Microsoft
ПРИГОТУЙСЯ
до НОВОГО ДНЯ
Windows Vista
Microsoft Office
Microsoft Exchange Server 2007

Создание приложений для Vista с использованием Windows Communication Foundation

Сергій Байдачний
Фахівець з питань розробки ПЗ
Майкрософт Україна

Віктор Шатохін
Фахівець з питань розробки ПЗ
Майкрософт Україна

for Developers
Office
Windows Vista
Microsoft Exchange Server 2007

Windows Communication Foundation

"Indigo[WCF] is lubrication for software...it reduces friction between the moving parts." -- Don Box

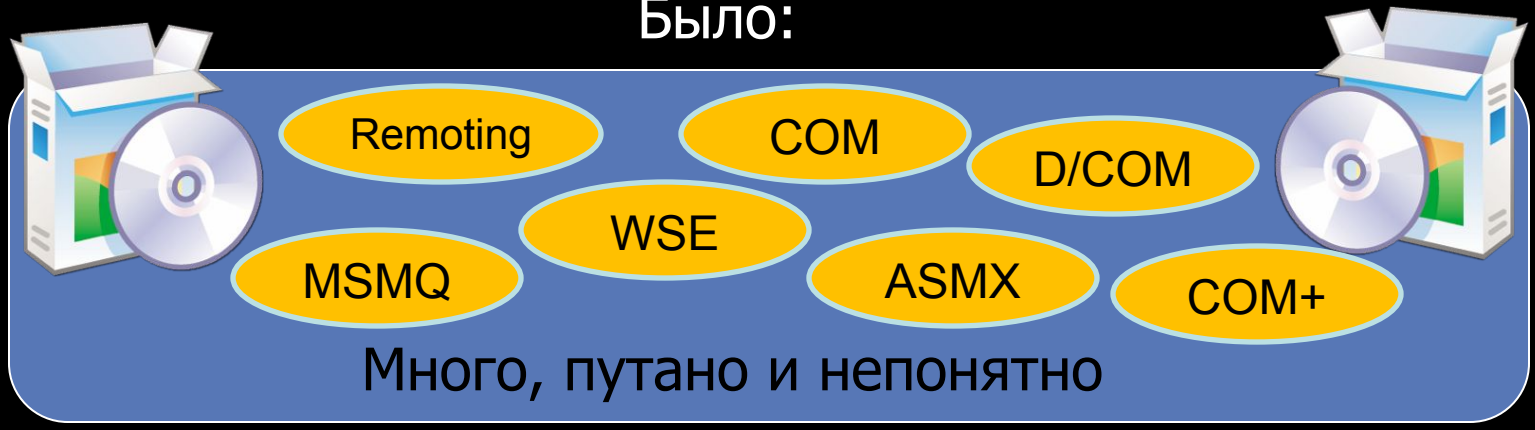
for Developers
Office
Windows Vista
Microsoft Exchange Server 2007

Windows Communication Foundation (WCF)

- Что такое WCF?
  - Унифицированная платформа построения распределенных приложений
  - Позволяет создавать безопасные, надежные и транзакционные Веб-сервисы
    - Взаимодействие с различными платформами
    - Интеграция с существующими решениями
  - Использует и расширяет существующие технологии построения распределенных систем Microsoft
    - Enterprise Services
    - System.Messaging
    - Microsoft .NET Remoting
    - ASMX
    - WSE

