



# Programare Web folosind API-urile Amazon, Google și eBay

Ciprian Dobre  
[ciprian.dobre@cs.pub.ro](mailto:ciprian.dobre@cs.pub.ro)

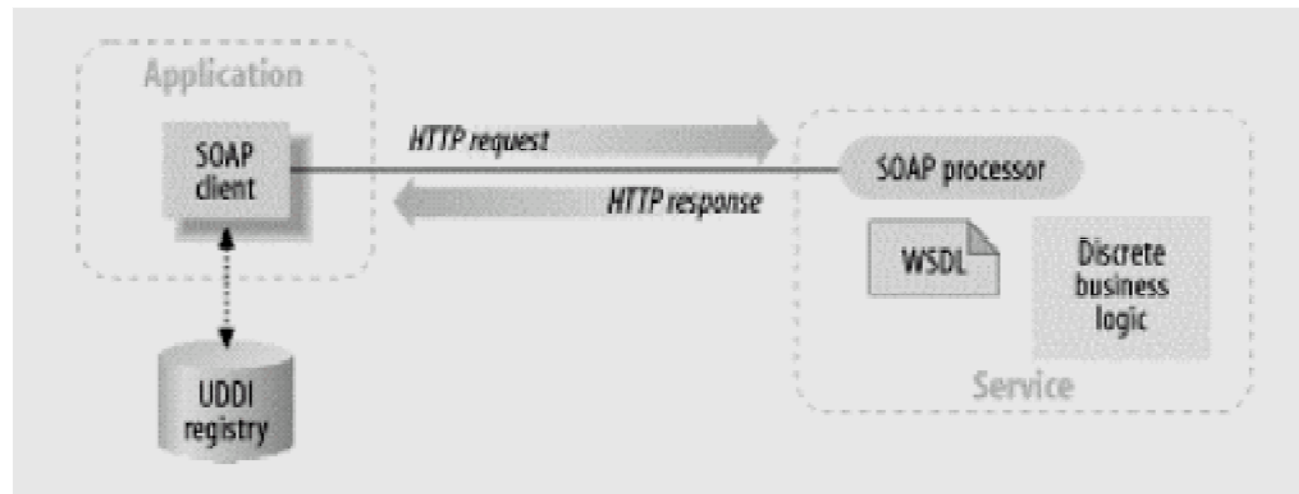


# Ce este un Serviciu Web?

- Serviciu Web:
  - Modalitate de re folosire/integrare de software
    - Indiferent unde este software-ul, pe ce platformă rezidă sau în ce limbaj a fost el scris
  - Asigură **interoperabilitatea** între aplicații posibil distribuite
  - La bază XML și protocoale Internet (HTTP, SMTP...)
- Avantaje:
  - Ușurința de integrare
  - Dezvoltarea rapidă a aplicațiilor

