Microsoft®
tech·ed
Russia | 2011

9–10 НОЯБРЯ 2011
МОСКВА

Microsoft tech·ed Russia 2011

9–10 НОЯБРЯ 2011 · МОСКВА

ARC208

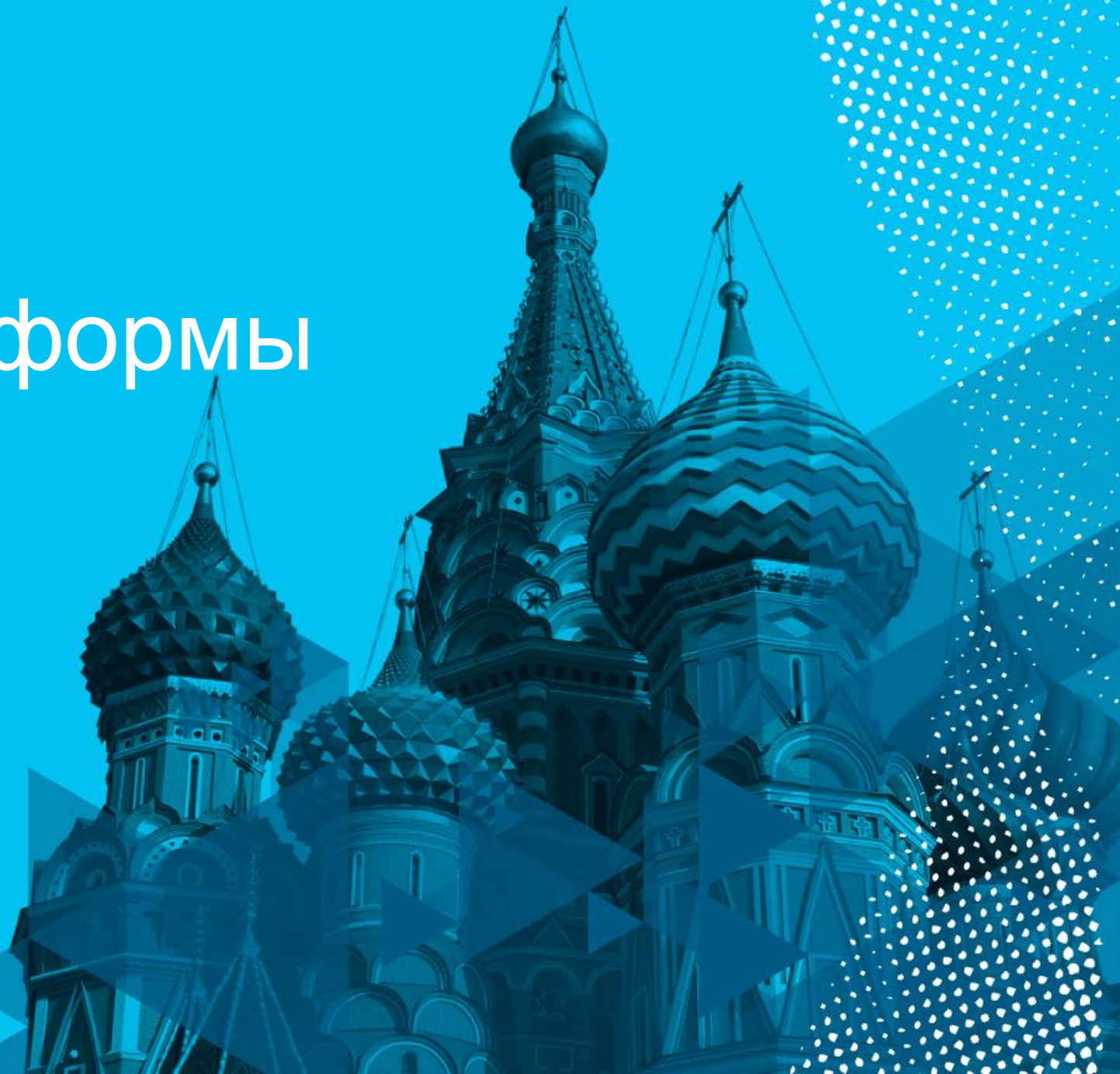
Подходы облачного проектирования в Windows Azure

Гайдар Магдануров
Руководитель направления веб-технологий
Microsoft

Содержание

- Облачные платформы
  - Предпосылки появления и возможности
- Windows Azure
  - Краткий обзор основных компонентов
- Типовая архитектура
  - … приложений в облаке
- Важные моменты
  - … при проектировании облачных приложений

