

Распределенные приложения и Azure Service Bus



Денис Иванов
@denisivanov

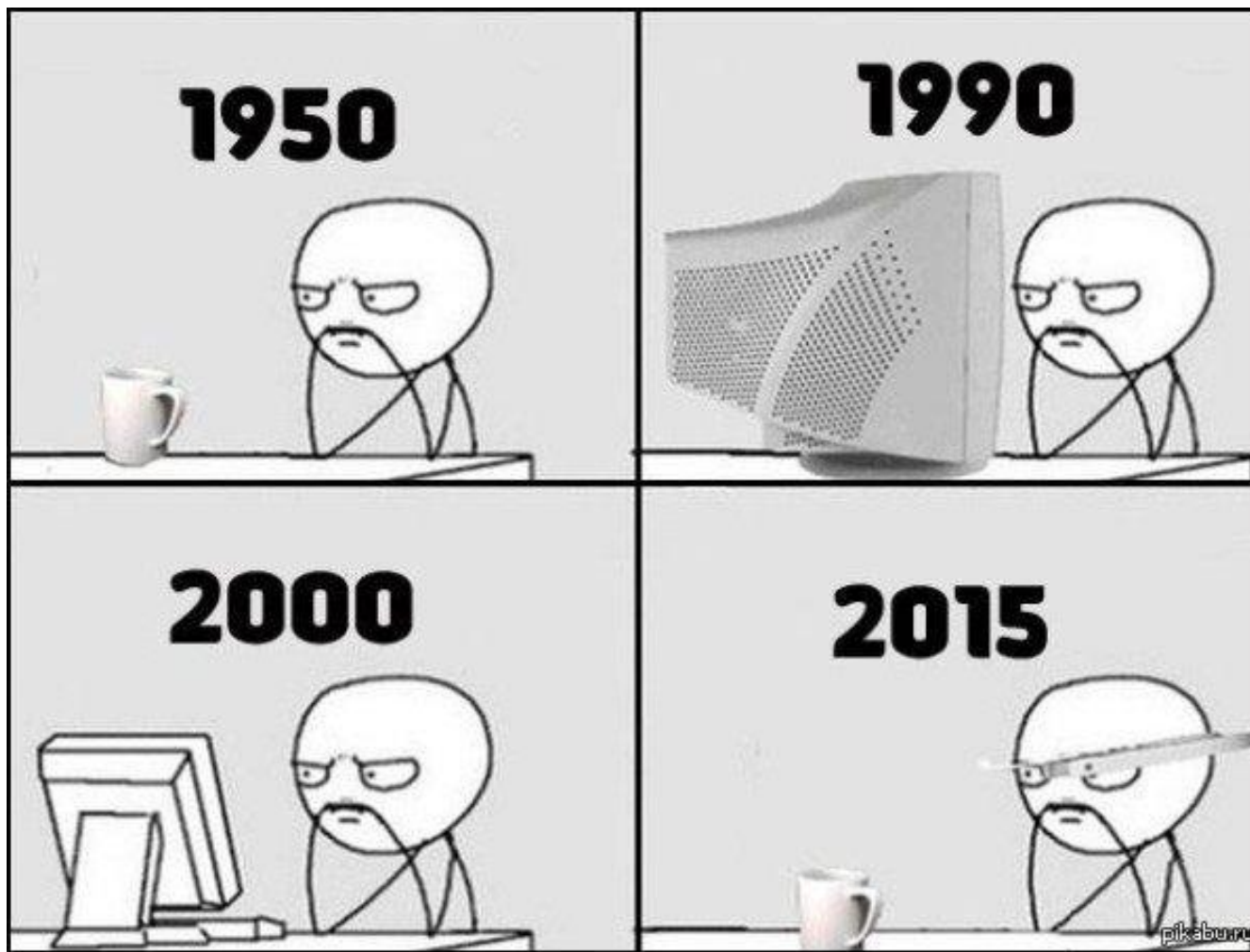


Microsoft®
.NET



Класс приложений

- Enterprise
- Важна предметная область
- Автоматизация бизнес-процессов
- Многоэтапные бизнес-операции





Эволюция

- Клиент-серверные, синхронное выполнение



Эволюция

– Фоновые операции



Эволюция

– Декомпозиция бек-энда



Эволюция

– Несколько фронт-эндов



Проблема





Проблема





Проблема





Подходы к проектированию приложений

- Эволюционный/легаси-система
- На основе прошлого опыта
- По примеру существующего приложения
- DDD



