

# Windows Communication Foundation

## Service-Orientierte Anwendungen mit .NET

**Henry Böhlert**

**.NET Developer Group Stuttgart**

**27.9.2006**

# Ziele

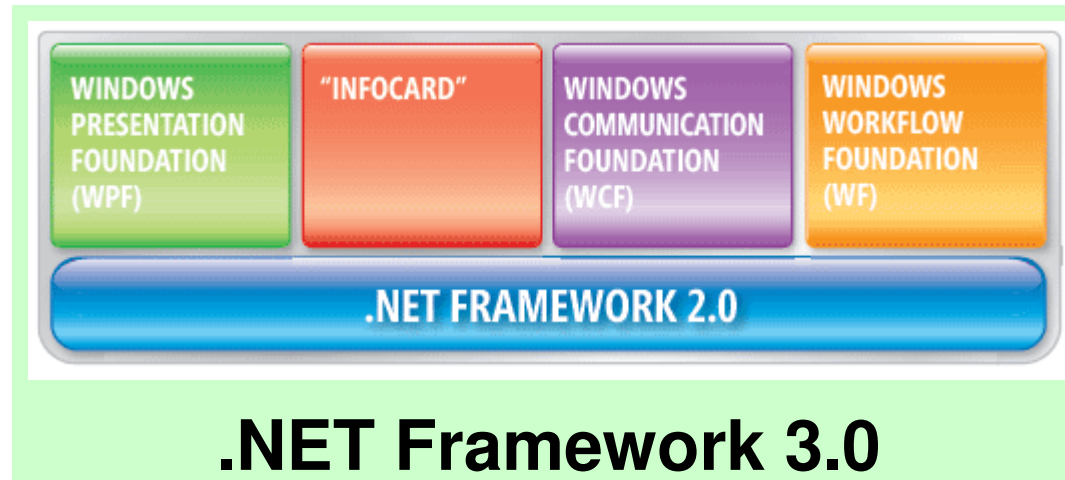
- **Hallo Welt!**
- **Konzept**
- **Konfigurieren**
- **Erweitern**
- **Testen und Fehler analysieren**

# Indigo

- <http://msdn.microsoft.com/Longhorn/understanding/pillars/Indigo/default.aspx>
- <http://channel9.msdn.com/>

# Windows Communication Foundation

- <http://msdn.microsoft.com/winfx/reference/communication/>
- <http://channel9.msdn.com/tags/WCF>



