

Çfarë janë Shërbimet Ueb

- Shërbimet Ueb janë një teknologji e shpërndarë, që realizojnë ndërveprim dhe bashkëpunim mes ofruesve të shërbimit dhe klientëve.
- Shërbimet Ueb janë njësi të vogla të kodit, secila me numër të caktuar detyrash.
- Shërbimet Ueb përdorin protokollet komunikuese të bazuara në XML.
- Shërbimet Ueb janë të pavarura prej sistemeve operative dhe gjuhëve programuese.
- Shërbimet Ueb mundësojnë që aplikacionet të shkëmbejnë të dhëna.
- Shërbimet Ueb janë një teknologji e re që mundësojnë qasjen e funksioneve në largësi përmes internetit.
- Shërbimi Ueb është një ndërsaqe e vendosur ndërmjet kodit të aplikacionit dhe përdoruesit të atij kodi.

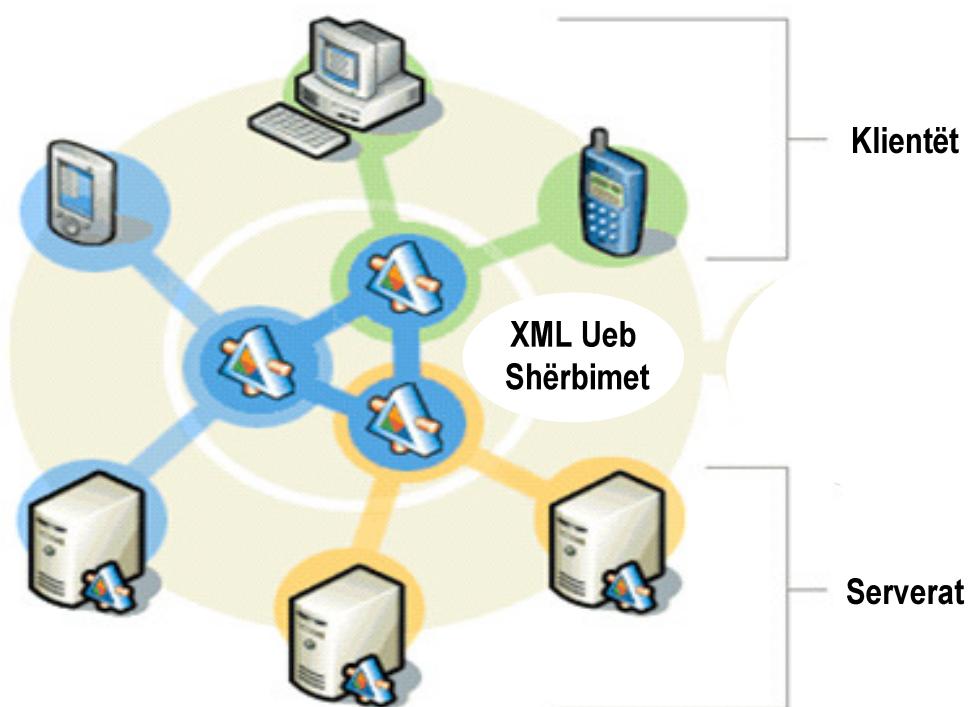


Fig.1. Arkitektura funksionale e Shërbimeve Ueb.



Koha e përgjigjes në funksion të ngarkesës

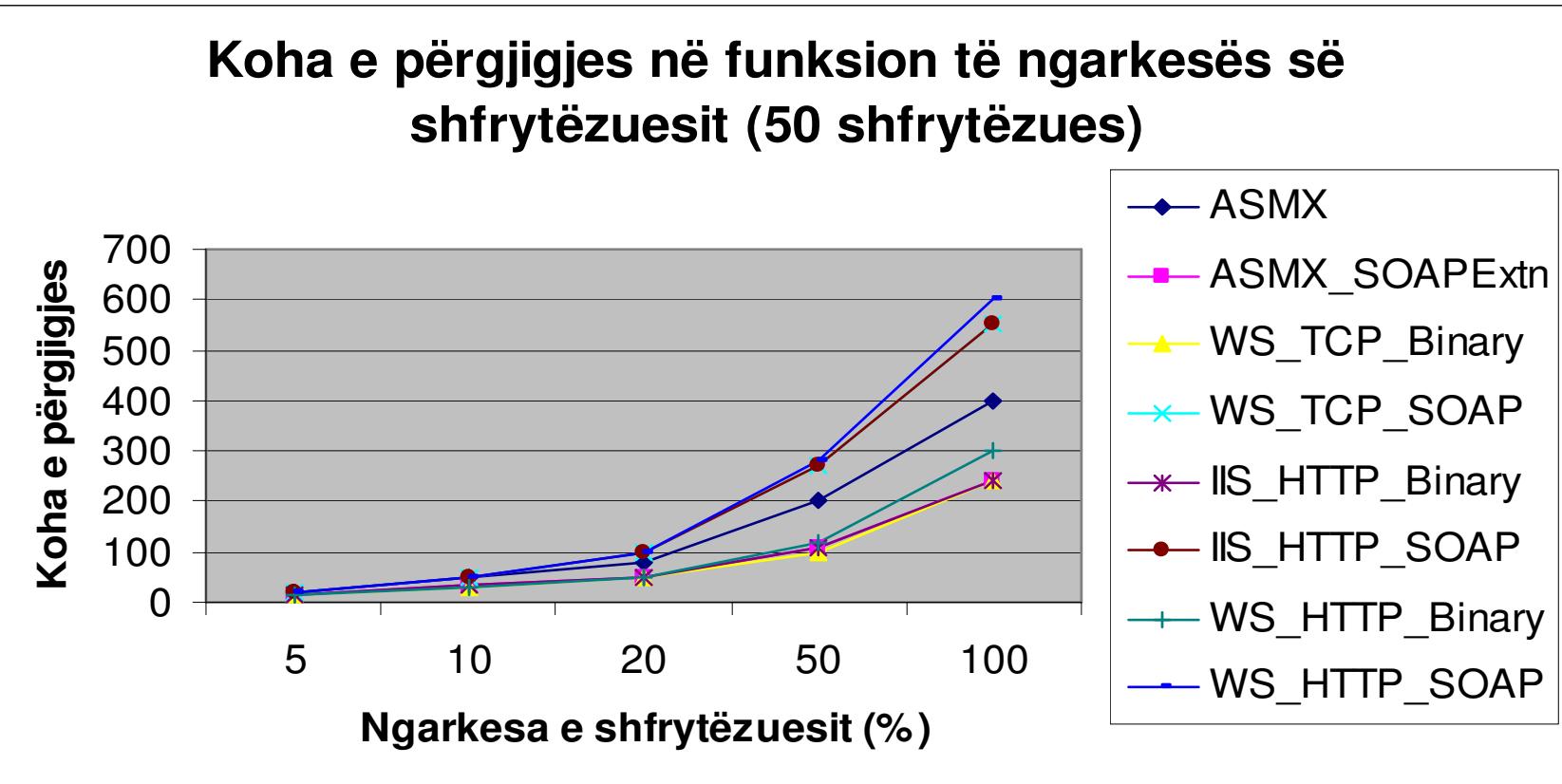


Fig.9. Koha e përgjigjes në funksion të ngarkesës së shfrytëzuesit

Modeli i sigurisë së Shërbimeve Ueb

- **Niveli i sigurisë platformë/transport**, ky nivel i sigurisë arrihet duke ofruar siguri pikë-pikë, ku kanali transportues në mes dy pikave te skajshme: Klientit të Shërbimit Ueb dhe Shërbimit Ueb mund te përdoret për te ofruar këtë siguri.
- **Niveli i sigurisë së aplikacionit**, Me këtë siguri, aplikacioni merr sigurinë dhe përdor veçoritë e saj te përshtatur . Për shembull:
 - Një aplikacion mund te përdor kokën e përshtatur SOAP për te kaluar kredencialet e përdoruesit për ta njojur përdoruesin me të cilin secili Shërbim Ueb kërkohet. Zgjidhje e thjeshte është te kalohet me tikete (apo emër te përdoruesit ose licencë) në kokën SOAP.
 - Aplikacioni ka fleksibilitet për te gjeneruar objektin e vet që i përmban këto rolet.
 - Aplikacioni ka mundësi qe ndaras te enkriptoj çka dëshiron të bëj, edhe pse kjo kërkon memorie të siguruar me çelës dhe zhvilluesit duhet te kenë dituri për kriptografinë relevante të API.
 - Një teknike alternative është te përdoret SSL për te mundësuar konfidencialitet dhe integritet dhe për te kombinuar atë me kokën e përshtatur SOAP për te kryer autentifikimin.
- **Niveli i sigurisë së Mesazhit**, Kjo paraqet mundësinë me fleksibile dhe me te fuqishme për sigurinë e Shërbimeve Ueb.

Tregu i Shërbimeve Ueb

- Sipas një analize të bërë nga komiteti i IDC (International Data Corporation), në vitin 2003 janë shpenzuar afro 0.88 miliardë Euro, ndersa deri në vitin 2009 tregu i Shërbimeve Ueb ka arritu shifra rreth 11.92 miliard eoro.
- Në figurën 12 është paraqitur një diagram ku shihet evoluimi i Shërbimeve Ueb në vite.

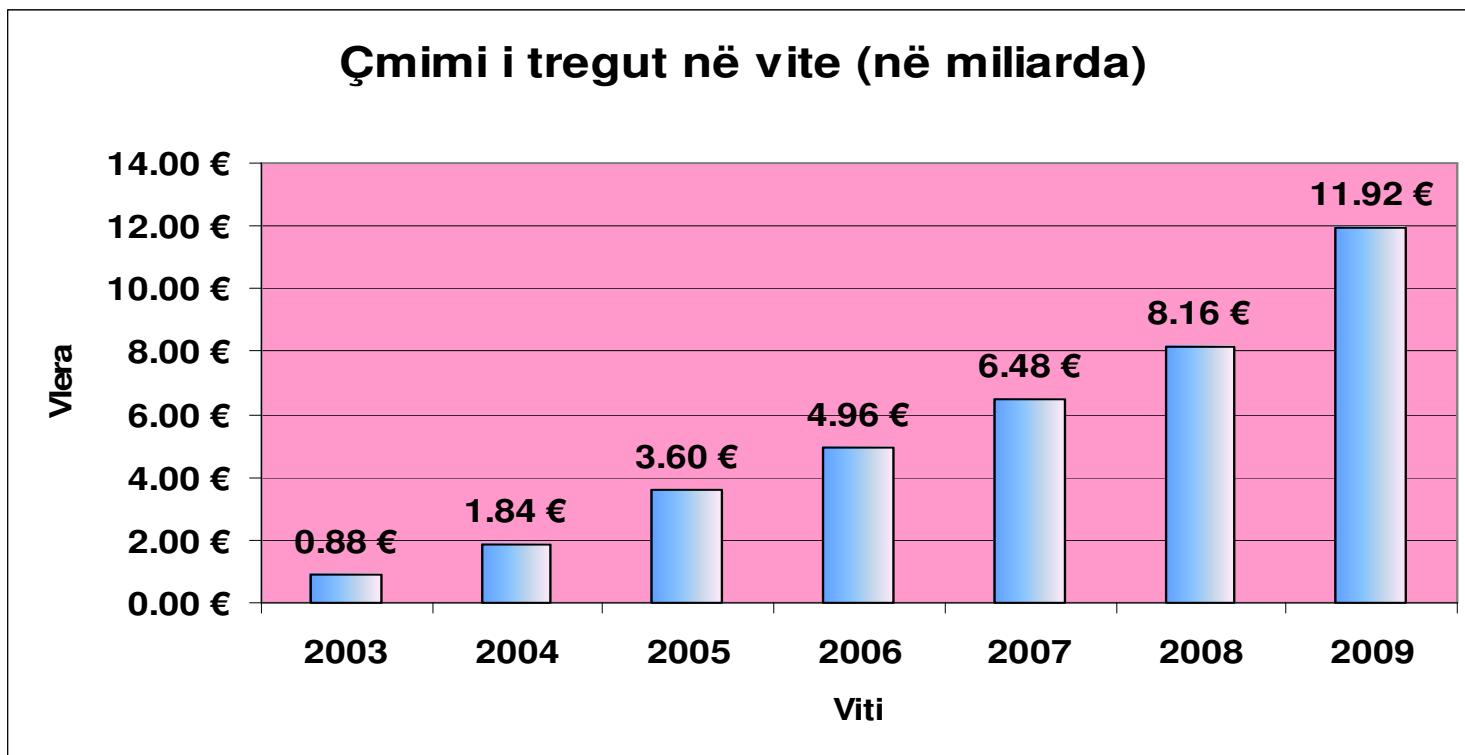
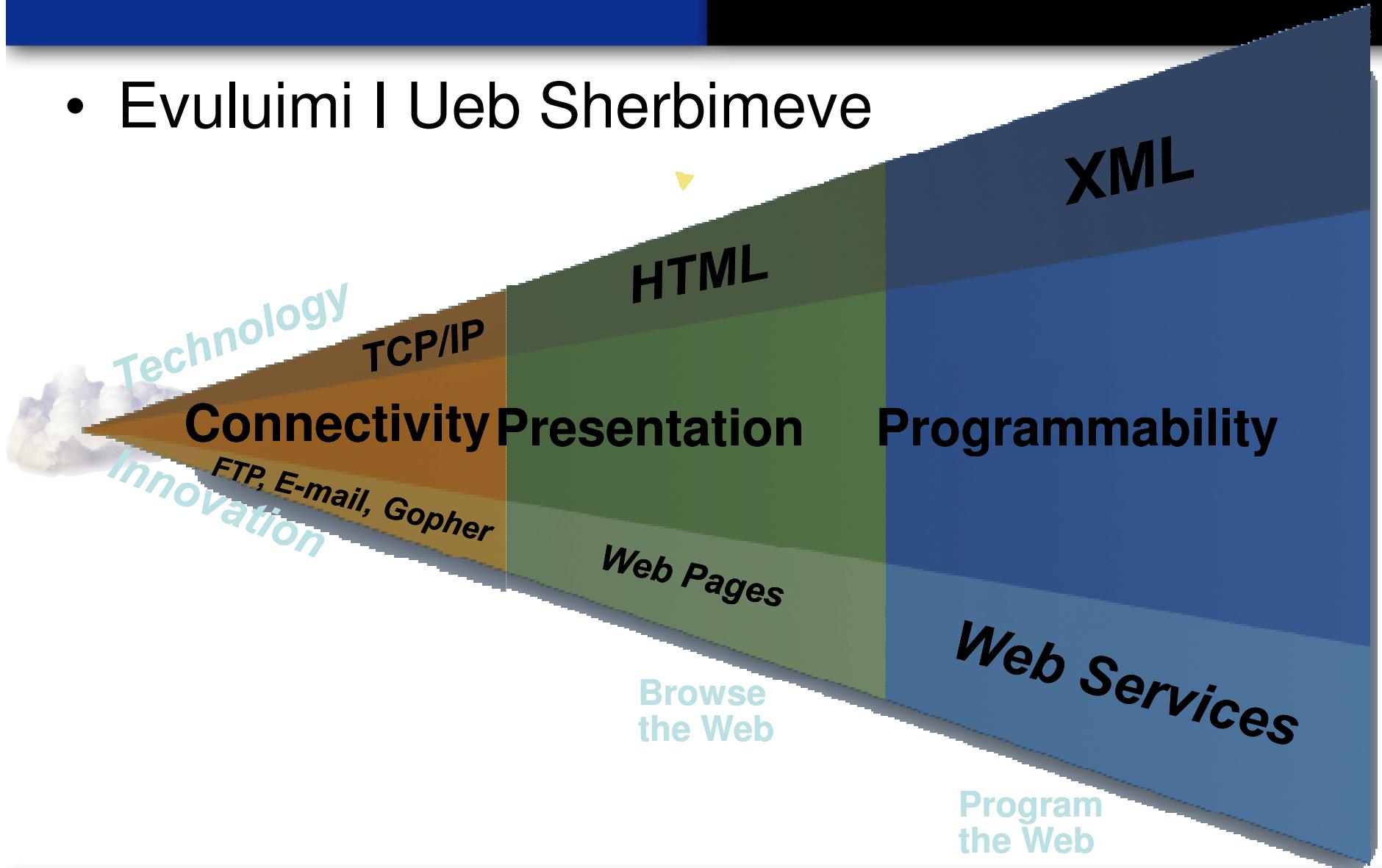


Fig.10. Tregu i ueb shërbimeve 2003-2009

- Evoluimi I Ueb Sherbimeve



Komponentët e Shërbimeve Ueb

- UDDI – Siguron mekanizmin për të regjistruar dhe kategorizuar Shërbimet Ueb që ofrohen si dhe për të gjetur Ueb shërbimet të cilat do të përdoren.
- WSDL – Bazohet në XML për përshkrimin e Ueb Shërbimeve. Përshkruan çfarë funksionaliteti ofron Shërbimi Ueb, mënyrën e komunikimit dhe ku mund të qaset.
- SOAP – Është protokoll i thjeshtë i bazuar në XML që i mundëson aplikacioneve shkëmbim të informatave përmes HTTP.
- XML - Paraqet formatin e shënimive që barten prej një aplikacioni në tjetrin.



Fig.2. Komponentet e Shërbimeve Ueb.

Çka është SOAP ?

- SOAP është protokoll për komunikim ndërmjet aplikacioneve.
- SOAP bazohet në XML dhe ju mundëson aplikacioneve që të komunikojnë edhe përmes internetit.
- SOAP në teknologjinë e Shërbimeve Ueb qëndron si protokoll i standardizuar për paketimin e porosive të cilat komunikojnë dhe përdoren nga aplikacionet.
- SOAP përmban **zarfin** që përbëhet nga **koka** dhe **trupi**.
- Zarfi SOAP** përshkruan përbajtjen e mesazhit dhe mënyrën e përpunimit dhe përmban informacionet për destinacionin final të mesazhit.
- Koka** përmban të dhëna shtesë rreth mesazhit por vetë nuk është pjesë e mesazhit. Për shembull koka mund të përbajë:
 - Informacionin për shfrytëzuesin që është duke e bërë kërkesën, që përfshinë ID-në e shfrytëzuesit, fjalëkalimin, certifikimin X.509, ose të dhënat e tjera identifikuuese.
 - Informacionin rreth transaksionit, gjendjes dhe rrugëtimit që tregon se kush e ka përpunuuar mesazhin SOAP
- Trupi** përmban ngarkesën fitimprurëse të mesazhit dhe një seksion që është opcional që është gabimi SOAP që përmban informacionin rreth gabimeve eventuale ose të gjendjeve.



Fig.3. Protokoli SOAP.

Çka është WSDL ?

- WSDL bazohet në gjuhë XML për përshkrimin e Shërbimeve Ueb.
- Për të kriuar një Shërbim Ueb duhet të krijohet një dokument WSDL i cili e përshkruan shërbimin e tillë. Dokumenti do të vendoset në server apo publikohet në regjistrin UDDI.
- Klienti i interesuar në këtë shërbim së pari merr një kopje apo referencë të këtij dokumenti duke e kërkuar regjistrin, mandej e kuption kontratën, krijon kërkesën SOAP duke u bazuar në kontratë dhe të njëjtën e dërgon në server.

Elementet e WSDL – së:

- definitions**
 - Definon emrin e shërbimit
 - Deklaron namespa-sat që shfrytëzohen në dokument
- types**
 - Përshkruan tipet e të dhënavë që shfrytëzohen nga Klienti dhe Serveri
- message**
 - Definon emrin mesazhit request/response
 - Definon gjithashtu elementet e mesazhit
- portType**
 - Definon kombinimin e elementeve të mesazhit të një forme me komplet operacionet
- binding**
 - Përfshin specifikat detale se si janë të lidhura mesazhet që transmetohen
- service**
 - Definon adresën ku është i vendosur shërbimi

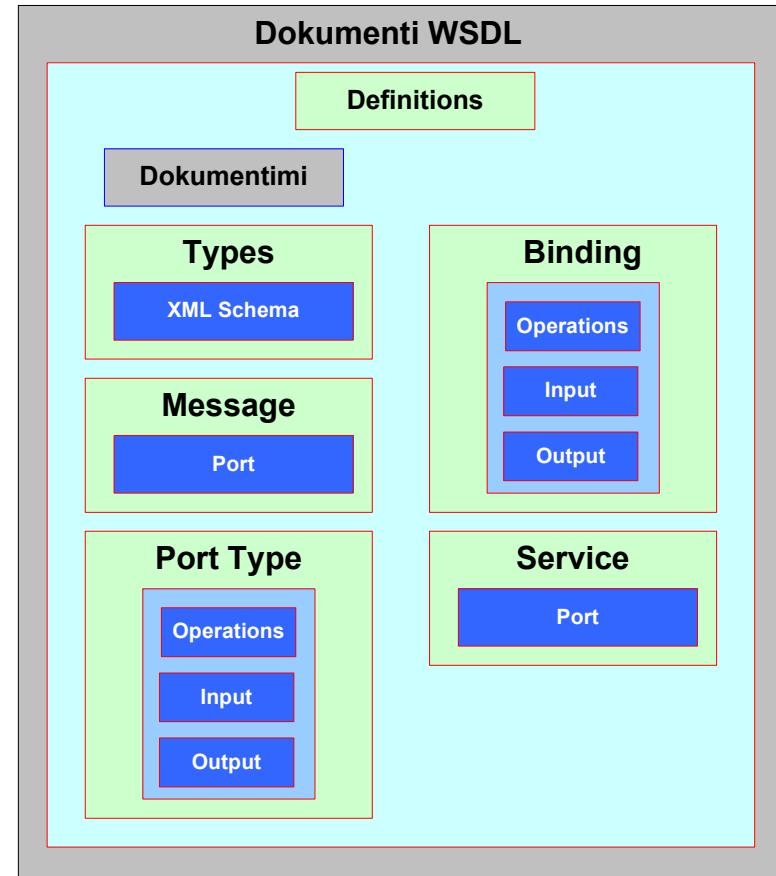


Fig.4. Dokumenti WSDL.

Çka është UDDI ?

- UDDI ka filluar si bashkëpunim ndërmjet Microsoft-it, IBM-it dhe Aribas për të ndihmuar adaptimin dhe përdorimin e standardeve të Shërbimeve Ueb.
- UDDI është një regjistër i cili shërben për ruajtjen dhe gjetjen e informacioneve të Shërbimeve Ueb.
- Regjistri publik i UDDI-së punon në mënyrë të ngjashme sikur edhe DNS (*Internet Domain Name Service*).
- Të dhënata e UDDI-së ndahen në tri kategori kryesore:
 - Faqet e bardha:** Emri i biznesit, adresa, të dhëna kontaktuese, Emri i Ueb fakes dhe numëri identifikues.
 - Faqet e verdha:** Lloji i biznesit, lokalizimi dhe produktet, duke përfshirë lokalizimin e biznesit, llojin e industrisë, numrin identifikues të biznesit, etj.
 - Faqet e gjelbërta:** Të dhënata teknike për shërbimet e biznesit, si p.sh. si të bashkëveprojmë me to, definime të proceseve të biznesit, etj.

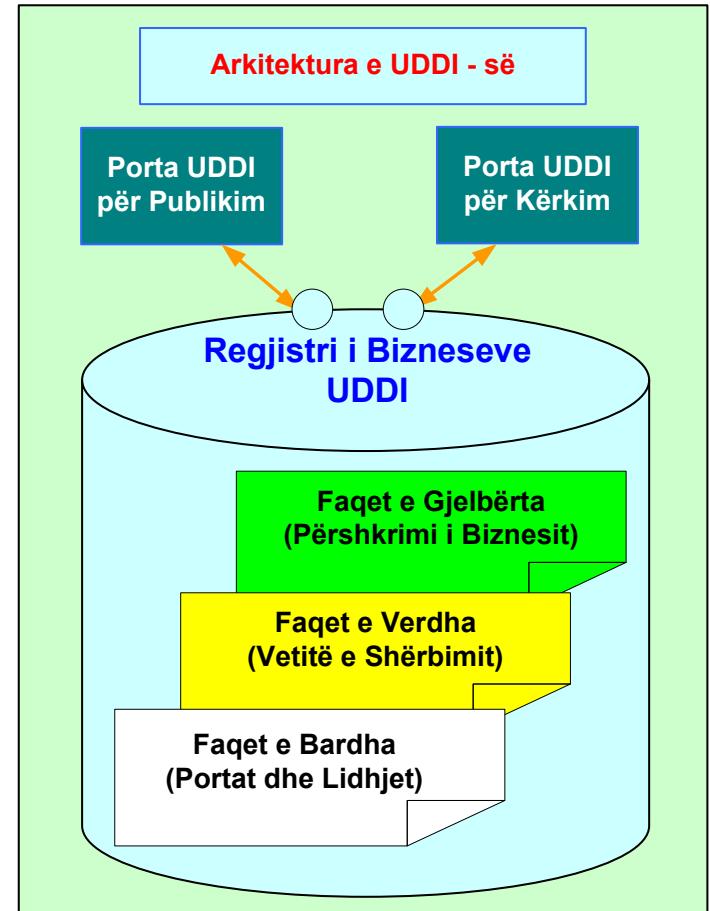


Fig.5. Arkitektura e UDDI-së.



Modeli i Shërbimet Ueb (Illustrim)



Modeli i Shërbimeve Ueb ndjek këto hapa:

- Publikon
- Gjen dhe
- Lidhë

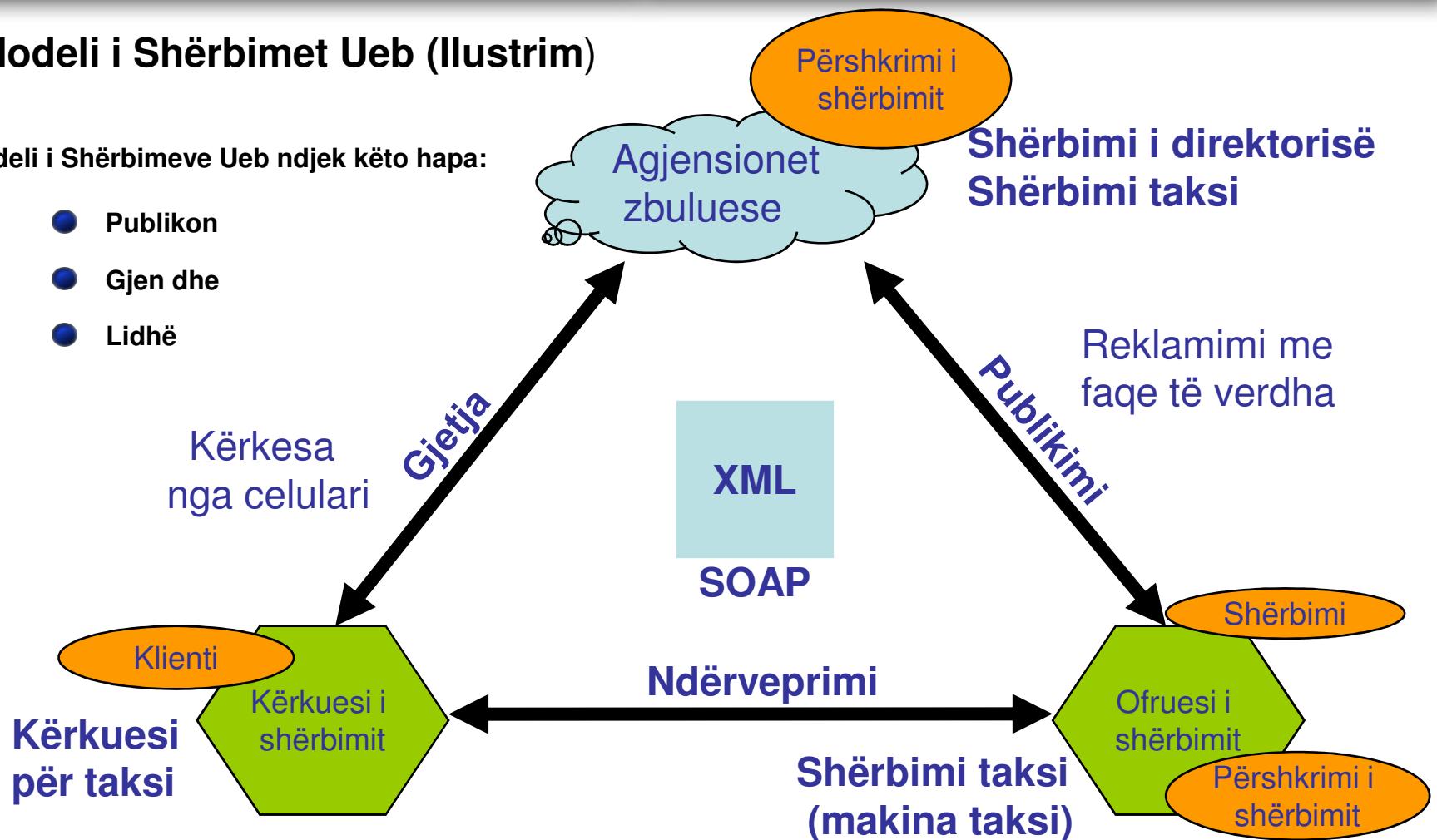


Fig.6. Modeli i shërbimeve Ueb



Arkitektura e kërkesës të një Shërbimi Ueb

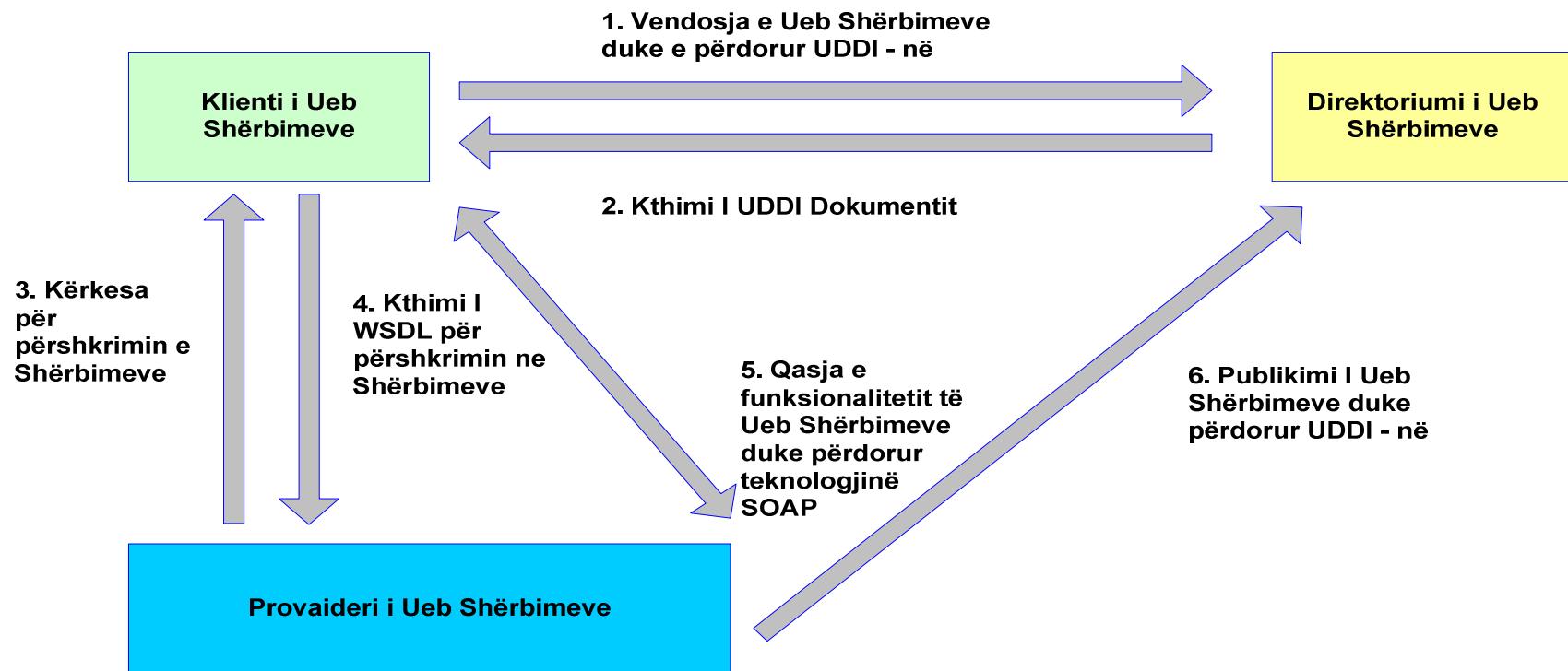


Fig.7. Arkitektura e Shërbimeve Ueb.

Zbatimi dhe ndërlidhja e Shërbimeve Ueb

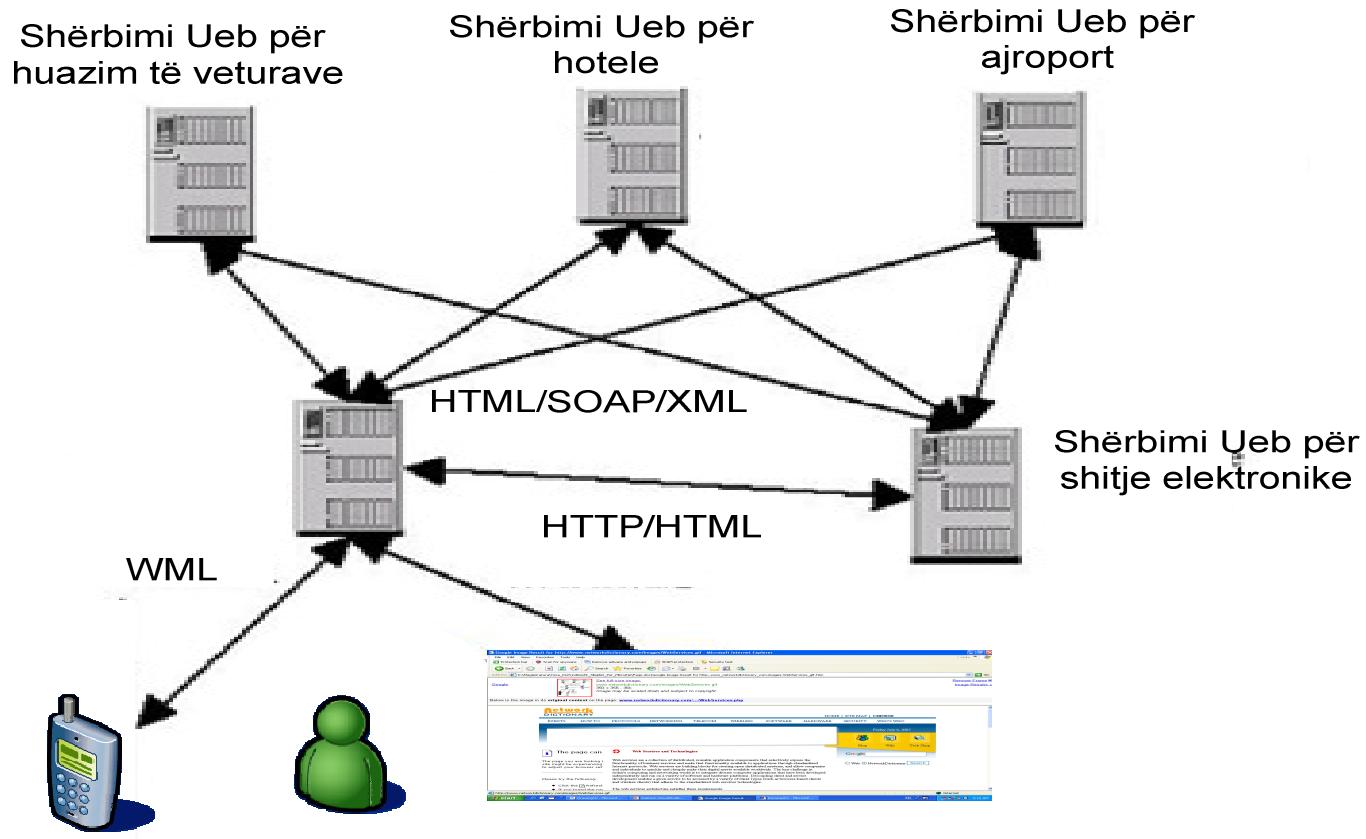


Fig.8. Zbatimi dhe ndërlidhja e Shërbimeve Ueb



eXtensible Markup Language **(XML)**

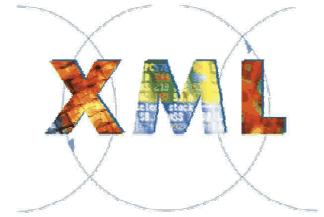
Prof. Ass. Dr. Isak Shabani



What is XML?

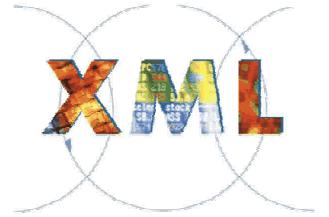
- **eXtensible Markup Language**
- **Markup language for documents containing structured information**
- **Defined by four specifications:**
 - XML, the Extensible Markup Language
 - XLL, the Extensible Linking Language
 - XSL, the Extensible Style Language
 - XUA, the XML User Agent

XML is not...



- **A replacement for HTML**
(but HTML can be generated from XML)
- **A presentation format**
(but XML can be converted into one)
- **A programming language**
(but it can be used with almost any language)
- **A network transfer protocol**
(but XML may be transferred over a network)
- **A database**
(but XML may be stored into a database)

But then – what is it?



**XML is a meta markup language for
text documents / textual data**

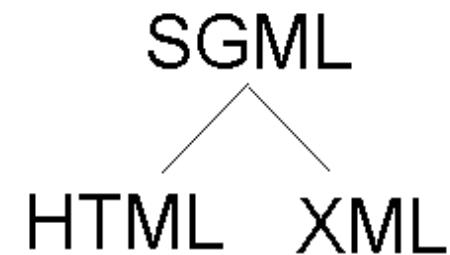


**XML allows to define languages
(„applications“) to represent text
documents / textual data**

XML....



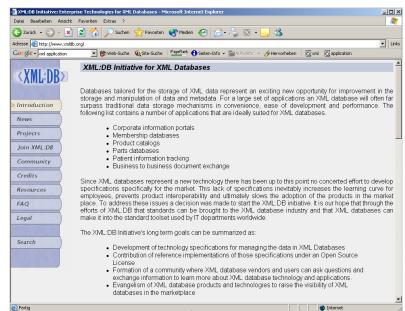
- Based on Standard Generalized Markup Language (SGML)
- Version 1.0 introduced by World Wide Web Consortium (W3C) in 1998
- Bridge for data exchange on the Web



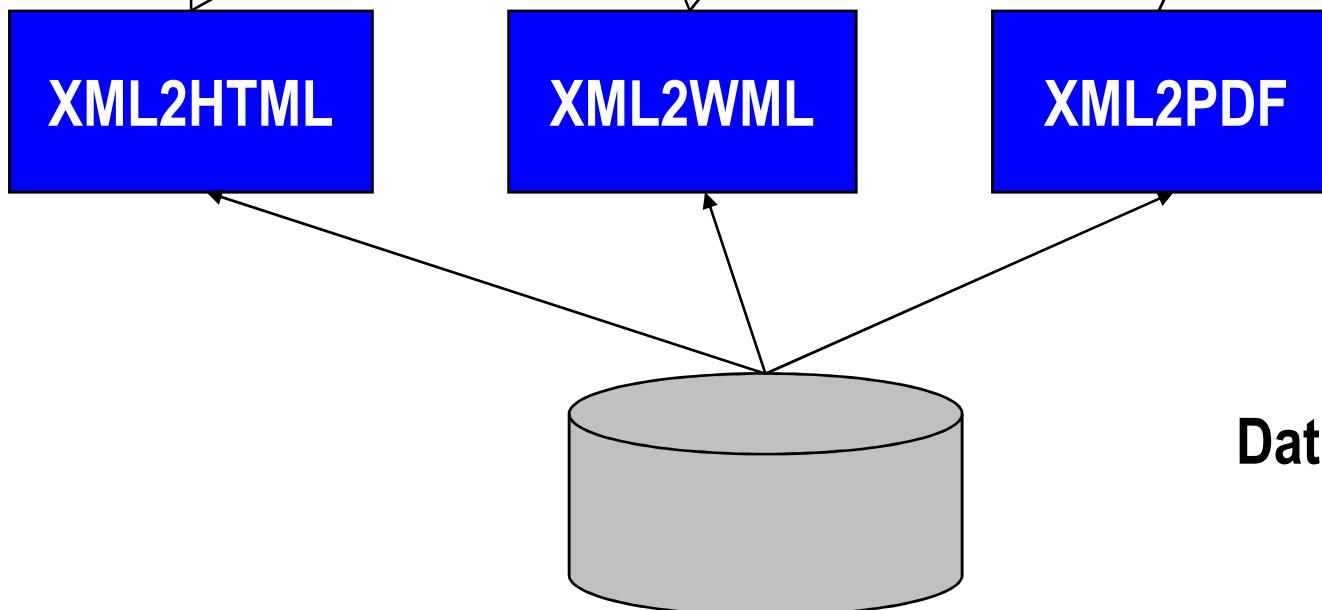
App. Scenario 1: Content Mgt.



