

Microsoft

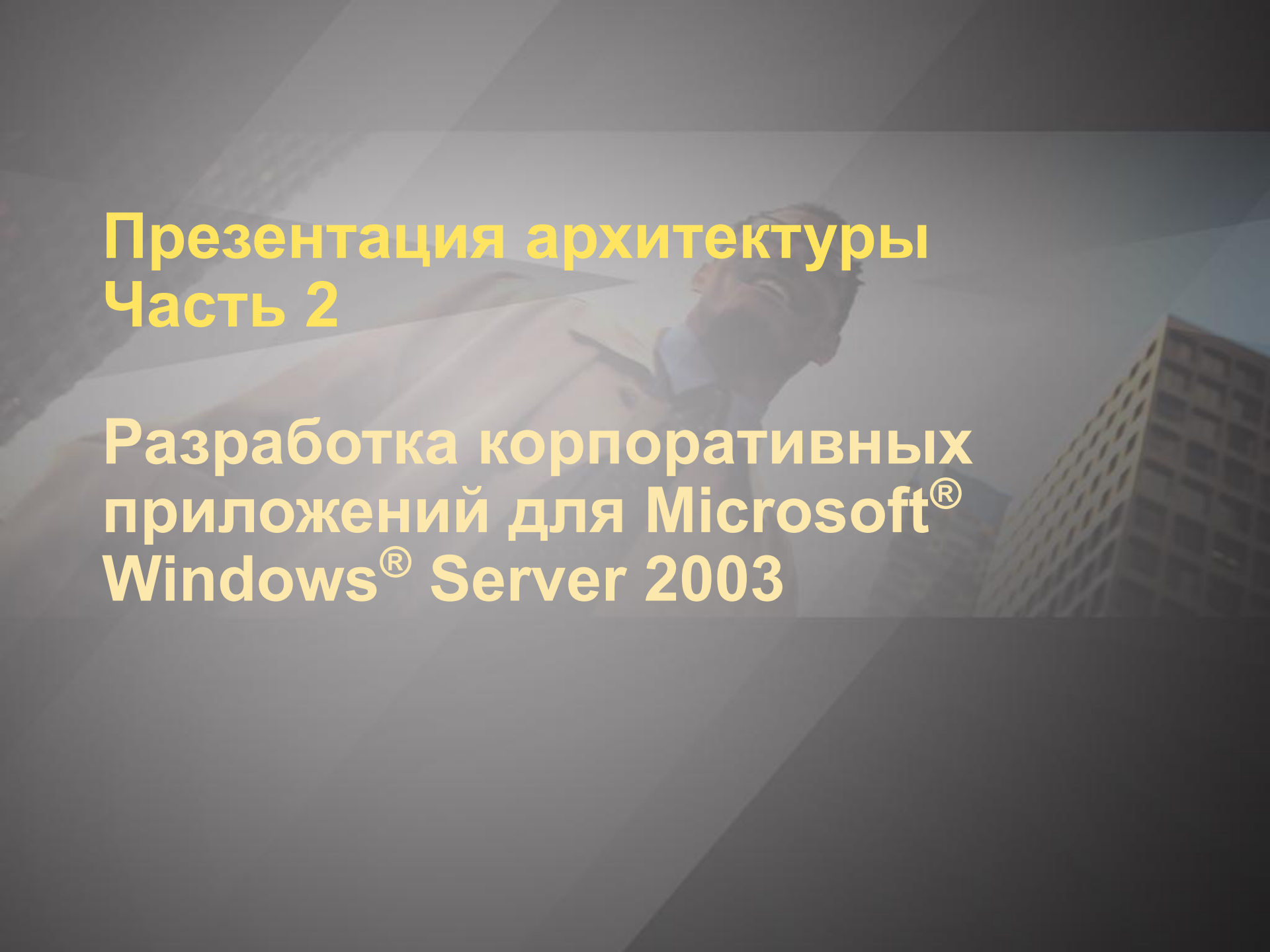
*do more with less*



# Windows Server 2003 **Launch**

  
Microsoft  
**Windows Server 2003**

  
Microsoft  
**Visual Studio.net**

A man in a light-colored suit and tie is looking upwards with a slight smile. In the background, a tall skyscraper with a grid of windows is visible against a clear sky. The image is overlaid with a semi-transparent dark grey layer.

# Презентация архитектуры Часть 2

Разработка корпоративных  
приложений для Microsoft®  
Windows® Server 2003

# Аудитория

- Проектировщики приложений, разработчики и специалисты по инфраструктуре
  - Те, кто отвечает за проектировку, разработку и управление корпоративными системами
  - Разработчики приложений, использующие преимущества Windows Server 2003
  - Те, кто заинтересован в технических подходах корпорации Майкрософт к архитектуре приложений
  - Те, кто заинтересован в построении приложений для архитектуры, ориентированной на службы (SOA)

# Реальные ожидания

- Архитектура – это сложная тема, которая имеет много аспектов и которую можно рассматривать с разных точек зрения
- Данная презентация касается лишь небольшой части вопросов
  - Ее никак нельзя считать всеобъемлющей
- По тем проблемам, которые не будут рассматриваться сегодня, имеются следующие ресурсы
  - Подробная документация, прилагаемая к данной презентации
  - DVD-диск с записью событий
  - Веб-узел MSDN<sup>®</sup> .NET Architecture Center
  - Веб-узел Microsoft patterns & practices

# Основные задачи

- Исследование проблем безопасности с точки зрения их влияния на архитектуру приложения
- Обзор влияния расширенных возможностей кэширования в Microsoft® .NET
- Обсуждение вопросов надежности и масштабируемости
- Обзор методологий доступа к данным
- Обсуждение планирования управления транзакциями