# Los geht's:

<http://msdn.microsoft.com/windowsvista/downloads/products/getthebeta/>

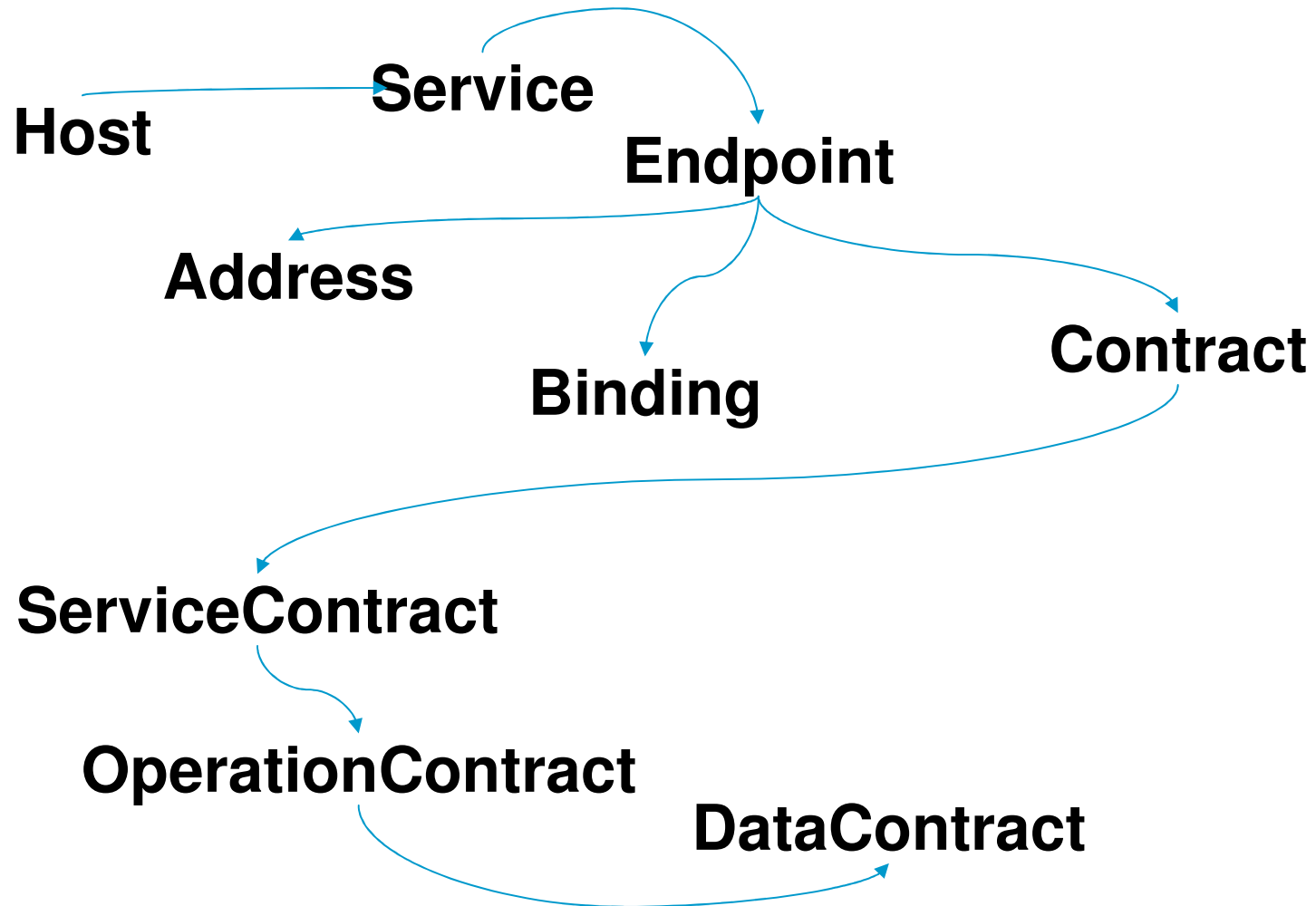
- .NET Framework 3.0 Runtime Components (Windows XP SP2, Server 2003)
- Windows SDK (Dokumentation, Beispiele und Hilfsprogramme)
- "Orcas" .NET Framework 3.0 Development Tools (Visual Studio 2005)
- [Visual Studio 2005 Extensions for Windows Workflow Foundation]

**Bitte nicht auf  
Produktionsmaschinen  
installieren!**

# Hallo Welt!

1. **Neues „WCF Service Library“-Projekt erstellen**
2. **Konsolenanwendung für den ServiceHost**
3. **MyServiceHost.cs und app.config erstellen**
4. **„System.ServiceModel“ und WCF Service Library einbinden**
5. **„StartService“ und „StopService“ in „Main“ aufrufen**

