



G6: Web-Zerbitzuak.

Deskribapena eta protokoloak

Rosa Arruabarrena, Xabier Arregi, Jose Ángel Vadillo
LSI, UPV/EHU

1



Helburuak

- Bereziki SOAn oinarritutako Web-zerbitzuen funtsezko kontzeptuen sarrera
- Web-zerbitzuen argitalpen eta kontsumoarekin erlazionaturik dauden printzipio eta teknologien sarrera egin:
 - WSDL eta SOAP

2



Web-zerbitzuak. Zer dira?

- Web-zerbitzuak software-piezak dira
- Protokolo estandarren bidez, beste aplikazioek erabil ditzakete
 - Aplikazioen arteko datu-trukea gertatzen da
 - Aplikazio horiek programazio-lengoaia desberdinetan idatzita egon daitezke, sistema eragile desberdinetan, kokapen fisiko desberdinetan...
- Garapen eredu bat ahalbidetzen dute, non weba erabiliz, urruneko prozeduren deien integrazioa/erabilpena egiten duten.
 - Erabiltzaile interfaze gabeko web-guneak bezala imajina genitzake

3



Zer dira? Zertarako? (II)

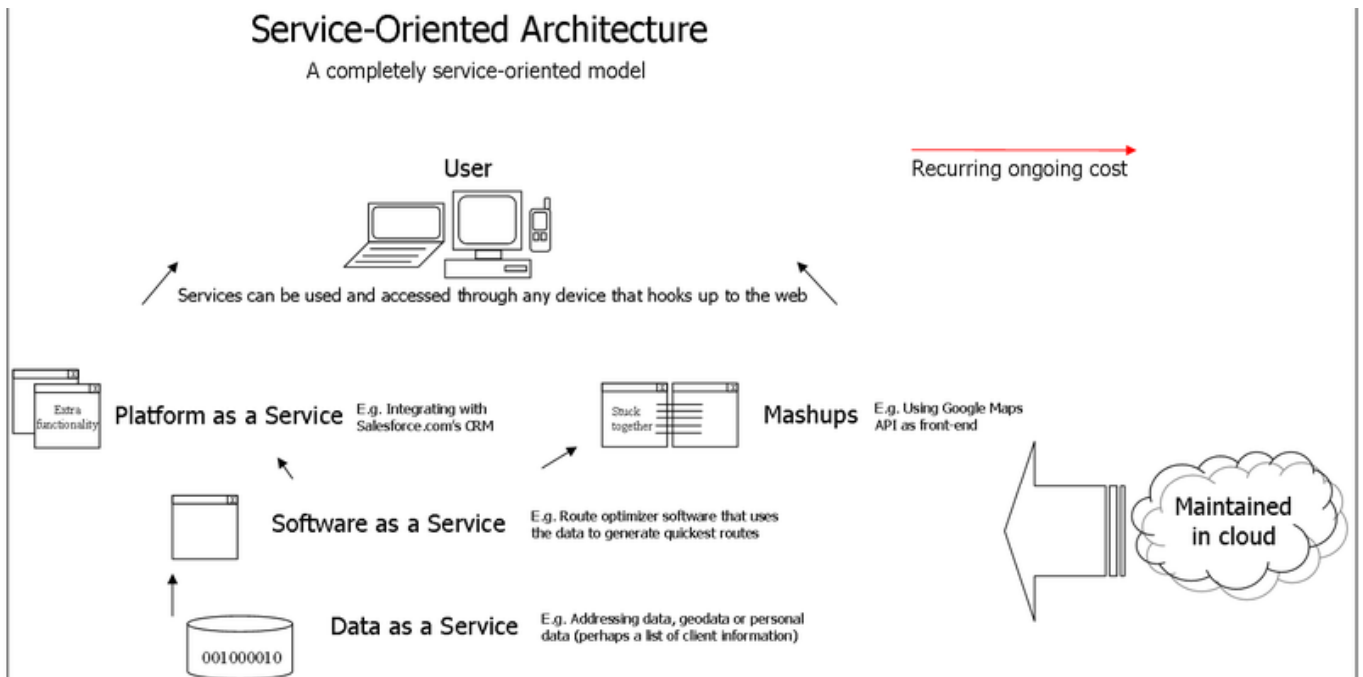
- Web-zerbitzuak auto-definitu eta auto-deskribatu egiten dira
- XML da komunikaziorako eta deskribapenerako erabiltzen den lengoaia nagusia.
 - SOAP eta XML ulertzen duen edozein bezerok atzi ditzake
- Estandarren erabilpena sustatzen du
- Mundu mailako sistema banatuak sor litezke, ezarrita dauden plataformetatik independente