Clients



Converters



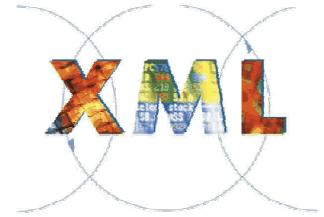
Database with XML
documents

XML Standards – an Overview



- XML Core Working Group:
 - XML 1.0 (Feb 1998), 1.1 (candidate for recommendation)
 - XML Namespaces (Jan 1999)
 - XML Inclusion (candidate for recommendation)
- XSLT Working Group:
 - XSL Transformations 1.0 (Nov 1999), 2.0 planned
 - XPath 1.0 (Nov 1999), 2.0 planned
 - eXtensible Stylesheet Language XSL(-FO) 1.0 (Oct 2001)
- XML Linking Working Group:
 - XLink 1.0 (Jun 2001)
 - XPointer 1.0 (March 2003, 3 substandards)
- XQuery 1.0 (Nov 2002) plus many substandards
- XMLSchema 1.0 (May 2001)
- ...

Comparisons



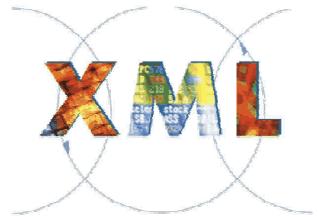
XML

- Extensible set of tags
- Content orientated
- Standard Data infrastructure
- Allows multiple output forms

HTML

- Fixed set of tags
- Presentation oriented
- No data validation capabilities
- Single presentation

XML Documents



What's in an XML document?

- Elements
- Attributes
- plus some other details

A Simple XML Document



```
<article>
  <author>Gerhard Weikum</author>
  <title>The Web in Ten Years</title>
  <text>
    <abstract>In order to evolve...</abstract>
    <section number="1" title="Introduction">
      The <index>Web</index> provides the universal...
    </section>
  </text>
</article>
```

Start Tag

<text>

<abstract>In order to evolve...</abstract>
<section number="1" title="Introduction">
 The <index>Web</index> provides the universal...

</text>

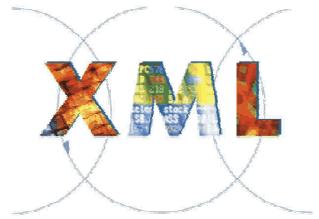
</article>

End Tag

Element

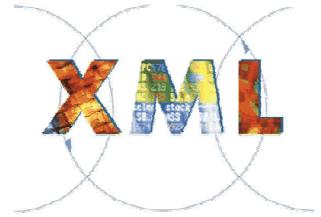
Content of the
Element
(Subelements
and/or Text)

A Simple XML Document



```
<article>
  <author>Gerhard Weikum</author>
  <title>The Web in Ten Years</title>
  <text>
    <abstract>In order to evolve...</abstract>
    <section number="1" title="Introduction">
      The <index>Web</index> provides the universal...
    </section>
  </text>
</article>
```

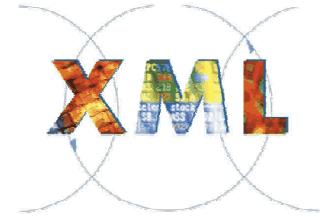
Attributes with
name and value



Authoring XML Elements

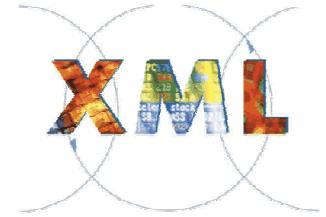
- An XML element is made up of a start tag, an end tag, and data in between.
- Example:
`<director> Matthew Dunn </director>`
- Example of another element with the same value:
`<actor> Matthew Dunn </actor>`
- XML tags are case-sensitive:
`<CITY> <City> <city>`
- XML can abbreviate empty elements, for example:
`<married></married>` can be abbreviated to
`<married/>`

Authoring XML Elements (cont'd)



- An attribute is a name-value pair separated by an equal sign (=).
- Example:
`<City ZIP="94608"> Emeryville </City>`
- Attributes are used to attach additional, secondary information to an element.

Authoring XML Documents



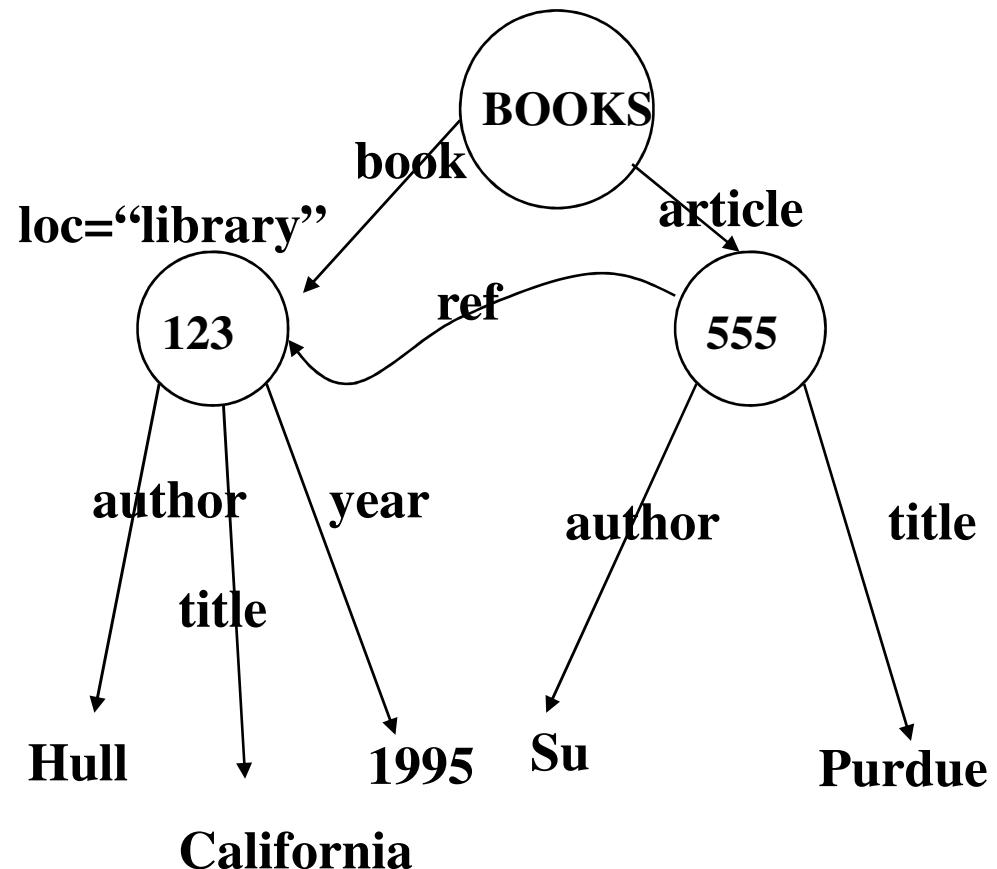
- A basic XML document is an XML element that can, but might not, include nested XML elements.
- Example:

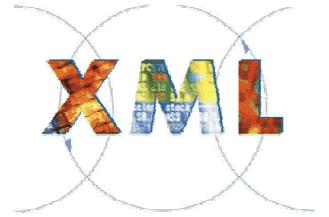
```
<books>
  <book isbn="123">
    <title> Second Chance </title>
    <author> Matthew Dunn </author>
  </book>
</books>
```

XML Data Model: Example



```
<BOOKS>
<book id="123"
      loc="library">
    <author>Hull</author>
    <title>California</title>
    <year> 1995 </year>
  </book>
  <article id="555" ref="123">
    <author>Su</author>
    <title> Purdue</title>
  </article>
</BOOKS>
```



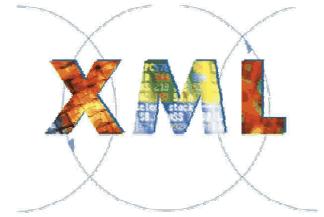


Authoring XML Data

- Example:

```
<XML ID="XMLID">  
  <customer>  
    <name> Mark Hanson </name>  
    <custID> 29085 </custID>  
  </customer>  
</XML>
```

Document Type Definitions (DTD)



- An XML document may have an optional DTD.
- DTD serves as grammar for the underlying XML document, and it is part of XML language.
- DTDs are somewhat unsatisfactory, but no consensus exists so far beyond the basic DTDs.
- DTD has the form:
`<!DOCTYPE name [markupdeclaration]>`



DTD (cont'd)

- Consider an XML document:

```
<db><person>  
    <name>Alan</name>  
    <age>42</age>  
    <email>agb@usa.net</email>  
  </person>  
  <person>.....</person>  
  .....  
</db>
```

- DTD for it might be:

```
<!DOCTYPE db [  
    <!ELEMENT db (person*)>  
    <!ELEMENT person (name,  
        age, email)>  
    <!ELEMENT name (#PCDATA)>  
    <!ELEMENT age (#PCDATA)>  
    <!ELEMENT email (#PCDATA)>  
>]
```

DTD Example: Elements



```
<!ELEMENT article      (title,author+,text)>
<!ELEMENT title        (#PCDATA)>
<!ELEMENT author       (#PCDATA)>
<!ELEMENT text         (abstract,section*,literature?)>
<!ELEMENT abstract     (#PCDATA)>
<!ELEMENT section      (#PCDATA|index)+>
<!ELEMENT literature   (#PCDATA)>
<!ELEMENT index        (#PCDATA)>
```

Content of the `title` element is parsed character data

Content of the `text` element may contain zero or more `section` elements in this position

Content of the `article` element is a `title` element, followed by one or more `author` elements, followed by a `text` element

Attribute Declarations in DTDs



Attributes are declared per element:

```
<!ATTLIST section number CDATA #REQUIRED
```

declares two required attributes for element

element name

attribute name

attribute type

attribute default

Attribute Declarations in DTDs



Attributes are declared per element:

```
<!ATTLIST section number CDATA #REQUIRED  
                  title  CDATA #REQUIRED>
```

declares two required attributes for element `section`.

Possible attribute defaults:

- `#REQUIRED` is required in each element instance
- `#IMPLIED` is optional
- `#FIXED default` always has this default value
- `default` has this default value if the attribute is omitted from the element instance

Attribute Types in DTDs



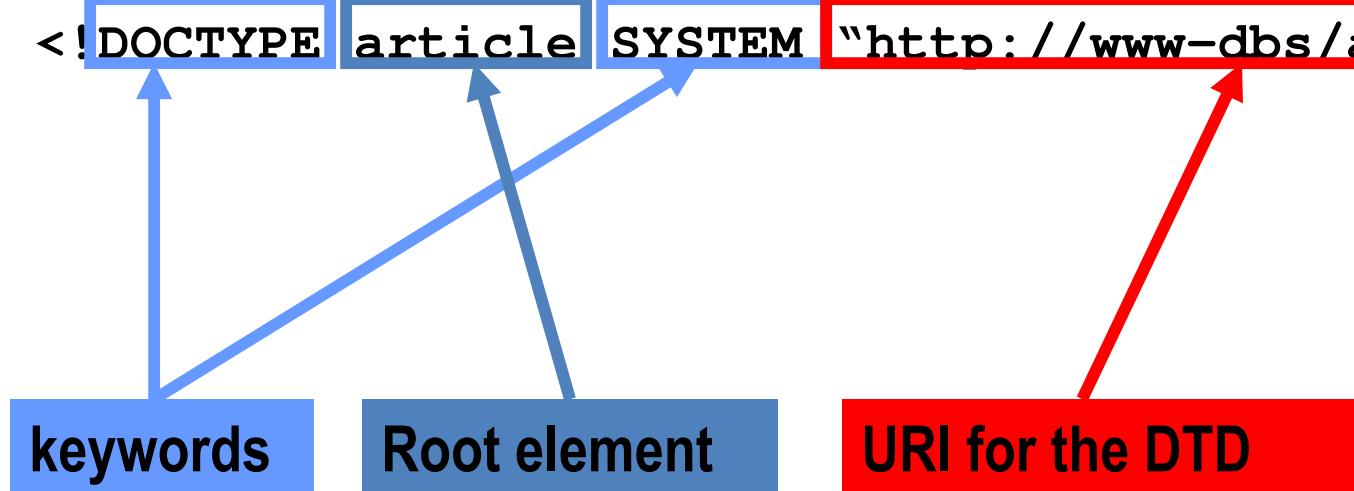
- **CDATA** string data
- **(A1 | ... | An)** enumeration of all possible values of the attribute (each is XML name)
- **ID** unique XML name to identify the element
- **IDREF** refers to **ID** attribute of some other element („intra-document link“)
- **IDREFS** list of **IDREF**, separated by white space
- plus some more

Linking DTD and XML Docs

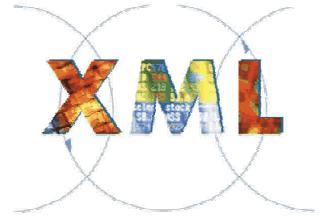


- Document Type Declaration in the XML document:

```
<!DOCTYPE article SYSTEM "http://www-dbs/article.dtd">
```



Linking DTD and XML Docs



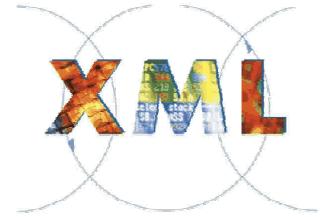
- Internal DTD:

```
<?xml version="1.0"?>
<!DOCTYPE article [
    <!ELEMENT article (title,author+,text)>
    ...
    <!ELEMENT index (#PCDATA)>
]>
<article>
...
</article>
```

- Both ways can be mixed, internal DTD overwrites external entity information:

```
<!DOCTYPE article SYSTEM „article.dtd“ [
    <!ENTITY % pub_content (title+,author*,text)
]>
```

XML Query Languages



- The first XML query languages
 - LOREL (Stanford)
 - XQL
- Several other query languages have been developed (e.g. UNQL, XPath)
- XML-QL considered by W3C for standardization
- Currently W3C is considering and working on a new query language: XQuery

A Query Language for XML: XML-QL



- Developed at AT&T labs
- To extract data from the input XML data
- Has variables to which data is bound and templates which show how the output XML data is to be constructed
- Uses the XML syntax
- Based on a where/construct syntax
 - **Where** combines **from** and **where** parts of SQL
 - **Construct** corresponds to SQL's **select**



XML-QL Query: Example 1

- Retrieve all authors of books published by Morgan Kaufmann:

```
where <book>
    <publisher><name>
        Morgan Kaufmann
    </name> </publisher>
    <title> $T </title>
    <author> $A </author>
</book> in "www.a.b.c/bib.xml"
construct <result> $A </result>
```

XML-QL Query: Example 2



- XML-QL query asking for all bookstores that sell *The Java Programming Language* for under \$25:

where <**store**>

```
<name> $N </name>
<book>
  <title> The Java Programming Language </title>
  <price> $P </price>
</book>
</store> in “www.store/bib.xml”
$P < 25
```

construct <**result**> \$N </result>

XQuery Examples



// find Web-related articles by Dan Suciu from the year 2014

```
<results> {
FOR $a IN document("literature.xml")//article
  FOR $n IN $a//author, $t IN $a/title
  WHERE $a/@year = "2014"
    AND contains($n, "Suciu") AND contains($t, "Web")
RETURN <result> $n $t </result> } </results>
```

// find articles co-authored by authors who have jointly written a book after 2010

```
<results> {
FOR $a IN document("literature.xml")//article
  FOR $a1 IN $a//author, $a2 IN $a//author
  WHERE SOME $b IN document("literature.xml")//book SATISFIES
    $b//author = $a1 AND $b//author = $a2 AND $b/@year > "2010"
RETURN <result> $a1 $a2 <wrote> $a </wrote> </result> }
</results>
```

Converting Relational Database to XML

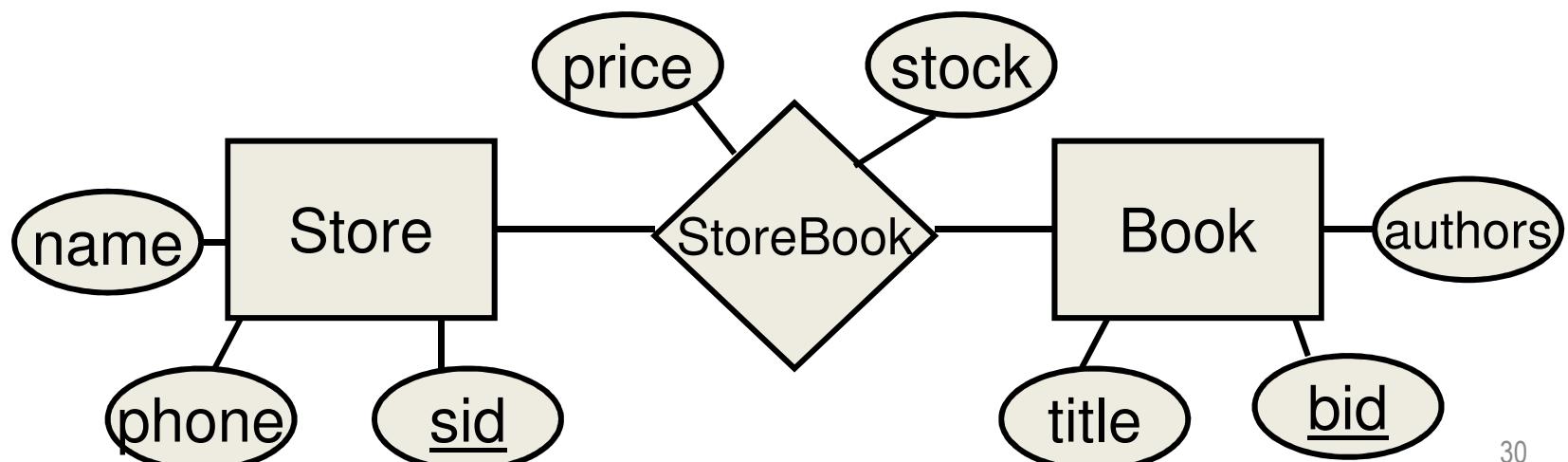


Example: Export the following data into XML and group books by store

- Relational Database:
Store (sid, name, phone)
Book (bid, title, authors)
StoreBook (sid, bid, price, stock)

XML:

```
<store> <name> ... </name>
      <phone> ... </phone>
      <book> <title>... </title>
            <authors> ... </authors>
            <price> ... </price>
      </book>
      <book>...</book>
...
</store>
```



Karakteristikat e Sistemeve të Shpërndara

Domene të aplikacioneve të zgjedhura dhe programet e shoqëruara në Ueb

Prof. Ass. Dr. Isak Shabani

<i>Financa dhe komerciale</i>	eCommerce e.g. Amazon dhe eBay, PayPal, bankat online dhe tregtia
<i>Shoqëria informative</i>	Informata nga web dhe makina kërkuëse, eBooks, Wikipedia; rrjete sociale: FaceBook dhe MySpace.
<i>Industritë kreative dhe argëtim</i>	Lojra online, muzikë dhe film në shtëpi, përbajtje e gjeneruar nga shfrytëzuesi, p.sh., YouTube, Flickr
<i>Kujdes Shëndetësor</i>	Informatika shëndetsore, në regjistrat online të pacientëve, monitorimi i pacientëve
<i>Edukim</i>	e-learning, ambiente virtuale mësimi; të mësuarit nga distanca
<i>Transporti dhe logjistika</i>	GPS në sistemetë për gjetjen e rrugëve, shërbime të hartave: Google Maps, Google Earth
<i>Shkencë</i>	Gridi (Grid Computers) si teknologji e cila mundëson bashkëpunimin në mes të shkencëtarëve
<i>Menaxhimi i ambientit</i>	Teknologjia e sensorëve për të monitoruar tërmetet, vërshimet ose dallgët (cunami).

Të mirat dhe të metat e Sistemeve të shpërndara

Prof. Ass. Dr. Isak Shabani

Të mirat

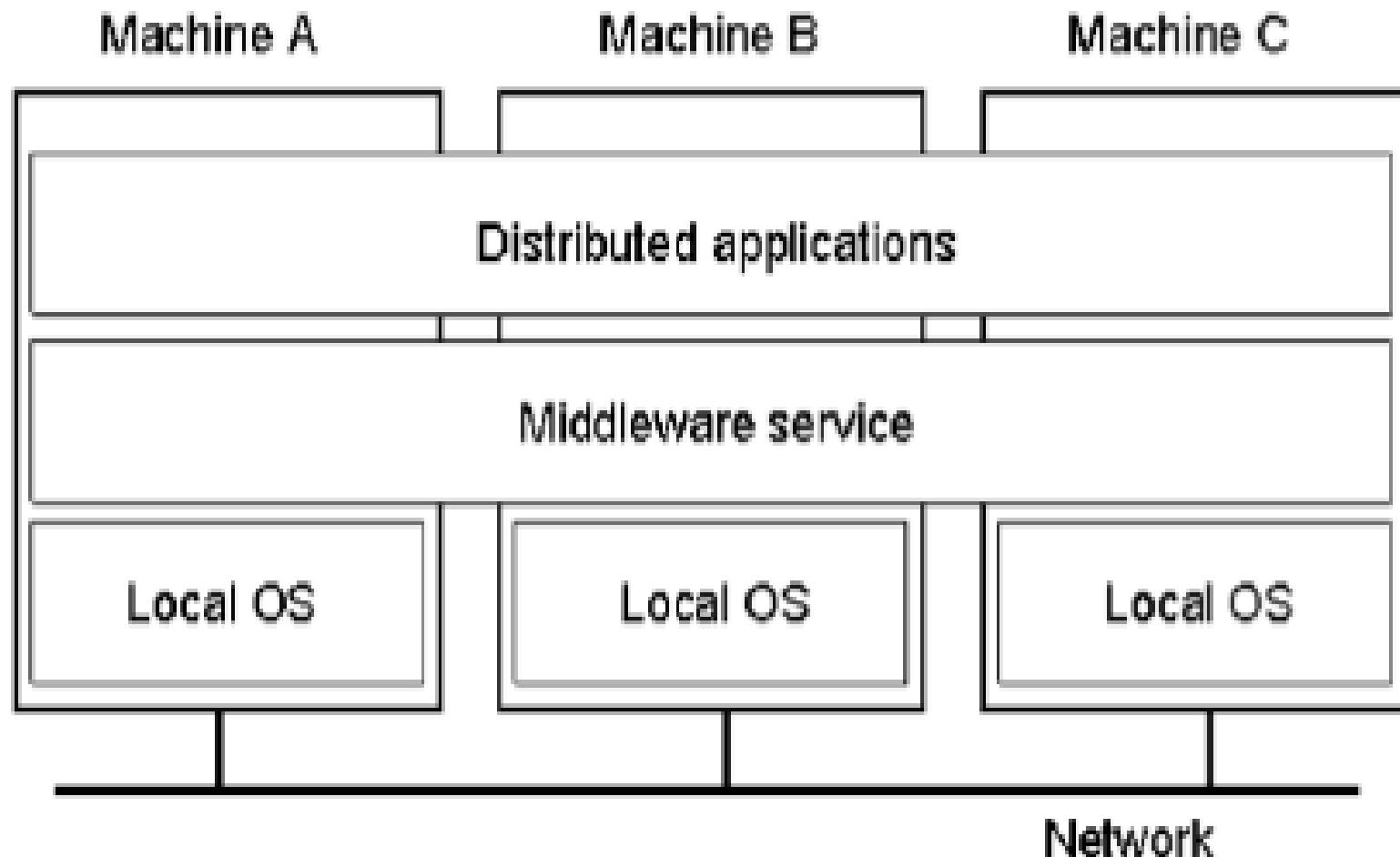
- Ekonomik
- Shpejtë
- Shpërndarje e përhershme (eng., inherent distribution)
- Qëndrueshëm
- Zgjerim

Të metat

- Softueri
- Rrjeta
- Më shumë komponente në amortizim
- Siguria

Organizimi i Sistemeve te Shperndara

Prof. Ass. Dr. Isak Shabani



Kushtet në të cilat punojnë Sistemet e Shpërdara

Prof. Ass. Dr. Isak Shabani

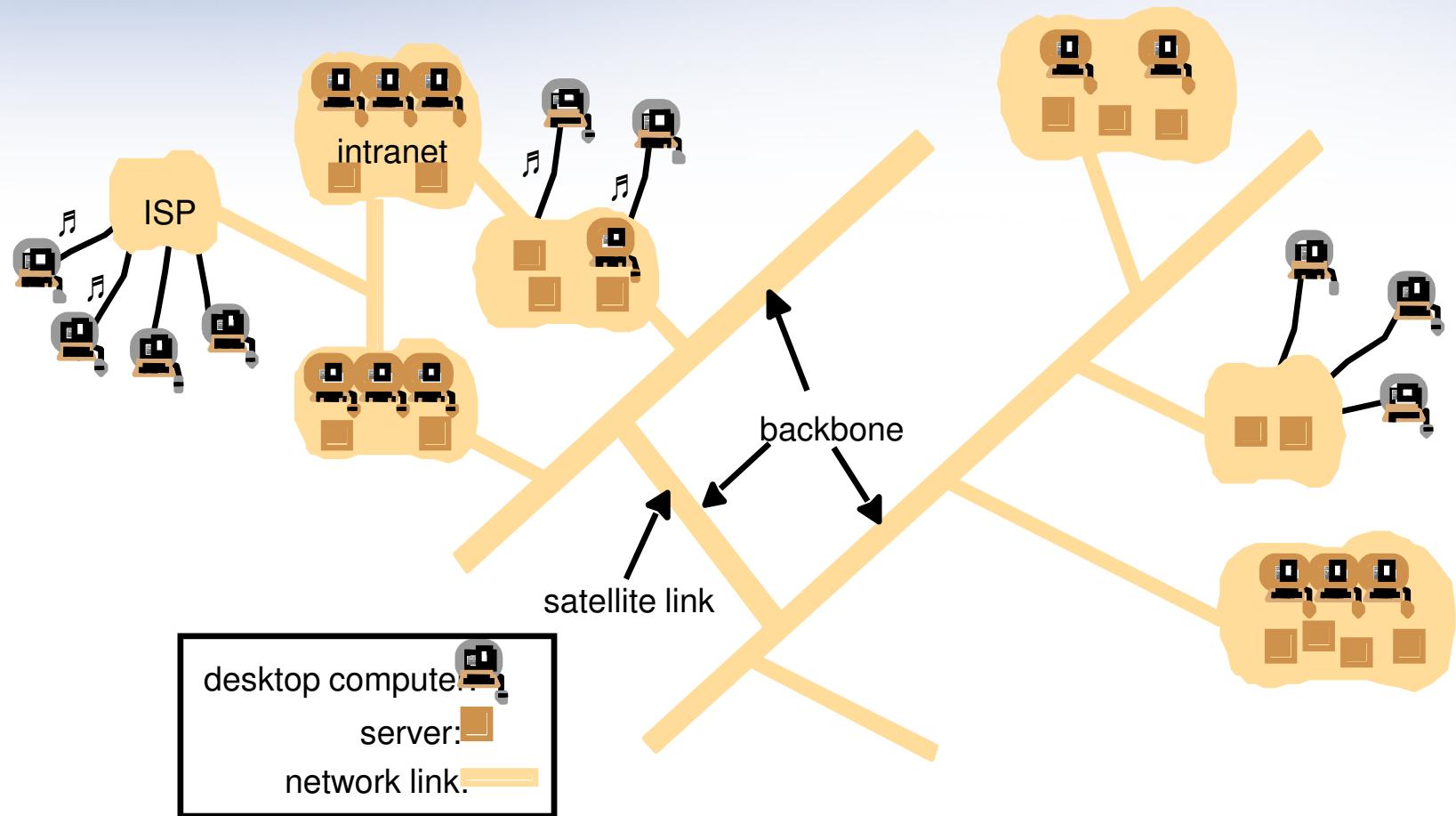
- Sistemet e shpërndara në botën reale janë të projektuar që të punojnë në kushte të ndryshme që lidhen me:
 - Harduerin
 - Softuerin
 - Kanalet e komunikimit
 - Problemet e sigurisë së sistemit
 - Ndarjen e përgjegjësive në mes të komponentëve të sistemeve të shpërndara
 - Vendosjen e këtyre komponentëve në kompjuterët e lidhur në rrjet.
- Këto janë elementet kryesor për projektim të këtyre sistemeve, pasi këto ndikojnë direkt në performancën, qëndrueshmërinë dhe sigurinë e sistemit.

Shembuj të Sistemeve të Shpërndara

- The Internet
- Intranets
- Mobile and Ubiquitous Computing
- Computational Grids
- The Word Wide Web
- The world wide web – information, resource sharing
- Clusters, Network of workstations
- Distributed manufacturing system (e.g., automated assembly line)
- Network of branch office computers - Information system to handle automatic processing of orders
- Network of embedded systems
- New Cell processor (PlayStation 3)

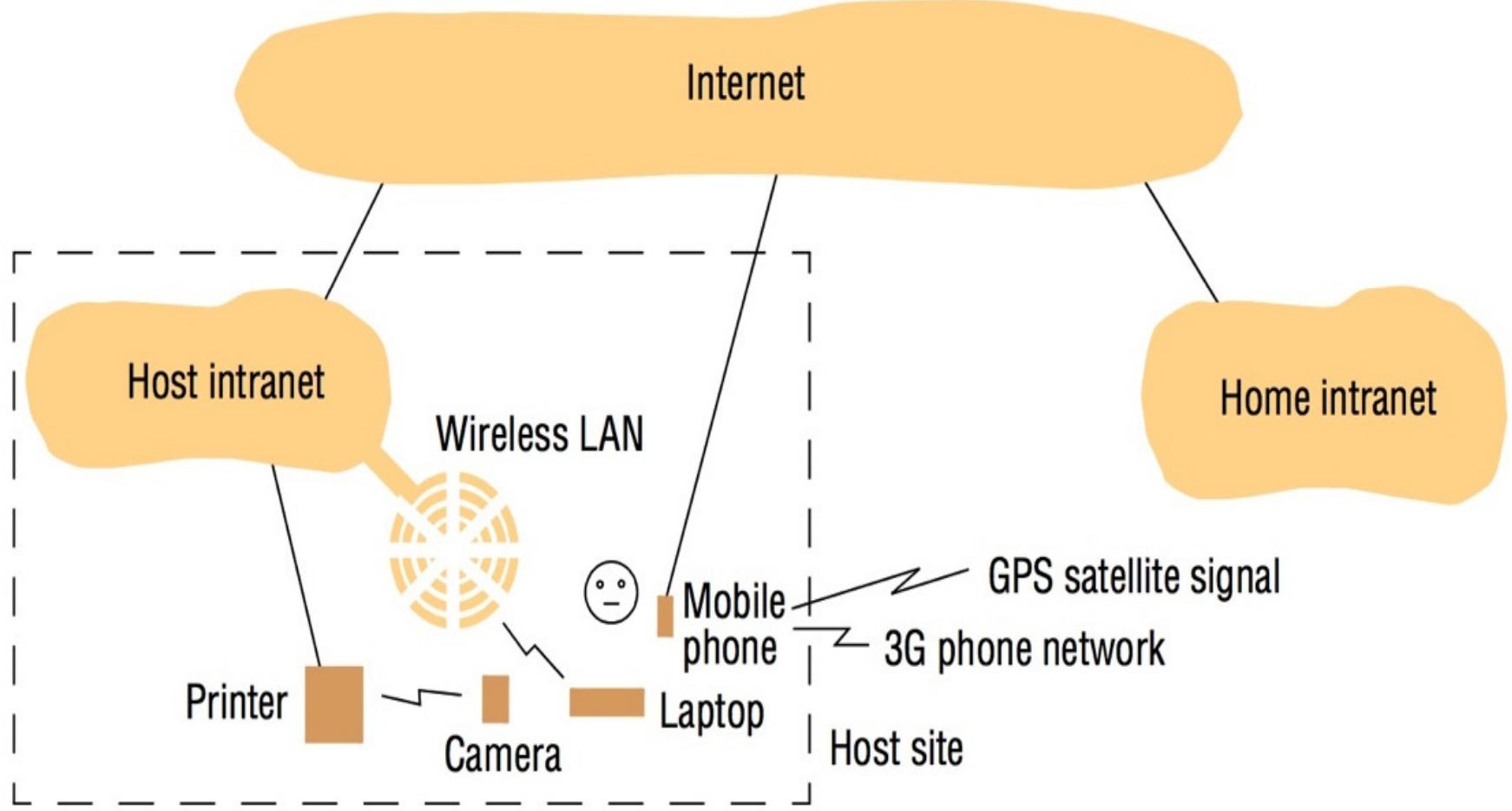
Një pjesë tipike e Internetit

Prof. Ass. Dr. Isak Shabani



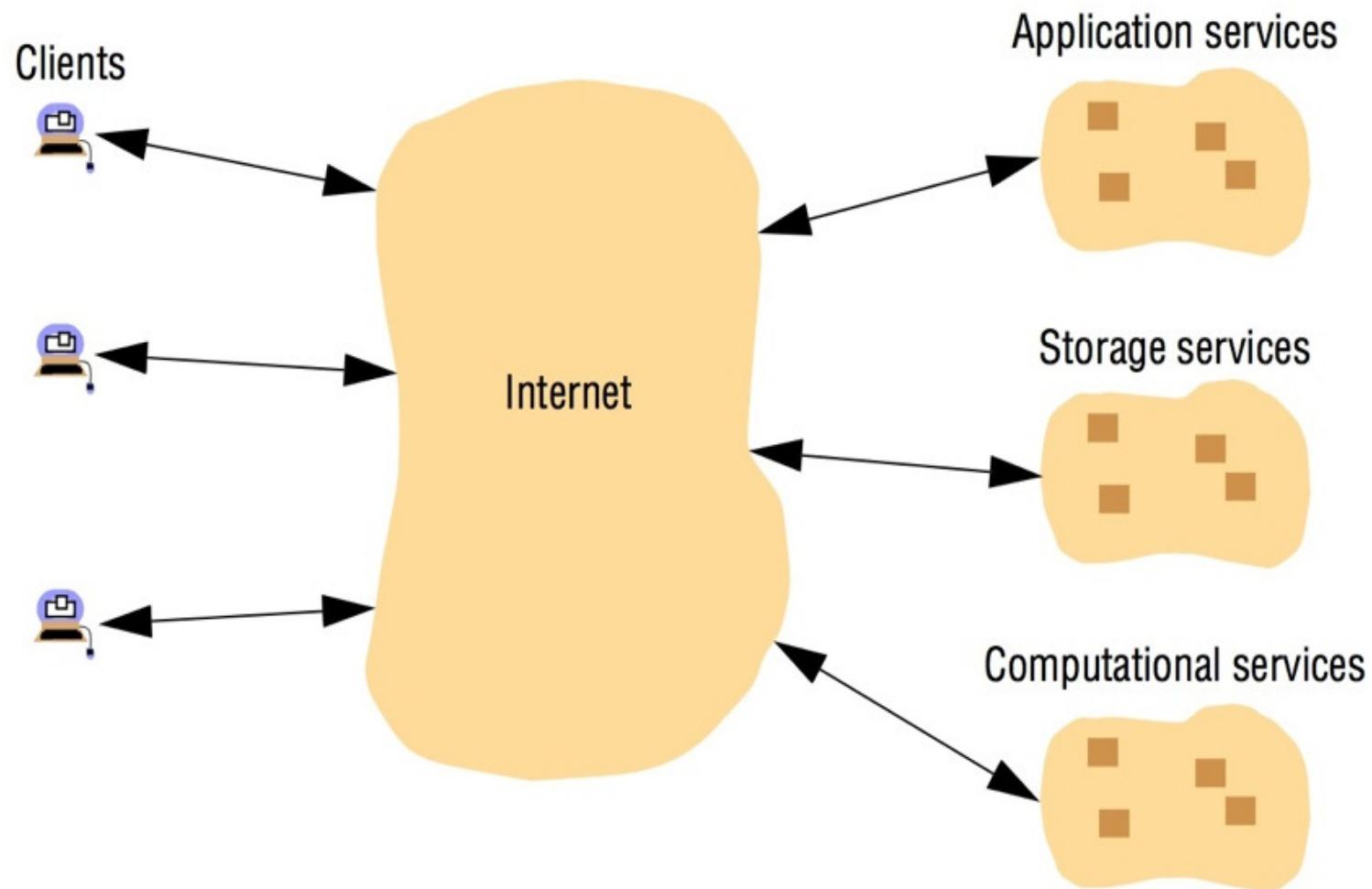
Paisjet portabël dhe handheld në sistemet e shpërndara

Prof. Ass. Dr. Isak Shabani



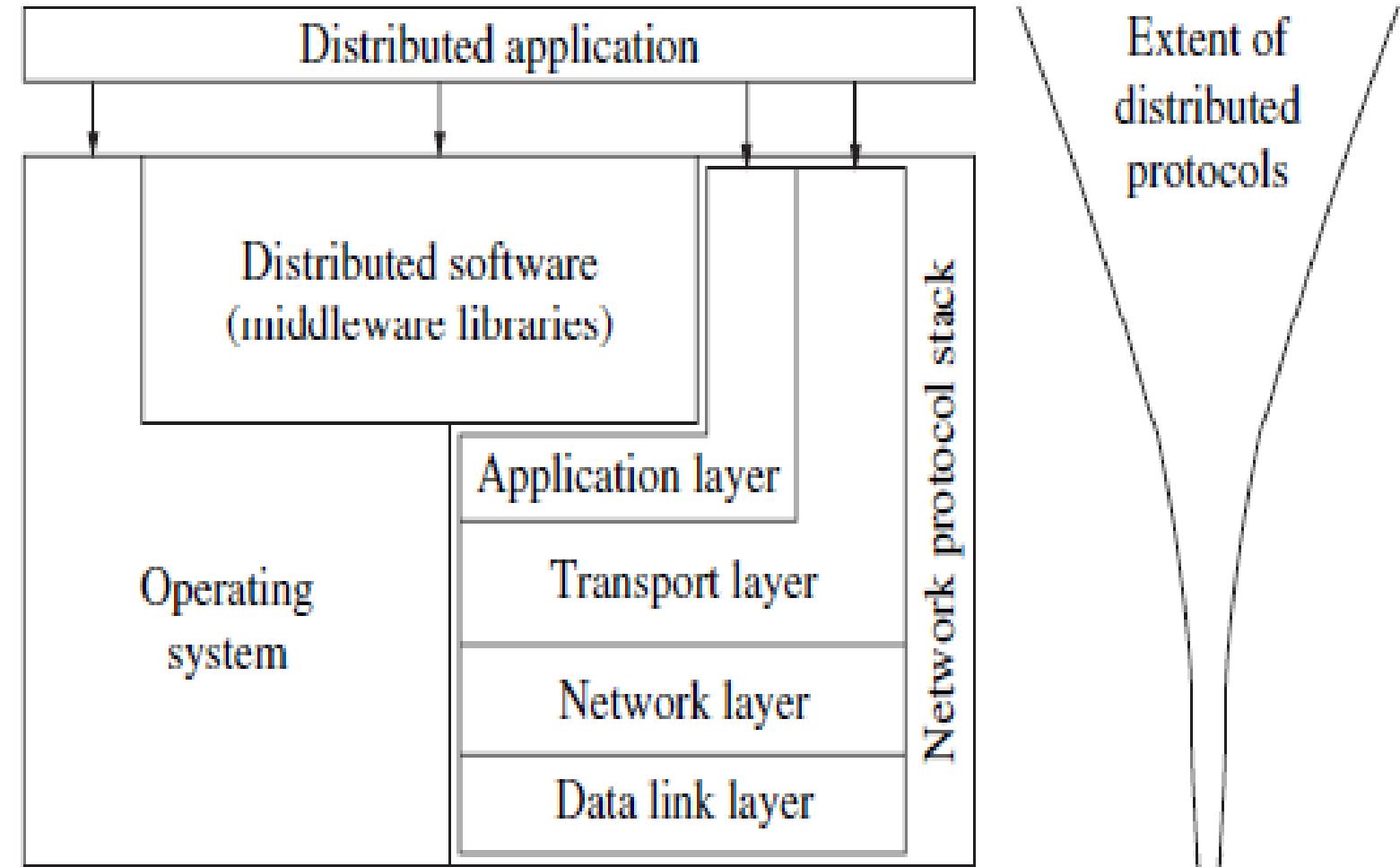
Cloud computing

Prof. Ass. Dr. Isak Shabani



Arkitektuar e aplikacioneve te shperndara

Prof. Ass. Dr. Isak Shabani



Rritja e Internetit (kompjuterët dhe ueb serverët)

