5.– 8. September 2011
in Nürnberg

Herbstcampus

Wissenstransfer
par excellence

# Verbindlichkeiten

## Einführung in die Grundlagen benutzerdefinierter Bindings in WCF

Tobias Krügel
conplement AG

# Verbindlichkeiten

Einführung in benutzerdefinierte Bindings in WCF

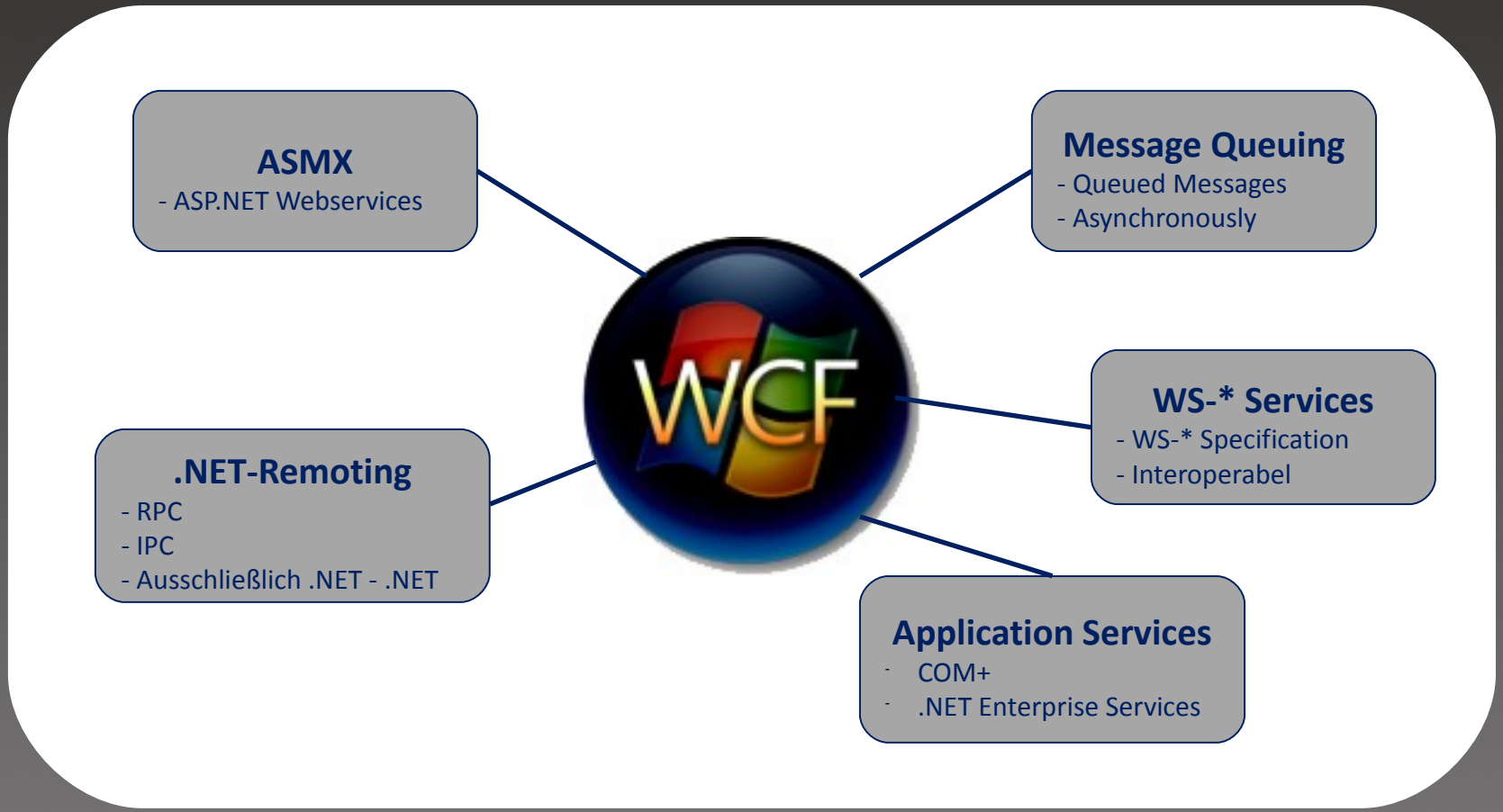
Tobias Krügel
conplement AG

# Agenda

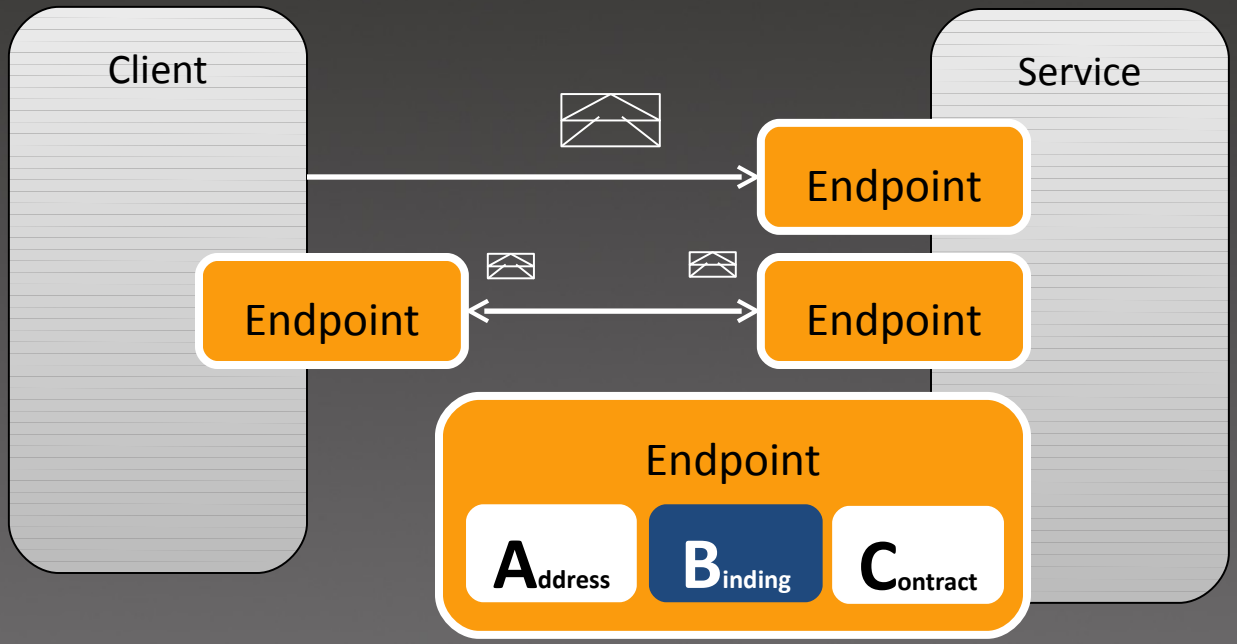
- Essentials
  - WCF
  - Endpoints
  - Bindings
- Channel Layer
  - Channels
  - Channel States
  - Channel Shapes
  - Channel Managers
- Service Model Layer
  - Bindings
  - BindingElements

