

# WINDOWS COMMUNICATION FOUNDATION (WCF)

Veerle Ongenaë

## WCF - overzicht

2

- Wat is WCF
- Architectuur WCF
  - Contracten, Gedrag, Berichten, Hosting
- Ontwikkeling WCF
  - Ontwerp servicecontract
  - Implementatie contract
  - Configuratie
  - Hosting
  - Ontwikkelen client

Industrieel Ingenieur Informatica, UGent

## WCF

3

- Gedistribueerde applicaties
  - Communicatie tussen applicaties
    - Op dezelfde computer
    - Over een intranet
    - Over het internet
- Service-oriented applications (~SOA)
- Webservices
  - SOAP
  - REST
- Transport over
  - Named pipes
  - TCP
  - HTTP
  - MSMQ
- Berichten
  - Binair
  - Tekst

Industrieel Ingenieur Informatica, UGent

## WCF Concepten

4

- Client-Services
- Communicatie
  - Berichten
  - Tussen "endpoints"
- Service
  - Bestaat uit verschillende "endpoints"
- Endpoint
  - Waarnaar berichten sturen
  - Hoe berichten sturen
  - Structuur berichten

Industrieel Ingenieur Informatica, UGent

## WCF - overzicht

5

- Wat is WCF
- Architectuur WCF
  - Contracten, Gedrag, Berichten, Hosting
- Ontwikkeling WCF
  - Ontwerp servicecontract
  - Implementatie contract
  - Configuratie
  - Hosting
  - Ontwikkelen client

Industrieel Ingenieur Informatica, UGent

## WCF-architectuur

6



<https://msdn.microsoft.com/en-us/library/ms733128%28v=vs.110%29.aspx>

Industrieel Ingenieur Informatica, UGent

## WCF Architectuur - Contracten

7

- Data Contract
  - Parameters in bericht
  - XSD
- Message Contract
  - Structuur bericht
- Service Contract
  - Beschikbare interface
  - Signatuur methodes
- Bindings, Policies
  - Hoe communiceren met service?
  - Beveiliging?

Industrieel Ingenieur Informatica, UGent

## WCF Architectuur – gedrag

8

- Service runtime
- Eigenlijke gedrag applicatie
  - Aantal berichten dat verwerkt kan worden
  - Wat bij fouten?
  - Welke metadata beschikbaar voor de buitenwereld?
  - Hoeveel instanties tegelijk van service?
  - Transacties
  - ...

Industrieel Ingenieur Informatica, UGent

## WCF Architectuur - Berichten

9

- Verschillende kanalen verwerken berichten
  - Zowel hoofdingen als berichten
    - Bv. Kanaal voor authenticatie
- Twee type kanalen
  - Transport
    - Lezen en schrijven van berichten van en naar het netwerk
  - Protocol
    - Lezen en schrijven van hoofdingen
    - ...

Industrieel Ingenieur Informatica, UGent

## WCF Architectuur - Hosting

10

- Een service is een programma
- Moet uitgevoerd worden
  - Als executable
    - Self-hosted service
  - Gehost in een externe omgeving
    - Webserver
    - Als Windows service

Industrieel Ingenieur Informatica, UGent

## Voorbeeld

11

- Service op IIS
  - tijdService
  - tijdClient
  - tijdWebserviceClient
- Service als console-applicatie
  - tijdServer
  - toonTijdVanConsole (Client)

Industrieel Ingenieur Informatica, UGent

## Overzicht

12

- Ontwerp servicecontract
- Implementatie contract
- Configuratie
- Hosting
- Ontwikkelen client

Industrieel Ingenieur Informatica, UGent

## Ontwerp servicecontract

13

- Service – dienst
  - Bestaat uit operations – opdrachten
    - Methode
      - AttribootOperationContract
      - Parameters en teruggeefwaarde
        - Serialiseerbaar
        - Geen referenties, maar kopies
    - Eigen datatypes
      - Contract
        - AttribootDataContract
        - AttribootDataMember
  - Interface
    - AttribootServiceContract
- Namespace: System.ServiceModel;

Industrieel Ingenieur Informatica, UGent

## Servicecontract

14

- Interface met attributen

```
[ServiceContract(Namespace = "http://Microsoft.ServiceModel.Samples")]
public interface ICalculator {
    [OperationContract]
    double Add(double n1, double n2);
    [OperationContract]
    double Subtract(double n1, double n2);
    [OperationContract]
    double Multiply(double n1, double n2);
    [OperationContract]
    double Divide(double n1, double n2);
}
```

Industrieel Ingenieur Informatica, UGent

## Type communicatie

15

- Vraag/antwoord
  - Standaard
- Eénrichtingscommunicatie (one-way contract)
  - Client → server
    - Client verwacht geen antwoord
  - Methode zonder returnwaarde
  - Geen uitvoerparameters
- Tweerichtingscommunicatie (duplex contract)
  - Client → server
    - Client verwacht geen antwoord
  - Server → client
    - Events op client
    - Informatie opvragen van client