# Terminologie



# Konzept „Service-Orientierung“

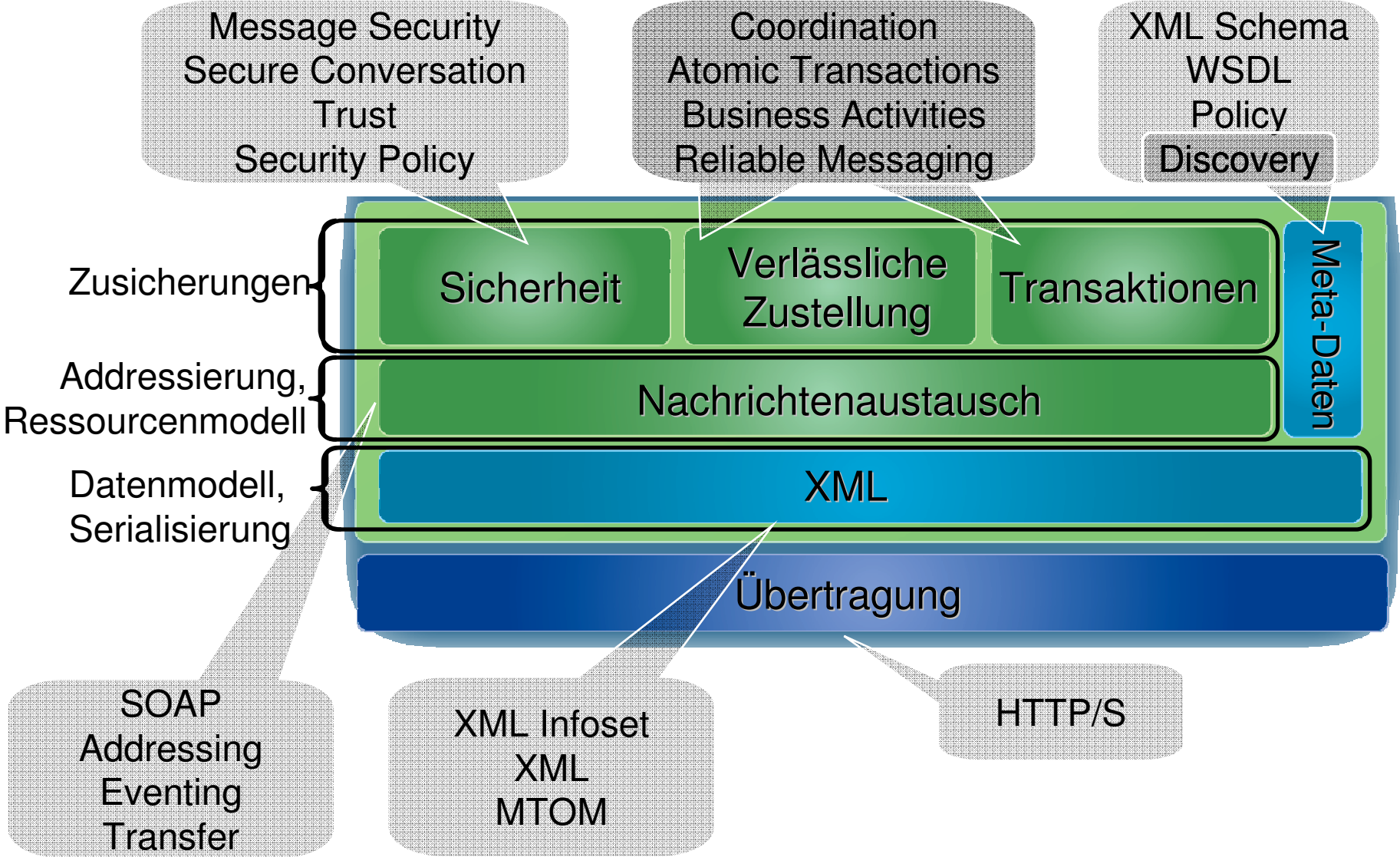
## Neue Stufe der Entkopplung in Verteilten Anwendungen

1. **Remote Procedure Call**
  - **Direkter, synchroner Aufruf einer bestimmten Funktion**
2. **Objekt-Orientierung, .NET Remoting, CORBA**
  - **Synchroner Aufruf einer (virtuellen) Methode eines Objekts, unabhängig von der tatsächlichen Methodenimplementierung**
3. **Komponentenarchitektur**
  - **Synchroner Aufruf von Methoden dynamisch geladener Objekte anhand von Schnittstellendefinitionen und Meta-Daten**
4. **Web Services**
  - **Synchrone Anfrage/Antwort-Kommunikation, unabhängig von Technologie und Implementierung**
5. **Service-Orientierung, WCF**
  - **Asynchroner oder synchroner Austausch von Nachrichten, anhand vertraglich vereinbarter Regeln, unabhängig von Technologie und innerer Architektur**