4

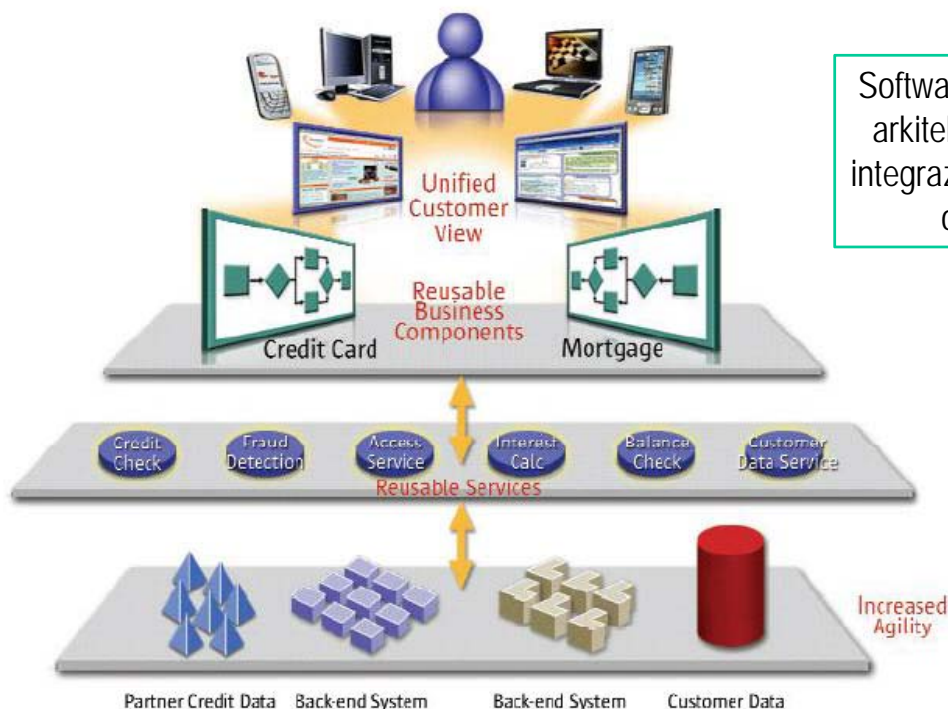


Web-zerbitzuak. Grafikoki

- Web-zerbitzuen azpiegitura: XML + HTTP
- Zerbitzuei Orientatutako Arkitektura



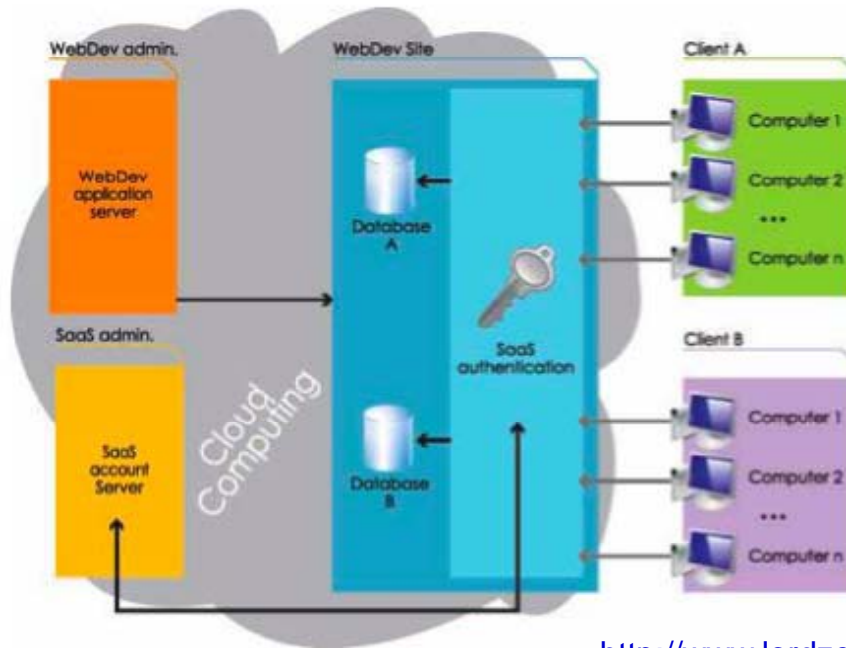
SOA arkitektura. Zerbitzuetara Orientaturiko Arkitektura



Softwarearen garapenerako
arkitektura web-zerbitzuen
integrazioan eta erabilpenean
oinarritzen dena

SaaS

(Softwarea zebitzu bat bezala ulertua)



Adib.:

- Google Docs (ofimatika)
- Prezi.com (aurkezpenak)
- Amazon Elastic Compute Cloud (Amazon EC2)

Lizentziarik ez,
erabilpenagatik ordaintzen da

<http://www.lordzealon.com>

7

Abantailak



- Elkarrekintza errazten dute (lengoiatik independente bai eta plataformatik ere)
- Estandarren hedapena sustatzen dute
- Aplikazioen integrazioa sistema banatuetan errazten dute
- Berrerabilpena errazten dute kapsulaketa (interfazea) bidez
- Negozio aukerak zabaltzen dituzte

8



Web-zerbitzuen bizitza-zikloa

- **Sorrera eta garapena**

- **Balio** bat **gehitzen** duten **negozioen arauak eta prozesuen identifikazioan** oinarrituak
- **Prozesu horien atzipena gauatzeko interfazeen** definizioa
- Interfaz horien deskribapena modu **estandarrean webeko softwarearen garapenerako industriarentzat**

- **Erakusketa publikoa**

- Interesgarri gerta dakioken **komunitateak** ikusgai eta ulergai izan ditzaten interfaze horiek argitaratzea komenigarria da

- **Erabilpena**

- **Zerbitzuen eskaerak onartu** behar dira (eta dagozkien erantzunak ekoitzi) web-eko **protokolo estandarrak erabiliz**
- Onartutako eskaera estandarren arteko eta web-zerbitzuen inplementazioen arteko erlazioa ezarri behar da

9



Web-zerbitzuetan oinarritutako negozio-ereduen adibideak

- Web-zerbitzu komertzialen merkatua

- <http://www.strikeiron.com/>
- <http://www.fraudlabs.com/>
- ...

