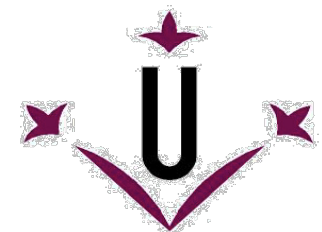


Enginyeria del Software III

# SERVEIS WEB SOAP

Roberto García  
Despatx EPS 3.15  
rgarcia@diei.udl.cat



# SOAP

## SIMPLE OBJECT ACCESS PROTOCOL

- **Estàndard** per a la interconnexió
- Permet l'intercanvi d'informació estructurada i amb tipus entre entitats descentralitzades
- **Codificació i empaquetament** basat en XML per a intercanviar dades
- SOAP proporciona principalment:
  - El element “envelope”
  - Un conjunt de regles de codificació
  - La representació de crides de mètodes remots



# SOAP ESPECIFICACIÓ

- Bàsicament formada per:
  - Una **sintaxi** per la definició de missatges en forma de documents XML (referits com missatges SOAP)
  - Un **model** per l'intercanvi de missatges
  - Un conjunt de **regles** per representar dades dins dels missatges SOAP (codificació SOAP)
  - Pautes pel **transport** de missatges SOAP amb HTTP
  - Convenció per realitzar crides a **procediments remots**



# SOAP

## SIMPLE OBJECT ACCESS PROTOCOL

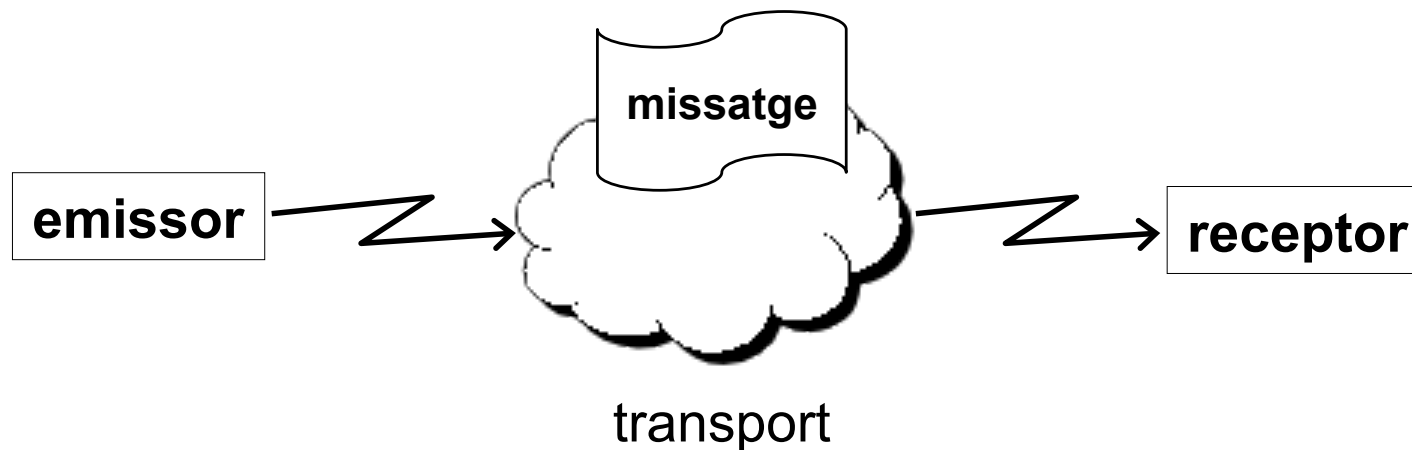
- Estàndard del W3C”
  - *“Mecanisme de definició simple i lleuger per l’intercanvi estructurat d’informació entre punts d’un entorn descentralitzat i distribuït utilitzant XML”*
  - <http://www.w3.org/TR/soap12-part0>



# SOAP

## MODEL D'INTERCANVI DE MISSATGES

- El model d'intercanvi de missatges depèn de tres conceptes bàsics:
  - Els missatges són **documents** XML, que
  - viatgen d'un **emissor** a un **receptor**



# INTERCANVI DE MISSATGES XML

- Tots els missatges SOAP són **documents XML**
- Avantatges sobre altres protocols de missatgeria:
  - Edició i/o lectura simple (ex.: editor de text)
  - Depuració més simple que si utilitzéssim un protocol binari
  - Popularitat XML
  - Moltes eines disponible en tots el sistemes i plataformes