# Essentials

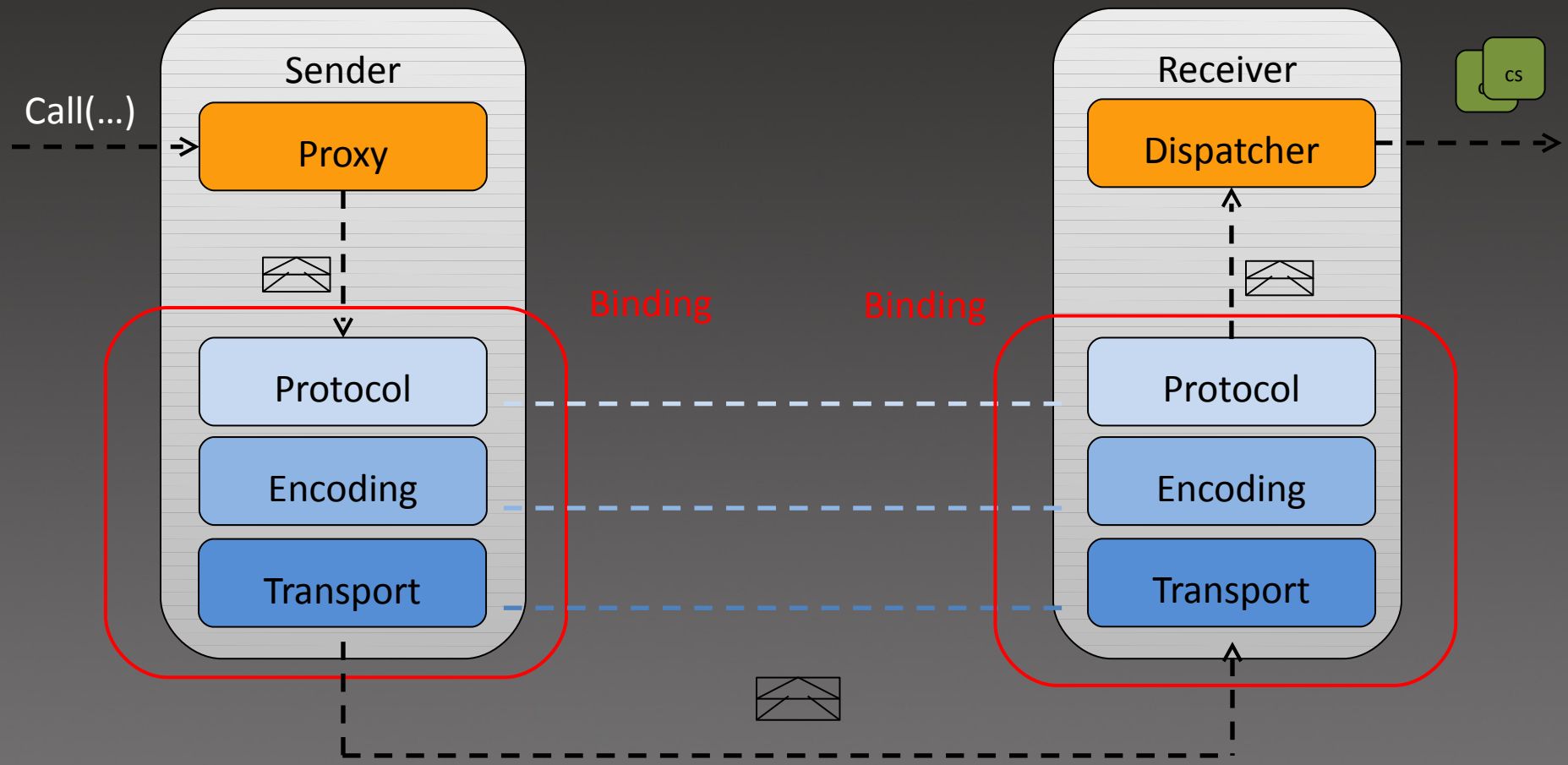
- Windows Communication Foundation



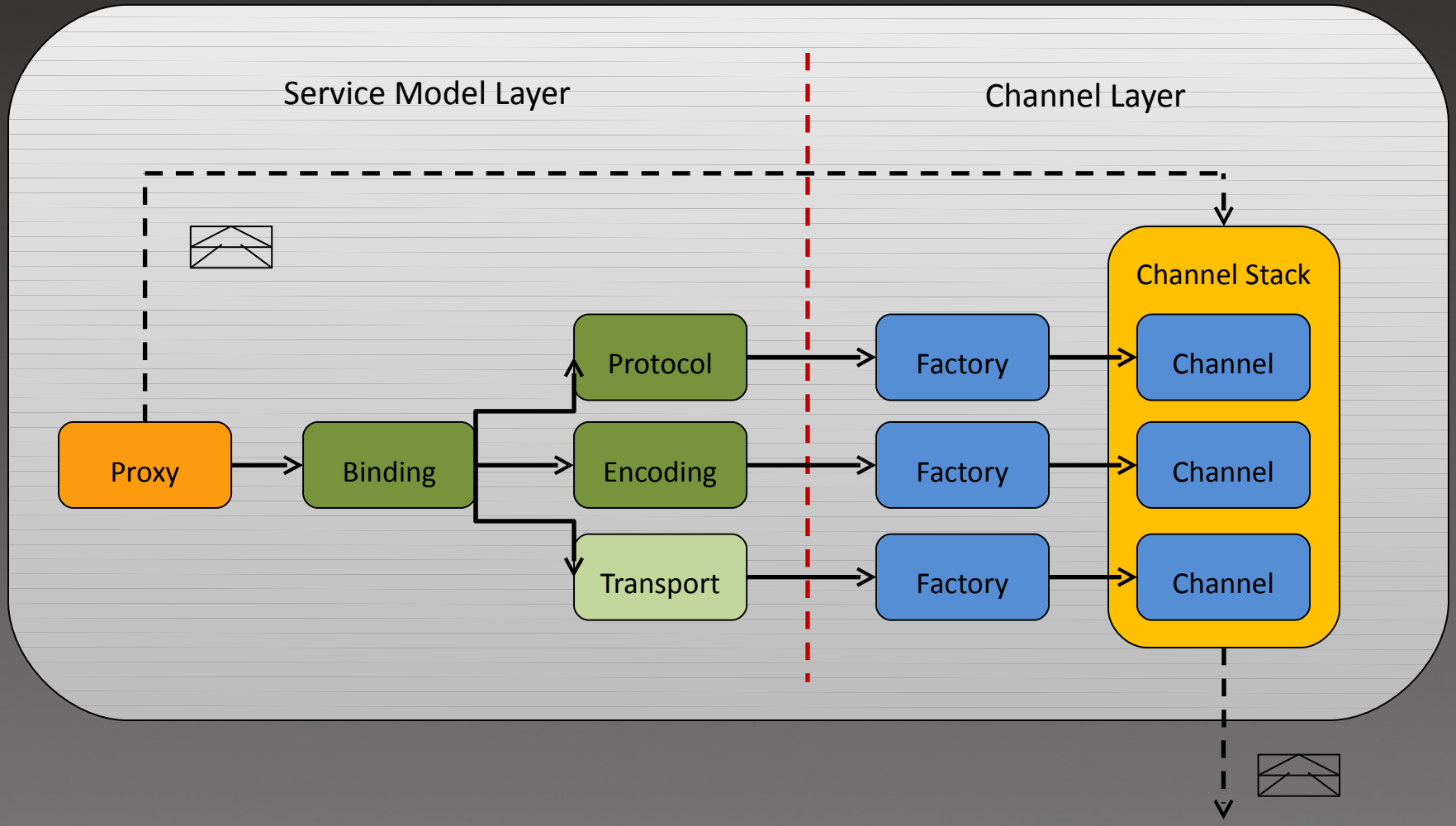
# Endpoints



## WCF Gross Anatomy

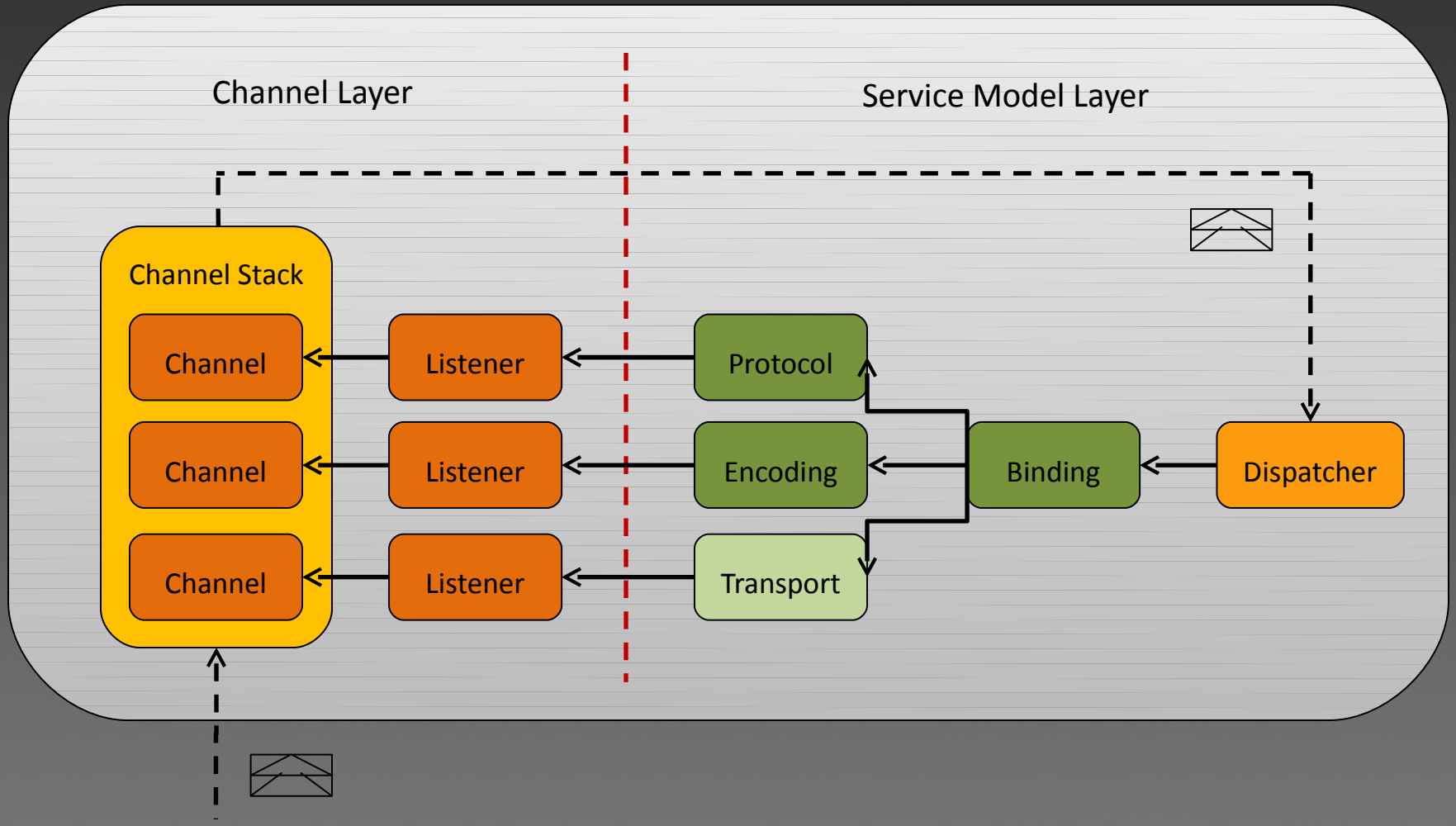


## Architecture (Sender)



# Essentials

## Architecture (Receiver)



## Bindings

- Common Bindings

| Type | Transport | Encoding | Interoperable |
|---|---|---|---|
| **BasicHttpBinding** | HTTP/HTTPS | Text/MTOM | beliebige Partner |
| **WSHttpBinding** | HTTP/HTTPS | Text/MTOM | beliebige WS*-Partner |
| **NetTcpBinding** | TCP | Binary | WCF to WCF |
| **NetNamedPipesBinding** | IPC | Binary | Selber Rechner |
| **NetMsmqBinding** | MSMQ | Binary | MS Message Queue Services |