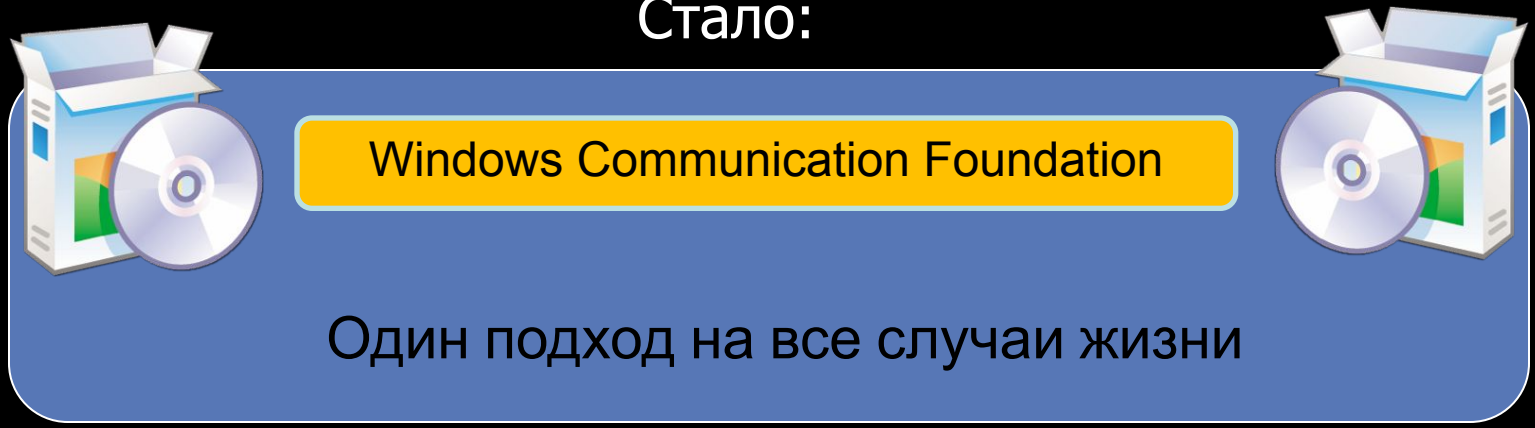
Что такое Windows Communication Foundation?

Было:



Много, путано и непонятно

Стало:



Один подход на все случаи жизни

Разработка в Visual Studio .NET 2003

```
class HelloService {

  [WebMethod]
  public String Hello(String Greeting) {

      X509CertificateCollection collection = new
      X509CertificateCollection();
      IntPtr blob = Marshal.AllocHGlobal(Marshal.SizeOf(
         typeof(CRYPTOAPI_BLOB)));
      IntPtr data = (IntPtr)((int)blob +
        Marshal.SizeOf(typeof(CRYPTOAPI_BLOB)));
      …

      SeqAckRange range = new SeqAcknRange(id, low, high );
      SeqAckRange[] ranges = { range };
      ReliableQueue.ProcessAcks( ranges );
      …

      BeginTransaction (NULL,
        ISOLATIONLEVEL_SERIALIZABLE, 0, pITxOptions,
        &pITransaction);
      …

     return Greeting;
  }
}
```

20,379 строк кода
безопасность

5,988 строк кода
надежная доставка

25,507 строк кода
транзакционность

4,442 строк кода
вспомогательные ф-ии

Всего
56,296

Разработка в Visual Studio .NET 2003 с использованием WSE

```
class HelloService {

  [WebMethod]
  public String Hello(String Greeting) {

    foreach ( SecurityToken tok in
      requestContext.Security.Tokens
    {
      X509SecurityToken token = tok as X509SecToken
    }

    …
    SeqAckRange range = new SeqAcknRange(id,low,high );
    SeqAckRange[] ranges = { range };
    ReliableQueue.ProcessAcks( ranges );
    …

    BeginTransaction (NULL,
      ISOLATIONLEVEL_SERIALIZABLE, 0, pITxOptions,
      &pITransaction);
    …

    return Greeting;
  }
}
```