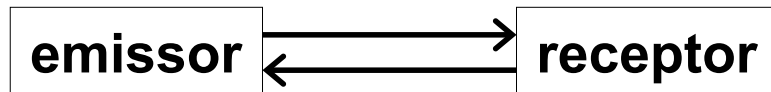
# SOAP

## DISSENY MODULAR

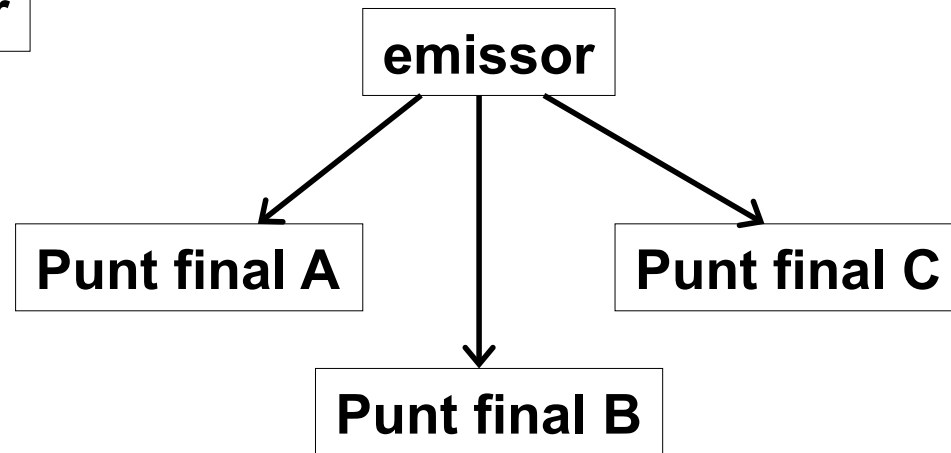
- Al ser obert i extensible es pot utilitzar en situacions molt diferents emprant el mateix model
- Està dissenyat per ser extensible en tots aquests sectors:
  - **Sintaxi del missatge:** un sector independent de capçalera (header) pensat per ampliacions futures
  - **Dades:** qualsevol tipus de dades
  - **Transport:** SOAP només defineix com s'intercanvien els missatges via HTTP, però es pot implementar qualsevol altre protocol



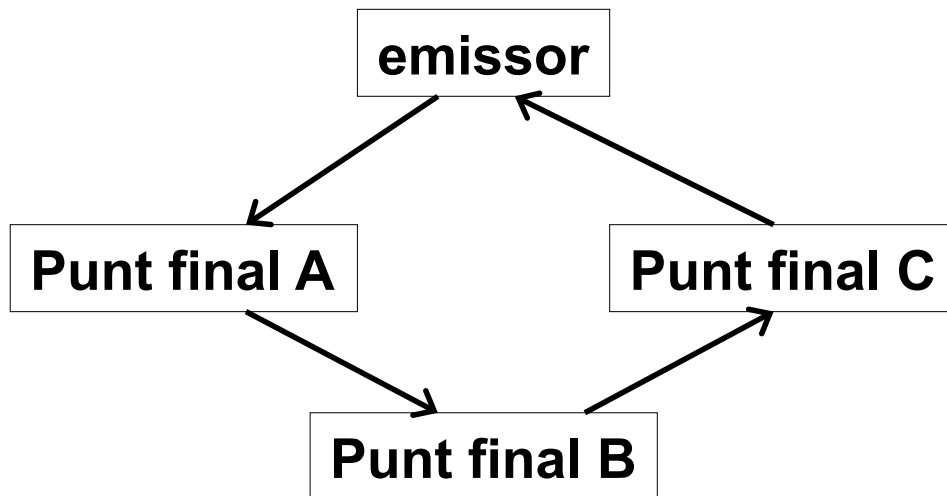
# CADENES DE MISSATGES



Petició-resposta



difusió



Fluxe de treball





# COMPORTAMENT DELS PUNTS DE SERVEI (ENDPOINTS)

- El **patró emisor-receptor** es repeteix per compondre diferents cadenes de missatges
- Plantejar-se SOAP en termes dels **punts de servei** ajuda a comprendre la flexibilitat de la missatgeria SOAP
  - Independentment de la ruta que pren un missatge o del n° de punts de servei que el processin, tots ells han de processar el missatge d'una determinada manera