Diseinu baten adibidea, non Web-zerbitzuen erabilpena aukera egin litekeen



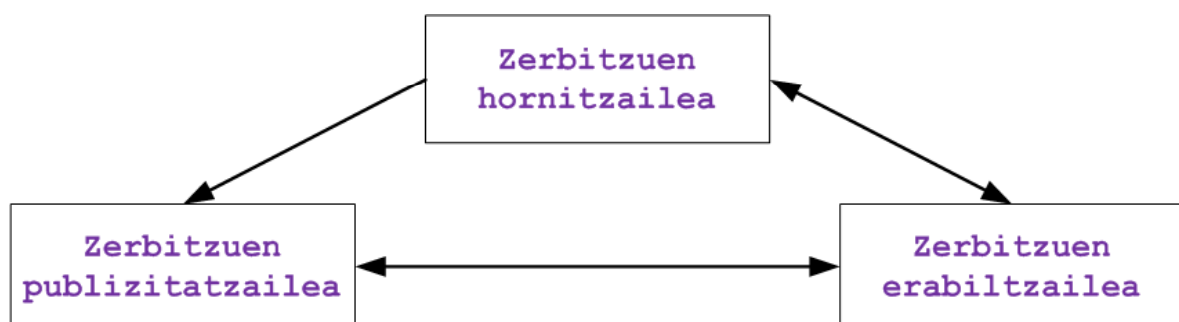
- Ingurunea: e-commerce zerbitzu bat garatu eta ezarri nahi da, eta bertako salmenta kreditu txartelak bidez gauzatu
 - Arazo eratorria: kreditu txartelen identifikatzaileetan gerta liteke iruzurrezko erabilpena egitea norbaitek
 - Errekerimendua: erosketa aginduak iragazi, eta kreditu-txartelaren identifikazioan iruzurra egiten ari dela susmoa balego
 - IP helbide, kreditu txartel edota kutxa entitateen inguruan informazioa eman dezaketen web-zerbitzu desberdinetara jotzen da
 - <http://www.fraudlabs.com/contact.aspx>

11

Nola erabil genezake web-zerbitzu bat?



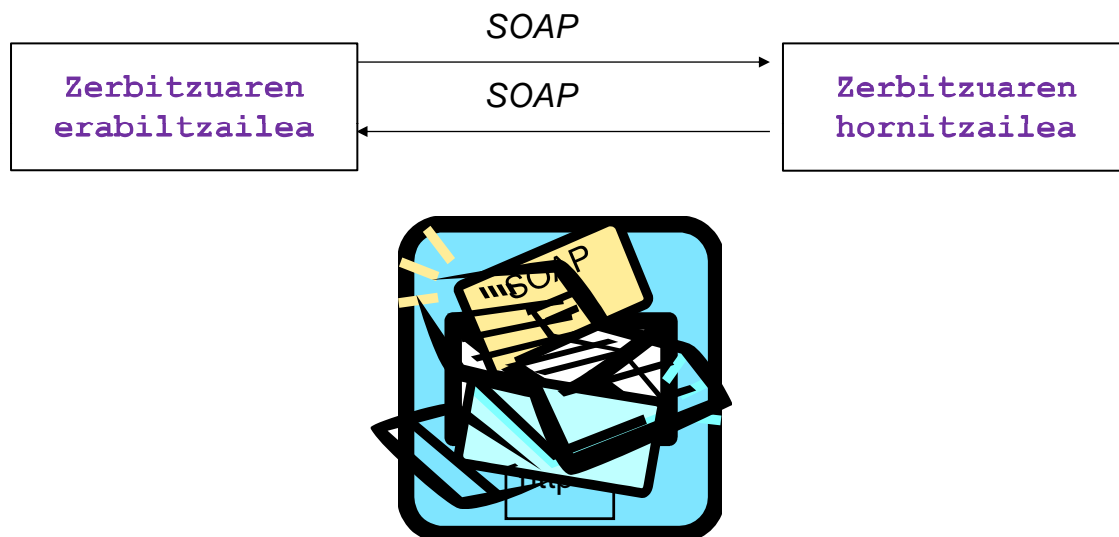
- Zeintzuk zerbitzu daude erabilgarri?
- Web-zerbitzu zehatz batekin nola ezartzen da komunikazioa?
- Goazen aurrera ... exekuta zaitez!



12



Web-zerbitzu baten “kontsumoa”



