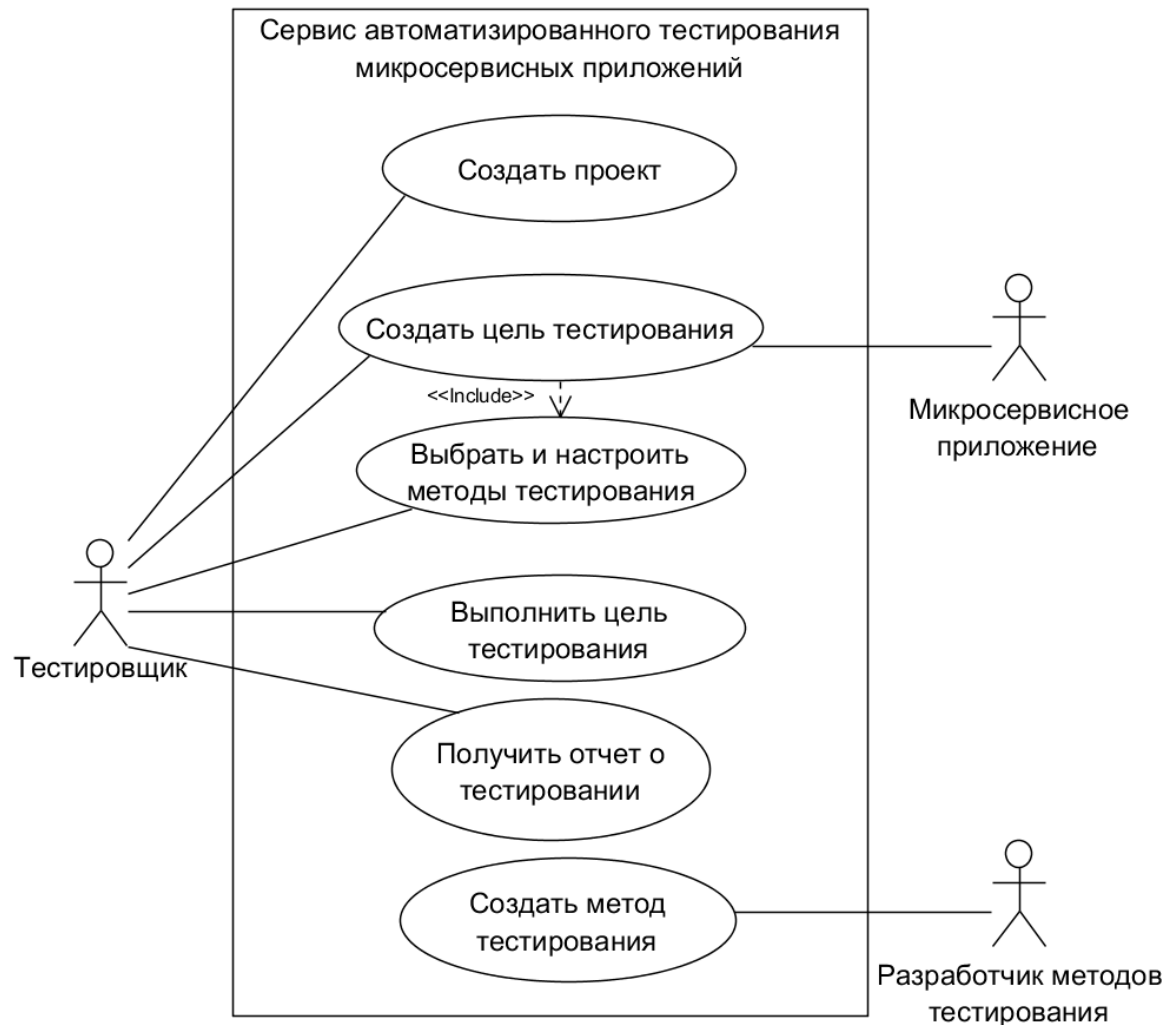
Облачный сервис для тестирования микросервисных приложений



