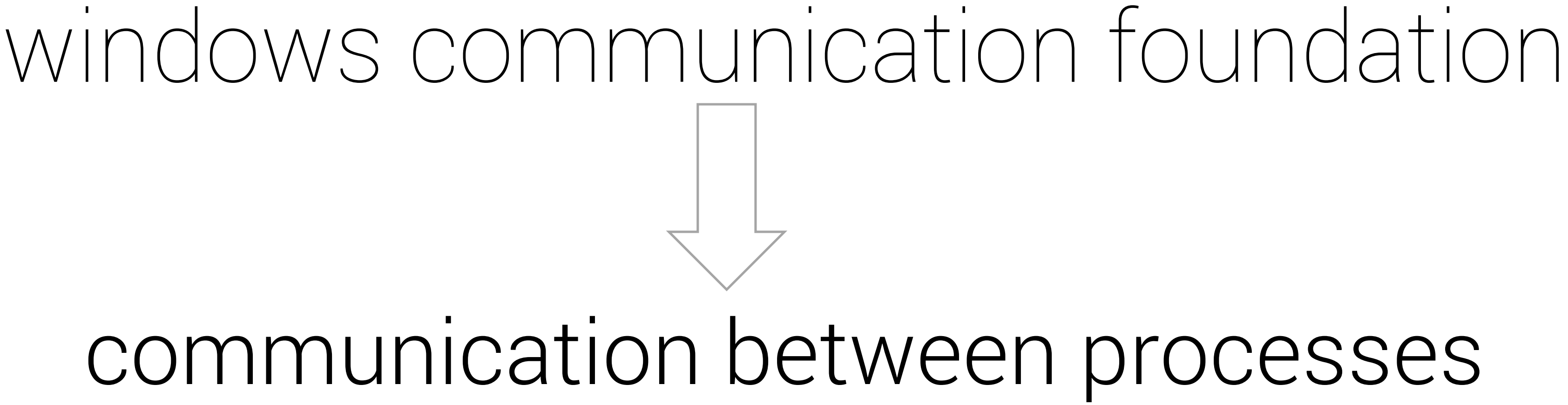
how to make computers talk?

