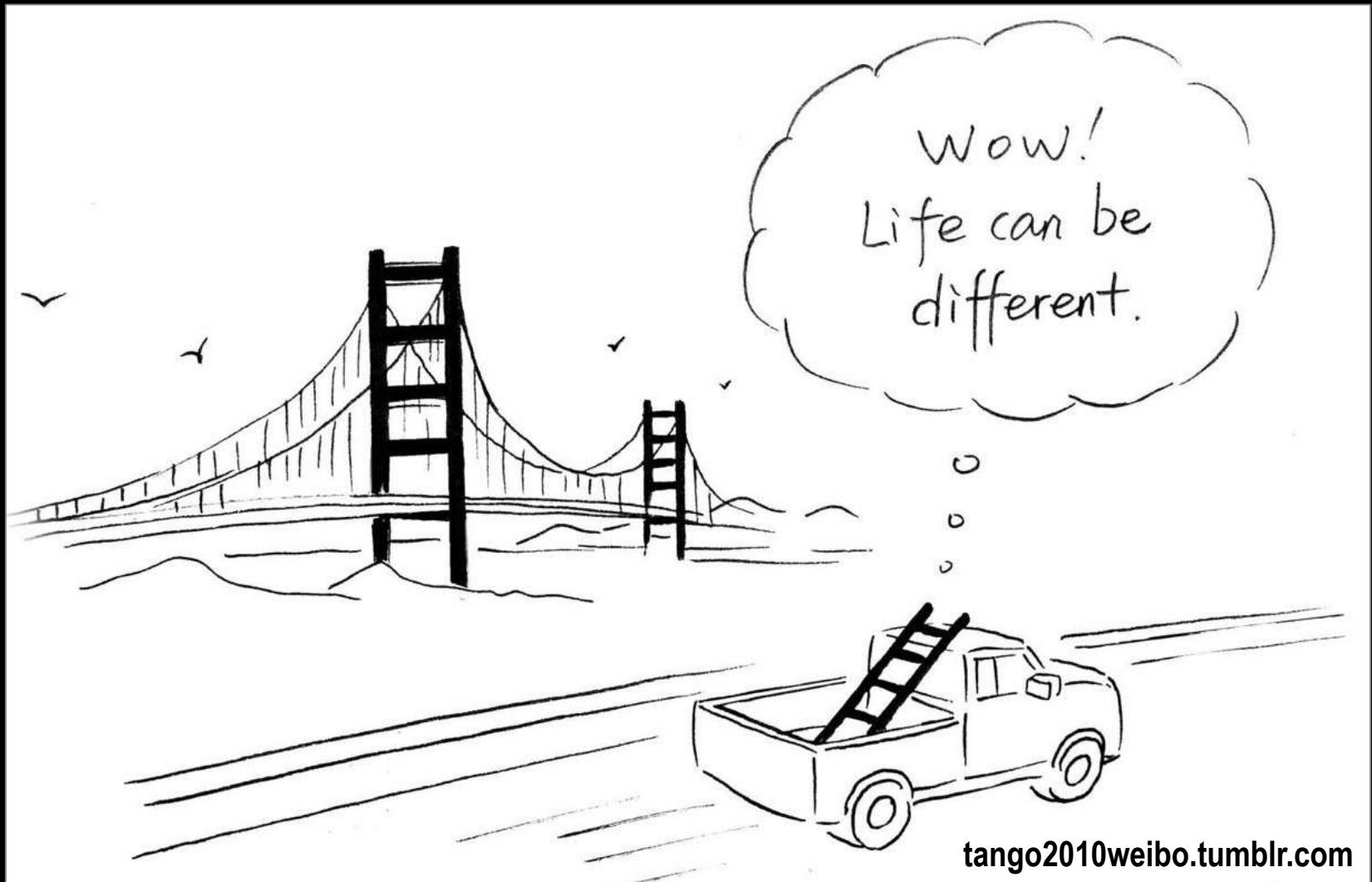


Tehnologii Web



servicii Web (I)
de la SOA la SOAP, WSDL și UDDI

„Prețuim ceea ce înțelegem.”

Kevin Budelmann

Care sunt scopurile Web-ului?

Constituirea și interacțiunea cu un spațiu de comunicare inter-umană

- ▶ partajarea cunoștințelor

Constituirea și interacțiunea cu un spațiu de comunicare inter-umană

- ▶ partajarea cunoștințelor

Web social (“Web 2.0”), Web al datelor (semantic),...

Exploatarea puterii computaționale

accesul la Web se poate realiza
via dispozitive având resurse reduse

Exploatarea puterii computaționale

accesul la Web se poate realiza
via dispozitive având resurse reduse

Web ubicuu (omniprezent): **Web mobil, Web 3D,...**
performanță ► asigurarea scalabilității

remarcă

Interacțiunea dintre om și Web se rezolvă
prin intermediul formularelor Web și
explorarea legăturilor via adrese Web – URI-uri

Cum pot fi accesate și procesate resursele
– date, informații, cunoștințe –
disponibile pe Web?

nevoi ale dezvoltatorilor Web

Soluții multi-platformă, slab-conectate

integrare (în timp-real) la nivel de Internet/Web
a aplicațiilor, serviciilor și sistemelor

nevoi ale dezvoltatorilor Web

Soluții multi-platformă, slab-conectate

integrare (în timp-real) la nivel de Internet/Web
a aplicațiilor, serviciilor și sistemelor

exemplificare: găsirea ofertelor de servicii,
pe baza localizării geografice a utilizatorului,
în contextul dispozitivelor mobile

nevoi ale dezvoltatorilor Web

Soluții multi-platformă, slab-conectate

datele să poată fi descrise pentru a fi „înțelese”
de calculatoare și pentru a fi interconectate facil

nevoi ale dezvoltatorilor Web

Soluții multi-platformă, slab-conectate

datele să poată fi descrise pentru a fi „înțelese” de calculatoare și pentru a fi interconectate facil

Web “puzzles”

- ▶ inter-conectarea mai multor servicii oferind date de interes, conform preferințelor utilizatorului

nevoi ale dezvoltatorilor Web

Servicii ataşabile (*pluggable*) & versatile

Software as a Service – SaaS

Application Service Provider – ASP

soluție

Divizarea aplicațiilor în **servicii** – independente –
care se pot **compune**,
menite a se **conecta** și **orchestra** în mod **spontan**
în cadrul proceselor de afaceri/tehnice

Web component-based software

soluție

“The Web is the computer”

disponibilitatea unei/unor arhitecturi care...

oferă suport pentru paradigme de comunicare
– bazată pe actualele tehnologii Web –
între aplicații eterogene

soluție

“The Web is the computer”

disponibilitatea unei/unor arhitecturi care...

permit(e) localizarea transparentă a serviciilor

soluție

“The Web is the computer”

disponibilitatea unei/unor arhitecturi care...

facilitează adăugarea, înlocuirea, eliminarea
serviciilor în mod dinamic

soluție

“The Web is the computer”

disponibilitatea unei/unor arhitecturi care...

ascund(e) dezvoltatorului detaliile de sistem

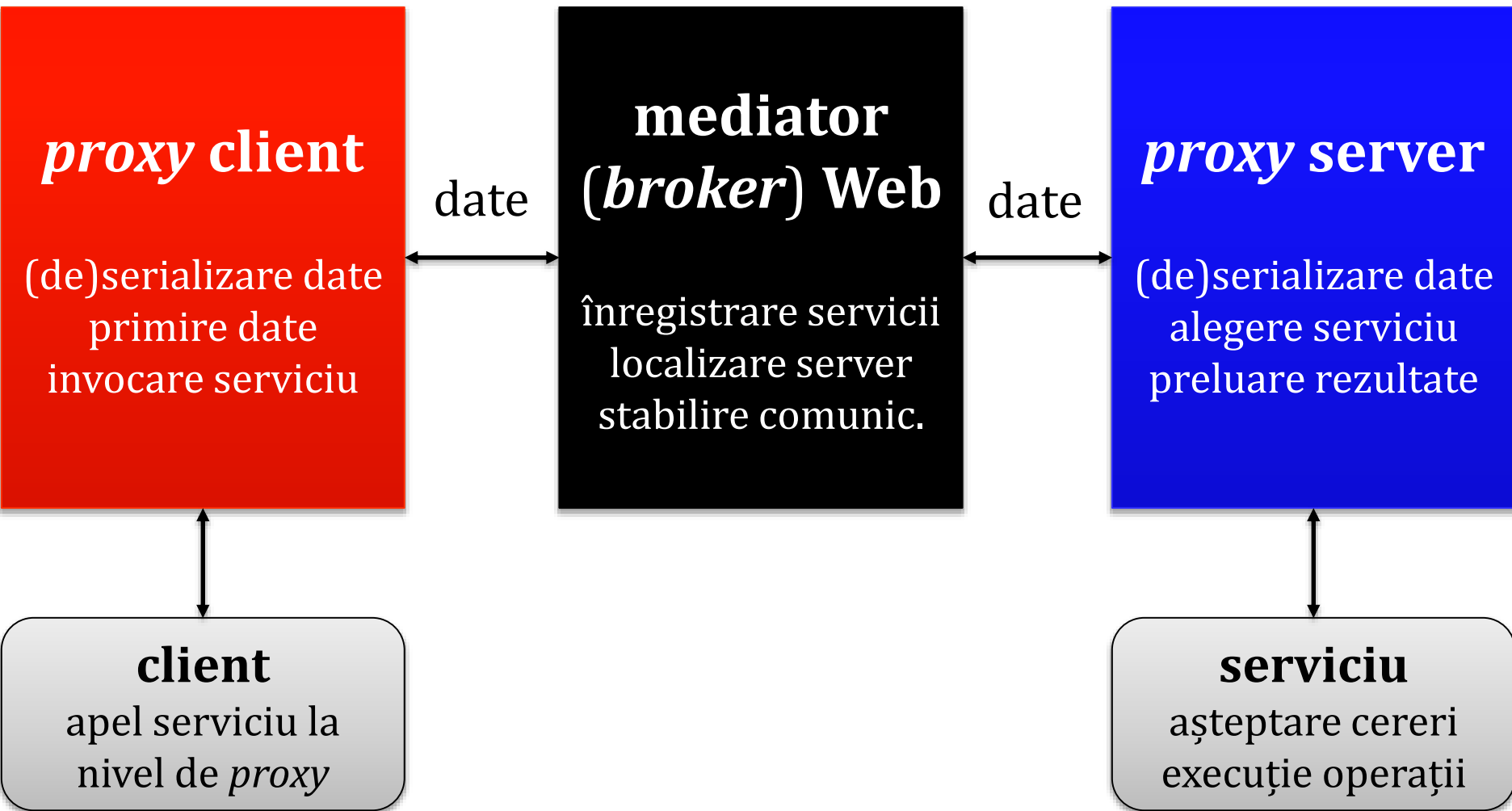
soluție

“The Web is the computer”

disponibilitatea unei/unor arhitecturi care...