# Arhitectura Serviciilor Web

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





# Dar întâi ... REST

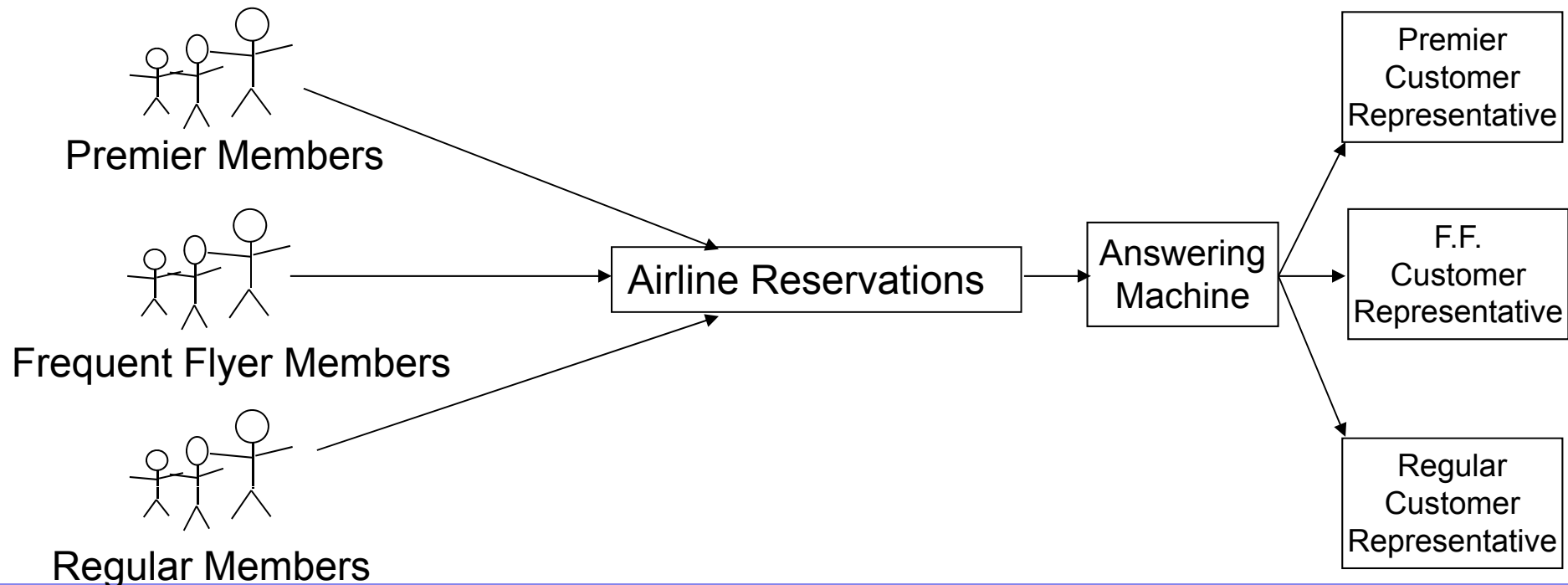
- **REST** (Representational State Transfer) este o paradigmă de dezvoltare.
- Abordare în crearea Serviciilor Web.
- Exemplu: Serviciu de rezervare zboruri
- Să presupunem că o companie aeriană dorește crearea unui serviciu pentru rezervări telefonice.
- Compania dorește ca acei clienți premium să aibă acces la serviciu imediat, clienții fideli să primească servicii cu diverse bonificații și toți ceilalți clienți să primească serviciile obișnuite.
- Există două abordări în proiectarea serviciului de rezervare...

# Abordarea 1

## "Press 1 for Premier, Press 2 for..."

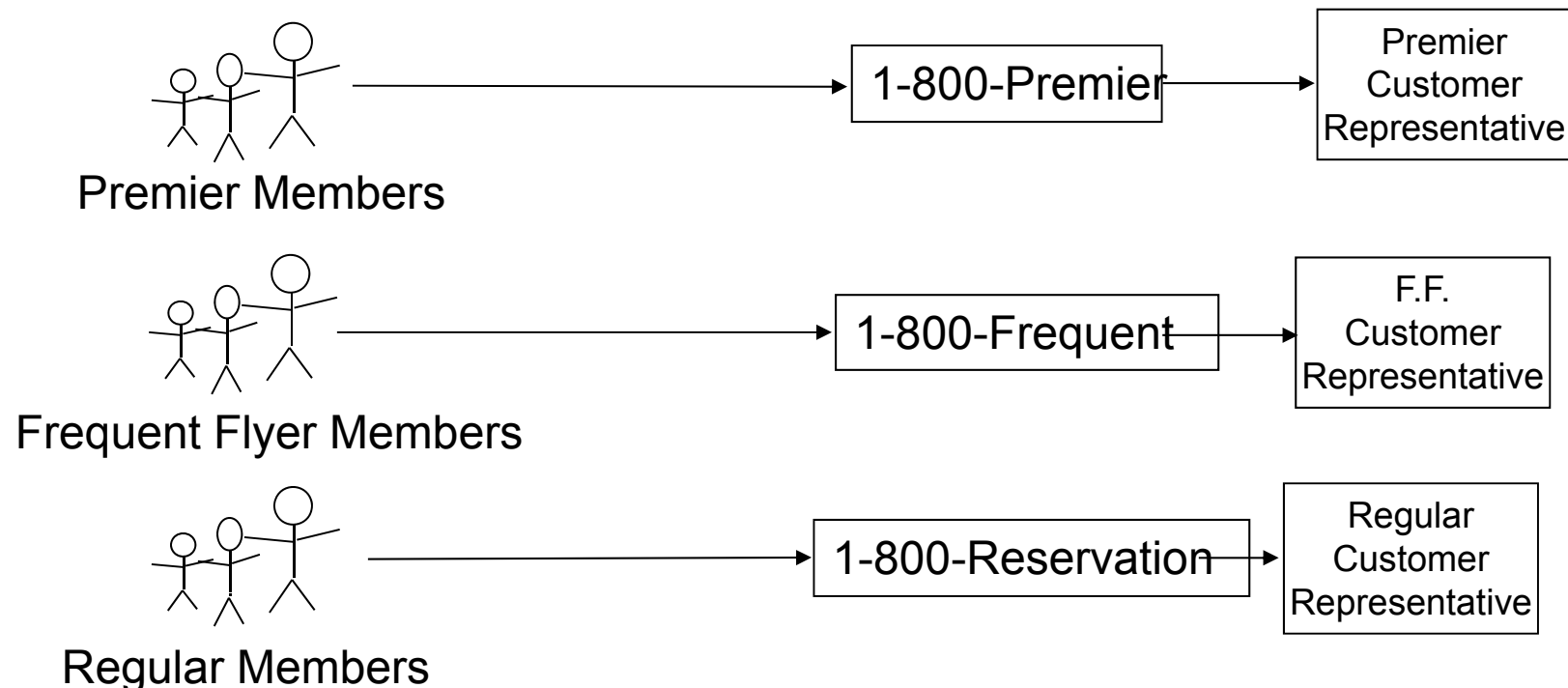
Compania poate furniza un singur număr de telefon.