@tomkuj

rule FINANCIAL

Part of the GFT Group

nowadays



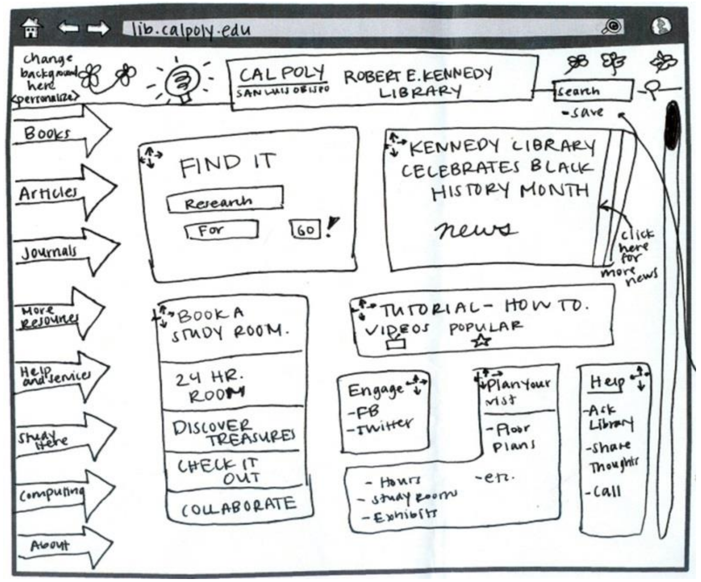
windows communication foundation

communication between processes

Address
Binding
Conctract

Address

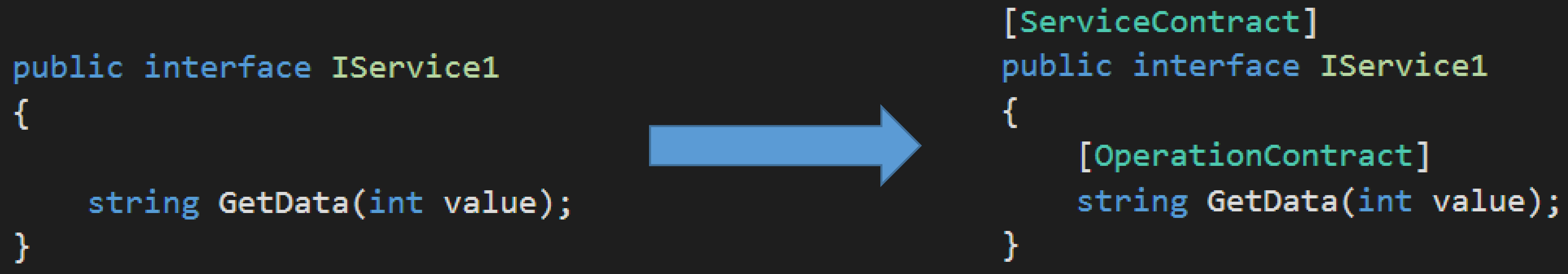
http://localhost:51134/Service.svc



Binding

Conctract

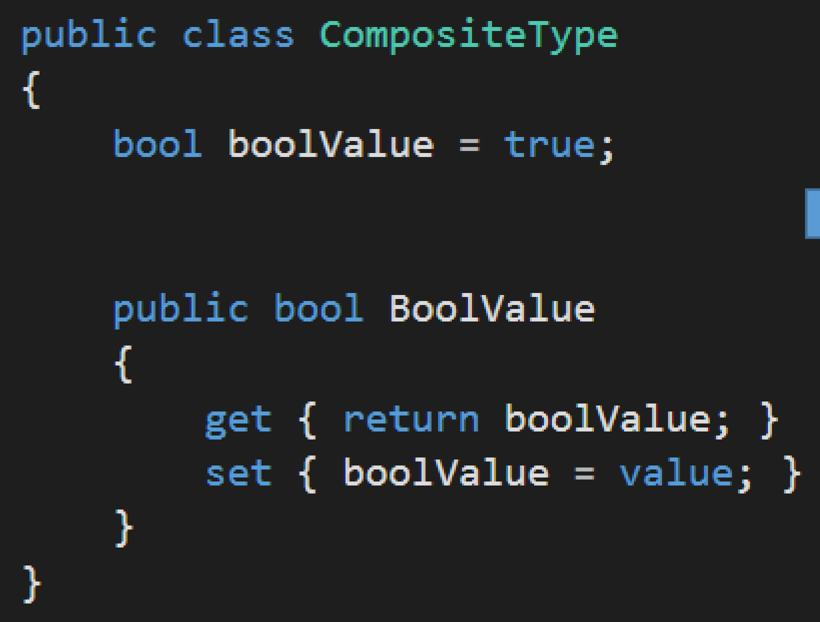
Service Contract



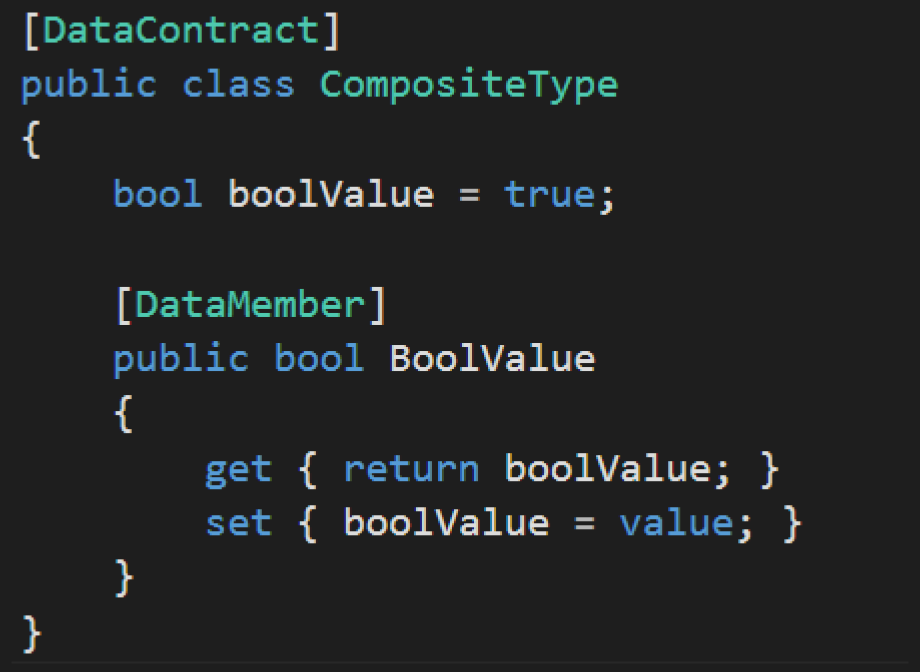
```