10 строк кода
безопасность

1,804 строк кода
надежная доставка

25,507 строк кода
транзакционность

Всего
27,321

Разработка в Visual Studio 2005 + WCF

```
[ServiceContract]
class HelloService
{
  [OperationContract]
  [PrincipalPermission(SecurityAction.Demand,
    Role = “Adminstrators")]
  [TransactionFlow(TransactionFlowOption.Mandatory)]
  [OperationBehavior(TransactionScopeRequired = true,
    TransactionAutoComplete = true)]
  String Hello(String Greeting) {return Greeting;}
}
```

```
<bindings>
  <wsHttpBinding>
    <binding name="Binding1“  transactionFlow="true">
      <security mode="Message">
        <message clientCredentialType="Windows“/>
      </security>
      <reliableSession enabled="true" />
    </binding>
  </wsHttpBinding>
</bindings>
```

1 строка кода + **2 конф. эл.**
безопасность

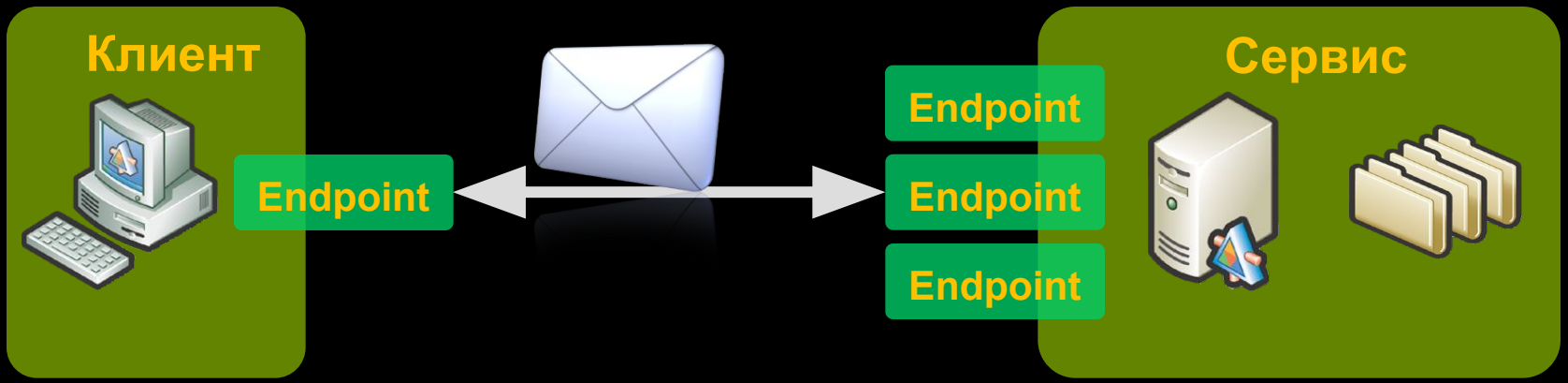
0 строк кода + **1 конф. эл.**
надежная доставка

