



DataArt



Ещё одна история миграции приложения
в облако

Монолит, бинарники и два дымящихся разработчика



Исходная позиция



Компания:

- не имела собственного IT штата
- разработка велась силами third-party вендора

Компания планировала:

- значительный рост в течение следующих нескольких лет
 - глобальное расширение
 - запуск новых продуктов
 - повышение операционной эффективности за счет инвестиций в ИТ
- улучшение User Experience

Исходная позиция

Компания:

- прекратила сотрудничество с вендором
- осталась у разбитого корыта с монолитным приложением
 - устаревшие технологии
 - ненадёжное

Вендор:

- не оставил документацию/артефакты
- не оставил исходных кодов от части приложения

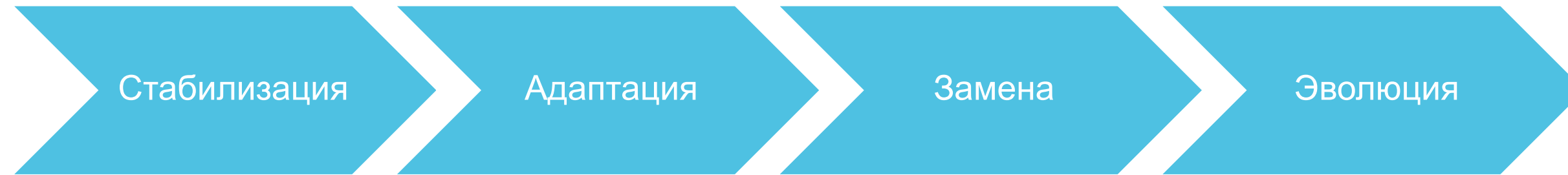


Исходная позиция с точки зрения новой команды разработки

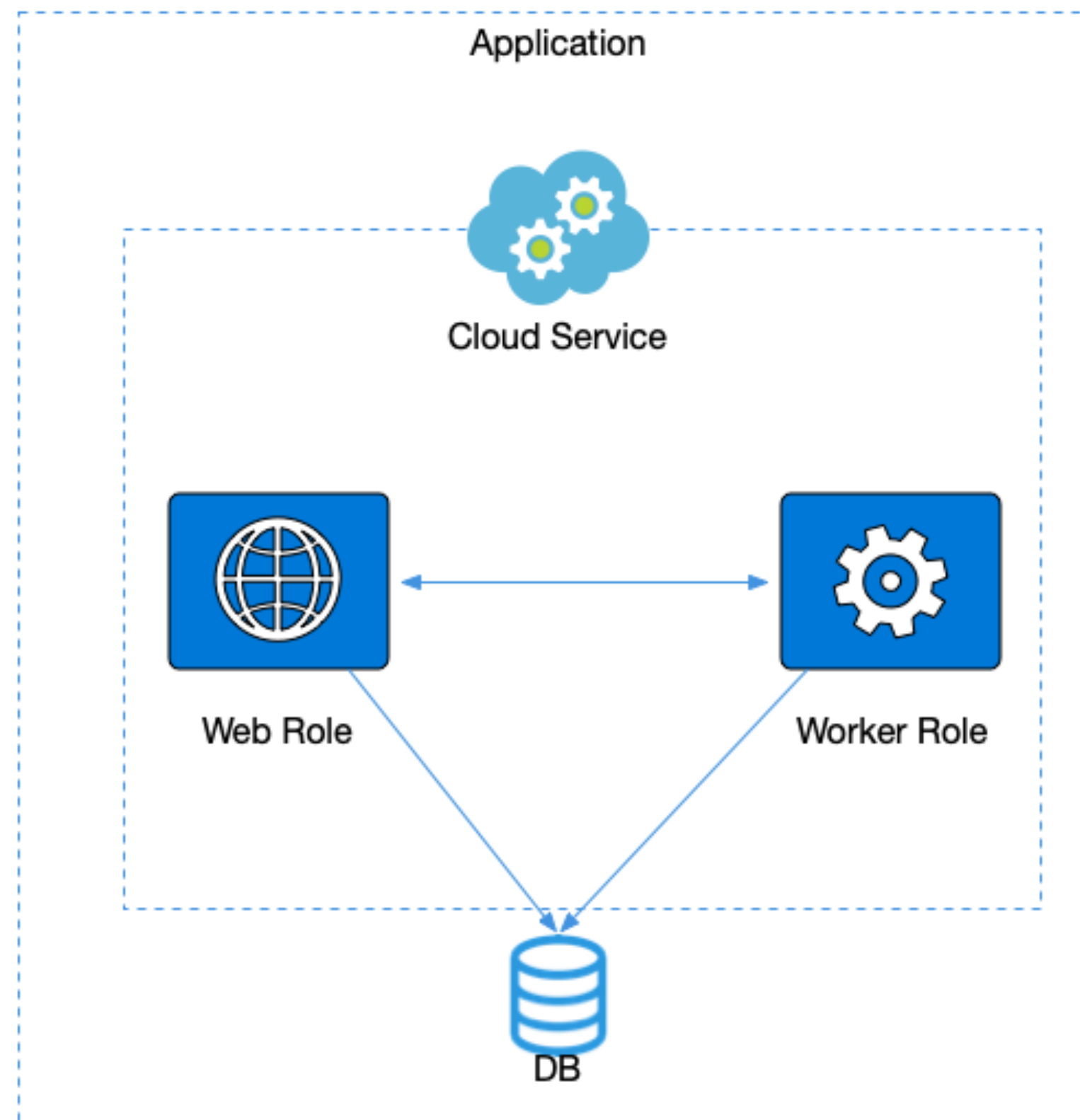
Нет смысла развивать продукт без значительного re-design:

- *отсутствие гибкости* – трудно изменить программное обеспечение в соответствии с потребностями бизнеса
- *недостаток информации* – никто на самом деле не знает, как это работает, нет документации, руководств, нет доступа к людям, которые его построили
- *высокая стоимость обслуживания* – трудно устранять неисправности, трудно поддерживать надежность
- *устаревшая технология* – высокая стоимость инфраструктуры, сложность при интеграции с другими системами; требует от дорогих людей знания устаревших технологий, вызывает проблемы безопасности, производительности и взаимодействия с пользователем

Подход



Стабилизация



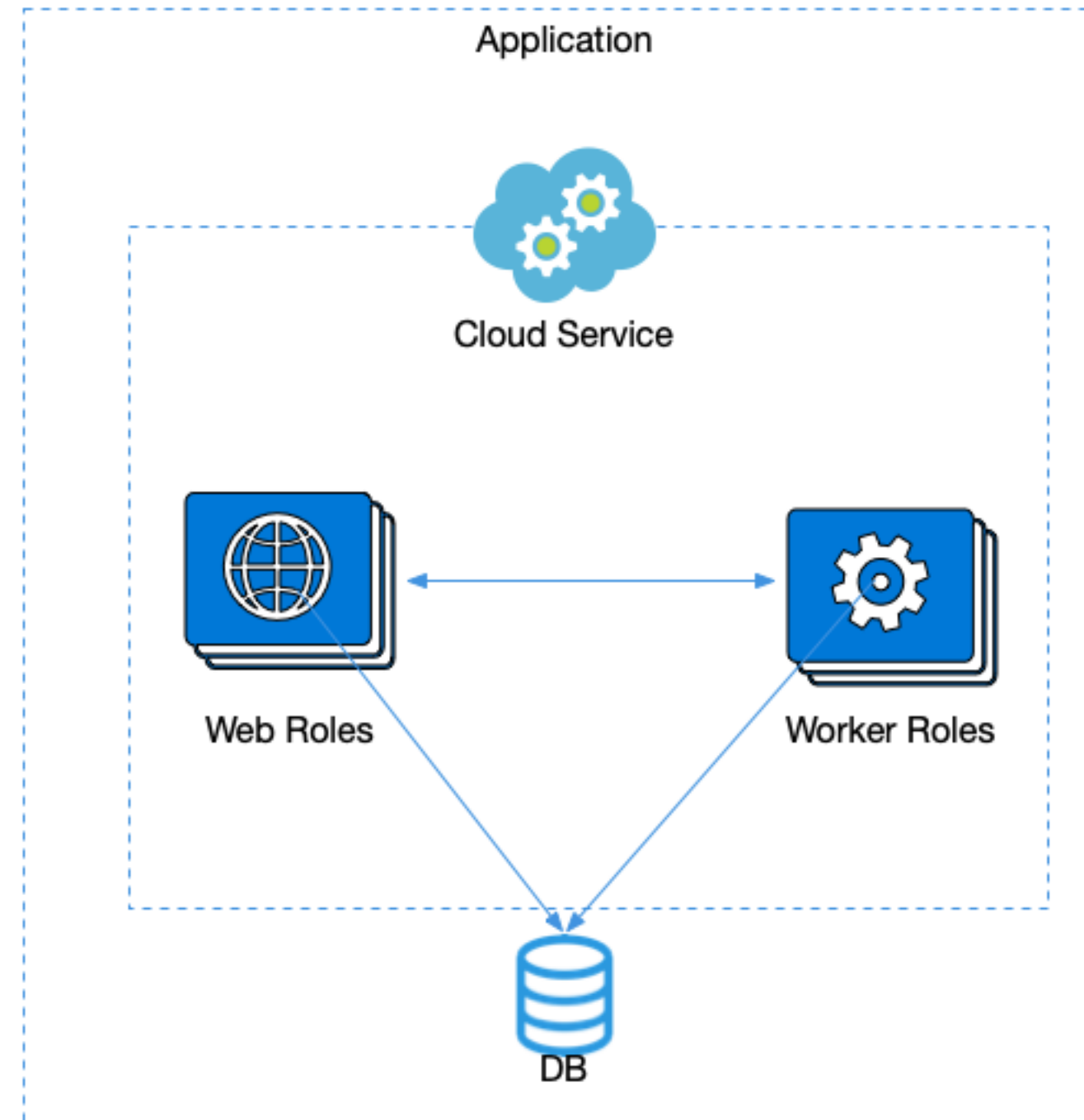
Демо

Стабилизация

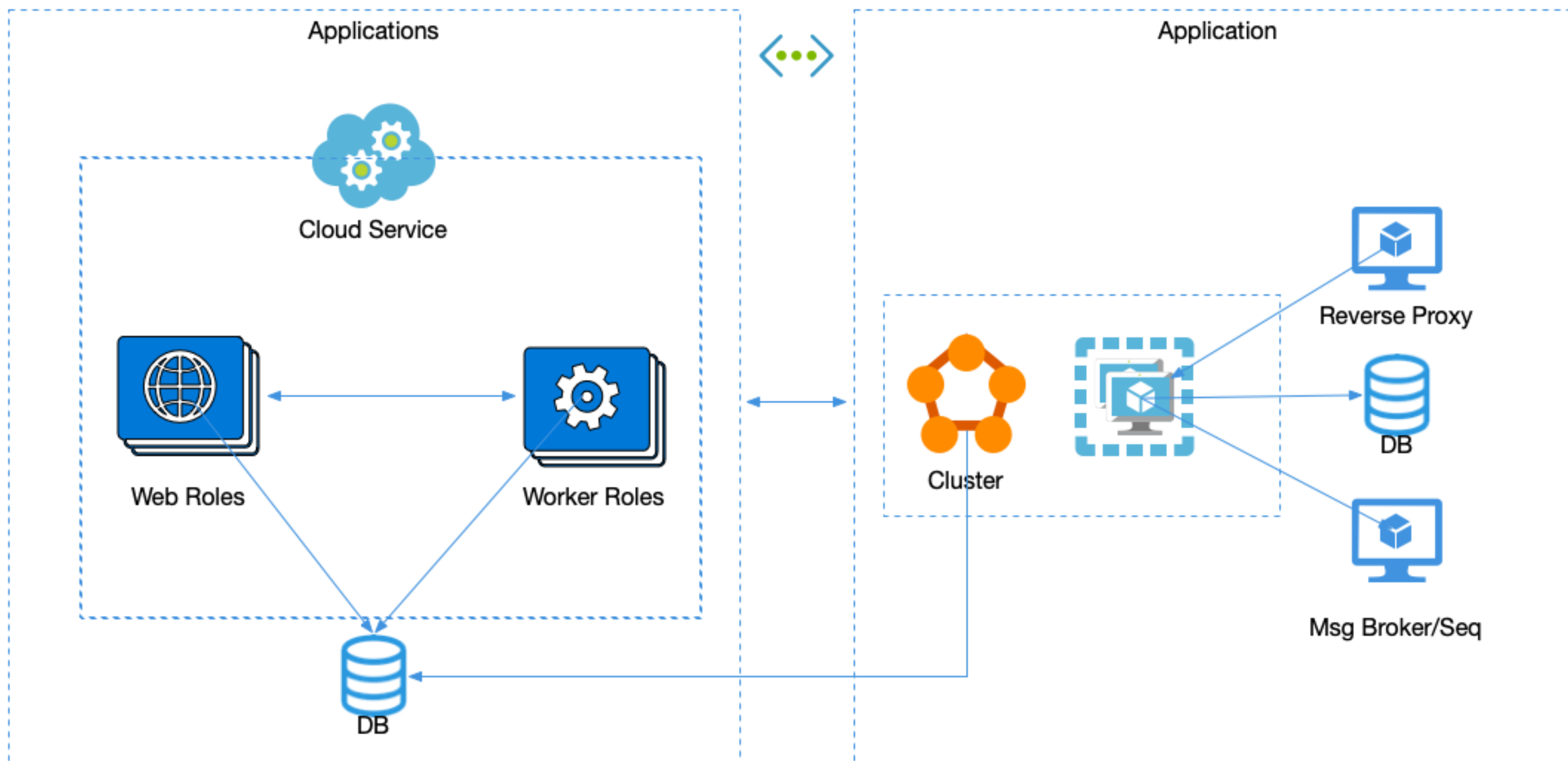
- Построение облачной инфраструктуры
 - Cloud Services
 - Virtual Machines (MS SQL, ...)

- Процесс разработки
 - CI/CD (TeamCity, Octopus)
 - Тестовые окружения
 - Git flows
 - Jira, Confluence

