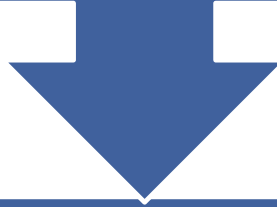


CONF.DR. CRISTIAN KEVORCHIAN
FACULTATEA DE MATEMATICA
SI INFORMATICA
UNIVERSITATEA DIN
BUCUREȘTI

ARHITECTURI SOA

Un serviciu web este o colecție de protocoale și standarde deschise utilizate pentru schimbul de date dintre aplicații sau medii de calcul. Aplicațiile software scrise în diverse limbaje de programare și executate pe platforme eterogene pot utiliza servicii web pentru a schimba date peste rețele de calculatoare(ex. Internet) într-o manieră similară comunicării între procesele unui sistem de calcul. Interoperabilitatea dintre funcționalități scrise în C#, Java sau Python cu aplicații rulate pe Windows sau Linux se realizează în baza standardelor deschise.



Sintetizând, un serviciu web este un serviciu, care îndeplinește următoarele condiții:

Este disponibil peste Internet

Utilizează un sistem de mesaje bazat pe un XML standardizat

Utilizarea lui nu depinde de sistemul de operare și limbaje de programare utilizat

Se auto-descrie prin intermediul unei gramatici XML comune

Mecanism de identificare a serviciului foarte simplu.

SERVICII WEB-DEFINIȚIE

COMPONENTELE SERVICIILOR WEB

Platforma serviciilor web este XML + HTTP. Standardul WS utilizează următoarele elemente:

- SOAP (Simple Object Access Protocol)
- UDDI (Universal Description, Discovery and Integration)
- WSDL (Web Services Description Language)

SOAP

SIMPLE OBJECT APPLICATION PROTOCOL
-DEFINIȚIE-

SOAP este un protocol simplu destinat schimbului de informații structurate într-un mediu distribuit. SOAP utilizează tehnologii XML pentru a defini un framework extensibil de mesagerie, care oferă o structură a mesajului capabilă să interopereze cu o familie de protocoale compozite.

Frameworkul a fost conceput pentru a fi independent de orice model particular de programare, precum și de alte semantici de implementare.



WS-Secure
Conversation

WS-Federation

WS-Authorization

WS-Policy

WS-Trust

WS-Privacy

WS-Security

SOAP

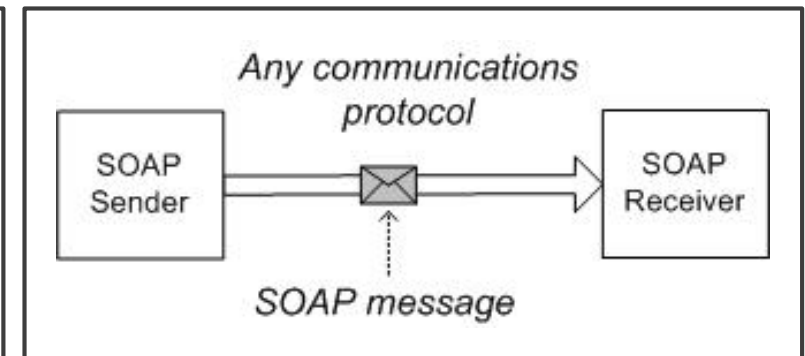
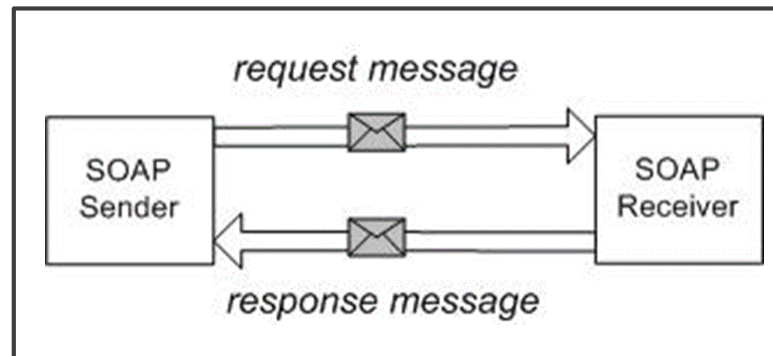
SOAP-PROTOCOLS COMPOSITE

FRAMEWORK-UL DE MESAGERIE SOAP(I)