public interface IService1
{

    string GetData(int value);
}
```

```
[ServiceContract]
public interface IService1
{
    [OperationContract]
    string GetData(int value);
}
```

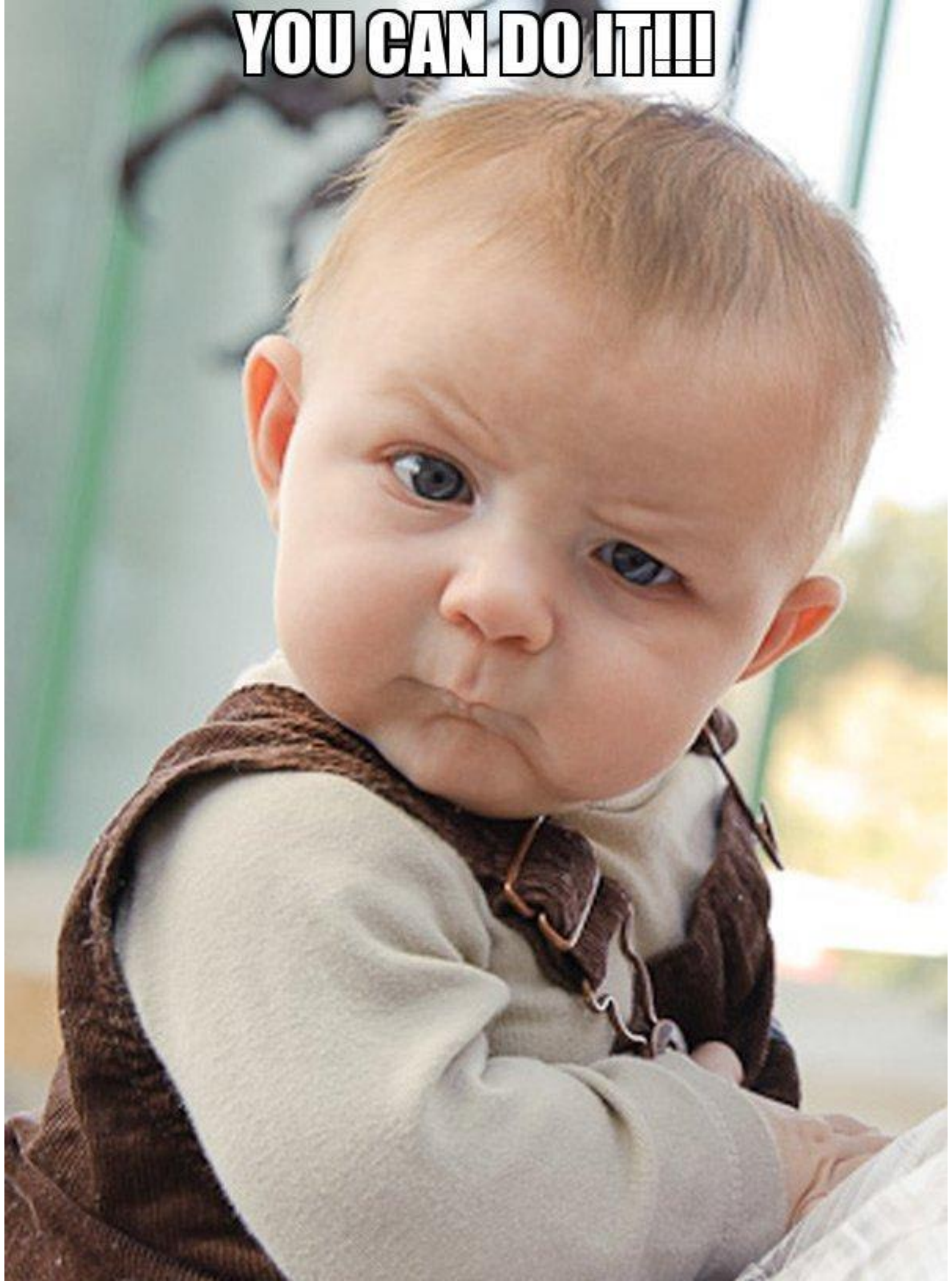
Data Contract



```
public class CompositeType
{
    bool boolValue = true;


    public bool BoolValue
    {
        get { return boolValue; }
        set { boolValue = value; }
    }
}
```