Prof. Ass. Dr. Isak Shabani

<i>Data</i>	<i>Kompjuterë</i>	<i>Ueb serverë</i>	<i>Përqindja (%)</i>
1993, Korrik	1,776,000	130	0.008
1995, Korrik	6,642,000	23,500	0.4
1997, Korrik	19,540,000	1,203,096	6
1999, Korrik	56,218,000	6,598,697	12
2001, Korrik	125,888,197	31,299,592	25
2003, Korrik	~200,000,000	42,298,371	21
2005, Korrik	353,284,187	67,571,581	19

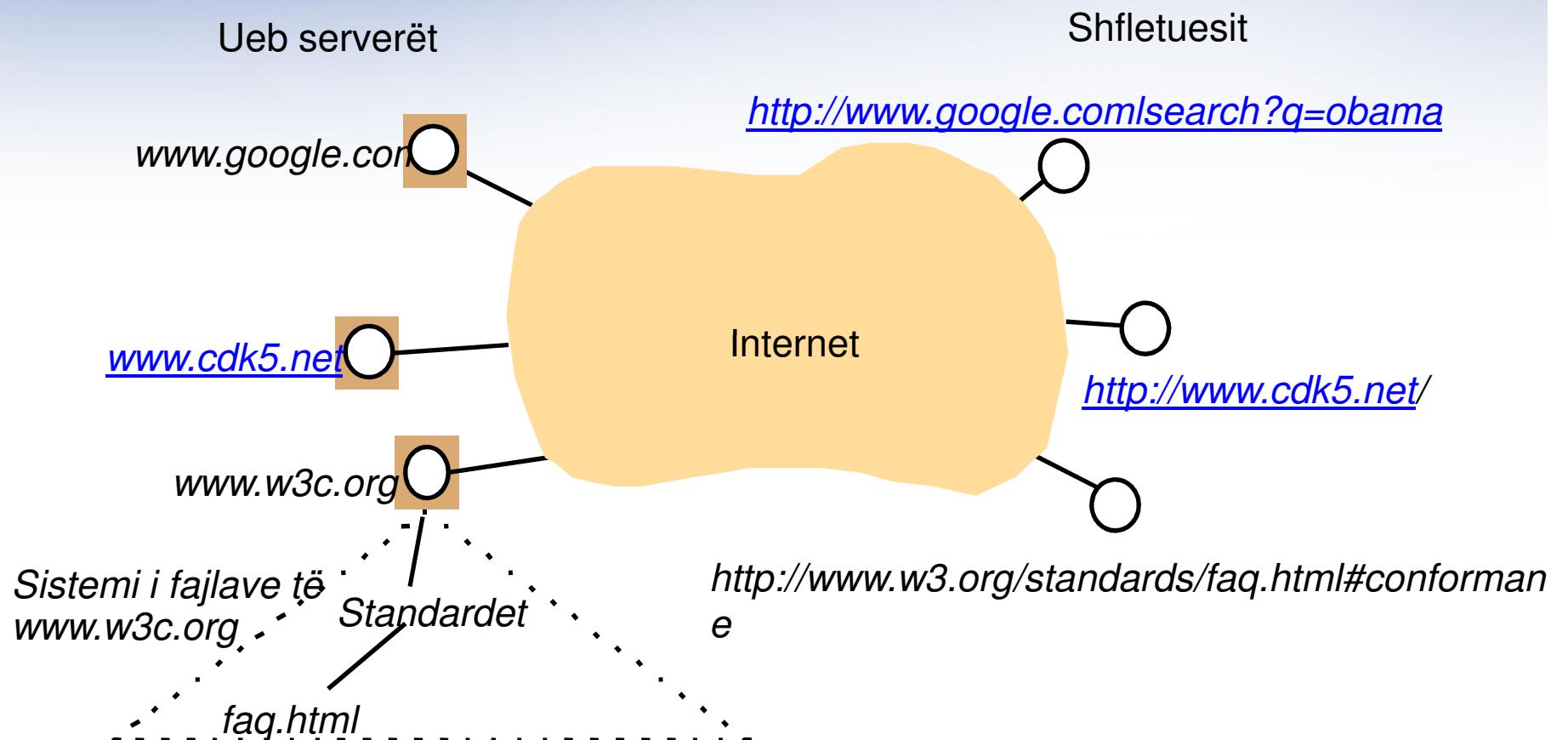
Transparenca

Prof. Ass. Dr. Isak Shabani

- **Transparenca në qasje:** mundëson që burimeve lokale dhe në distancë t'u qasemi me operacione identike.
- **Transparenca në lokacion:** mundëson qasjen në burime pa dijeninë e lokacionit fizik të tyre ose lokacionin në rrjetë (për shembull, cila ndërtësë ose IP Adresë).
- **Transparenca në Konkurrencë:** mundëson operimin e disa proceseve të operojnë në mënyrë konkurrente duke përdorur resurse të ndara pa interferencë në mes tyre.
- **Transparenca në replikim(përsëritje):** mundëson disa instanca të resurseve të përdoren për të rritur qëndrueshmërinë dhe performancën pa njojuri të përsëritjeve/replikave nga shfrytëzues ose programer të aplikacionit.
- **Transparenca në prishje:** mundëson fshehjen e gabimeve, duke lejuar shfrytëzuesit dhe programet aplikative të përfundojnë punët e tyre pavarësisht nga prishjet në komponentët harduerike ose softuerike.
- **Transparenca mobile:** lejon lëvizjen e resurse nga klientët në sistem pa ndikimin në operacione të shfrytëzuësve ose programeve.
- **Transparenca në performancë:** lejon që sistemet të konfigurohet dhe të përmirësojnë performancën në varësi të ngarkesës.
- **Transparenca e shkallëzimit:** lejon sistemin dhe aplikacionet të zgjerohen në shkallë pa ndryshime në strukturën e sistemit ose të algoritmeve të aplikacionit.

Ueb sherbimet dhe Ueb browseret

Prof. Ass. Dr. Isak Shabani



Arkitektura e Ueb Shërbimeve

WIKISKEPËR S OEN DUGJIMIUSAG

Prof. Ass. Dr. Isak Shabani

Teknologjite “Distributed Object Computing”

Teknologjite “Distributed Object Computing” bazohet në tri familje të mëdha teknologjike fig.1:

- **Teknologjite e bazuara në Ueb**, në mënyrë të veçantë të përshtatura për përhapjen e shërbimeve;
- **Middleware me objekte të shpërndara**, për të organizuar një larmi objektesh, shpesh të ofruara nga organizata të ndryshme, në kontekstin e një teknologjie unitare;
- **Teknologjite e inkapsulimit të sistemeve legale**, për të ofruar informacione dhe shërbime legale në mënyra të ndryshme.

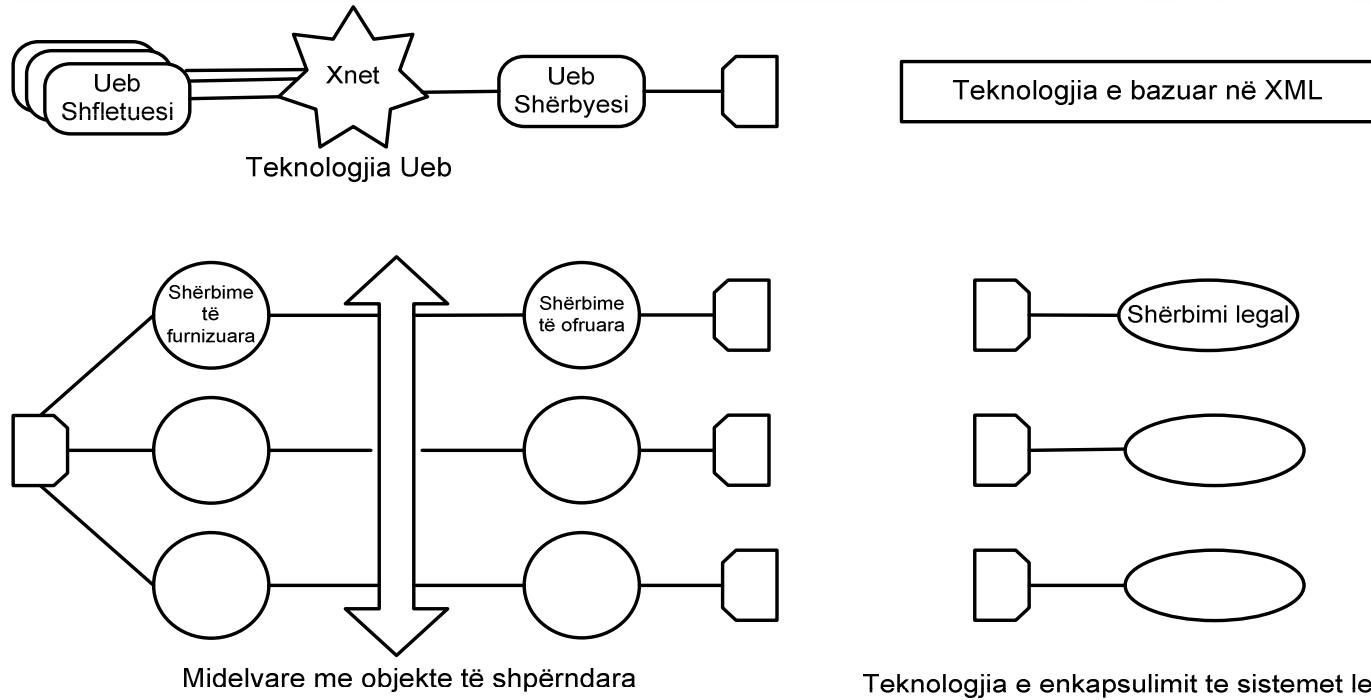


Fig.1 Teknologjite “Distributed Object Computing”

Teknologjitetë “Distributed Object Computing”

Prof. Ass. Dr. Isak Shabani

Me qëllim që të tregojmë në mënyrë sistematike të paktën një nëngrup përfaqësues nga të gjithë mundësitë aplikative, të cilat mund të paraqiten në realitet, do të analizojmë rastin që paraqitet me ndryshimin e katër parametrave përfaqësues të mëposhtëm:

- Ofrimi i shërbimit nga një organizatë ose nga më shumë organizata;
- Ofrimi i shërbimeve IT ose transaksioneve;
- Ofrimi i shërbimit për përdoruesit e brendshëm ose të jashtëm;
- Nevoja për të inkapsuluar sistemet legale në kufijtë e shërbimit të ofruar.

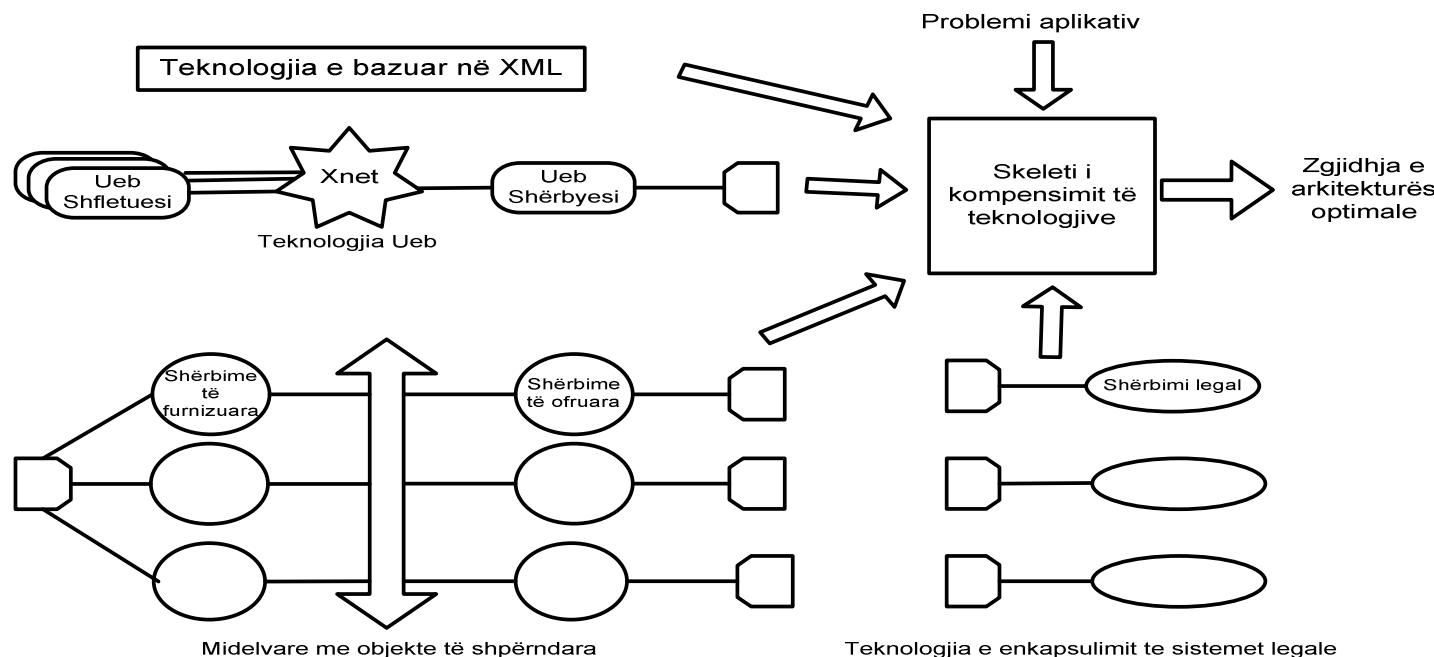


Fig.2 Strategjia e udhëhequr nga nevojat aplikative.

Niveli i softuerit në sistemet e shpërndara

- Fillimisht termi arkitektura softuerike ka nënkuptuar organizimin e softuerit në module në një kompjuter, kurse tani nënkupton shërbimet e ofruara dhe të kërkuara nga proceset që gjendet në të njëjtin apo në kompjutera të ndryshëm.
- Ky përcaktim i proceseve dhe shërbimeve mund të shprehet si nivel i shërbimeve.
- Shërbyesi në kuptim abstrakt është një proces që pranon një kërkesë dhe kryen një punë të caktuar pas pranimit të kërkesës.
- Një shërbim i shpërndarë mund të ofrohet nga një apo më shumë procese të shërbyesit të cilët bashkëpunojnë në mes tyre vetëm dhe me klientin për të realizuar një punë.
- Në fig 4. përmenden dy terma shumë të rëndësishme: platforma dhe middleware.

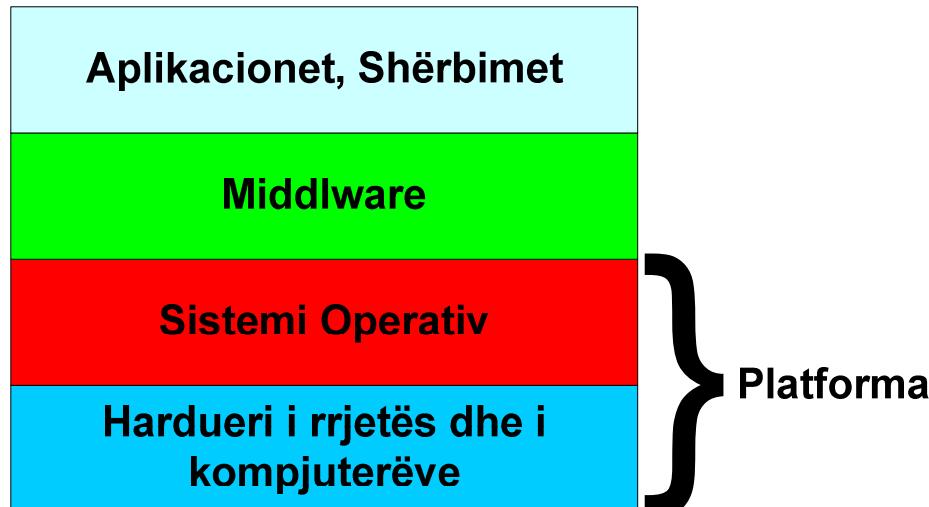


Fig 3. Shtresat softuerike dhe harduerike të sistemit të shpërndarë.

Arkitektura e Shërbimeve Ueb

Shërbimet Ueb janë teknologji e shpërndarë, që realizojnë ndërveprim dhe bashkëpunim mes ofruesve të shërbimit dhe klientëve.

Modeli i Shërbimeve Ueb ndjek këto hapa: publikon, gjen dhe lidh.

- Në hapin e parë, një ofrues shërbimi publikon një Shërbim Ueb në regjistrin e Shërbimeve Ueb.
- Së dyti, klienti i cili është në kërkim të një shërbimi për nevojat e tij shikon në regjistër.
- Pasi të ketë gjetur me sukses disa mundësi, ai zgjedh shërbimin sipas preferencave të veta dhe shkarkon përshkrimin e Shërbimeve dhe lidhet me atë që të thërrasë dhe të shfrytëzojë shërbimin.

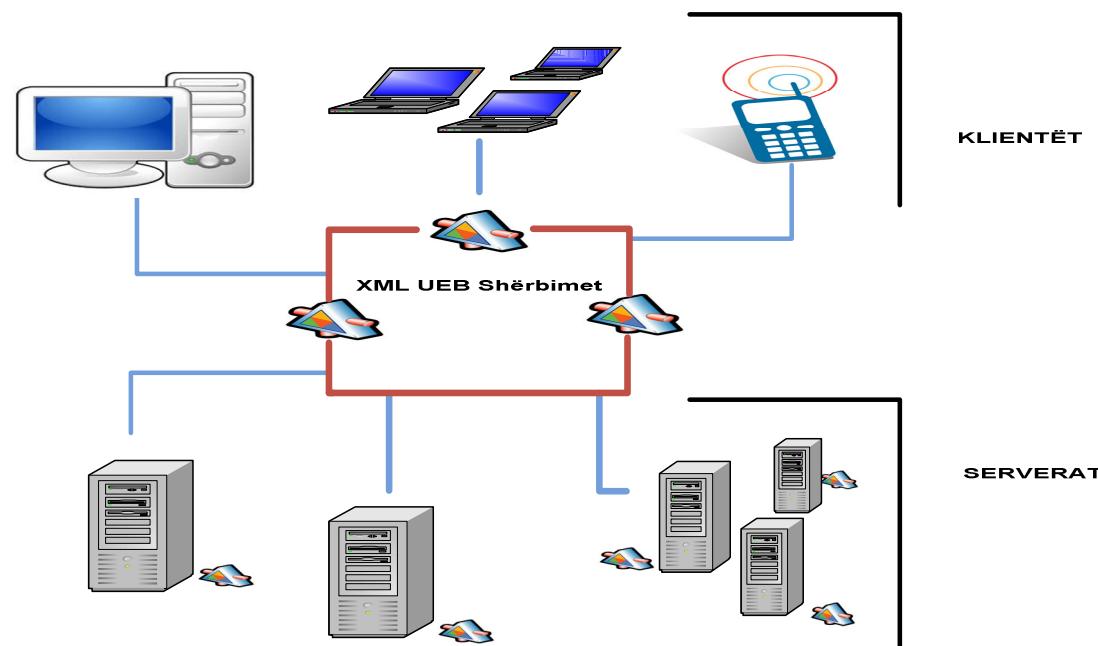


Fig. 4. Arkitektura funksionale e Shërbimeve Ueb.

Bashkëveprimi me Ueb Shërbime

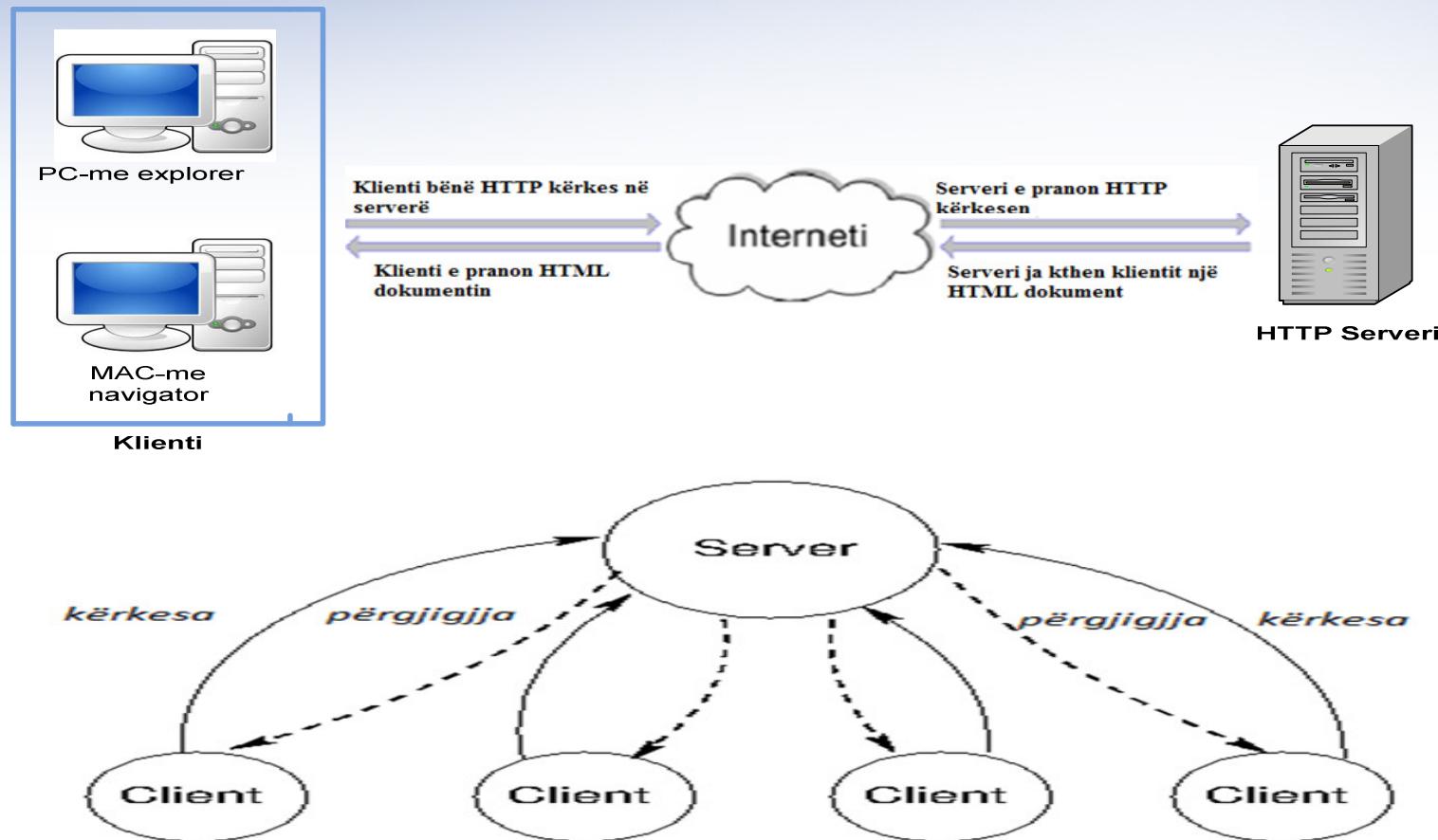


Fig. 5. Komunikimi i klientit me serverin

Bashkëveprimi me Ueb Shërbime

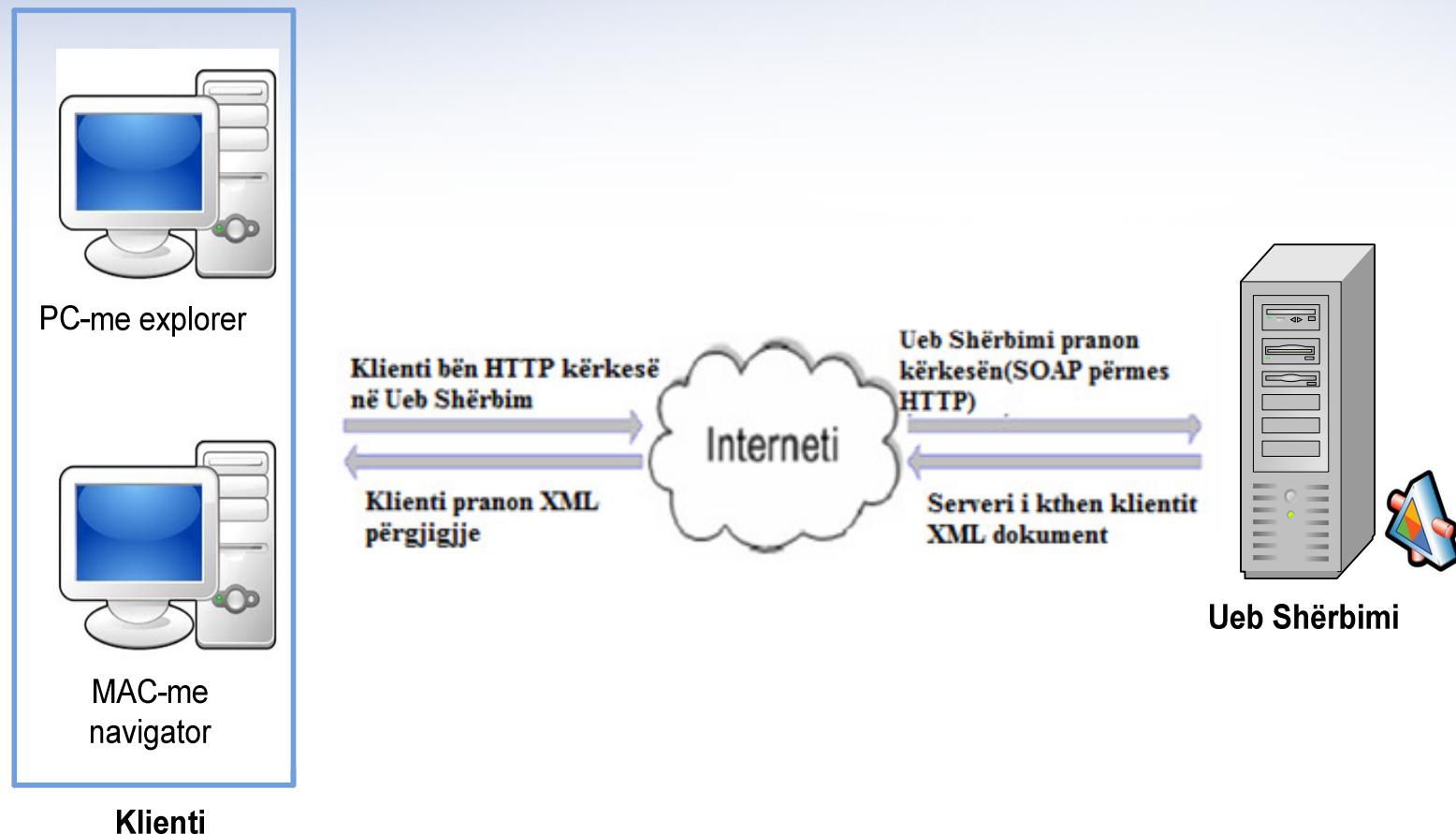


Fig. 6. Komunikimi i klientit me ueb shërbimin përmes një ueb serveri

Arkitektura e modelit te Ueb Sherbimeve

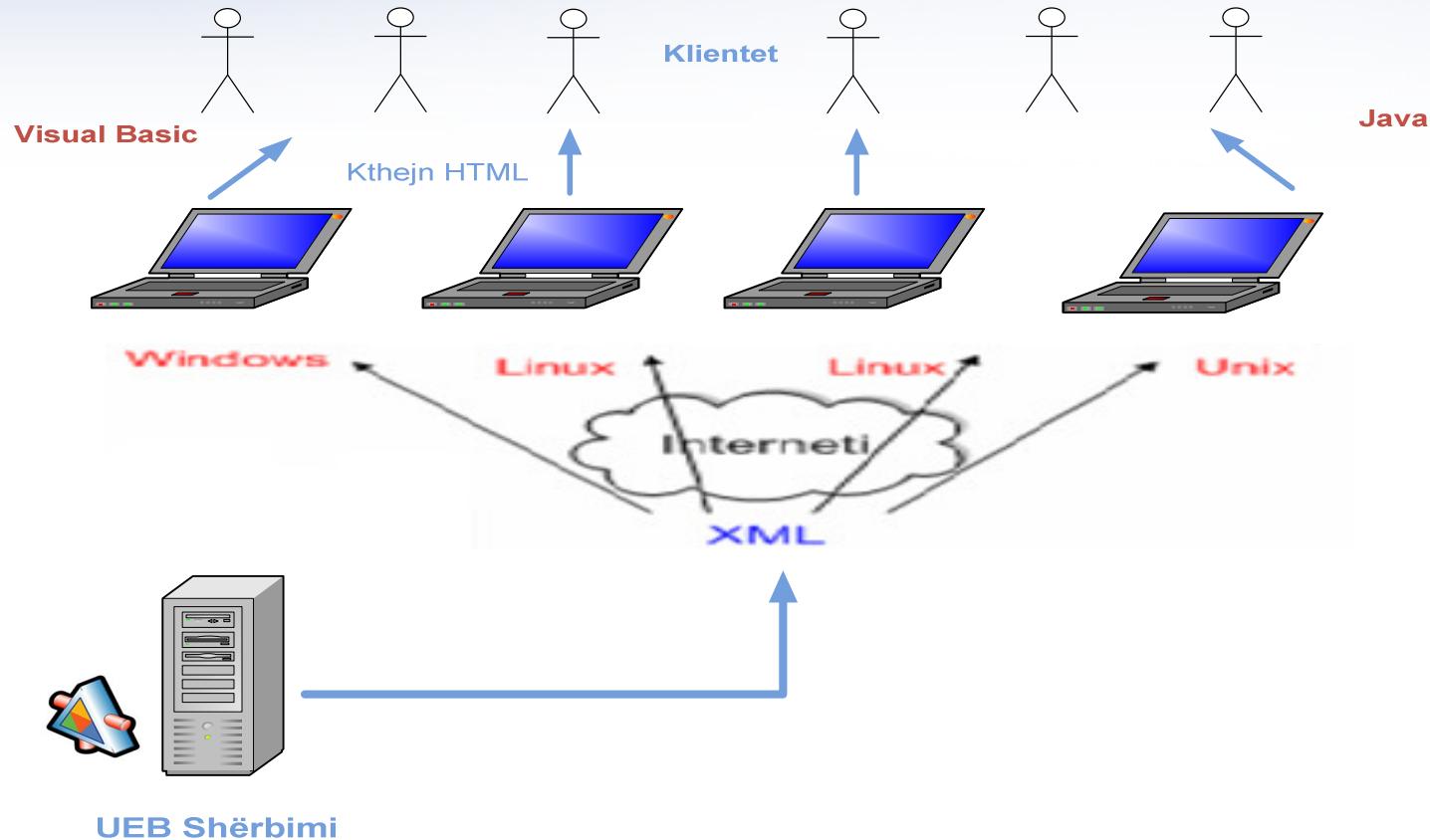


Fig. 7. Modeli i integrimit të ueb shërbimi

Arkitektura e Ueb Sherbimeve

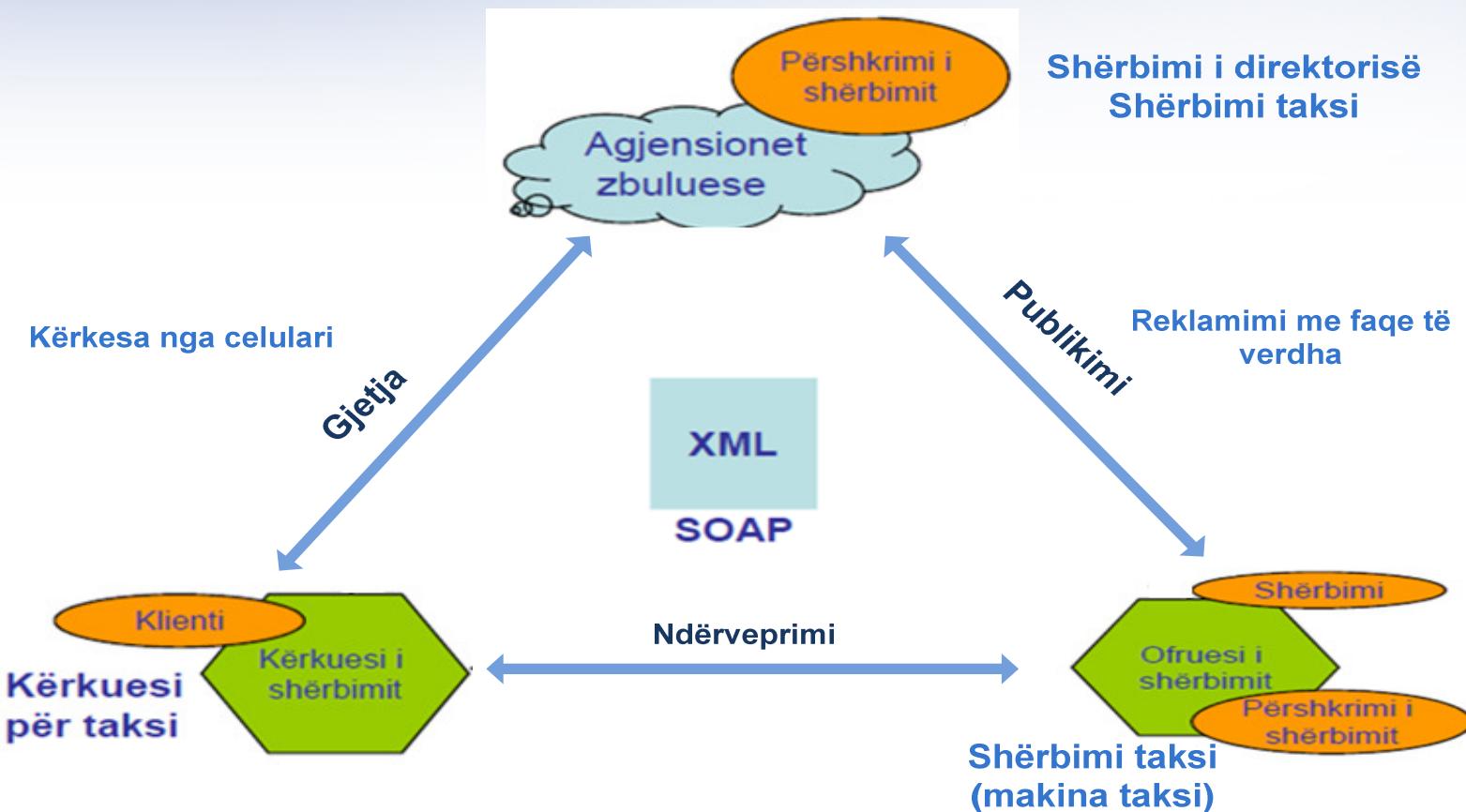


Fig. 8. Arkitektura e Shërbimeve Ueb.