Облачные платформы



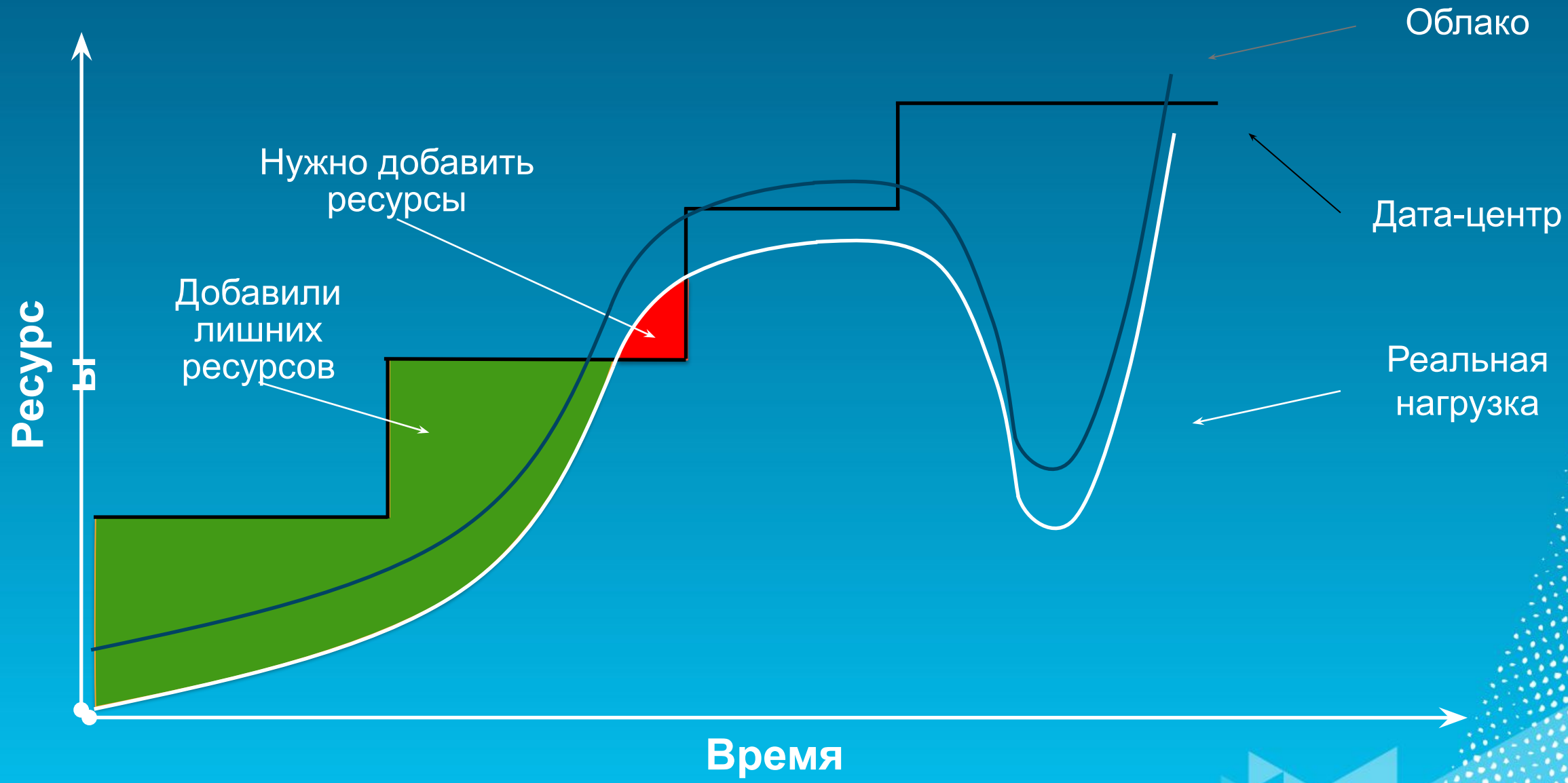
Предпосылки появления облачных платформ

- Рост нагрузки на частные дата-центры
  - Увеличение количества пользователей
  - Распространение мобильных решений
  - Одновременная совместная работа
  - Повышение ожиданий от приложений
  - Непредсказуемые пики нагрузки
- Высокие расходы на дата-центы
  - Ресурсы: электричество, охлаждение, сети, персонал
  - Проблемы масштабирования: большие первоначальные инвестиции, необходимость платить «деньги вперед» за непредсказуемую нагрузку

Возможности облачных платформ

- Масштабирование системы по необходимости
- Высокая доступность и отказоустойчивость
- Эффективное управление расходами
- Возможность фокусироваться на создании и обслуживании продукта, а не на инфраструктуры
- Быстрая публикация решений любого масштаба

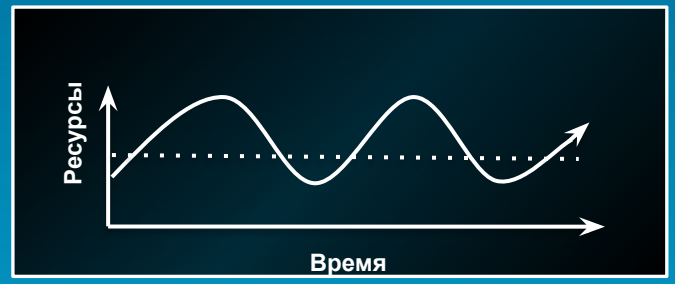
Классический дата-центр и облако



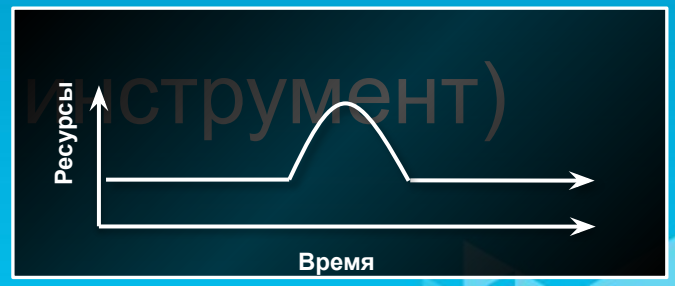
Эффективные облачные сценарии нагрузки

1. Периодическое включение (выборы)

2. Рост нагрузки (социальная сеть)



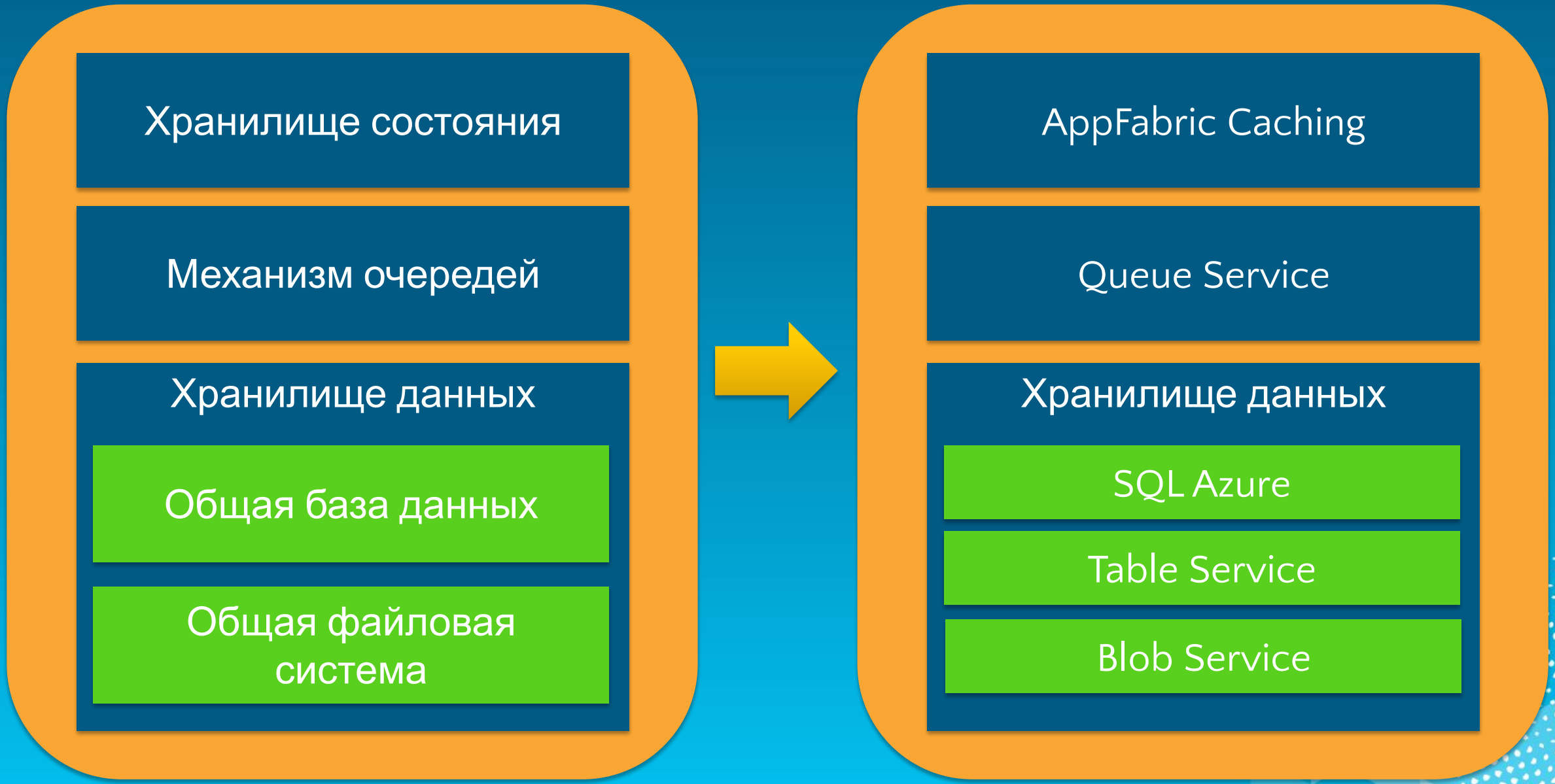
3. Периодическая нагрузка (рабочий инструмент)