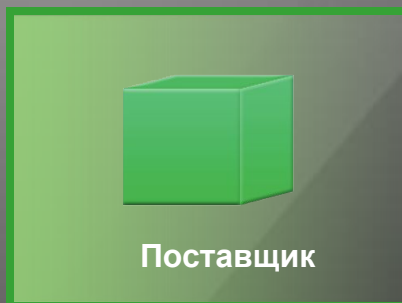
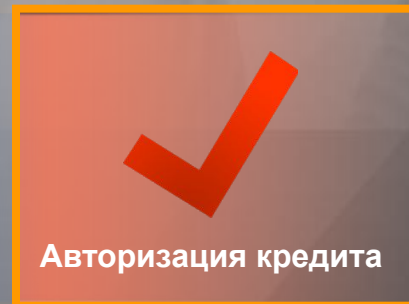
# Содержание занятия

- **Службы**
- **Безопасность**
- **Управление состоянием сети и транзакциями**
- **Доступность и масштабируемость**

# Создание служб:

## Вопросы проектирования

- Распределение данных и установка границ
  - Доверие / безопасность
  - Транзакции и состояние сети
  - Доступность и масштабируемость





# Безопасность

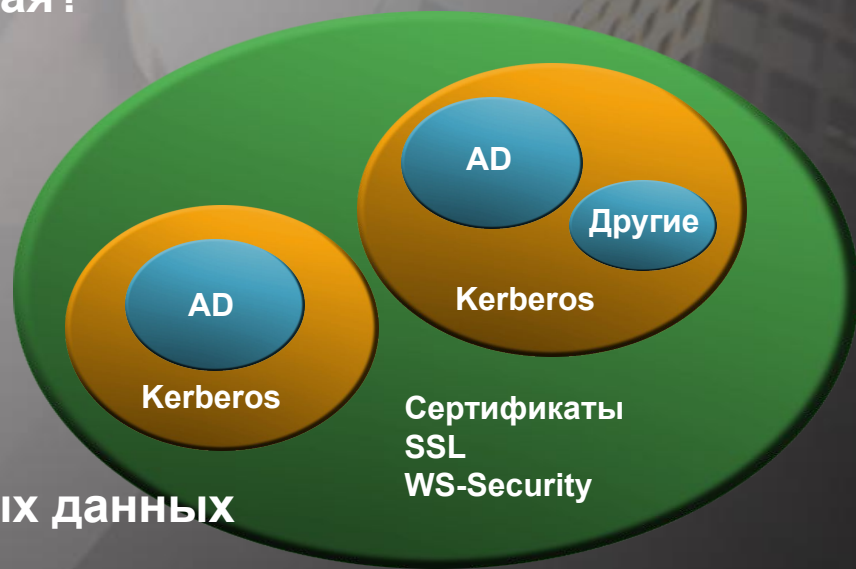
- Установка и подключение границ доверий
- Ключевые области
  - Проверка подлинности
  - Авторизация
  - Безопасная связь