2 строки кода + **1 конф. эл.**
транзакционность

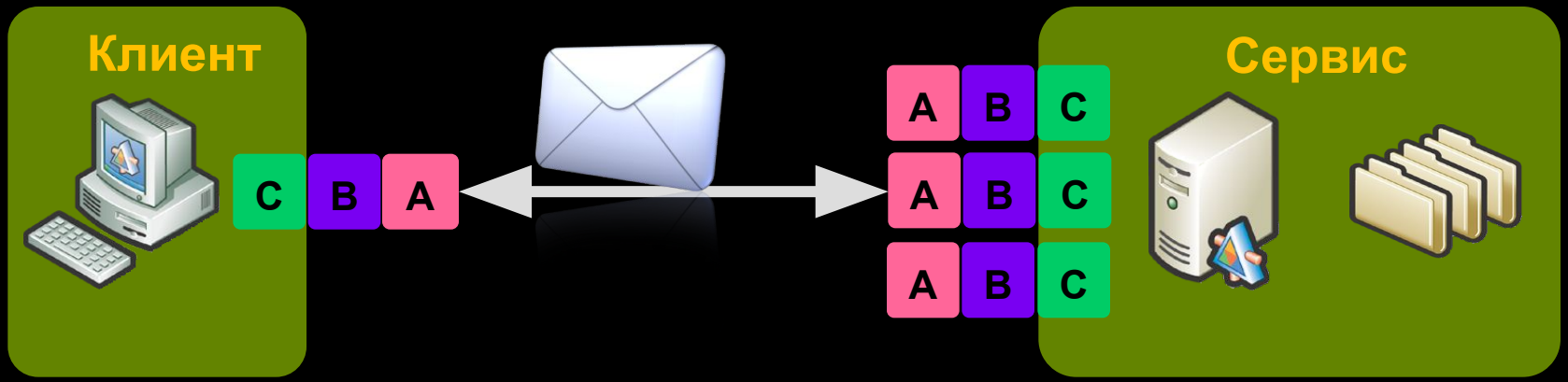
Всего:
3 строки кода+ **4 конф. эл.**

Как это работает?