asigură calitatea dezvoltării și exploataării
aplicațiilor distribuite și/sau paralele:
**standardizare, securitate, disponibilitate,
reutilizare, mentenanță etc.**

Web-ul ca tehnologie *middleware*



Ce sunt serviciile Web?

servicii web

*“A **service** is an abstract resource that represents a capability of performing tasks that form a coherent functionality from the point of view of providers entities and requesters entities.”*

www.w3.org/TR/ws-gloss/

servicii web

“A **Web service** *is a software system designed to support interoperable machine-to-machine interaction over a network.*”

www.w3.org/TR/ws-gloss/

servicii web

Software oferind o funcționalitate specifică

- acces la resurse – Instagram, Pinterest, Slideshare, Vimeo
- agregare de știri – Feedly, Reddit
- cartografiere – Google Maps, Nokia HERE, OpenStreetMap
- mesagerie instantanee – Telegram, Twilio etc.
- procesări – Amazon Rekognition, IBM Watson, Microsoft Azure Cognitive Services, Google TensorFlow...
- realizare de statistici Web – e.g., Google Analytics
- rețele sociale – Facebook Open Graph, LinkedIn, Twitter
- spelling checking* – Bing Spell Check, xSpell,...
- stocare de date – Amazon S3, Dropbox, OneDrive etc.

...

Search headlines...

All sites ▾ Any time ▾



CSS-Tricks

[latest](#) • [day](#) • [week](#) • [month](#)



Medium Programming

[latest](#) • [day](#) • [week](#) • [month](#)



Rearrange



Preferences



Subscribe



Send feedback

Made with love by [Browserling](#) and powered by alien technology.

- 11h [Inclusively Hidden](#)
- 12h [Tabs: It's Complicated™](#)
- 16h [How to Get a Progressive Web App into the Google Play Store](#)
- 2d [Faking env\(\) to Use it Now](#)
- 2d [Creating Reusable Base Classes in TypeScript with a Real-Life Example](#)
- 2d [An Event Apart Boston is Coming. Save Now!](#)
- 3d [Clever code](#)
- 4d [The Power of Named Transitions in Vue](#)

- 6h [Linear programming and discrete optimization with Python using PuLP](#)
- 9h [Git...Huh?](#)
- 9h [A beginner's guide to Docker — how to create a client/server side with docker-compose](#)
- 13h [How to create a Buttons UI Kit](#)
- 14h [How to write better conditionals in JavaScript](#)
- 15h [When to use a function declaration vs. a function expression](#)

- 2h [Learn Programming with Go - A Live Coding Crash Course!](#)
- 4h [Xcode help!](#)
- 6h [Microsoft Bing Fire Tokenizer – 10x Faster Than NLTK](#)
- 8h [Microsoft's New Programming Language Bosque Keeps Your Code Simple](#)
- 8h [Already know the interview code question? Don't admit it!](#)
- 9h [Free Angular Tutorial: Building](#)

utilizate – la distanță – de alte aplicații/servicii

servicii web

Accesate standardizat via Web

adresare de resurse cu URI

transfer de date via HTTP

mesaje adoptând formate de date: CSV, JSON, XML,...

servicii web: exemplu

Serviciul unei agenții de turism
oferirea – și vânzarea, eventual –
a unor formule de petrecere a vacanței

servicii web: exemplu

Serviciul unei agenții de turism
utilizează alte servicii (software)
disponibile la nivel de Web

servicii cartografice + meteo
servicii hoteliere

tranzacții financiare – *e.g., e-banking*

servicii de transport

servicii de recomandare socială

Cum am putea implementa un serviciu?

servicii web

Implementare standard

recurgerea la servere + *framework*-uri de aplicații Web



ASP.NET, Django, Node.js,
PHP (CodeIgnater, Laravel,...),
Play!, Ruby on Rails,...

servicii web

Tradițional, aplicația oferă o interfață-utilizator disponibilă pe Web

limbaj de marcare – *e.g.*, HTML

stiluri de prezentare a conținutului – CSS

interactivitate via JavaScript (+biblioteci/*framework*-uri)

servicii web

Tradițional, aplicația oferă o interfață-utilizator disponibilă pe Web

cererile sunt capt(ur)ate via formulare
+ legături hipermedia

servicii web

Tradițional, aplicația oferă o interfață-utilizator disponibilă pe Web

utilizatorii umani trebuie să interpreteze etichetele și câmpurile de dialog

33	Euro
157.01	Romanian Leu

servicii web

Tradițional, aplicația oferă o interfață-utilizator disponibilă pe Web


serviciul implementat oferă un răspuns
(o reprezentare a unei resurse Web)

uzual, un document HTML al cărui conținut
e transferat la client conform unui protocol: HTTP(S)

servicii web

Cum obținem răspunsul pentru a fi (re)folosit
în programele noastre?

procesarea datelor din codul HTML ► ***Web scraping***



a se revedea
cursul anterior

servicii web

Cum obținem răspunsul pentru a fi (re)folosit
în programele noastre?



orice modificare în marcaje ► rescrierea programului
de preluare a datelor din documentul HTML

servicii web: caracterizare

Serviciile Web fac explicite specificațiile implicite

datele de intrare și răspunsul pot fi specificate (riguros)
via diverse maniere de validare

servicii web: caracterizare

Utilizate la interacțiunea dintre aplicații

dinamice

lipsa unei cunoașteri *a-priori* a interacțiunii
cu alte aplicații/servicii Web

servicii web: caracterizare

Puncte finale utilizate pentru procesarea datelor,
în manieră publică – eventual, via API-uri deschise

servicii web: caracterizare

Dezvoltate pe baza platformelor, arhitecturilor,
tehnologiilor și limbajelor curente

Există un model arhitectural de dezvoltare
a serviciilor la nivel de Web?

soa

Arhitectura orientată spre servicii

Service Oriented Architecture

soa

Stil arhitectural de proiectare și dezvoltare
de aplicații considerate drept servicii
care pot fi invocate de alte aplicații

soa

Paradigmă de dezvoltare a software-ului
care adoptă folosirea de servicii,
oferind funcționalități solicitate de utilizatori

soa

Paradigmă de dezvoltare a software-ului
care adoptă folosirea de servicii,
oferind funcționalități solicitate de utilizatori

resursele sunt disponibile via o suită de servicii
independente ale căror implementări
nu trebuie să fie cunoscute (*black box*)

soa

Componentele sistemului în ansamblu
au un grad mare de **independență** (*de-coupling*)

soa

Componentele sistemului în ansamblu
au un grad mare de **independență** (*de-coupling*)

serviciile trebuie proiectate să interacționeze
fără a exista **dependențe** între acestea

soa

Serviciile partajează un **contract formal**

necesitatea unei descrieri formale a serviciului:
operații oferite (interfața serviciului)

maniera de interschimb a datelor (cerere + răspuns)

maniera de descoperire (*service discoverability*)

calitatea unui serviciu (SLA – *service-level agreement*)

soa

Serviciile pot fi recompuse/orchestrate
conform cerințelor sau contextului de exploatare

composable services
reusable services

soa

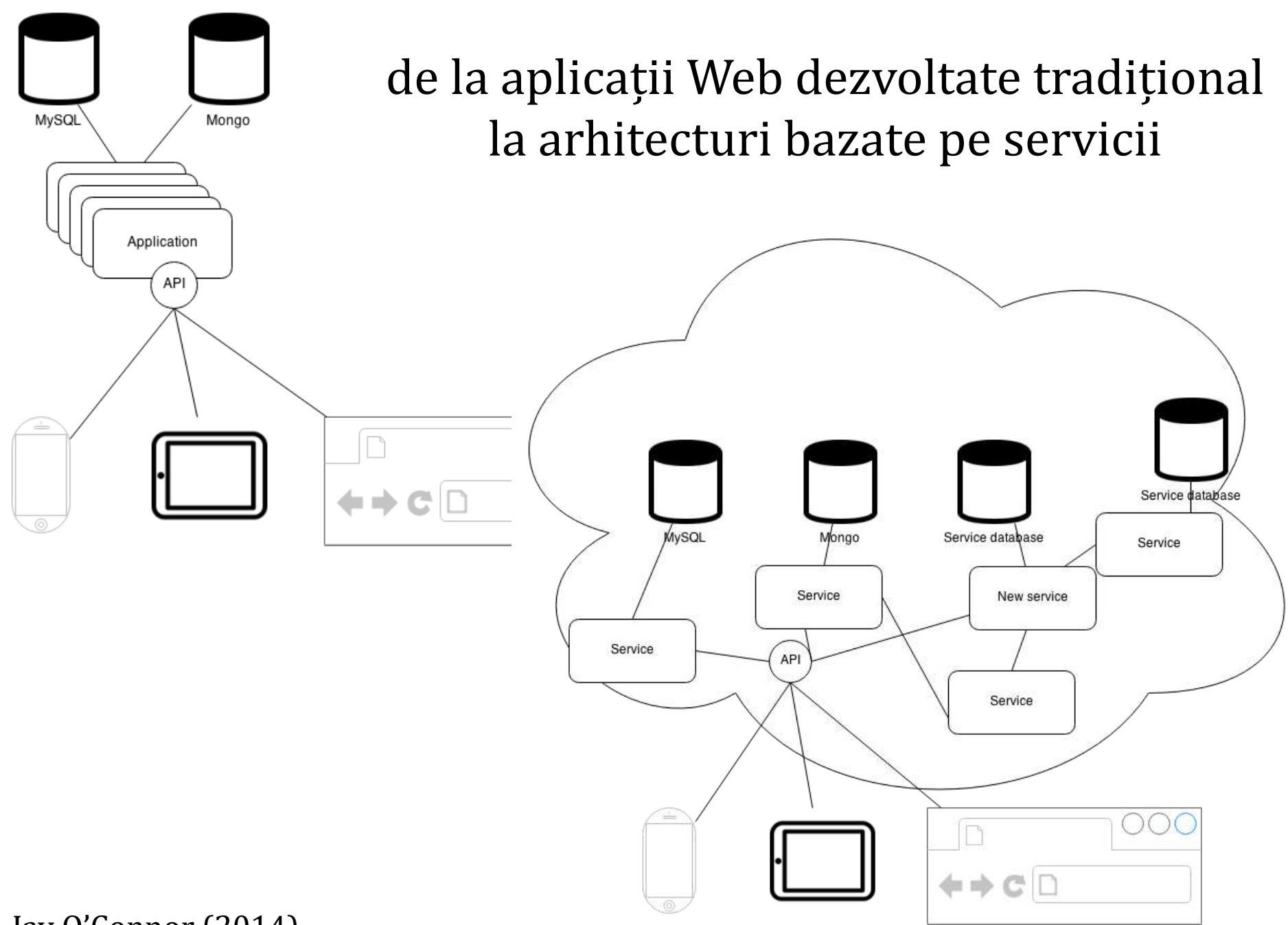
Serviciile nu vor depinde de starea comunicării
(*statelessness*)