- Frameworkul de mesagerie bazat pe XML se caracterizează prin:
 - Extensibilitate
 - Simplitatea a caracterizat design-ul pentru SOAP
 - SOAP definește un framework de comunicații care permite adăugarea de nivele suplimentare ca extensie pentru funcționalități cum ar fi: securitate, rutare și fiabilitate.
 - Interoperabilitate
 - SOAP poate fi folosit peste orice protocol de transport, cum ar fi TCP, HTTP, SMTP
 - SOAP expune o legătură explicită cu HTTP

FRAMEWORKUL DE MESAGERIE SOAP(II)

- Independență
 - SOAP interoperează cu orice model de programare și nu este legat de RPC(Remote Procedure Call)
 - SOAP permite lucrul cu un număr nelimitat de modele MEP(Message Exchange Pattern[W3C]).
 - SOAP definește un model de prelucrare individuală bazat pe mesaje uni-direcționale:



FORMATUL MESAJELOR SOAP

-
- **Un mesaj SOAP are trei părți:**
 - Înfășurătoarea SOAP(SOAP Envelope)
 - Opțional header-ul SOAP
 - Corpul mesajului SOAP

Sursa: <http://schemas.xmlsoap.org/soap/envelope>

ÎNFĂȘURĂTOAREA SOAP

- Este definit un framework general pentru a stabili structura mesajului și cum acesta este procesat.
- Este rădăcina structurii ierarhice XML asociată mesajului SOAP.
- Înfășurătoarea SOAP include opțional Headerul și în mod obligatoriu elementul **BODY**

FORMA GENERALA A CODULUI ÎNFĂȘURĂTORII SOAP

```
<soap:Envelope
  xmlns:soap="http://schemas.xml
  soap.org/soap/envelope/">
  <soap:Header> <!-- optional --
  >

    <!-- conținut "header"... -->
  </soap:Header>

  <soap:Body>

    <!--conținut "body"... -->
  </soap:Body>
</soap:Envelope>
```

HEADERUL SOAP

- Header-ul este un container generic destinat controlului informației.
- Acesta poate conține orice număr de elemente din orice namespace
- Blocurile de antet trebuie să conțină informații care influențează prelucrarea unui conținut informațional util.
- Header-ul este opțional

FORMA GENERALA A CODULUI HEADER-ULUI SOAP

```
<soap:Header>  
  <!-- credentiale asociate securității-->  
  <s:credentials xmlns:s="urn:examples-  
org:security">  
    <username>ckadmin</username>  
    <password>Cristi.1955</password>  
  </s:credentials>  
</soap:Header>
```

CORPUL(BODY) SOAP

Elementul "BODY" reprezintă conținutul informațional al mesajului SOAP:

```
<soap:Body>
```

```
  <x:TransferBani xmlns:x="urn:examples-org:banking">  <from>22-  
  222222</from>
```

```
    <to>33-333333</to>
```

```
    <amount>444.44</amount>
```

```
  </x:TransferFunds>
```

```
</soap:Body>
```

EXEMPLU

```
<soap:Envelope>
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
    <soap:Header>
      <!-- credentiale-->
      <s:credentials xmlns:s="urn:examples-org:security">
        <username>ckadmin</username>
      </s:credentials>
    </soap:Header>
    <soap:Body>
      <x:TransferBani xmlns:x="urn:examples-org:banking">
        <from>22-222222</from>
        <to>33-333333</to>
        <amount>444.44</amount>
        <curency>RON</curency>
      </x:TransferBani>
    </soap:Body>
  </soap:Envelope>
```

01

Specificațiile SOAP nu definesc criptarea pentru Servicii Web XML

02

Criptarea se plasează la nivelul protocolul de transport(SSL)

03

Evaluarea costurilor criptarii totale sau parțiale a datelor.

04

Aceast aspect este lăsat în seama celor care implementează protocolul SOAP.