Message driven architecture

- Bounded contexts
- Aggregates/aggregate roots
- Commands and events



Демо-приложение

2 bounded contexts

Продажа рекламы

- Подбор параметров заказа
- Создание заказа
- Жизненный цикл заказа



Демо-приложение

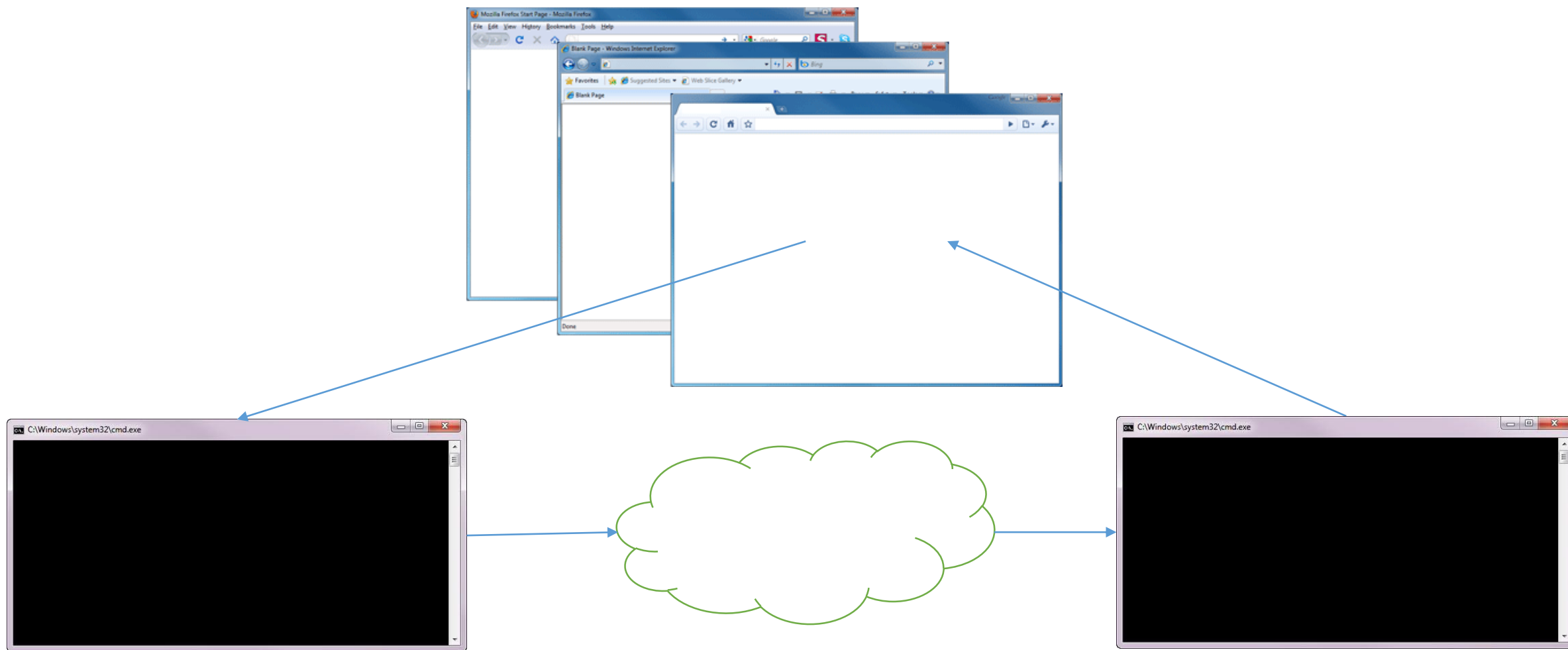
2 bounded contexts

Производство рекламы

- Валидация заказа
- Оформление рекламных материалов



Топология демо-приложения





Используемые технологии

- OWIN
- SignalR
- Service Bus For Windows 1.1

ON AIR



<http://devday-net-demo.azurewebsites.net/>



ВОЗМОЖНОСТИ Service Bus For Windows/ Azure Service Bus

- Namespaces
- Brokered/relayed messaging
- Queues/topics
- Subscriptions/routing
- Poison messages (dead letter queue)



ВОЗМОЖНОСТИ Service Bus For Windows/ Azure Service Bus

- Batching
- Duplicate detection
- Retries, delivery delay, message expires
- AMQP
- REST API

- Независимое изменение разных частей приложения
- Отказоустойчивость и динамическая балансировка
- Синхронное или асинхронное выполнение
- Легко реализуемый Task-based UI (wizards)

- Повышение гибкости ведет к усложнению инфраструктуры
- Выберите подходящий способ проектирования
- Определите цели и этапы
- Используйте наиболее удобные инструменты



Вопросы?

Денис Иванов

@denisivanov

<https://github.com/denisivan0v>



Спасибо

<https://github.com/denisivan0v/devday-demo>