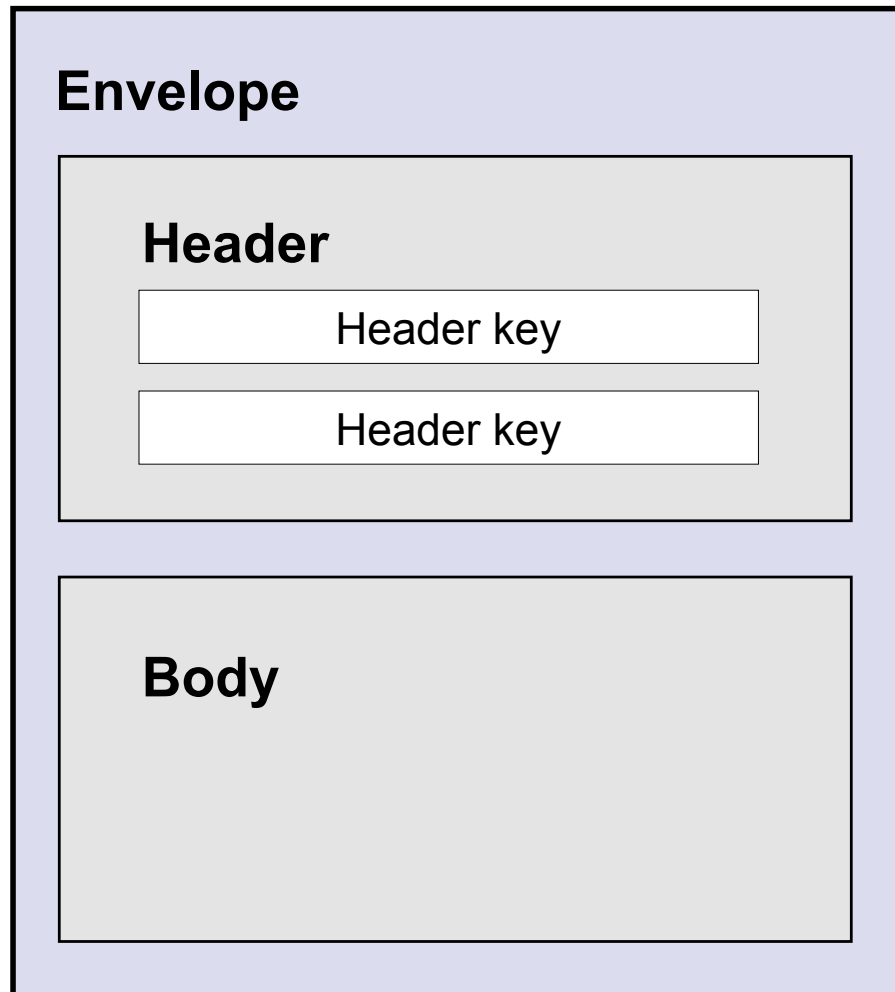


# COMPORTAMENT DELS PUNTS DE SERVEI (ENDPOINTS)

- Passos que ha de seguir per un missatge, per ajustar-se al procés definit a l'especificació:
  1. Analitzar el missatge SOAP per determinar si conté alguna informació destinada a ell
  2. Si té informació per ell i pot processar-la, ho fa, si no el missatge sencer es rebutjat
  3. Si el punt extrem és un intermediari, s'extrauen totes les parts identificades en el primer pas abans de passar-lo al punt de servei següent



# ESTRUCTURA D'UN MISSATGE SOAP



Lloc per les ampliacions del missatge

Lloc per la càrrega útil del missatge

# PRIMER EXEMPLE

*Espai de noms oficial pels missatges SOAP*

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"  
  soap:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
```

*Atribut que descriu com es representaran les dades.  
No és obligatori*

```
<soap:Header>  
  <h:from xmlns:h="http://www.wrox.com/Header">Soap@wrox.com</h:from>  
</soap:Header>
```

```
<soap:Body>  
  <w:GetSecretIdentity xmlns:w="http://www.wrox.com/heroes/">  
    <w:codename>XSLT-Man</w:codename>  
  </w:GetSecretIdentity>  
</soap:Body>
```