SECURITATE SOAP

EXEMPLU:
**SOAP CU CRIPTAREA
DATELOR**

```
<%@WebService Language="C#" Class="CreditCardService" %>
using System.Web.Services;

public class CreditCardService {
    [WebMethod]
    [EncryptionExtension(Encrypt=EncryptMode.Response)]
    public string GetCreditCardNumber() {
        return "MC: 4111-1111-1111-1111";
    }
}
```


CERERE/RASPUNS PENTRU CRIPTARE

CERERE	RASPUNS
<pre><?xml version="1.0" encoding="utf-8"?> <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"> <soap:Body> <GetCreditCardNumber xmlns= "http://tempuri.org/" /> </soap:Body> </soap:Envelope></pre>	<pre><soap:Body> <GetCreditCardNumber xmlns="http://tempuri.org/"> <GetCreditCardNumberResult>83 151 243 32 53 95 86 13 190 134 188 241 198 209 72 114 122 38 180 34 194 138 16 97 221 195 239 86 26 152 94 27 </GetCreditCardNumberResult> </GetCreditCardNumber> </soap:Body></pre>

MODELUL DE PROGRAMARE WCF

WCF(la inceputuri proiectul Indigo) este un framework pentru dezvoltarea de aplicații SOA (Service-Oriented Application)

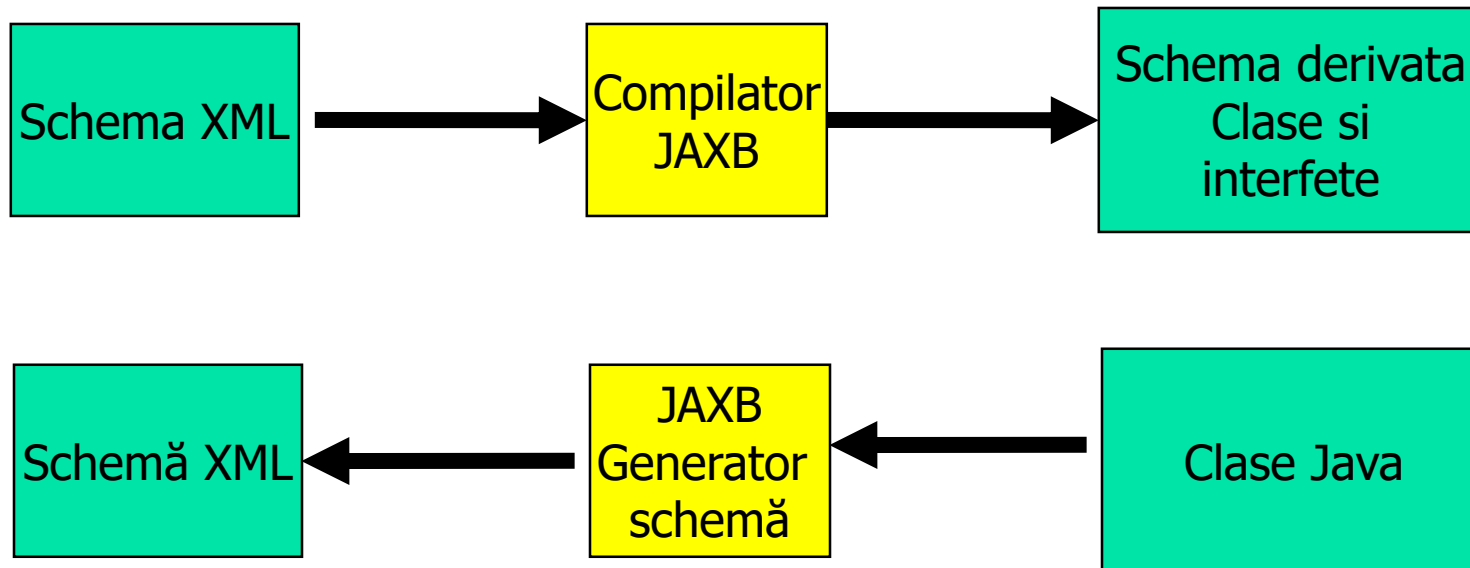
WCF implementează mai multe servicii web standard: WS-Adressing, WS-ReliableMessaging și WS-Security.

Au fost adăugate funcționalități pentru găzduirea soluțiilor WCF în IIS(Internet Information Services), serverul web din Windows, dar și în cloud.