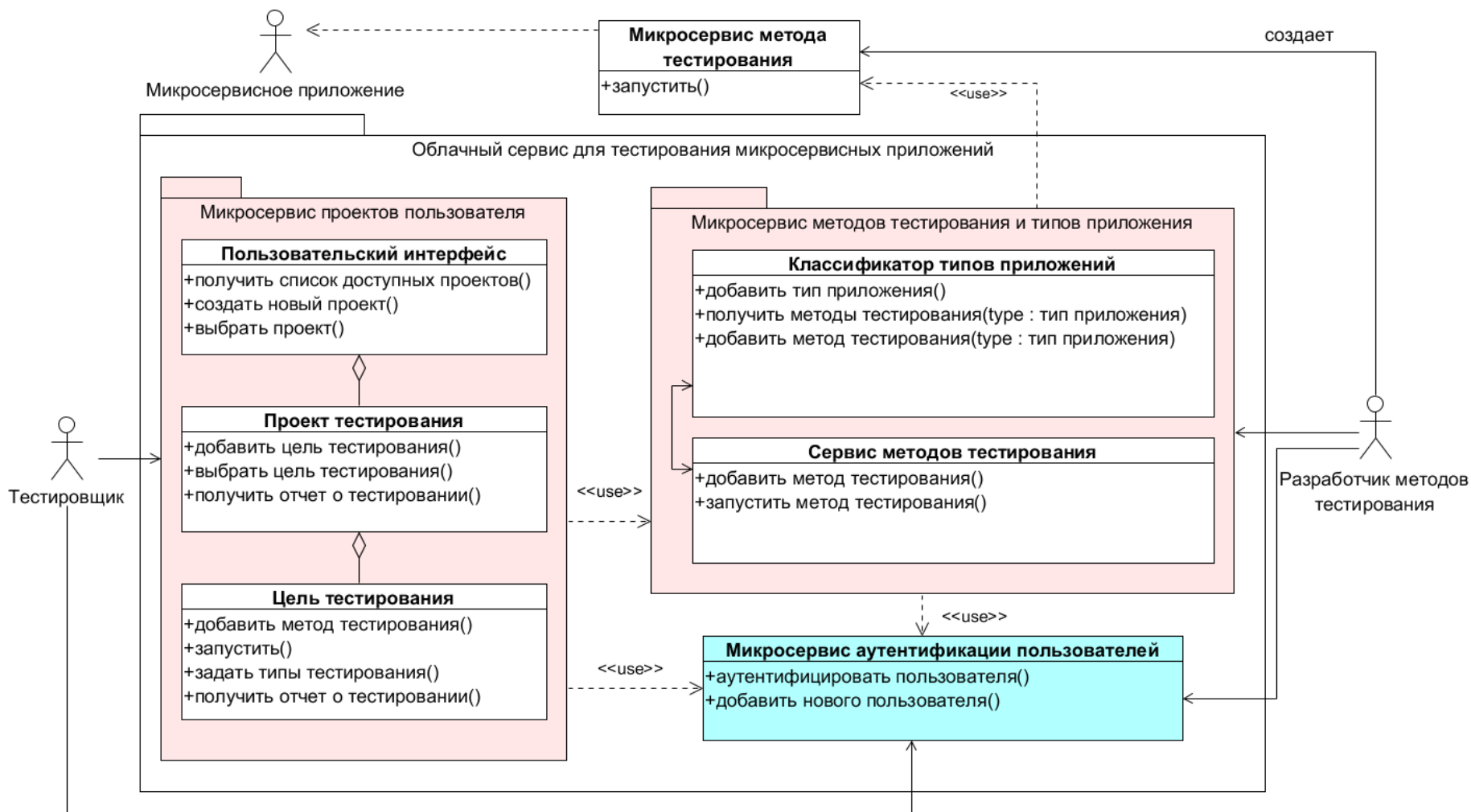
- В настоящий момент, активно разрабатывается целый ряд независимых методов, для тестирования различных аспектов облачных микросервисных приложений.
- Нами предлагается решение, обеспечивающее возможность организации автоматизированного всестороннего тестирования микросервисных облачных приложений
- Пользователь системы сможет применить интересующие его методы тестирования для собственного облачного приложения, указав методы доступа к тем или иным ресурсам микросервисного приложения (репозитории исходного кода, репозитории виртуальных машин, адреса облачных серверов и др.).