După intrarea în sistem utilizatorul întâlnește un mesaj automat: *"Press 1 if you are a premier member, press 2 if you are a frequent flyer, press 3 for all others."*



# Abordarea 2 – Folosirea mai multor numere de telefon

- Compania poate furniza mai multe numere de telefon – una pentru membrii premium, alta pentru clienții fideli și alta pentru ceilalți clienți.



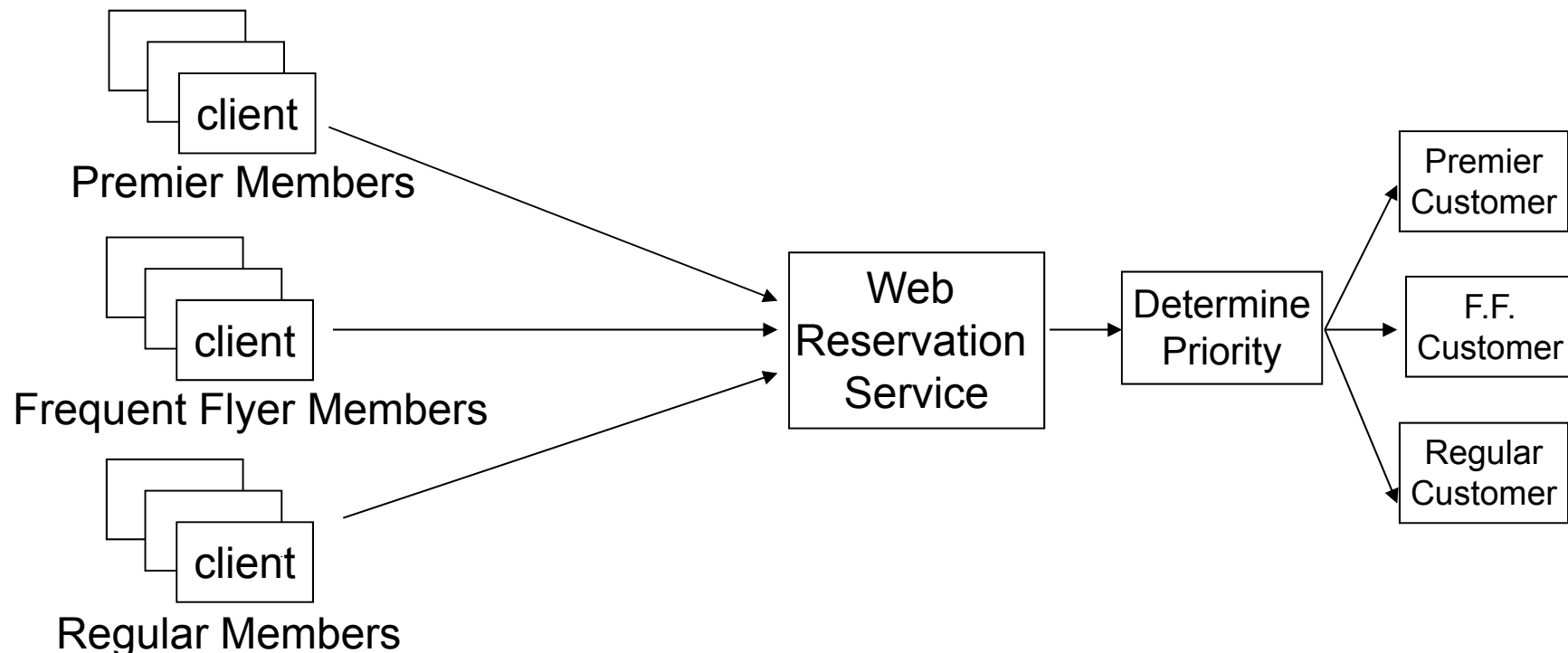