```
</soap:Envelope>
```



# PRIMER EXEMPLE BODY

```
<soap:Body>  
  <w:GetSecretIdentity xmlns:w="http://www.wrox.com/  
    heroes/">  
    <w:codename>XSLT-Man</w:codename>  
  </w:GetSecretIdentity>  
</soap:Body>
```

- Ubicació de les dades per una aplicació específica
  - Conté la “càrrega útil” del missatge (dades que representen l’objectiu del missatge)
- Pot ser:
  - Una crida a un procediment remot
  - Una comanda de compra
  - Un full d’estil
  - Qualsevol XML que necessiti intercanviar-se mitjançant el missatge
- Depèn directament de “envelope” i és obligatori



# PRIMER EXEMPLE HEADER

```
<soap:Header>  
  <h:from xmlns:h="http://www.wrox.com/Header">  
    Soap@wrox.com  
  </h:from>  
</soap:Header>
```

- **Objectiu:** encapsulació de les extensions del format del missatge
- Conté **meta-informació** que permet extensions com transaccions, encriptació de dades, referències d'objectes...
- És **opcional**, però si hi va s'ha de situar just després de “envelope”



# MÉS ASPECTES DE HEADER

- Conté, si és el cas, informació processable pels intermediaris
- Atributs predefinitos:
  - mustUnderstand
    - true (1): l'element és obligatori i si no pot processar-se ha de retornar un error
    - false (0): l'element és opcional. És el valor per defecte.

**<soap:Header>**

```
<h:from xmlns:h="http://www.wrox.com/Header"  
        soap:mustUnderstand="1">
```

```
    SoapGuy@wrox.com
```

```
</h:from>
```

**</soap:Header>**



# MÉS EXEMPLES

```
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
```

```
  <env:Header>
```

```
    <n:alertcontrol xmlns:n="http://example.org/alertcontrol">
```

```
      <n:priority>1</n:priority>
```

```
      <n:expires>2001-06-22T14:00:00-05:00</n:expires>
```

```
    </n:alertcontrol>
```

```
  </env:Header>
```

```
  <env:Body>
```

```
    <m:alert xmlns:m="http://example.org/alert">
```

```
      <m:msg>Pick up Mary at school at 2pm</m:msg>
```

```
    </m:alert>
```

```
  </env:Body>
```

```
</env:Envelope>
```





# MÉS ASPECTES DE HEADER

## ○ Atributs predefinitos:

### • actor

- Indica quin node ha de processar la informació
- Uneix un element Header amb un punt de servei particular
- En cas de no ser-hi, l'encarregat de processar-la és el node receptor final
- El valor és una URI que identifica aquest punt de servei
- Quan un punt intermediari troba una entrada per a ell ha d'eliminar els elements Header que processa
  
- Cas particular:
  - **actor="http://schemas.xmlsoap.org/soap/actor/next"**  
indica que el missatge es dirigeix al següent punt de servei amb el que es trobi el missatge



# SOAP: FAULT

## FORMAT PREDEFINIT DE MISSATGE D'ERROR

- L'element fault constitueix la forma estàndard de retornar a l'emissor d'un missatge un informe sobre un comportament inesperat
  - Venen a ser les **excepcions** en els serveis web
- En cas de ser-hi ha d'estar en la càrrega útil del missatge, o sigui, com un element fill de body



# SOAP: FAULT

## FORMAT PREDEFINIT DE MISSATGE D'ERROR

- Subelements de fault
  - **faultcode**: codi de l'error
    - **Valors predefinitos**:
      - **VersionMismatch**: hi ha un error en l'espai de noms de “envelop”
      - **MustUnderstand**: valor retornat quan un punt de servei rep una entrada que no reconeix (i li havíem dit que havia de reconèixer-la)
      - **Client/Receiver**: el client troba un problema en el missatge rebut
      - **Server/Sender**: un problema durant el processament no relacionat amb el contingut del missatge
      - **Altres**: DTDNotSupported, DataEncodingUnknown
    - Poden extendre's: exemple `Server.OutOfMemory`



# SOAP: FAULT

## FORMAT PREDEFINIT DE MISSATGE D'ERROR

- Subelements de fault (continuació)
  - **faultstring**: explicació dirigida a les persones
  - **detail**: informació específica de l'aplicació
    - Pot contenir elements XML
  - **faultactor**: URI del node que ha causat l'error



# APLICAT AL PRIMER EXEMPLE

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
    soap:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  ...
  <soap:Body>
    ...
    <soap:Fault>
      <faultcode>soap:Client.MissingParameter</faultcode>
      <faultstring>A parameter was missing</faultstring>
      <faultactor>http://www.wrox.com/heroes/endpoint.asp</faultactor>
      <detail>
        <w:error xmlns:w="http://www.wrox.com/">
          <code>178</code>
          <descr>The codename parameter was missing.</descr>
        </w:error>
      </detail>
    </soap:Fault>
  </soap:Body>
</soap:Envelope>
```