Технологическая реализация облака

Windows Azure

Windows Azure



Windows Azure

- Windows Azure Platform – окружение, управляющее облаком и набор сервисов (.NET, identity, storage). Набор виртуальных машин (web role, worker role)
- SQL Azure – распределенная реляционная база данных
- Table Service – не-реляционное хранилище сущностей (1 Мб, (255 – 3) свойства у каждой сущности)
- Blob Service – хранилище двоичных данных, может быть подключено как общий сетевой диск (1 Тб в page blob, 200 Гб в block blob)
- Queues – квази-транзакционная очередь (8Кб сообщение)

Windows Azure AppFabric

- Service Bus - связь между распределенными приложениями на основе сообщений
- Access Control – управление доступом
- Distributed Caching – распределенный кэш в памяти
  - Хранение состояния
  - Кеширование данных
  - Одинаковая модель программирования для приложений, размещаемых в облаке и частном дата-центре

Требования к облачной архитектуре

Требования к архитектуре в облаке

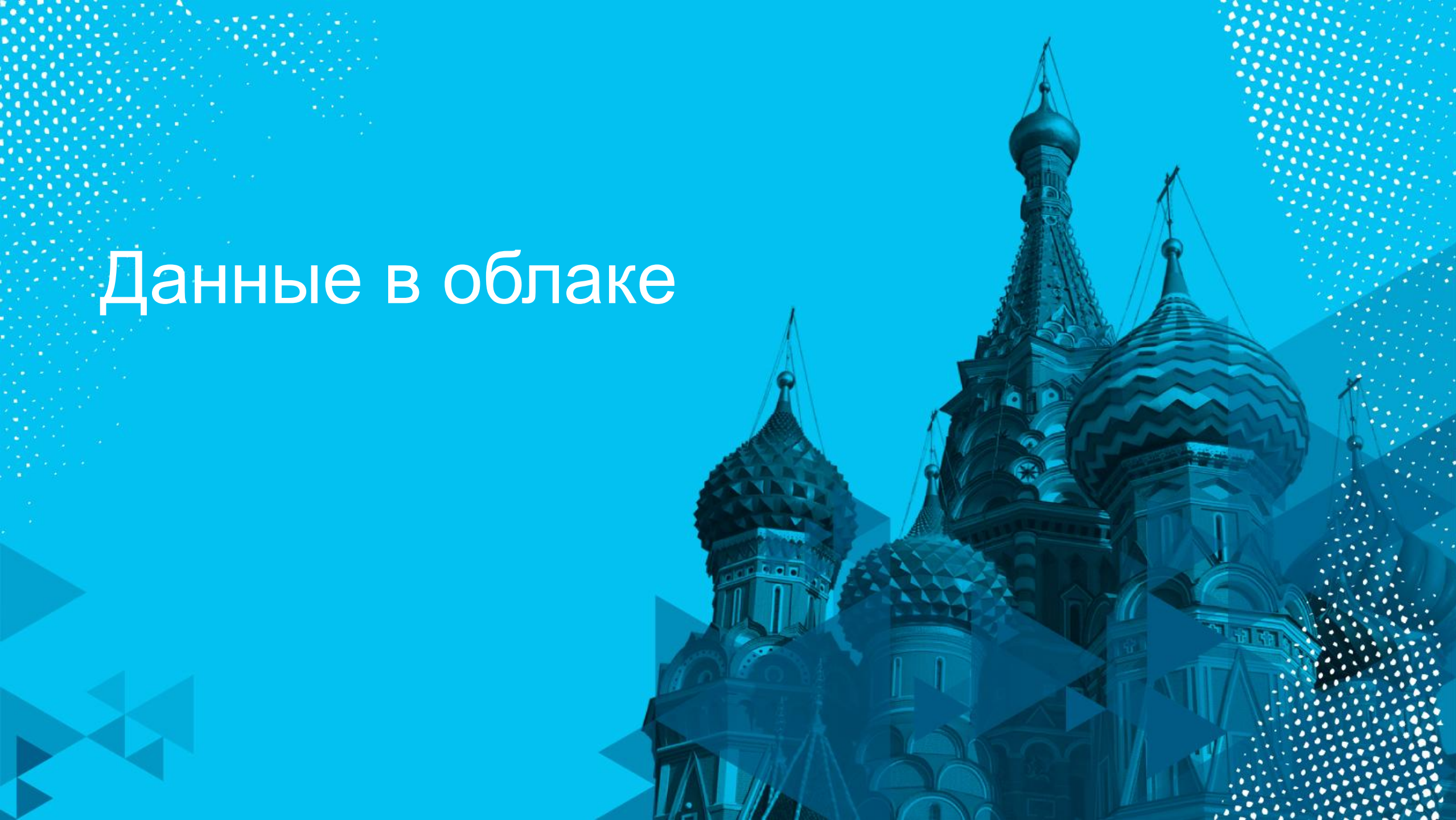
- Слабая связанность
  - Автономные компоненты, общающиеся сообщениями
- Масштабируемость
  - Независимое дублирование компонентов
- Отказоустойчивость
  - Независимая работа компонентов
- Параллелизм
  - Асинхронная обработка задач
- Сохранение целостности данных
  - Валидация данных и сообщений