soa

Serviciile nu vor depinde de starea comunicării
(*statelessness*)

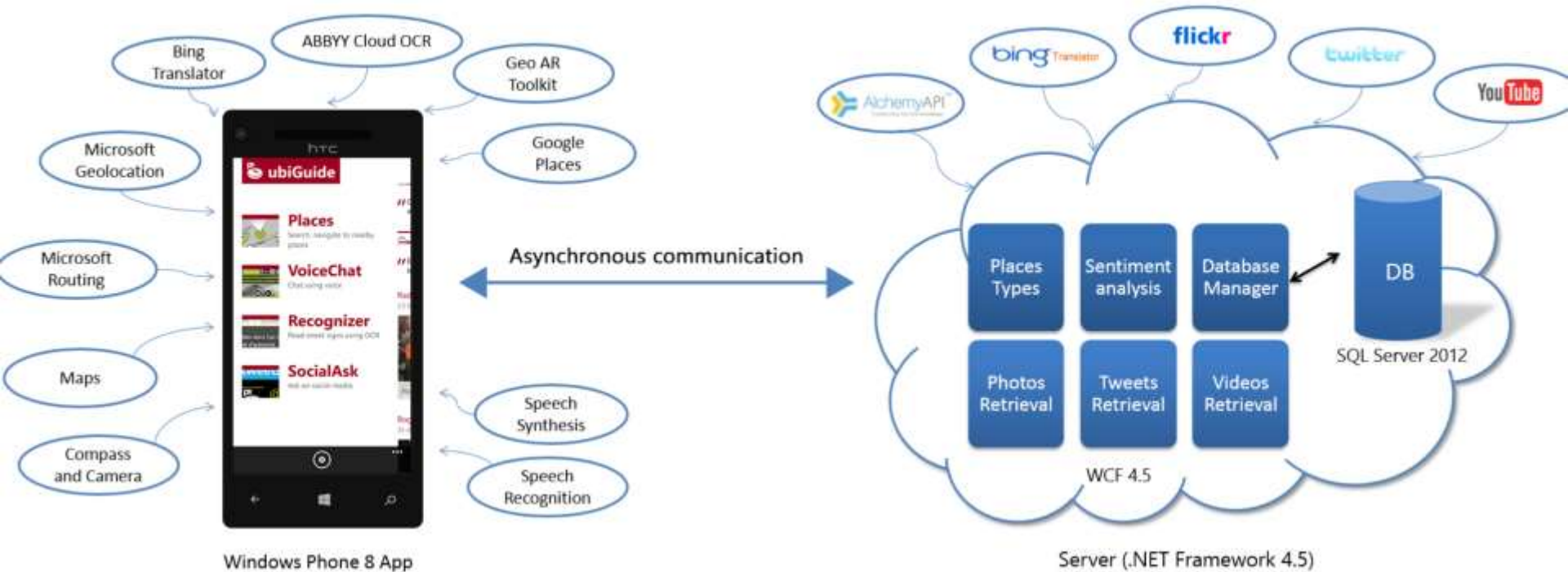
pentru a efectua o procesare, cantitatea de informație
ce trebuie reținută trebuie să fie minimală

de la aplicații Web dezvoltate tradițional la arhitecturi bazate pe servicii



Jay O'Connor (2014)

tech.bellycard.com/blog/migrating-to-a-service-oriented-architecture-soa/



proiectul **ubiGuide**

(absolvenții FII Ionuț Dănilă & Mihaela Ghimiciu, 2013—2014)

servicii publice folosite: ABBYY Cloud OCR, AlchemyAPI,

Google Places API, YouTube API,...

biblioteci: Flickr.NET, GART (*Geo Augmented Reality Toolkit*),

Hammock, TweetSharp etc.

www.youtube.com/watch?v=wygXE6hQ07c

www.slideshare.net/ionutdanila/ubi-guide

Putem indica într-un format standardizat
datele de intrare și răspunsul oferit de
serviciul Web?

invocare

Necesitatea unui **protocol de comunicare**
(**transport**) de date
între platforme/aplicații eterogene

invocare

Protocolul va trebui să ofere un mecanism de **invocare** și de **transmitere** a datelor în mod structurat

invocare

Protocolul va trebui să ofere un mecanism de **invocare** și de **transmitere** a datelor în mod structurat

facilitarea de interacțiuni complexe între aplicații

asigurarea extensibilității + securitate, fiabilitate, *caching*

invocare: soluție

XML-RPC (1999, 2003)

simplic de utilizat, nepretențios

bazat pe apeluri de proceduri la distanță
RPC (*Remote Procedure Call*)

mesajele sunt modelate în XML

xmlrpc.scripting.com/spec

invocare: soluție

SOAP

sofisticat, mai flexibil

suită de standarde W3C (2007)

utilizat cu precădere în aplicații de tip *enterprise*

www.w3.org/TR/soap12/

invocare: soluție

Recurgerea la alte reprezentări

CSV (*Comma Separated Values*)

POX (*Plain Old XML*)

JSON (*JavaScript Object Notation*)

dezvoltatorul realizează metode proprii de serializare

invocare: soap

Scop:

protocol de comunicație între două mașini
(client și server) pentru interschimb de date XML

independent de platformă/limbaj de programare

invocare: soap

Standard al Consorțiului Web (2003, 2007)

www.w3.org/TR/soap12-part0/

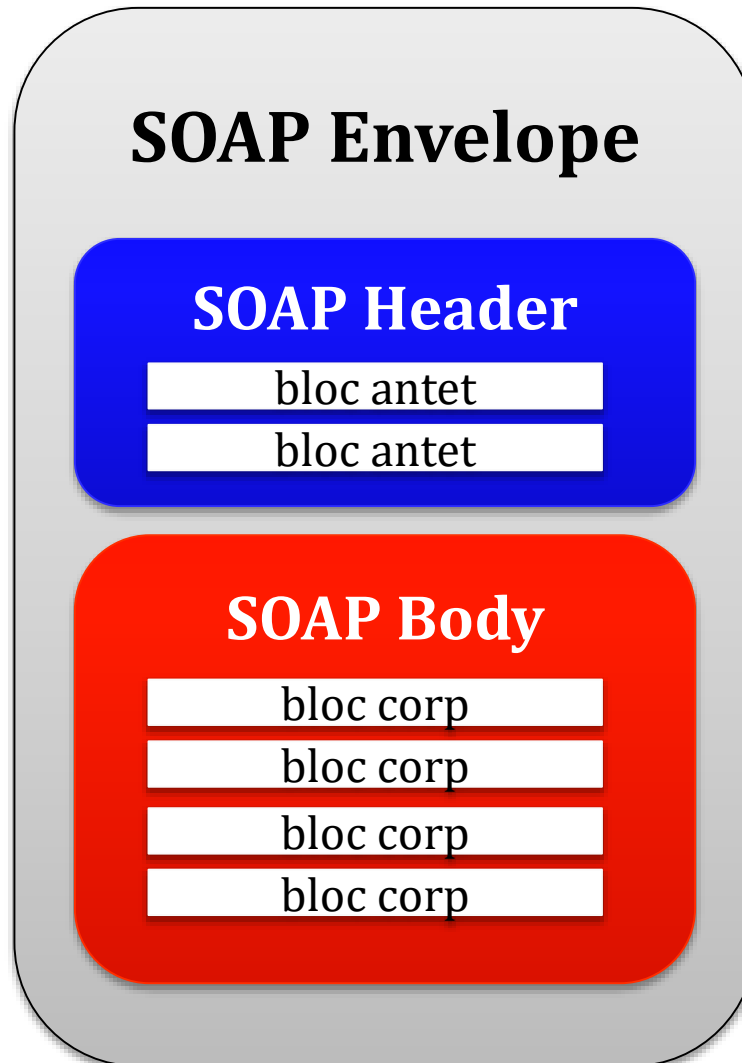
invocare: soap

„Dialogul” dintre aplicații are loc
via **mesaje SOAP** \equiv documente XML

plic (*envelope*)
antet + corp

XML messaging

invocare: soap



invocare: soap

Se poate descrie un model de procesare
a conținutului

SOAP encoding rules

invocare: soap

Poate specifica o cale de la expeditor la destinatar,
via un intermediar (*proxy*) opțional