13



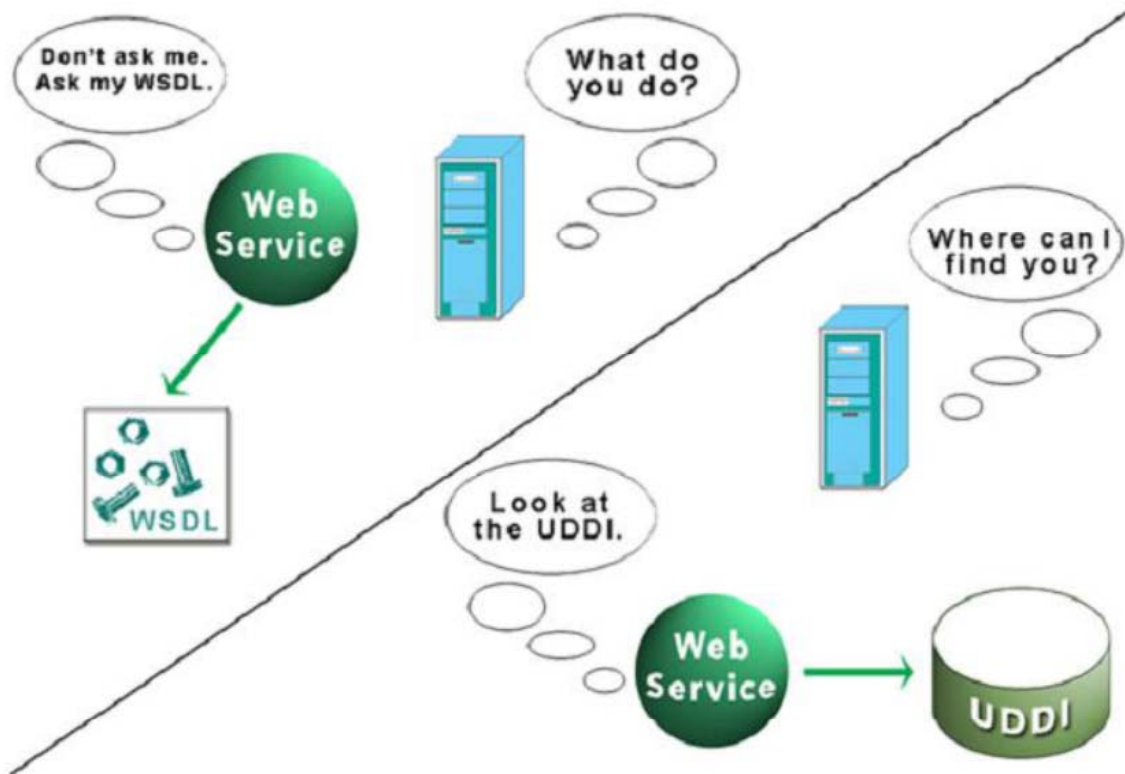
SOA arkitektura. Eredu bat

- SOAP-WSDL-UDDI eredua
 - Web-zerbitzuen deskribapena eta kontsumoa egiteko eredu bat hiru estandarretan oinarritzen dena
 - SOAP (Simple Object Access Protocol)
 - WSDL (Web Services Description Language)
 - UDDI (Universal Description, Discovery and Integration)

14



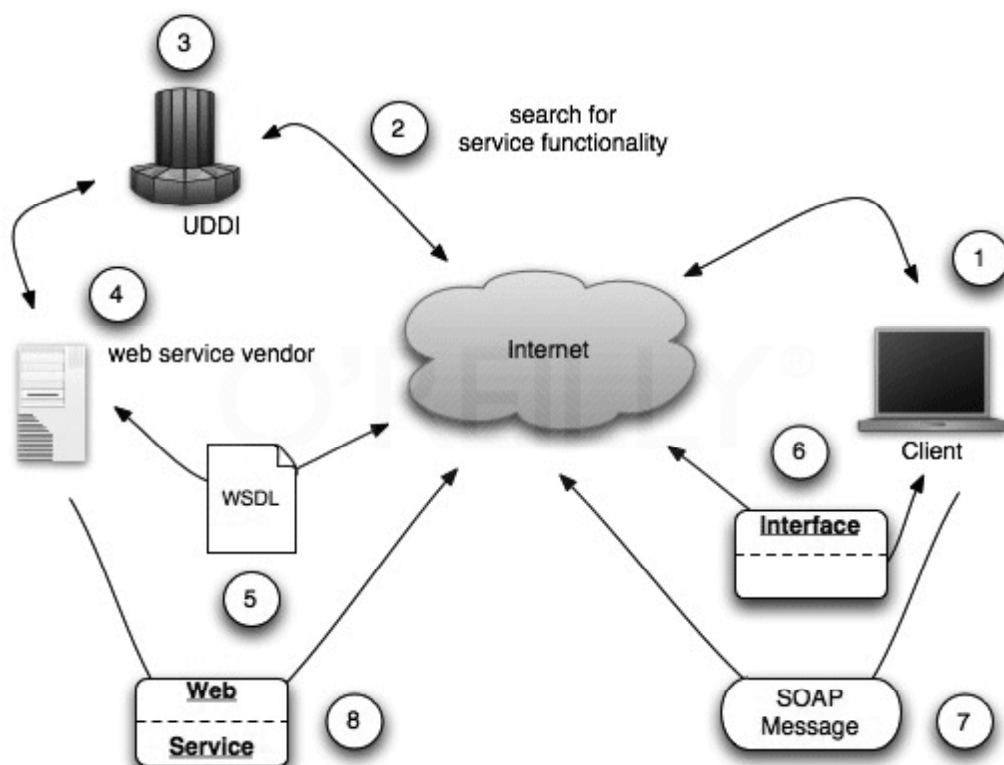
SOAP-WSDL-UDDI ereduua



15



SOAP-WSDL-UDDI ereduua



16

SOAP:

Simple Object Access Protocol



- Komunikazio-protokoloa
- Aplikazioen arteko komunikazioetarako
- Plataforma-independentea
- Lengoaia-independentea
- SOAP **mezuak** bidaltzeko formatua da
- Internet bidez komunikatzen da (**HTTP**)
- XMLn oinarritua: sinplea eta hedagarria
- Suhesiak saihesteko aukera
- Estandarra izan dadin W3Cren babesa du
 - <http://www.w3.org/TR/soap/>

17

SOAP mezuen sintaxia



- SOAP **mezua** XML dokumentua da, honako elementu hauek dituena:
 - Gutun-azala (**envelope**): *Elementu* nagusia, XML dokumentua SOAP mezua dela adierazten du
 - Goiburukoa (**header**): Goiburuko informazioa
 - **Body**: Deiaren eta erantzunaren informazioa
 - **Fault**: Erroreak eta egoerari buruzko informazioa

18

SOAP *eskaera* baten sintaxia (SOAP request)



POST /InStock HTTP/1.1

Host: `www.example.org`

Content-Type: `application/soap+xml; charset=utf-8`

Content-Length: `nnn`

`<?xml version="1.0"?>`

`<soap:Envelope`

`xmlns:soap="http://www.w3.org/2001/12/soap-envelope"`

`soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">`

`<soap:Body xmlns:m="http://www.example.org/stock">`

`<m:GetStockPrice>`

`<m:StockName>IBM</m:StockName>`

`</m:GetStockPrice>`

`</soap:Body>`

`</soap:Envelope>`

19

SOAP *Erantzun* baten sintaxia (SOAP response)