Arkitektura e Shërbimeve Ueb

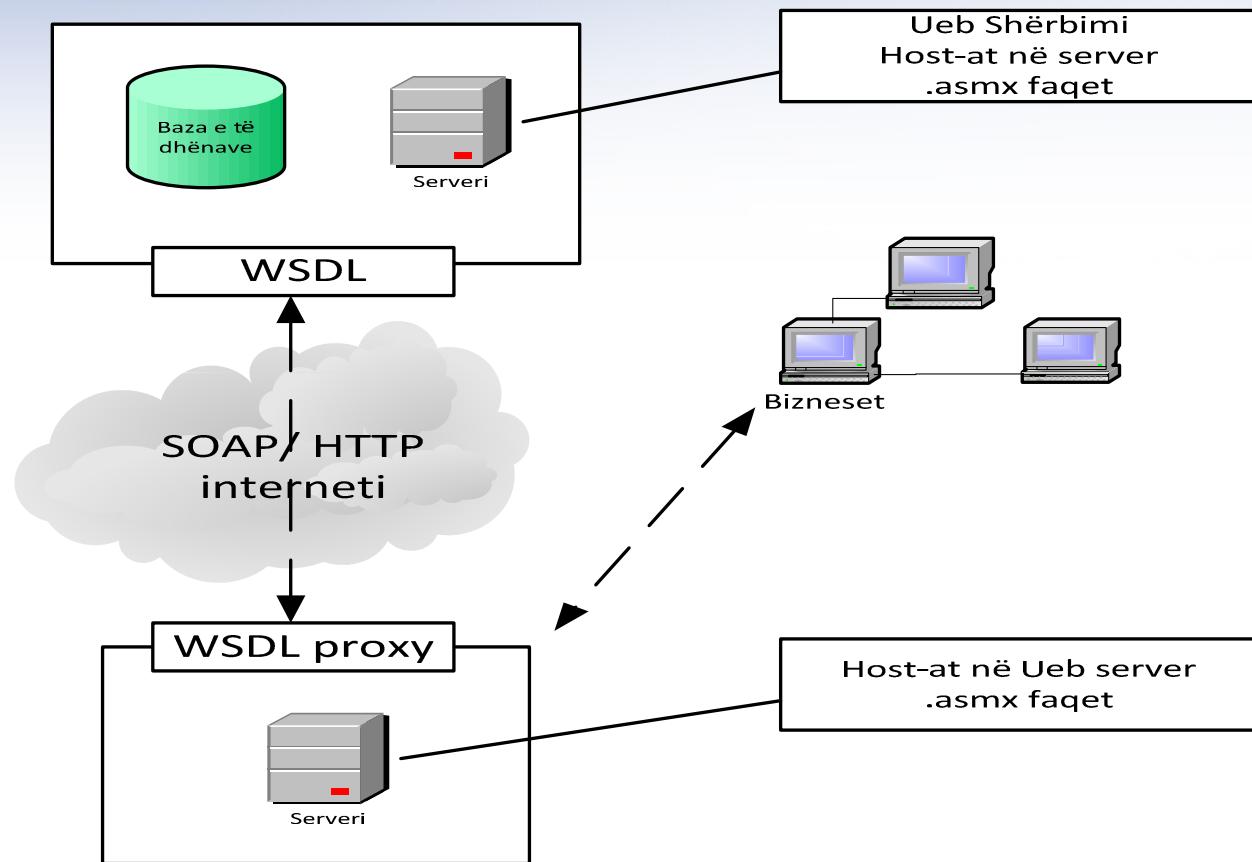


Fig. 9. Komunikimi i shërbimit Ueb me bizneset

Arkitektura e Shërbimeve Ueb për marrjen e një Ueb shërbimi

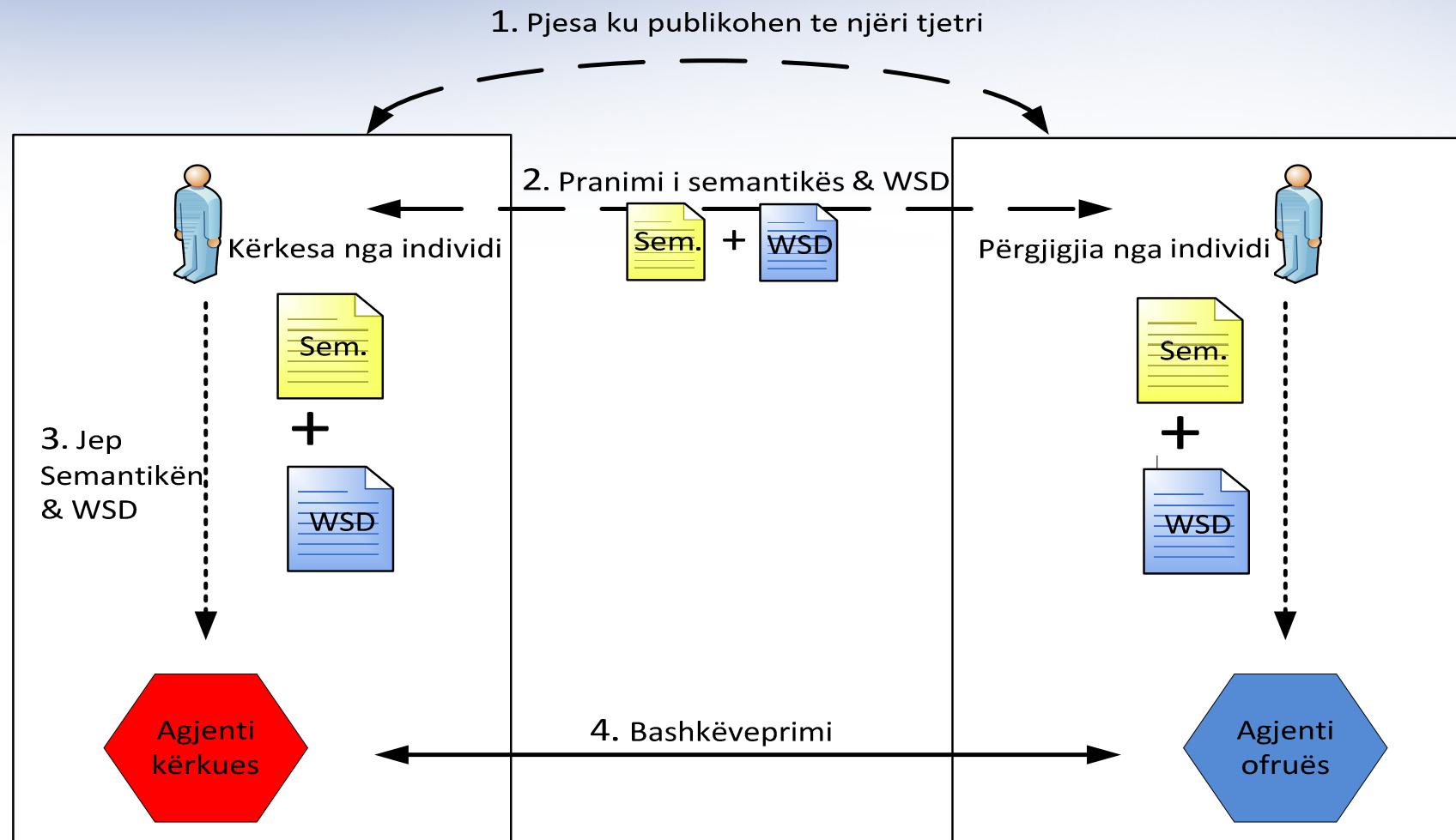


Figura 10. Arkitektuara e shërbimit Ueb

Ueb Shërbimet dhe standardet

Shërbimi Ueb është një ndërfaqe e vendosur ndërmjet kodit të aplikacionit dhe përdoruesit të atij kodi.

Ai vepron si një shtresë abstraktioni, duke bërë ndarjen e platformës dhe detajet specifike të gjuhës programuese, që thérret kodin e aplikacionit.

Kjo shtresë e standardizuar mundëson afrimin në funksionet e aplikacionit me ndihmën e çdo gjuhe programuese, që mbështet Shërbimet Ueb, fig.4.

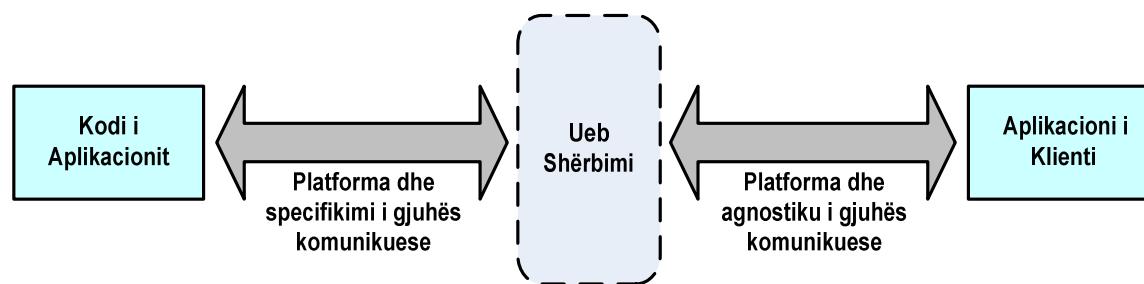


Figura 11. Shërbimet Ueb si shtresë abstraktioni ndërmjet aplikacionit klient dhe aplikacionit kod.

Teknologjia e Ueb Shërbimeve

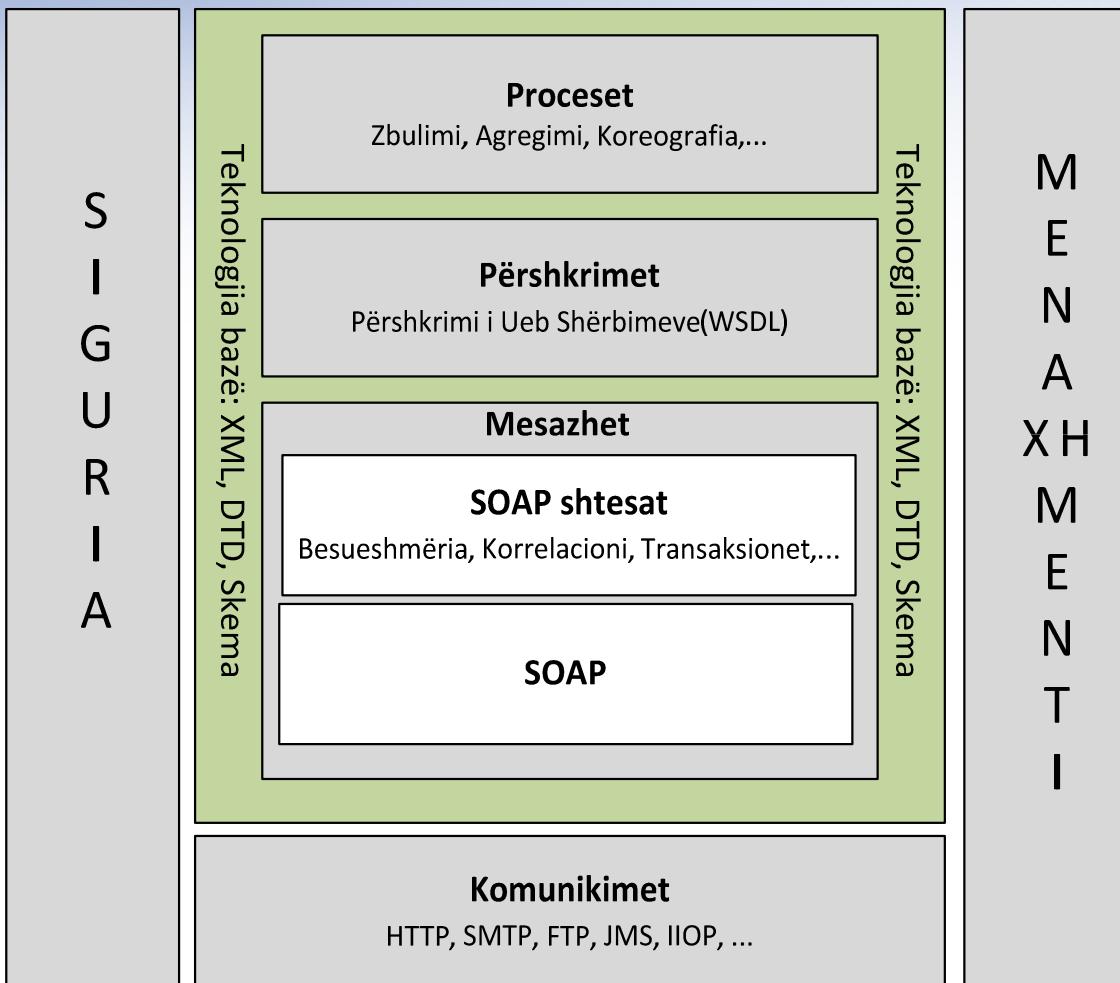


Figura 12 Teknologjet e sherbimeve
Ueb

Si funksionojnë Ueb Shërbimet

- Shërbimi Ueb është një strukturë e bazuar mbi porositë.
- Kërkesa e vetme, që i parashtronet Shërbimit Ueb është që të jetë në gjendje të dërgojë dhe të pranojë porosi duke përdorur disa kombinime të protokolleve standarde të Internetit.
- Forma më e rëndomtë e shërbimeve Ueb është thirrja e procedurave të cilat janë duke punuar në server, ku porosia kodohet si vijon: ***"Thirre këtë nën-rutinë me këto argumente"***, dhe përgjigjja do të ishte ***"Ja ku janë rezultatet e nën-rutinës së thirrur"***.

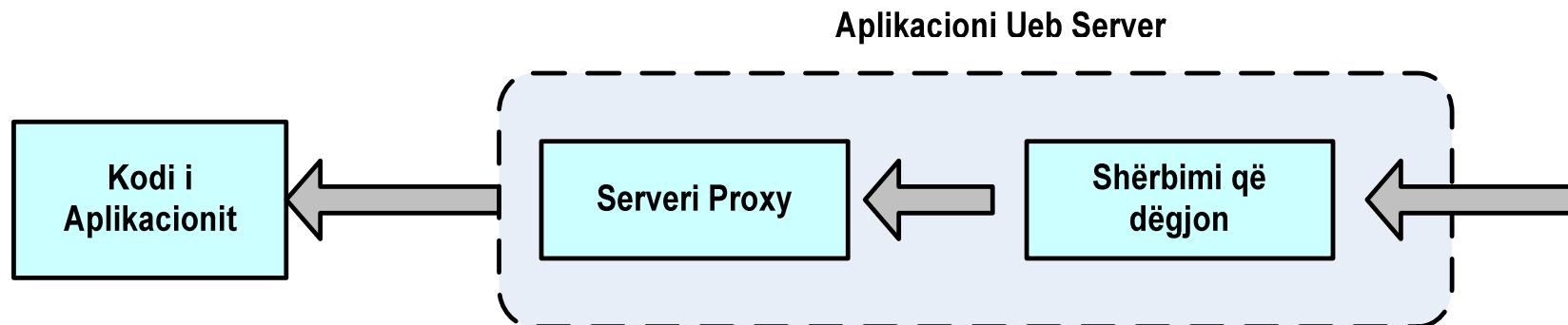


Figura 13. Një shërbim Ueb i përbërë nga disa komponentë kryesorë.

Komponentët e Shërbimeve Ueb

UDDI
Universal Description Discovery and
Integration

WSDL
XML Web Services Description
Language

SOAP
Simple Object Access Protocol

XML dhe HTTP

Figura 14. Komponentet e Ueb Sherbimeve

Zbulimi i Shërbimit Ueb

Kur kemi të bëjmë me zbulimin e shërbimeve Ueb, atëherë duhet ta kuptojmë si aktin e gjetjes së një përshkrimi të një shërbimi, i cili më parë mund të ketë qenë i panjohur, por që plotëson kushtet përkatëse të funksionalitetit.

Qëllimi pse bëhet një gjë e tillë është gjetja e ueb shërbimit sa më adekuat.

Figura 12. tregon hapat për zbulimin e një Shërbimi Ueb.

Zbulimi i Shërbimeve Ueb mund të jetë manual ose automatik.

Hapi 1: ofruesi dhe kërkuesi komunikojnë me njëri tjetrit.

Hapi 2: Ofruesi dhe kërkuesi pajtohen për semantikën që do të përdorin.

Hapi 3: Ofruesi dhe kërkuesi dërgojnë tek agjentët e tyre përshkrimet WSD dhe semantikën që do përdorin gjatë shkëmbimit të mesazheve.

Hapi 4: Agjenti ofrues dhe kërkues fillojnë shkëmbimin e mesazheve SOAP me interes dhe të përcaktuara nga krijuesit e tyre.

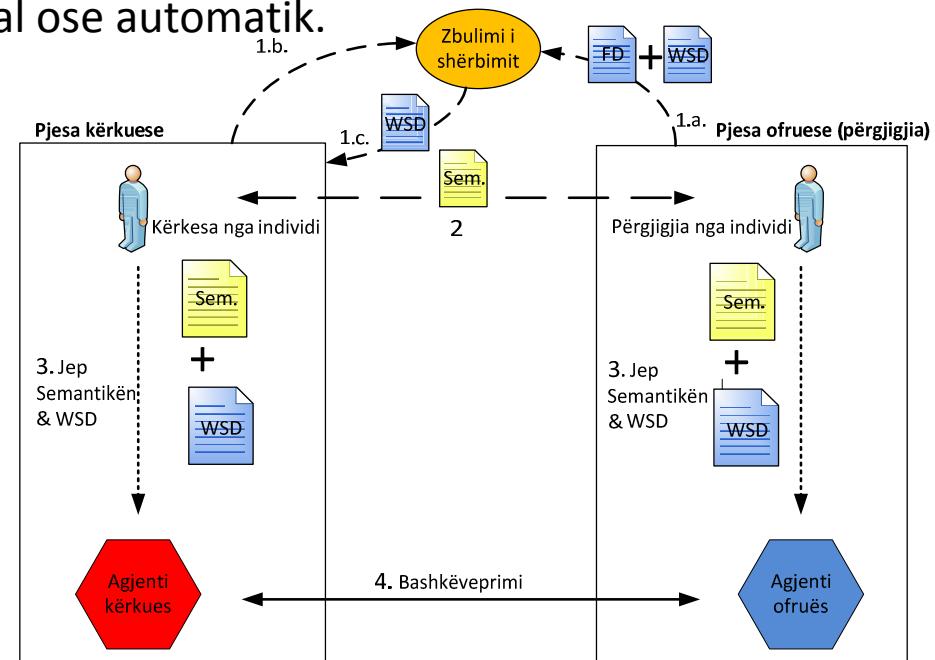


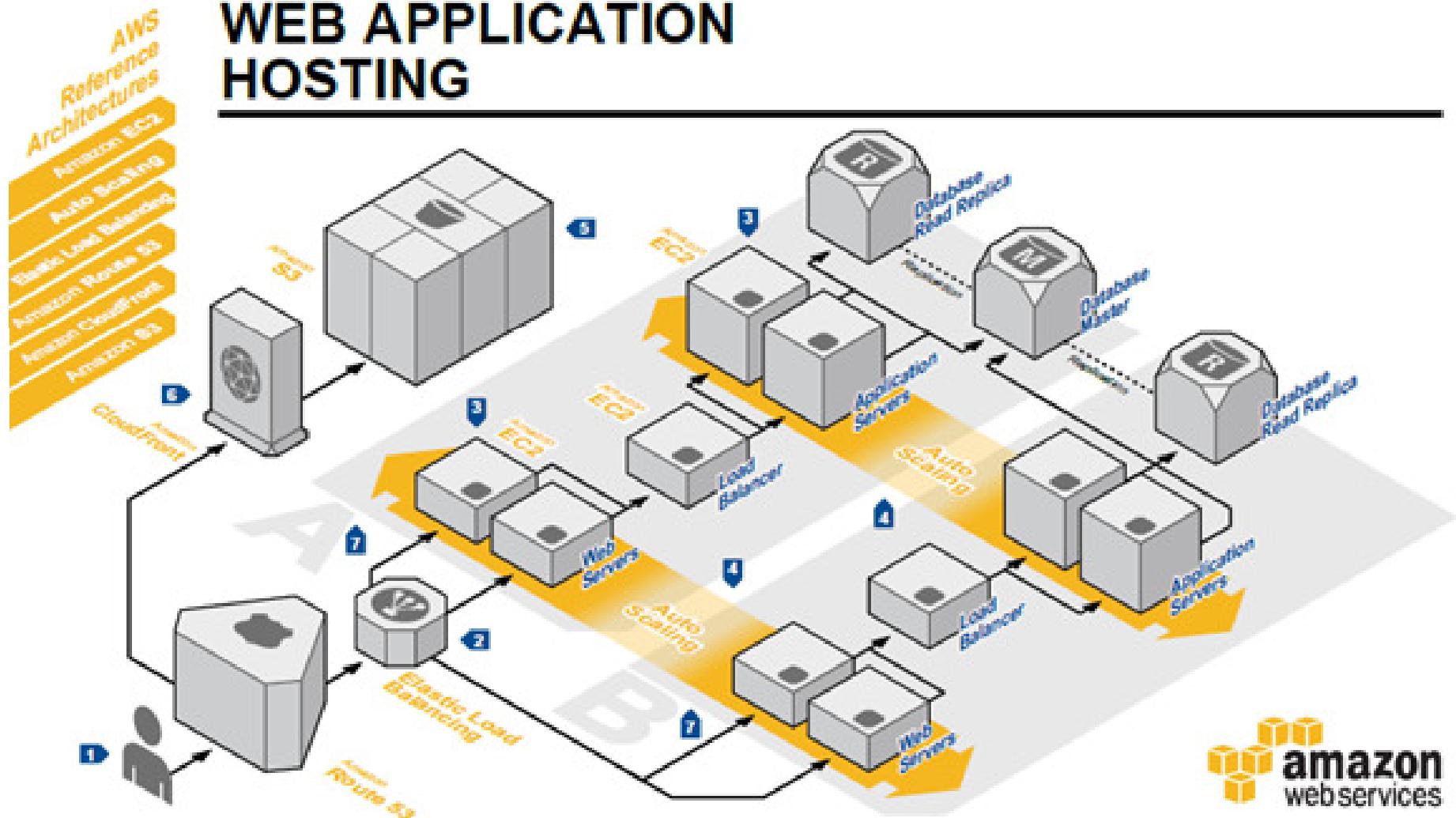
Figura 15. Zbulimi i Ueb shërbimit

Relacionet ndërmjet Shërbimeve Ueb

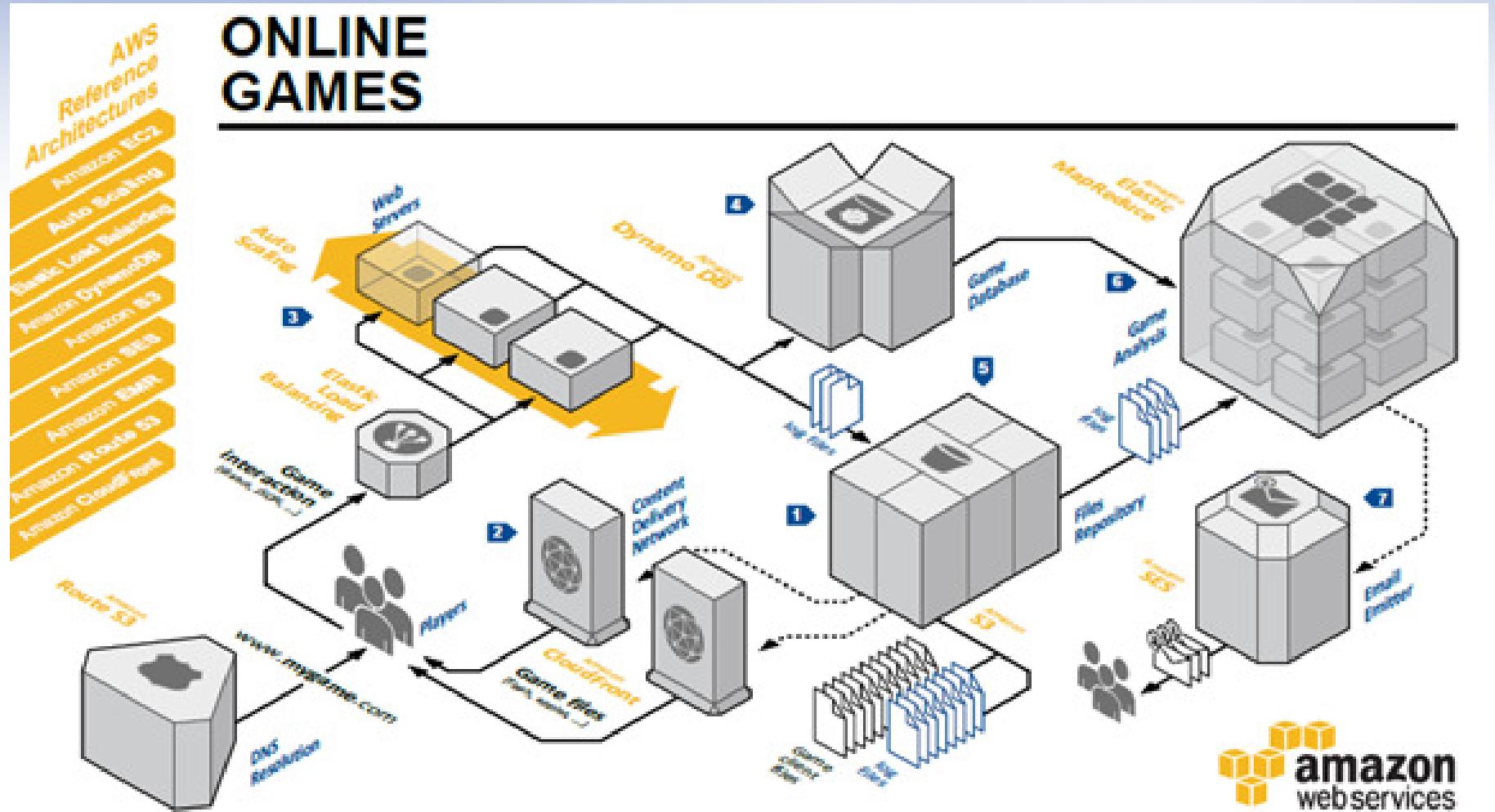
- Në arkitekturën e Shërbimeve Ueb rëndësi të madhe ka edhe lidhja në mes të shërbimeve Ueb.
- Relacioni në mes Shërbimeve Ueb, pohimi X është në relacion me Y, përcakton lidhjet në mes koncepteve X dhe Y, i tillë që çdo X është po ashtu edhe Y.
- Para së gjithash kur themi se koncepti X është Y, ne mendojmë se çdo veçori e Y është po ashtu veçori e X.
- Megjithatë, derisa X është koncept ndryshe nga Y, vetitë e X po ashtu mund të jenë të ndryshme nga vetitë e Y.
- Për shembull në konceptin e shërbimeve, themi se çdo shërbim ka një identifikues.
- Në koncept më specifik të Shërbimeve Ueb ne tregojmë se çdo shërbim Ueb e ka një identifikues në formën e identifikuesit URI (*Uniform Resource Identifier*).
- Sa herë që krijohet një shërbim Ueb, supozojmë se atij i shoqërohet një identifikues.
- Prania e këtij identifikuesi na siguron që Shërbimi Ueb ekziston, përndryshe ai është shkatërruar.

Amazon Ueb sherbimet

WEB APPLICATION HOSTING



Shembull i arkitekturës për hostimin e shërbimeve për Online Games permes Ueb Sherbimeve



amazon
web services

Modelet arkitekturale të Ueb Shërbimeve

Prof. Ass. Dr. Isak Shabani

- Arkitektura e Shërbimeve Ueb ka katër modele.
 - Modeli i orientuar në mesazhe (message-oriented)
 - Modeli i orientuar në shërbime (service-oriented)
 - Modeli i orientuar në burime (resource-oriented) dhe
 - Modeli i politikave zhvilluese (development policies)
- Secili model është emërtuar në atë mënyrë, që të shikohet si koncept kyç i komplet arkitekturës së Ueb shërbimeve .

Modeli message-oriented

Prof. Ass. Dr. Isak Shabani

Fokusohet në mesazhe, strukturën, transportin e tyre, por pa ndonjë referencë në arsyet për mesazhe, po ashtu edhe për përbajtjen e tyre.

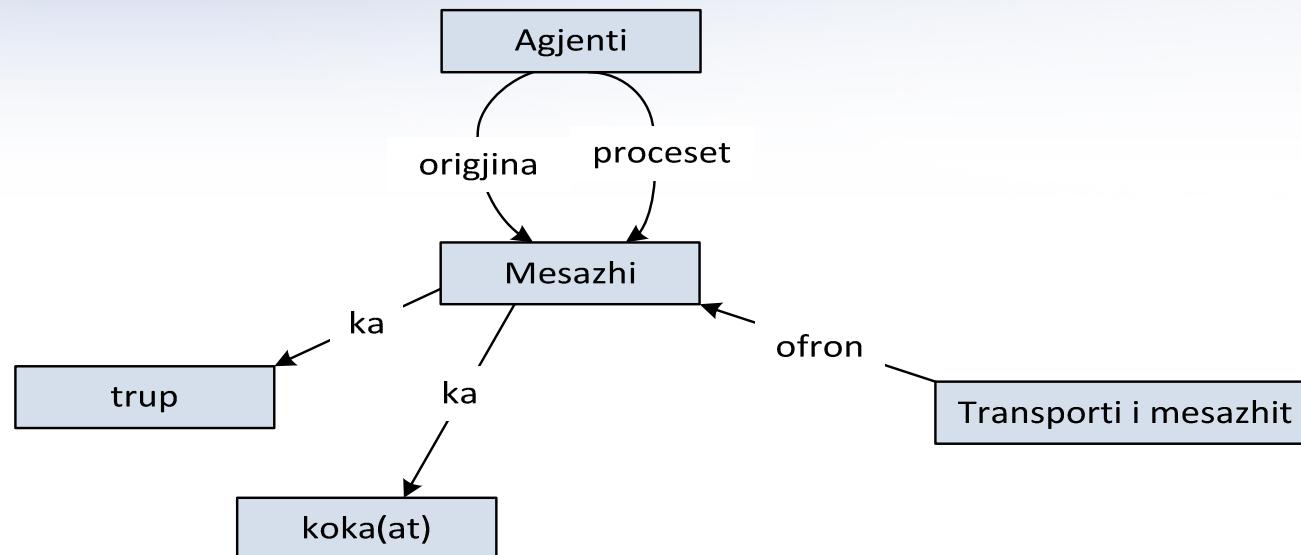


Fig 1.
*Modeli i thjeshtuar
message-oriented*

Thelbi i modelit të mesazheve sillet rreth koncepteve të ilustruara më lart:

- agjentit (që merr dhe dërgon mesazhe),
 - strukturës së mesazhit (në elementet e kokës dhe trupit të mesazhit) dhe
 - mekanizmit të shpërndarjes së mesazhit.
- Natyrisht duhet të merren parasysh edhe detajet e tjera shtesë: roli i linjave të veprimit (si duhet të veprojnë ato në nivelet e modelit të caktuar të mesazhit).

Modeli service-oriented

Prof. Ass. Dr. Isak Shabani

- Fokusohet në aspektet e shërbimeve, veprimeve e kështu me radhë.
- Në sistemet e shpërndara, shërbimet nuk mund të realizohen pa ndonjë teknikë të përdorimit të mesazheve, e kundërta nuk vlen: mesazhet nuk kanë nevojë të jenë të lidhura me shërbime.

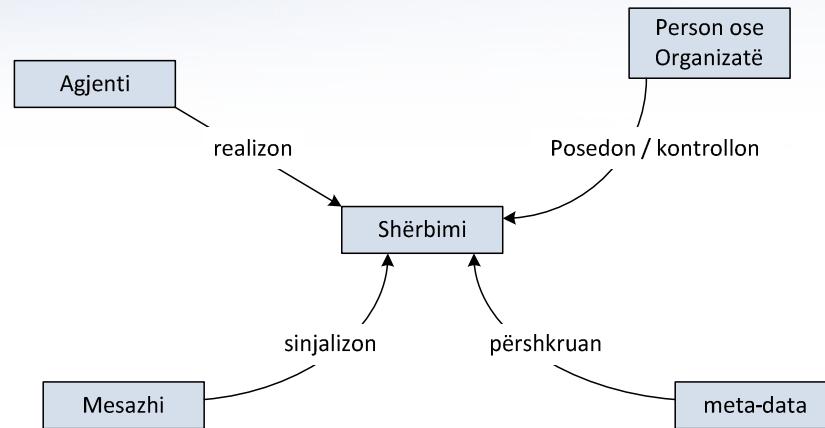


Fig 2.
*Modeli i thjeshtuar
service-oriented*

- Shërbimi realizohet nga një agjent dhe shfrytëzohet nga një agjent tjeter.
- Shërbimet janë të ndërmjetësuara nga kuptimi i shkëmbimeve të mesazheve ndërmjet agentëve kërkues dhe atyre ofrues.
- **Një aspekt shumë i rëndësishëm i shërbimeve është lidhja e tyre me botën reale, kryesisht shërbimet shpërndahen për të ofruar funksionalitet në botën reale.**
- Ne po e modelojmë këtë duke e sqaruar konceptin e pranuesit të shërbimit, i cili në qoftëse është person ose organizatë, ka një përgjegjësi për shërbimin.
- Gjithashtu edhe ky model lejon përdorimin e të dhënave të përafërtë(ang. meta-data).

Modeli resource-oriented

Prof. Ass. Dr. Isak Shabani

- Fokusohet në burimet që ekzistojnë dhe që kanë pronar.

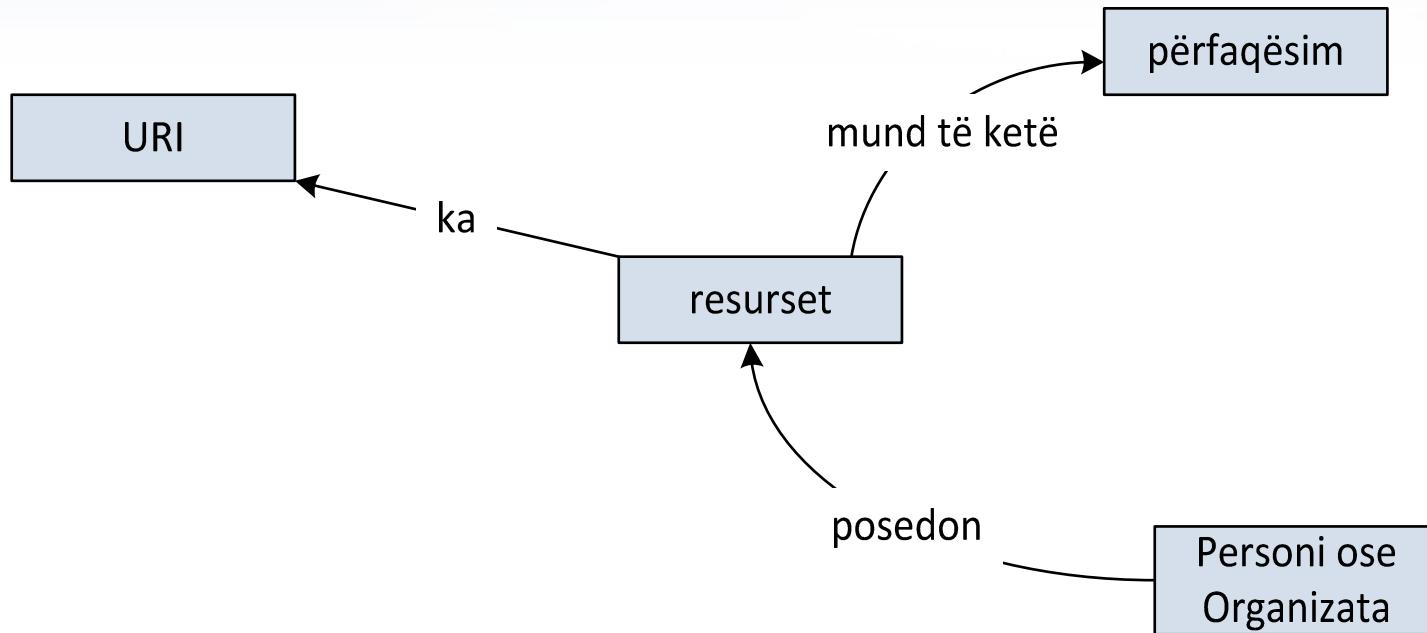


Fig 3. Modeli i thjeshtuar resource-oriented

Modeli i politikave zhvilluese

Prof. Ass. Dr. Isak Shabani

- Fokusohet në kushtëzimet në sjelljen e agjentëve dhe të shërbimeve.

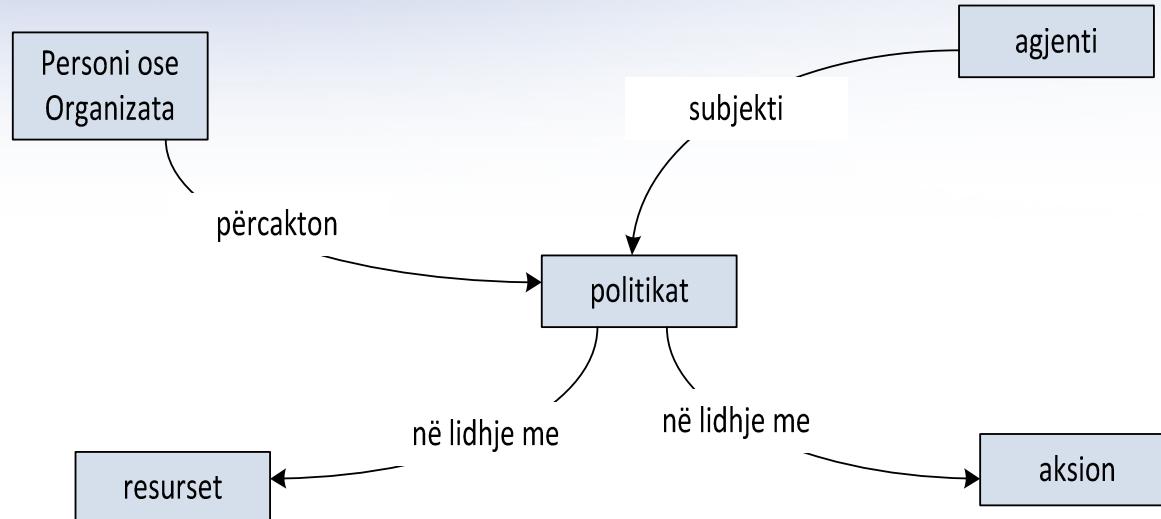
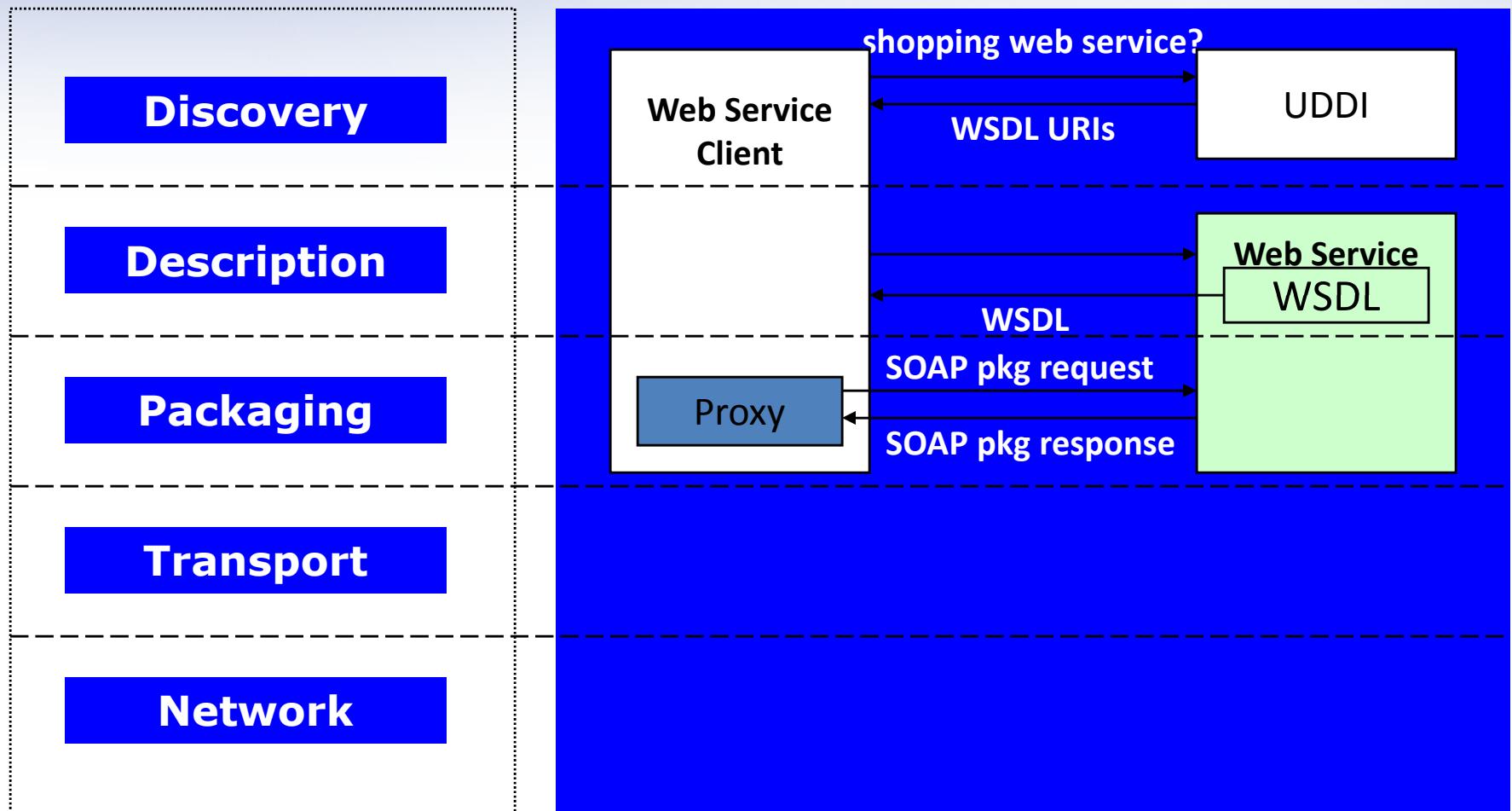


Fig 4.
Modeli i thjeshtuar i politikave zhvilluese

- Politikat zhvilluese bëhen për resurse (burime).
- Aplikohen te agjentët që tentojnë të qasen në këto resurse.
- Vendosen në një vend ose zhvillohen nga njerëzit që kanë përgjegjësi për këto resurse.
- Politikat zhvilluese mund të miratohen për të treguar çështjet e sigurisë, kualitetit, menaxhimit dhe aplikimit të shërbimeve.