SOAP routing

invocare: soap

Anteturile pot fi procesate de intermediari diferiți

invocare: soap

Datele XML din corp pot fi transportate indiferent
de protocolul folosit

uzual, HTTP

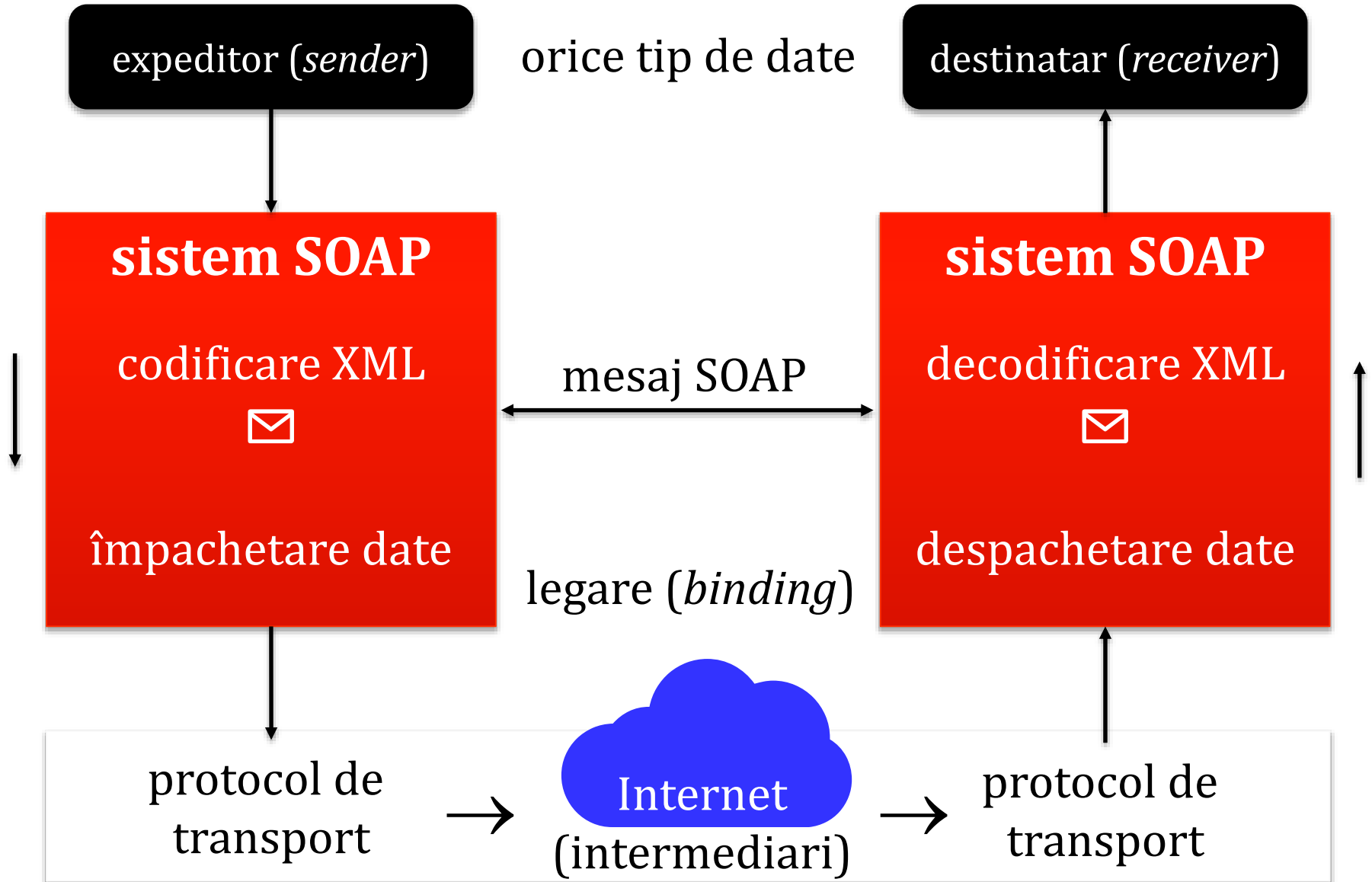
pot fi adoptate și alte protoacoale – *e.g.*, SMTP, XMPP

invocare: soap

Prin SOAP poate fi specificat un apel de metodă disponibilă pe alt calculator

SOAP RPC representation

Vehicularea mesajelor SOAP



(în loc de) pauză



invocare: soap – exemplu

Scenariu:

un client trimite serviciului
un nume de sortiment de portocale (argument de intrare)
pentru a afla cantitatea disponibilă – *i.e.* răspunsul

invocare: soap – exemplu

Abordare clasică TCP/IP ☹️

socket-uri ► se specifică un port stabilit de utilizator
(*e.g.*, 7777) + o convenție de interschimb de date

invocare: soap – exemplu

Abordare RPC ☹️

apel la o procedură la distanță, executată de server
(utilizând un port al dispecerului RPC),
folosind XDR – *External Data Representation* –
ca protocol binar de (de)serializare a datelor

implementări tipice în C, C++
sau Java (cu RMI – *Remote Method Invocation*)

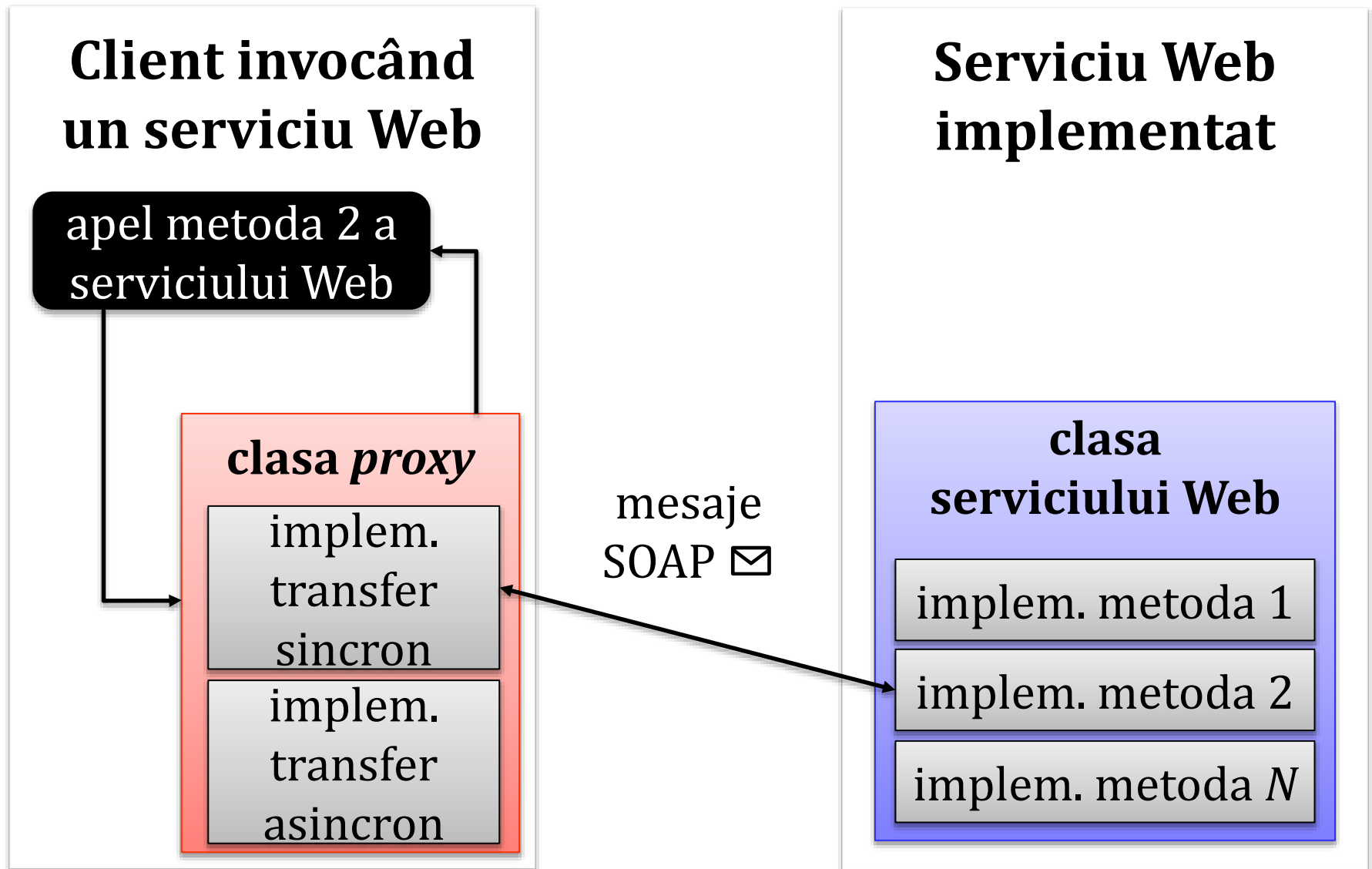
invocare: soap – exemplu

Abordare SOAP – *XML messaging*

datele vehiculate vor fi împachetate
de „plicuri” (mesaje) SOAP, transportate via HTTP

independentă de platformă și de limbaj

Invocarea unui serviciu Web – în stilul RPC




invocare: un mesaj SOAP – cerere HTTP

POST <http://web.info/porto/>

Accept: text/xml

Content-Type: text/xml



spațiu de nume
XML specific SOAP

```
<S-ENV:Envelope
  xmlns:S-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <S-ENV:Body>
    <!-- se apelează metoda de furnizare a stocului -->
    <v:getQuantity xmlns:v="http://web.info/porto/">
      <!-- date de intrare: numele sortimentului de portocale -->
      <v:arg0>blue</v:arg0>
    </v:getQuantity>
  </S-ENV:Body>
</S-ENV:Envelope>
```




parametru
de intrare

invocare: un mesaj SOAP – răspuns HTTP

HTTP/1.1 200 OK

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

```
<soap:Envelope
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <tns:getQuantityResponse
      xmlns:tns="http://web.info/porto/"
      <!-- răspunsul propriu-zis recepționat -->
      <return>74</return>
    </tns:getQuantityResponse>
  </soap:Body>
</soap:Envelope>
```



spațiu de nume XML
definit de serviciul
nostru

semnalarea erorilor – un mesaj *SOAP fault*

```
<s:Envelope
  xmlns:s="http://www.w3.org/2003/05/soap-envelope">
  <s:Body>
    <s:Fault>
      <faultcode>flickr.error.0</faultcode>
      <faultstring>Invalid SOAP envelope.</faultstring>
      <faultactor>
        http://www.flickr.com/services/soap/
      </faultactor>
      <details>
        Please see http://www.flickr.com/services/api/
        for more details
      </details>
    </s:Fault>
  </s:Body>
</s:Envelope>
```

aici, invocare eronată
a serviciului Web
SOAP oferit de Flickr

invocare: soap

SOAP \equiv RPC la nivel de Web

cerere + răspuns incluzând parametri de intrare/ieșire (+tipurile lor) specificate în XML

invocare: soap

SOAP \equiv protocol de mesagerie (serializare)

cererea conține un obiect-cerere serializat

răspunsul include un obiect-răspuns serializat

specificare

Necesitatea unui limbaj de descriere
a serviciilor Web

specificare

Necesitatea unui limbaj de descriere
a serviciilor Web

Cum poate fi găsit un serviciu Web?