Типовая сценарии использования облака

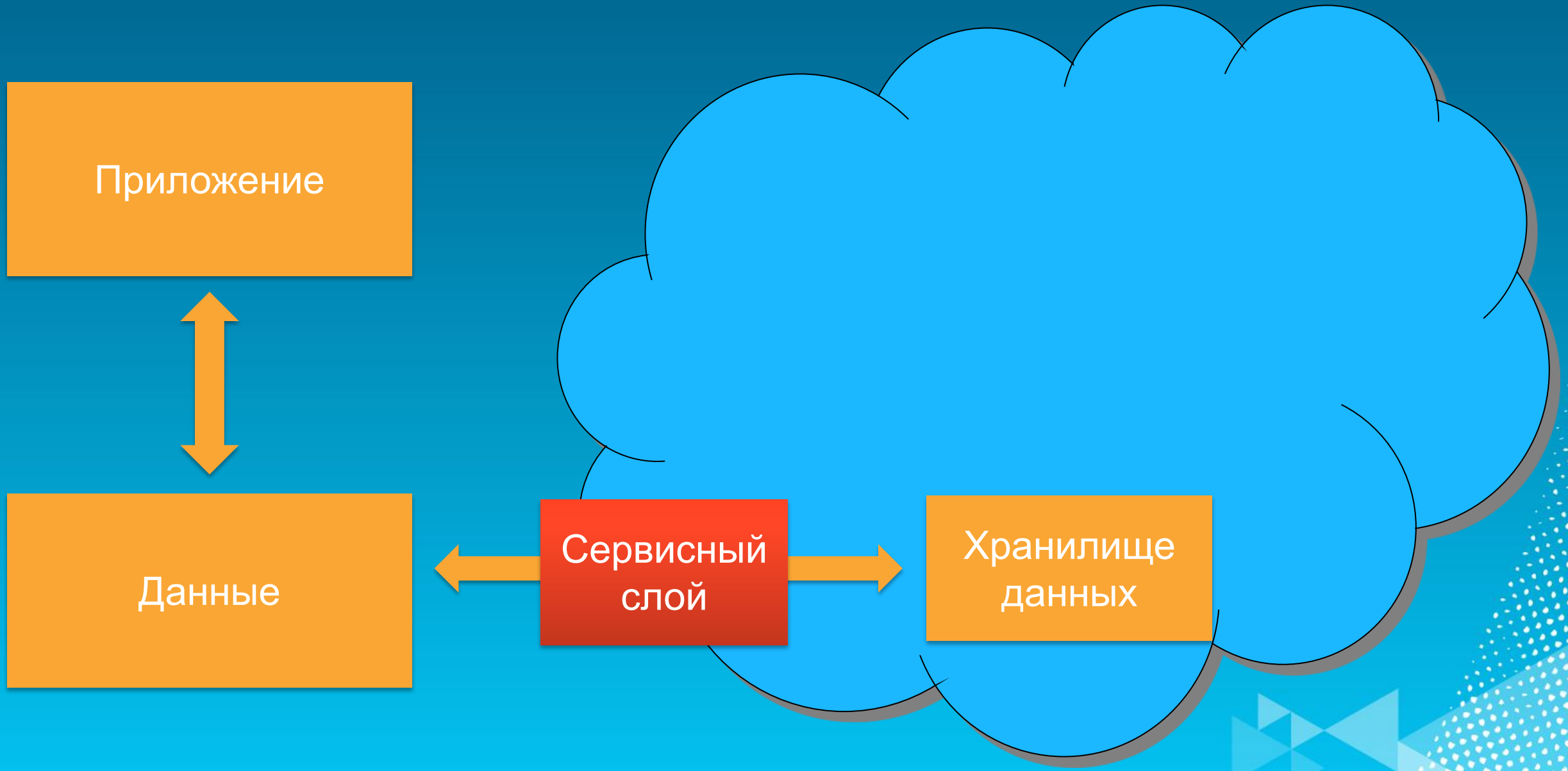
Они же – возможный путь миграции существующего приложения в облако.

- Размещение данных в облаке
- Размещение фоновой обработки в облаке
- Размещение приложения в облаке

Данные в облаке



Данные в облаке

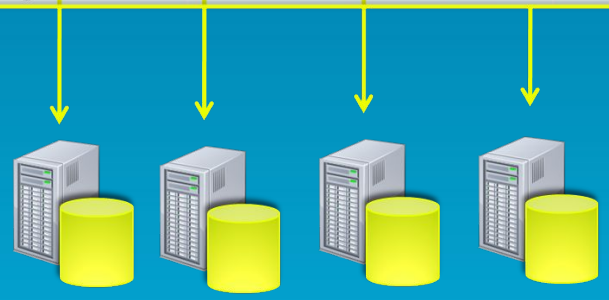


Проектирование: данные в облаке

- Разбиение данных
  - Горизонтальное
  - Вертикальное
- Требуемый эффект
  - Уменьшение объемов данных
  - Уменьшение количества транзакций
  - Уменьшение стоимости эксплуатации хранилища
  - Повышение эластичности в период пиковых нагрузок

Горизонтальное разбиение

FirstName	LastName	Email	Thumbnail	Photo
David	Alexander	davida@contoso.com	3kb	3MB
Jarred	Carlson	jaredc@contoso.com	3kb	3MB
Sue	Charles	suec@contoso.com	3kb	3MB
Simon	Mitchel	simonm@contoso.com	3kb	3MB
Richard	Zeng	richardz@contoso.com	3kb	3MB



- Данные «размазаны» между несколькими нодами
- Возможно масштабирование
- Дешевые запросы внутри одной партиции
- Дорогие запросы между разными партициями

Горизонтальное разбиение - Table Storage

- Партиции автоматически балансируются
  - Нет необходимости разбивать на равномерные части
- «Горячие» активные партиции могут быть масштабируемы
  - Windows Azure может выделить больше ресурсов более загруженным партициям
- Partition Key и Row Key = уникальный ID
  - PartitionKey должен быть указан для Create, Update, Delete
  - Выборки между партициями выполняются последовательно
- Данные могут быть возвращены несколькими страницами (continuation tokens)

Горизонтальное разбиение – SQL Azure

- Партиции – разные базы данных в SQL Azure
  - Необходимо для объемов данных более > 50GB
  - Большая транзакционная нагрузка (возможны сбои)
- Логика разбиения полностью на разработчике
- Нет автоматической балансировки партиций
  - Необходимо равномерно распрелелять нагрузку
  - Размер партиции не играет роли, важна нагрузка
- Партиции стоят денег
  - Оптимизация расходов за счет создания дополнительных партиций под высокую нагрузку и удаление после заверешия высокой нагрузки

Вертикальное разбиение

FirstName	LastName	Email	Thumbnail	Photo
David	Alexander	davida@contoso.com	3kb	3MB
Jarred	Carlson	jaredc@contoso.com	3kb	3MB
Sue	Charles	suec@contoso.com	3kb	3MB
Simon	Mitchel	simonm@contoso.com	3kb	3MB
Richard	Zeng	richardz@contoso.com	3kb	3MB

SQL Azure Tables BLOBs

- Распределение данных между хранилищами
- Часто используемые данные хранятся в «дорогом» индексированном хранилище
- Большие объемы данных в «дешевом» бинарном хранилище
- Получение всей строки требует более одного запроса