## Bindings

### DURABLE SERVICE BINDINGS

- BasicHttpContextBinding
- NetTcpContextBinding
- WSHttpContextBinding

### WS BINDINGS

- WSDualHttpBinding
- WSFederationHttpBinding
- WS2007FederationHttpBinding
- WS2007HttpBinding

### SONSTIGE

- NetPeerTcpBinding
- MSMQIntegrationBinding

### SERVICE BUS BINDINGS

- BasicHttpRelayBinding
- NetEventRelayBinding
- NetOnewayRelayBinding
- NetTcpRelayBinding
- WebHttpRelayBinding
- WS2007HttpRelayBinding
- WSHttpRelayBinding

### MEX BINDINGS

- MexHttpBinding
- MexTcpBinding
- MexNamedPipeBinding

### CUSTOM BINDINGS

- CustomBinding

# Demo - 1

## Bindings

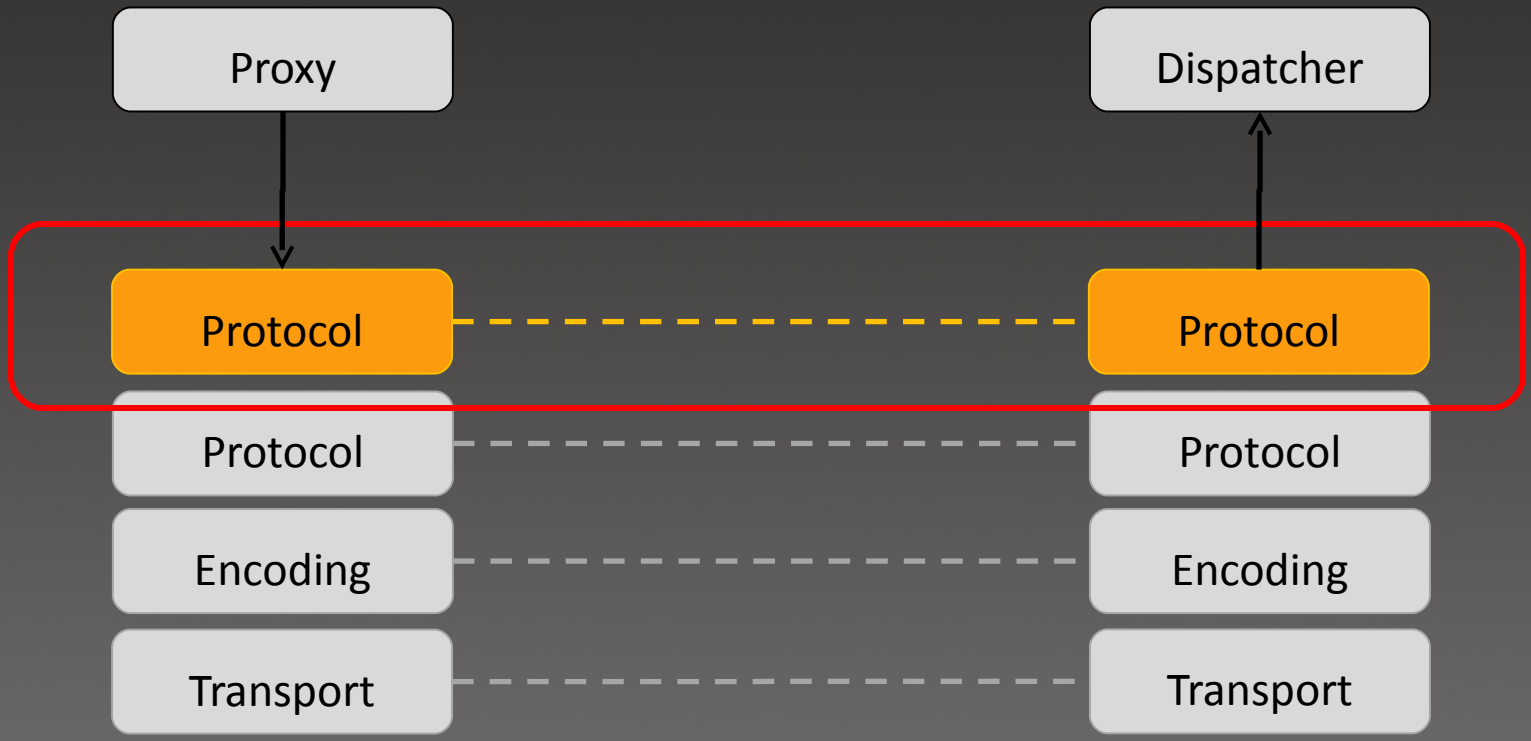
- Common Binding
- Adapted Binding
- Custom Binding