Care este sintaxa mesajelor vehiculate?

Cum se desfășoară transferul de date?

specificare: **wsdl**

Web Service Description Language

recomandare a Consorțiului Web (2007)

www.w3.org/TR/wsdl20/

specificare: **wSDL**

Un serviciu Web e descris în format XML
de un document **.wSDL**

tipurile de date (argumente de intrare + răspuns oferit)
se definesc via scheme XML

specifică sintaxa, nu semantica unui serviciu

specificare: **wSDL**

Interfața serviciului (definiție abstractă)	Mesaje (<i>messages</i>)
	Operații (<i>operation</i>)
	Interfață (<i>interface</i>)
Implementarea serviciului Web (specificație concretă)	Atașare (<i>binding</i>)
	Serviciu (<i>service</i>)
	Punct terminal (<i>endpoint</i>)

specificare: **wSDL**

Interfața serviciului (definiție abstractă)	Mesaje (<i>messages</i>)
	Operații (<i>operation</i>)
	Interfață (<i>interface</i>)
Implementarea serviciului Web (specificație concretă)	Atașare (<i>binding</i>)
	Serviciu (<i>service</i>)
	Punct terminal (<i>endpoint</i>)

conceptual, se folosește un model de date reprezentat printr-un set de componente (Service, Operation, Message,...) având atașate proprietăți, inclusiv vocabularul XML (elemente + attribute) specific serviciului expus

În ce manieră
pot fi (re)găsite serviciile Web existente?

regăsire: uddi

Universal Description, Discovery, and Integration

catalog distribuit, universal, al listei de servicii Web
disponibile (înregistrate)

versiunea curentă: UDDI 3.0.2 – standard OASIS (2004)

www.oasis-open.org/committees/uddi-spec/doc/tcspecs.htm

regăsire: uddi

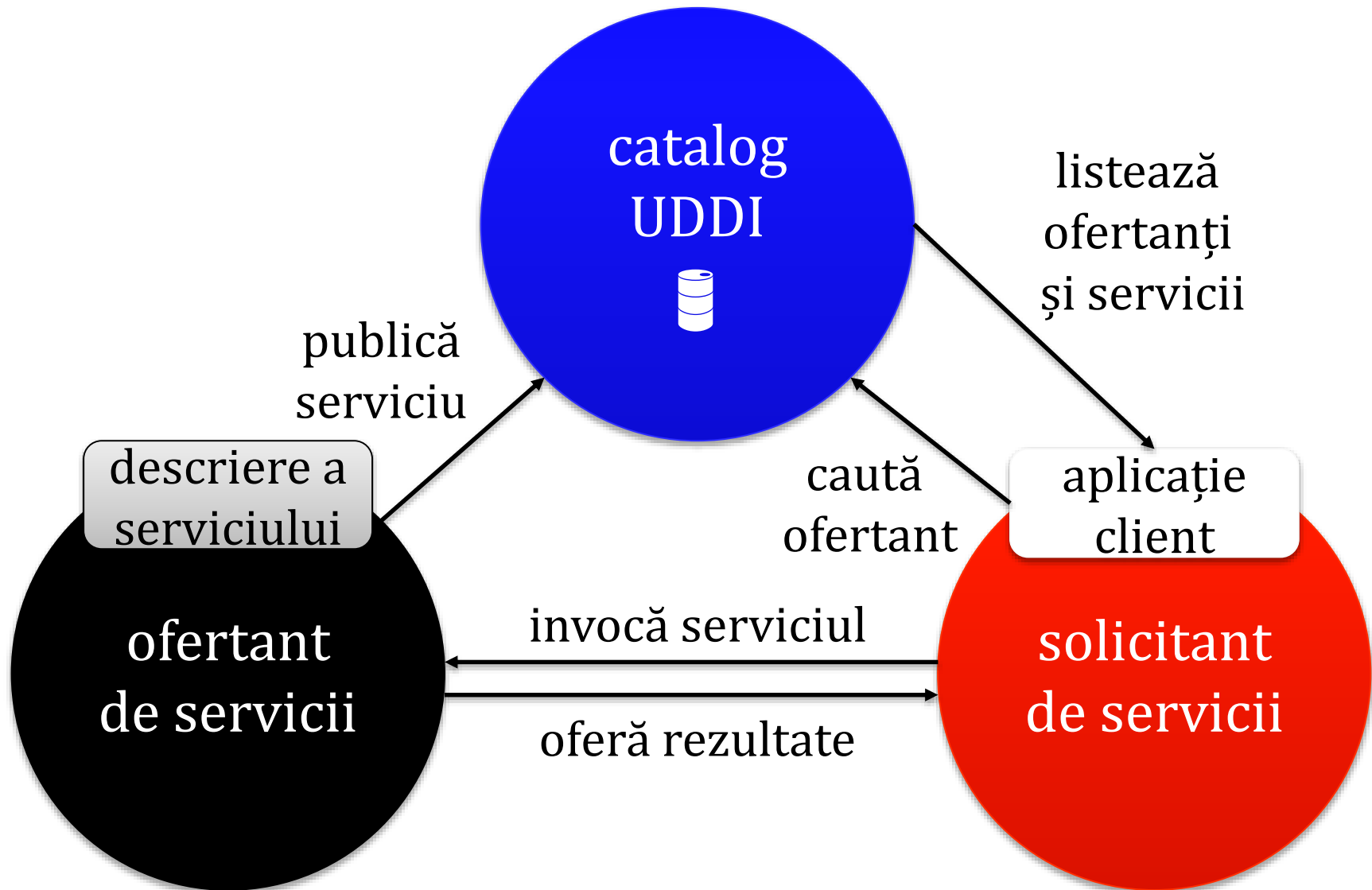
Universal Description, Discovery, and Integration

UDDI este în fapt serviciu Web, invocabil prin SOAP

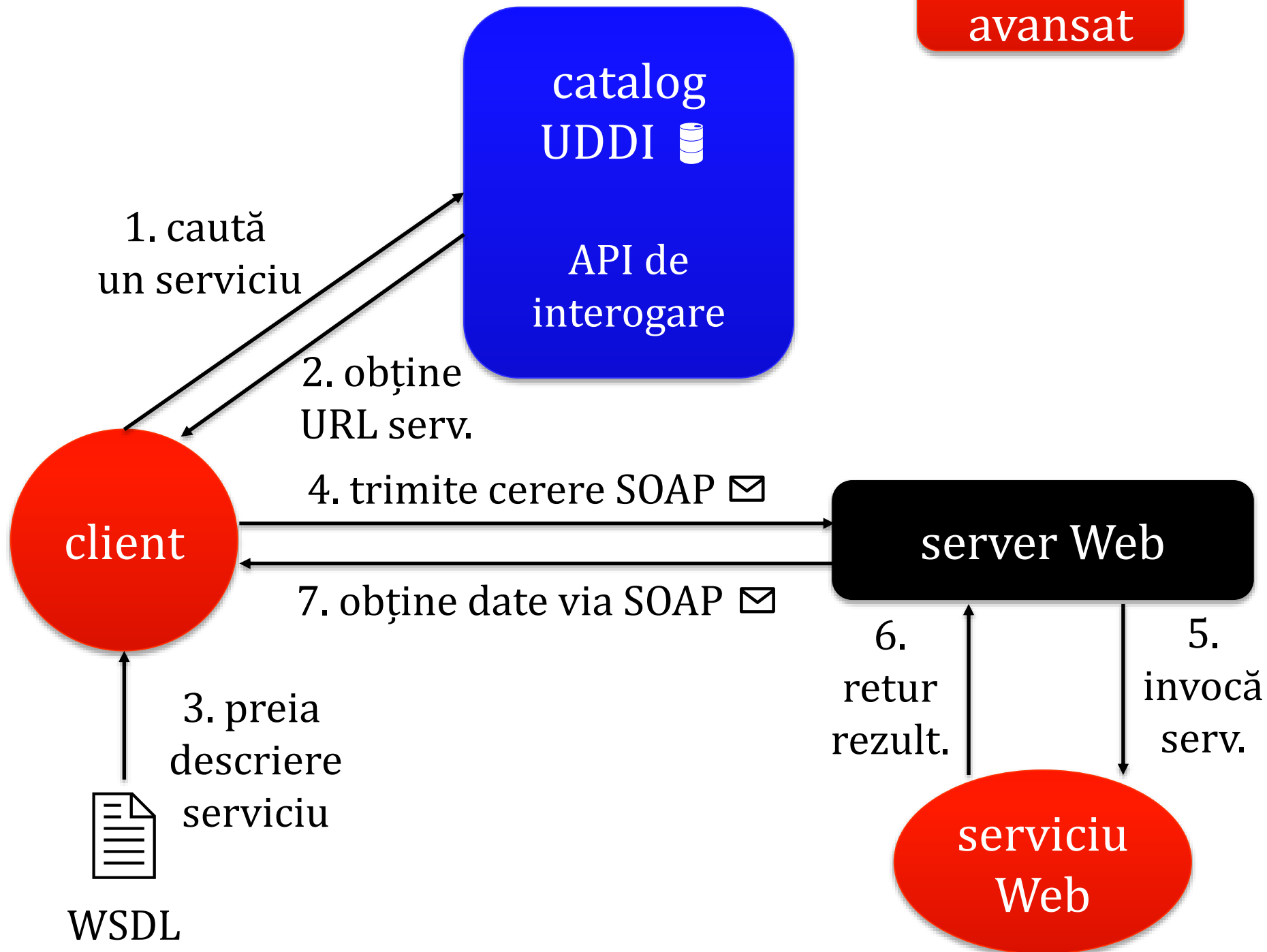
înregistrările sunt replicate

actualmente, disponibil la nivel privat (*enterprise*)

regăsire: uddi



avansat



regăsire: uddi – implementare

Instrumente software – exemple:

Web Services Tools – parte din **Eclipse Web Tools Platform**
www.eclipse.org/webtools/ws/

jUDDI (Apache)
permite gestionarea unui catalog UDDI privat
juddi.apache.org

WebSphere Application Server (IBM)
oferă facilități pentru managementul cataloagelor UDDI
www.ibm.com/cloud/websphere-application-platform

specificații & inițiative adiționale (WS-*)

Adresare: WS-Addressing

Descoperire: WS-Inspection, WS-Discovery

Mesagerie: Reliable HTTP (HTTPR),
WS Attachments, WS-Routing,...