# Serviciu de rezervare online

- Să presupunem că dorim crearea unui serviciu Web pentru clienții ce doresc crearea unei rezervări online.
- Situația este oarecum similară cu serviciul telefonic...

# Soluția 1: punct unic de acces

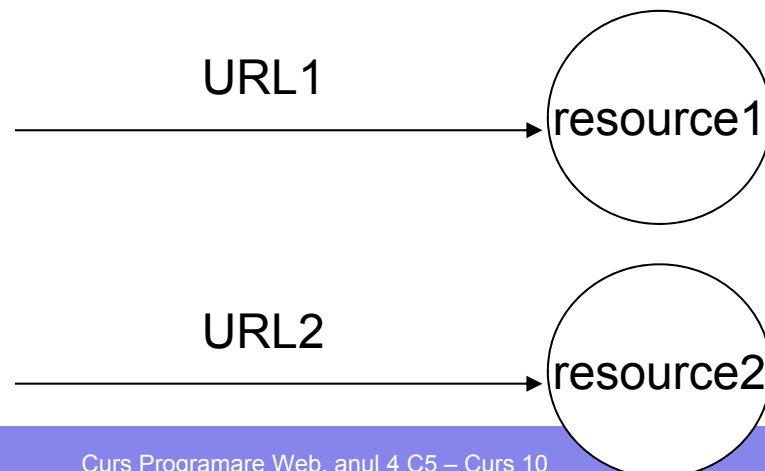
- Compania aeriană poate furniza un singur URL. Serviciul Web este responsabil cu verificarea cererii și apelarea funcției corecte.





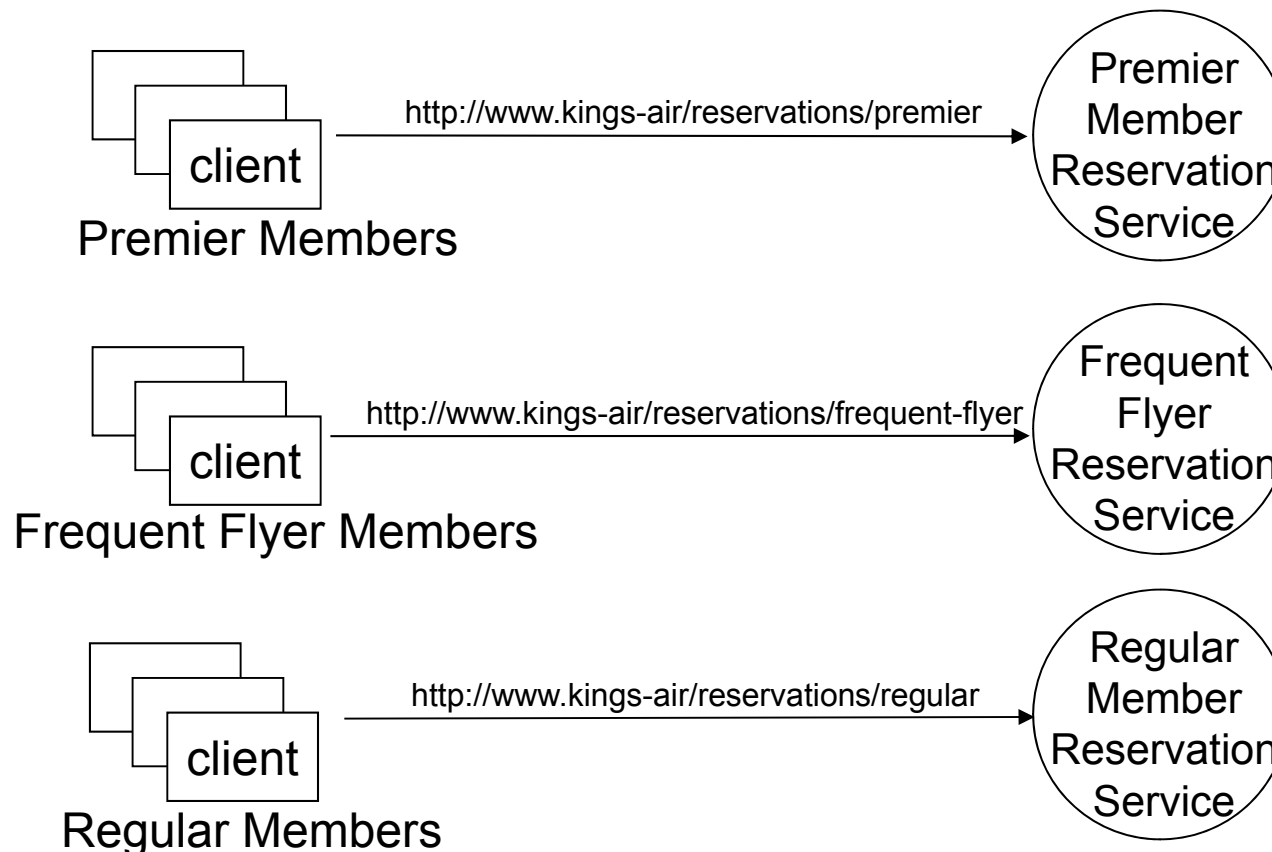
# Soluția 1: dezavantaje...