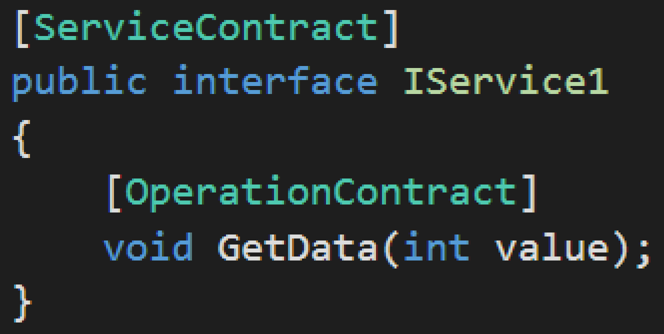
```
[DataContract]
public class CompositeType
{
    bool boolValue = true;

    [DataMember]
    public bool BoolValue
    {
        get { return boolValue; }
        set { boolValue = value; }
    }
}
```


YOU CAN DO IT!!!

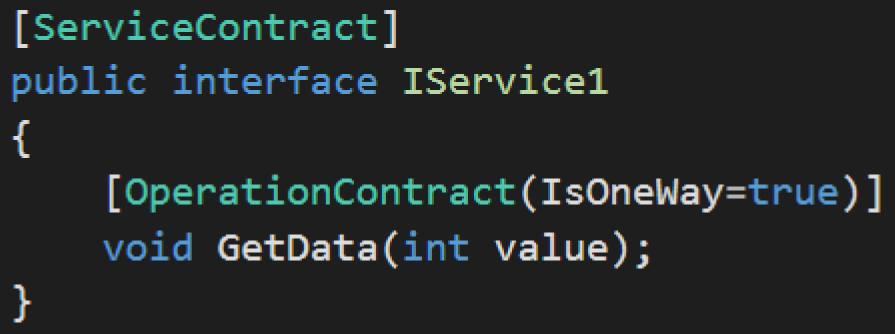
message patterns

one way
request-response
duplex (AKA asynchronous)

one way

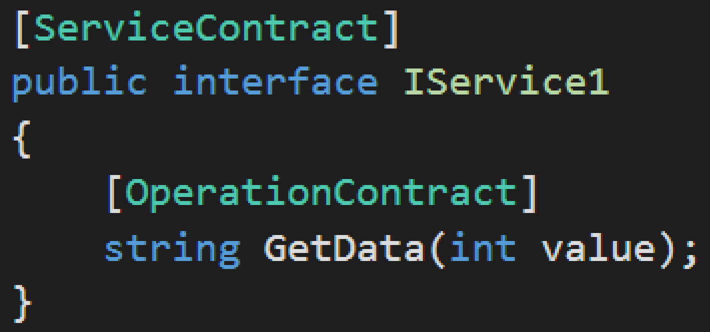


```
[ServiceContract]
public interface IService1
{
    [OperationContract]
    void GetData(int value);
}
```