# RECAPITULEM FUNCIONAMENT DEL PUNT DE SERVEI

- Passos (transparentes per l'usuari) del processament del missatge:
  1. Examinar el missatge per comprovar si conté informació dirigida al punt de servei concret
    - Header → si hi ha entrades tant per posició (següent, últim) o per URI
    - Si és l'últim també s'analitza el contingut
  2. S'estudien les entrades de l'encapçalament dirigides a aquest punt de servei
    - Si hi un mustUnderstand="1" que no entén ha de retornar el fault corresponent
  3. Si el punt de servei és un intermediari, ha d'eliminar les entrades de l'encapçalament que s'han processat abans d'enviar el missatge al punt següent
    - Recordem que el body només el processa l'últim punt



# ESTIL DE CODIFICACIÓ

- L'atribut **encodingStyle** defineix les regles de codificació que utilitza el missatge
  - NO n'hi ha cap de predeterminada
  - Per tant si no hi és o si encodingStyle="" el receptor no té cap referència per verificar els tipus de dades que rep
- Disposem d'alguns tipus bàsics predefinitos
  - sencers, cadenes, coma flotant,...
- Regles específiques per a:
  - Estructures, Arrays i Referències
- Es complementa amb els XML Schemas



# ESTIL DE CODIFICACIÓ

## TIPUS BÀSICS

Els tipus primitius  
provenen de  
l'schema XML

```
<?xml version='1.0' ?>  
<soap:Envelope xmlns:soap='http://www.w3.org/2001/12/soap-envelope'  
  xmlns:xsi="http://www.w3.org/2001/XMLSchema" ←  
  encodingStyle='http://www.w3.org/2001/12/soap-encoding'>
```

```
  <soap:Body>  
    <g:article>  
      <g:codi xsi:type='soap:int'>234</g:codi>  
      <g:tamany xsi:type='soap:string'>article llarg</g:tamany>  
    </g:article>  
  </soap:Body>
```

```
</soap:Envelope>
```

*xsi:type* serveix per facilitar la comunicació entre una ampla varietat de llenguatges





# ESTIL DE CODIFICACIÓ ENUMERACIONS

- Reutilitzats de XML, SOAP permet definir tipus enumerats
  - **enumeració**: conjunt de valors basats en un tipus bàsic

## Definició

```
<simpleType name="Regio" base="xsd:string">  
  <enumeration value="Nord">  
  <enumeration value="Sud">  
  <enumeration value="Est">  
  <enumeration value="Oest">  
</simpleType>
```

## Utilització

```
<soap:Envelope xmlns:soap="http ..."  
  soap:encodingStyle="http ..."  
  xmlns:xsi="http ...">  
  
  <soap:Body>  
    <t:ObtenirVendes xmlns:t="http://example.org/regions"  
      <t:reg xsi:type="t:Regio">Oest</t:reg>  
    </t:ObtenirVendes>  
  </soap:Body>  
  
</soap:Envelope>
```



# ESTIL DE CODIFICACIÓ

## TIPUS COMPOSTOS: ESTRUCTURES

```
<Book xmlns="http://example.org/2001/06/books" >  
  <author>Henry Ford</author>  
  <title>Prefactory</title>  
  <year>1998</year>  
</Book>
```

**Exemple d'una  
estructura d'un  
llibre**

**Fragment d'un  
esquema  
descriuint aquesta  
estructura**

```
<xsd:element name="Book"  
  xmlns:xsd='http://www.w3.org/2001/XMLSchema' >  
  <xsd:complexType>  
    <xsd:sequence>  
      <xsd:element name="author" type="xsd:string" />  
      <xsd:element name="title" type="xsd:string" />  
      <xsd:element name="year" type="xsd:gYear" />  
    </xsd:sequence>  
  </xsd:complexType>  
</xsd:element>
```

**Estructura Genèrica**

```
struct Book{  
  string author;  
  string title;  
  int year;  
}
```

**Ús estructura en SOAP**

```
<Book>  
  <author xsi:type="xsd:string">Joan</author>  
  <title xsi:type="xsd:string">Puig Pelat</title>  
  <year xsi:type="xsd:gYear">43</year>  
</Book>
```