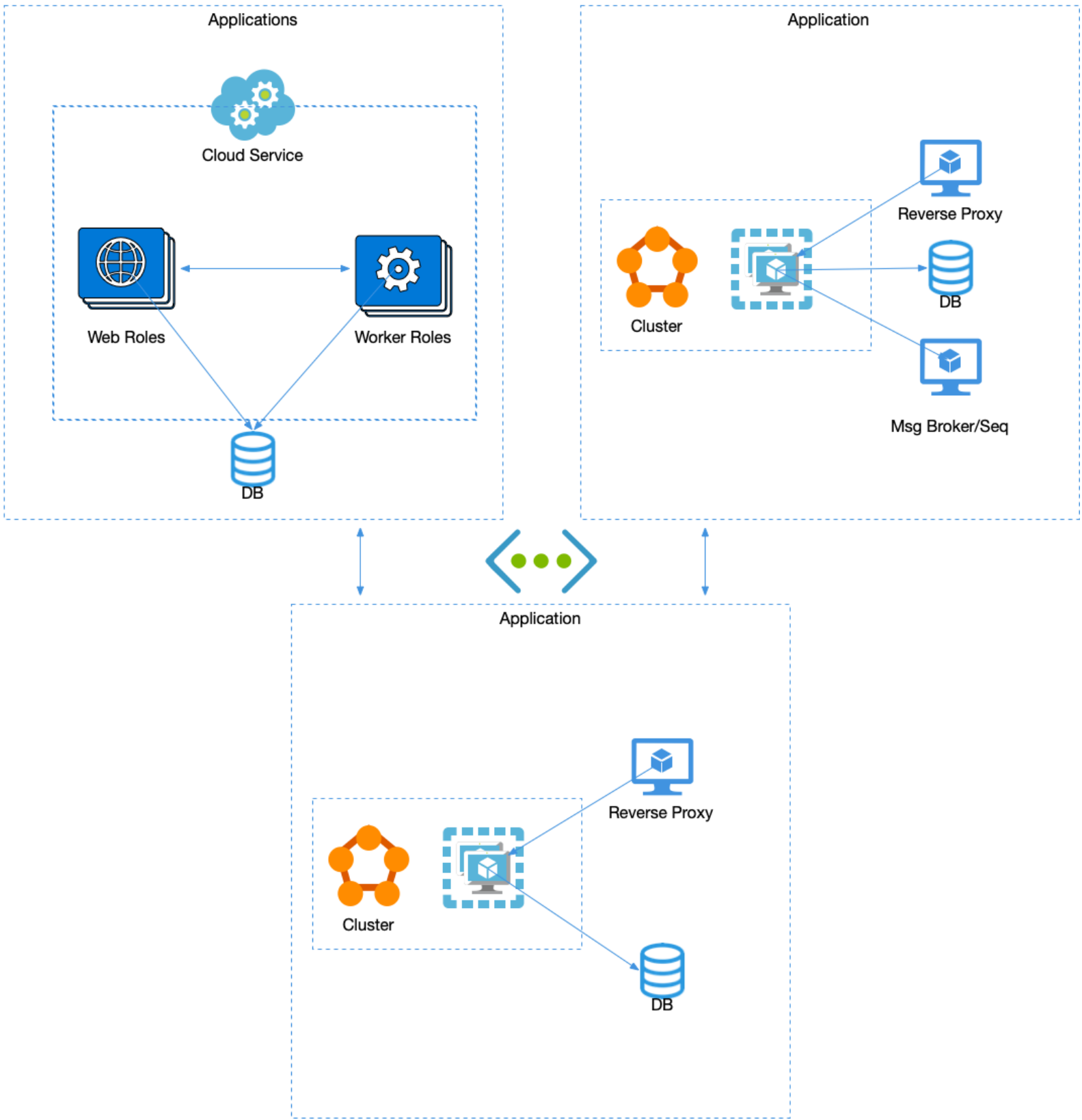
Адаптация



Замена



Эволюция



Disaster Recovery



Промежуточные итоги



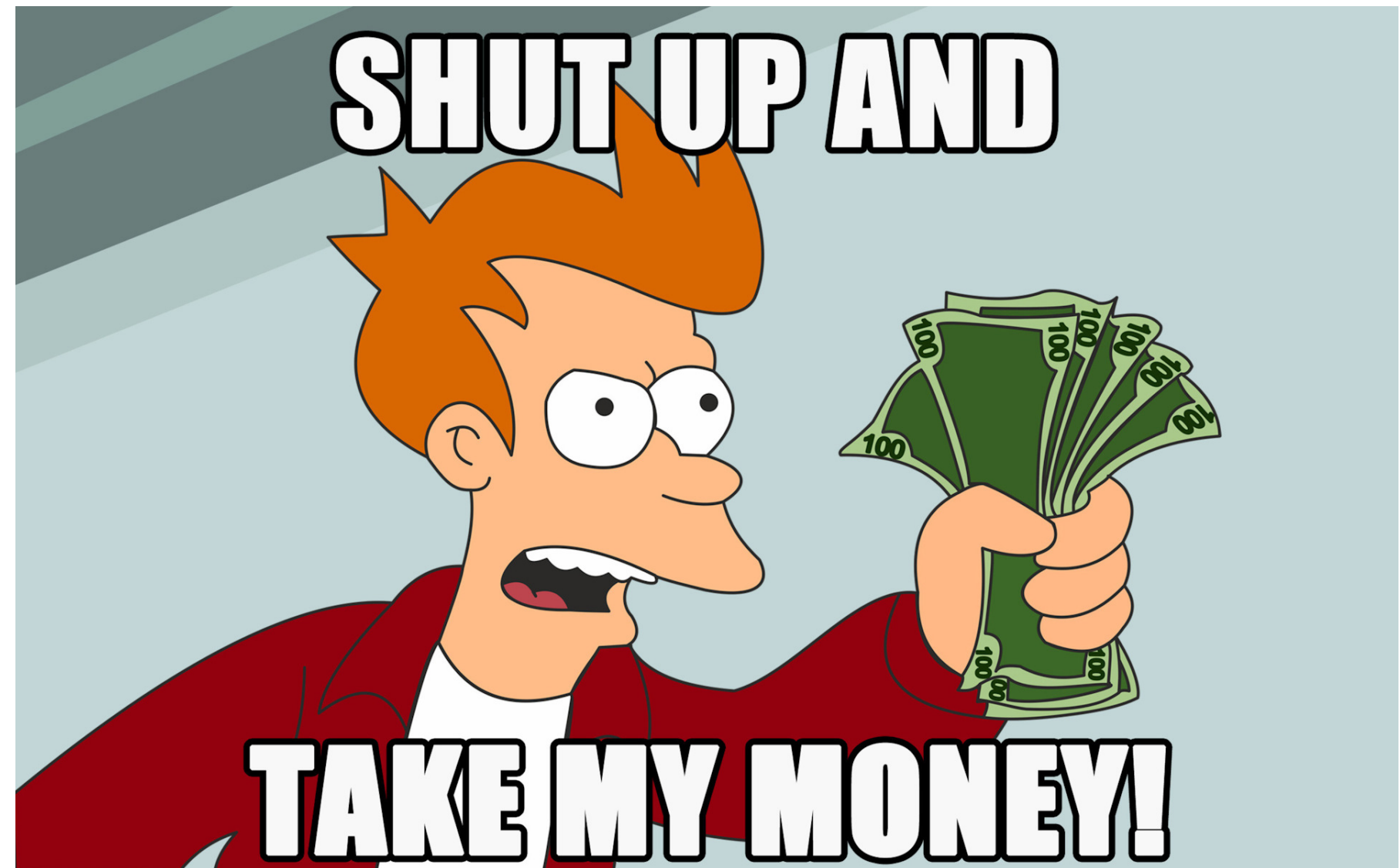
-
- CI/CD
 - Octopus Deploy
 - MSSQL Servers
 - 78 cloud services
 - 13 Service Fabric Clusters
 - 25 Windows VMs
 - 7 Linux VMs

Одно НО

Стоимость окружений:

Dev окружения ~ 14к\$

Prod окружение ~ 13к\$



Оптимизация стоимости



Работы с инфраструктурой:

- Infrastructure as a Code (templates, arm, terraform)
- Start/Stop service instances

Изменение типов сервисов:

- Docker
- Azure Container Instances
- Web App (App Services)