HTTP/1.1 200 OK

Content-Type: `application/soap+xml; charset=utf-8`

Content-Length: `nnn`

`<?xml version="1.0"?>`

`<soap:Envelope`

`xmlns:soap="http://www.w3.org/2001/12/soap-envelope"`

`soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">`

`<soap:Body xmlns:m="http://www.example.org/stock">`

`<m:GetStockPriceResponse>`

`<m:Price>34.5</m:Price>`

`</m:GetStockPriceResponse>`

`</soap:Body>`

`</soap:Envelope>`

20

WSDL

Web Services Description Language



- XML lengoaia bat da, web-zerbitzuak deskribatzeko eta web-zerbitzu horiek nola atzitu behar diren adierazteko
- “kontratua” deskribatzeko erabiltzen da (web-zerbitzuak emango duenaren konpromisoa). Zer eta nola eska litekeen, zer eta nola erantzungo den.
- W3C gomendioa da, Hark sustatua
 - <http://www.w3.org/TR/wsdl20/>

21

WSDL dokumentuak



Lau *elementu* nagusi:

- **<portType>**: Klase kontzeptuaren pareko; **eragiketak** eta sarrera/irteerako ‘**mezuak**’ deskribatzen ditu. *Hau da, eragiketa multzo bat biltzen du, deskribatutako zerbitzuak eskaintzen dituenak.*
- **<message>**: **eragiketen datuen deskribapena**. *Hau da, eragiketa baten sarrera-irteerako mezuen definizio motatua.*
 - ‘mezu’ bakoitzak **<part>** bat edo gehiago eduki dezake. Part *elementu* horiek **funzio-dei tradizionalen parametroekin** pareka daitezke.
- **<types>**: web-zerbitzuan erabiltzen diren datu-moten definizioak. XML eskema batean erabilitakoen berdinak erabiltzen dira.
- **<binding>**: portType jakin baterako protokoloa eta datu-formatua

22

WSDL

Dokumentu (laburtu) baten adibidea



```
<message name="getTermRequest">
  <part name="term" type="xs:string"/>
</message>

<message name="getTermResponse">
  <part name="value" type="xs:string"/>
</message>

<portType name="glossaryTerms">
  <operation name="getTerm">
    <input message="getTermRequest"/>
    <output message="getTermResponse"/>
  </operation>
</portType>
```

23

WSDL: beste dokumentu baten adibidea



```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<definitions xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema" .....
```

```
<types>
  <xsd:schema targetNamespace="http://localhost/nussoap-0.9.5/samples">
    <xsd:import namespace="http://schemas.xmlsoap.org/soap/encoding/" />
    <xsd:import namespace="http://schemas.xmlsoap.org/wsdl/" /> </xsd:schema>
  </types>

  <message name="batuRequest">
    <part name="x" type="xsd:int" />
    <part name="y" type="xsd:int" />
  </message>

  <message name="batuResponse">
    <part name="z" type="xsd:int" />
  </message>

  <portType name="batuPortType">
    <operation name="batu">
      <input message="tns:batuRequest" />
      <output message="tns:batuResponse" />
    </operation>
  </portType>

  .....

  <service name="batu">
    <soap:address location="http://localhost/nussoap-0.9.5/EskolakoAdibideak/batuWSDL.php" />
  </service>
```

SOAP adibidea. PHP bezeroa



Euskarazko testuen analizatzaile morfologikoa

- Zerbitzariaren wsdl fitxategia:

<http://ixa2.si.ehu.es/glabaka/Morfeus.xml> :

```
<xs:element name="morfeusetik_pasa">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="maila" nillable="true" type="xs:string"/>
      <xs:element minOccurs="0" name="sarrera" nillable="true" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="morfeusetik_pasaResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="morfeusetik_pasaResult" nillable="true"
        type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

<wsdl:message name="morfeusetik_pasaRequest">
  <wsdl:part name="parameters" element="ns:morfeusetik_pasa"/>
</wsdl:message>
<wsdl:message name="morfeusetik_pasaResponse">
  <wsdl:part name="parameters" element="ns:morfeusetik_pasaResponse"/>
</wsdl:message>
<wsdl:portType name="MorfeusPortType">
  <wsdl:operation name="morfeusetik_pasa">
    <wsdl:input message="ns:morfeusetik_pasaRequest" wsaw:Action="urn:morfeusetik_pasa"/>
    <wsdl:output message="ns:morfeusetik_pasaResponse"
      wsaw:Action="urn:morfeusetik_pasaResponse"/>
  </wsdl:operation>
</wsdl:portType>
```

25

SOAP adibidea. PHP bezeroa (II)



```
<?php
```

```
$bezeroa = new
  SoapClient( "http://ixa2.si.ehu.es/glabaka/Morfeus.xml" );

$datuak = array( 'maila'=>'1',
                  'sarrera'=>base64_encode( 'probarako testua' ) );

$emaitza = $bezeroa->morfeusetik_pasa( $datuak );

$testua = base64_decode( $emaitza->morfeusetik_pasaResult );

print_r( $testua );
?>
```

26

UDDI

Universal Description, Discovery and Integration



- Direktorio edo biltegi bat da. Bertan web-zerbitzuak, beren interfazea eta beren kokapena deskribatzen da.
- Estandarren arabera diseinatua.
- Gaur egun, enpresek edo erakundeek beren zerbitzuen katalogo propioak izan ohi dituzte.