# Channel Layer

## Channels

# Channels

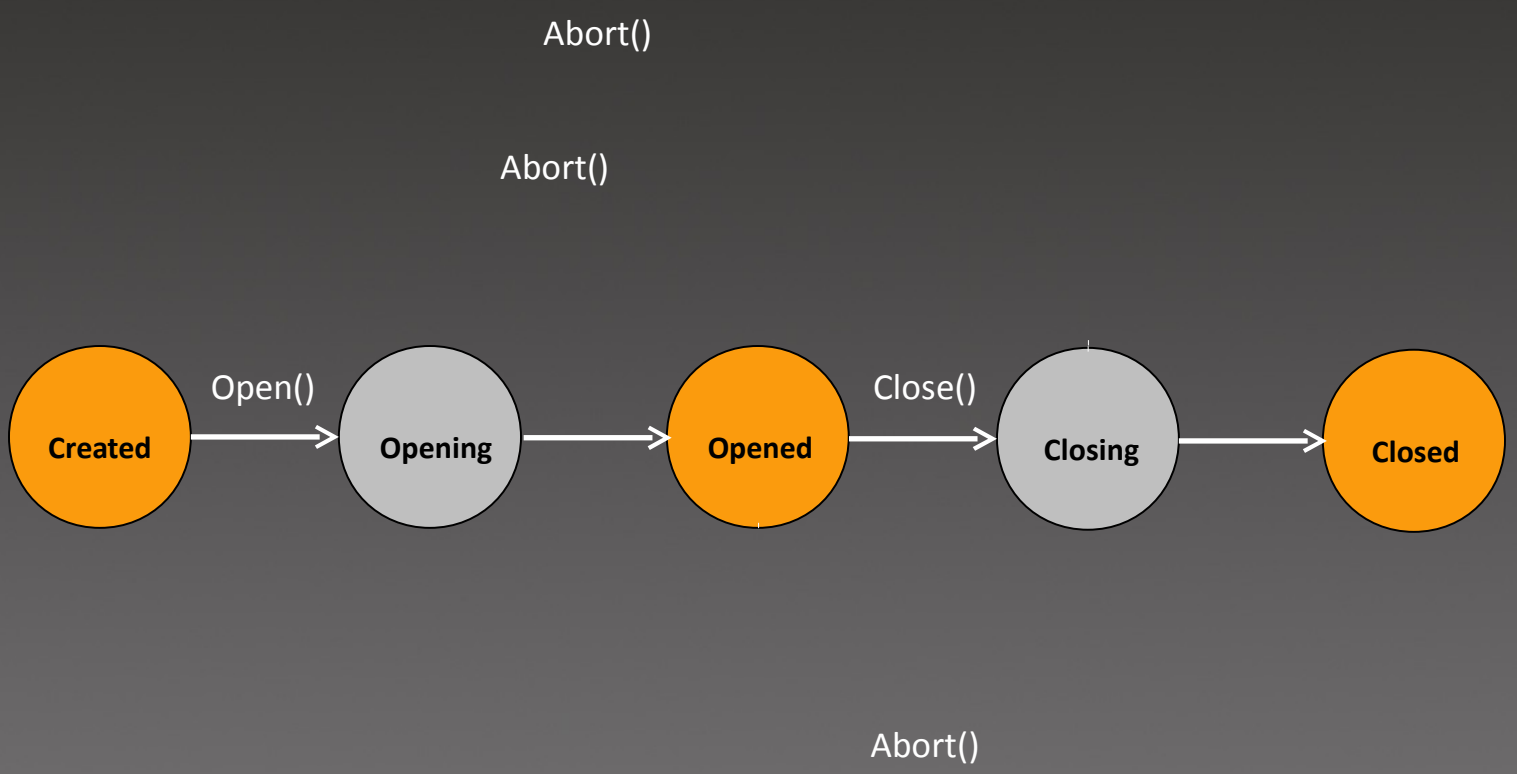


# Channel Layer

## State Machine

# Channel Layer

## States



# Channel Layer

## States

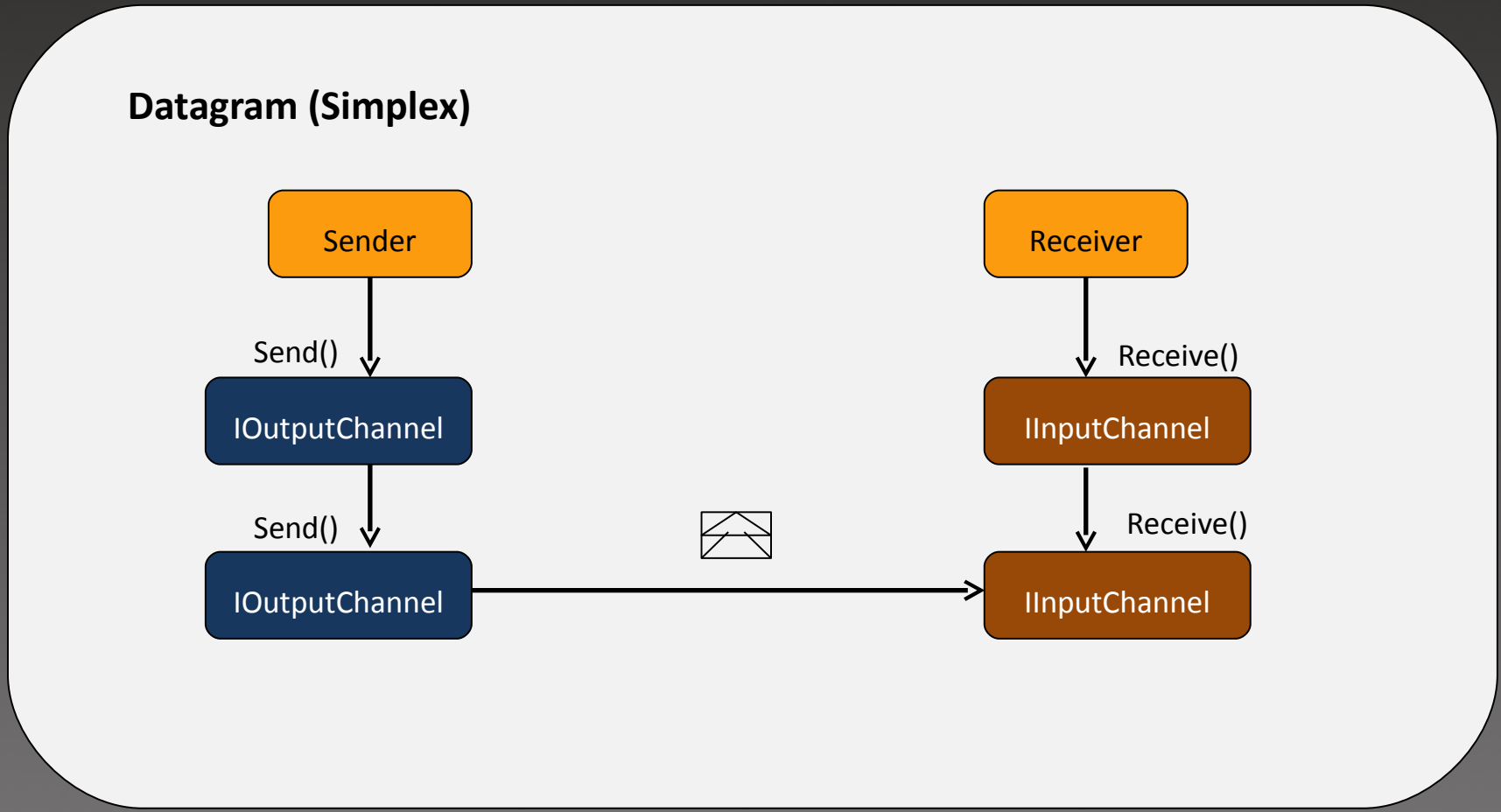
```
public interface ICommunicationObject
{
    CommunicationState State { get; }

    void Open();
    void Close();
    void Abort();

    event EventHandler Opening;
    event EventHandler Opened;
    event EventHandler Closing;
    event EventHandler Closed;
    event EventHandler Faulted;

    IAsyncResult BeginOpen(AsyncCallback callback, object state);
    void EndOpen(IAsyncResult result);

    IAsyncResult BeginClose(AsyncCallback callback, object state);
    void EndClose(IAsyncResult result);
}
```