Securitate și autorizare:
WS-Security, WS-Trust, WS-Policy,...

Suport pentru tranzacții:
WS-Coordination, WS-Transaction

specificații & inițiative adiționale (WS-*)

Interacțiunea dintre servicii Web și utilizatori:

WS for Remote Portlets (WSRP),

WS for Interactive Applications (WSIA)

Workflow-uri: Business Process Execution Language (BPEL), WS-Choreography, WS Flow Language (WSFL),...

...și multe altele

Existența serviciilor Web este suficientă?

dezvoltare

Datele și serviciile trebuie să fie accesibile
de pe fiecare dispozitiv și de oriunde

a se considera ubicuitatea Web-ului

dezvoltare

Necesitatea unei infrastructuri
orientate către servicii

o „magistrală” de comunicare între servicii/componente

dezvoltare

Noile servicii pot fi compuse din serviciile Web deja existente și accesate în mod transparent

dezvoltare

Noile servicii pot fi compuse din serviciile Web
deja existente și accesate în mod transparent

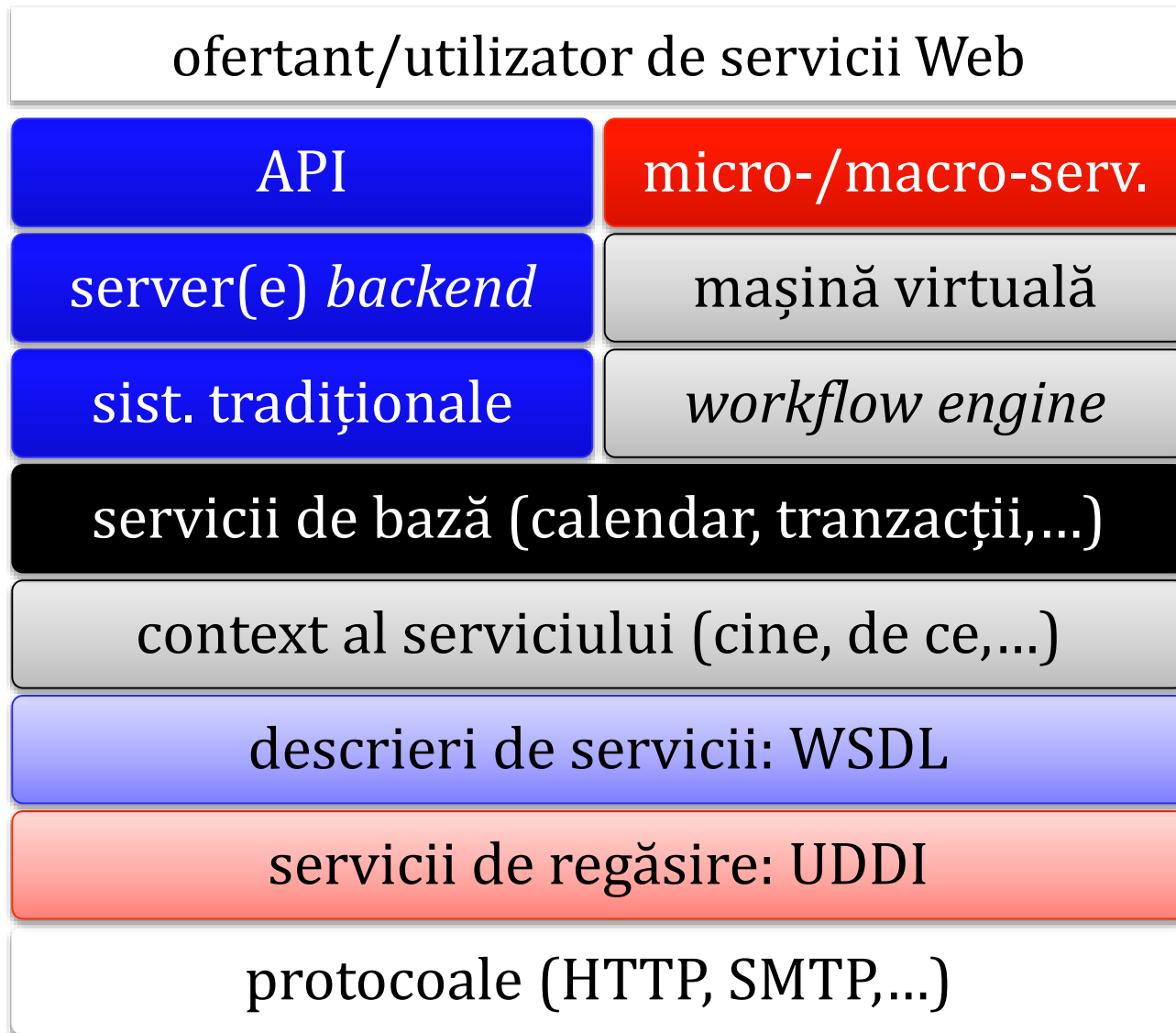
necesitatea unei platforme independente de dispozitiv,
permițând rularea serviciilor

middleware oferind funcționalități + inter-operabilitate

dezvoltare

Suport pentru conținut Web în stil „vechi”
(*e.g.*, CGI, servere de aplicații Web) + servicii Web

servere Web \equiv „porți” spre pagini și/sau servicii Web



structura conceptuală stratificată a unei platforme
bazate pe servicii Web

dezvoltare

Tehnologii, produse, aplicații – exemplificări:

Apache Axis2 (C și Java), Synapse, Tuscany (C++, Java)

EasyWSDL – generator de clase pentru Java
(inclusiv Android), Objective-C, Swift: easywsdl.com

Eclipse SOA Tools – www.eclipse.org/soa/

gsoap (pachet Go) – github.com/tiaguinho/gsoap

gSOAP Toolkit (C și C++): sourceforge.net/projects/gsoap2/

JAX-WS (Java) – specificație: jcp.org/en/jsr/detail?id=224

vezi javax.xml.ws – github.com/eclipse-ee4j/metro-jax-ws

dezvoltare

Tehnologii, produse, aplicații – exemplificări:

suport nativ în **.NET** (C# *et al.*) – vezi System.Web.Services
parte din WCF (*Windows Communication Foundation*)

PySimpleSOAP, **SUDS**, **Zeep** (implementări Python)

facilități oferite de *framework*-ul **Play!** (Java, Scala)

suport nativ în PHP – php.net/manual/en/refs.webservice.php

Red Hat JBoss Enterprise SOA Platform (Java)

Savon (bibliotecă Ruby) – savonrb.com

soap (pachet Haskell) – www.stackage.org/package/soap

dezvoltare

Tehnologii, produse, aplicații – exemplificări:

SOAPEngine (client SOAP scris în Objective-C pentru aplicații macOS, iOS, tvOS) – github.com/priore/SOAPEngine
soap, **easysoap** (Node.js) – www.npmjs.com/search?q=soap
SOAP::Lite (modul Perl) – metacpan.org/release/SOAP-Lite

la nivel de *browser* Web:

tinysoap – bibliotecă JavaScript: github.com/mhzed/tinysoap
Boomerang, **Wizdler** – extensii pentru Chrome

...și altele

dezvoltare: servicii publice via soap

Exemple diverse:

Continental Clothing, Culture24, FedEx, FitnessWizard, Who's Hurt, InsideSales, LyricWiki, Magento, OnTime, PDFReactor, Salesforce, ScrumWorks Pro, SugarCRM, TrackRoad Routing, ViaMichelin, Wikispaces, XML Soccer

Caz concret autohton:

serviciile Web oferite de Sistemul Informatic Unic Integrat al Casei Naționale de Asigurări de Sănătate (CNAS)

siui.casan.ro/cnas/siui_3.7/specificatii

dezvoltare: studiu de caz

Invocarea unui serviciu Web public
pe baza descrierii WSDL a acestuia

dezvoltare: studiu de caz

Invocarea unui serviciu Web public
pe baza descrierii WSDL a acestuia

utilizăm situl **Programmable Web**

– www.programmableweb.com –

pentru a obține lista serviciilor invocabile prin SOAP

avansat



LEARN ABOUT APIS

API NEWS

API DIRECTORY

SPECS

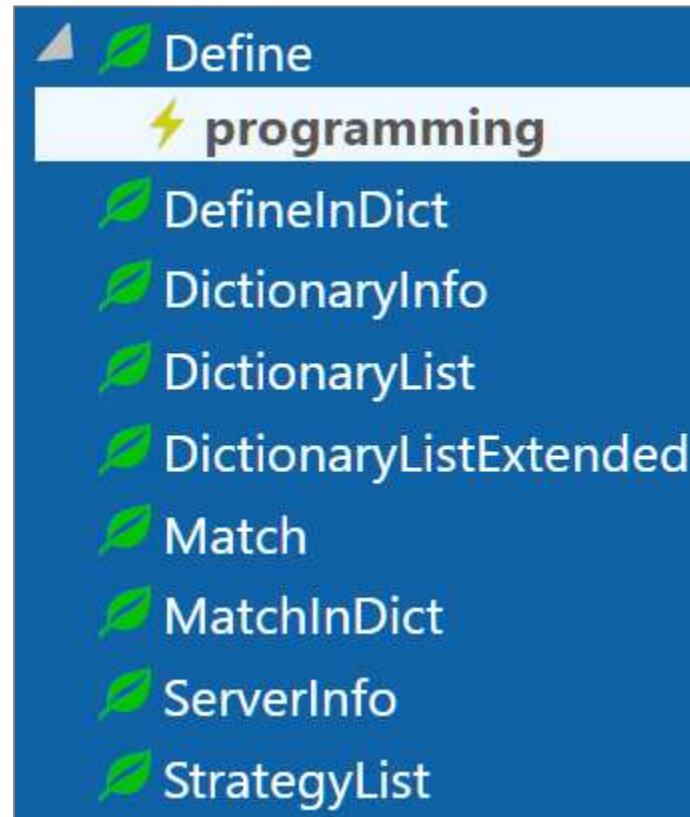
API Portal / Home Page	http://services.aonaware.com/DictService/
Primary Category	Reference
Secondary Categories	Dictionary
Support Email Address	website@aonaware.com
Is the API Design/Description Non-Proprietary ?	No
Scope	Single purpose API
Device Specific	No
Docs Home Page URL	http://services.aonaware.com/DictService/
Architectural Style	RPC
Supported Request Formats	XML, SOAP
Supported Response Formats	SOAP, XML

Aonaware Dictionary API

oferă descrierea
funcționalităților
via WSDL

acces fără
autentificare

nu necesită
cheie de utilizare
(*developer key*)
obținută
în prealabil



utilizăm **Boomerang** – extensie Chrome

pe baza URL-ului descrierii WSDL a serviciului Web,
se oferă lista operațiilor implementate:
Define DictionaryInfo DictionaryList Match etc.