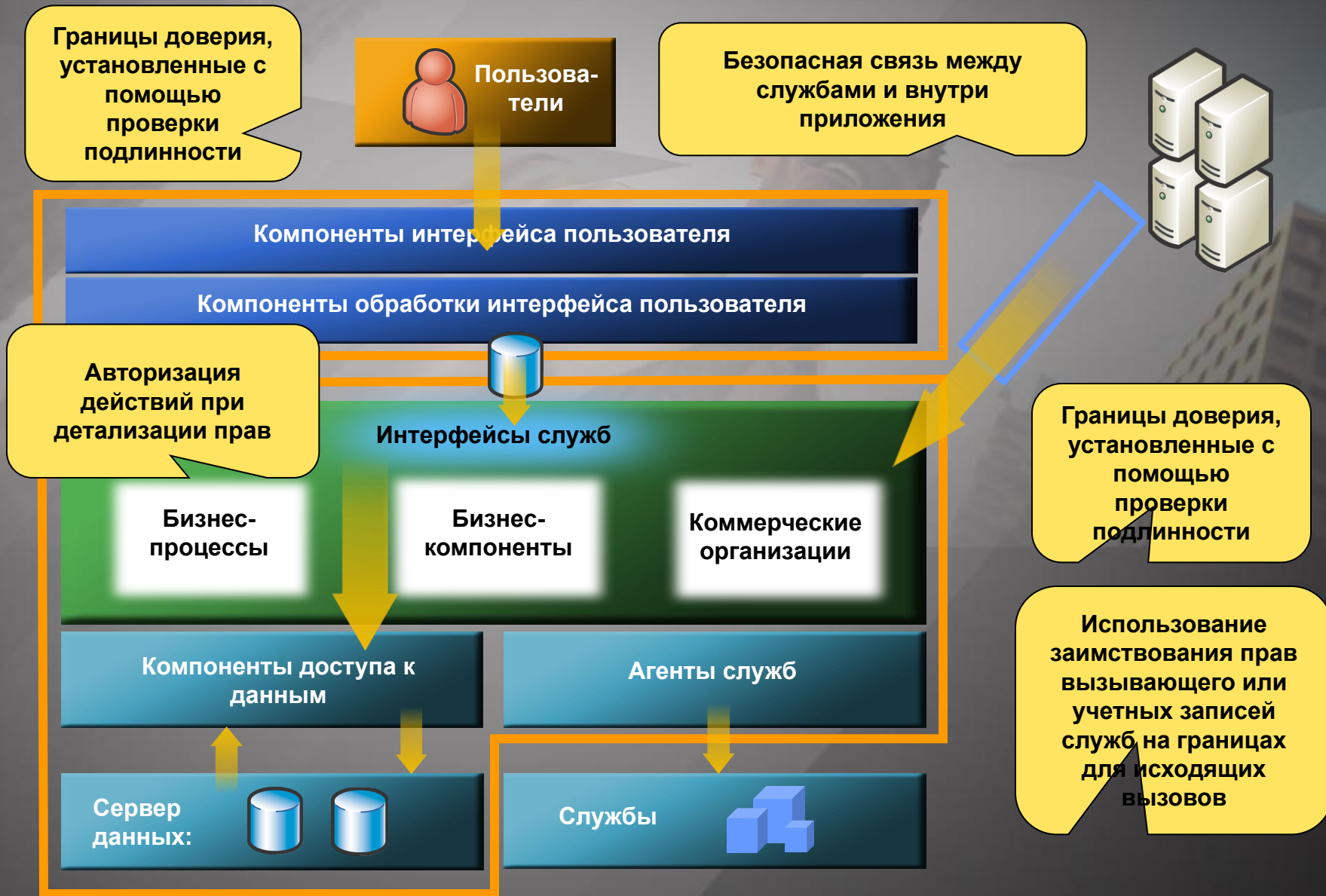
# Безопасность



- Проверка подлинности
  - Специальная или платформенная?
- Авторизация
  - Общая инфраструктура и управление
  - Объединение операций
- Безопасная связь
  - Внутри приложения
  - Между службами
- Аудит
  - Сохранение идентификационных данных
- Управление профилями
  - Объединение, масштабируемость