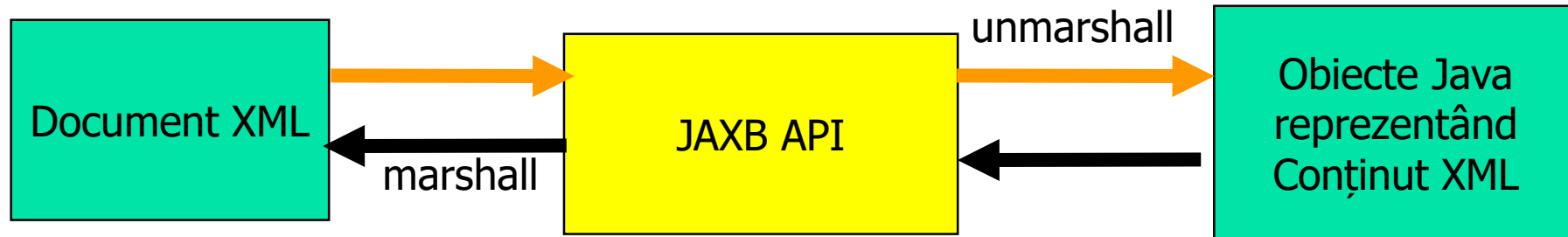
MODELUL DE PROGRAMARE JAX-WS-JAXB

JAVA ARCHITECTURE FOR XML BUILDING

- Furnizează o legătură între XML și Java
- Asociază o familie de clase Java cu XML
- Abordează clasele Java ca reprezentând instanțe XML



JAXB LA RUNTIME



JAX-WS

- JAX-WS poate fi utilizat pentru implementare SOAP atât la nivel de server cât și la nivel de client.
- Terminologie, a se citi jargon SOA:

XML XSD

XSLT Xpath JAXP

SAX DOM JAXB StaX

SOAP WSDL UDDI Rest

JAX-RPC JAX-WS JAX-RS

SAAJ WS* BP

Axis Metro

ESB

SOA

CONSIDERATII ARHITECTURALE

- Simplificat, un serviciu poate fi interpretat drept o funcționalitate expusă în scopul utilizării locale sau “remote”.
- Serviciile și aplicațiile care consumă aceste funcționalități interacționează prin intermediul mesajelor.
- Mesajele pot fi transeferate direct de la serviciu la consumator, sau prin servicii în cloud cum ar fi Azure ServiceBus (mesajele fiind SOAP)
- Serviciul este “opac” din exterior. WCF expune metadatele prin WSDL
- Un **endpoint** este o resursă în rețea de unde pot fi trimise mesaje.
- Un serviciu poate fi văzut ca o familie de endpoint-uri.
- Clienții nu interacționează cu serviciile în mod direct, fiind utilizat un proxy.

CONTRACT ASOCIAT SERVICIULUI IN WCF

- În cadrul WCF termenul de contract asociat serviciului(service contract) reprezintă o familie de operații(metode) pe care le expune un serviciu WCF și este marcat cu atributului [ServiceContract].
- Contractul descrie funcționalitățile serviciului disponibilzate consumatorului.
- Un contract este compus dintr-un **contract interfață** și un **endpoint**
- In economia serviciului acesta poate fi comparat cu un contract de tip SLA(Service Level Agreement).
- Contractul interfață este compus din:
 - **Contractul operațional**-este definiția unei operațiuni asociată serviciului WCF, metoda, sau expunerea funcționalității serviciului ca parte a **contractului interfață**
 - **Contractul date** – este utilizat pentru a genera o schemă XML peste tipurile CLR. Modelul datelor impune structura și tipul mesajului.

WSDL și XSD

Contractul este scris în WSDL și XSD(XML Schema Definition) și lucrează cu tipuri CLR.WCF facilitează această mapare:

- Serviciu se structurează astfel: **[ServiceContract]**, **[OperationContract]**, **[FaultContract]**,**[MessageContract]** și/sau **[DataContract]**.
- Pentru codul programului care consuma serviciul cerem serviciului detalii despre contracte, în plus trebuie să fie generată o clasă proxy ce expune interfața serviciului, pe care o putem apela din cod.
- Svcutil.exe –este un utilitar care expune metadate legate de serviciu
- Este generat codul WSDL și tipurile .NET și .XSD
- Svctraceviewer.exe – utilitar grafic pentru citit log-uri WCF, formatul mesajelor catre si din endpoint-uri.

ENDPOINT

Fiecare serviciu este asociat cu o **adresa** care definește unde se afla serviciul, iar o **legatura** definește cum se poate comunica cu serviciul. **Contractul** stabilește ce face serviciul.