Aonaware Dictionary API

folosind specificația WSDL, putem determina structura parametrilor de intrare pentru operația **Define**

```
<s:element name="Define">  
  <s:complexType>  
    <s:sequence>  
      <s:element minOccurs="0" maxOccurs="1"  
        name="word" type="s:string"/>  
    </s:sequence>  
  </s:complexType>  
</s:element>
```

definiție XML Schema

operația **Define** are un singur argument de intrare opțional de tip **string**

POST

Request Response

BODY HEADERS PARAMS AUTH SCRIPT DOCS XML No Environment

```
1 <x:Envelope xmlns:x="http://schemas.xmlsoap.org/soap/envelope/" xmlns:web
2   = "http://services.aonaware.com/webservices/">
3   <x:Header/>
4   <x:Body>
5     <web:Define>
6       <web:word>programming</web:word>
7     </web:Define>
8   </x:Body>
</x:Envelope>
```

cererea SOAP realizată via extensia **Boomerang**

```
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:s="http://services.aonaware.com/webservices/">
  <SOAP-ENV:Body>
    <!-- precizează operația (metoda) ce va fi invocată -->
    <s:Define>
      <!-- parametrul de intrare; aici, un termen (șir de caractere) -->
      <s:word>
        programming
      </s:word>
    </s:Define>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Dictionary API – invocarea operației (cerere SOAP)

Aonaware Dictionary API

pe baza WSDL, putem cunoaște tipul răspunsului furnizat de operația **Define** (executată la distanță)

```
<s:element name="DefineResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="0" maxOccurs="1"
        name="DefineResult"
        type="WordDefinition"/>
    </s:sequence>
  </s:complexType>
</s:element>
<s:complexType name="WordDefinition">
  <s:sequence>...</s:sequence>
</s:complexType>
```

```

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <DefineResponse xmlns="http://services.aonaware.com/webservices/">
      <DefineResult>
        <Word>programming</Word>
        <Definitions>
          <Definition>
            <Word>programming</Word>
            <Dictionary><Id>wn</Id><Name>WordNet</Name></Dictionary>
            <WordDefinition>program: a system...</WordDefinition>
          </Definition>
          <Definition> <!-- alte definiții --> </Definition>
        </Definitions>
      </DefineResult>
    </DefineResponse>
  </soap:Body>
</soap:Envelope>

```

Dictionary API – datele obținute (răspuns SOAP)

POST ▾

http://services.aonaware.com/DictService/DictService.asmx ▾

Send

Request

Response

BODY

HEADERS

Length: 4.2 KB Duration: 144 ms

SUCCESS 200

XML ▾

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http
  ://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
3   <soap:Body>
4     <DefineResponse xmlns="http://services.aonaware.com/webservices/">
5       <DefineResult>
6         <Word>programming</Word>
7         <Definitions>
8           <Definition>
9             <Word>programming</Word>
10            <Dictionary>
11              <Id>foldoc</Id>
12              <Name>The Free On-line Dictionary of Computing (27 SEP 03
13                )</Name>
14              </Dictionary>
15              <WordDefinition><WordDefinition>
31            </Definition>
32            <Definition><Definition>
49            <Definition><Definition>
89            <Definition><Definition>
103           <Definition><Definition>
113          </Definitions>
114        </DefineResult>
115      </DefineResponse>
116    </soap:Body>
117  </soap:Envelope>
```

răspunsul SOAP vizualizat
cu extensia **Boomerang**

dezvoltare: php

Clasa **SoapServer** deservește cereri SOAP

metode utile:

addFunction () – adaugă o funcție procesând o cerere

setClass () – specifică o clasă implementând un serviciu

handle () – „ascultă” cereri SOAP de la clienți

fault () – emite un mesaj de eroare (*SOAP fault*)

setPersistence () – stabilește persistența stării conexiunii

addSoapHeader () – include un câmp-antet în răspuns

amănunte la php.net/manual/en/book.soap.php

```
try {  
    $server = new SoapServer (null,           // nu oferim nicio descriere WSDL  
        [ 'uri' => 'http://web.info/porto' ] ); // spațiul de nume al serviciului  
  
    // adăugăm metodele (i.e. funcționalitățile, operațiile) implementate  
    $server->addFunction ('getQuantity');  
    $server->handle ();                       // așteptăm cereri SOAP  
} catch (SOAPFault $exception) {           // a apărut o excepție :(  
    die ('Ah, o problemă... ' . $exception);  
}  
  
// funcție oferind cantitatea dintr-un sortiment de portocale  
function getQuantity ($product) {  
    // uzual, vom efectua o interogare SQL, o procesare de date  
    // (CSV, JSON, XML,...), o invocare a altui serviciu Web etc.  
    switch ($product) {  
        case 'gray'    : return 33;  
        case 'blue'    : return 74;  
        default        : return 'n/a';  
    }  
}
```



vezi arhiva
cu exemple

dezvoltare: php

Clasa **SoapClient** realizează cereri SOAP către un serviciu

metode de interes:

- __soapCall ()** – trimite o cerere (apel de operație) SOAP
- __doRequest ()** – realizează o cerere (varianta PHP7)
- __getLastRequest ()** – oferă informații despre ultima cerere
- __getLastResponse ()** – oferă informații vizând răspunsul
- __setSoapHeaders ()** – stabilește anteturi (vezi **SoapHeader**)
- __setCookie ()** – stabilește un *cookie*

a se consulta php.net/manual/en/book.soap.php

dezvoltare: php

```
$client = new SoapClient (null, // nu recurgem la WSDL
[ 'location' => 'http://web.info/porto.php', // adresa serviciului Web
  'uri'       => 'http://web.info/porto',    // spațiul de nume asociat
  'trace'     => 1                          // furnizează info de depanare
]);
// realizăm o suită de apeluri ale metodei (operației) dorite
foreach ([ 'blue', 'gray', 'celestial' ] as $product) {
    $res = $client->__soapCall ('getQuantity', [ $product ]);
    echo "<p>The quantity of $product oranges
        is <strong>$res</strong>.</p>";
}
```

SOAP request:

```
<?xml version="1.0" encoding="UTF-8"?> <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ns1="http://web.info/porto" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/" SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"><SOAP-ENV:Body><ns1:getQuantity><param0 xsi:type="xsd:string">blue</param0></ns1:getQuantity></SOAP-ENV:Body></SOAP-ENV:Envelope>
```

SOAP response:

```
<?xml version="1.0" encoding="UTF-8"?> <SOAP-ENV:Envelope xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/" SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"><SOAP-ENV:Body><ns1:getQuantityResponse><return xsi:type="xsd:int">74</return></ns1:getQuantityResponse></SOAP-ENV:Body></SOAP-ENV:Envelope>
```

```
<ns1:getQuantity>
  <param0 xsi:type="xsd:string">
    blue
  </param0>
</ns1:getQuantity>
  ▼
<ns1:getQuantityResponse>
  <return xsi:type="xsd:int">74</return>
</ns1:getQuantityResponse>
```

The quantity of blue oranges is **74**.

trimiterea datelor de intrare – încapsulate ca XML în mesajul de cerere (plicul) SOAP pentru metoda **getQuantity()** oferită de serviciul Web implementat în PHP și primirea rezultatului – date XML incluse în mesajul de răspuns (plicul) SOAP

SOAP request:

```
<?xml version="1.0" encoding="UTF-8"?> <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ns1="http://web.info/porto" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/" SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"><SOAP-ENV:Body><ns1:getQuantity><param0 xsi:type="xsd:string">celestial</param0></ns1:getQuantity></SOAP-ENV:Body></SOAP-ENV:Envelope>
```

SOAP response:

```
<?xml version="1.0" encoding="UTF-8"?> <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ns1="http://web.info/porto" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/" SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"><SOAP-ENV:Body><ns1:getQuantityResponse><return xsi:type="xsd:string">n/a</return></ns1:getQuantityResponse></SOAP-ENV:Body></SOAP-ENV:Envelope>
```

The quantity of celestial oranges is n/a.

```
<ns1:getQuantity>
  <param0 xsi:type="xsd:string">
    celestial
  </param0>
</ns1:getQuantity>

▼

<ns1:getQuantityResponse>
  <return xsi:type="xsd:string">n/a</return>
</ns1:getQuantityResponse>
```


răspunsul poate fi eterogen – aici, șirul de caractere "n/a"
pentru un sortiment necunoscut de portocale

dezvoltare: php

Recurgând la **SoapClient** preluăm definiții de termeni pe baza serviciului Web **Aonaware Dictionary**

folosim descrierea WSDL
expusă de serviciul implementat în .NET

mediul PHP va genera „din zbor” un obiect corespunzător rezultatului transmis de serviciu la invocarea unei metode



detalii în arhiva
cu cod-sursă

```
// URL-ul serviciului Web apelat prin SOAP
define ('WS_URL', 'http://services.aonaware.com/DictService/DictService.asmx');
// adresa Web a documentului WSDL descriind serviciul
define ('WS_WSDL', 'http://services.aonaware.com/DictService/DictService.asmx?WSDL');
// adresa spațiului de nume al serviciului Web apelat
define ('WS_NS', 'http://services.aonaware.com/webservices/');
try {
    $client = new SoapClient(WS_WSDL, // folosim descrierea WSDL a serviciului
        [ 'location'    => WS_URL,      // adresa serviciului Web
          'uri'          => WS_NS        // spațiul de nume specificând serviciul Web
        ]
    );
    /* invocăm metoda Define() a serviciului Web pentru a obține sensul termenului dorit */
    $res = $client->Define ([ 'word' => 'knowledge' ]);

    /* preluăm prima definiție a termenului pe baza proprietăților generate automat asociate
       elementelor documentului XML încapsulat în plicul SOAP al răspunsului furnizat */
    echo "<pre><strong>Definition</strong>:\n" .
        $res->DefineResult->Definitions->Definition[0]->WordDefinition . "\n</pre>";
} catch (SOAPFault $exception) { // eroare :(
    echo 'An exception occurred: ' . $exception->faultstring;
}
```

Definition:

knowledge

The objects,
concepts and relationships that are assumed to exist in some
area of interest. A collection of {knowledge}, represented
using some {knowledge representation} language is known as a
{knowledge base} and a program for extending and/or querying a
knowledge base is a {knowledge-based system}.