- CommunicationObject

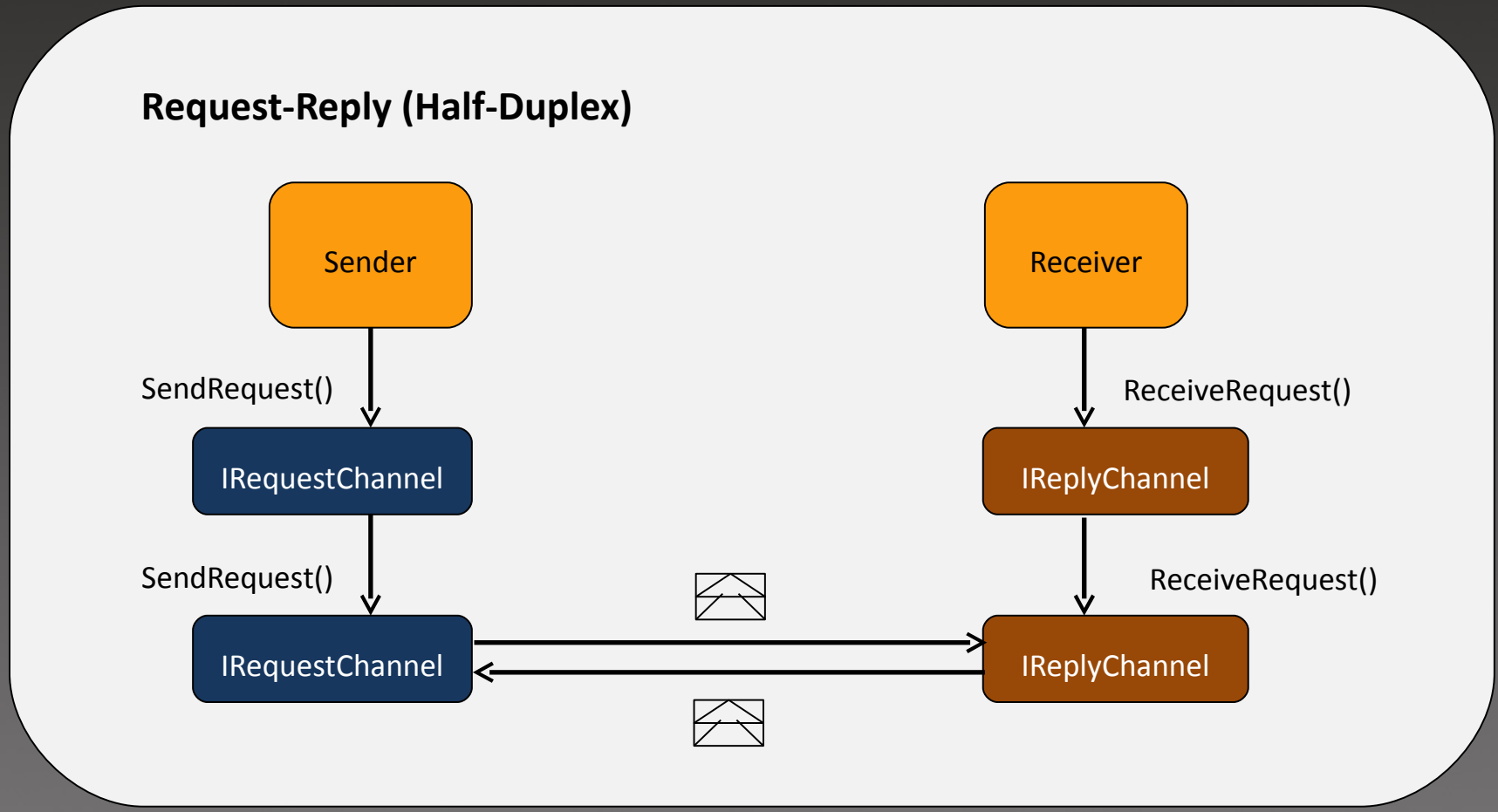
# Channel Layer

## Channel Shapes

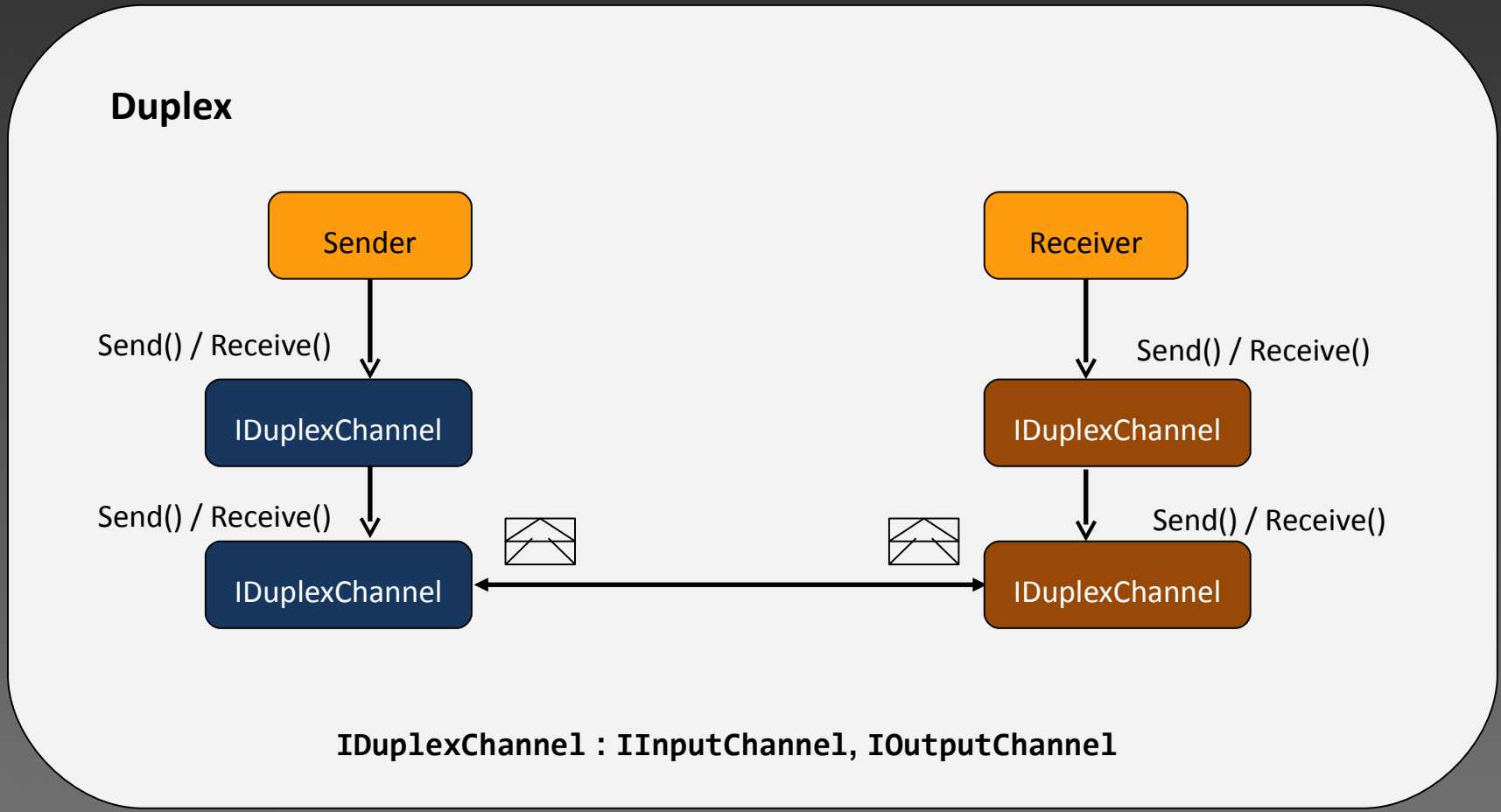
# Message Exchange Patterns (1/4)