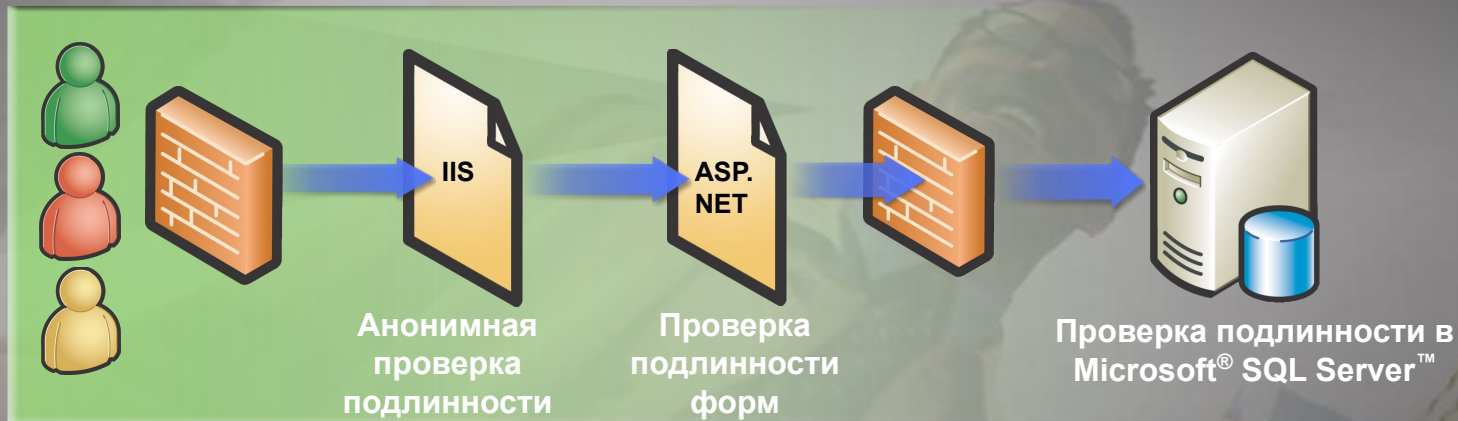


# Проектирование системы безопасности



# Безопасность

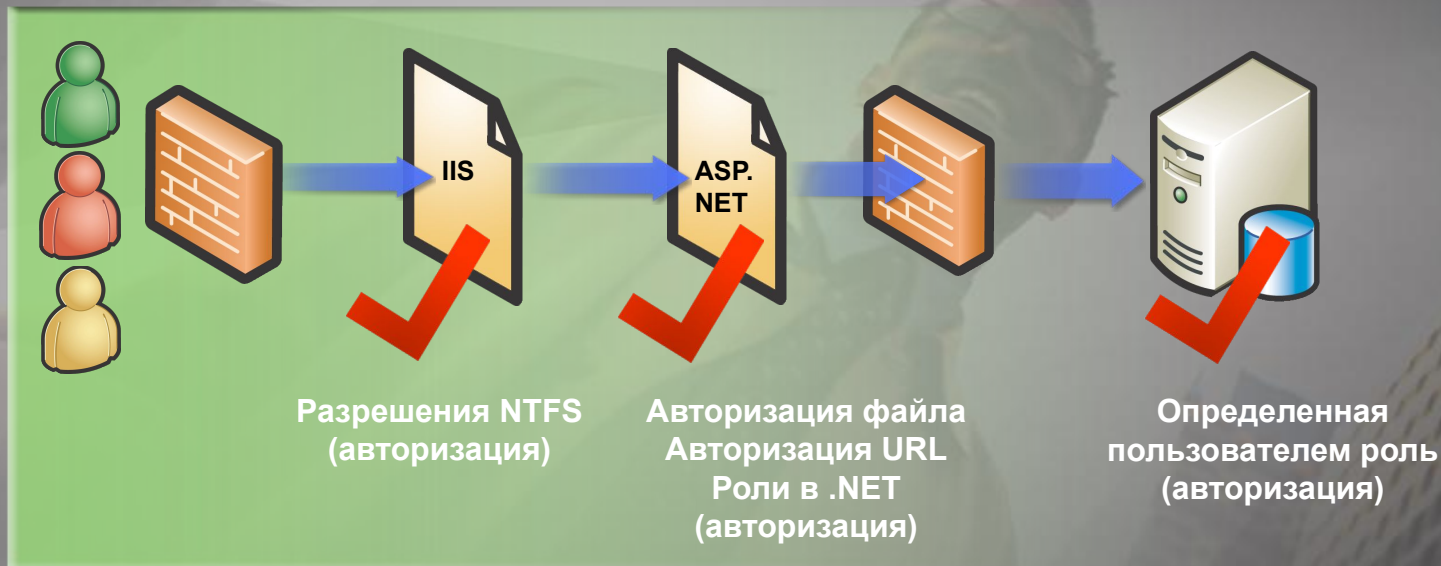
## Проверка подлинности



- Где должна производиться проверка подлинности?
  - Microsoft Internet Information Server (IIS) для приложений интрасетей
  - Проверка подлинности на основе форм в Microsoft® ASP.NET
- Проверка подлинности в Microsoft® .NET Passport

# Безопасность

## Авторизация



- **ASP.NET на основе ролей**
  - Совместимость с формами и цифровым паспортом .NET
  - Доступность обычных объектов и объектов Windows
  - Возможность настройки с помощью `IPrincipal` и `IIdentity`

# Безопасность

## Авторизация

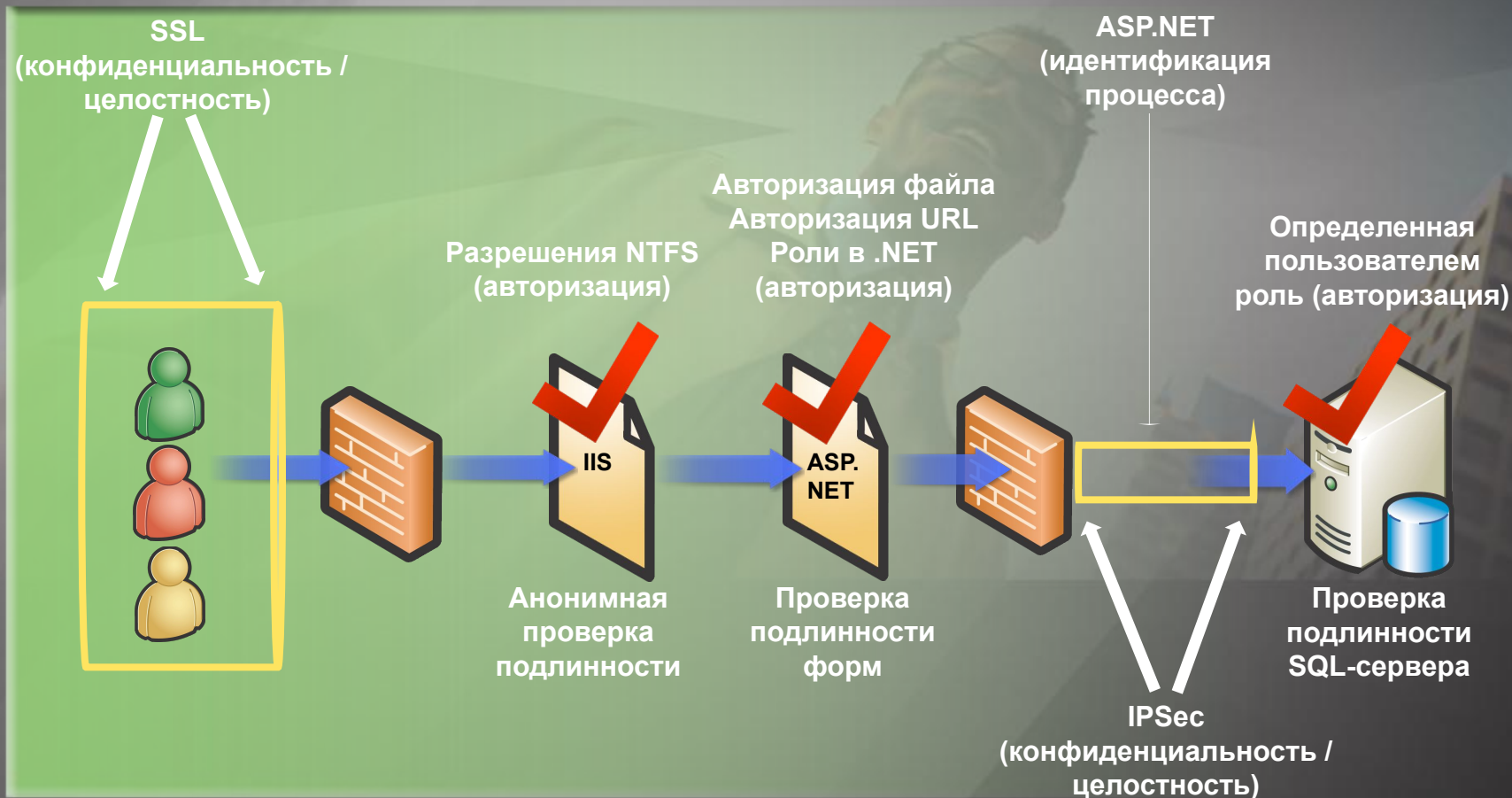


- **ASP.NET на основе ролей**
  - **Определяемые пользователем роли баз данных SQL-сервера**
  - **Роли приложений SQL-сервера**