27

NuSOAP liburutegia



- SOAPn oinarritutako web-zerbitzuak sortzeko eta kontsumitzeko PHP liburutegia da
- WSDL fitxategiak erabiliz edo erabili gabe lan egiteko aukera ematen du
- Web-zerbitzuaren zehaztapen fitxategien (WSDL) sorrera automatikoa ahalbidetzen du
- SOAP zehaztapenen xehetasunak ezkutatzeko, SOAPRequest eta SOAPResponse klaseen instantziak sortuz.

28



NuSOAP

- <http://www.scottnichol.com/nusoapintro.htm>
- <http://sourceforge.net/projects/nusoap/>
(liburutegiaren deskarga)
- <http://www.desarrolloweb.com/manuales/61/>
- <http://www.slideshare.net/fulvio.corno/web-services-in-php-using-the-nusoap-library>

29

SOAP eta PHP

<http://php.net/manual/es/book.soap.php>

- PHPn SOAP hedapenak ematen du aukera SOAP zerbitzariak eta bezeroak erabiltzeko
- SoapClient klasea:

```
SoapClient {  
    // Methods  
public mixed __call( DEPRECATED)  
    public SoapClient ( mixed $wsdl [, array $options ] )  
    public string __doRequest ( string $request , string $location , string  
        $action , int $version [, int $one_way = 0 ] )  
    public array __getFunctions ( void )  
    public string __getLastRequest ( void )  
    public string __getLastRequestHeaders ( void )  
    public string __getLastResponse ( void )  
    public string __getLastResponseHeaders ( void )  
    public array __getTypes ( void )  
    public void __setCookie ( string $name [, string $value ] )  
    public string __setLocation ( [ string $new_location ] )  
    public bool __setSoapHeaders ( [ mixed $soapheaders ] )  
    public mixed __soapCall ( string $function_name , array $arguments [,  
        array $options [, mixed $input_headers [, array &$amp;output_headers ]]]  
    )  
    public SoapClient ( mixed $wsdl [, array $options ] )  
}
```

30





soap

Soap Client	enabled
Soap Server	enabled

Directive	Local Value	Master Value
soap.wsdl_cache	1	1
soap.wsdl_cache_dir	/tmp	/tmp
soap.wsdl_cache_enabled	1	1
soap.wsdl_cache_limit	5	5
soap.wsdl_cache_ttl	86400	86400

- -

31

SOAP eredua. Alderdi onak eta ez hain onak



- SOAP+WSDL ereduko web-zerbitzuen erabilerak kontratu bidezko protokoloa finkatzen du, estandar irekietan oinarrituta
- Hobetzen da komunikazioa, erabilgarritasuna eta plataforma-independente izatea
- Baina, zerbitzuaren menpe geratzen da bezeroa, eta kontratua aldatzen bada?

32

Beste eredu bat: *REST* *REpresentational State Transfer*



- Web-zerbitzu *arinak*
- ROA eredua (*Resource-Oriented Architecture*)
- Zuzenean HTTP erabiltzen da, ez dago bestelako gutun-azalik

33

Beste eredu bat: *REST*: *REpresentational State Transfer*



- CORBA, RPC edo SOAP bezalako eredu konplexuak erabili ordez, makinen arteko HTTP dei sinpleak erabiltzen dira.
- Baina Web Zerbitzuen ezaugarriei eusten dio ereduak
 - Plataformatik independentea
 - Lengoaiatik independentea
 - Estandarretan oinarritua
 - Suhesiekin-eta erabiltzeko modukoa
- RESTful aplikazioek HTTPko eskaerak erabiltzen dituzte (GET, POST, PUT, DELETE) beste inolako gutun-azalik gabe. HTTP eskaeran doa behar den informazio guztia.
- HTTP bidez dator erantzuna: testu soila, XML, JSON

34



REST arkitektura

- *Baliabideak* (hizkeran ez da erabiltzen "metodo" edo "zerbitzu" bezalako terminorik)
 - URL logikoen bidez adierazten dira
- Bezero-zerbitzari eredua
- Elkarrekintza bakoitza beregaina da, eta konexioak aldiro ezartzen dira
- Eskaerak eta erantzunak: arinak

35



REST Javascript-ez (AJAX)

- Neurri handi batean, AJAXek REST oinarriak jarraitzen ditu

XmlHttpRequest

- Diseinua:
 - Kontuan hartu GET eta POST bereizketa
 - Zerbitzariaren egoera aldatetarako ez erabili GET
 - Erabili URL logikoak, ez fisikoak
 - Erantzunak:
 - Ez daitezela pisuak izan
 - Ondo dokumentatua

36



REST *PHPz*

- Get eskaeretarako:
 - `file_get_contents` funtzioa
- Bestelako eskaeretarako, HTTP eskaera osoa antolatu behar da



REST vs. SOAP

- Konparaziorako: *Postalak eta eskutitzak*
- *Batzuen iritziz:*
- *SOAP helduagoa da, baina REST gero eta gehiago erabiltzen da (bere erraztasun eta arintasunagatik)*

REST baliabideen dokumentazioa: WADL



- ***WADL, the Web Application Description Language***
- XMLz idazten da
- WSDL baino arinagoa eta ulerterrazagoa