# Message Exchange Patterns (2/4)



# Message Exchange Patterns (3/4)



# Message Exchange Patterns (4/4)

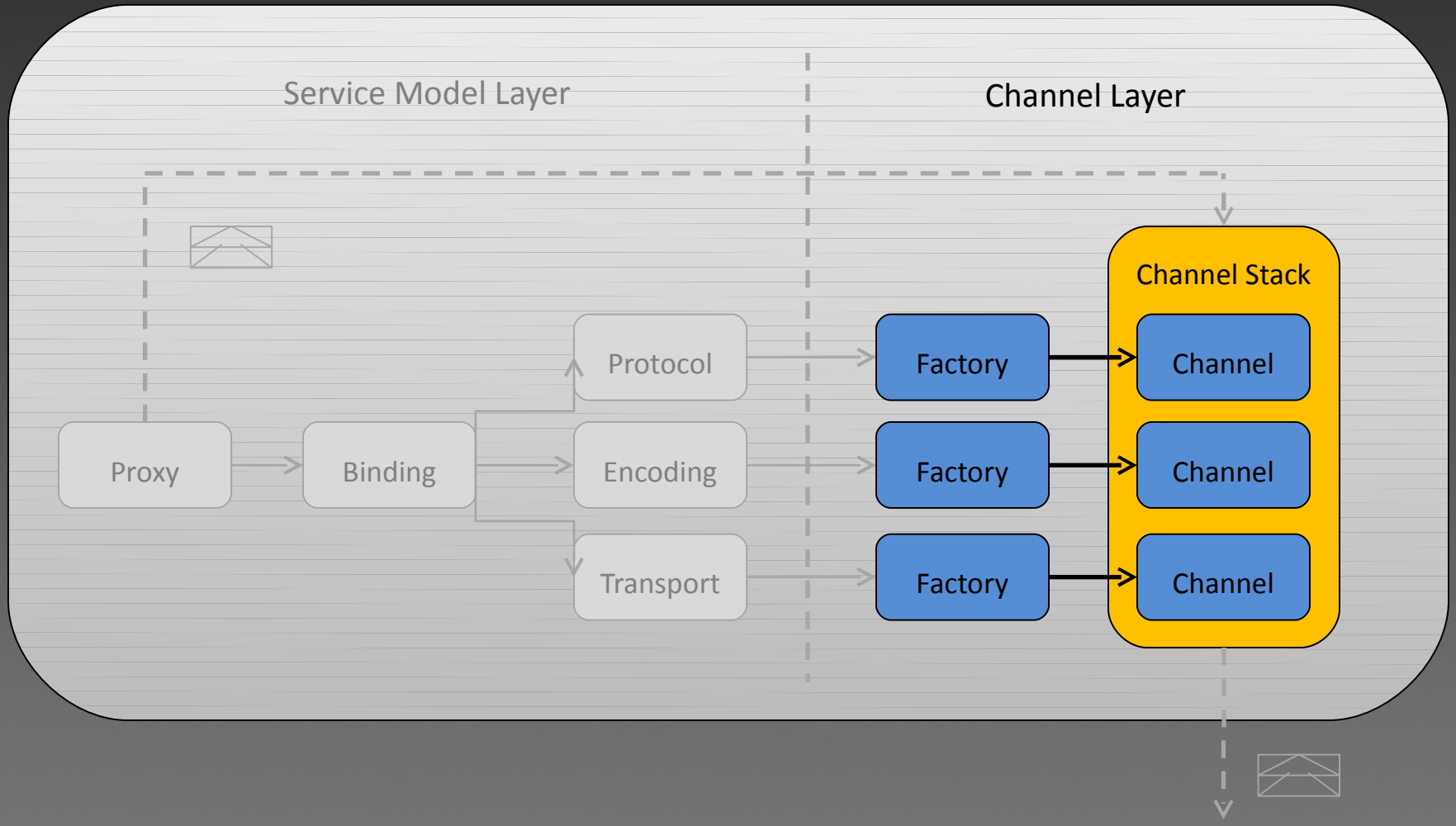
- ISessionChannel

```
public interface ISessionChannel<TSession> where TSession : ISession
{
    TSession Session { get; }
}
```

- IInputSessionChannel
- IOutputSessionChannel
- IRequestSessionChannel
- IReplySessionChannel
- IDuplexSessionChannel