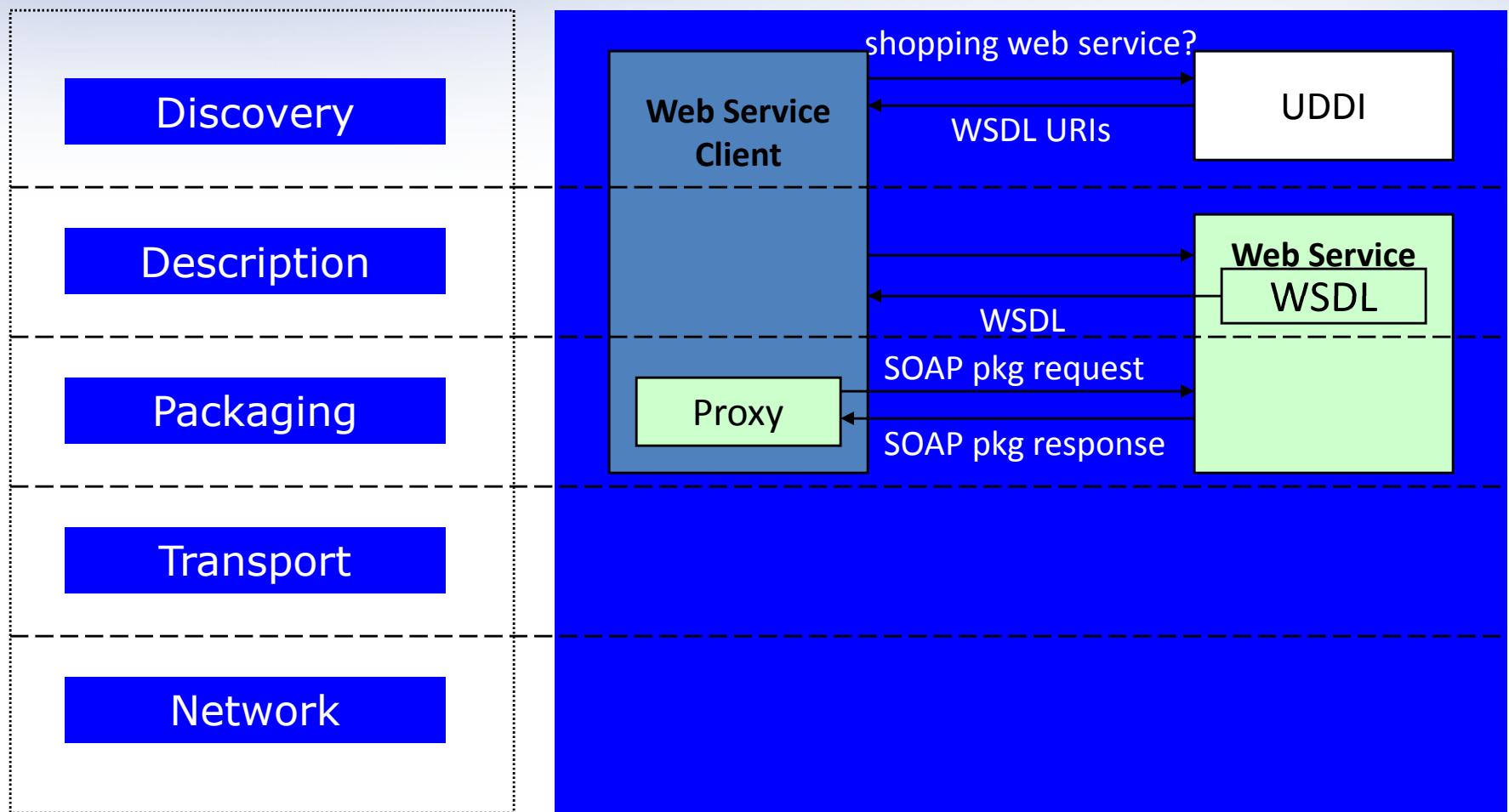
Step3. Write Proxy to Access Web Service

Prof. Ass. Dr. Isak Shabani



Step4. Write Client to Invoke Proxy

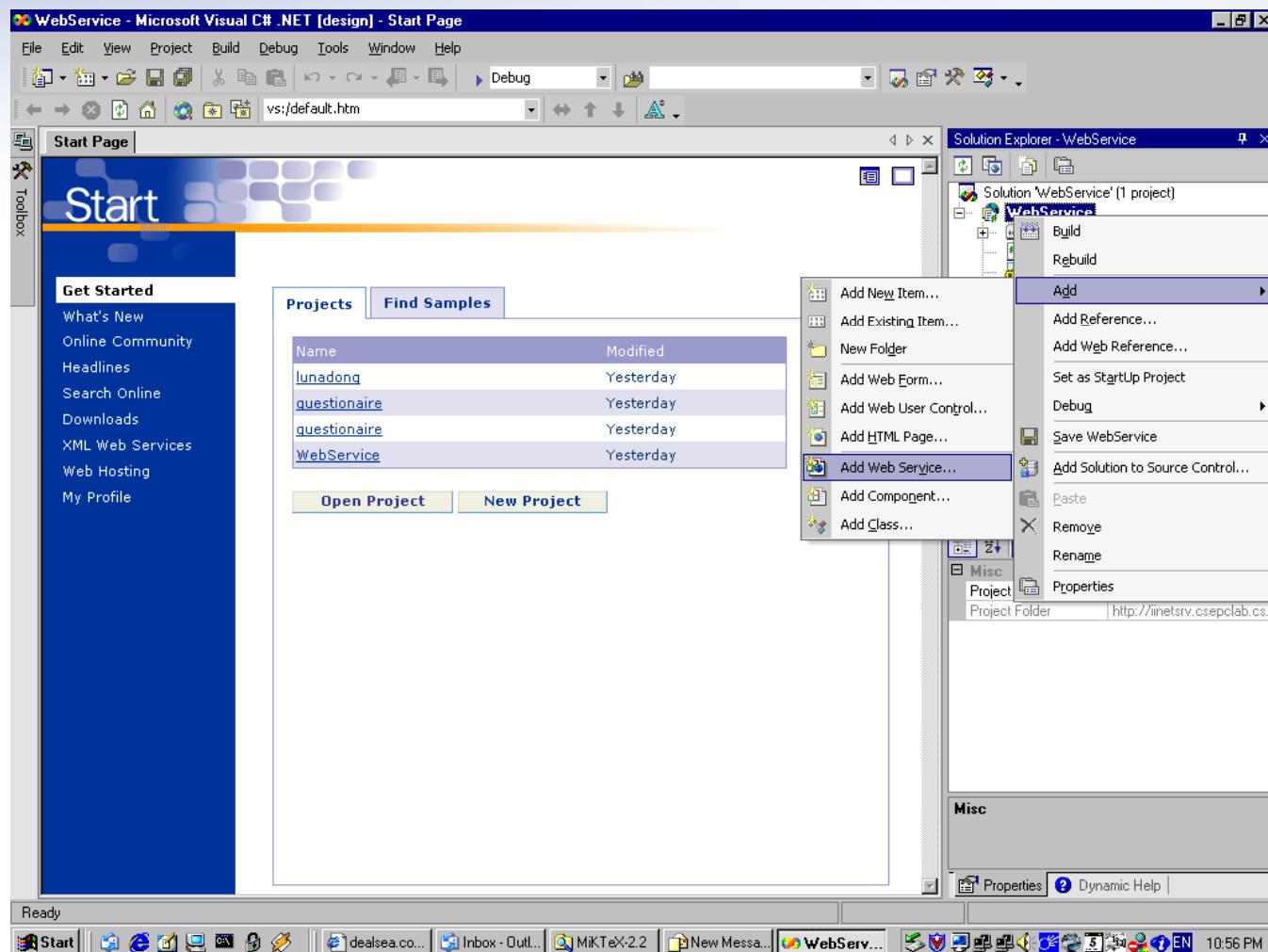
Prof. Ass. Dr. Isak Shabani



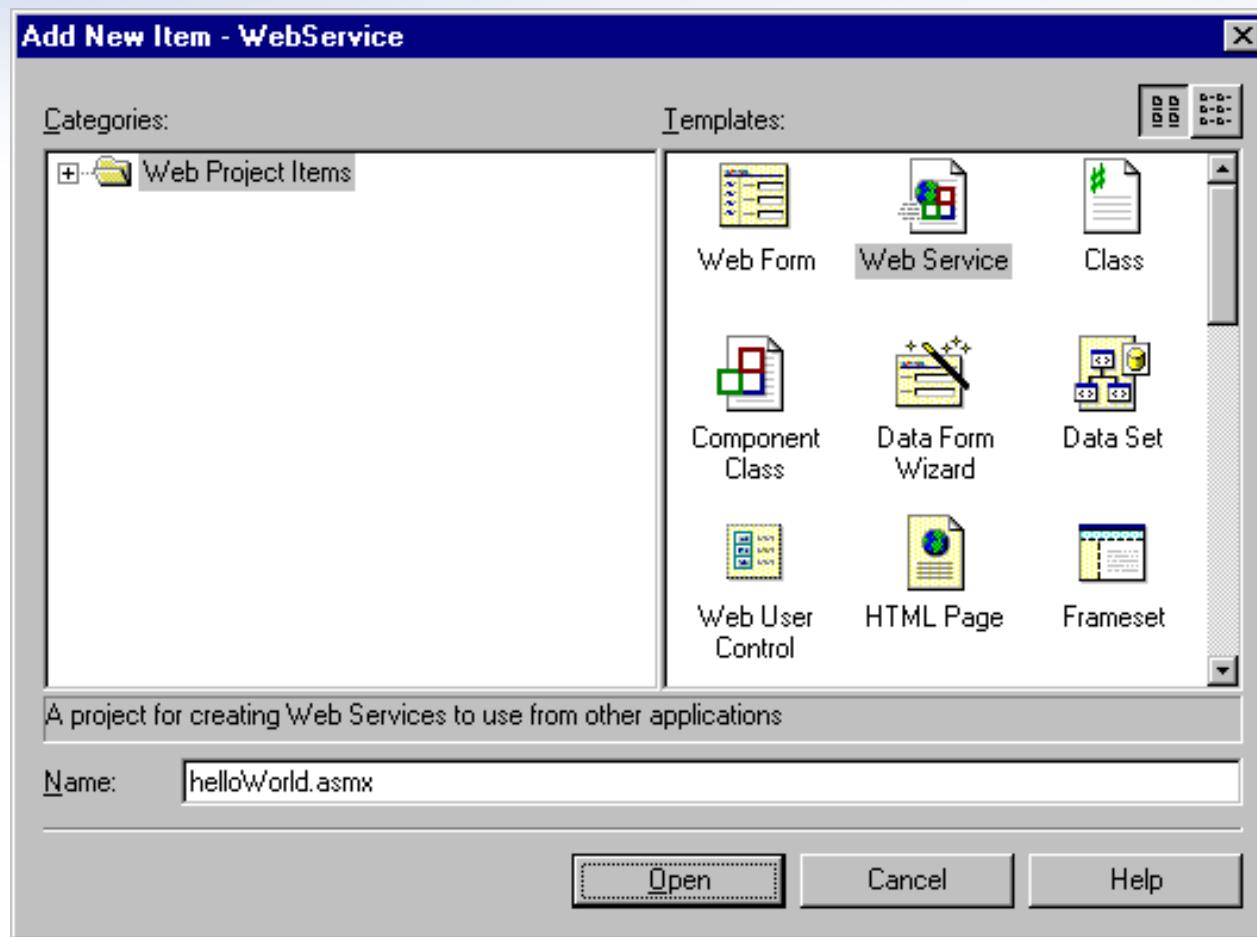
Step1. Create a Web Service

- Functionality is implemented in .asmx.vb or .asmx.cs files.
 - <%@WebService Language="C#" Class="helloWorld" %>
- Use System.Web.Services Namespace
 - Using System.Web.Services
- Inherit your own class from WebService Base Class
 - public class helloWorld : System.Web.Services.WebService
- Declare the WebMethod Attribute
 - [WebMethod]
public string HelloWorld(string name)
{...}

Creating a Web Service with .Net

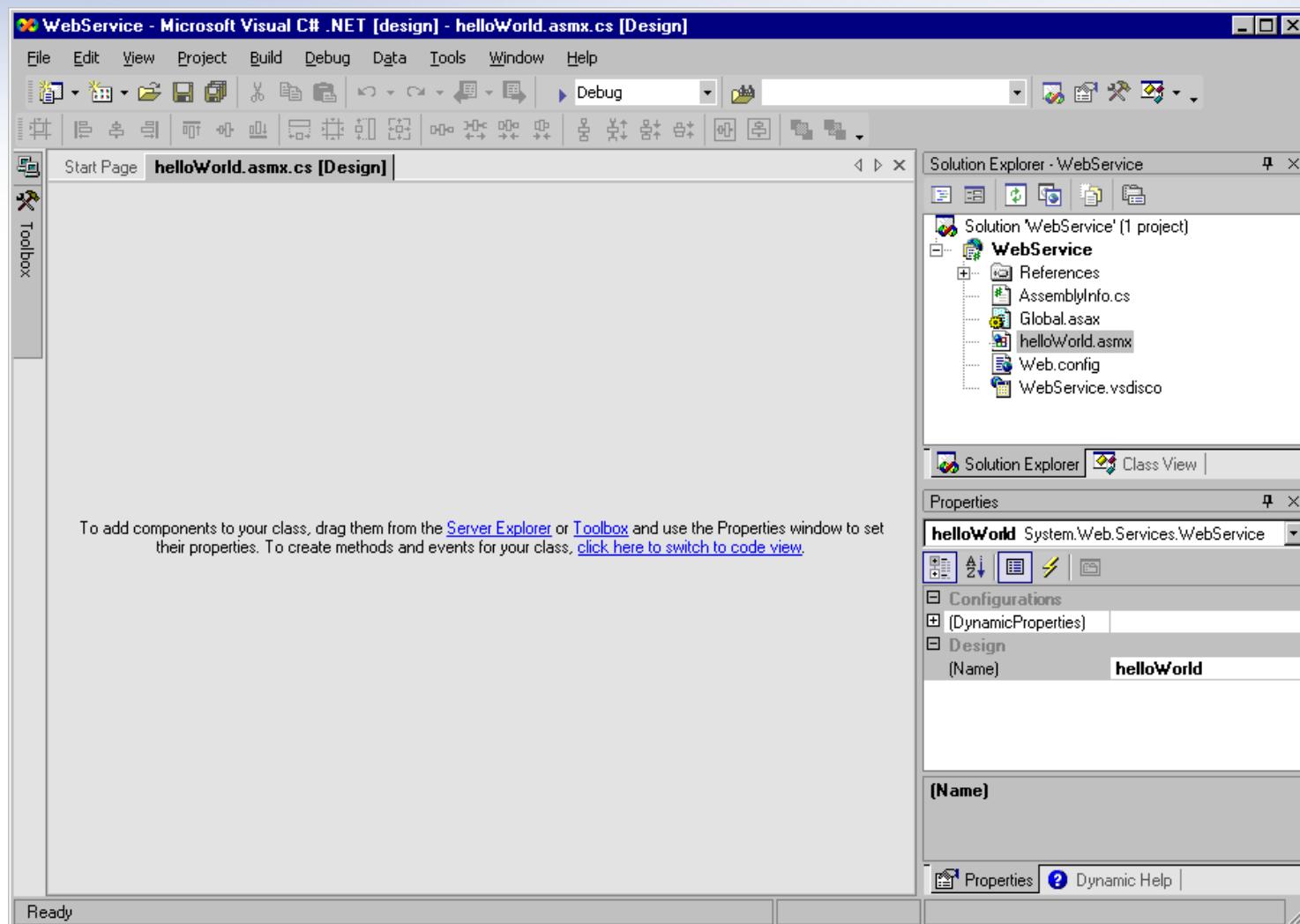


Creating a Web Service with .Net

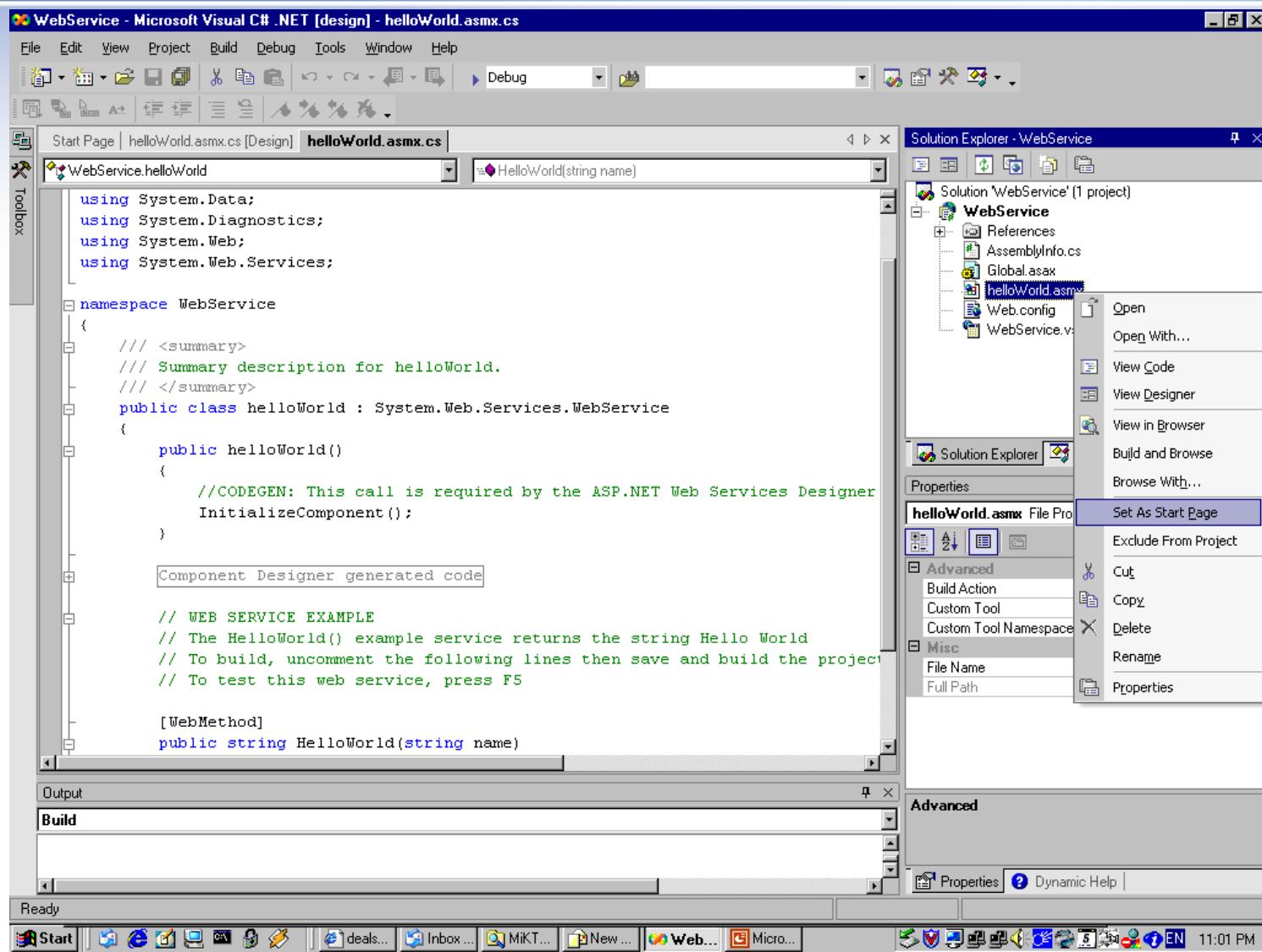


Creating a Web Service with .Net

Prof. Ass. Dr. Isak Shabani



Creating a Web Service with .Net



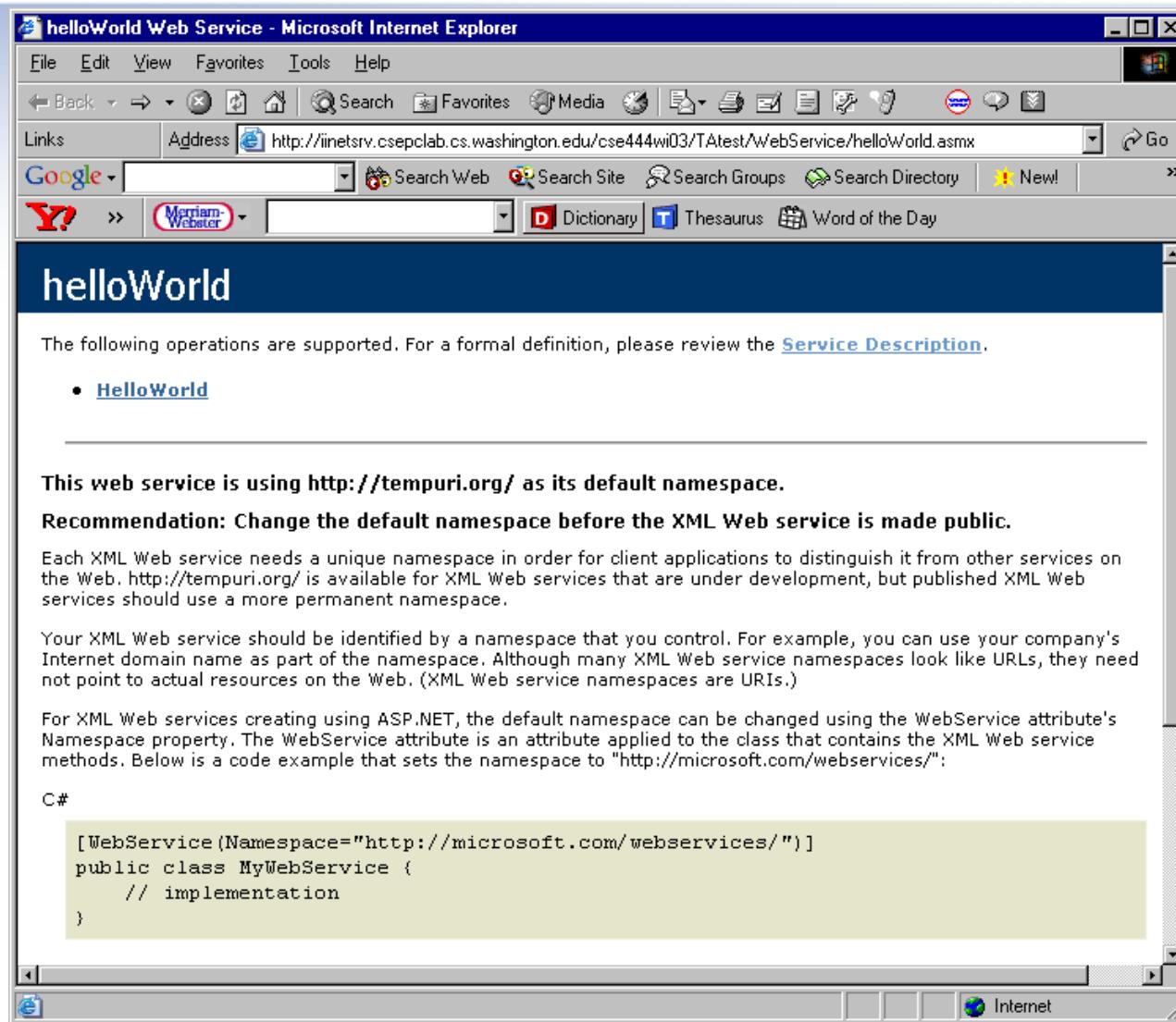
Creating a Web Service with .Net

Prof. Ass. Dr. Isak Shabani

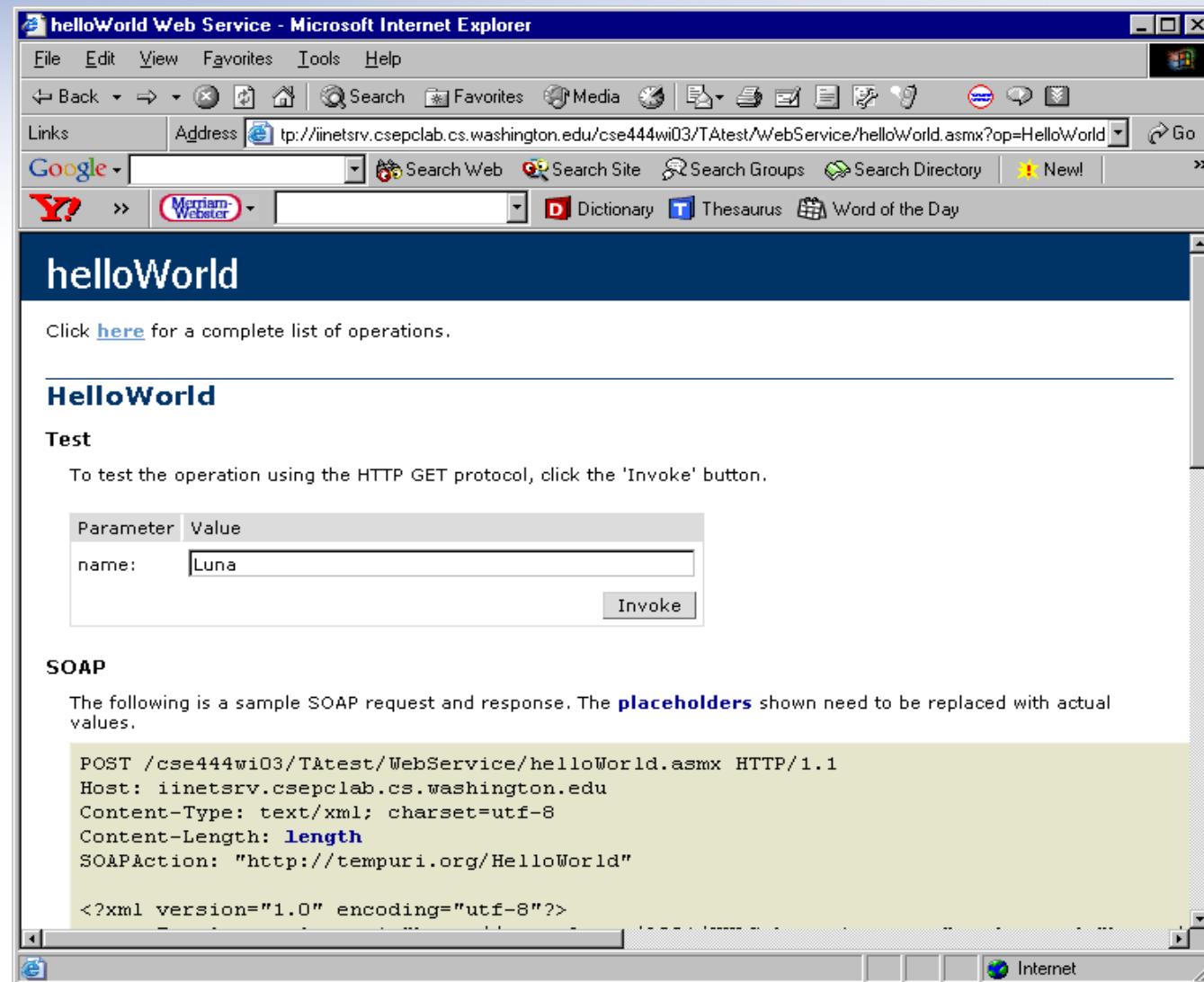
[WebMethod]

```
public string HelloWorld(string name)  
{  
    return "Hello " + name;  
}
```

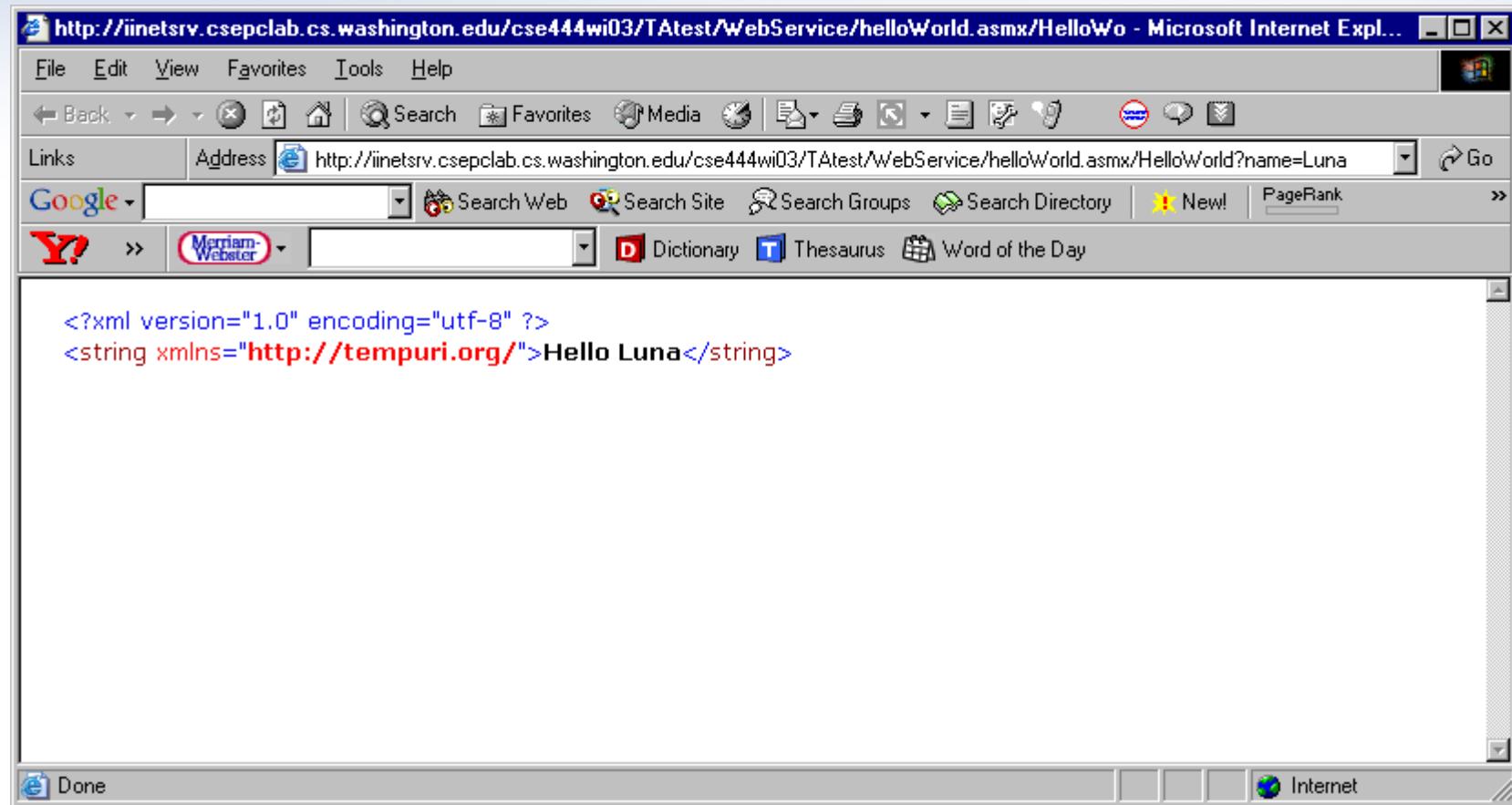
Compile and View Your Web Service



Compile and View Your Web Service



Compile and View Your Web Service



Step2. Describe Web Service using WSDL

- WSDL (Web Services Description Language)
- Describes 3 ways to access web service: GET, POST, SOAP
- Element:
 - <types>: XML schema for input/output
 - <message>:
 - **HelloWorldSoapIn, ...SoapOut, ...HttpGetIn, ...HttpGetOut, ...HttpPostIn, ...HttpPostOut**
 - <porttype>
 - **helloWorldSoap, ...HttpGet, ...HttpPost**
 - <binding>
 - **s0:helloWorldSoap, s0:...HttpGet, s0:...HttpPost**
 - <service name = “helloWorld”>
 <port name = “...Soap” binding = “s0:...Soap”>
 <soap:address location = “http://...”/>
 </port></service>

WSDL Generated by .NET

The screenshot shows a Microsoft Internet Explorer window titled "helloWorld Web Service - Microsoft Internet Explorer". The address bar contains the URL <http://inetsrv.csepclab.cs.washington.edu/cse444wi03/TATest/WebService/helloWorld.asmx>. The page content is as follows:

helloWorld

The following operations are supported. For a formal definition, please review the [Service Description](#).

- [HelloWorld](#)

This web service is using <http://tempuri.org/> as its default namespace.

Recommendation: Change the default namespace before the XML Web service is made public.

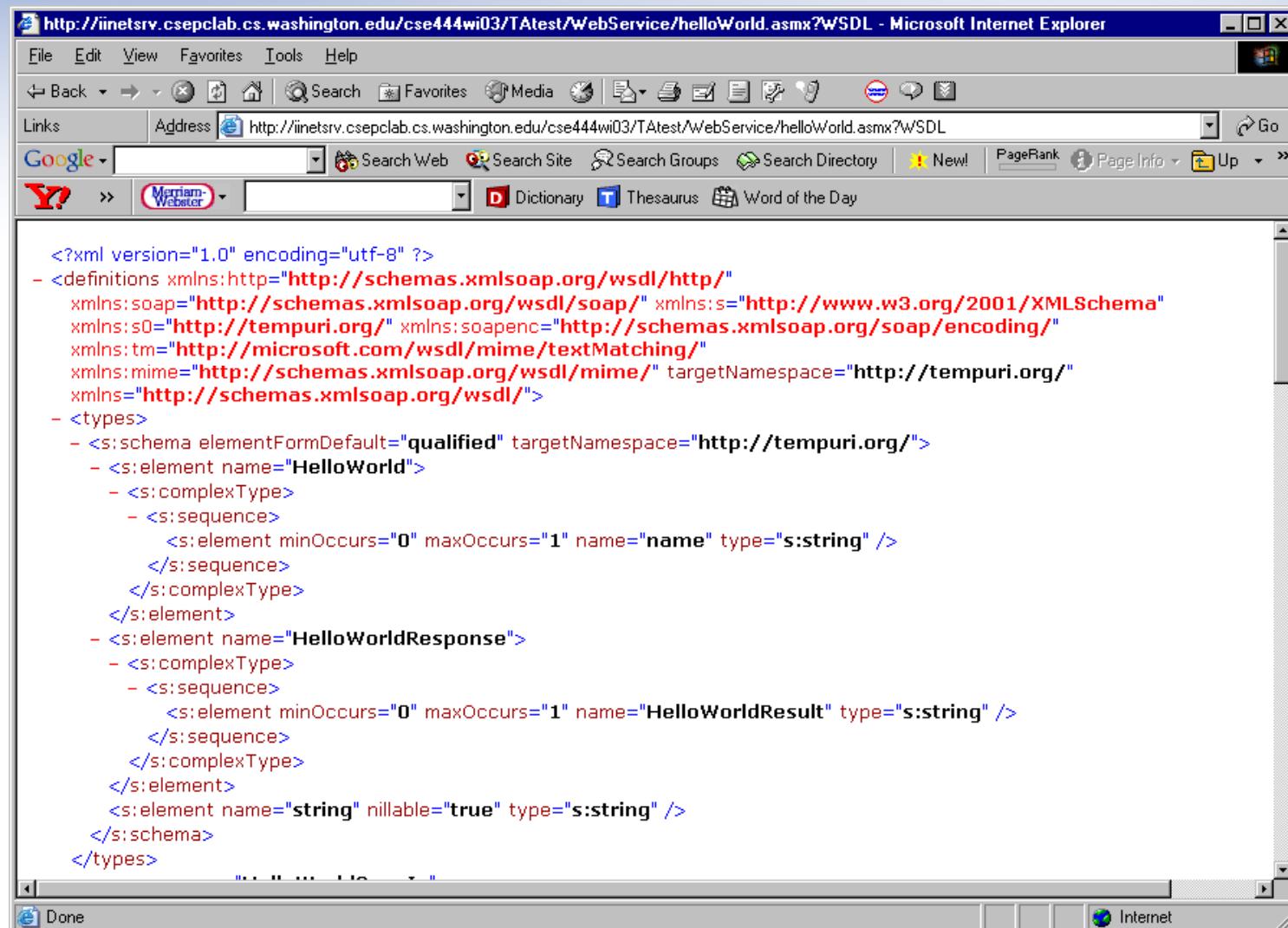
Each XML Web service needs a unique namespace in order for client applications to distinguish it from other services on the Web. <http://tempuri.org/> is available for XML Web services that are under development, but published XML Web services should use a more permanent namespace.

Your XML Web service should be identified by a namespace that you control. For example, you can use your company's Internet domain name as part of the namespace. Although many XML Web service namespaces look like URLs, they need not point to actual resources on the Web. (XML Web service namespaces are URIs.)