Industrieel Ingenieur Informatica, UGent

## Eénrichtingscommunicatie

16

- Methode

- Geen returnwaarde
- Geen uitvoerparameters
- AttribootOperationContract
  - Eigenschap IsOneWay op true

```
[OperationContract(IsOneWay=true)]
void Hello(string greeting);
```

Industrieel Ingenieur Informatica, UGent

## Duplex communicatie

17

- Tweerichtingsverkeer
  - Geen vraag/antwoord
  - Communicatie kan in beide richtingen
    - Client start communicatie op (sessie)
      - Houdt kanaal open
    - Server kan later antwoorden
- Twee interfaces
  - Client → Server
  - Server → Client
    - Call back contract

Industrieel Ingenieur Informatica, UGent

## Duplex communicatie - syntax

18

```
[ServiceContract(Namespace =
    "http://Microsoft.ServiceModel.Samples",
    SessionMode=SessionMode.Required,
    CallbackContract=typeof(ICalculatorDuplex
    Callback))]
public interface ICalculatorDuplex {
    [OperationContract(IsOneWay = true)]
    void Clear();
    [OperationContract(IsOneWay = true)]
    void AddTo(double n);
    [OperationContract(IsOneWay = true)]
    void SubtractFrom(double n);
    [OperationContract(IsOneWay = true)]
    void MultiplyBy(double n);
    [OperationContract(IsOneWay = true)]
    void DivideBy(double n);
}
```

```
public interface ICalculatorDuplexCallback
{
    [OperationContract(IsOneWay = true)]
    void Result(double result);
    [OperationContract(IsOneWay = true)]
    void Equation(string eqn);
}
```

Implementatie op client

Implementatie op server

Industrieel Ingenieur Informatica, UGent

## Overzicht

19

- Ontwerp servicecontract
- Implementatie contract
- Configuratie
- Hosting
- Ontwikkelen client

Industrieel Ingenieur Informatica, UGent

## Implementatie servicecontract

20

□ Klasse die interface implementeert

```
[ServiceContract]
public interface IMath {
    [OperationContract]
    double Add(double A, double B);
    [OperationContract]
    double Multiply (double A, double B);
}

public class MathService : IMath {
    public double Add (double A, double B) { return A + B; }
    public double Multiply (double A, double B) { return A * B; }
}
```

Industrieel Ingenieur Informatica, UGent

## Overzicht

21

- Ontwerp servicecontract
- Implementatie contract
- Configuratie
- Hosting
- Ontwikkelen client

Industrieel Ingenieur Informatica, UGent

## Configuratie

22

- Specificiëren endpoint
  - Welke service?
    - Interface
    - Implementatie
  - Waar?
    - URL
  - Hoe communiceren? (binding)
    - TCP? HTTP?
    - Tekst? Binair?
    - Beveiligd?
- Configuratie = belangrijk deel WCF-programmatie

Industrieel Ingenieur Informatica, UGent

## Specifiëren endpoint

23

- Meestal in configuratie
- Kan ook in code

Industrieel Ingenieur Informatica, UGent

## Specifiëren endpoint in code

24

```
Uri baseAddress = new Uri("http://localhost:8000/ServiceModelSamples/Service");
// Adres

ServiceHost selfHost = new ServiceHost(typeof(CalculatorService),
baseAddress);
// Implementatie service

try {
    selfHost.AddServiceEndpoint( typeof(ICalculator), new WSHttpBinding(),
"CalculatorService");
    // Welke service? Binding
    ... // gedrag instellen, service runnen
} catch (CommunicationException ce) { Console.WriteLine("An exception occurred:
{0}", ce.Message); selfHost.Abort();
}
```

Industrieel Ingenieur Informatica, UGent

## Specifiëren endpoint in configuratie

25

```
<system.ServiceModel>
  <services>
    <service name="Microsoft.ServiceModel.Samples.CalculatorService">
      <binding name="myBindingConfiguration1" closeTimeout="00:01:00"/>
      <endpoint address="http://localhost:8000/ServiceModelSamples/Service"
        binding="basicHttpBinding" bindingConfiguration="myBindingConfiguration1"
        contract="Microsoft.ServiceModel.Samples.Calculator" />
    </service>
  </services>
  <bindings>
    <binding name="myBindingConfiguration1" />
  </bindings>
  <behaviors>
    <serviceBehaviors>
      <behavior>
        <serviceMetadata httpGetEnabled="True"/>
        <serviceDebug includeExceptionDetailInFaults="False" />
      </behavior>
    </serviceBehaviors>
  </behaviors>
</system.ServiceModel>
```