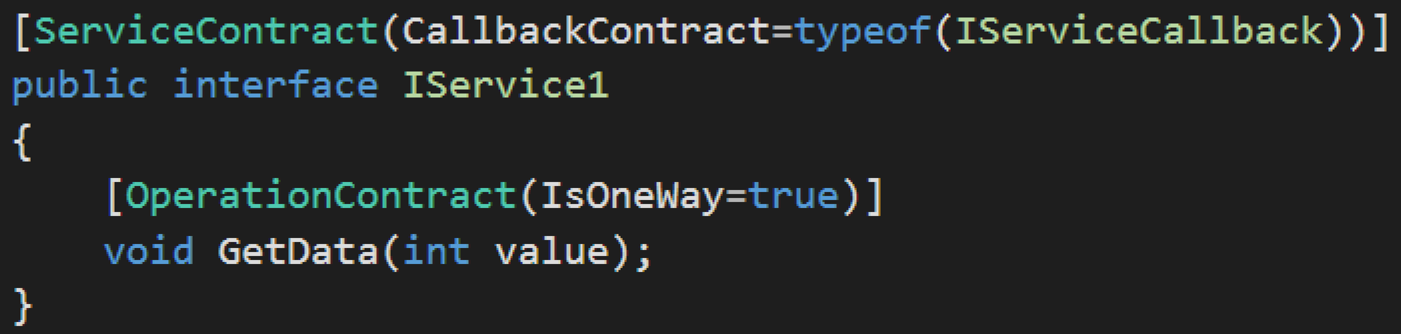
```
[ServiceContract]
public interface IService1
{
    [OperationContract(IsOneWay=true)]
    void GetData(int value);
}
```

request-response



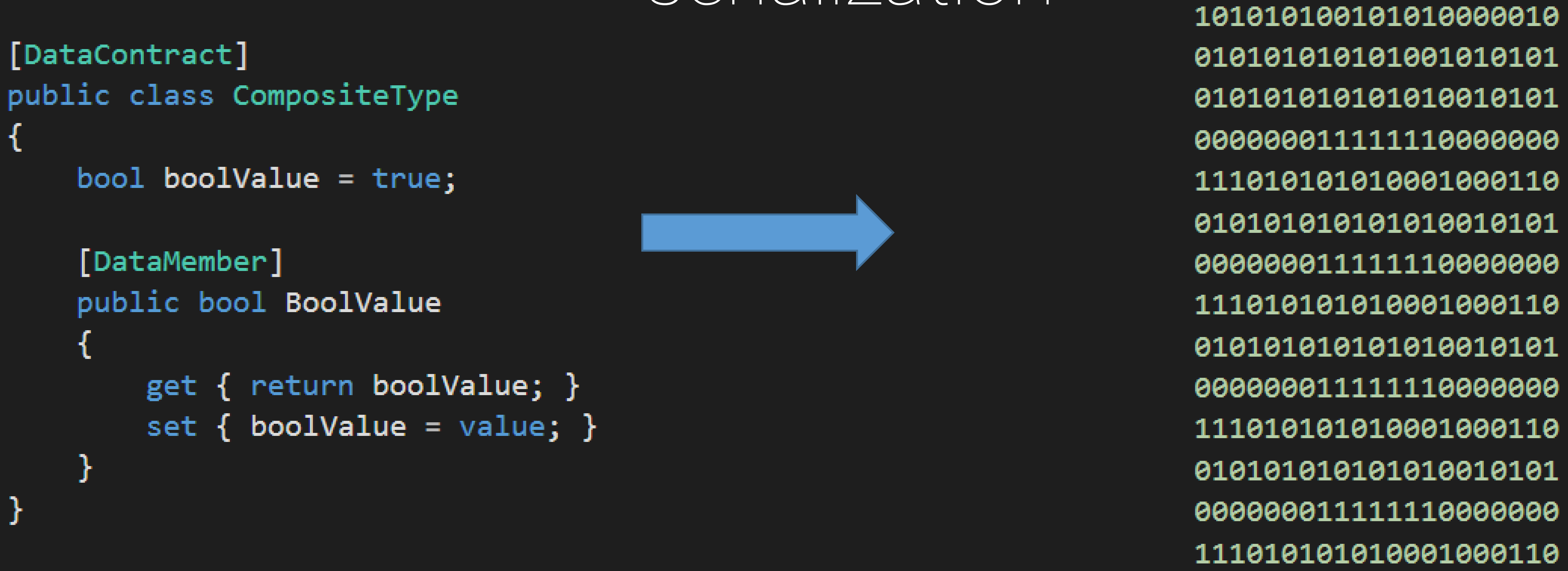
```
[ServiceContract]
public interface IService1
{
    [OperationContract]
    string GetData(int value);
}
```