# Grundsätze der „Service-Orientierung“

- **Service-Grenzen sind explizit**
- **Services sind eigenständig**
- **Services kommunizieren anhand gemeinsamer Schemata und Vereinbarungen, nicht durch eine Klassenhierarchie**
- **Service-Kompatibilität basiert auf Richtlinienpezifikation**

# Spezifikationen für Web Services (WS-\*)



# Entscheidungshilfe Technologie

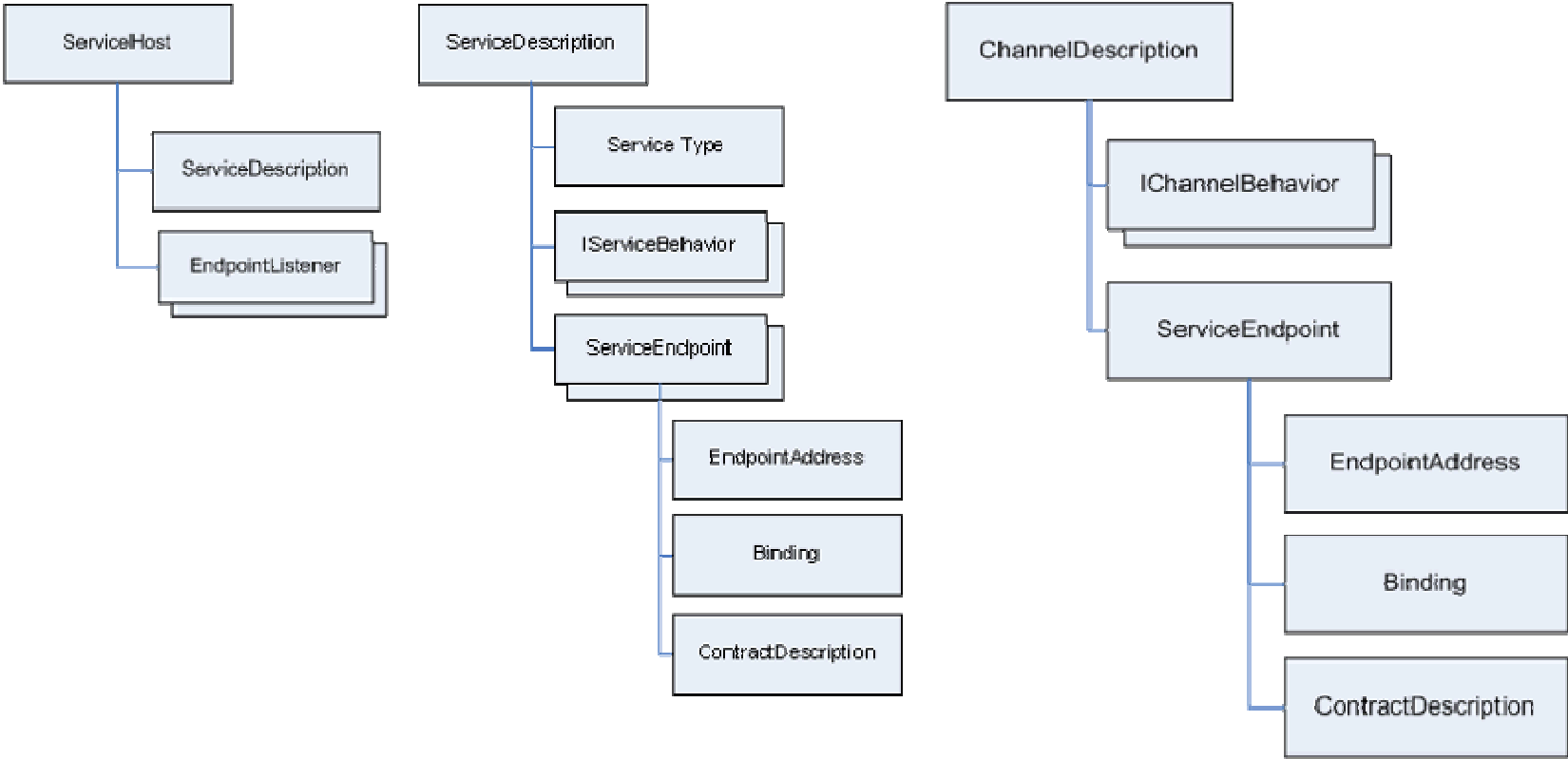
## Remoting

- **Beidseitig kontrollierte .NET-Objekt-Objekt-Kommunikation**
  - **Kann verschlüsselten TCP-Transport benutzen**
- **Lokal (zwischen AppDomains oder Prozessen auf einer Maschine)**