Конечные точки

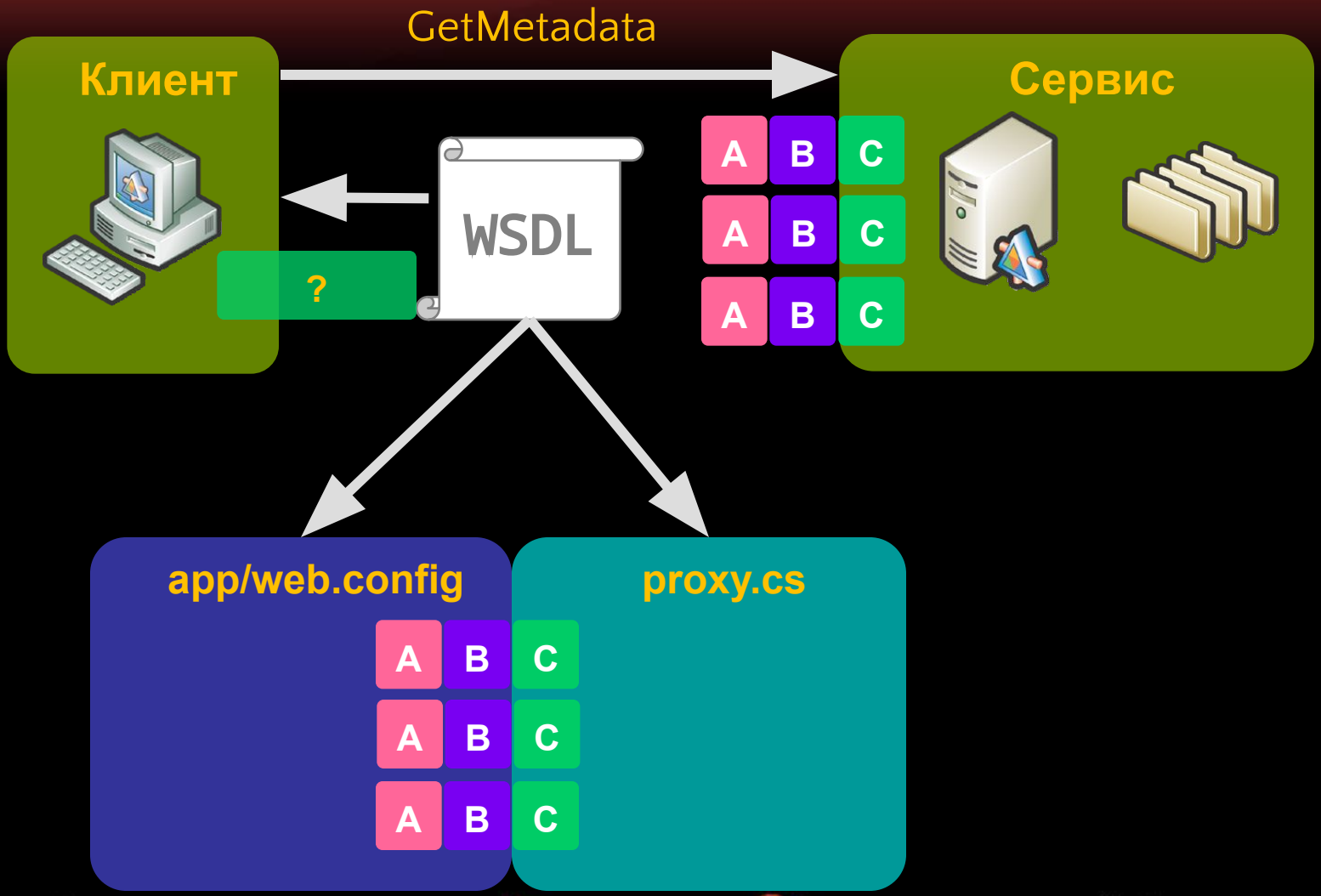


Адрес, Привязка, Контракт

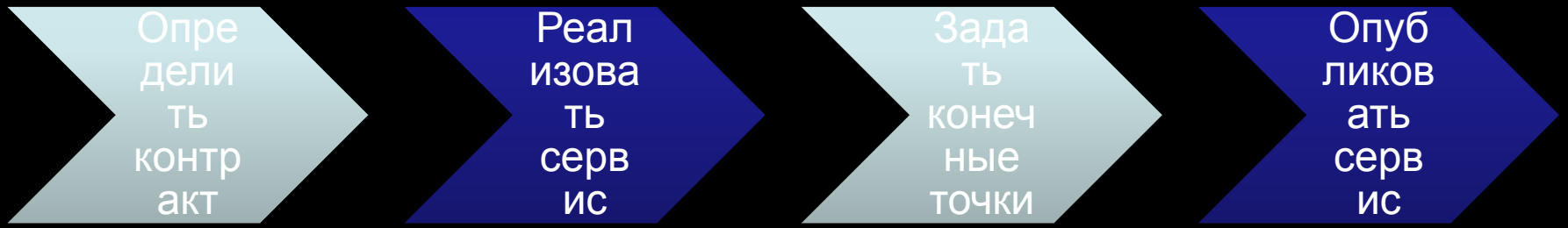


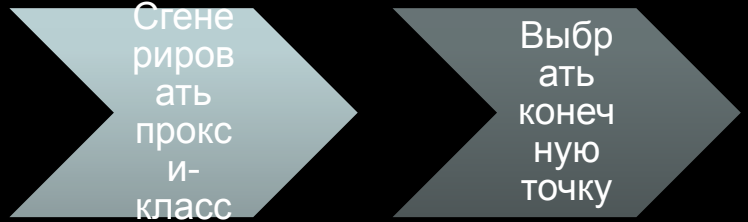
Адрес	Привязка	Контракт
Где?	Как?	Что?

Публикация & конфигурирование конечных точек



Создание сервиса и клиента





for Developers
Office
Windows Vista
Microsoft Exchange Server 2007

Создание сервиса

```
[ServiceContract]
public interface IMyInterface {
   [OperationContract]
   public MyOutputType MyMethod(MyInputType myData);
}
```

1

Определить контракт

```
[ServiceBehavior(InstanceContextMode=Single]
public class MyService: IMyInterface {
   public MyOutputType MyMethod(MyInputType myData) {
     //my code
     …
   }
}
```

2

Реализовать сервис

```
<service
  name="MyService">
  <endpoint
    address="MyAddress"
    binding="netTcpBinding"
    contract="IMyInterface" />
<service/>
```

3

Задать Конечную точку

Контракты

Overview

- Контракт сервиса
  - Определяет список операций
  - Отражение CLR типов в WSDL
- Контракт данных
  - Определяет структуру данных
  - Отражение CLR типов в XSD
- Контракт сообщения
  - Определяет структуру сообщения
  - Отражение CLR типов в SOAP сообщения

Контракты

```
[ServiceContract]
public interface IMyInterface
{
   [OperationContract]
   public MyOutputType MyMethod(MyInputType myData);
}
```

Как возвращаемые и передаваемые типы сариализуются в сообщение?