39

REST API bat PHPz. Adibidea



- <http://blog.ijasoneverett.com/2013/02/rest-api-a-simple-php-tutorial/>

40

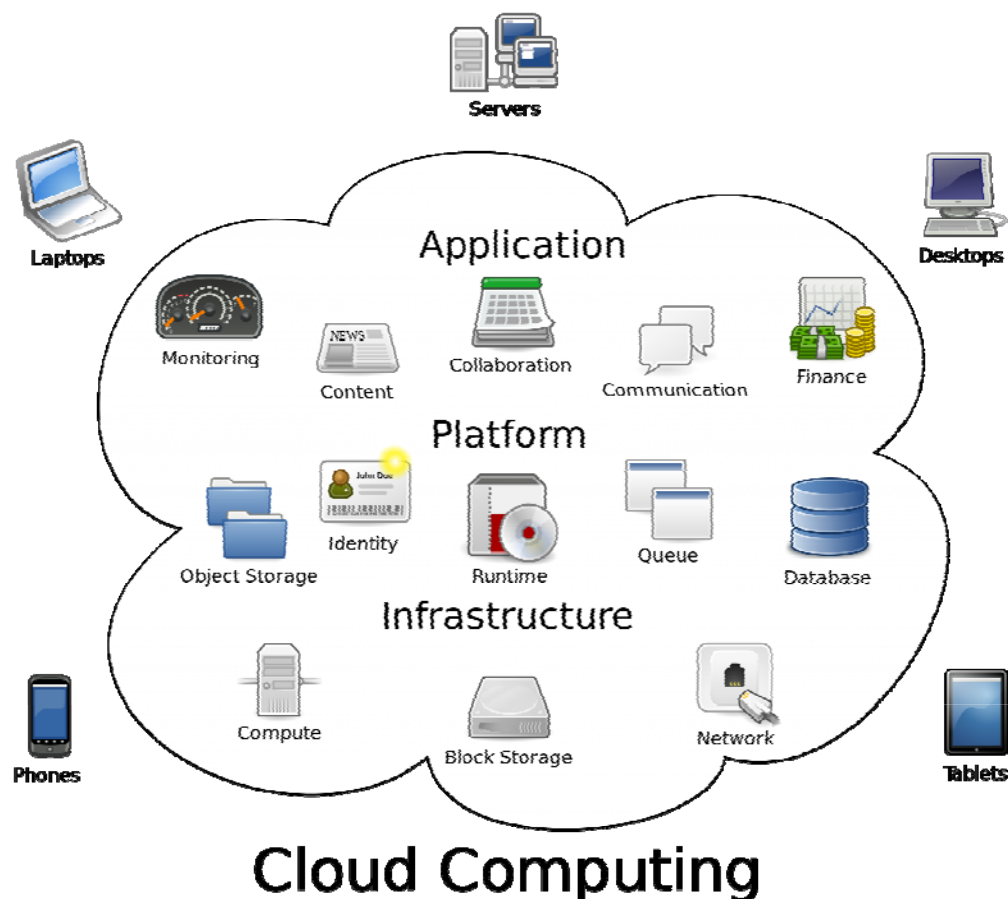


Mashup

- Webgune bat, edo web-aplikazio bat, iturri desberdinetako edukia erabiltzen dituen, bere interfaze grafikoan txertatuz.
- Iturri horien APIak dira baliatzen direnak
- Adibidez, *Google Maps webgune askotan txertatzen da, edo Twitter*
- Adibideak ikusi eta aztertzeko:
 - <http://www.programmableweb.com>

41

Cloud computing



42

Adibidea1: batu.php

SOAP “batu” metodo baten sorrera



```
<?php
//nusoap.php klasea gehitzen dugu
require_once('../lib/nusoap.php');
require_once('../lib/class.wsdlcache.php');

//soap_server motako objektua sortzen dugu
$server = new soap_server;

//inplementatu nahi dugun funtzioa erregistratzen dugu
//funtzio bat baino gehiago erregistra liteke ...
$server->register('batu');

//funtzioa inplementatzen da
function batu($x, $y){
return $x + $y;
}
//nusoap klaseko service metodoari dei egiten diogu
$HTTP_RAW_POST_DATA = isset($HTTP_RAW_POST_DATA) ? $HTTP_RAW_POST_DATA : '';
$server->service($HTTP_RAW_POST_DATA);
?>
```

43

`$server->service($HTTP_RAW_POST_DATA);`



- SOAP eskaerako xml zatia eduki behar du
`$server->service($HTTP_RAW_POST_DATA);`
- **Service** metodoak , XMLa aztertzen du, funtzioari dei egiten dio eta erantzuna sortzen du
- **Service** metodoari deia egin aurretik aldagai horren instantzia batek existitu behar du. Parametrorik pasa ez bada, orduan balio bezala string hutsa esleitzen zaio
- `HTTP_RAW_POST_DATA= isset(HTTP_RAW_POST_DATA)?
$HTTP_RAW_POST_DATA:'';`

44

Adibidea2: Bezero baten sorrera ...



```
<FORM NAME="datuak" ID=" datuak" ACTION= "batuketaEskatu.php" METHOD="POST">
<input type='text' name='batugail'></input>
<input type='text' name='batugai2'></input>
<input type='submit' name='batu' value='BATU'></input>
</form>

<?php
//nusoap.php klasea gehitzen dugu
require_once('../lib/nusoap.php');
require_once('../lib/class.wsdlcache.php');

//soapclient motadun objektua sortzen dugu. http://www.mydomain.com/server.php
//erabiliko den SOAP zerbitzua non dagoen zehazten url horrek
$soapclient = new nusoap_client( 'http://localhost/nusoap-
0.9.5/samples/batu.php?wsdl', false);

//Web-Service-n inplementatu dugun funtzioari dei egiten diogu
//eta itzultzen diguna inprimatzen dugu
if (isset($_POST['batugail'])) {
echo '<h1>Batuketa da: ' . $soapclient->call('batu',array( 'x'=>$_POST['batugail'],
'y'=>$_POST['batugai2batugail'])). '</h1>';
echo '<h2>Request</h2><pre>'.htmlspecialchars($soapclient->request, ENT_QUOTES).'</pre>';
echo '<h2>Response</h2><pre>'.htmlspecialchars($soapclient->response, ENT_QUOTES).'</pre>';

//echo '<h2>Debug</h2>';
//echo '<pre>' . htmlspecialchars($soapclient->debug_str, ENT_QUOTES) . '</pre>';
}
?>
```