# Безопасность

## Двухуровневый сценарий



- Для защиты глобальной сети использован протокол SSL
- Для защиты локальной сети использован брандмауэр + IPSec

# Безопасность

## Решения по безопасности работы в Интернете

- Обычная архитектура Интернета
  - Проверка подлинности .NET на основе форм
  - Проверка подлинности IPPrincipal на основе ролей
  - Протокол SSL и стандарт безопасности протокола IP (IPSec).
- Двухуровневая модель имеет ограниченное количество параметров проверки подлинности базы данных



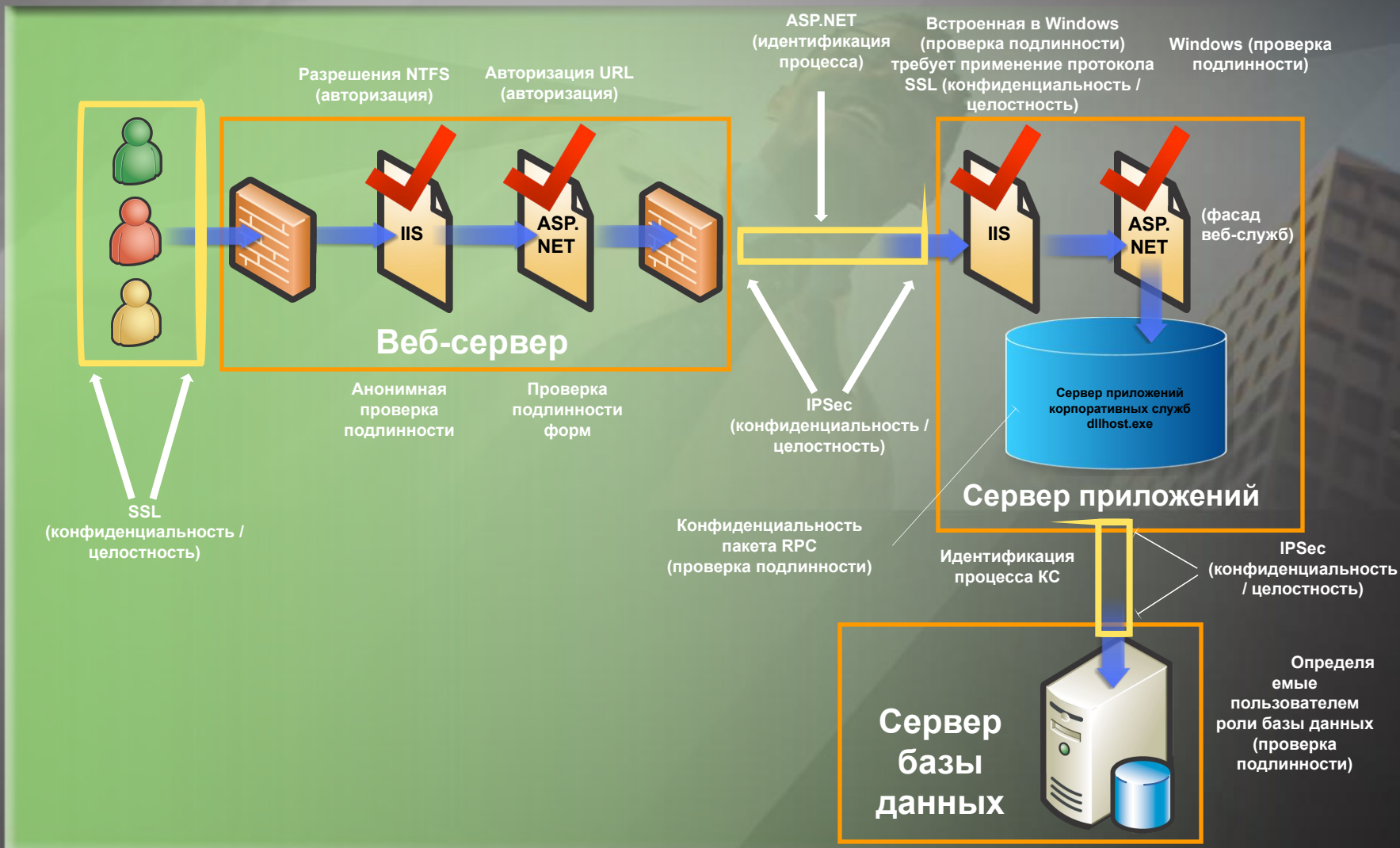
# Безопасность

## Безопасность сервера приложений

- Большая надежность, адаптируемость, масштабируемость
  - Уровни свободно связаны между собой с помощью протокола SOAP и других интернет-протоколов
  - Доступна дополнительная защита: проверка подлинности и роли на основе службы каталогов Microsoft® Active Directory®, а также AuthzMan
  - Дополнительная масштабируемость на физическом уровне

# Безопасность при использовании сервера приложений

## Развертывание с тремя уровнями



# Краткие рекомендации.

## Безопасность

- Следуйте инструкциям, приведенным в статье «Разработка безопасных приложений ASP.NET»
- Используйте учетные записи служб, когда не требуется идентификация пользователя для доступа к базе данных
- Используйте повышенную безопасность и службу каталогов Active Directory на бизнес-уровне
- Используйте диспетчер авторизации для применения дополнительных средств авторизации
- **Советы и рекомендации**
  - Руководство «Разработка безопасных приложений ASP.NET»

<http://msdn.microsoft.com/library/en-us/dnnetsec/html/secnetlpMSDN.asp>

# Содержание занятия

- Службы
- Безопасность
- Управление состоянием сети и транзакциями
- Доступность и масштабируемость