- Сериализация осуществляется с помощью **DataContractSerializer** по умолчанию
  - Обработка“встроенных” .NET типов автоматически
  - Пользовательские типы необходимо определить как Data Contracts:

```
[DataContract]
public class MyDataContract
{    [DataMember]
     public string MyField;
}
```

- При желании, можно использовать **XmlSerializer**:

```
[ServiceContract]
[XmlSerializerFormat]
public interface IMyInterface
```

Адрес

Привязка	Схема
BasicHttpBinding, WsHttpBinding	http://...
NetTcpBinding	net.tcp://...
NetMSMQBinding	net.msmq://...
NetNamedPipesBinding	net.pipe://...

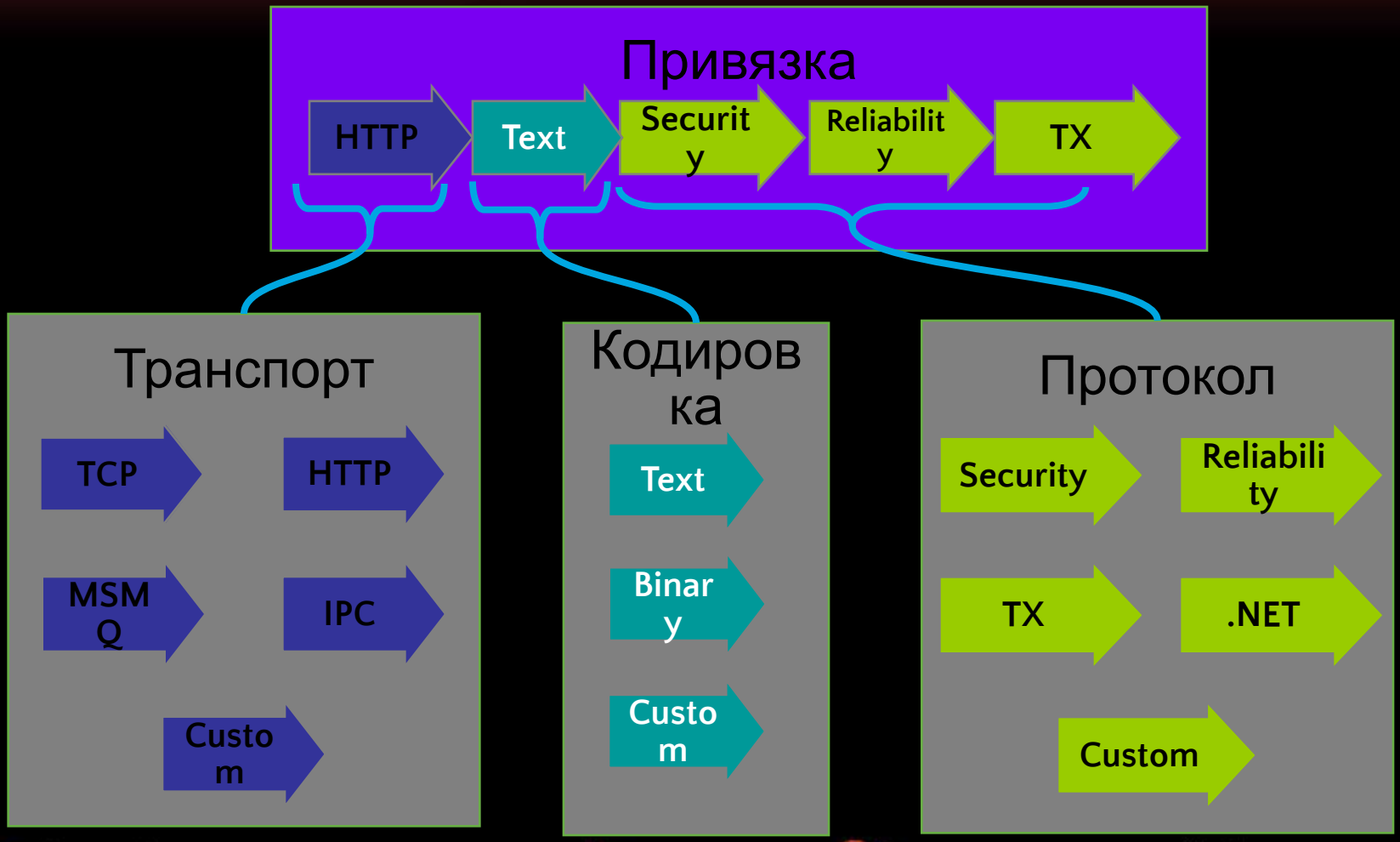
Часть адреса определяет транспортный протокол.