# Demo - 2

## Channel Layer

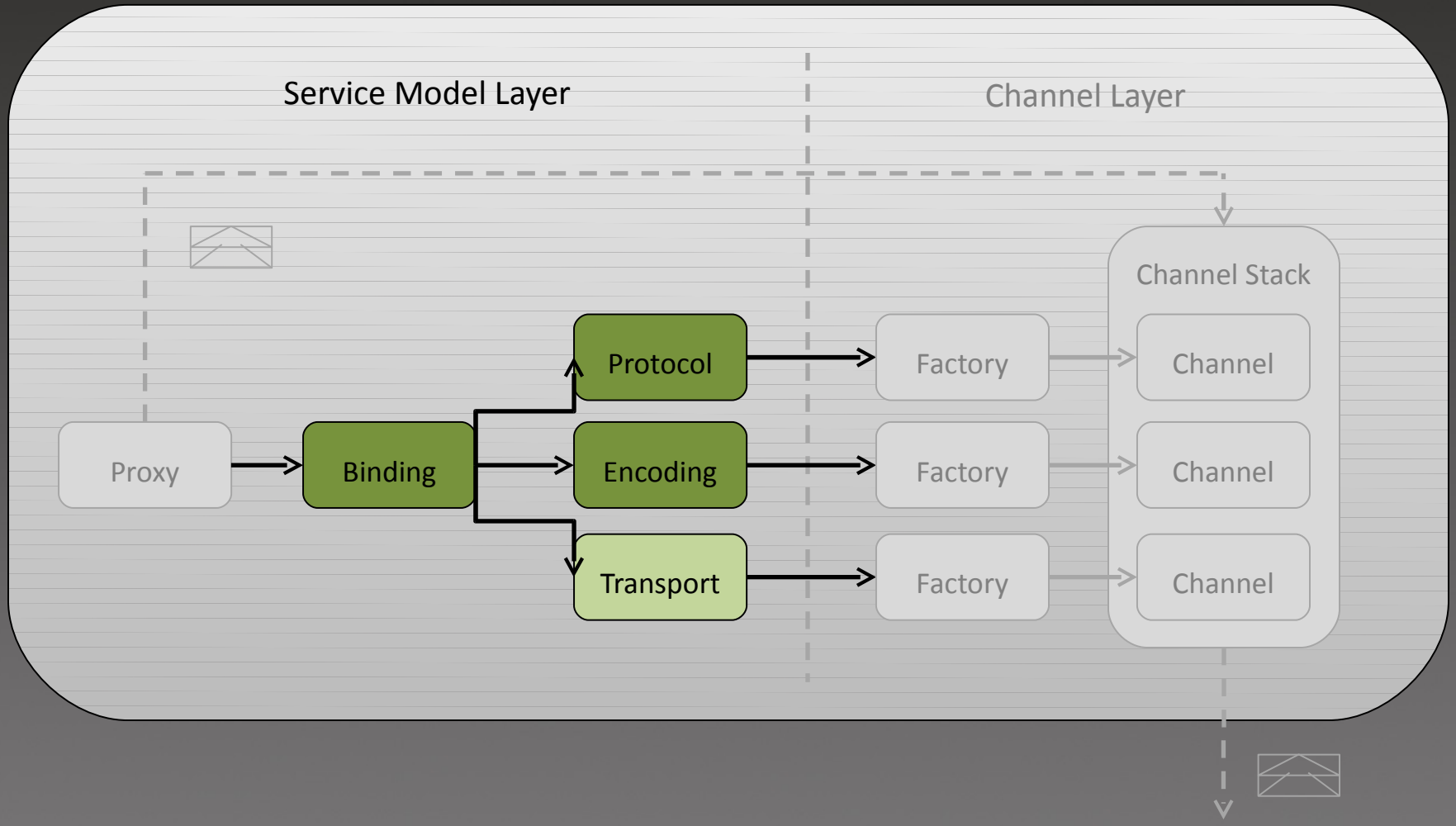


# Demo - 2

- Channels
- Channel States
- Channel Shaping
- Channel Manager

# Demo - 3

## Service Model Layer



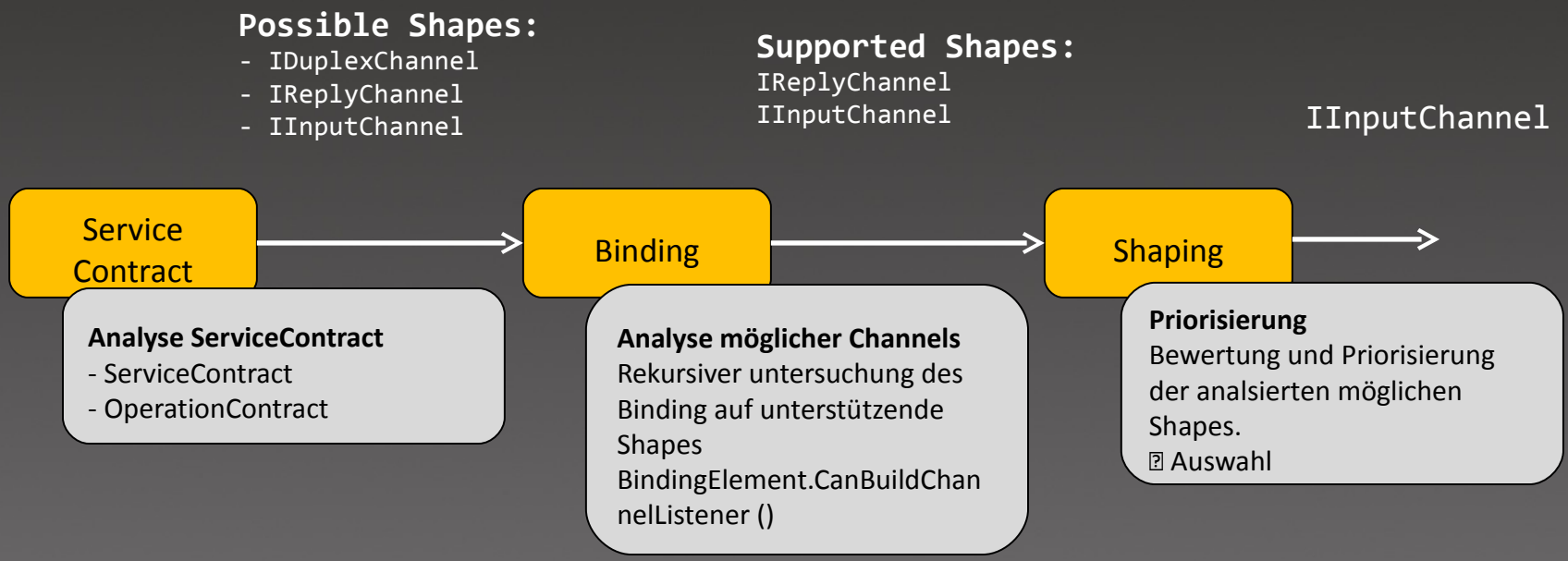
# Demo - 3

- Binding Elements
- Binding

# Shaping Channels

# Shaping Channels



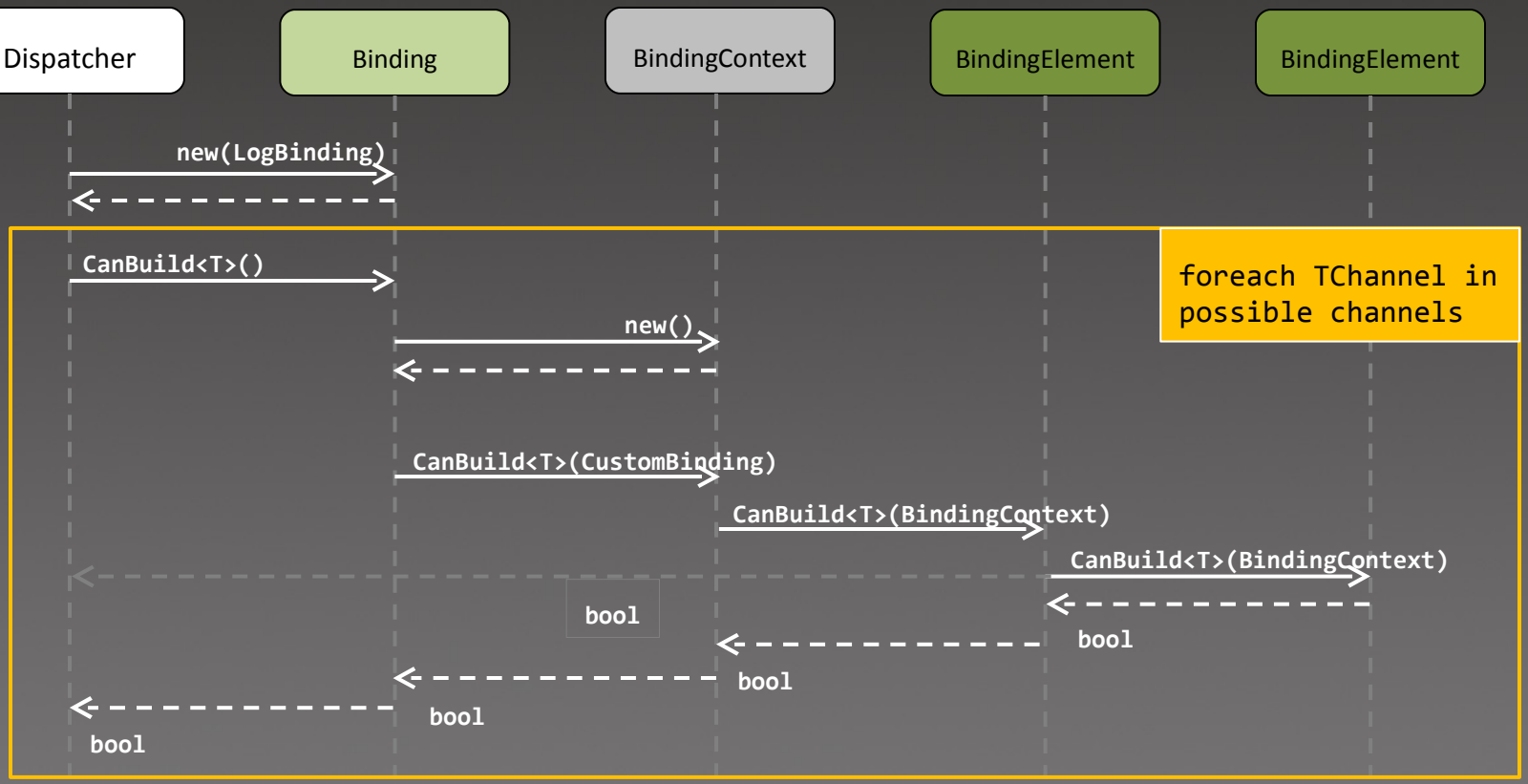
# Shaping Channels

- ServiceContract Analysis

```
[ServiceContract(
     SessionMode=SessionMode.Required,
CallbackContract=typeof(ICalculatorCallback)
)]
public interface ICalculator
{
    [OperationContract(IsOneWay = true)]
     void Write(string message);
}
```