Цели вертикального разбиения

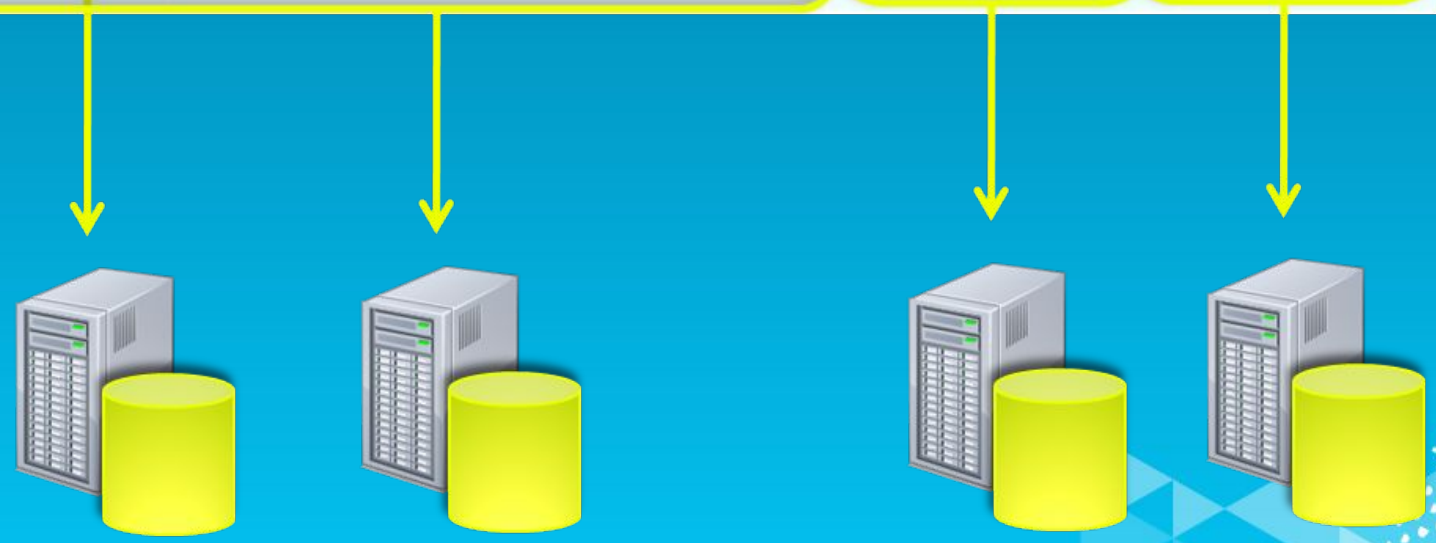
- Баланс производительности и стоимости
- SQL Azure
  - Индексируемое
  - Нет платы за транзакцию
  - Фиксированная плата за объем хранилища
- Windows Azure Storage
  - Ограниченные возможности индексирования
  - Оплата за запрос
  - Плата зависит от объемов передаваемых данных

Пример вертикального разбиения

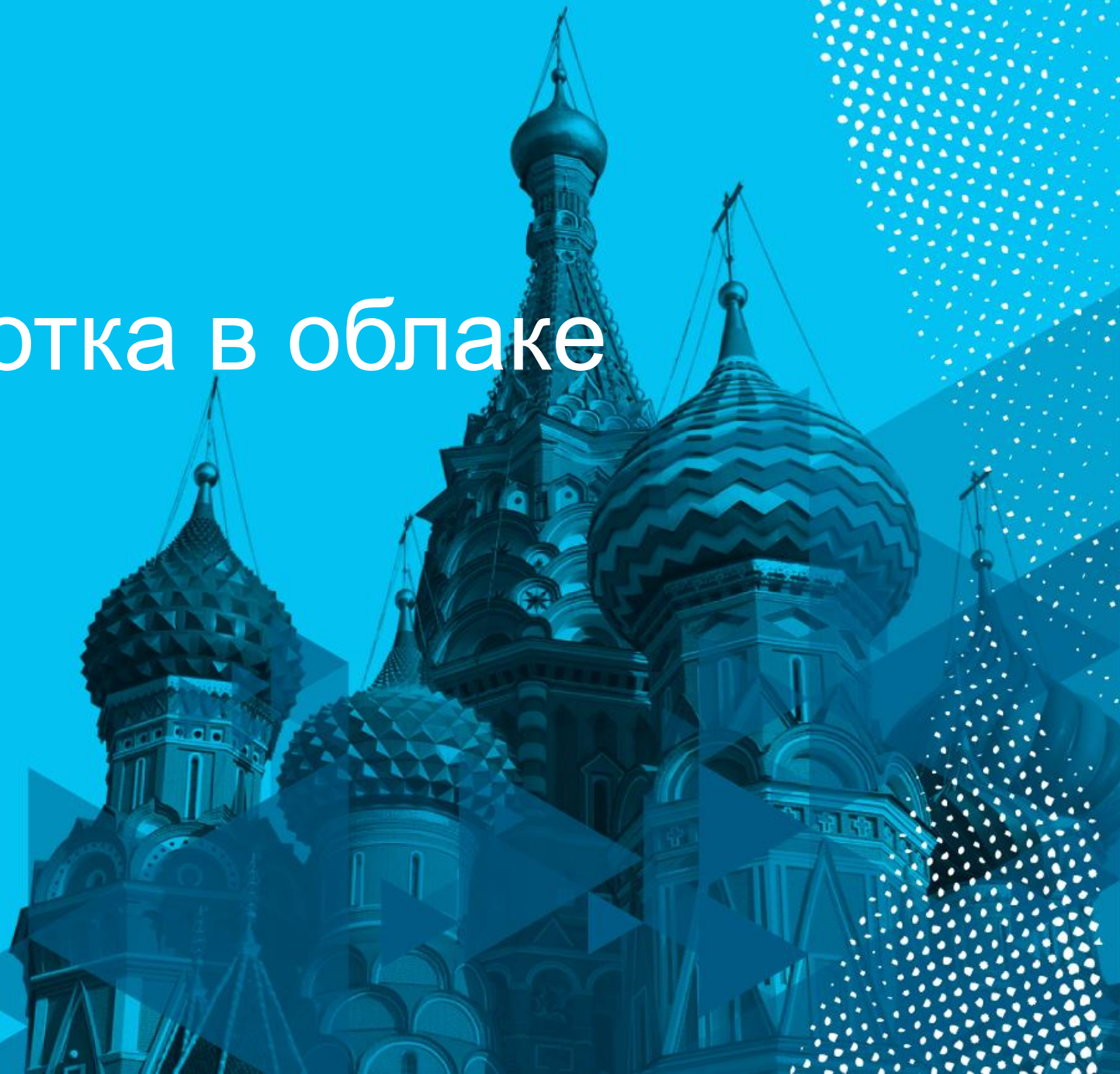
- Данные с возможностью поиска в Table Storage или SQL Azure
  - Индексация (SQL Azure)
  - Нет оплаты за запрос (SQL Azure)
  - Ниже расходы на объем хранилища (Windows Azure Table Storage)
- Небольшие изображения в Table Storage
  - Двоичное содержимое менее 64кб
  - Групповые выборки позволяют экономить на транзакциях
- Полные изображения в Blob Storage
  - Большие объемы данных
  - Есть возможность отдавать изображения напрямую по HTTP и CDN