Варианты использования сервиса тестирования

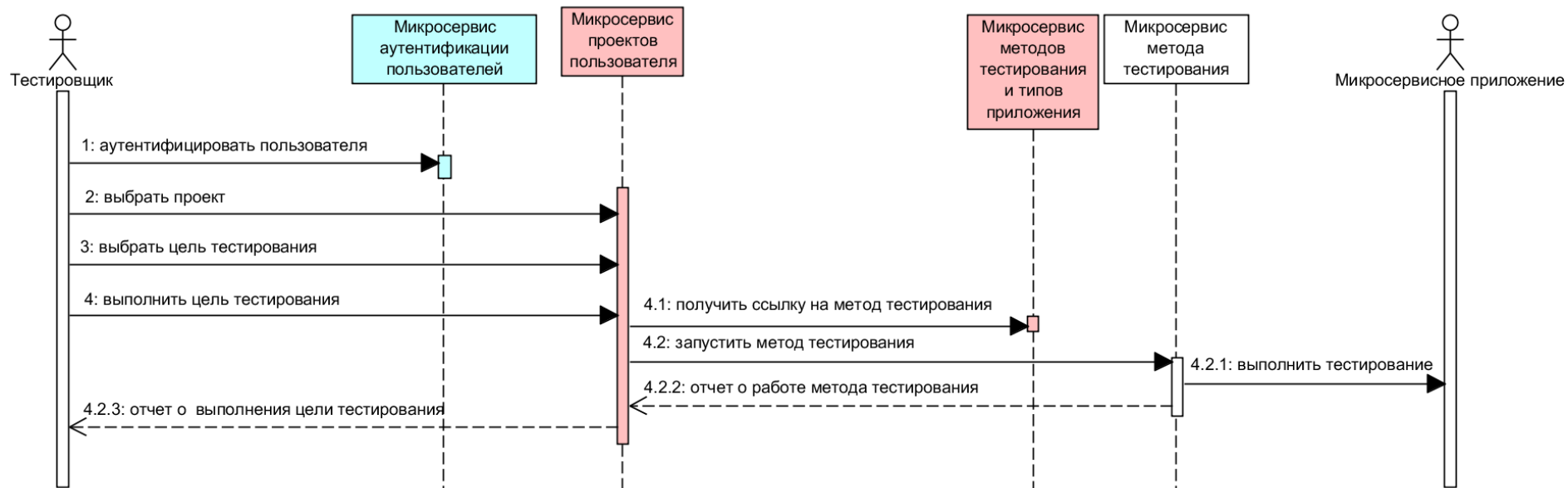


Архитектура сервиса тестирования микросервисного приложения





Процесс тестирования приложения

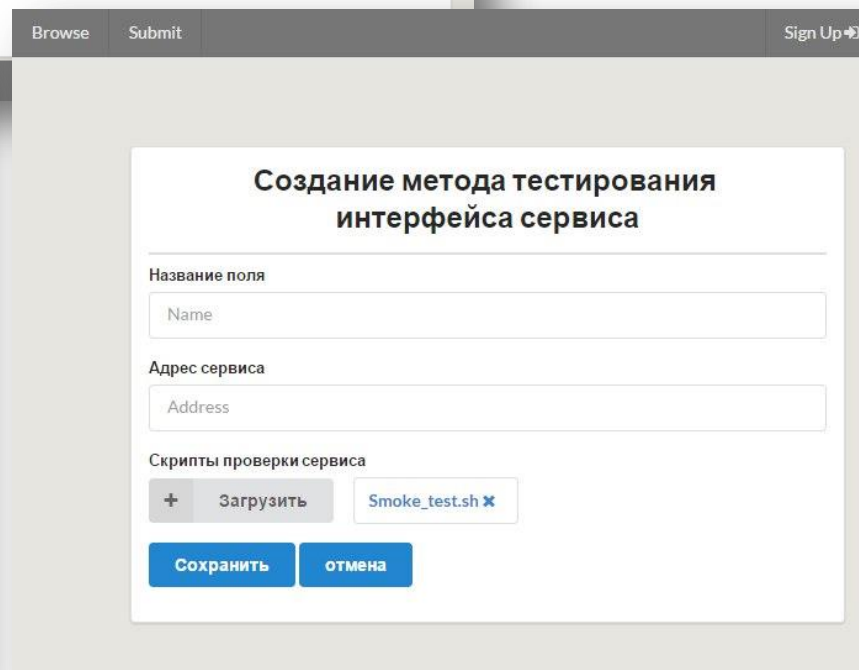
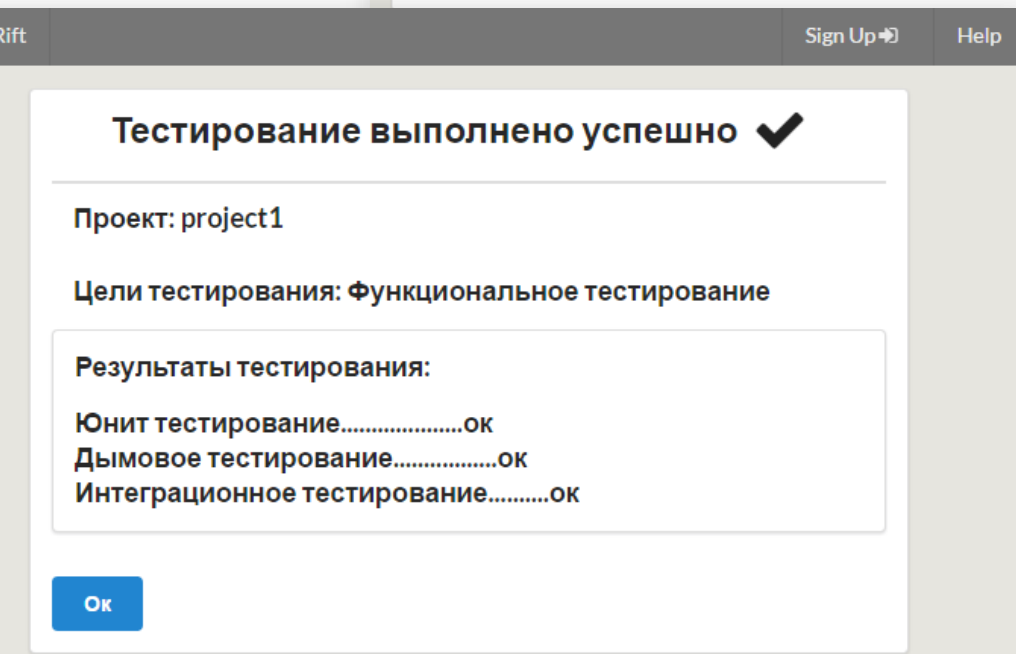
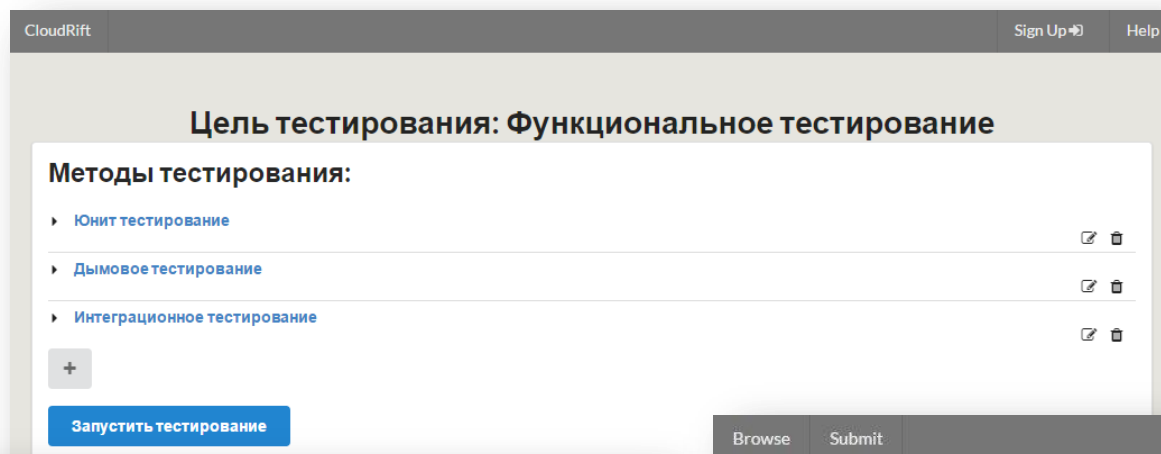




- На основе предложенной архитектуры реализуется сервис CloudRift:
 - Платформа: **Ruby-On-Rails**
 - Контейнеризация: **Docker**
 - Авторизация: **Oauth2**
 - Программный интерфейс: **JSON REST API**
 - UI: **Semantic Ui**
 - Облачная система: **Heroku**



CloudRift: Реализация сервиса тестирования





Результаты

- Определены требования и разработана архитектура сервиса для тестирования микросервисных приложений.
- Начата реализация сервиса тестирования, а также отдельных микросервисов для компонентного тестирования и тестирования безопасности.
- Развитие сервиса тестирования:
 - внедрение методов нагрузочного тестирования, тестирования стабильности (ChaosMonkey)
 - открытие API и поддержка для создания собственных методов тестирования приложений