For XML Web services created using ASP.NET, the default namespace can be changed using the WebService attribute's Namespace property. The WebService attribute is an attribute applied to the class that contains the XML Web service methods. Below is a code example that sets the namespace to "<http://microsoft.com/webservices/>":

```
C#
[WebService(Namespace="http://microsoft.com/webservices/")]
public class MyWebService {
    // implementation
}
```

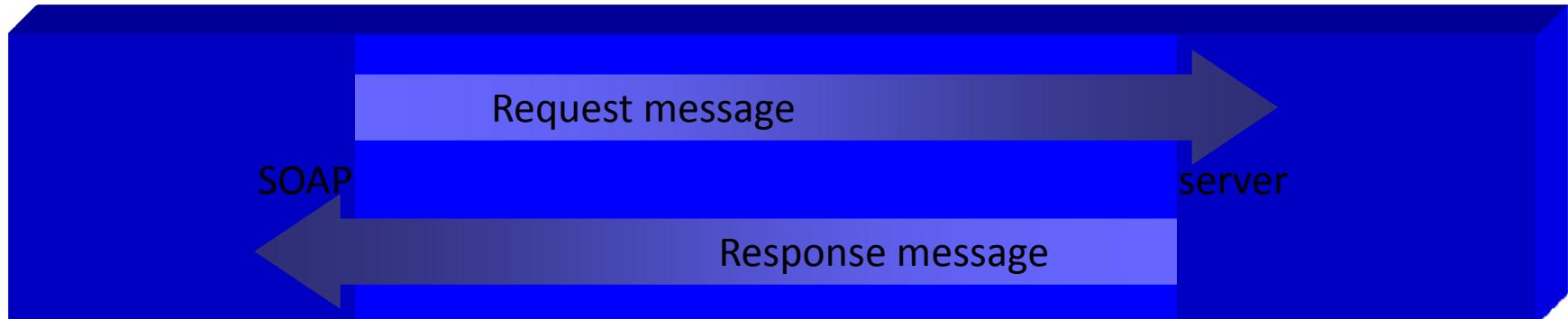
WSDL Generated by .NET



The screenshot shows a Microsoft Internet Explorer window displaying the WSDL (Web Services Description Language) for a .NET web service. The URL in the address bar is `http://iinetsrv.csepclab.cs.washington.edu/cse444wi03/TATest/WebService/helloWorld.asmx?WSDL`. The page content is the XML WSDL code, which defines the service interface and its operations.

```
<?xml version="1.0" encoding="utf-8" ?>
- <definitions xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:s="http://www.w3.org/2001/XMLSchema"
  xmlns:s0="http://tempuri.org/" xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/"
  xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/" targetNamespace="http://tempuri.org/"
  xmlns="http://schemas.xmlsoap.org/wsdl/">"
- <types>
- <s:schema elementFormDefault="qualified" targetNamespace="http://tempuri.org/">
- <s:element name="HelloWorld">
- <s:complexType>
- <s:sequence>
<s:element minOccurs="0" maxOccurs="1" name="name" type="s:string" />
</s:sequence>
</s:complexType>
</s:element>
- <s:element name="HelloWorldResponse">
- <s:complexType>
- <s:sequence>
<s:element minOccurs="0" maxOccurs="1" name="HelloWorldResult" type="s:string" />
</s:sequence>
</s:complexType>
</s:element>
<s:element name="string" nillable="true" type="s:string" />
</s:schema>
</types>
```

- SOAP Messages
- Using SOAP as RPC (Remote Procedure Call) Messages



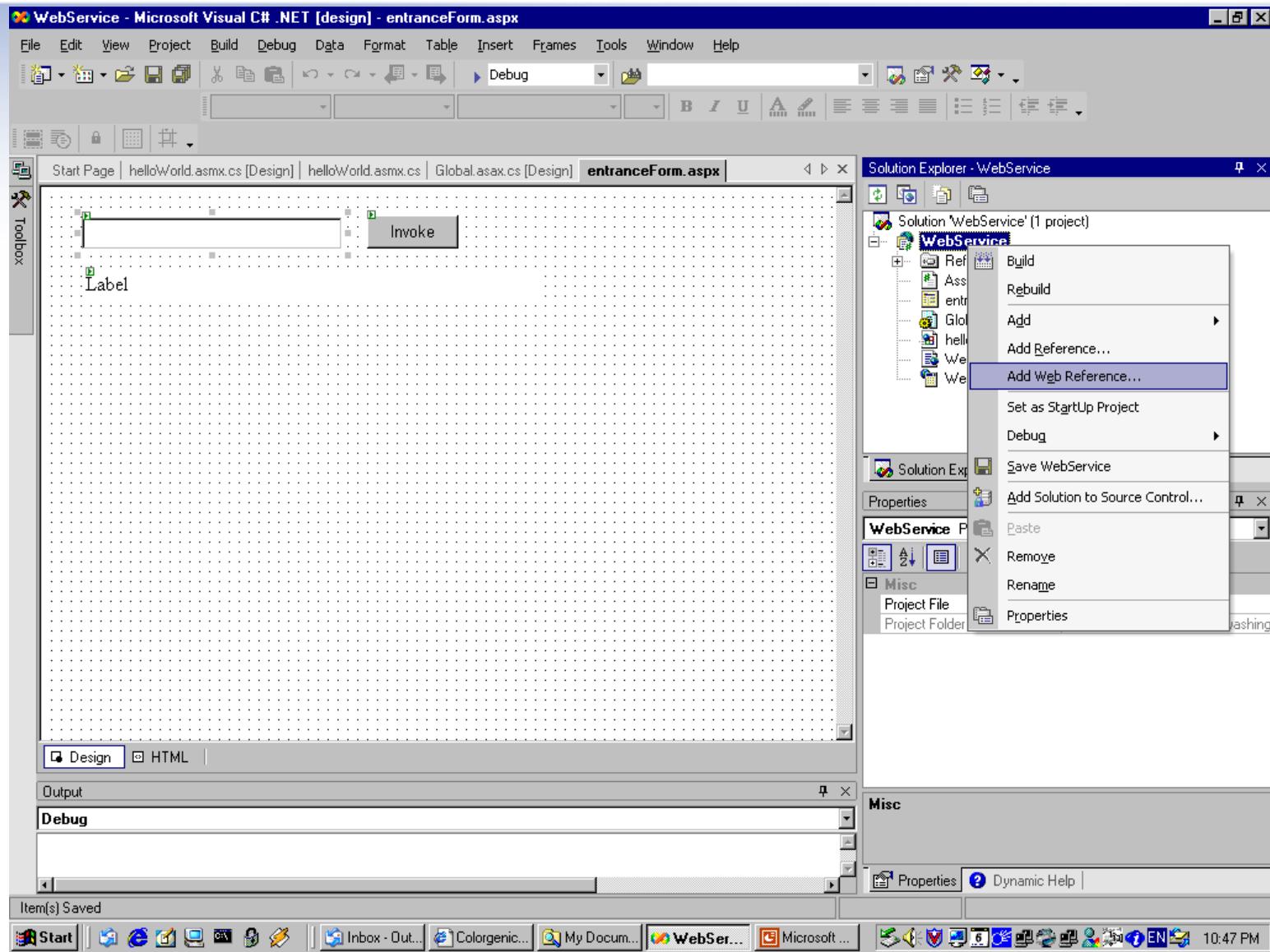
Step3. Write Proxy to Access Web Service

Prof. Ass. Dr. Isak Shabani

- In software, a proxy is the code that does work on behalf of other code.
 - **For web service users**, the proxy represents the web service the users wish to call
 - **For web service servers**, the proxy makes request on the behalf of the user

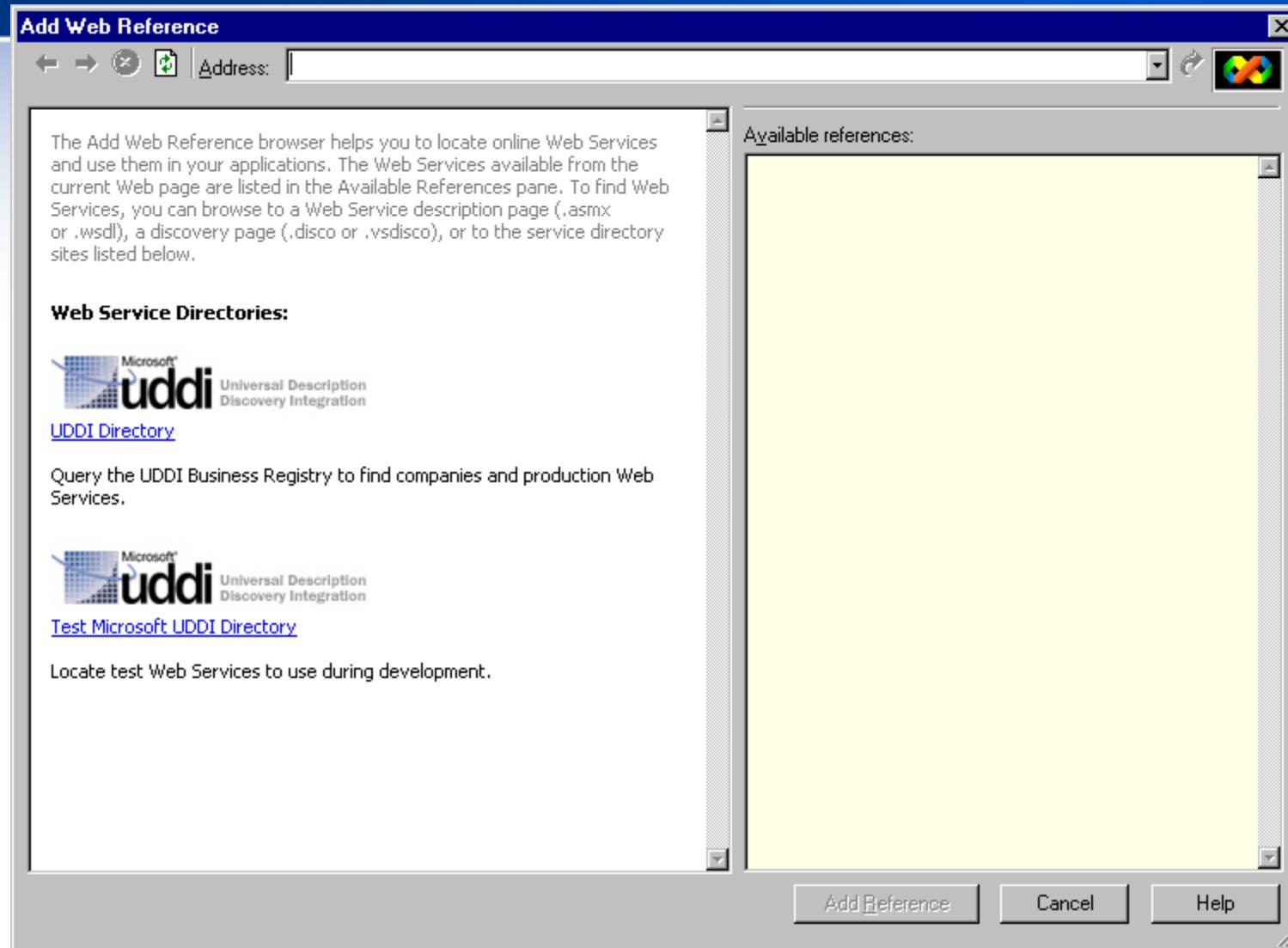
Create a Proxy (Web Reference) with .Net

Prof. Ass. Dr. Isak Shabani



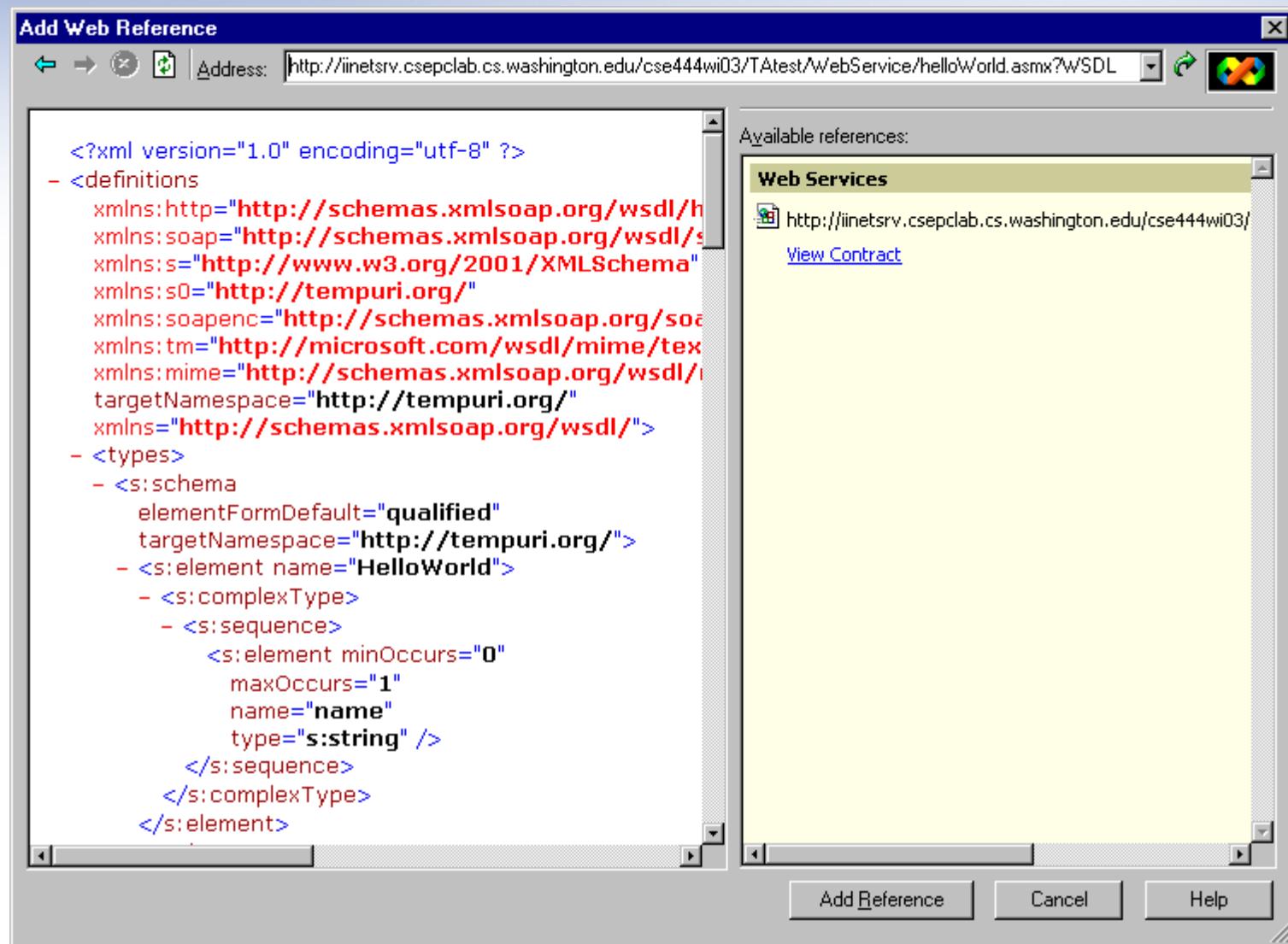
Create a Proxy (Web Reference) with .Net

Prof. Ass. Dr. Isak Shabani



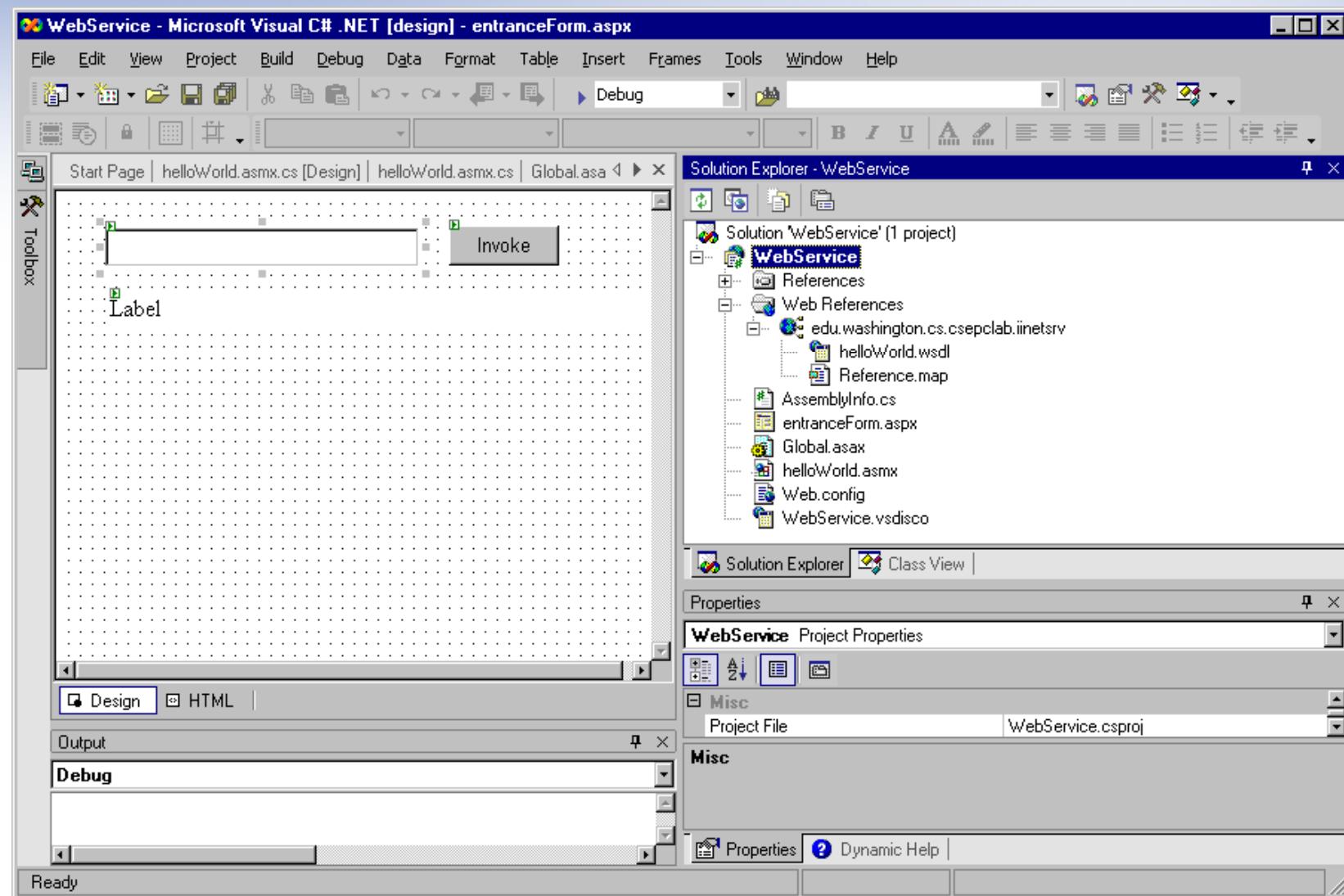
Create a Proxy (Web Reference) with .Net

Prof. Ass. Dr. Isak Shabani

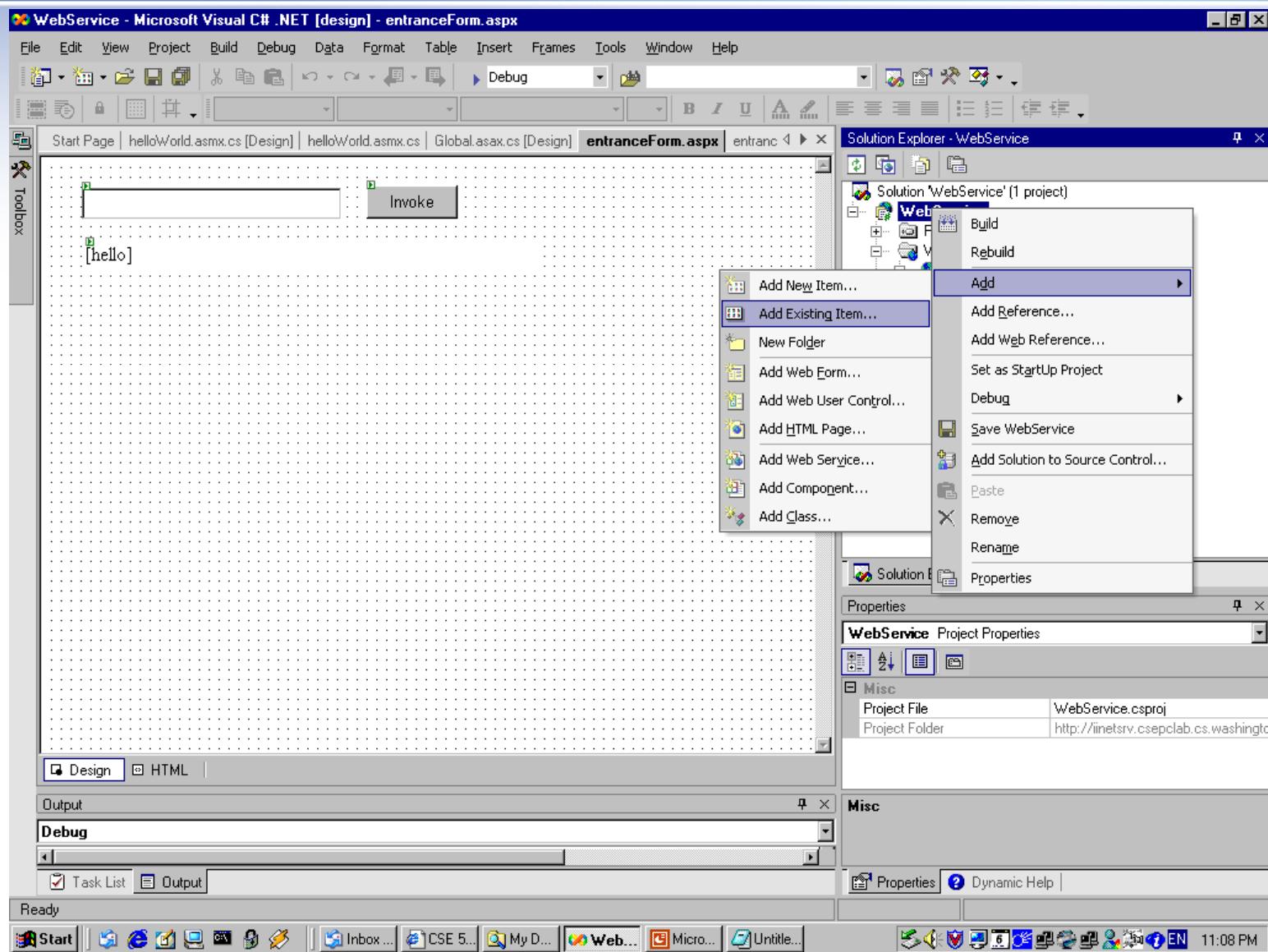


Create a Proxy (Web Reference) with .Net

Prof. Ass. Dr. Isak Shabani

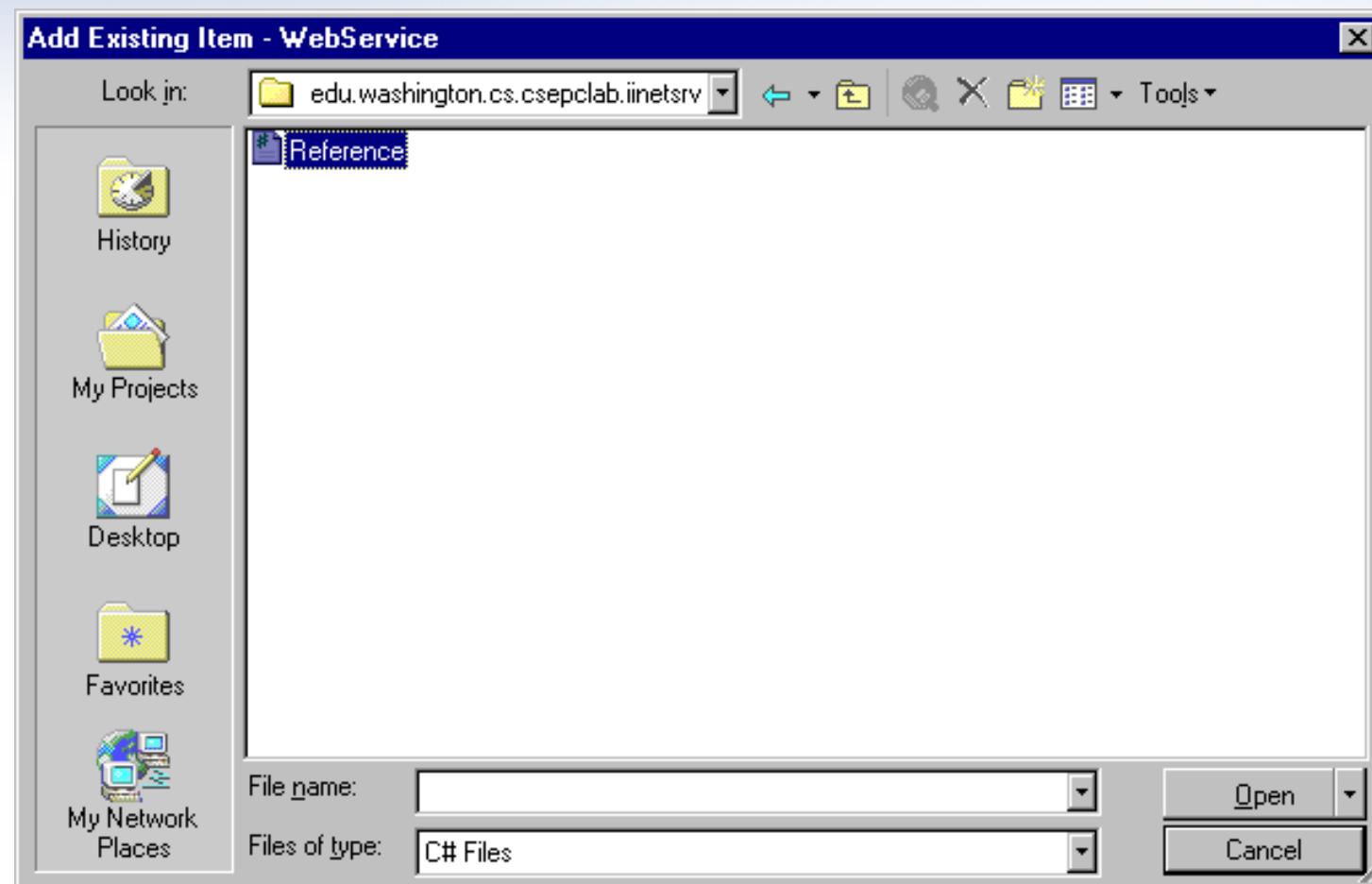


Read Code for Proxies

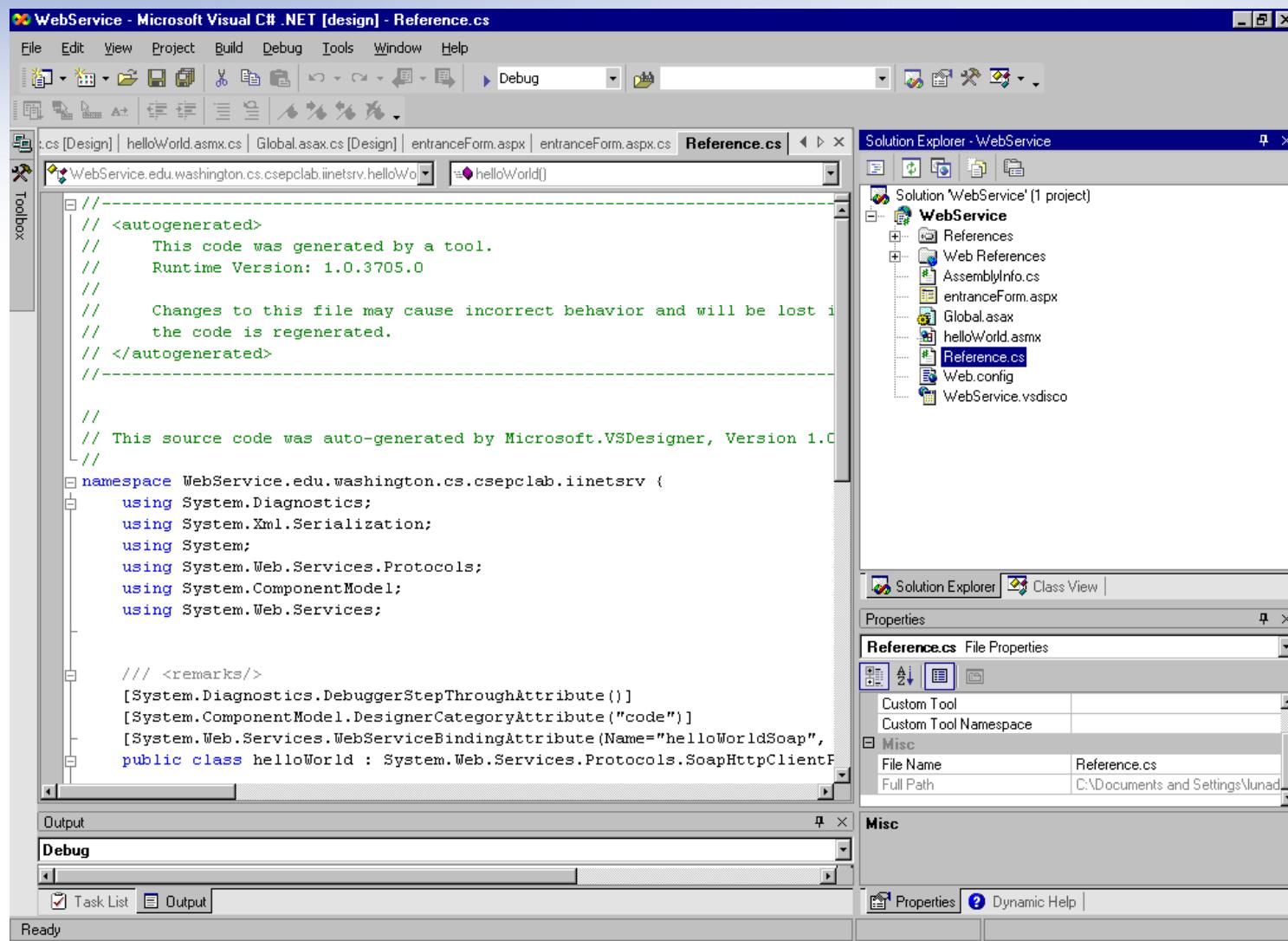


Read Code for Proxies

Prof. Ass. Dr. Isak Shabani



Read Code for Proxies



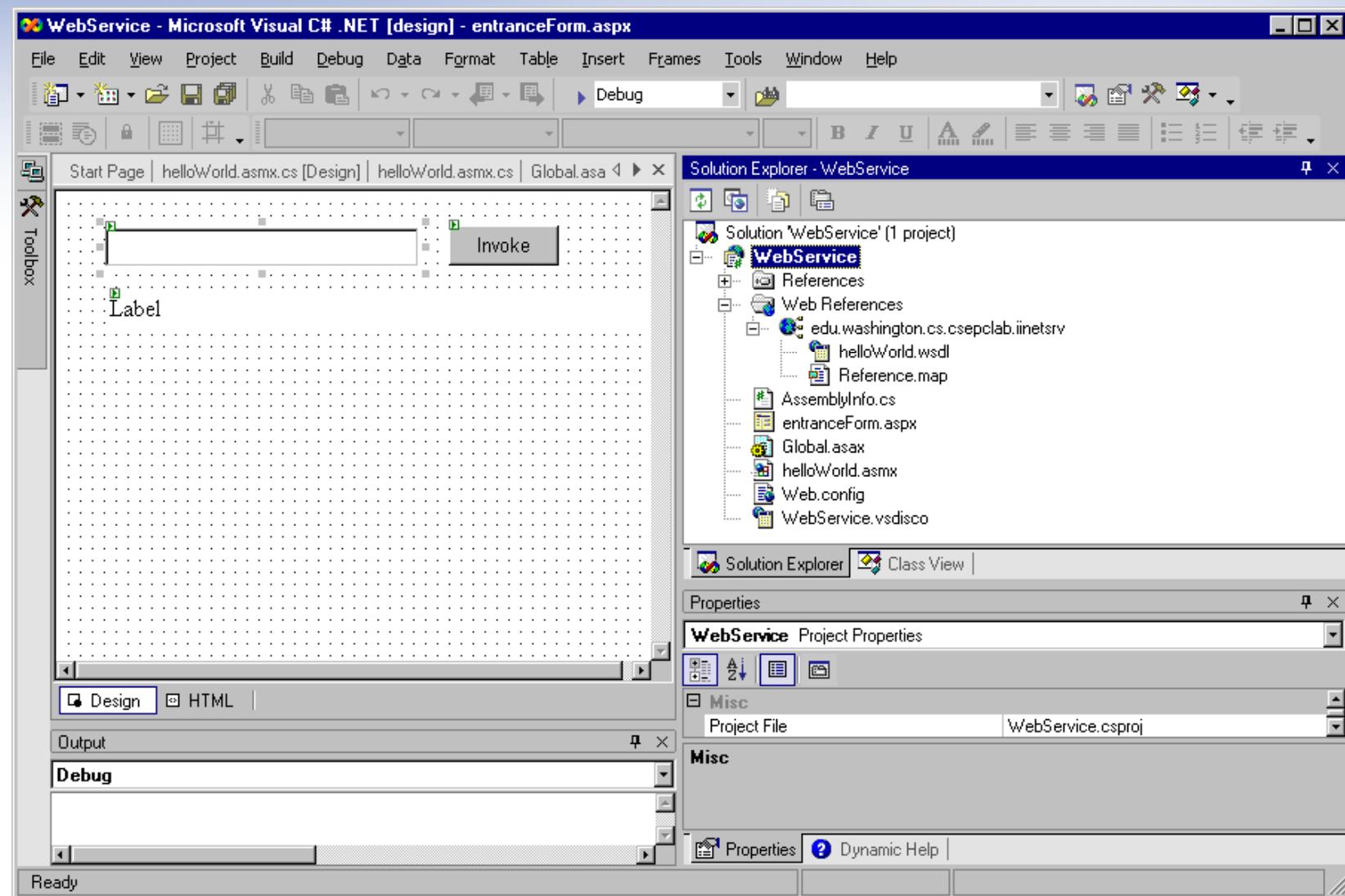
Read Code for Proxies

Prof. Ass. Dr. Isak Shabani

```
public class helloWorld : System.Web.Services.  
    Protocols.SoapHttpClientProtocol  
{  
    public helloWorld() {  
        this.Url = "http://iinetsrv.csepclab.cs.washington  
            .edu/cse444wi03/TATest/WebService/helloWorld.  
            asmx";  
    }  
    ...
```

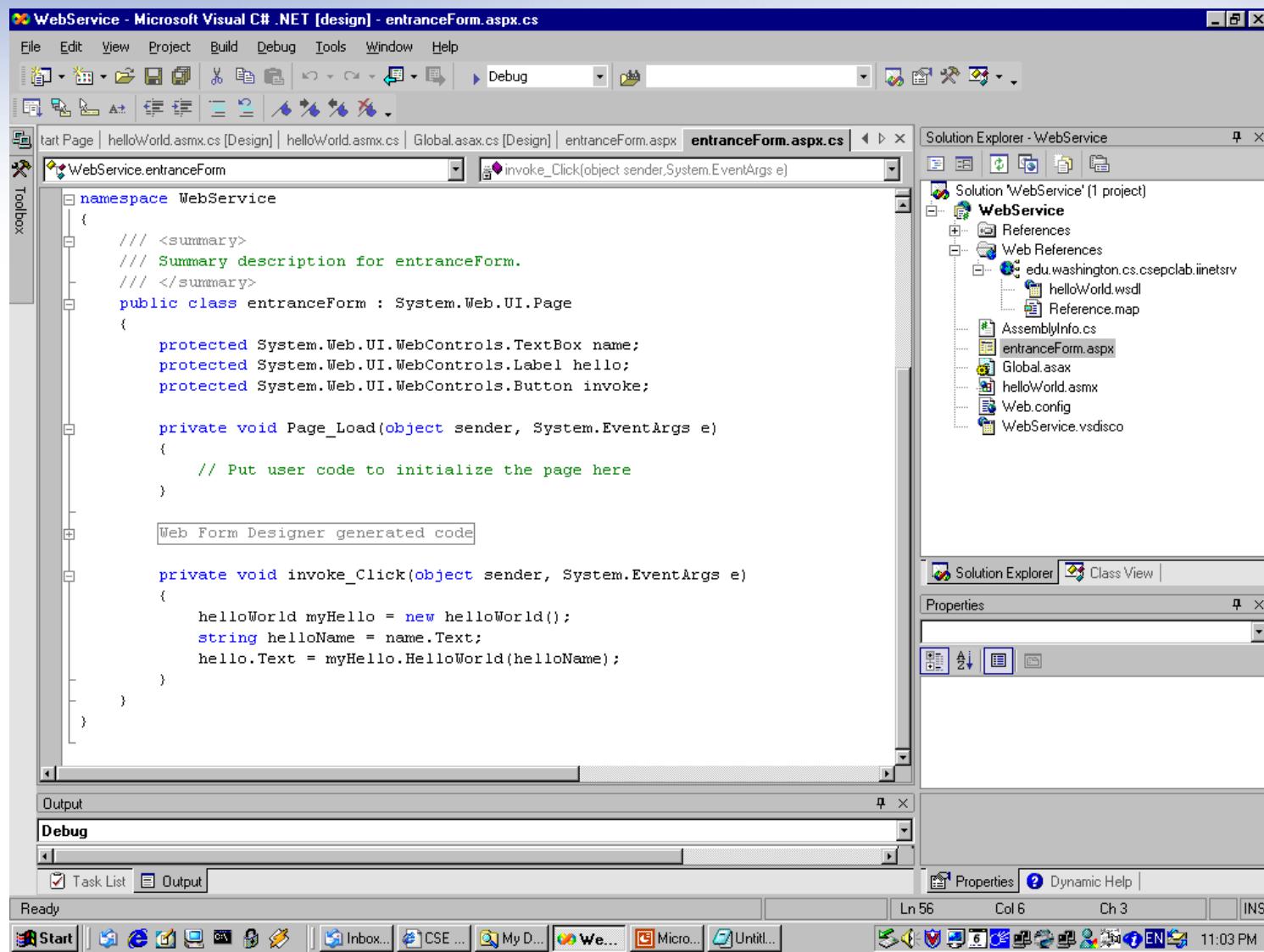
Step4. Write a Client to Invoke Proxy

Prof. Ass. Dr. Isak Shabani



Step4. Write a Client to Invoke Proxy

Prof. Ass. Dr. Isak Shabani

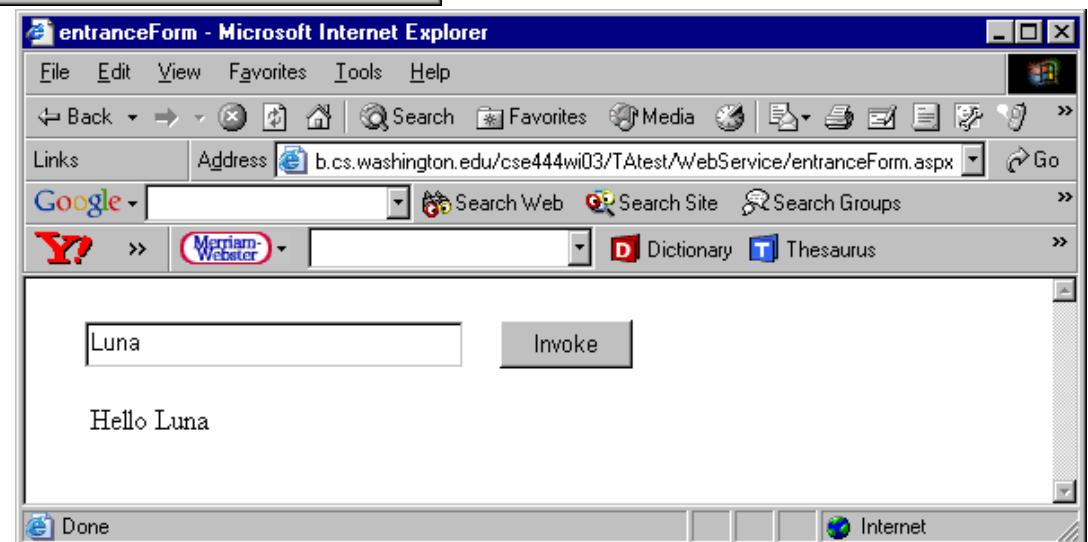
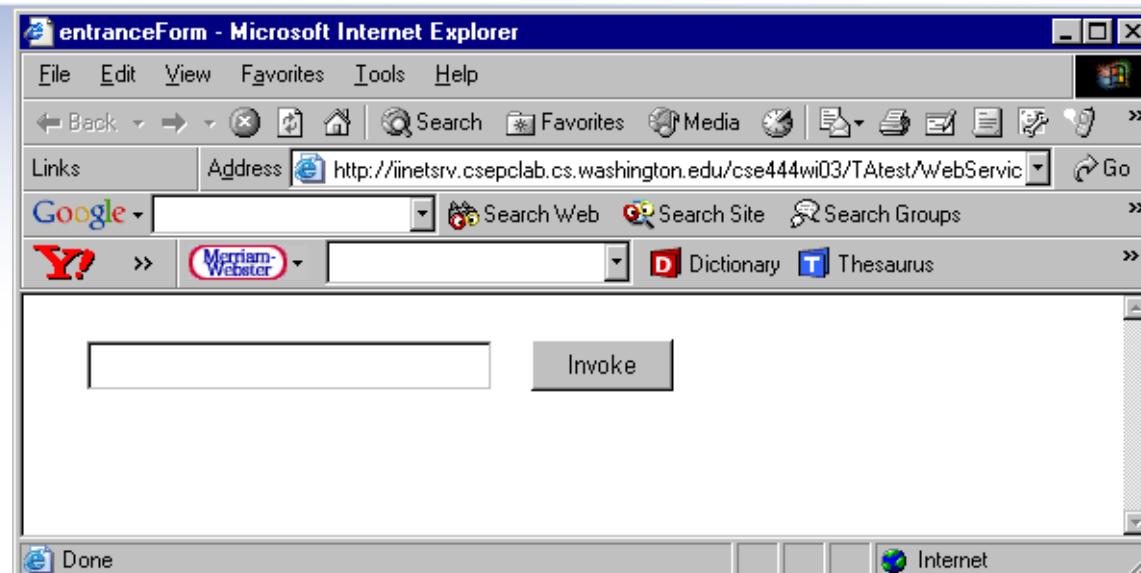


Step4. Write a Client to Invoke Proxy

```
private void invoke_Click(object sender, System.EventArgs e)
{
    helloWorld myHello = new helloWorld();
    string helloName = name.Text;
    hello.Text = myHello.HelloWorld(helloName);
}
```