  - **Sollte IPC-Channel benutzen (Named Pipes)**

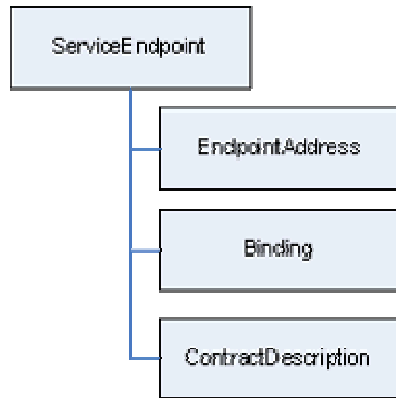
## Windows Communication Foundation

- **Web Services Standards (WS-\*)**
- **Interop (BEA, IBM, SAP, Oracle, ...)**
- **Services im Extranet oder Internet**

# Architektur



# Endpoint



## Adresse

- „Wo“
- **Binding-typisch**

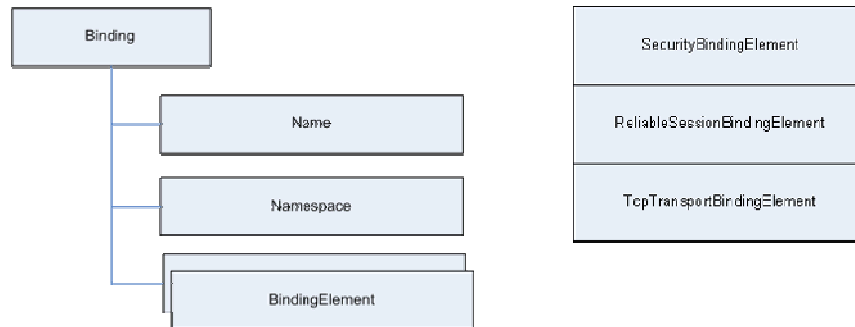
## Binding

- „Wie“
- **Konfiguration des Kommunikationspfades (Channel)**

## Contract

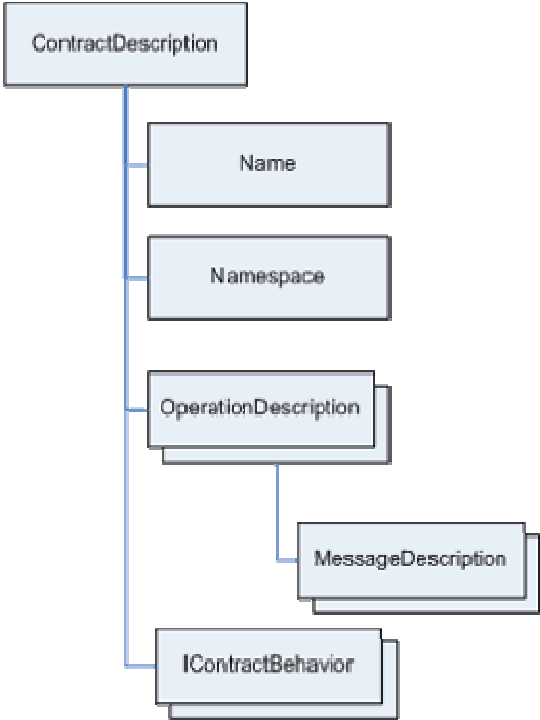
- „Was“
- **Operationen und Datenformate**