duplex channel



```
[ServiceContract(CallbackContract=typeof(IServiceCallback))]
public interface IService1
{
    [OperationContract(IsOneWay=true)]
    void GetData(int value);
}
```

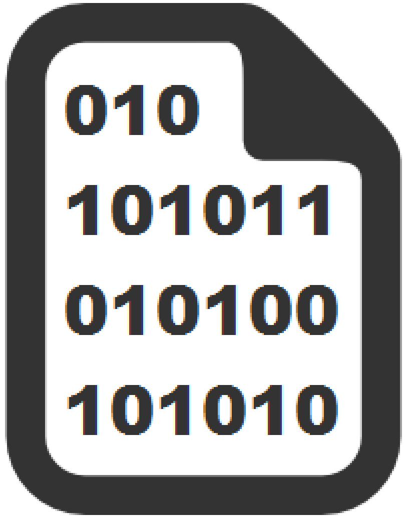
serialization

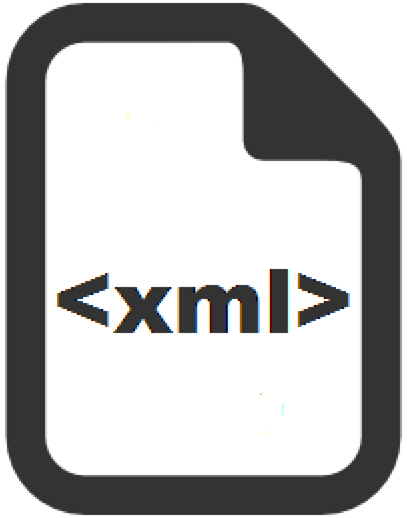


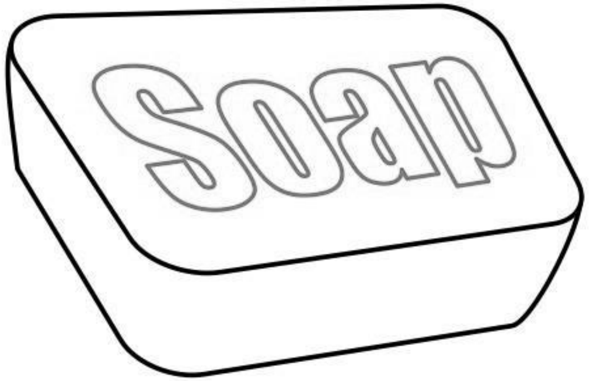
```
[DataContract]
public class CompositeType
{
    bool boolValue = true;

    [DataMember]
    public bool BoolValue
    {
        get { return boolValue; }
        set { boolValue = value; }
    }
}
```

```
101010100101010000010
010101010101001010101
010101010101010010101
000000011111110000000
111010101010001000110
010101010101010010101
000000011111110000000
111010101010001000110
010101010101010010101
000000011111110000000
111010101010001000110
010101010101010010101
000000011111110000000
111010101010001000110
```

serialization types







unified development model

SOAP serialization – interplatform objects exchange

👍

performance (e.g. real-time games)
concurrency – special care required
critical sections – easy to overlook
dead locks