Step4. Write a Client to Invoke Proxy

Prof. Ass. Dr. Isak Shabani



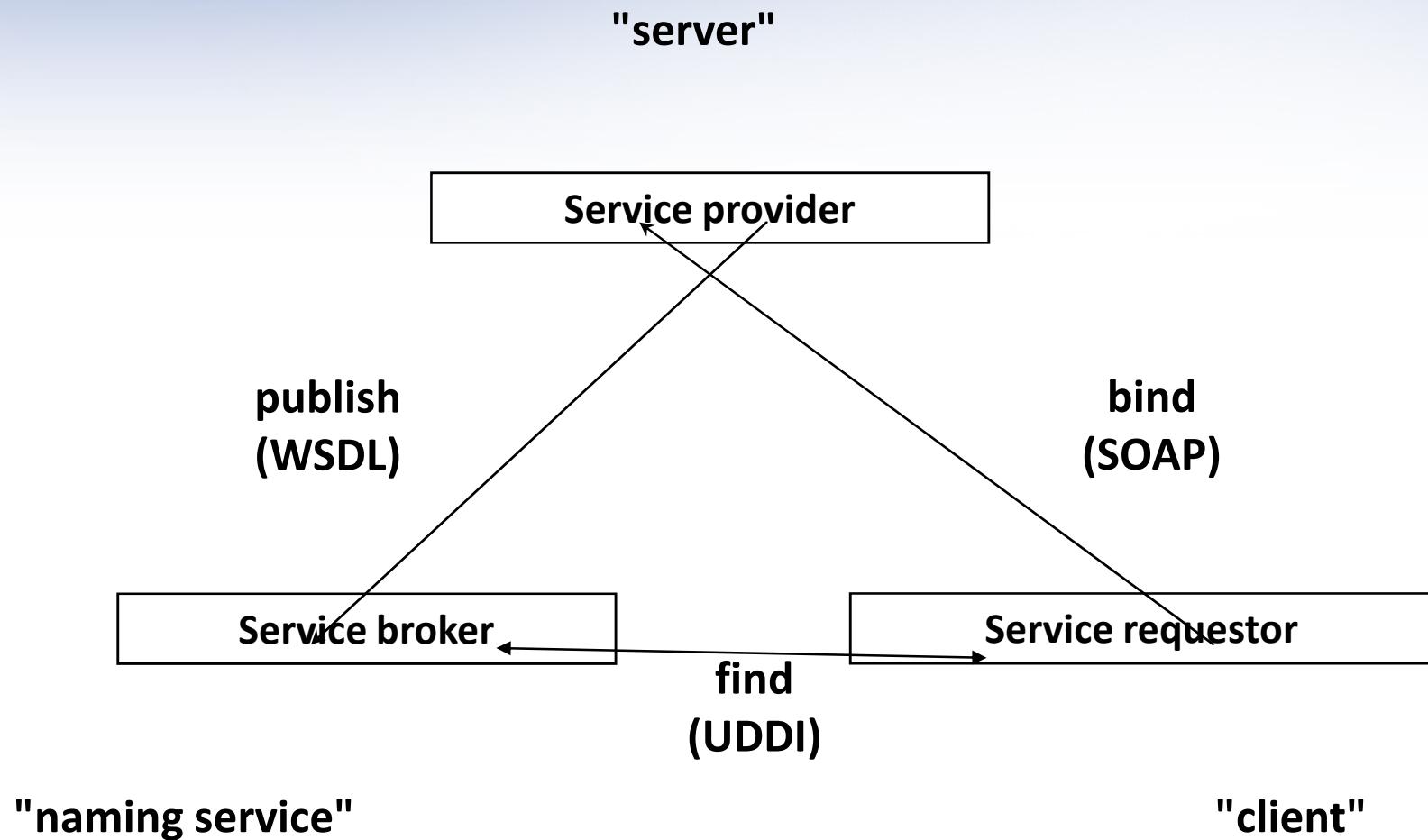
Evolulimi i Ueb-it

Prof. Ass. Dr. Isak Shabani

Teknologja	TCP/IP	HTML	XML
Qëllimi	Connectivity	Presentation	Programmability
Aplikacionet	E-Mail, FTP...	Web Pages	Web Services
Rezultati	Create the Web	Browse the Web	Program the Web

Arkitektura e Ueb Sherbimeve

Prof. Ass. Dr. Isak Shabani



Web Service Stack

Prof. Ass. Dr. Isak Shabani

- A set of standards for implementing web services

Publication and Discovery: **UDDI**

extends URI

Service Description: **WSDL**

extends HTML

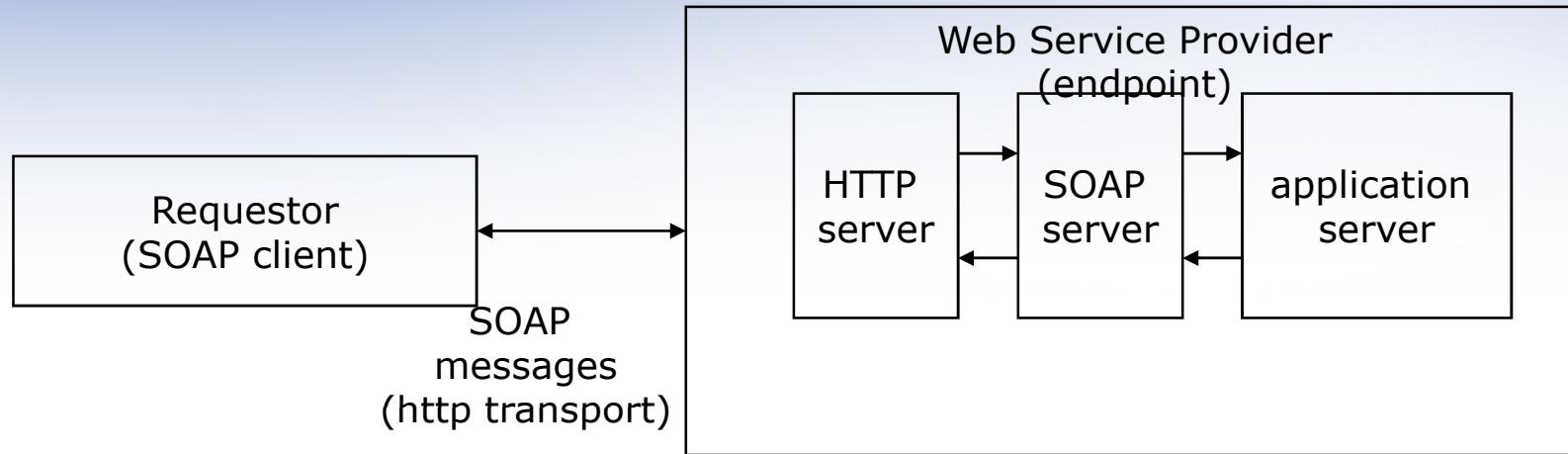
Messaging: **SOAP**

extends HTTP

Transport: HTTP, SMTP, FTP, ...

Web Services Implementation

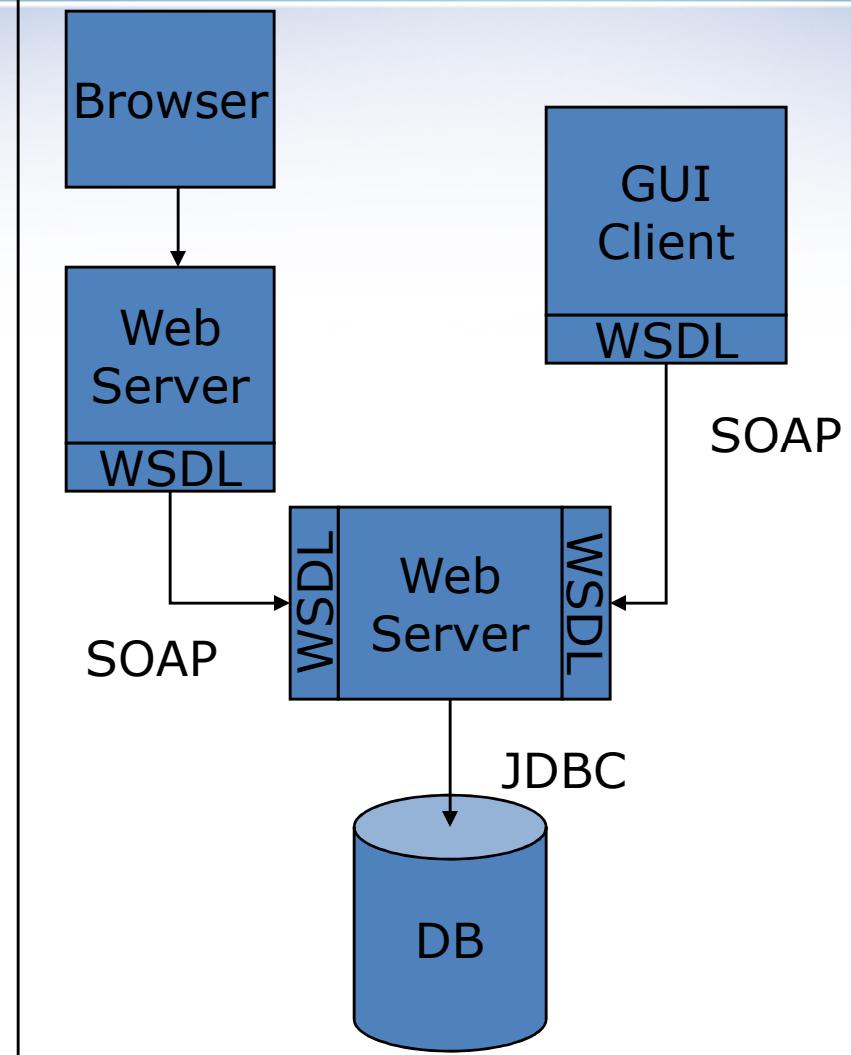
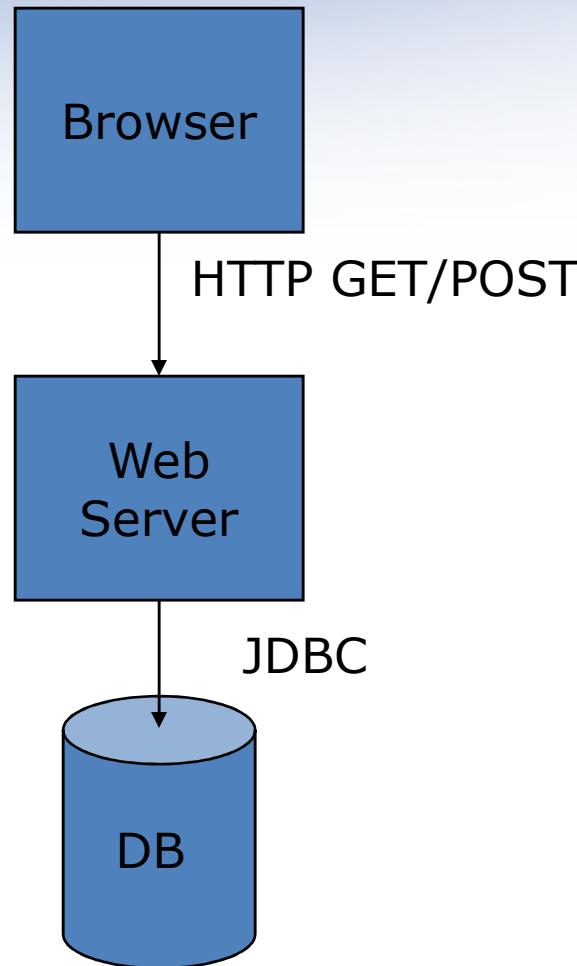
Prof. Ass. Dr. Isak Shabani



- Application Server (web service-enabled)
 - provides implementation of services and exposes it through WSDL/SOAP
 - implementation in Java, as EJB, as .NET (C#) etc.
- SOAP server
 - implements the SOAP protocol
- HTTP server
 - standard Web server
- SOAP client
 - implements the SOAP protocol on the client site

Servlets/CGI Compared to Web Services

Prof. Ass. Dr. Isak Shabani



More Examples of Web Services

Prof. Ass. Dr. Isak Shabani

- Geographical Information Systems are perfect candidates for WS
 - The Open Geospatial Consortium defines several relevant standards
 - Geographic Markup Language (GML) exchanges info.
 - Web Feature Service works with abstract GML feature data.
 - Web Map Service creates maps (images)
 - Lots more at <http://www.opengeospatial.org/specs/?page=specs>
- XMethods
 - Lots and lots of contributed examples, live demos
 - Try them
 - <http://www.xmethods.com/>
- Lots more for bioinformatics.
 - Easiest way to find is to download Taverna from SourceForge.
 - Then check out <http://communitygrids.blogspot.com> for guidelines.
- CICC is building many new one for chemical informatics.

Making RDAHMM into a Web Service

- RDAHMM takes GPS (or other) time-series data as input, along with various command line parameters.
- GPS data comes from GRWS or other services.
 - <http://geoapp.ucsd.edu/sci gnDataPortal/grwsSummar y.jsp>
- It creates 11 output files.
- Results are superimposed on the input time series.

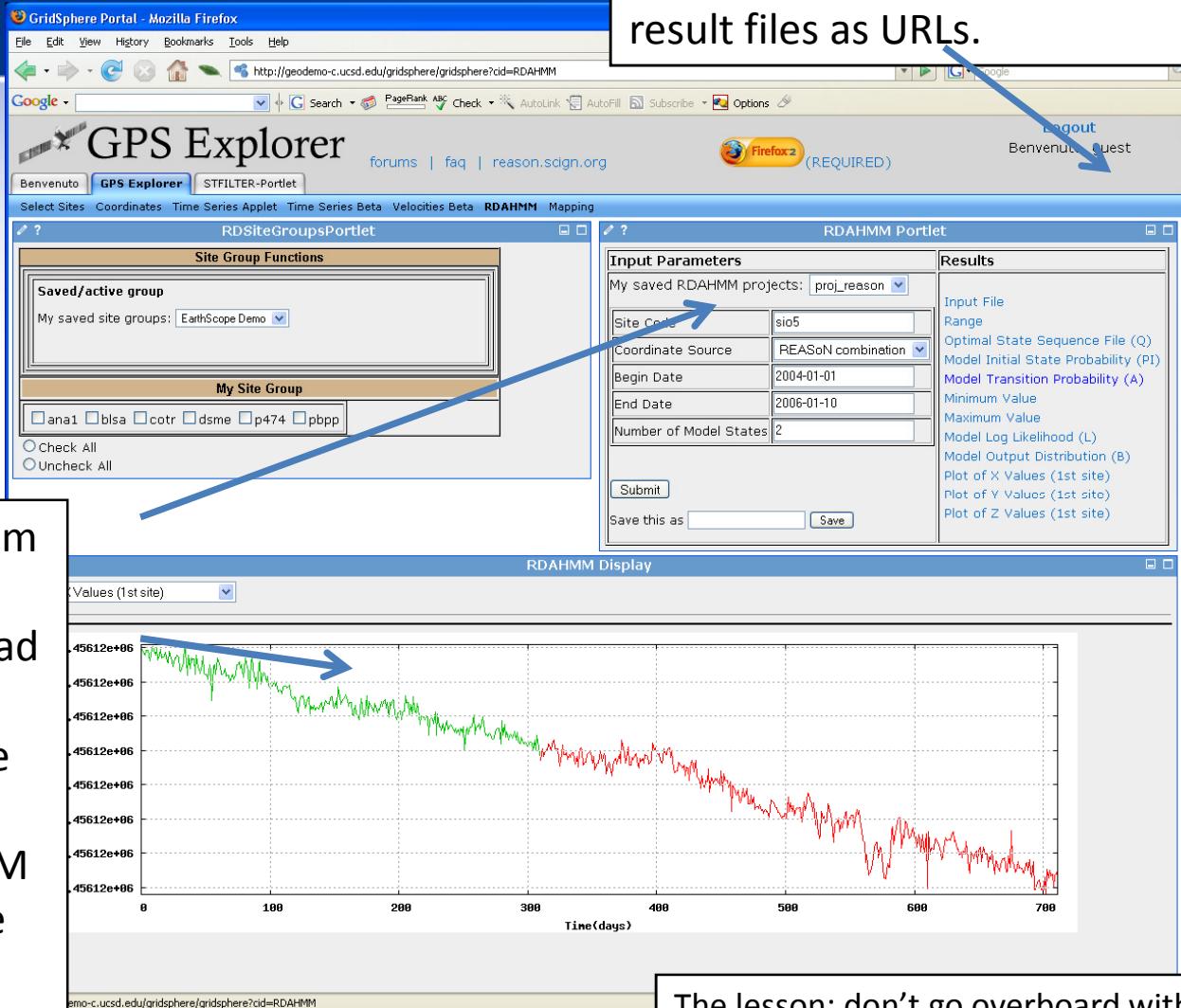
```
USAGE: GEMCodes/RDAHMM2/bin/rdahmm -data 'input  
observation sequence file'  
[-L 'output model log likelihood file']  
[-Q 'output optimal state sequence file']  
[-pi 'output model initial state probability file']  
[-A 'output model transition probability file']  
[-B 'output model output distribution file']  
[-minvalfile 'data minimum value file']  
[-maxvalfile 'data maximum value file']  
[-rangefile 'data range file']  
[-covarsweightsfile 'covariance component weightings file']  
[-covgraphfile 'covariance graph connectivity file']  
-T 'number of observations'  
-D 'dimension of observations'  
-N 'number of model states'  
-output_type 'type of HMM output distribution {gauss}'  
[-init_type 'type of HMM parameter initialization {random}']  
.....
```

Making RDAHMM into a Web Service

Prof. Ass. Dr. Isak Shabani

This is a portal client to a data mining service that I built. The web service

GPS data comes from the Scripps GRWS Web Service. Instead of defining a data type for this file, we just pass around URLs. The RDAHMM service receives the URL as input.



The service returns output result files as URLs.

Portal courtesy of NASA REASoN project.

The lesson: don't go overboard with XML message definitions. You will regret it. Use URLs and keep your SOAP/WSDL simple.

Komunikimi ndërmjet aplikacioneve softuerike të ndërtuara në platforma të ndryshme me ndihmën e Shërbimeve Ueb

Prof. Ass. Dr. Isak Shabani

Që dy aplikacione të komunikojnë me sukses duhet që së pari të plotësohen disa kushte:

- Duhet të ekzistojë lidhja fizike;
- Duhet të ekzistojë një “marrëveshje” në lidhje me formatin e shkëmbimit të të dhënave si p.sh. nëse duhet të bëhet me XML apo diçka tjetër;
- Duhet që të ekzistojë një “marrëveshje” e përbashkët për kuptueshmërinë e të dhënave si p.sh. nëse kemi të bëjmë me prezantim HTML apo me ndonjë transaksion;
- Duhet që të ekzistojë një “marrëveshje” e përbashkët për mënyrën dhe formën e dërgimit të mesazheve ndërmjet palëve.

Komunikimi .NET me J2EE

- Microsoft. NET është platformë e XML Microsoft Web Services.
- Nëpërmjet .NET ofrohet mundësia e krijimit dhe përdorimit të aplikacioneve, proceseve dhe faqeve Ueb si një Shërbim Ueb i bazuar në XML.
- Përmes një rreshti të vetëm të kodit dhe vendosjen e një vlere të një atributi, është e mundur që aplikacioni të kthehet në një shërbim Ueb në mjedisin .NET.

Komunikimi ndërmjet aplikacioneve softuerike të ndërtuara në platforma të ndryshme me ndihmën e Shërbimeve Ueb

Prof. Ass. Dr. Isak Shabani

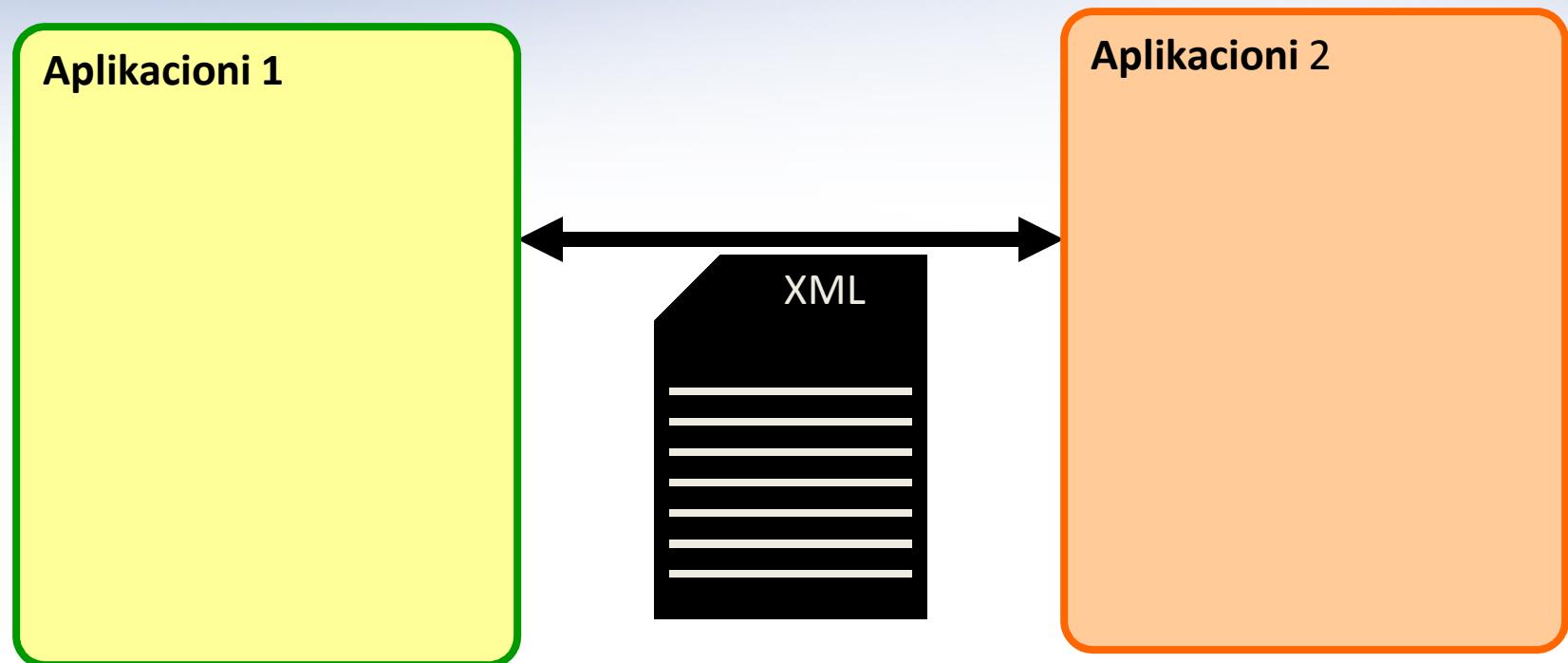


Fig 1. Shkëmbimi i informacioneve ndermjet dy aplikacioneve permes XML

Komunikimi ndërmjet aplikacioneve softuerike të ndërtuara në platforma të ndryshme me ndihmën e Shërbimeve Ueb

Prof. Ass. Dr. Isak Shabani

Në vijim është dhënë mënyra vizuale e përshkrimit të procesit të dyanshëm të shkëmbimit të informacioneve ndërmjet dy platformave nëpërmjet arkitekturës JNBridgePro.

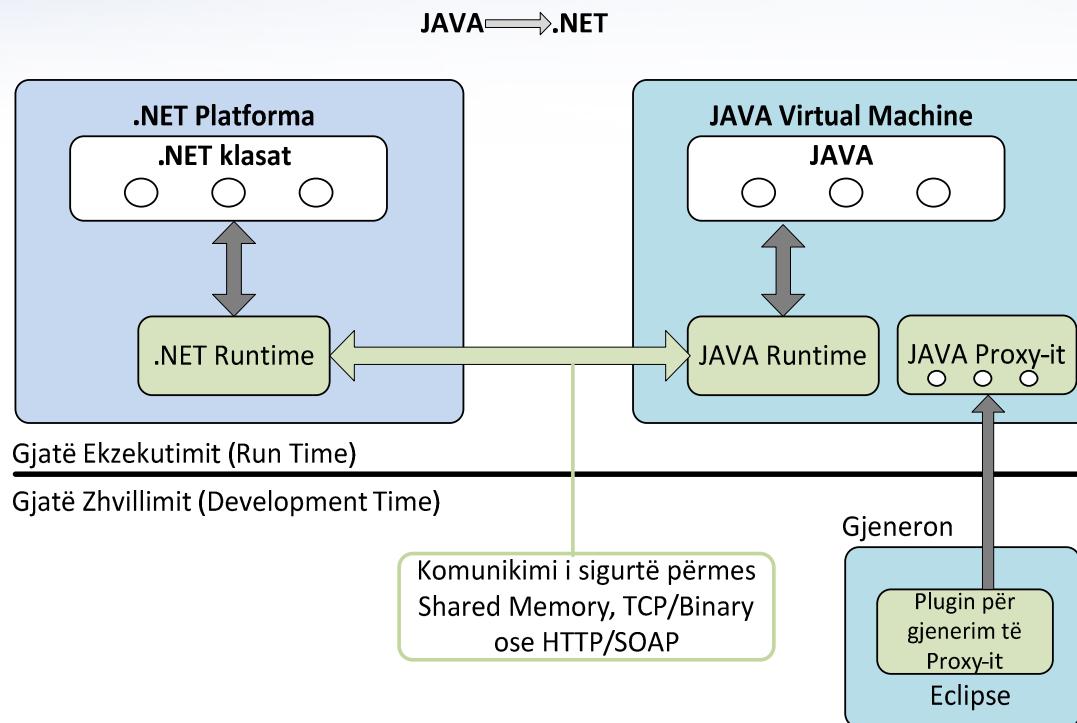


Fig 2. Shkëmbimi i informacioneve Java në .NET

Komunikimi ndërmjet aplikacioneve softuerike të ndërtuara në platforma të ndryshme me ndihmën e Shërbimeve Ueb

Prof. Ass. Dr. Isak Shabani

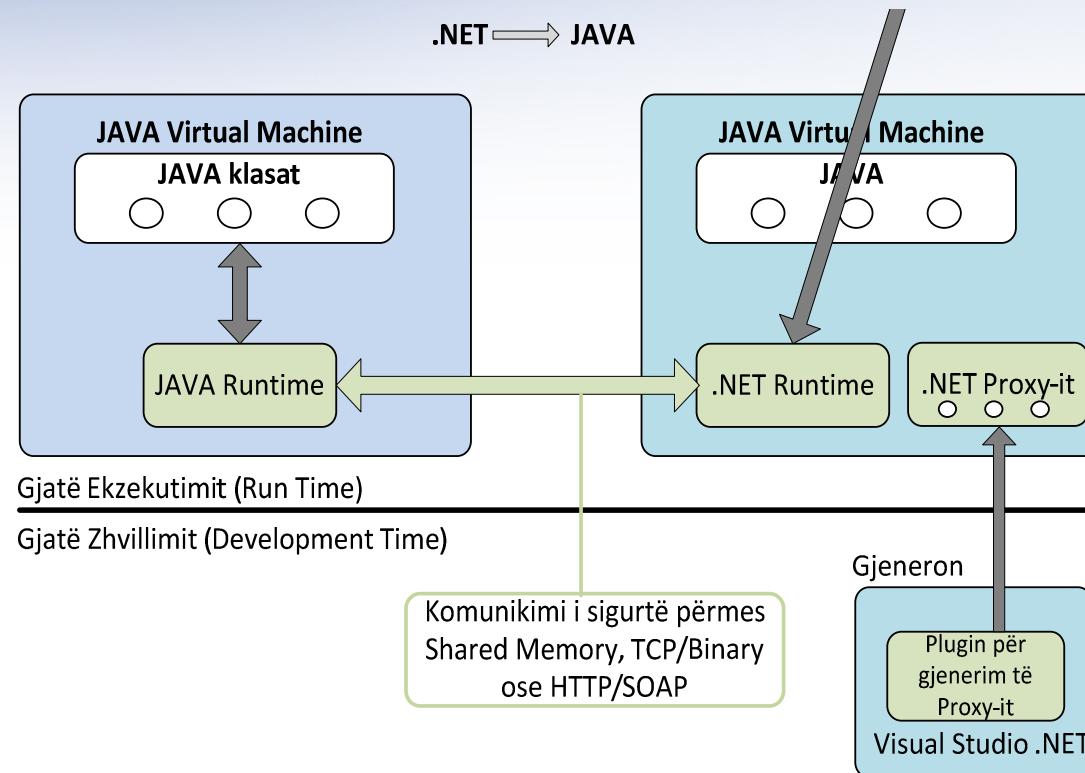


Fig 3 Shkëmbimi i informacioneve .NET në JAVA

Shembuj te krijimit te Ueb Sherbimeve

Prof. Ass. Dr. Isak Shabani

- Java
 - Client
 - Server
- VBScript
 - Client –high level API
 - Client – low level API
- .NET
 - C# Client
 - C# Server

Rekomandimet e projketimit te Ueb sherbimeve

Prof. Ass. Dr. Isak Shabani

- Krijimi i klases lokale
- Krijimi i metodes me emer te njejte

Shembuj (Java Client)

Prof. Ass. Dr. Isak Shabani

```
URL endpointURL = new URL(endpoint);  
  
Call call = new Call();  
  
call.setSOAPTransport(m_httpconn);  
  
call.setTargetObjectURI("MessageService");  
  
call.setMethodName("setMessage");  
  
call.setEncodingStyleURI(Constants.NS_URI_SOAP_ENC);
```

Shembuj (Java Client)

Prof. Ass. Dr. Isak Shabani

```
Vector params = new Vector() ;  
params.addElement (  
    new Parameter("name",  
        java.lang.String.class, name,  
        null)) ;  
params.addElement (  
    new Parameter("colour",  
        java.lang.String.class, colour,  
        null)) ;  
call.setParams(params) ;  
Response response =  
    call.invoke(endpointURL, "") ;
```

Shembuj (Java Client)

Prof. Ass. Dr. Isak Shabani

- Demostrimi

- A Web service Server is simple:
 - New class with method
- Then:
 - Register class with soap router
 - Or
 - Place the source code in a jws file

- High Level API (After adding a Web Service Reference)

```
Dim serv As clsws_MessageService
```

```
Set serv = New clsws_MessageService
```

```
serv.wsm_setMessage txtName.Text, txtColor.Text
```

Shembuj (VB Client)

Prof. Ass. Dr. Isak Shabani

```
Serializer.Init Connector.InputStream  
    Serializer.startEnvelope , ENC  
        Serializer.SoapNamespace "xsi", XSI  
        Serializer.SoapNamespace "SOAP-ENC", ENC  
        Serializer.SoapNamespace "xsd", XSD  
    Serializer.startBody
```

```
        Serializer.startElement Method, URI, , "method"  
            Serializer.startElement "parameter"  
                Serializer.SoapAttribute "type", ,  
                "xsd:string", "xsi"  
                Serializer.writeString username  
            Serializer.endElement
```

```
        Serializer.endBody  
    Serializer.endEnvelope  
Connector.EndMessage
```

Shembuj (VB Client)

Prof. Ass. Dr. Isak Shabani

Name

Bill

Colour

blue

Call Web Service

Shembuj (C# Client)

Prof. Ass. Dr. Isak Shabani

- Add a Web References to a project

```
localhost.MessageService serv = new  
localhost.MessageService();  
  
serv.setMessage(x, y);
```

Shembuj (C# Client)

Prof. Ass. Dr. Isak Shabani

- Demostrimi

Shembuj (C# Server)

Prof. Ass. Dr. Isak Shabani

```
public class Demo :  
System.Web.Services.WebService {
```

```
    public Demo() {  
        InitializeComponent();  
    }
```

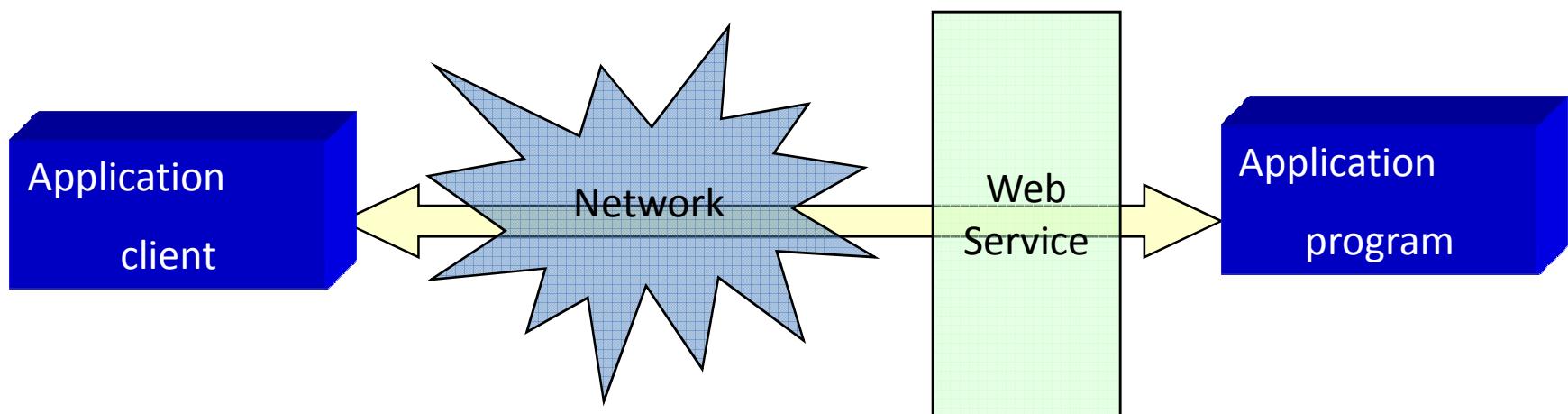
[Web Method]

```
    public string HelloWorld() {  
        return "Hello World";  
    }
```

Arkitektura standarte e Ueb Sherbimeve

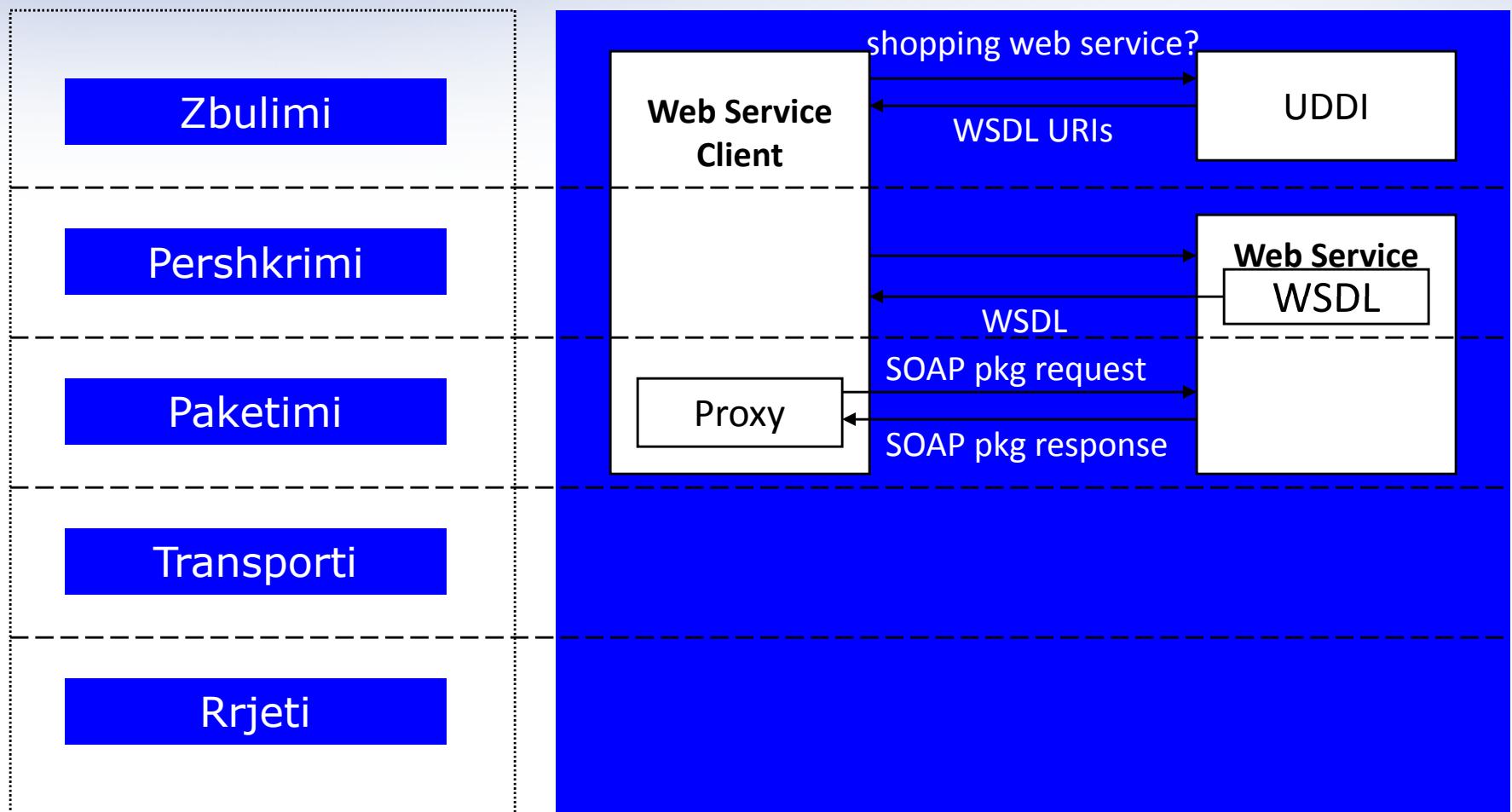
Prof. Ass. Dr. Isak Shabani

- A web service is a network accessible interface to application programs, built using standard Internet technologies.
- Clients of web services do **NOT** need to know how it is implemented.