Klassenaam implementatie

URL

Binding

Interface

Opvragen metadata service

Gedrag service

Industrieel Ingenieur Informatica, UGent

## Bestaande bindingen

- 26
- Voorzien in WCF
  - Een aantal voorbeelden
    - BasicHttpBinding
      - HTTP
      - Tekst/XML
      - Bv. Asmx
    - WSHHttpBinding
      - Beveiligd, niet-duplex
    - NetTcpBinding
      - Beveiligd, communicatie tussen WCF-applicaties op verschillende machines
    - NetNamedPipeBinding
      - Beveiligd, geoptimaliseerd tussen WCF-applicaties op één machine
- Industrieel Ingenieur Informatica, UGent

## Overzicht

- 27
- Ontwerp servicecontract
  - Implementatie contract
  - Configuratie
  - Hosting
  - Ontwikkelen client
- Industrieel Ingenieur Informatica, UGent

## Hosting

- 28
- Publiceren
    - Runtime omgeving
      - Beheren levensloop en context
  - Windows proces
    - Ondersteunen managed code
    - Voorbeeld
      - IIS
      - Console-applicatie (zelf "hosten")
      - Windows service
      - ...
  - Code service onafhankelijk van host
- Industrieel Ingenieur Informatica, UGent

## Hosten via console-applicatie

29

```
Uri baseAddress = new Uri("http://localhost:8000/ServiceModelSamples/Service");
using (ServiceHost host = new ServiceHost(typeof(CalculatorService), baseAddress)) {

    host.AddServiceEndpoint(typeof(ICalculator), new WSHHttpBinding(), "CalculatorService");
    ServiceMetadataBehavior smb = new ServiceMetadataBehavior();
    smb.HttpGetEnabled = true;
    host.Description.Behaviors.Add(smb);

    host.Open();
    Console.WriteLine("The service is ready at {0}", baseAddress);
    Console.WriteLine("Press <Enter> to stop the service.");
    Console.ReadLine();

    host.Close();
}
```

Industrieel Ingenieur Informatica, UGent

## Hosten als Windows service

30

```
public class CalculatorWindowsService : ServiceBase {
    public ServiceHost serviceHost = null;
    public CalculatorWindowsService() {
        ServiceName = "WCFWindowsServiceSample";
    }

    public static void Main() {
        ServiceBase.Run(new CalculatorWindowsService());
    }

    protected override void OnStart(string[] args) {
        if (serviceHost != null) { serviceHost.Close(); }
        serviceHost = new ServiceHost(typeof(CalculatorService));
        serviceHost.Open();
    }

    protected override void OnStop() {
        if (serviceHost != null) {
            serviceHost.Close();
            serviceHost = null;
        }
    }
}

[RunInstaller(true)]
public class ProjectInstaller : Installer {
    private ServiceProcessInstaller process;
    private ServiceInstaller service;
    public ProjectInstaller() {
        process = new ServiceProcessInstaller();
        process.Account = ServiceAccount.LocalSystem;
        service = new ServiceInstaller();
        service.ServiceName = "WCFWindowsServiceSample";
        Installers.Add(process);
        Installers.Add(service);
    }
}
```

Industrieel Ingenieur Informatica, UGent

## Hosten in IIS

31

- Maak een service in IIS
  - TijdService.svc en TijdService.svc.cs
- Publiceer/Run
- WSDL
  - `http://localhost:49695/TijdService.svc?wsdl`

Industrieel Ingenieur Informatica, UGent

## Overzicht

32

- Ontwerp servicecontract
- Implementatie contract
- Configuratie
- Hosting
- Ontwikkelen client

Industrieel Ingenieur Informatica, UGent

## Client ontwikkelen

33

- Ophalen
  - Contract
  - Bindings
  - Adres
- Svcutil.exe
  - `svcutil.exe /language:cs /out:generatedProxy.cs /config:app.config`  
`http://localhost:8000/ServiceModelSamples/service`
  - Genereert
    - Configuratiebestand
    - Proxy-klasse
    - Beiden toevoegen aan clientproject
  - Kan ook via WSDL indien gehost op IIS

Industrieel Ingenieur Informatica, UGent

## Proxy gebruiken

34

- Proxy aanmaken
- Methodes proxy oproepen
- Proxy afsluiten

Industrieel Ingenieur Informatica, UGent

## Proxy gebruiken

35

```
class Client {
    static void Main() {
        CalculatorClient client = new CalculatorClient();
        double value1 = 100.00D;
        double value2 = 15.99D;
        double result = client.Add(value1, value2);

        client.Close();
    }
}
```

Industrieel Ingenieur Informatica, UGent

## Informatie

37

- WCF (<https://docs.microsoft.com/en-us/dotnet/framework/wcf/index>)

Industrieel Ingenieur Informatica, UGent