Knowledge differs from {data} or {information} in that new
knowledge may be created from existing knowledge using logical
{inference}. If information is data plus meaning then
knowledge is information plus processing.

definiția termenului *knowledge*
rezultată în urma invocării serviciului Web public
pe către programul PHP – *e.g.*, un client SOAP

dezvoltare: java

Utilizarea adnotărilor pentru specificarea serviciului

```
package ro.infoiasi.portocale;
```

```
import javax.jws.WebService;  
import javax.jws.WebMethod;
```



vezi arhiva
cu exemple

@WebService

```
public class Portocale { // clasa ce implementează serviciul Web
```

```
    @WebMethod           // o metoda publică oferind stocul de portocale
```

```
    public Integer furnizeazaStoc (String sortiment) { ... }
```

```
    @WebMethod           // o altă metodă publică furnizând prețul
```

```
    public Double furnizeazaPret () {... }
```

```
}
```

dezvoltare: java

Publicarea serviciului Web

```
package ro.infoiasi.portocale.serviciu;  
  
import javax.xml.ws.Endpoint;  
import ro.infoiasi.portocale.*;  
  
public class ServiciuExpus {  
    public static void main (String[] args) {  
        try {  
            // publicăm la URL-ul specificat serviciul Web  
            Endpoint.publish ("http://localhost:8888/porto", new Portocale ());  
        } catch (Exception e) { /* a survenit o excepție... */ }  
    }  
}
```


dezvoltare: java

Accesarea (consumarea) serviciului Web de către un client

```
public class ClientDorindPortocale {  
    public static void main (String[] args) {  
        try { // instanțiem serviciul pe baza clasei proxy...  
            // generate în prealabil cu utilitarul 'wsimport'  
            PortocaleService serviciu = new PortocaleService ();  
            Portocale porto = serviciu.getPortocalePort ();  
            // apelăm metodele expuse de serviciu  
            System.out.println ("Stocul de portocale albastre are valoarea " +  
                porto.furnizeazaStoc ("albastre") * porto.furnizeazaPret ());  
        } catch (Exception e) { /* a survenit o excepție... */ }  
    }  
}
```

vezi arhiva
cu exemple

dezvoltare: direcții

Servicii Web bazate pe Java
conform modelului **ESB** (*Enterprise Service Bus*)
de experimentat Eclipse GlassFish
projects.eclipse.org/projects/ee4j.glassfish/

Servicii Web recurgând la .NET
WCF (*Windows Communication Foundation*)
docs.microsoft.com/en-us/dotnet/framework/wcf/

de studiat și L. Alboaie, S. Buraga, *Servicii Web*, Polirom, 2006:
www.slideshare.net/busaco/l-alboaie-s-buraga-servicii-web-concepte-de-baz-i-implementri-2006

...Dar testul #1?

Fie datele CSV privind numărul locurilor de muncă disponibile pe situri de profil. Ele vor fi gestionate via un serviciu Web SOAP.

WebSite,PHP,Node.js,XML ejobs.ro,333,44,91 bestjobs.eu,272,18,149 ro.indeed.com,466,172,420 indeed.com,17939,8863,17372
--

[A—L] Specificați operațiile serviciului Web capabile să realizeze căutări multicriteriale. Creați o structură XML stocând numărul de *job*-uri oferite/solicitate și media lor pentru un sit de profil și/sau o tehnologie dată.

[M—Z] Propuneți operațiile serviciului Web pentru actualizarea *job*-urilor vizând un sit de profil. Ce structură XML s-ar preta pentru a modela și oferi dinamica locurilor de muncă într-o perioadă de timp și o zonă geografică?

Fie datele CSV vizând
locurile de muncă propuse
de mai multe companii.
Ele vor fi gestionate via
un serviciu Web SOAP.

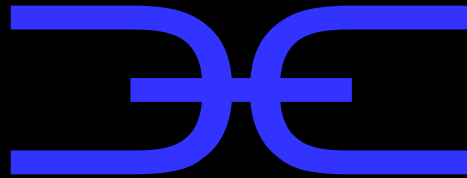
company,topic,geo-area,type
Amazon,Web,lași,AWS Dev
IBM,Node.js,Denver,Web Dev
emag,PHP,lași,Intern
IBM,XML,Hyderabad,API Dev

[A—M] Specificați operațiile serviciului Web capabile
să șteargă ofertele de *job*-uri pe baza unor criterii date.
Creați o structură XML stocând arhiva salariilor pentru
anumite categorii de locuri de muncă *per* companie/arie.

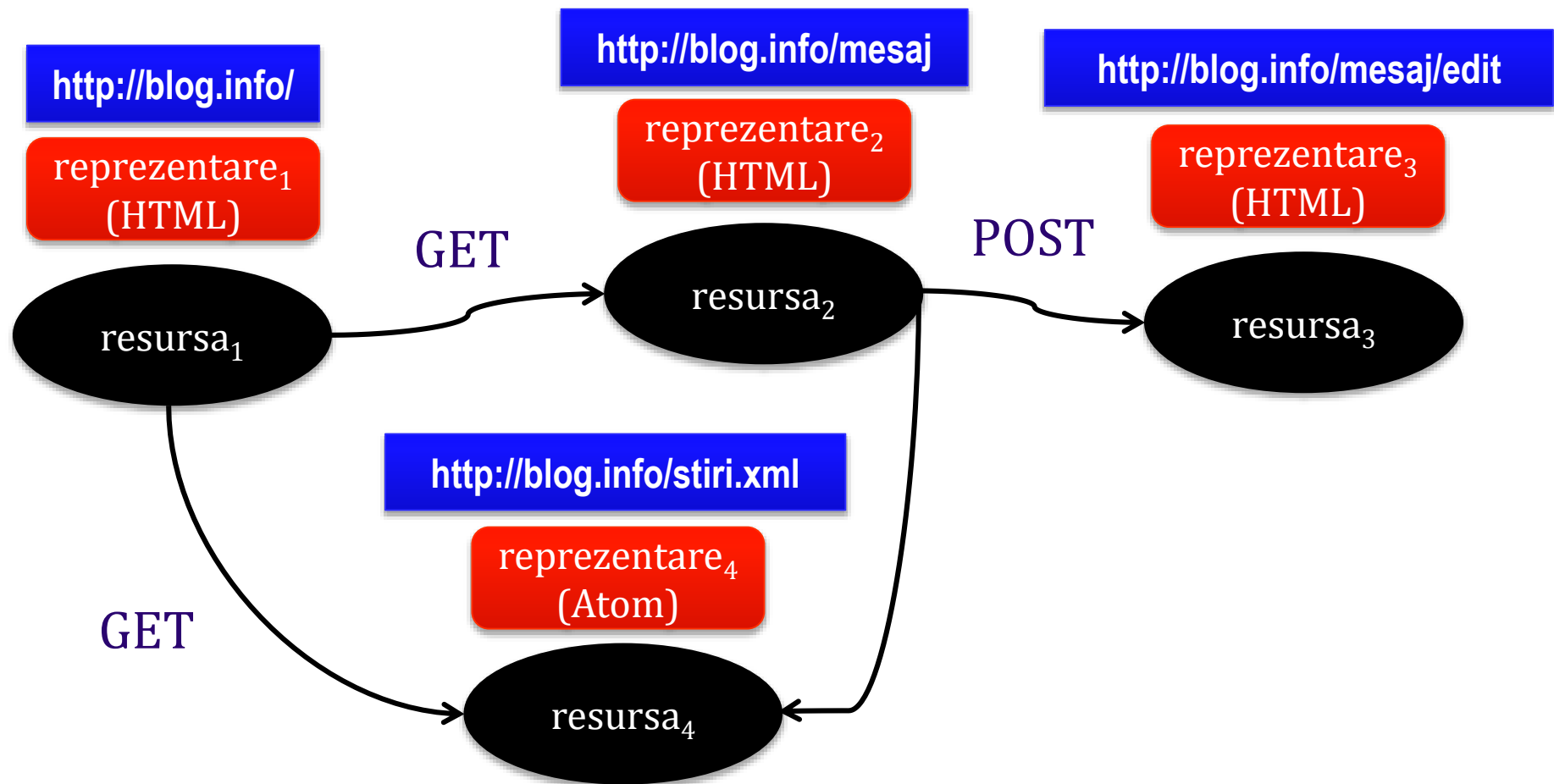
[N—Z] Propuneți operațiile serviciului Web pentru
compararea atractivității ofertelor de muncă. Furnizați
structura XML pentru a modela și expune dinamica
numărului și tipului de *job*-urilor la diverse companii.

rezumat

servicii Web



„definiții”, caracterizare, arhitectură,
SOAP, tehnologii și aplicații



episodul viitor: **servicii Web** prin REST