- Clienții trebuie să înțeleagă pașii și regulile impuse pentru accesarea funcției conform rolului acestora.
- Serviciul Web: punct de defect; bottleneck; greu de asigurat load balancing.
- Nu se potrivește Axiomei 0 formulată de Tim Berners-Lee cu privire la Web Design, Axiom 0.
- Axioma 0: toate resursele pe Web trebuie să fie unic identificate printr-un URI.



## Abordarea 2: Mai multe URI

- Compania aeriană poate furniza mai multe URL-uri: unul pentru clienții premium, altul pentru clienții frecvenți și altul pentru ceilalți clienți.

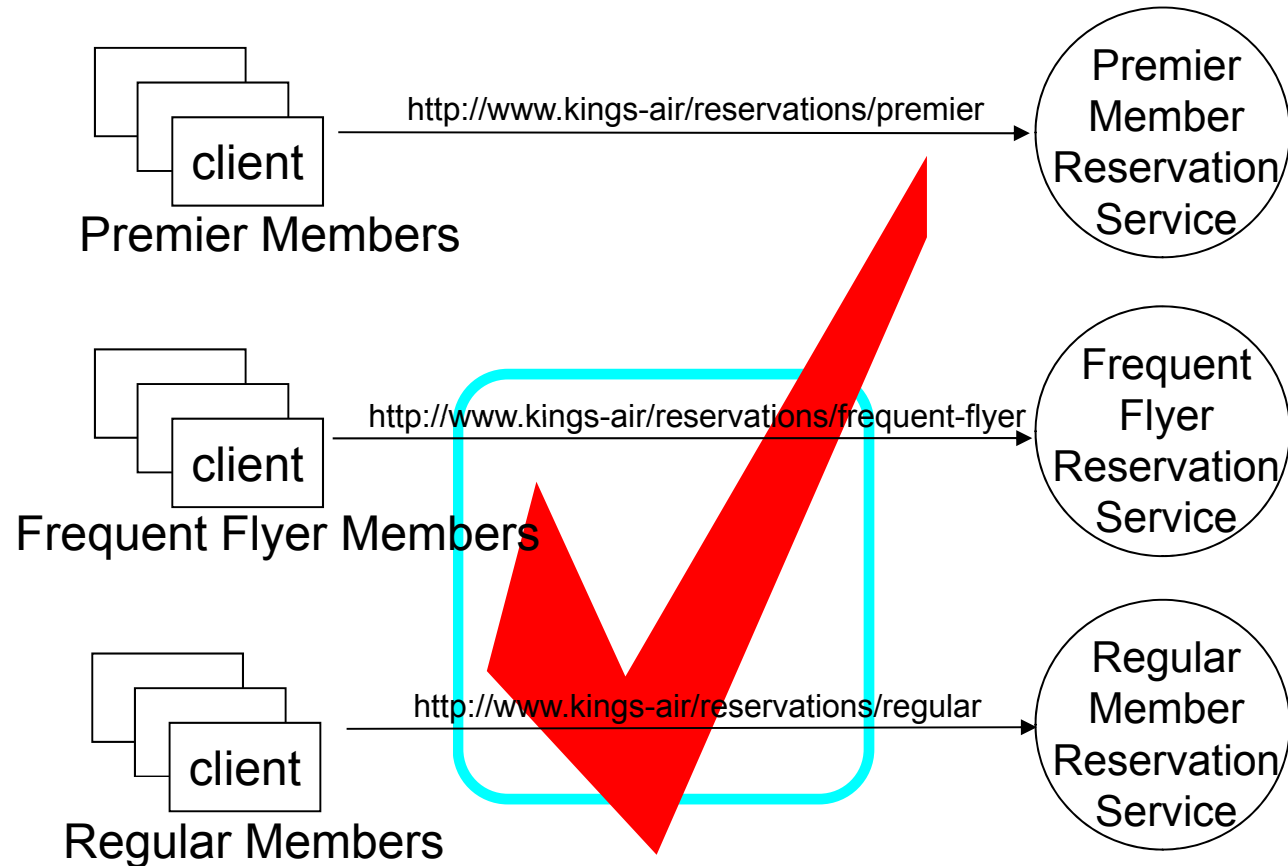




## Abordarea 2: avantaje

- Diversele URL-uri pot fi descoperite de către motoarele de căutare.
- E ușor de înțeles funcția unui serviciu chiar din numele acestuia.
- Nu e nevoie să introducem reguli de acces.  
"What you see is what you get."
- E ușor de implementat o politică de priorități – putem de exemplu asigna o mașină puternică care să răspundă cererilor pentru serviciul clienților premium.

# Aceasta este o abordare REST





# Două aspecte în modelul de dezvoltare REST

- **Resource**

Orice entitate este o resursă. O resursă poate fi un site Web, o pagină HTML, un document XML, un serviciu Web, etc.

- **URL-urile Identifică Resurse**

Orice resursă este unic identificată printr-un URL.



# REST pentru Servicii Web

- URL-urile sunt substantive