# Binding



| Binding                        | Interoperability  | Security (Default)          | Session (Default)                   | Transactions | Duplex |
|--------------------------------|-------------------|-----------------------------|-------------------------------------|--------------|--------|
| <b>BasicHttpBinding</b>        | Basic Profile 1.1 | (None), Transport, Message, | None, (None)                        | (None)       | n/a    |
| <b>WSHttpBinding</b>           | WS                | Transport, (Message), Mixed | (None), Transport, Reliable Session | (None), Yes  | n/a    |
| <b>WSDualHttpBinding</b>       | WS                | (Message)                   | (Reliable Session)                  | (None), Yes  | Yes    |
| <b>WSFederationHttpBinding</b> | WS-Federation     | (Message)                   | (None), Reliable Session            | (None), Yes  | No     |
| <b>NetTcpBinding</b>           | .NET              | (Transport), Message        | Reliable Session, (Transport)       | (None), Yes  | Yes    |
| <b>NetNamedPipeBinding</b>     | .NET              | (Transport)                 | None, (Transport)                   | (None), Yes  | Yes    |
| <b>NetMsmqBinding</b>          | .NET              | Message, (Transport), Both  | (None)                              | (None), Yes  | No     |
| <b>NetPeerTcpBinding</b>       | Peer              | (Transport)                 | (None)                              | (None)       | Yes    |
| <b>MsmqIntegrationBinding</b>  | MSMQ              | (Transport)                 | (None)                              | (None), Yes  | n/a    |