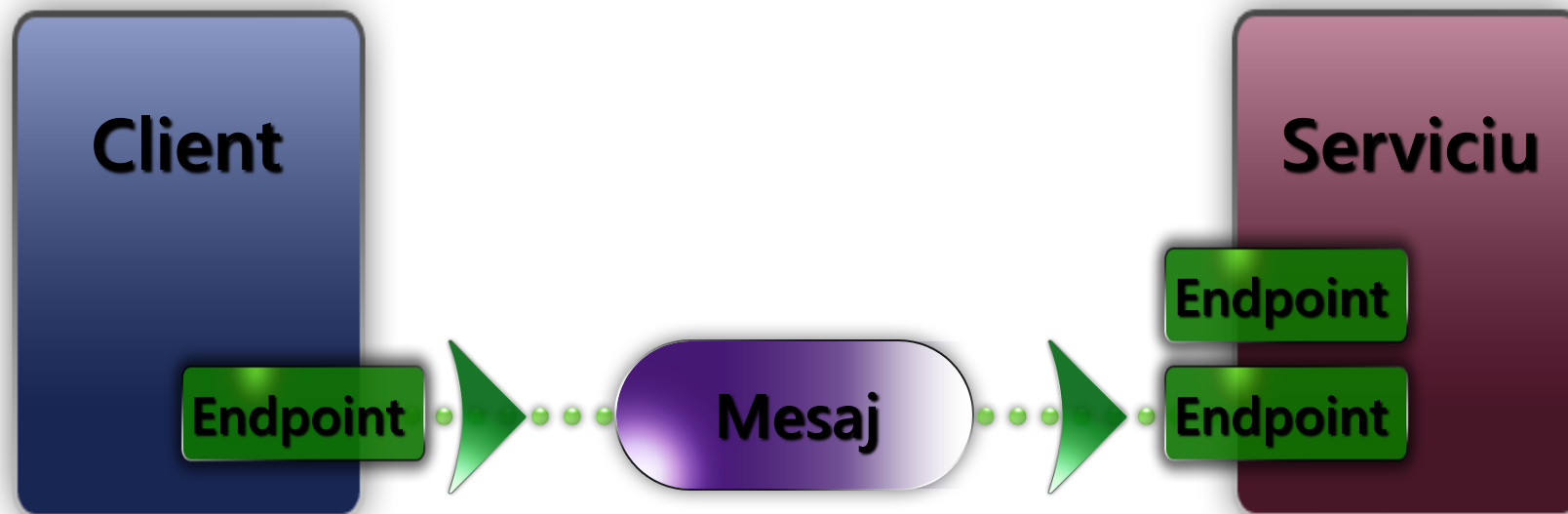
Tripleta (Adresa, Legatura, Contract) este formalizată în WCF sub forma unui endpoint

Fiecare endpoint trebuie să înregistreze aceste trei elemente

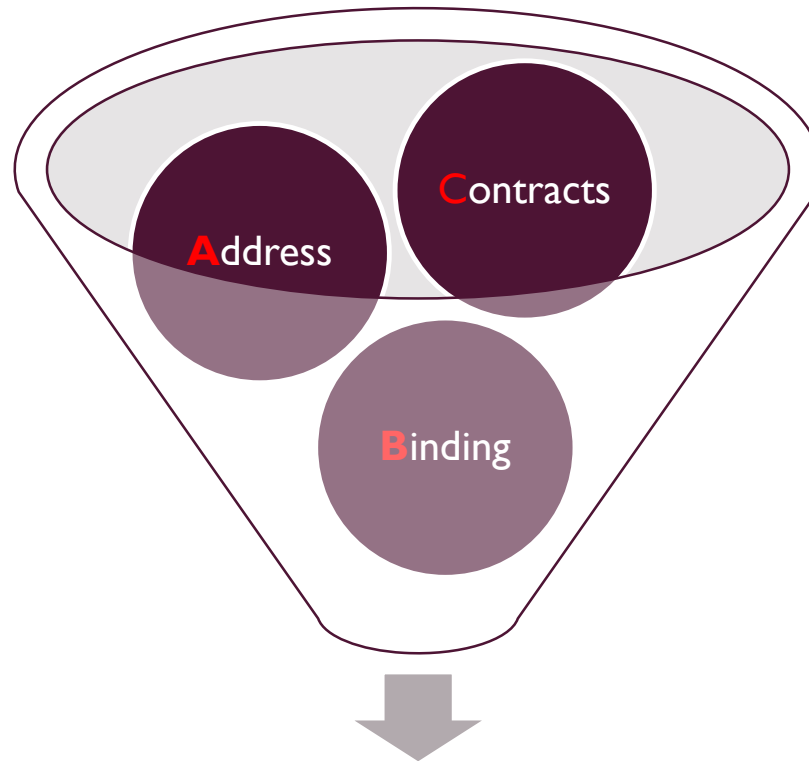
Practic un endpoint este o interfață similară cu o interfață COM(**C**omponent **O**bject **M**odel).