# Основные задачи архитектуры, ориентированной на службы (SOA)

- Инкапсуляция службами всех функций управления состоянием
- Разговор с помощью сообщений, передаваемых через ненадежные протоколы
- Основные проблемы
  - Поддержание состояния сеанса связи
  - Управление транзакциями
  - Управление одновременным доступом к данным
  - Кэширование данных для повышения быстродействия, масштабируемости и доступности

# Состояние и транзакции



# Связь на основе сообщений

- **Сообщения – это деловые документы, обмен которыми необходим для осуществления бизнес-процесса**
- **Службы могут требовать от других служб сохранения данных**
  - **Пример: код покупателя**
- **Сеанс связи на основе сообщений требует сохранения состояния и его восстановления в случае необходимости для осуществления деловых операций**
  - **Этот процесс напоминает разговор между людьми**



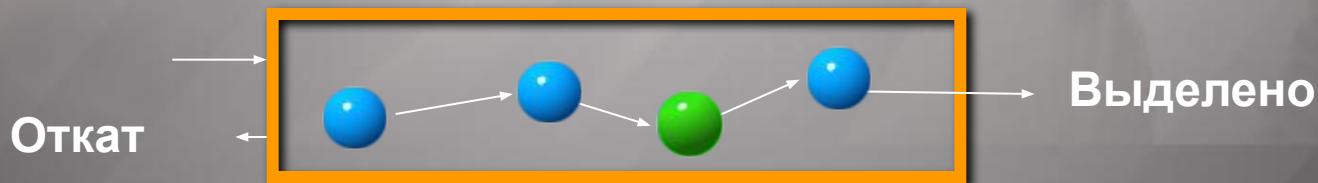
# Состояние разговора

- **Выделение управления состоянием разговора в бизнес-процессах**
  - Вне используемой для ведения бизнеса базы данных или интерфейса служб
  - Использование расписаний сервера Microsoft® BizTalk®
- **Сохранение однородности бизнес-компонентов**
  - Более благоприятные условия для деловой активности
- **Сообщения могут быть потеряны или доставлены дважды**
  - Если не используется транзактный протокол S & F
  - Или протокол SOAP в MSMQ 3.0 через HTTP
  - Надежность веб-служб в качестве долгосрочной политики

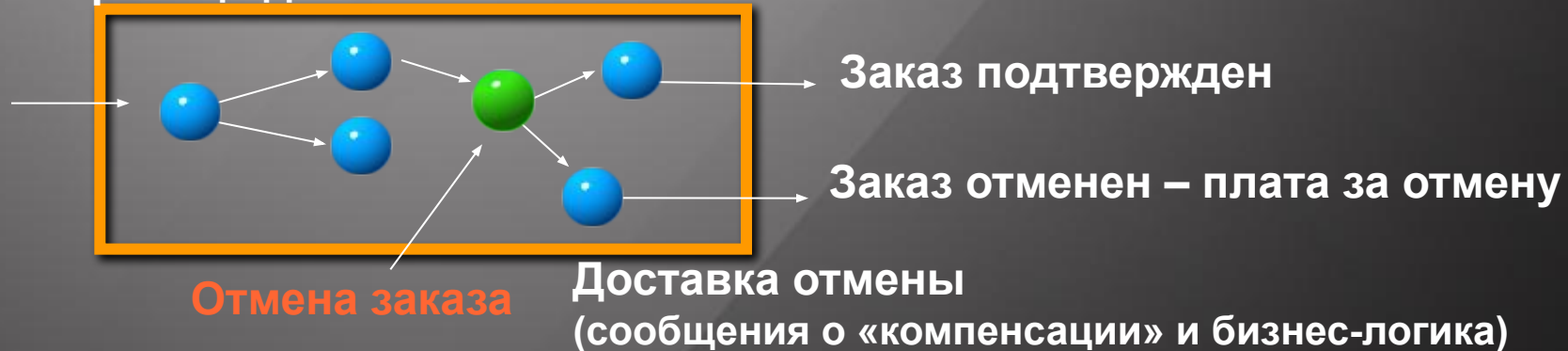
# Управление транзакциями

- Службы как граница области
- Взаимодействие служб через коммерческие транзакции
- Коммерческая транзакция  $\leftrightarrow$  выделение/откат
  - Только не ACID!
- Коммерческие транзакции больше подходят для систем на основе сообщений