- API-ul de bază (verbele) sunt C.R.U.D
  - Create (HTTP POST)
  - Retrieve (HTTP GET)
  - Update (HTTP PUT)
  - Delete (HTTP DELETE)



# Stil REST

Substantiv: `http://www.server/database/myResultSet`

Verbe:

**Search – crează setul rezultat**

```
POST /database/myResultSet HTTP/1.1
Content-Type: text/xml; charset=utf-8
<query>title = fish</query>
```

**Sort – actualizează setul**

```
PUT /database/myResultSet HTTP/1.1
Content-Type: text/xml; charset=utf-8
<sortBy>author</sortBy>
```

**Retrieve – regăsește setul**

```
GET /database/myResultSet HTTP/1.1
Content-Type: text/xml; charset=utf-8
<recordSet><startRecord>1</startRecord><endRecord>10</endRecord></r
ecordSet>
```

**Finish – șterge setul**

```
DELETE /database/myResultSet HTTP/1.1
Content-Type: text/xml; charset=utf-8
```



# SOAP

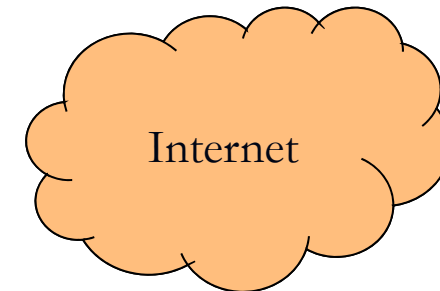
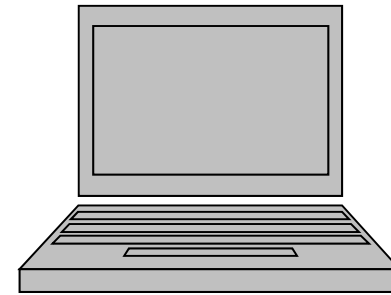
- Simple Object Access Protocol
  - Mai nou SOAP vine de la SOAP!
- Originea în tehnologia Remote Procedural Call peste XML și HTTP
- SOAP este
  - un protocol de comunicație
  - folosit pentru comunicația între aplicații
  - format pentru trimiterea de mesaje
  - proiectat pentru comunicarea în Internet
  - independent de platformă
  - independent de limbaj
  - bazat pe XML
- Inițiat de Microsoft, acum susținut de W3C
- O serie de competitori
  - XML-RPC etc...