Гибридное разбиение

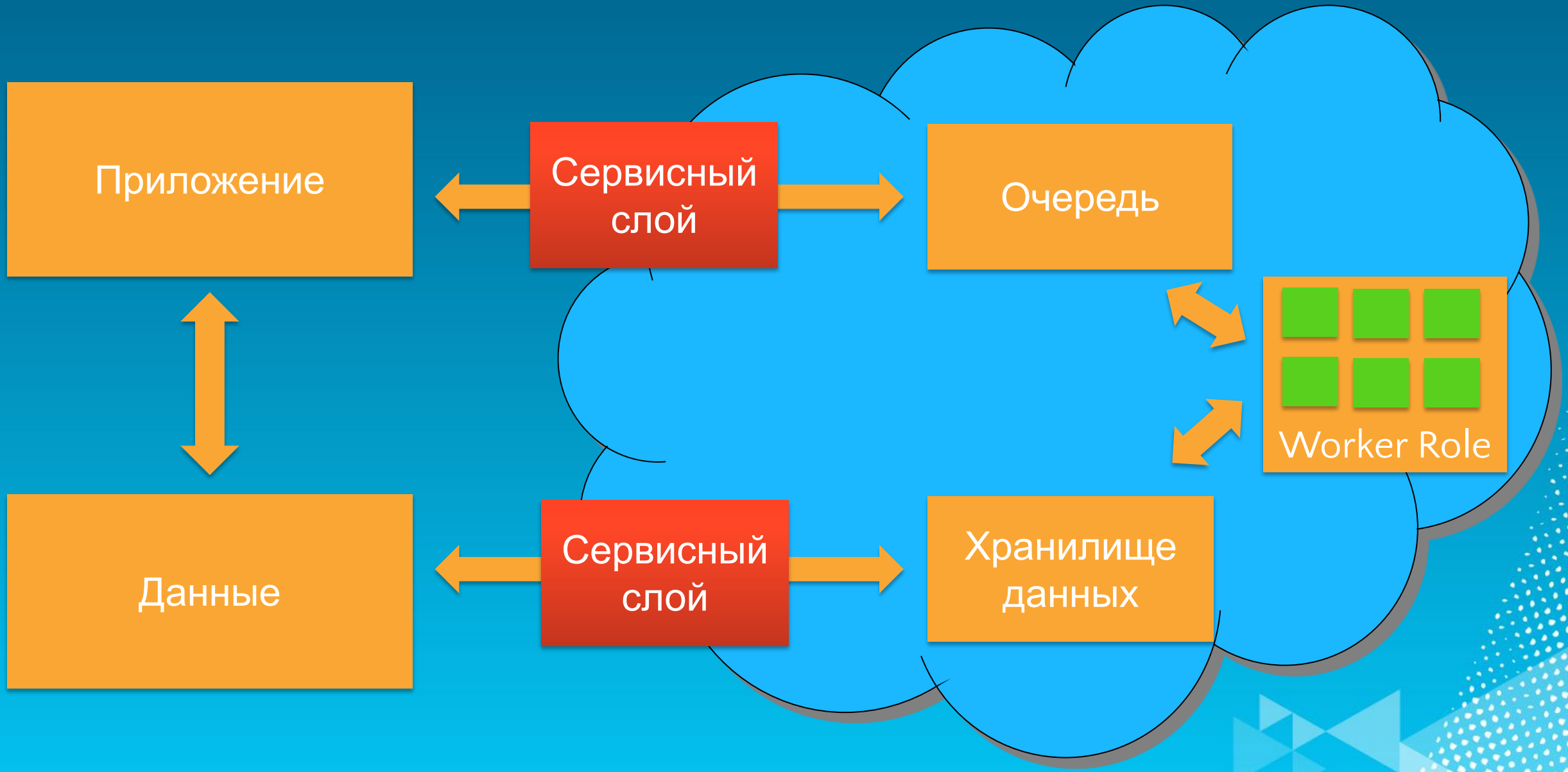
FirstName	LastName	Email	Thumbnail	Photo
David	Alexander	davida@contoso.com	3kb	3MB
Jarred	Carlson	jaredc@contoso.com	3kb	3MB
Sue	Charles	suec@contoso.com	3kb	3MB
Simon	Mitchel	simonm@contoso.com	3kb	3MB
Richard	Zeng	richardz@contoso.com	3kb	3MB