# Shaping Channels

## ChannelStack Analysis (Receiver)



# Shaping Channels

- **Priorisierung**

| Priorität | Sender | Receiver |
|---|---|---|
| 1 | IOutputChannel | IInputChannel |
| | IOutputSessionChannel | IInputSessionChannel |
| 2 | IRequestChannel | IReplyChannel |
| | IRequestSessionChannel | IReplySessionChannel |
| 3 | IDuplexChannel | IDuplexChannel |
| | IDuplexSessionChannel | IDuplexSessionChannel |

# Demo - 4

- WCF Channel Shaping

# Shaping Channels

- **TcpTransportBindingElement**

```
public override bool CanBuildChannelFactory<TChannel>(BindingContext context)
{
    if (context == null)
    {
        throw DiagnosticUtility.ExceptionUtility.ThrowHelperArgumentNull("context");
    }
    if (this.TransferMode == TransferMode.Buffered)
    {
        return (typeof(TChannel) == typeof(IDuplexSessionChannel));
    }
    return (typeof(TChannel) == typeof(IRequestChannel));
}
```

# Summary

- Essentials
  - WCF
  - Endpoints
  - Bindings
- Channel Layer
  - Channels
  - Channel States
  - Channel Shapes
  - Channel Managers
- Service Model Layer
  - Bindings
  - BindingElements

# Ihre Fragen?


- Tobias Krügel
- Senior Software Engineer
- tobias.kruegel@conplement.de

## Bindings

### Default Binding Characteristics

| Type | Encoder | Security | Session | Transactions | Duplex | Streaming |
|---|---|---|---|---|---|---|
| **BasicHttpBinding** | Text<br>MTOM | Transport | | | | X |
| **WSHttpBinding** | Text<br>MTOM | Message | X | X | X | X |
| **NetTcpBinding** | Binary | Transport<br>Message | X | X | X | X |
| **NetNamedPipesBinding** | Binary | Transport<br>Message | X | X | X | X |
| **NetMsmqBinding** | Binary | Transport<br>Message | X | X | | |

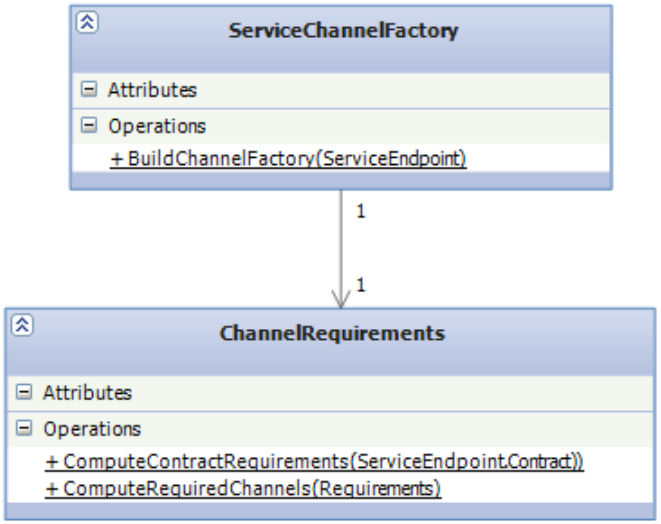
# Appendix

## Channel Shapes (MEP)

- Ermittlung der Channel Shapes durch WCF

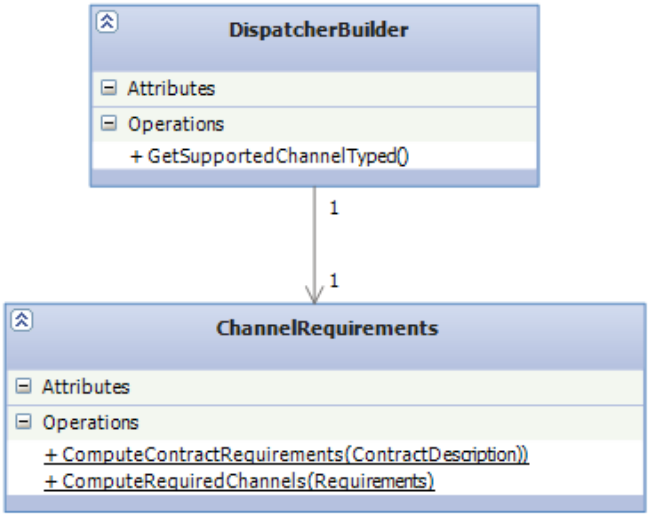
### Sender

System.ServiceModel.Channels




### Receiver

System.ServiceModel.Channels



# Appendix

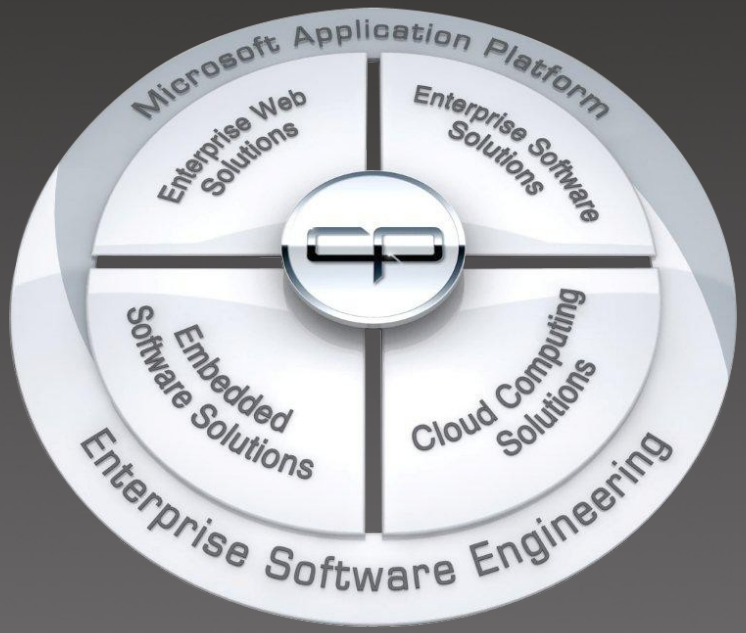
## Literatur

- **Inside Windows Communication Foundation , Microsoft Press**
  Justin Smith

- **Professional WCF 4, Wrox**
  Pablo Cibraro
  Kurt Claeys
  Fabio Cozzolino
  Johann Grabner

- **Programming WCF Services, O'Reilly**
  Juval Löwy

# conplement AG – getting ideas done!




## ENTERPRISE SOFTWARE ENGINEERING

Kundenspezifische Beratung und Optimierung von Software-Prozessen und -Plattformen

## ENTERPRISE WEB SOLUTIONS

Beratung zu webbasierten Systemen und SharePoint Lösungen im Enterprise-Umfeld

## ENTERPRISE SOFTWARE SOLUTIONS

Beratung von Enterprise-Kunden im Themenbereich Windows und Rich Client Applikationen

## EMBEDDED SOFTWARE SOLUTIONS

Design der Hard- und Software von Embedded Lösungen sowie deren Integration in Kundensysteme