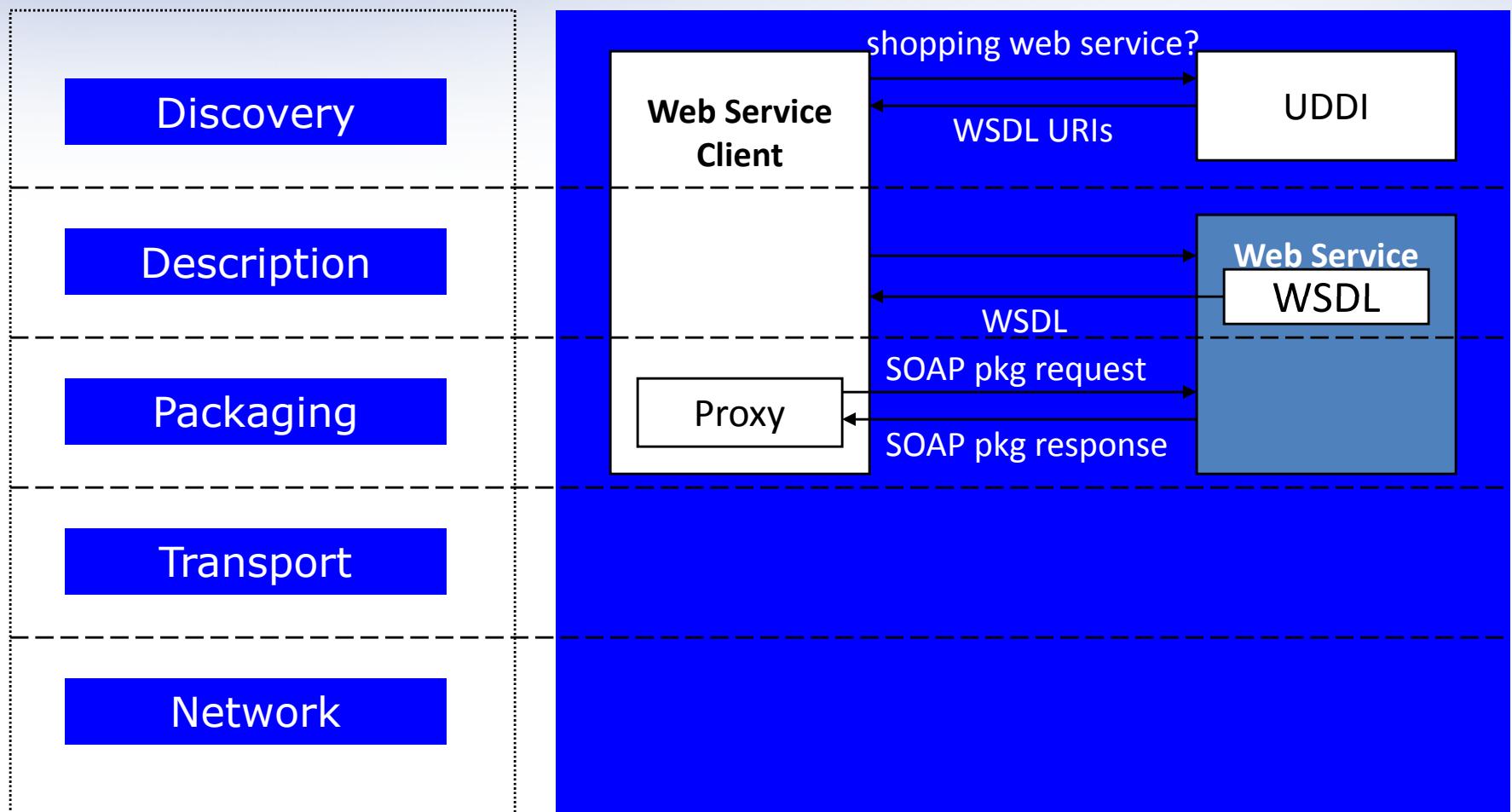


Web Service Technology Stack

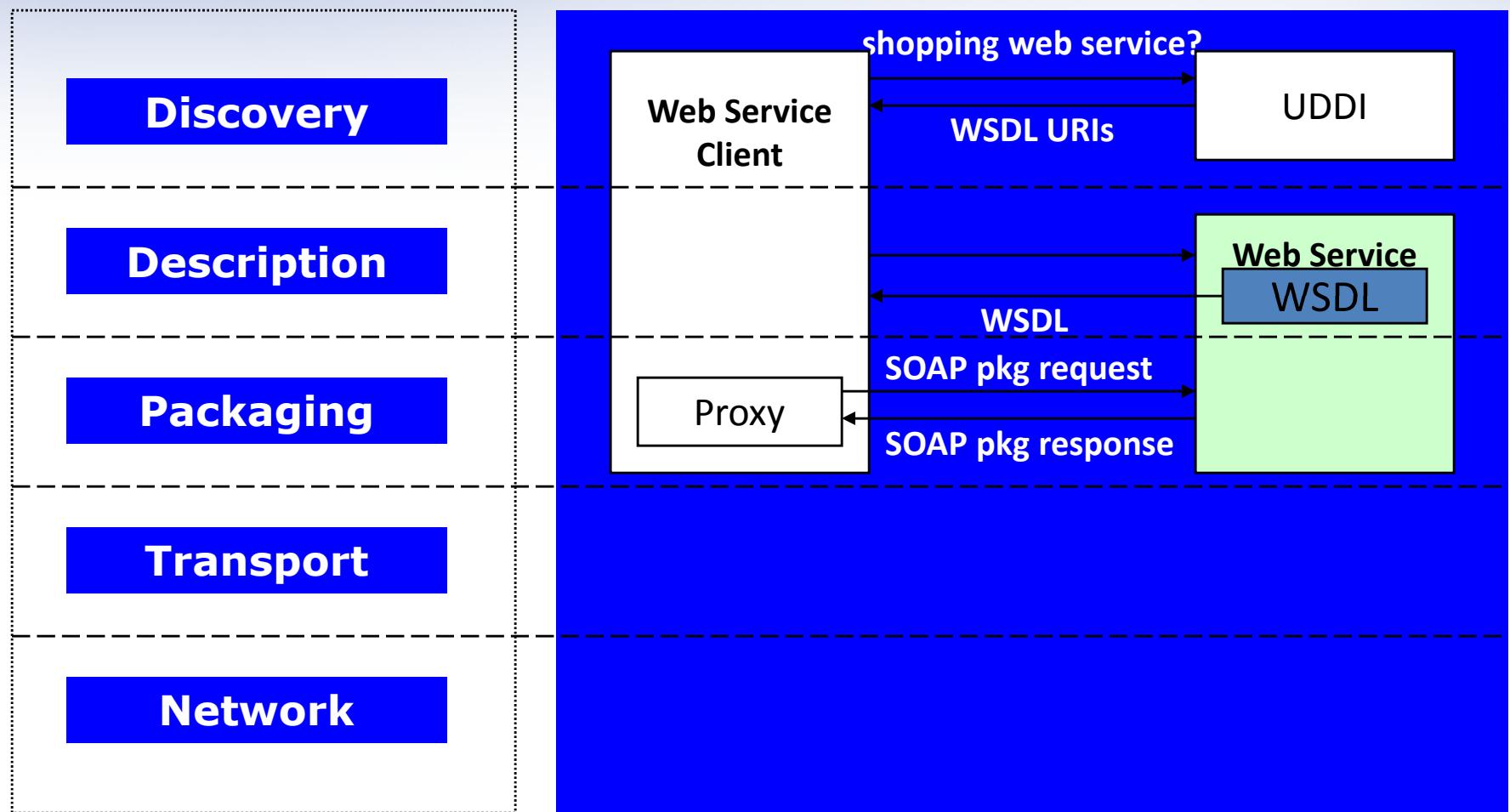
Prof. Ass. Dr. Isak Shabani



Step1. Write Web Service Method



Step2. Describe Web Service using WSDL



TEKNOLOGJITË DHE KOMPONETET E UEB SHËRBIMEVE

Prof. Ass. Dr. Isak Shabani

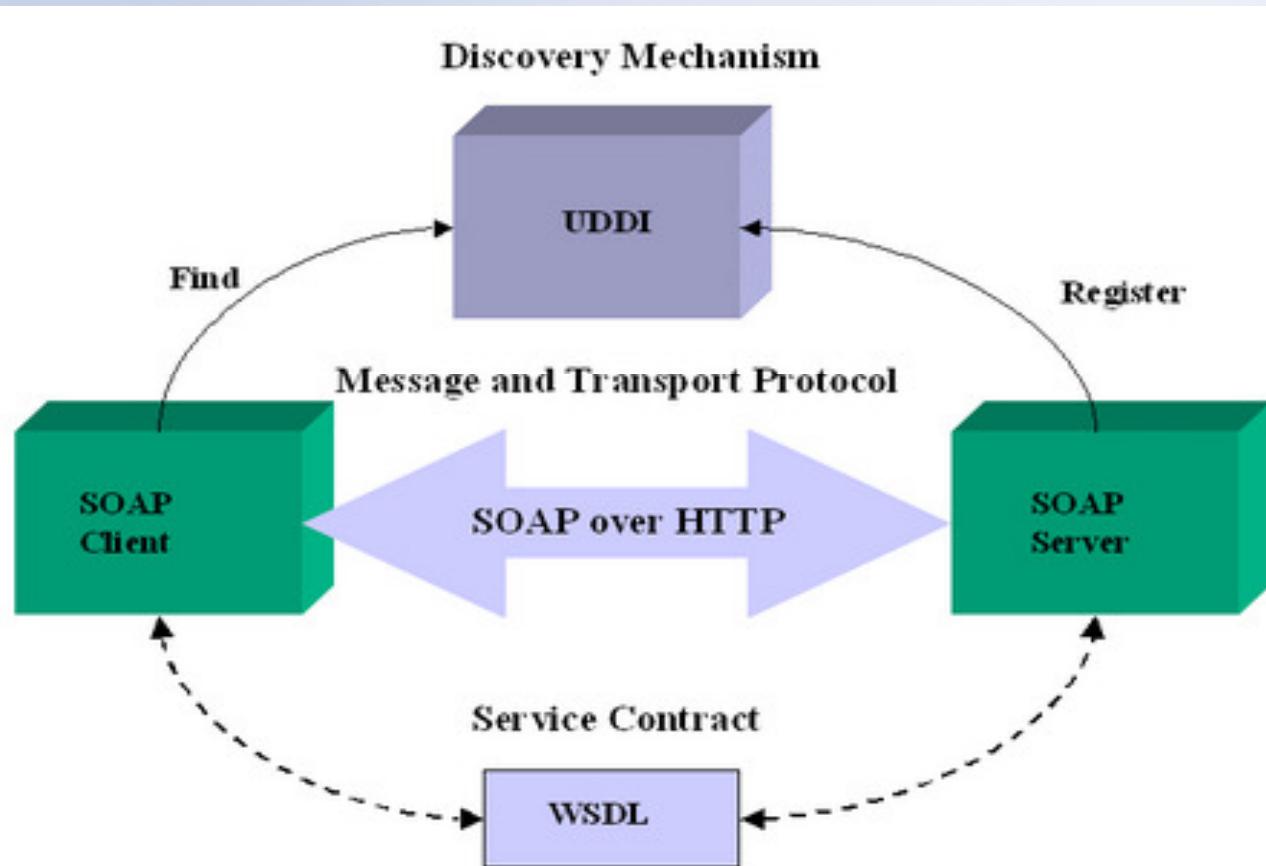
Teknologjite e Ueb Shërbimeve

- Ueb shërbimet janë komponentë softuerike që komunikojnë duke përdorur teknologjite Ueb të bazuara në standarde, duke përfshirë HTTP dhe mesazhet e bazuara në XML.
- Ueb shërbimet janë dizajnuar që të jenë të qasshme nga aplikacionet tjera dhe ndryshojnë në kompleksitet, duke filluar nga operacione të thjeshta e deri te ato më komplekse.
- Ueb Shërbimet bazuhe në standardet e hapura si HTTP dhe protokollet e bazuara në XML duke përfshirë SOAP dhe WSDL.
- Ueb shërbimet janë të pavarura nga hardueri, gjuha programuese dhe sistemi operativ.
- Kjo nënkupton që aplikacionet e shkruara në gjuhë të ndryshme programuese dhe që ekzekutohen në platforma të ndryshme mund të shkëmbejnë të dhëna përmes intranet-it ose Internet-it duke përdorur Ueb shërbimet.

Teknologjite e Ueb Shërbimeve

- Ueb shërbimet mundësohen nga XML dhe tre teknologji tjera thelbësore: WSDL, SOAP dhe UDDI.
- Para se të ndërtohet një Ueb shërbim, zhvilluesit krijojnë një përkufizimin në formë të një dokumenti WSDL që përshkruan vendndodhjen e lokacionit në Ueb dhe funksionalitetin që ofron shërbimi.
- Informacionet rreth shërbimit mund të futen në një regjistër UDDI i cili lejon konsumatorët e Ueb shërbimit të kërkojnë dhe gjejnë shërbimet që atyre i`u nevojiten.
- Ky hap nuk është i detyrueshëm por është i dobishëm kur kompania dëshiron që Ueb shërbimet e saj të zbulohen nga ana e konsumatorëve të shërbimeve të brendshme ose të jashtme.
- Bazuar në informacionin në regjistrin UDDI, zhvilluesi i Ueb shërbimit klient shfrytëzon instrukzionet në WSDL për të ndërtuar mesazhet SOAP për shkëmbimin e të dhënavë me shërbimin, përmes HTTP-së.

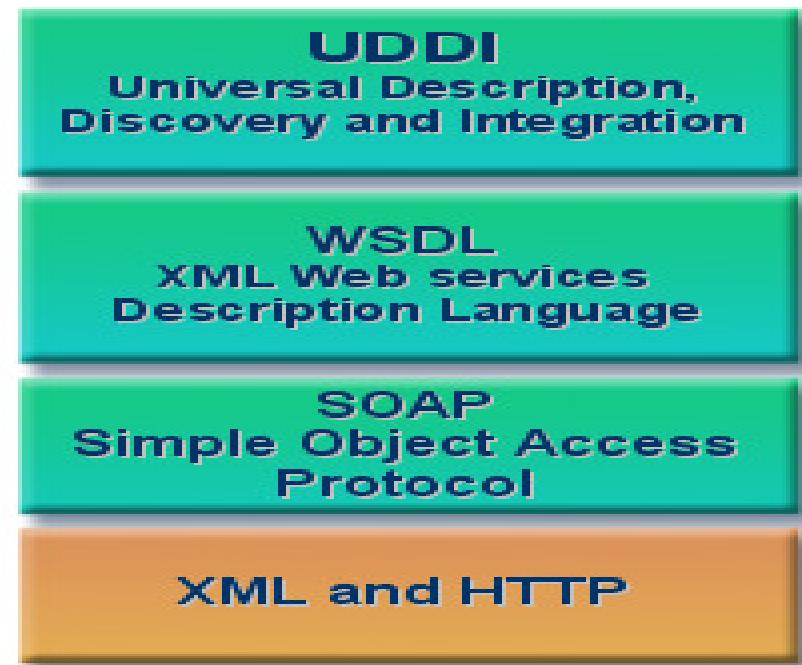
Teknologjite e Ueb Shërbimeve



Arkitektura e orientuar në shërbime duke shfrytëzuar Ueb Shërbimet

Komponetet e Ueb Shërbimet

- **UDDI** – Siguron mekanizmin për të regjistruar dhe kategorizuar Shërbimet Ueb që ofrohen si dhe për të gjetur Ueb shërbimet të cilat do të përdoren.
- **WSDL** – Bazohet në XML për përshkrimin e Ueb Shërbimeve. Përshkruan çfarë funksionaliteti ofron Shërbimi Ueb, mënyrën e komunikimit dhe ku mund të qaset.
- **SOAP** – Është protokoll i thjeshtë i bazuar në XML që i mundëson aplikacioneve shkëmbim të informatave përmes HTTP.
- **XML** - Paraqet formatin e shënimive që barten prej një aplikacioni në tjetrin.



SOAP

- SOAP është protokoll për komunikim ndërmjet aplikacioneve.
- SOAP bazohet në XML dhe ju mundëson aplikacioneve që të komunikojnë edhe përmes internetit.
- SOAP në teknologjinë e Shërbimeve Ueb qëndron si protokoll i standardizuar për paketimin e porosive të cilat komunikojnë dhe përdoren nga aplikacionet.
- SOAP përmban **zarfin** që përbëhet nga **koka** dhe **trupi**.
- Zarfi SOAP** përshkruan përbajtjen e mesazhit dhe mënyrën e përpunimit dhe përmban informacionet për destinacionin final të mesazhit.
- Koka** përmban të dhëna shtesë rreth mesazhit por vetë nuk është pjesë e mesazhit. Për shembull koka mund të përbajë:



Protokolli SOAP

- Informacionin për shfrytëzuesin që është duke e bërë kërkesën, që përfshinë ID-në e shfrytëzuesit, fjalëkalimin, certifikimin X.509, ose të dhënat e tjera identifikuuese.
- Informacionin rreth transaksionit, gjendjes dhe rrugëtimit që tregon se kush e ka përpunuuar mesazhin SOAP
- Trupi** përmban ngarkesën fitimprurëse të mesazhit dhe një seksion që është opcional që është gabimi SOAP që përmban informacionin rreth gabimeve eventuale ose të gjendjeve.

SOAP

- SOAP është një protokoll i specifikuar për thirrjen e metodave në server, shërbime, komponente dhe objekteve.
-
- SOAP kodifikon praktikën ekzistuese të përdorur XML dhe HTTP si një mekanizëm i thirrjeve të metodave.
- Specifikimi SOAP shfrytëzon fjalorin XML që është përdorur për të përfaqësuar vlerat e parametrave të metodave, kthimin e vlerave dhe përjashtimet.
- Sikurse XML-RPC, SOAP është zhvilluar si platformë e pavarur nga gjuhët programuese, metodat e kalimit të informacionit midis shërbimeve, objekteve dhe serverat.
- SOAP ofron një mekanizëm të thjeshtë të standardizuar për të lëvizur informacione të strukturuara.

Si ka ka ardhur SOAP

- Microsoft, UserLand, DevelopMentor, IBM, Ariba, Commerce One, Compaq, HP, IONA, Lotus, SAP
~ Submitted to W3C May 2000.
- SOAP bazohet në XML.
- SOAP, protokoll për shkëmbim të informacionit.
- Përdorimi më i madh është mbi HTTP në Internet.

W3C

- 9 Janar, 2001
- W3C shtyp draftin e parë publik për versionin SOAP 1.2.
- Drafti iprodhuar dhe punuar nga grupei punues i XML Protokollit.
- Specifikimin mund ta gjeni në linkun:
<http://www.w3.org/TR/soap12/>

SOAP Motivimi

- Shumë aplikacione të shpërndarë komunikojnë duke përdorur RPC në mes të objekteve të shpërndara si DCOM dhe CORBA.
- HTTP nuk është i dizajnuar për këto objekte, kështu që thirrjet RPC nuk janë të përshtatura dhe të lehta të përdoren në Internet.
- Problemet e sigurisë ekzistojnë për metodat RPC, kështu që shumica e serverëve firewall dhe proxy janë të vendosur për të bllokuar këtë trafik.
- HTTP është përkrahur nga të gjithë Shfletuesit të Internetit dhe serverat, kështu SOAP paraqet një protokoll të mirë për të zëvnduar RPC-në.

SOAP

SOAP është:

- Protokoll komunikimi.
- Për komunikim në mes të aplikacioneve: kërkesë/përgjigje, multicast, etj.
- Projektuar për të komunikuar nëpërmjet HTTP.
- Jo të lidhura me ndonjë komponent teknologjike.
- Jo i lidhur me ndonjë gjuhë programimi.
- Bazuar në XML.
- E thjeshtë dhe e zgjerueshme.

SOAP

- SOAP është protokoll komunikimi për XML Ueb shërbimet.
- SOAP është dizajnuar për të përkthyer metodat, parametrat si informata XML tek serverët e largët.
- Edhe pse SOAP nuk është një objekt model, në fakt është një XML që është i definuar nga skema XML dhe ka elementet e tija ndërtuese.
- Këto elemente mundë të merren parasysh në terma të objekteve dhe këto tri objekte janë:
 - Zarfi SOAP (*eng. SOAP Envelope*)
 - SOAP kryetitulli(*eng. Header*)
 - SOAP trupi - pjesa kryesore (*Body*)

SOAP

- Secili mesazh SOAP fillon me një përbajtës të quajtur zarfi (mbështjellësi).
- Brenda zarfit janë Headeri (kryetitulli) dhe trupi (pjesa kryesore).
- Headeri është opsional, por trupi duhet të ekzistojë.
- Brenda trupit janë ngarkesa e dobishme dhe seksioni i quajtur defekti SOAP.
- Ngarkesa e dobishme është e dhëna që mesazhi ishte krijuar të transmetojë.
- Merr formën e një fragmenti XML kuptimi i të cilit komplet është i varur në Ueb shërbimin e thirrur.
- Seksioni SOAP defekti përmban informatë të gabimit dhe normalisht mund të gjendet në mesazhet kthyese (ku jepet përgjigja) .

SOAP

- SOAP është një protokol për shkëmbim të mesazheve në një sistem kompjuterik të shpërndarë dhe e luan një rol krucial në Ueb Shërbime.
- Duke u mbështetur në XML mundësohet dërgimi dhe pranimi i mesazheve ndërmjet dy aplikacioneve.
- Thënë shkurt SOAP e bën formatizimin dhe përshkrimin mesazhit dhe kështu mundëson komunikim mes dy aplikacioneve pavarësisht arkitekturës të ndryshme të tyre dhe gjuhëve programuese me te cilat janë krijuar ato.
- Ne anën tjetër SOAP klasifikohet edhe si XML dokument (SOAP envelope) i cili enkapsulohet me njëren nga protokolet e shtresës me të ulët zakonisht me anë te HTTP në portin 80 kjo meqenëse ky port nuk është i ndaluar nga shumica e firewall-ëve,,
- Në përgjithësi mund të transferohet me shumicën e protokolleve p.sh. me SMTP, FTP, TCP etj.

Struktura gjenerale e SOAP mesazheve

- Zarfi
Përcakton (definon) përmbajtjen e mesazhit
- Koka (opcionale)
Përmban informacionet në kokë
- Trupi
përmban thirrjet dhe përgjigjet e informacionit

Zarfi

- Zarfi SOAP është dokumenti i cili transferohet nëpër rrjet prej dërguesit te marrësi.
- Zarfi është një kontejner i të dhënave që barten dhe përmban të dy pjesët: *Header* dhe *Body*, i pari është opcional kurse ky i fundit është i domosdoshëm dhe duhet të paraqitet vetëm një herë përgjatë dokumentit.
- Duhet të shoqërohet me SOAP zarfi namespace:
<http://www.w3.org/2001/06/soap-envelope>
- SOAP serializimi namespace:
<http://www.w3.org/2001/06/soap-encoding>
Kodimi i stilit të attributeve që mund të përbajnë një URI që përshkruan se si të dhënat duhet të serizlizohen.
- Mesazhi SOAP nuk përmban DTD.

Koka (Header)

- Opcionale.
- Në elementin *Header* ruhen të dhëna opcionale siç janë ato për:
 - siguri,
 - transaksione,
 - instruksione përrugëtim, informacione për debagim dhe
 - informacione tjera të rëndësishme për procesimin e mesazhit.
 - Këto informata janë të futura nëpër bloqe të veçanta.
- Nëse është i pranishëm, duhet menjëherë të ndiqni elementin zarf SOAP XML ndjekur nga çdo hyrje header. Shpesh përmban meta-informacion në lidhje me thirrjen e metodës.
- Atributi aktor ← të cilët duhet të përpunonjë mesazhin
- Duhet të kuptojnë atributin ← si të procesohen (e prezgjedhur është "0" në qoftë se nuk është i pranishëm))

```
<env:Header>
  <t:Transaction
    xmlns:t="some-URI" env:mustUnderstand="1">
    5
  </t:Transaction>
</env:Header>
```

Trupi (Body)

- Elementi *Body* përbahen të gjitha të dhënat të cilat shkëmbohen në mes dy aplikacioneve.
- Pasi që në këtë element nuk specifikohet se çfarë lloji mesazhi mund të dërgohet kjo na jep një fleksibilitet të madh pasi që ai mund të jetë çfarëdo XML mesazhi.
- Në *Body* zakonisht futen parametrat për metodat e Ueb shërbimeve po ashtu dhe tipi i tyre.
- Përderisa gjatë rrugës nëpër shumë stade për elementin *Header* lejohet të ndryshohen informatat që ndodhen në të, për elementin *Body* nuk duhet të ndodh e njëta gjë, ai duhet të ndryshohet vetëm nga marrësi i fundit.
- SOAP nuk i ndalon ndonjë ndërmjetësuesi të ndryshoj përbajtjen e elementit *Body*, kështu që marrësi final nuk është i sigurt a ka ndryshuar mesazhi gjatë rrugës apo jo.
- Kjo e paraqet një rrezik për sigurinë e aplikacionit.
- Shkëmbimi i mesazheve .
- Më shpesh për thirrjet RPC dhe raportim të gabimeve (errorr).
- Përmban metodë e serializuar me argumente.
- Emri metodës në distancë është përdorur për të thirrur elementet XML dhe duhet menjëherë të I bashkangjiten trupit SOAP per hapjen e XML tagjeve.

Një shembull i thjesht i zarfit SOAP

Një shembull i thjeshtë SOAP XML dokument qrkon cmimin e Librit.

```
<env:Envelope>
  <env:Body>
    <m:GetCmimi>
      <Item>Libri 50 Euro</Item>
    </ m:GetCmimi >
  </env:Body>
</env:Envelope>
```

Shembull HTTP POST Request

- POST /StockQuote HTTP/1.1
Host: www.stocksserver.com
Content-Type: text/xml; charset="utf-8" ←SOAP HTTP Header info
Content-Length: nnnn
SOAPAction: "Some-URI" ←SOAP HTTP Header info
<env:Envelope
 xmlns:env="http://www.w3.org/2001/06/soap-envelope">
 <env:Body>
 <m:GetStockQuote xmlns:m="Some-URI"
 env:encodingStyle="http://www.w3.org/2001/06/soap-encoding">
 <symbol>SUNW</symbol>
 </m:GetStockQuote >
 </env:Body>
</env:Envelope>
- Sent from the client to the service. First few lines are HTTP headers and SOAP HTTP headers. Followed by the SOAPAction HTTP header and SOAP Envelope XML element. The SOAP Body contains application defined element(s), as defined by the service.

Shembull HTTP Response

HTTP/1.1 200 OK

Content-Type: text/xml; charset="utf-8" ←SOAP HTTP Header info
Content-Length: nnnn

```
<env:Envelope xmlns:env="http://www.w3.org/2001/06/soap-envelope">
  <env:Body>
    <m:GetStockQuoteResponse xmlns:m="Some-URI"
      env:encodingStyle="http://www.w3.org/2001/06/soap-encoding">
      <price>5.00</price>
    </m:GetStockQuoteResponse>
  </env:Body>
</env:Envelope>
```

W3C Draft: Shembull i SOAP Messages

<http://www.w3.org/TR/2001/WD-soap12-20010709/>

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

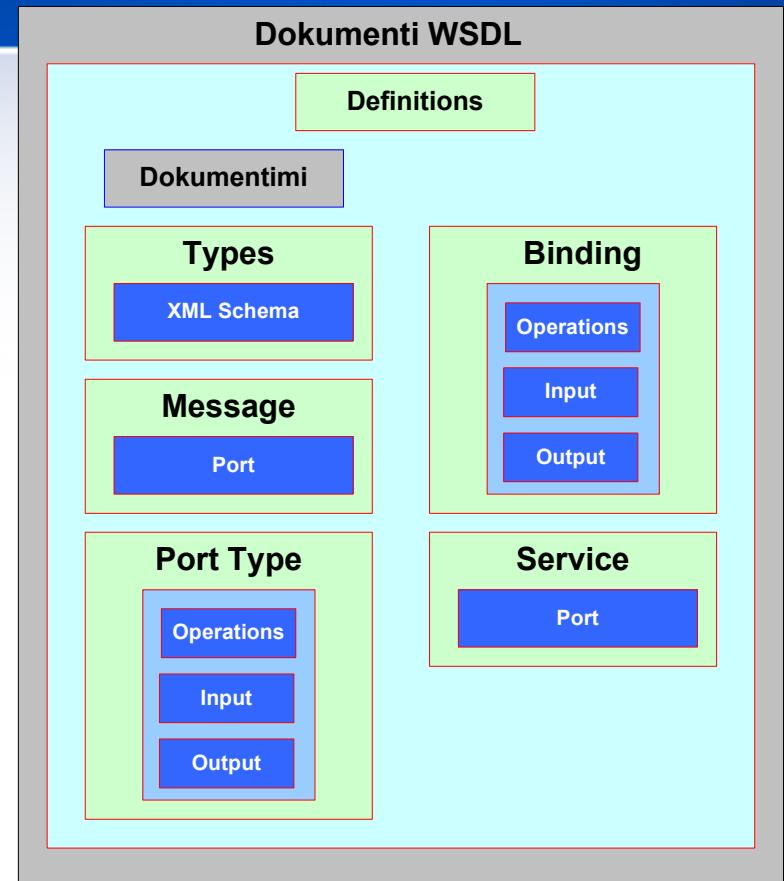
WSDL bazohet në gjuhë XML për përshkrimin e Shërbimeve Ueb.

Për të kriuar një Shërbim Ueb duhet të krijohet një dokument WSDL i cili e përshkruan shërbimin e tillë. Dokumenti do të vendoset në server apo publikohet në regjistrin UDDI.

Klienti i interesuar në këtë shërbim së pari merr një kopje apo referencë të këtij dokumenti duke e kërkuar regjistrin, mandej e kuption kontratën, krijon kërkesën SOAP duke u bazuar në kontratë dhe të njëjtën e dërgon në server.

Elementet e WSDL – së:

- **definitions**
 - Definon emrin e shërbimit
 - Deklaron namespa-sat që shfrytëzohen në dokument
- **types**
 - Përshkruan tipet e të dhënave që shfrytëzohen nga Klienti dhe Serveri
- **message**
 - Definon emrin mesazhit request/response
 - Definon gjithashtu elementet e mesazhit
- **portType**
 - Definon kombinimin e elementeve të mesazhit të një forme me komplet operacionet
- **binding**
 - Përfshin specifikat detale se si janë të lidhura mesazhet që transmetohen
- **service**
 - Definon adresën ku është i vendosur shërbimi



Dokumenti WSDL.

WSDL

- WSDL-ja është një XML dokument dhe përshkruan metodat dhe karakteristikat që ka një Ueb Shërbim.
- WSDL-ja është një format XML për të përshkruar se si një sistem softuerik mund të lidh dhe shfrytëzoj shërbimet e një sistemi tjetër softuerik nëpërmes internetit.
- Specifikat WSDL përkrahin ndërtimin e informatikës të bazuar në Ueb shërbime që kanë për objektiv programet kompjuterike se sa përdoruesit.
- WSDL përdor një sintaksë të bazuar në XML për të përshkruar specifikat e qasjes së Ueb Shërbimit, të tillë si lloji dhe numri i parametrave të kaluar tek një shërbim dhe lloji dhe struktura e rezultatit të kthyer.
- Dokumenti WSDL që përshkruan Ueb shërbimin vepron si kontratë mes Ueb shërbim klientit dhe serverit.
- Duke respektuar këtë kontratë ofruesi i shërbimit dhe konsumatori janë në gjendje të shkëmbejnë të dhëna në mënyrë standarde, pavarësisht platformave bazë dhe aplikacioneve në të cilat operojnë.

WSDL

- WSDL-ja është një XML dokument dhe përshkruan metodat dhe karakteristikat që ka një Ueb Shërbim.
- Një WSDL dokument definon një Ueb Shërbim duke përdorur këto elemente kryesore:

<portType>
<mesazhi>
<llojet>
<lidhja>

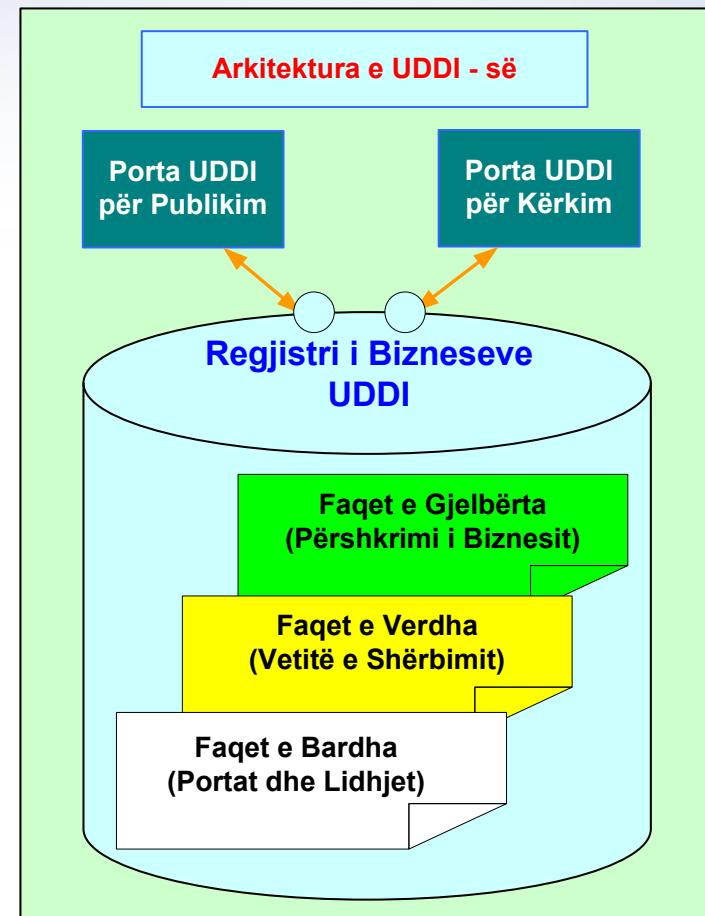
<**portType**>- përfaqëson një strukturë abstrakte të përkrahur nga një apo më shumë pika përfundimtare.

<**message**> një definicion abstrakt i shtypur i të dhënave të komunikuar.

<**types**> një përbajtës(kontejner) për definicionet e llojit te të dhënave që përdorin një lloj sistemi.

<**binding**> një protokoll konkret dhe format specifik i të dhënave për një lloj porti i posaçëm .

- UDDI ka filluar si bashkëpunim ndërmjet Microsoft-it, IBM-it dhe Aribas për të ndihmuar adaptimin dhe përdorimin e standardeve të Shërbimeve Ueb.
- UDDI është një regjistër i cili shërben për ruajtjen dhe gjetjen e informacioneve të Shërbimeve Ueb.
- Regjistri publik i UDDI-së punon në mënyrë të ngjashme sikur edhe DNS (*Internet Domain Name Service*).
- Të dhënat e UDDI-së ndahen në tri kategori kryesore:
 - **Faqet e bardha:** Emri i biznesit, adresa, të dhëna kontaktuese, Emri i Ueb faqes dhe numëri identifikues.
 - **Faqet e verdha:** Lloji i biznesit, lokalizimi dhe produktet, duke përfshirë lokalizimin e biznesit, llojin e industrisë, numrin identifikues të biznesit, etj.
 - **Faqet e gjelbërtë:** Të dhënat teknike për shërbimet e biznesit, si p.sh. si të bashkëveprojmë me to, definime të proceseve të biznesit, etj.



Arkitektura e UDDI-së.

UDDI

- UDDI është standard i sponzoruar nga OASIS (Organization for the Advancement of Structured Information Standards).
- UDDI është një specifikim për krijimin e një regjistri të bazuar në XML që liston informacionet rreth bizneseve dhe Ueb shërbimeve që ato ofrojnë.
- UDDI i`u ofron bizneseve një mënyrë uniforme për listimin e shërbimeve të tyre dhe zbulimin e shërbimeve të ofruara nga organizatat tjera.
- Edhe pse implementimet ndryshojnë, UDDI shpesh përshkruan Shërbimet duke përdorur WSDL dhe komunikon përmes SOAP mesazheve.
- Regjistrimi i Ueb shërbimit në regjistrin UDDI është hap opcional dhe regjistrat UDDI mund të jenë publik ose privat.

UDDI

- UDDI mundëson përdoruesit përfundimtarë për të kërkuar dhe për të gjetur shërbimet e internetit në lidhje me internet
- Paraqet një platformë të pavarur për përshkrimin e shërbimeve, gjetjen e bizneseve dhe integrimin tyre përmes internetit.
- Bizneset mund të gjejnë dhe të shpërndajnë informacione të biznesit.
- Registri UDDI është një shërbim publik që u mundëson bizneseve që me një mënyrë të standardizuar të përshkruajnë shërbimet e tyre, të hulumtojnë shërbimet e kompanive tjera dhe të kuptojnë metodat e nevojshme për të bërë biznes përmes ueb-it.
- UDDI është ekuivalent me DNS te rrjetat kompjuterike por në vend se të bëj përkthimin e adresave të serverëve ajo bën hulumtimin e shërbimeve të kompanive.
- Registri i UDDI është i ndarë në tri pjesë kryesore:
 - faqet e bardha,
 - të verdha dhe
 - të gjelbra.

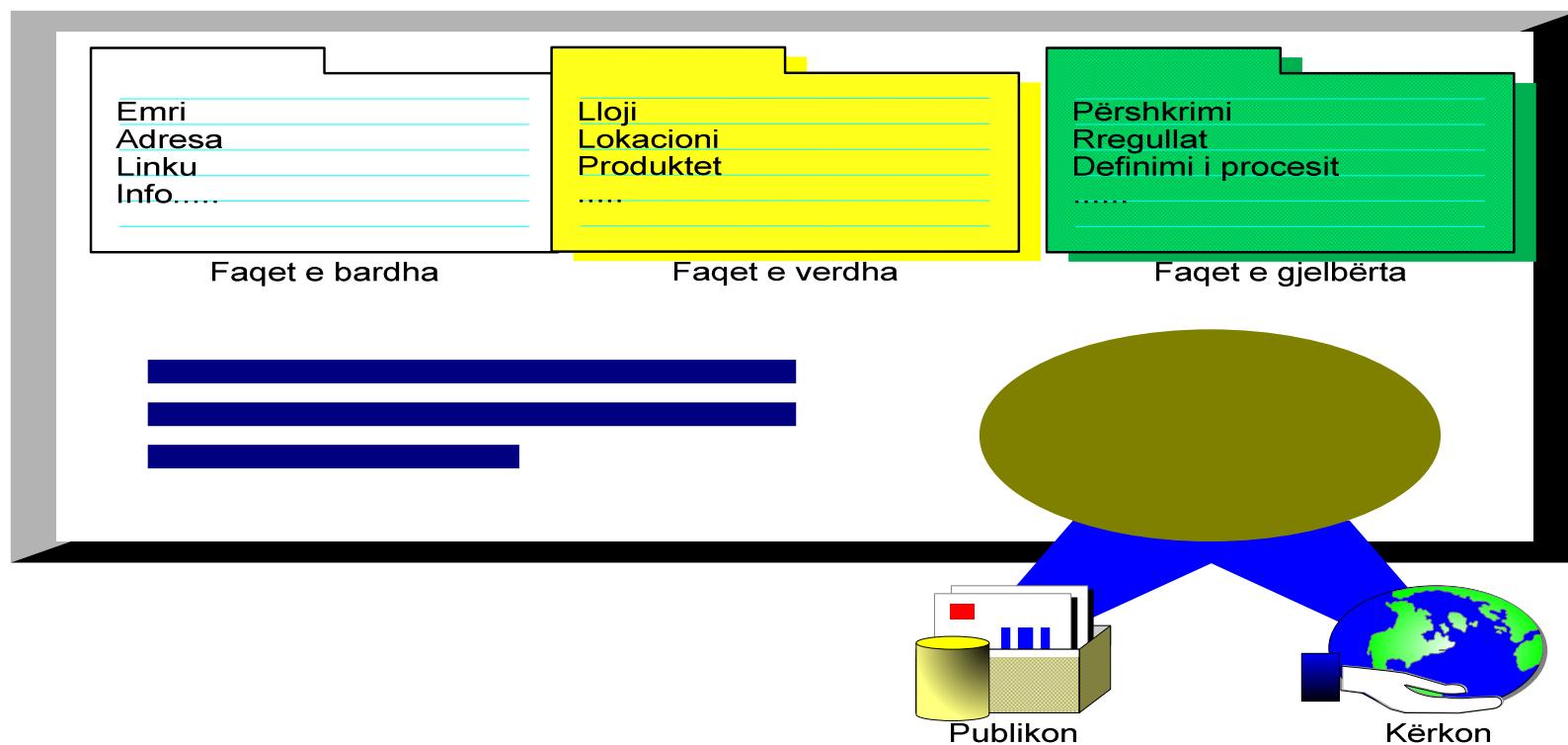
Faqet e bardha përmbajnë informacione themelore për kompanitë si emri, numri i tel., URL etj.

Faqet e verdha përmbajnë informata më të detajizuara dhe klasifikon shërbimet me të cilat merret ajo kompani.

Faqet e gjelbra përmbajnë të dhëna teknike për Ueb Shërbimet dhe mënyrës se si të qasemi në to.

Arkitektura e UDDI regjistrat

- UDDI është një urë lidhëse ndërmjet ofruesit të shërbimeve dhe kërkuesit të tyre.
- Ueb shërbimet kanë kuptim vetëm atëherë kur përdoruesit mund të gjejnë informata që i përgjigjen kërkuesave të tyre.
- Një ofrues i shërbimeve publikon shërbimin e vet të përshkruar duke përdorur WSDL dokumentin, pastaj kërkuesi i këtyre shërbimeve merr përsipër të gjej ato shërbime duke përdorur UDDI dhe në fund, ai kërkues merr shërbimin duke përdorur SOAP.



Rolucionet ndërmjet komponentëve: SOAP, WSDL dhe UDDI