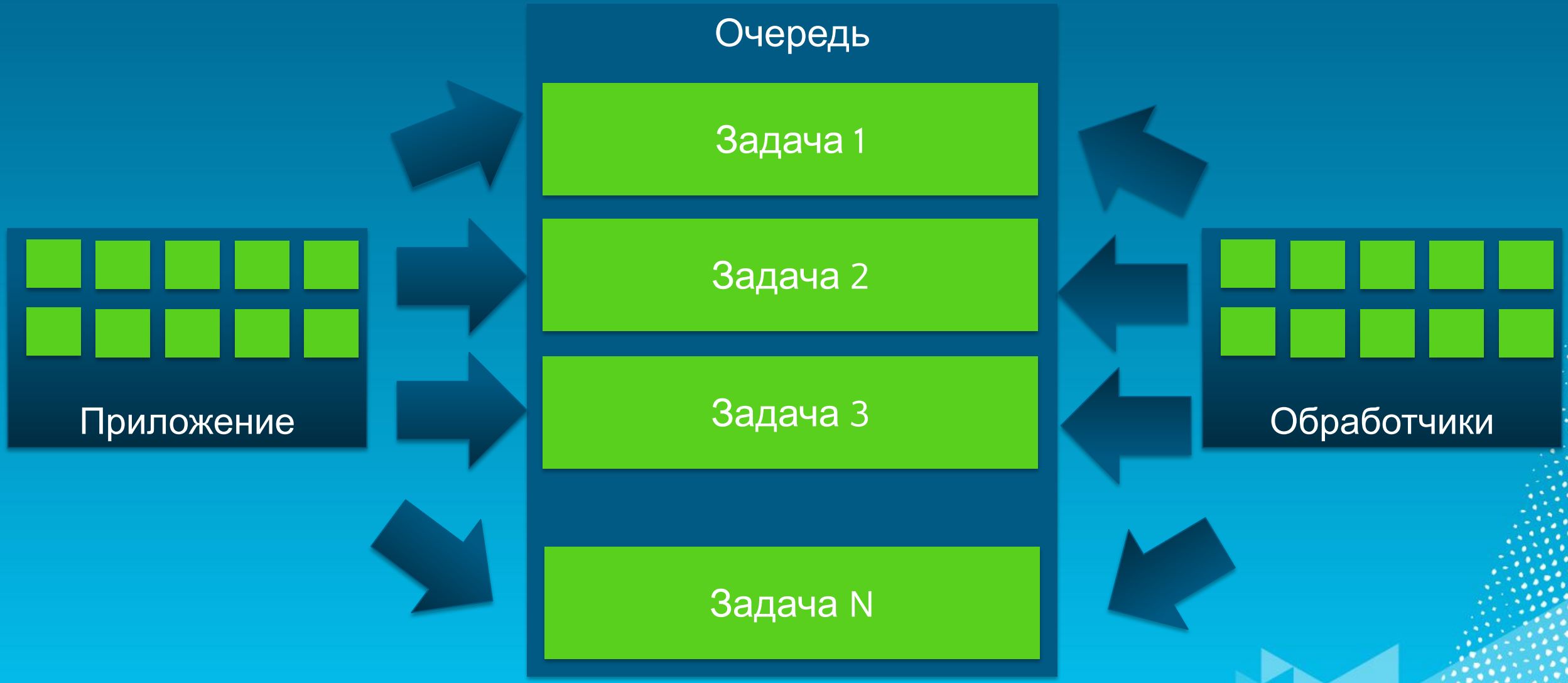
Фоновая обработка в облаке



Фоновая обработка в облаке



Асинхронная обработка в облаке



Проектирование: очереди в облаке

- Основные проблемы обработки в очереди
  - Повторная обработка сообщения
  - Многократные попытки обработать сообщения, вызывающие сбои обработки
  - Простой ресурсов обработчиков сообщений
  - Большие объемы данных, подлежащих обработке

Повторная обработка сообщений

- Проблема: сообщение обработано Worker, результат записан, однако Worker не удалил сообщение из очереди.
- Решение: ведение лога со статусом обработки сообщений.
  - Уникальный идентификатор транзакции
  - Запись результата в рамках одной транзакции с обновлением лога обработки сообщения.
  - В SQL Azure – транзакции уровня базы данных
  - В Table Storage - Entity Group Transaction в рамках одной партиции

Сообщение, вызывающее ошибки

- Проблема: сообщение вызывает сбой при обработке, «выбивая» по очереди Worker из пула.
- Решение: проверка значения счетчика количества попыток обработать сообщение (DequeueCount) и установка лимитов на количество попыток.
  - Проверка DequeueCount должна быть первой операцией!
  - «Проблемные сообщения» можно записывать в отдельную очередь или лог для последующего анализа.
  - В случае, если «проблемное сообщение» может быть исправлено, его можно вернуть в основную очередь (например, снова доступны необходимые ресурсы).

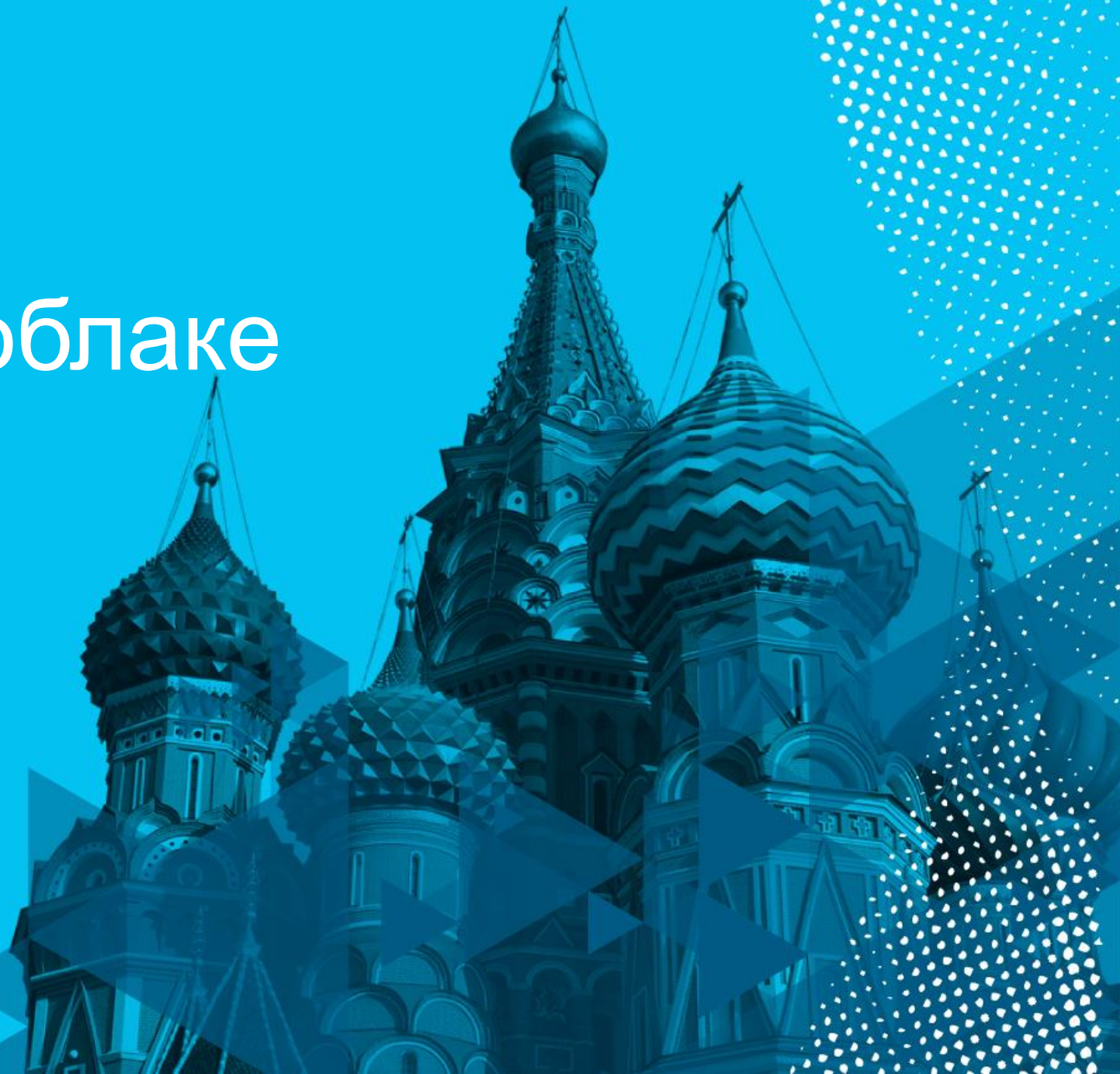
Простой ресурсов обработчиков сообщений

- Проблема: есть несколько типов обработчиков сообщений, часть из которых не загружена на 100%.
- Решение: использование общей очереди сообщений для разных типов задач с указанием типа задачи в самом сообщении.
  - Динамическая загрузка сборки под каждую конкретную задачу.
  - Загрузки сборки в новый AppDomain, чтобы не нарушать работу всего Worker в случае сбоя обработки.
  - Новые типы задач можно добавлять загружая новые сборки в Blob Storage, соответственно при изменении кода одной сборки не нужно обновлять все решение.