# Contract



**ServiceContract**

**OperationContract**

**MessageContract**

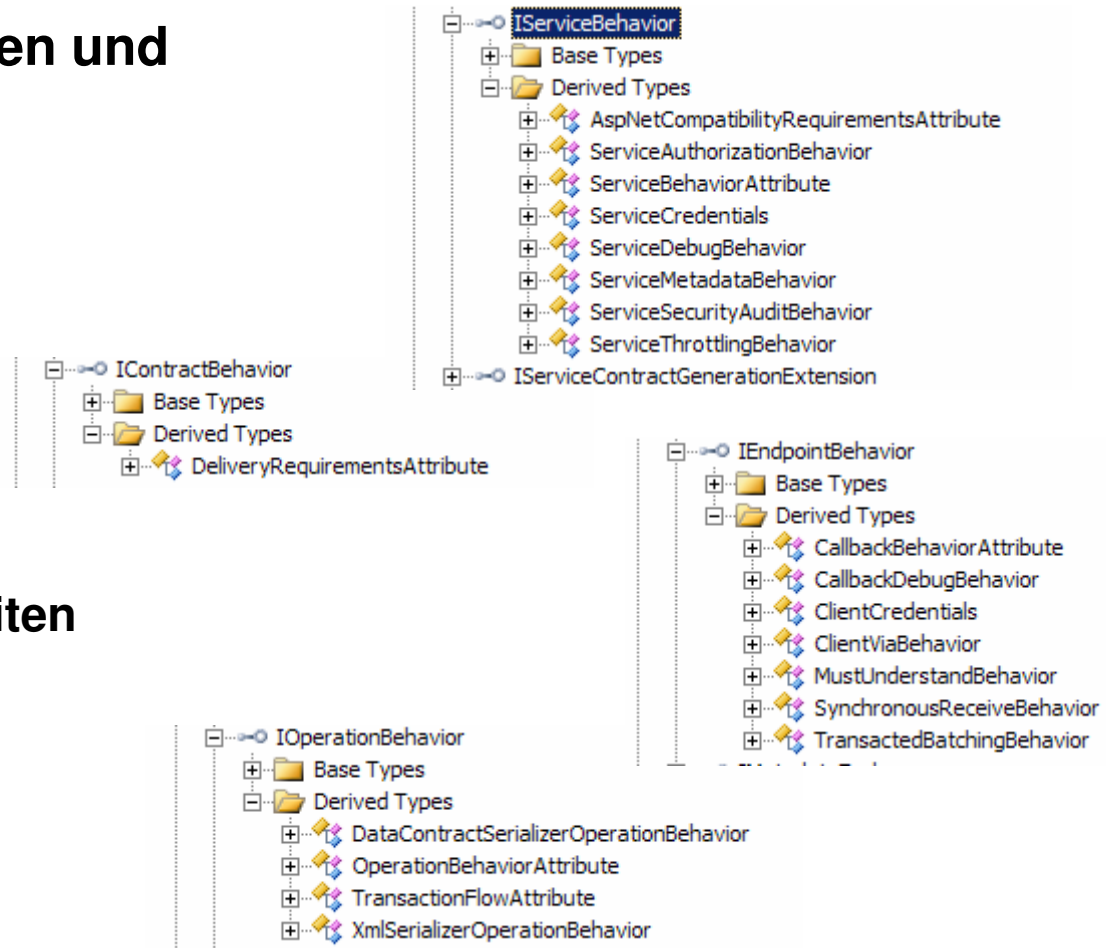
**DataContract**

**FaultContract**

# Behaviors

- **Zusätzliche Funktionen und Einschränkungen**

- **Service**
- **Endpoint**
- **Binding**
- **Contract**
- **Operation**
- **Eigene Behaviors ableiten**
  - **IServiceBehavior**
  - **IEndpointBehavior**
  - **IContractBehavior**
  - **IOperationBehavior**



# Hosting

- **Windows-Anwendung**
  - ServiceHost
  - Interaktion mit Benutzeroberfläche (eher für Testzwecke)
- **Windows-Dienst**
  - Sicherheit durch spezifischen Benutzeraccount
- **IIS**
  - .svc, Prozessverwaltung
  - 5.1/6.0: nur HTTP/S
- **Windows Activation Service (Windows Vista)**
  - IIS-ähnliches Hosting ohne IIS für alle Transportprotokolle (HTTP, TCP, Named Pipes, MSMQ)

# Konfiguration per XML

## •XML-Konfiguration

- Deklarative Konfiguration
- Service Configuration Editor im SDK (SvcConfigEditor.exe)
- Erlaubt weit gehende Änderungen ohne Code-Eingriff
- Intellisense in Visual Studio 2005
- Korrespondierender Namespace  
**System.ServiceModel.Configuration**

```
<system.serviceModel>  
  <behaviors>  
  </behaviors>  
  <commonBehaviors>  
  </commonBehaviors>  
  <client>  
    </client>  
  <diagnostics>  
  </diagnostics>  
  <serviceHostingEnvironment>  
  </serviceHostingEnvironment>  
  <comContracts>  
  </comContracts>  
  <services>  
  </services>  
  <bindings>  
  </bindings>  
</system.serviceModel>
```

# Konfiguration per Code

- **Inspektion von ServiceHost oder ChannelFactory**
  - Verwenden von Eigenschaften und Methoden
- **Ableitung von ServiceHost oder ChannelFactory**
  - Überschreiben von CreateDescription oder InitializeRuntime
- **Attribute**
  - Behaviors

# Diagnose und Fehleranalyse

- **Exceptions**
- **Trace**
  - **Service Trace Viewer im SDK (SvcTraceViewer.exe)**
- **Performance Counter**
  - **ServiceModelEndpoint**
  - **ServiceModelOperation**
  - **ServiceModelService**
- **WMI**
  - **root\ServiceModel namespace**

# Ressourcen

- **Community Website**
  - <http://wcf.netfx3.com>
- **Virtual Labs**
  - <http://msdn.microsoft.com/virtuallabs/wcf/default.aspx>
- **Windows SDK**
  - <http://windowssdk.msdn.microsoft.com>