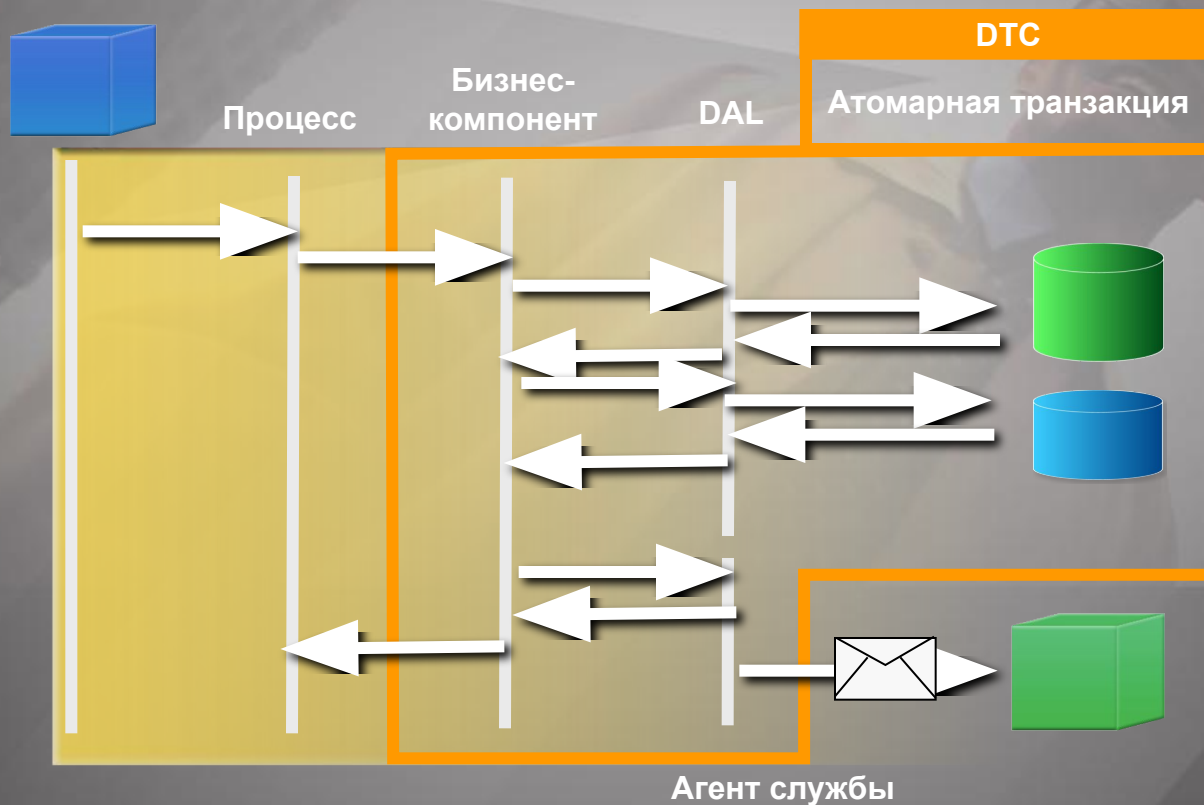
Граница ACID-транзакции



Граница делового сеанса связи



# Модель транзакции



Бизнес-оборот обеспечивает осуществление бизнес-процессов.

Более обособленные бизнес-компоненты инициируют атомарные транзакции и управляют ими.

Диспетчеры транзактных ресурсов принимают участие в атомарных транзакциях.

При условии, что другие службы не являются частью ACID-транзакции (если они не используют MSMQ).

Откаты производятся с компенсациями.

# Управление транзакциями

## Использование ACID-транзакций

- **Использование транзакций только в случае необходимости**
  - Далеко не всякое обновление требует транзакции
  - Чтение интерфейсов пользователей не требует транзакций
- **Обмен данными за пределами границы транзакции отключен и не является актуальным**
- **Требуется рассмотреть тип параллельных подключений**
  - Оптимистический вариант: отключенная среда
  - Пессимистический вариант: подключенная среда

# Управление атомарными транзакциями

## Выбор модели атомарной транзакции



# Предпочтительный вариант блокировки

## Использование одновременного отключенного доступа к данным

- **Целостность плюс одновременность плюс масштабируемость**
- **Эффективно для отключенных сред**
- **Полученные данные являются потенциально устаревшими**
  - **Время (период ожидания или штампы времени)**
  - **Сравнение данных или их отдельных частей**  
Пример: отправка заказа с ценами из устаревшего каталога
- **Необходимость разрешения конфликтов**
  - **Сравнение для определения изменившихся полей**
  - **Автоматически или вручную**  
Пример: обновление записей заказчиков

# Краткие рекомендации.

## Управление транзакциями

- **Руководства по транзакциям**
  - Глава «How to Code ADO.NET Manual Transactions» (Инструкции по кодированию транзакций ADO.NET вручную) в руководстве .NET Data Access Architecture (Архитектура доступа к данным .NET)
  - Статья Designing Data Tier Components and Passing Data Through Tiers (Разработка компонентов уровней данных и передача данных между уровнями)

<http://msdn.microsoft.com/practices/>



# Кэширование

## Роли кэшей в архитектуре, ориентированной на службы

- **Для чего нужно кэширование?**
  - Для повышения быстродействия?
  - Для масштабируемости? Для доступности?
- **Где выполнять кэширование?**
  - Кэширование ASP.NET – страница и фрагмент
  - SQL-сервер – постоянное кэширование
  - MMF – общий доступ к процессам в памяти
  - Таблицы статического кэширования – в пределах домена приложения
- **Какие данные следует кэшировать**
  - Нетранзактные справочные данные
  - Редко изменяемые данные
  - Данные, применимые к максимальному числу пользователей

# Кэширование

## Уровень представления и интерфейсы служб

- **ASP.NET**
  - **Кэширование выходящей страницы**
    - Очень быстрое, но без пользовательской настройки
    - Можно рассматривать как форматированные бизнес-данные
  - **Кэширование фрагмента страницы**
    - Позволяет производить общую настройку страницы
    - Не такое быстрое, но оптимальное для данных с большим объемом форматирования
- **Пакет Windows Forms**
  - **Снижение интенсивности обмена данными между веб-службами**

# Кэширование

## Компоненты доступа к данным

- Три основных сценария
- Кэширование приложений в пределах домена
  - Кэширование статических переменных (таких как ASP.NET)
- Кэширование в пределах компьютера
  - Кэширование в файлы, отображенные в памяти
  - Проблемы взаимодействия
- Кэширование в пределах центра обработки данных
  - SQL-сервер