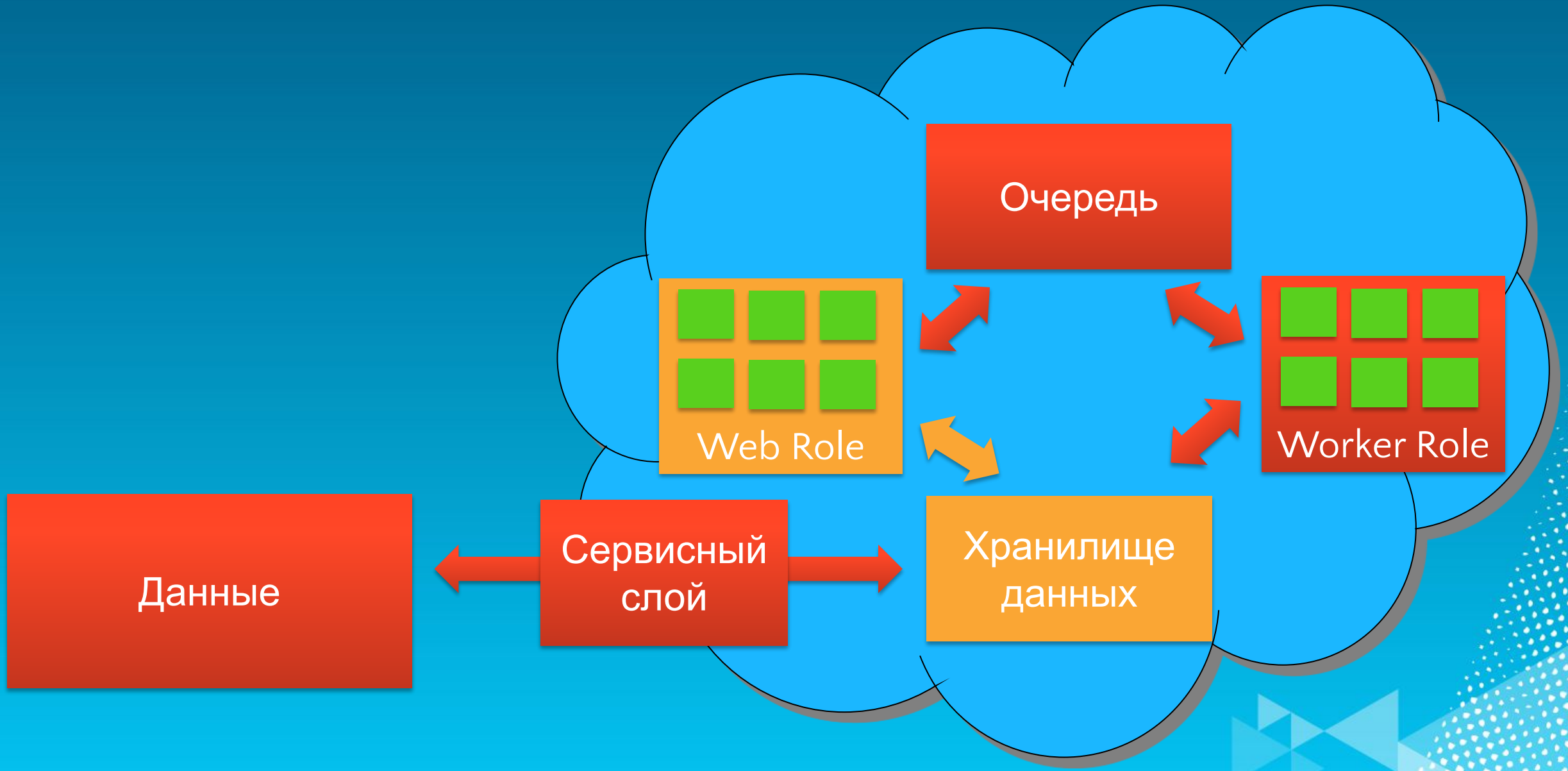
Большие объемы данных

- Проблема: задача требует обработки слишком большого объема данных.
- Решение: разбиение всего объема данных на части.
  - Паттерн MapReduce
  - Разбиение на части с уникальными идентификаторами
  - Финальная стадия – объединение результатов обработки.
  - В этом случае обработка происходит асинхронно, то есть подходит только для тех задач, которые могут быть разбиты на независимые компоненты.

Приложение в облаке



Приложение в облаке



Проектирование: приложение в облаке

- Карусельная диспечеризация запросов
  - Не гарантируется, что последовательные запросы приходят одной машине
  - Каждый элемент страницы может быть получен из разных источников (включая Ajax обновления)
- У всех Web Roles должен быть один Machine Key для хеширования View State
  - Обеспечивается Windows Azure Fabric
- Мульти-тенантность

Общее владение состоянием

- AppFabric Caching
  - Microsoft.Web.DistributedCache
- SQL Azure
  - Два обращения в базу (чтение и запись) на каждый запрос
  - Постоянное хранилище, нет оплаты за запрос
- Table Storage
  - Требуется написание соответствующего провайдера
  - Требуется оплата за транзакции
- Cookies
  - Избыточная нагрузка (на каждый запрос отправляется cookie, к статическим ресурсам в том числе)

Мульти-тенантность

- Проблема: несколько клиентов используют один сервис, требуется обеспечить разные базы данных.
- Решение: привязка базы данных к DNS имени.
  - Вариант 1: набор A-записей
  - Вариант 2: CNAME для *.domain

Загрузка файлов в ASP.NET

- Проблема: ASP.NET буферизует загружаемые файлы во временную директорию, в Windows Azure для веб-роли доступно не более 100 Мб локального хранилища.
- Решение: создать собственный механизм загрузки.
  - Вариант 1: IHttpHandler для буферизации загружаемого файла в Storage или на подключенный диск
  - Вариант 2: загружать непосредственно в Blob Storage со стороны клиента (например, используя контрол на Silverlight)

Заключение – требования к архитектуре

- Слабая связанность
  - Автономные компоненты, общающиеся сообщениями
- Масштабируемость
  - Независимое дублирование компонентов
- Отказоустойчивость
  - Независимая работа компонентов
- Параллелизм
  - Асинхронная обработка задач
- Сохранение целостности данных
  - Валидация данных и сообщений

Полезные ссылки

- Документация по Windows Azure
  - http://msdn.microsoft.com/en-us/library/windowsazure/
- Azure Design Patterns
  - http://azuredesignpatterns.com/
- Пример архитектуры для Azure
  - http://cloudsample.codeplex.com/

Обратная связь

Уважаемые участники!

Ваше мнение очень важно для нас!

В блокноте, который находится в инфопаке участника, вы найдете анкету для оценки докладов

Пожалуйста, оцените доклад и сдайте анкету при выходе из зала модератору

Для участия в конкурсе заполненных анкет, отметьте в анкете номер, который указан на вашем бейдже

Спасибо!

Вопросы

- ARC208
- Гайдар Магдануров
  - Руководитель направления веб-технологий
  - gaidarma@microsoft.com
  - www.radiag.ru

- Вы сможете задать вопросы докладчику в зоне Microsoft в зале №17 в течение часа после завершения этой сессии

Microsoft®
tech·ed
Russia | 2011

9–10 НОЯБРЯ 2011
МОСКВА