Задание адреса

```
<services>
  <host>
    <baseAddresses>
      <add baseAddress="http://localhost:8000/MyBaseAddress"/>
    </baseAddresses>
  </host>
  <service
    name="MyService">
    <endpoint
      address="MyEndpointAddress"
      binding="wsHttpBinding"
      bindingConfiguration="MyReliableBinding"
      contract="IMyInterface" />
  <service/>
</services>
```

for Developers Office Windows Vista Exchange Server 2007

Привязка



for Developers
Office
Windows Vista
Microsoft Exchange Server 2007

Встроенные привязки

```
<endpoint
    name="MyService"
    address="MyAddress"
    binding="netTcpBinding"
    contract="IMyInterface"
/>
```

Привязка	Назначение
BasicHttpBinding	Basic Profile 1.1 для работы с ASMX
WsHttpBinding	WS-* Поддержка WS-Security, WS-RM, WS-Tx
NetTcpBinding	.NET⇔.NET Безопасность, надежная доставка, дуплекс
NetNamePipesBinding	.NET⇔.NET Безопасность, надежная доставка, дуплекс
NetMSMQBinding	.NET⇔.NET черезMSMQ
NetPeerTcpBinding	.NET Peer ⇔ Peer

Модификация встроенных привязок

```
<services>
  <service
    name="MyService">
    <endpoint
      address="MyAddress"
      binding="wsHttpBinding"
      bindingConfiguration="MyReliableBinding"
      contract="IMyInterface" />
  <service/>
</services>
<bindings>
  <wsHttpBinding>
    <binding name="MyReliableBinding">
     <reliableSession
           enabled="true"
           ordered="true"
    </binding>
  </wsHttpBinding>
</bindings>
```

Создание собственной привязки

```
<?xml version="1.0" encoding="UTF-8" ?>
<configuration>
  <system.serviceModel>
    <services>
      <service
          name="MyService">
          <endpoint
            address="MyAddress"
            binding="customBinding"
            bindingConfiguration="MyCustomBinding"
            contract="IMyInterface" />
      </service>
    </services>
    <bindings>
      <customBinding>
        <binding name="MyCustomBinding">
           <reliableSession advancedFlowControl="true" />
           <security authenticationMode="Kerberos" />
        <binaryMessageEncoding />
        <tcpTransport maxMessageSize="9223372036854775807" />
        </binding>
      </customBinding>
    </bindings>
  </system.serviceModel>
</configuration>
```

Поведение

- Поведение локально
- Разработчики могут изменять поведение
  - Параллельное выполнение, модель создания, …
  - Любые аспекты, которые могут повлиять на корректность работы сервиса
- Администраторы могут изменить поведение
  - Пропускная способность, публикация метаданных, информация о маршрутизации, …
  - Любые аспекты, которые могут повлиять на выполнение сервиса
- Все что можно задать в конфигурационном файле, можно сделать в коде
  - Кто главнее – элементы заданные в коде имеют приоритет над элементами в конфигурационном файле (Атрибуты -> Конфигурационный файл -> Исходный код)