# ESTIL DE CODIFICACIÓ

## TIPUS COMPOSTOS: ARRAYS

- Array de enters:

```
<myFavoriteNumbers xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:enc="http://www.w3.org/2001/06/soap-encoding"
  xsi:type='enc:Array' enc:arrayType="xs:int[2]" >

  <number>3</number>
  <number>4</number>
</myFavoriteNumbers>
```

- Array de pizzas:

```
<pizzas xsi:type='enc:Array' enc:arrayType='p:Pizzas[2] '>
  <pizza>
    <codi>234</codi>
    <nom>Barbacoa</nom>
  </pizza>
  <pizza>
    <codi>237</codi>
    <nom>Barbacoa</nom>
  </pizza>
</pizzas>
```



# ESTIL DE CODIFICACIÓ

## TIPUS COMPOSTOS: ARRAYS

### *Array amb elements de varis tipus de dades*

```
<things xmlns:enc="http://www.w3.org/2001/06/soap-encoding"  
  xmlns:xs="http://www.w3.org/2001/XMLSchema"  
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
  xsi:type='enc:Array' enc:arrayType="xs:anyType[4]">
```

```
<thing xsi:type="xs:int">12345</thing>
```

```
<thing xsi:type="xs:decimal">6.789</thing>
```

```
<thing xsi:type="xs:string">
```

```
  Of Mans First Disobedience, and the Fruit  
  Of that Forbidden Tree, whose mortal tast  
  Brought Death into the World, and all our woe,
```

```
</thing>
```

```
<thing xsi:type="xs:anyURI">
```

```
  http://www.dartmouth.edu/~milton/reading_room/
```

```
</thing>
```

```
</things>
```



# ESTIL DE CODIFICACIÓ

## TIPUS COMPOSTOS: ARRAYS

### ○ Arrays parcials

```
<pizzas xsi:type='enc:Array' enc:arrayType='p:Pizzas[10]' enc:offset='[4] '>
  <pizza>
    <codi>234</codi>
    <nom>Barbacoa</nom>
  </pizza>
  <pizza>
    <codi>237</codi>
    <nom>Barbacoa</nom>
  </pizza>
</pizzas>
```

*Elements 5è i 6è*

```
<pizzas xsi:type='enc:Array' enc:arrayType='p:Pizzas[10] '>
  <pizza enc:position='2' >
    <codi>234</codi>
    <nom>Barbacoa</nom>
  </pizza>
  <pizza enc:position='5' >
    <codi>237</codi>
    <nom>Barbacoa</nom>
  </pizza>
</pizzas>
```

*Elements 2n i 5è*



# SOAP

## TRANSPORT DE MISSATGES

- Una de les principals virtuts de SOAP és la separació entre la definició del missatge i el **protocol de transport** utilitzat
  - Podem utilitzar
    - HTTP
    - SMTP
    - Arxius
    - Sockets
    - ...
  - Tot i que gairebé tothom utilitza HTTP



# SOAP

## TRANSPORT DE MISSATGES - HTTP

- Dues regles bàsiques per la utilització de HTTP com transport de SOAP
  - Enviament de missatge → **HTTP POST**
  - Envia un bloc de dades a un URI particular en el servidor web
    - En el cas dels missatges SOAP aquest bloc de dades és el propi missatge SOAP
  - Com SOAP és XML l'encapçalament del tipus contingut del HTTP POST ha de ser **text/xml**



# TRANSPORT HTTP

## ENVIAMENT DEL MISSATGE

- 1ra línia: mètode HTTP POST, fragment URI del punt de servei
  - En el context del servidor que rep el missatge
- 2na línia: tipus de contingut i la codificació
- 3ra línia: mida del contingut en bytes
- 4ta línia: mètode SOAP (detalls més endavant)

```
POST /StockQuote HTTP/1.1
Content-Type: text/xml; charset="utf-8"
Content-Length: nnnn
SOAPAction: "http://electrocommerce.org/abc#MyMessage"

<env:Envelope xmlns:env="http://www.w3.org/2001/06/soap-envelope" >
...
</env:Envelope>
```





# TRANSPORT HTTP SOAPACTION

- Encapçalament específic de SOAP quan utilitzem el protocol HTTP
- Format: SOAPAction: [ <"> URI <"> ]
  - Exemples:
    - SOAPAction: "http://electrocommerce.org/abc#MyMessage"
    - SOAPAction: "myapp.sdl"
    - SOAPAction: ""
    - SOAPAction:
    - ...
- L'encapçalament SOAPAction permet als tallafocs fer un processament condicional del missatges