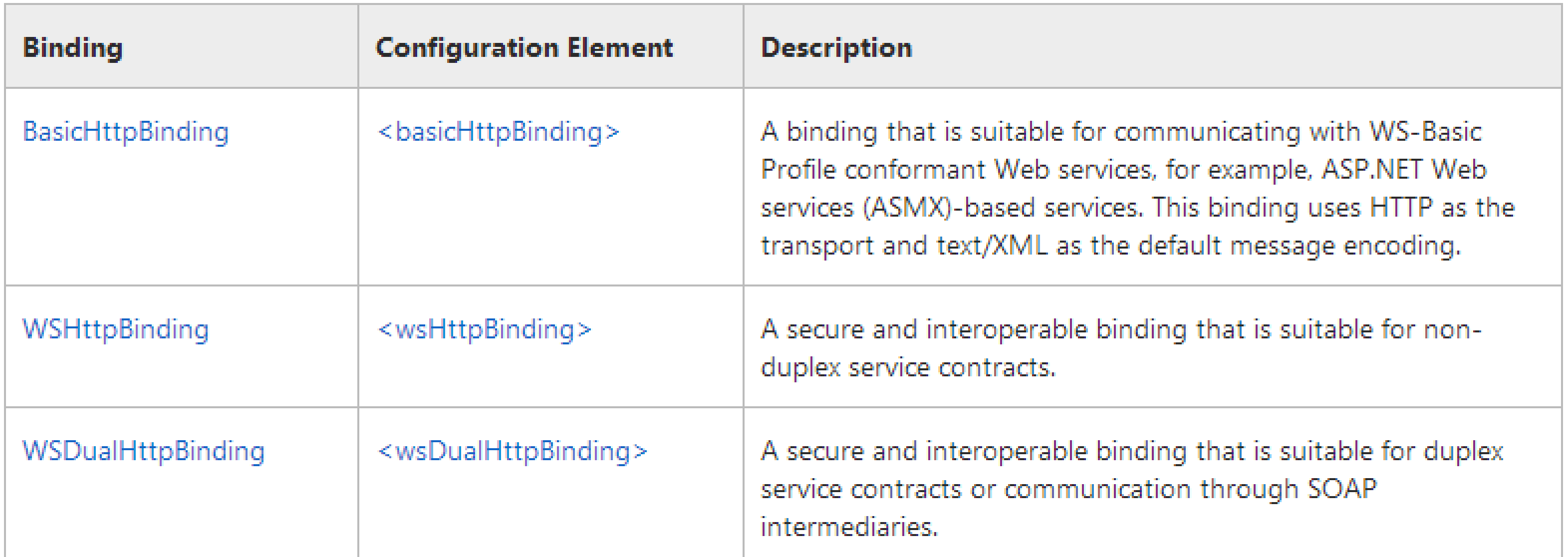
exercise

1. Launch Visual Studio
2. File / New / New Project / WCF Service Application
3. Run it
    1. Investigate WSDL file
    2. How the information is transformed?
    3. How is this data used? (*brain storm*)
4. Add method to the contract SayHello(string inputName)
    1. Method should return text „Hello {inputName}”.
5. Execute service and connect with WCF Test Client
6. Check if service works correctly

q&a

system-provided bindings

http://tinyurl.com/pwncws2



Binding	Configuration Element	Description
BasicHttpBinding	<basicHttpBinding>	A binding that is suitable for communicating with WS-Basic Profile conformant Web services, for example, ASP.NET Web services (ASMX)-based services. This binding uses HTTP as the transport and text/XML as the default message encoding.
WSHttpBinding	<wsHttpBinding>	A secure and interoperable binding that is suitable for non-duplex service contracts.
WSDualHttpBinding	<wsDualHttpBinding>	A secure and interoperable binding that is suitable for duplex service contracts or communication through SOAP intermediaries.