Возможности изменения поведения

- Значения времени ожидания (закрытия, открытия, простоя)
- Параллельное выполнение, создание, привязка к потоку
- Пропускная способность
- Обработка ошибок и исключительных ситуаций
- Имперсонификация, авторизация, аудит
- AutoEnlist, AutoComplete, Timeout, Isolation
- Serialization, MustUnderstand
- Метаданные
- …

Гарантированная доставка

Демонстрация

Really Simple Syndication (RSS)

Платформа RSS

Общий список RSS веб-каналов

Просмотр

Общее хранилище RSS веб-каналов

Механизм синхронизации RSS

Поиск

for Developers

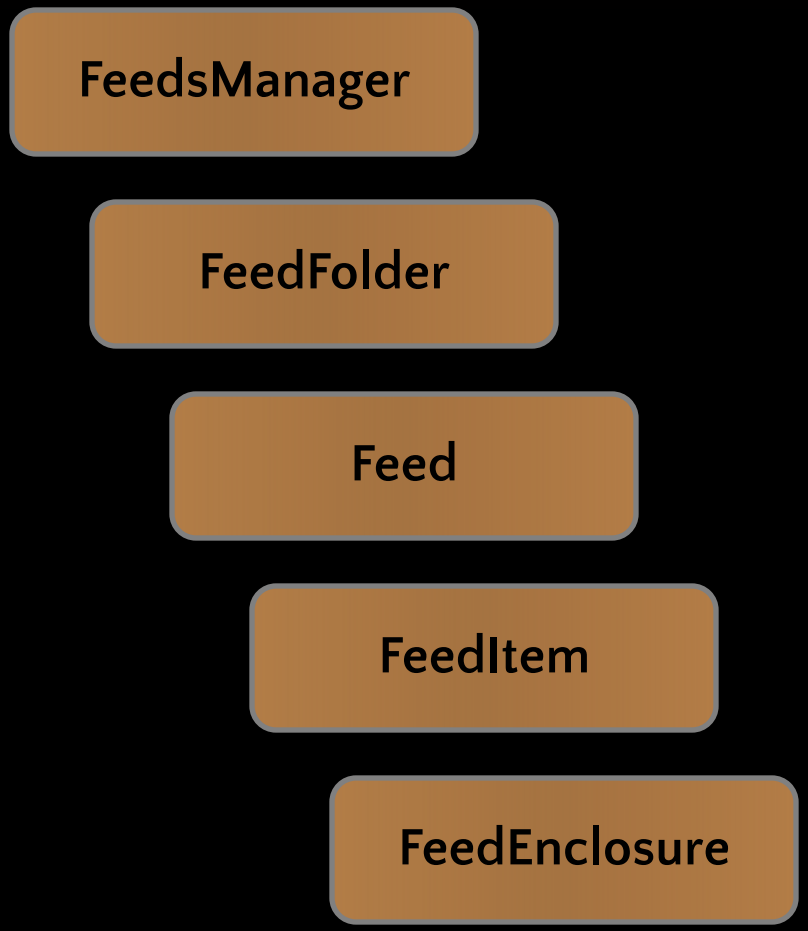
Office

Windows Vista

Exchange Server 2007

Платформа RSS веб-каналов

API веб-каналов



- COM API
- Два типа:
  - Automation-friendly (IFeed...)
  - Early-bound (IXFeed...)
- Возможно использование из управляемого кода
  - Поставляется библиотека типов

Работа с подписками

Демонстрация

Q&A

for Developers

Microsoft Office

Windows Vista

Microsoft Exchange Server 2007

© 2007 Microsoft Corporation. All rights reserved. Microsoft, Windows, Windows Vista and other product names are or may be registered trademarks and/or trademarks in the U.S. and/or other countries.
The information herein is for informational purposes only and represents the current view of Microsoft Corporation as of the date of this presentation. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information provided after the date of this presentation.
MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, AS TO THE INFORMATION IN THIS PRESENTATION.

for Developers

Office

Windows Vista

Microsoft Exchange Server 2007