Демо

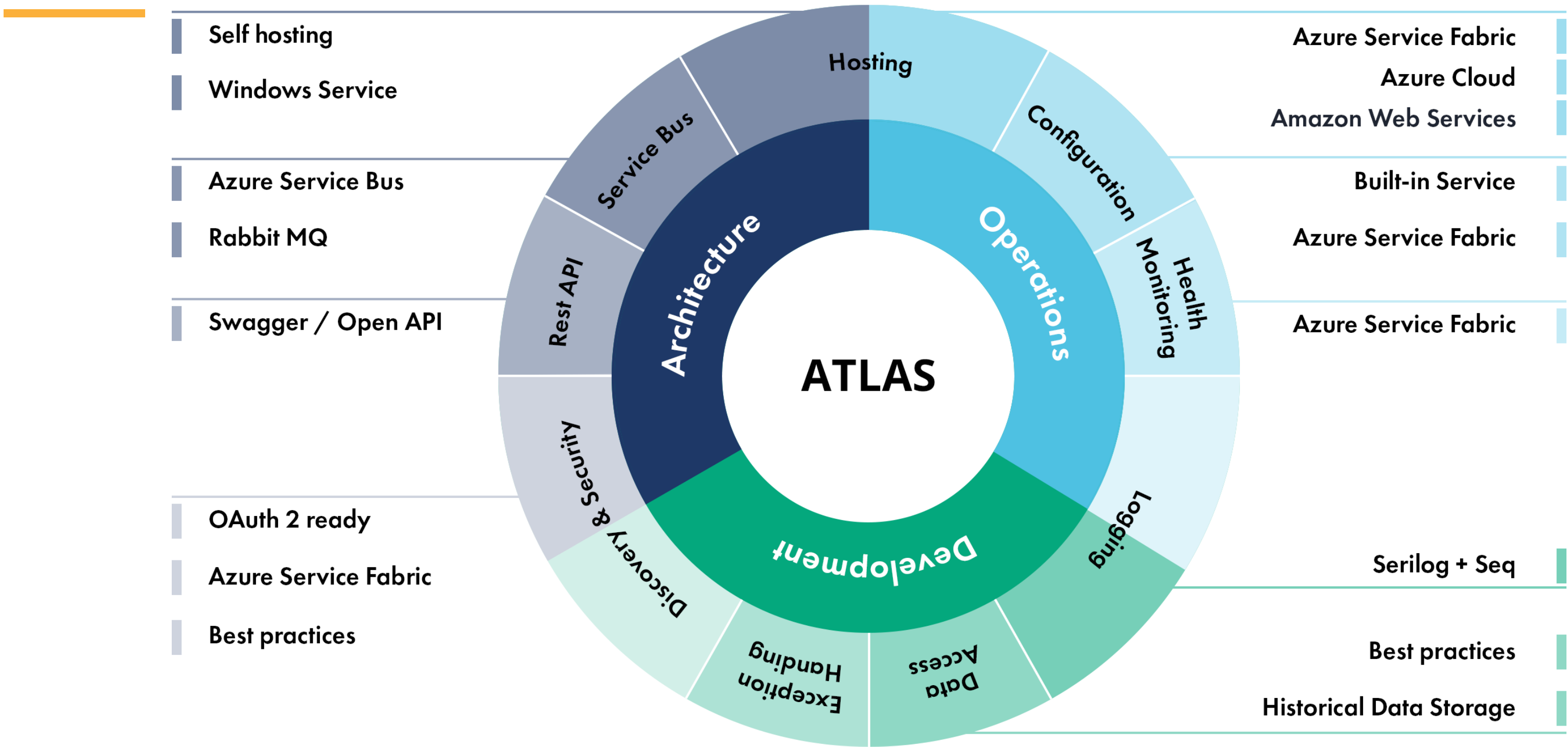
Что-то общее



Унификация:

- логирование
- работа с ошибками
- корреляция вызовов
- локальная разработка vs облако
- конфигурации
- ...

The Atlas



- Определить цели и задачи бизнеса перед миграцией
- Разбить процесс миграции на этапы
- Процесс миграции должен обеспечивать потребности бизнеса на каждом своём этапе
- Оптимизация инфраструктуры – такой же естественный процесс, как и рефакторинг кода
- Infrastructure as Code (templates/terraform) – must have!
- Если делать по уму, то есть простор для создания собственных фреймворков :)

Где найти



<https://github.com/DataArt/meetup-azure-monolith-migration>

<https://github.com/DataArt/Atlas>

Спасибо