# ENVIAMENT DEL MISSATGES CODIS D'ESTAT

- HTTP retorna informació referent a l'estat mitjançant codis
  - Ex: 200-299: codis indicatius d'èxit
  - El codi 500 és específic de SOAP



# ENVIAMENT DEL MISSATGE CODIS D'ESTAT

- Resposta amb èxit a la crida GetSecretIdentity

**HTTP/1.1 200 OK**

**Content-Type: text/xml**

**Content-Length: ###**

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  soap:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/1999/XMLSchema">
  <soap:Body>
    <m:GetSecretIdentityResponse xmlns:m="http://www.wrox.com/heroes/">
      <return xsi:type="xsd:string">Michael Kay</return>
    </m:GetSecretIdentityResponse>
  </soap:Body>
</soap:Envelope>
```



# LA CRIDA SERIA

**POST /endpoint.asp HTTP/1.1**  
**Content-Type: text/xml**  
**Content-Length: ###**  
**SOAPAction: "urn:wroxheroes"**

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  soap:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <soap:Header>
    <h:from xmlns:h="http://www.wrox.com/Header">
      SoapGuy@wrox.com
    </h:from>
  </soap:Header>
  <soap:Body>
    <w:GetSecretIdentity xmlns:w="http://www.wrox.com/heroes/">
      <codename>XSLT-Man</codename>
    </w:GetSecretIdentity>
  </soap:Body>
</soap:Envelope>
```



# SI HAGUÉSSIM ENVIAT AQUEST MISSATGE...

```
POST /endpoint.asp HTTP/1.1
Content-Type: text/xml
Content-Length: ###
SOAPAction: "urn:wroxheroes"

<Envelope
  <Body>
    <w:GetSecretIdentity xmlns:w="http://www.wrox.com/heroes/">
      <codename>XSLT-Man</codename>
    </w:GetSecretIdentity>
  </Body>
</Envelope>
```

**Fixem-nos que no hi ha l'espai de noms SOAP i per tant el punt extrem ha de retornar un error**



# MISSATGE D'ERROR

**HTTP/1.1 500 Internal Server Error**

**Content-Type: text/xml**

**Content-Length: ###**

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  soap:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/1999/XMLSchema">
  <soap:Body>
    <soap:Fault>
      <faultcode>soap:VersionMismatch</faultcode>
      <faultstring>The SOAP namespace is incorrect.</faultstring>
      <faultactor>http://www.wrox.com/endpoint.asp</faultactor>
      <detail>
        <w:errorinfo xmlns:w="http://www.wrox.com/">
          <desc>The SOAP namespace was blank.</desc>
        </w:errorinfo>
      </detail>
    </soap:Fault>
  </soap:Body>
</soap:Envelope>
```



# PROVANT SERVEIS WEB

ACTIVITAT 2.2

- Servei de localització geogràfica de adreces IP:
  - <http://www.websvcicex.net/geoip/service.asmx?op=GetGeoIP>

- **Probar-lo:**

Parameter	Value
IPAddress:	<input type="text"/>
<input type="button" value="Invoke"/>	

- El missatge corresponent amb SOAP/HTTP:

```
POST /geoip/service.asmx HTTP/1.1
Host: www.websvcicex.net
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://www.websvcicex.net/GetGeoIP"
```

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap=http://
schemas.xmlsoap.org/soap/envelope/>
<soap:Body>
  <GetGeoIP xmlns="http://www.websvcicex.net">
    <IPAddress>string</IPAddress>
  </GetGeoIP>
</soap:Body>
</soap:Envelope>
```



# PROVANT SERVEIS WEB

ACTIVITAT 2.2

- La web anterior també mostra exemples de com cridar serveis web sense SOAP

- Mitjançant HTTP Post
- Mitjançant HTTP GET

- Tan simple com la URL:

`http://www.websvcex.net/geoip/service.asmx/GetGeoIP?IPAddress=[ADREÇAIP]`

- Que es tradueix en el següent paquet HTTP:

`GET /geoip/service.asmx/GetGeoIP?IPAddress=[ADREÇAIP] HTTP/1.1`

`Host: www.websvcex.net`

- Triar i provar més serveis a:  
<http://www.websvcex.net>