# Управление состоянием сеанса

- Средство для управления временным состоянием после запроса, полученного сервером через Интернет
- В ASP.NET допускаются три расположения
  - В памяти: единая емкость
  - Служба сеанса: в памяти для веб-фермы
  - SQL-сервер: постоянно используемый веб-фермой

# Содержание занятия

- **Корпоративные службы**
- **Безопасность**
- **Управление состоянием сети и транзакциями**
- **Доступность и масштабируемость**

# Высокая доступность

## Проблемный домен

### Вопросы проектирования

- |   |   |
|---|---|
| <ul style="list-style-type: none"><li>• Ни одной точки сбоя</li></ul>   | <ul style="list-style-type: none"><li>• Балансировка нагрузки</li><li>• Кластеризация</li></ul>   |
| <ul style="list-style-type: none"><li>• Архитектура, ориентированная на службы, и связь на основе сообщений</li></ul> | <ul style="list-style-type: none"><li>• Изолированные сбои</li></ul>  |
| <ul style="list-style-type: none"><li>• Время восстановления</li></ul>  | <ul style="list-style-type: none"><li>• Автоматически восстанавливающаяся система может казаться более доступной</li></ul>  |
| <ul style="list-style-type: none"><li>• Постановка в очередь и кэширование</li></ul>                                  | <ul style="list-style-type: none"><li>• Повышение доступности</li><li>• Повышение воспринимаемого быстродействия</li><li>• Повышение воспринимаемой доступности</li></ul> |
| <ul style="list-style-type: none"><li>• Распределение данных</li></ul>  | <ul style="list-style-type: none"><li>• Требуется решение проблемы масштабирования базы данных</li></ul>  |

# Высокая доступность

- Критерий: планирование сбоев
- Приложение
  - Управление состоянием
  - Быстрое и устойчивое восстановление
- Инфраструктура
  - Ни одной точки сбоя
    - Как на серверах, так и на сетевых элементах
  - Архитектура систем Майкрософт
- Данные
  - Репликация, доставка журнала, распределение данных



# Высокая доступность

## Распределение нагрузки сети

- Для клонированных служб без отслеживания состояния  
Таких как ASP.NET, службы предприятия (COM+)
- Расширение фермы сервера приложений
- Встроенная поддержка для NLB в .NET Server
  - Управление распределением нагрузки
  - Поддержка управления кластером, узлом и портом
  - Возможность распределения на уровне для представления и бизнес-служб с отдельной балансировкой
- Упрощенное управление с помощью Application Center

# Высокая доступность

## Кластеризация

- Предназначено для служб, требующих хранилище данных
- Автоматическое управление передачей нагрузки при сбое
  - Обеспечение гибкости и надежности
  - Масштабируемость: рост системы вверх и распределение данных
- Предназначено не только для серверов баз данных, но и
  - Для всех компонентов, которые поддерживают постоянное состояние
  - Для серверов Exchange
  - Для серверов BizTalk
  - Для MSMQ

# Высокая доступность

## Программные решения

- Доступность и масштабируемость достигаются за счет создания инфраструктуры и приложения
- Постановка сообщений в очередь и асинхронный режим
  - Освобождение клиента от большого объема работы
  - Системные элементы могут быть отключены, не влияя на просмотр приложения клиентом
- Распределение данных
  - Не рекомендуется, но может быть использовано, если базе данных требуется улучшенная масштабируемость

# Краткие рекомендации.

## Высокая доступность

- Балансировка нагрузки и кластеризация влияют на надежность и масштабируемость
- Увеличение масштабируемости возможно за счет создания программного обеспечения, которое позволяет оптимизировать использование ресурсов
  - Высокодоступные системы являются результатом гармоничного сочетания: люди – процесс – технология
  - Люди разрабатывают приложения, для внедрения которых используется определенная технология, а также процесс создания высокодоступной системы
- Архитектура систем Майкрософт

# Итоги презентации

- Microsoft Windows Server 2003 + Microsoft Visual<sup>®</sup> Studio<sup>®</sup> .NET 2003 + советы и рекомендации = законченная платформа приложения
  - Безопасность, управление и связь
  - Быстродействие, доступность и масштабируемость
  - Мощная платформа для разработки многочисленных сценариев приложений
- Ресурсы для разработчиков архитектуры:  
<http://msdn.microsoft.com/architecture/>
- Инструкции и советы по применению:  
<http://msdn.microsoft.com/practices/>

# Важные веб-ресурсы

- Советы и примеры корпорации Microsoft  
<http://msdn.microsoft.com/practices/>
- Веб-узел MSDN .NET Architecture Center  
<http://msdn.microsoft.com/architecture/>
- Visual Studio .NET 2003  
[http://msdn](http://msdn.microsoft.com/vstudio)  
[n.microsoft.com/vstudio/](http://msdn.microsoft.com/vstudio/)
- Windows Server 2003  
<http://www.microsoft.com/windows>



A man in a light-colored suit and tie is laughing heartily, his head tilted back. The background is a blurred cityscape with tall buildings under a bright sky. The Microsoft logo is overlaid in the center.

# Microsoft®

© Корпорация Майкрософт (Microsoft Corporation), 2003. Все права защищены.

Microsoft, Visual Studio, эмблема Visual Studio, Windows, эмблема Windows, BizTalk, Active Directory, SQL Server и MSDN являются охраняемыми товарными знаками корпорации Майкрософт в США и других странах. Другие названия компаний и продуктов, упомянутые в данном документе, могут являться товарными знаками соответствующих владельцев. Эта презентация предназначена только для ознакомления. В этом документе корпорация Майкрософт не предоставляет никаких гарантий, явных или подразумеваемых.