Fiecare serviciu trebuie să expună un singur **<<business endpoint>>**

ENDPOINT



ENDPOINT



Endpoint

MICROSOFT INNOVATION & PRACTICE TEAM, MSCOE

MODELE DE DATE ȘI ATRIBUTUL **DataContract**

- Atributul **DataContract** este utilizat pentru a genera o schemă XML dintr-un tip CLR. Membrii unui tip de date nu vor fi serializati cu excepția atributului **DataMember**
- **DataContract** utilizează **XmlFormatter** pentru a lucra cu serializarea și tipurilor CLR, respectiv deserializarea schemei XML în tipuri CLR.
- **WCF** vine cu doua instrumente de serializare **XmlSerializer** și **XmlFormatter** succesor al primului și implicit implementat în WCF.
- Utilizat când **DataContract** și **DataMember** sunt utilizate într-un tip de dată CLR.

EXEMPLU

```
public class Cont
{
    public string Număr_Cont;
    public string Nume_Cont;
    public string Tip;
    public string Cash;
    public string Investitii;
}
```

```
using System.Runtime.Serialization;
[DataContract(Namespace = "Financiar", Name = „ContContract")]
public class Cont
{
    [DataMember(Name = „Numar_Cont")]
    public string Numar_Cont;
    [DataMember(Name = "Nume_Cont")]
    public string Nume_Cont;
    [DataMember(Name = „Tip", Order = 1)]
    public string Tip;
    [DataMember(Name = "Cash", Order = 1)]
    public string Cash;
    [DataMember(Name = "Investitii", Order = 2)]
    public string Investitii;
    [DataMember(Name = "Total", Order = 2)]
    public string Total;
}
```

MESAJE ȘI ATRIBUTUL MESSAGECONTRACT

- WCF utilizează mesaje SOAP alinate standardelor industriale SOAP și schema XML
- Mesajele pot fi transmise folosind protocoale de transport(HTTP, HTTPS, MSMQ sau TCP)
- MEP(Message Exchange Pattern) include varianta sincronă și asincrona.
- WCF include un framwork pentru managementul mesajelor(System.ServiceModel).
- Clasele MessageContract, MessageHeader și MessageBody utilizate ca attribute la descrierea structurii mesajului SOAP

EXEMPLU

```
[DataContract]
    public class MsgHeader {
[DataMember]
    public string loginName;
[DataMember]
    public string password;
    }
```

```
[DataContract]
public class MsgBody
{
[DataMember(Order=1)]
public string name;
[DataMember(Order=2)]
public int phone;
[DataMember(Order=3)]
public int email;
}
```

```
[MessageContract]
public class ContactMessage
{
[MessageHeader]
public MsgHeader msgHeader;
[MessageBody]
public MsgBody msgBody;
}
```

SERVICIUL ENDPOINTS ȘI ELEMENTUL ENDPOINT

- **Un endpoint conține:**
 - **Adresa**-locatia serviciului
 - **Legătura**-cerinte invocate de serviciu(ex. securitate și fiabilitate).
 - **Contract**

```
<system.serviceModel>  
<services>  
<service name="AccountService">  
<endpoint name="EndPoint1"  
address="net.tcp://localhost:1234"  
binding="netTcpBinding"  
contract="IAccount" />  
</service>  
</services>  
</system.serviceModel>
```

```
<system.serviceModel>  
<services>  
<service name="AccountService">  
<endpoint name="EndPoint1"  
address="net.tcp://localhost:1234"  
binding="netTcpBinding"  
contract="IAccount" />  
<endpoint name="EndPoint3"  
address="http://localhost:8000"  
binding="basicHttpBinding"  
contract="IAccount" />  
</service>  
</services>  
</system.serviceModel>
```




Q&A