45

Adibidea3: WSDL zerbitzuaren sorrera



```
<?php
//nusoap.php klasea gehitzen dugu
require_once('../lib/nusoap.php');
require_once('../lib/class.wsdlcache.php');

//soap_server motako objektua sortzen dugu
$ns="http://localhost/nusoap-0.9.5/samples"; //name of the service
$server = new soap_server;
$server->configureWSDL('batu', $ns);
$server->wsdl->schemaTargetNamespace=$ns;

//inplementatu nahi dugun funtzioa erregistratzen dugu
$server->register('batu',
array('x'=>'xsd:int', 'y'=>'xsd:int'),
array('z'=>'xsd:int'),
$ns);

//funtzioa inplementatzen dugu
function batu($x, $y){
return $x + $y;
}
//nusoap klaseko service metodoari dei egiten diogu
$HTTP_RAW_POST_DATA = isset($HTTP_RAW_POST_DATA) ? $HTTP_RAW_POST_DATA : '';
$server->service($HTTP_RAW_POST_DATA);
?>
```

46

Adibidea3: WS zerbitzu publiko bat kontsumitzen



```
<?php
/////nusoap.php klasea gehitzen dugu
require_once(' ../lib/nusoap.php');
require_once(' ../lib/class.wsdlcache.php');

//soapclient motako objektua sortzen dugu http://www.mydomain.com/server.php
//erabiliko den SOAP zerbitzua non dagoen zehazten url horrek
$soapclient = new soapclient(
    'http://footballpool.dataaccess.eu/data/info.wso?Wsd1',true);

//Web-Service-n inplementatu dugun funtzioari dei egiten diogu
//eta itzultzen diguna inprimatzen dugu
$result = $soapclient->call('YellowAndRedCardsTotal');
print_r($result);
echo '<br>';
echo '<b>EUROCOPA 2012:</b> <br>';
echo 'Txartel Horiak:' . $result['YellowAndRedCardsTotalResult']['iYellow'];
echo '<br>';
echo 'Txartel Gorriak:' . $result['YellowAndRedCardsTotalResult']['iRed'];
?>
```

47

Adibidea4: WS zerbitzu publiko bat kontsumitzen parametroak pasaz



```
<?php
//nusoap.php klasea gehitzen dugu
require_once(' ../lib/nusoap.php');
require_once(' ../lib/class.wsdlcache.php');

//soapclient motako objektua sortzen dugu
//erabiliko den SOAP zerbitzua non dagoen zehazten urla
$soapclient = new soapclient(
    http://footballpool.dataaccess.eu/data/info.wso?Wsd1',true);

//Web-Service-n inplementatu dugun funtzioari dei egiten diogu
//eta itzultzen diguna inprimatzen dugu
$result = $soapclient->call('TopGoalScorers', array('iTopN'=>'10'));

echo '<br>';
echo '<b>EUROCOPA 2012: GOLEATZAILE ONENAK</b> <br>';

for ($i = 0; $i <= count($result['TopGoalScorersResult']['tTopGoalScorer'])-1; $i++) {

    echo 'JOKALARIA:' .
        $result['TopGoalScorersResult']['tTopGoalScorer'][$i]['sName'] . "<br />";
    echo 'GOLAK:' .
        $result['TopGoalScorersResult']['tTopGoalScorer'][$i]['iGoals'] . "<br />";
    echo '<br>';
}
?>
```

48

Adibidea5: “batu” metodoari SOAP eskaera



POST /WebService4/Service.asmx HTTP/1.1

Host: localhost

Content-Type: text/xml; charset=utf-8

Content-Length:

SOAPAction: "http://www.ehu.es/Batu"

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema/">
  <soap:Body>
    <Batu xmlns="http://www.ehu.es/">
      <bat>6</bat>
      <bi>4</bi>
    </Batu>
  </soap:Body>
</soap:Envelope>
```

49

Adibidea5: “batu” metodoari SOAP erantzuna



HTTP/1.1 200 OK

Content-Type: text/xml; charset=utf-8

Content-Length:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema ...">
  <soap:Body>
    <BatuResponse xmlns="http://www.ehu.es/">
      <BatuResult>10</BatuResult>
    </BatuResponse>
  </soap:Body>
</soap:Envelope>
```

50



Helbide interesgarriak

- SOAP
 - www.w3schools.com/webservices/
- SOAP eta PHP:
 - <http://www.php.net/manual/en/book.soap.php>
- WSDL
 - http://www.w3schools.com/xml/xml_wsdl.asp

51



Helbide interesgarriak

- <http://www.programmableweb.com/>
- REST:
 - <http://rest.elkstein.org/2008/>
 - <http://net.tutsplus.com/tutorials/other/a-beginners-introduction-to-http-and-rest/>
 - <http://www.restapitutorial.com/>
 - <http://blog.ijasoneverett.com/2013/02/rest-api-a-simple-php-tutorial/>
- Mashup
 - <http://es.slideshare.net/webdirections/javascript-apis-and-mashups>

52



Adibideak

- Tresna linguistikoak
 - <http://clarin-es-lab.org/>
 - http://langrid.org/service_manager/language-services/profile/kyoto1.langrid/GoogleTranslate
- Biltegiak
 - <http://www.service-repository.com/>