# Stil RPC

...

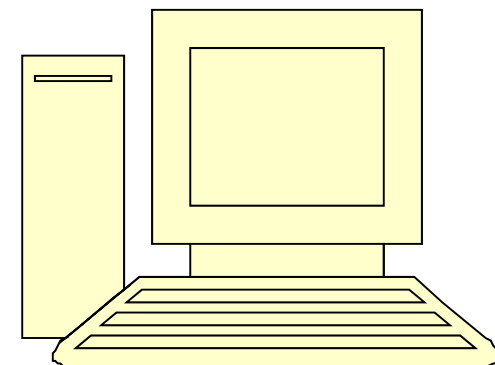
```
myVar = foo("hello world", 6)
```



```
String foo(String s, Integer i) {
```

...

```
    return result;  
}
```



# Mai mult despre SOAP...

```
<SOAP:Envelope xmlns:SOAP="..." >
  <SOAP:Header > ...
</SOAP:Header>

<SOAP:Body>

  <SRW:searchRetrieveRequest xmlns:SRW="..." >
    <SRW:version>1.1</SRW:version>
    <SRW:query>(dc.author exact "jones" and
dc.title >=
      "smith" )</SRW:query>
    <SRW:startRecord>1</SRW:startRecord>
    <SRW:maximumRecords>10</SRW:maximumRecords>

    <SRW:recordSchema>info:.../mods</SRW:recordsSc
hema>
  </SRW:searchRetrieveRequest>

</SOAP:Body>
</SOAP:Envelope>
```



# Componente SOAP

- SOAP:Header
  - Folosit pentru introducerea de informații privind mesajul și rutare
  - e.g. putem instrui un proxy să rescrie sru:baseUrl
- SOAP:Body
  - Partea de XML payload
- SOAP:Faults
  - Condiții de eroare



# WSDL 1.1

- Web Service Description Language
- Descrie mesajele SOAP.
- Format standard pentru descrierea unui serviciu web.
- Exprimat în XML, o definiție WSDL descrie modalitatea de acces la un serviciu web și ce operații sunt posibile.



# WSDL

- Componente
  - Type
    - Reprezentare XML a unui tip de date (exprimată în XML Schema)
    - e.g. searchRetrieveRequestType
  - Message
    - Mesaj XML construit din tipurile pasate între client și server și invers
    - e.g. searchRetrieveRequest, searchRetrieveResponse
  - portType
    - Ce mesaje sunt trimise în ce direcție și care este răspunsul
    - e.g. searchRetrieveRequest este trimis de la client la server, iar searchRetrieveResponse înapoi
  - Binding
    - Cum se codează mesajul (e.g. document SOAP)
    - Cum se transportă (e.g. HTTP, SMTP)
  - Service
    - e.g. <http://voyager.loc.gov:7090/>



# Problemele serviciilor Web

- WSDL nu era bine definit/descris
  - Specificațiile originale foloseau definiții prin exemplificare
- Primele instrumente WSDL erau incompatibile
  - Diverse interpretări
  - e.g. când se folosește `xsd:import`; sau `wsdl:import`
- Astăzi problemele sunt rezolvate de instrumentele de dezvoltare a serviciilor Web.



# SOAP Encoding

- Problemă: Format RPC vs. Format stil document
  - Stil RPC :  
String foo(String s, Integer i)
  - Stil document:  
Class foo {  
    String s;  
    Integer I;  
}  
String foobar(foo);



# SOAP Encoding – alegerea formatului de reprezentare

- Literal – format XML simplu

```
<stuff>hello</stuff>
<numbers">
  <item>10</item>
  <item>20</item>
  <item>30</item>
  <item>40</item>
  <item>50</item>
</numbers>
```

- SOAP – folosirea unor construcții particulare SOAP pentru array-uri etc.

```
<stuff xsi:type="xsd:string">hello</stuff>
<numbers xsi:type="SOAP-ENC:Array" SOAP-
ENC:arrayType="xsd:integer[5]">
  <item>10</item>
  <item>20</item>
  <item>30</item>
  <item>40</item>
  <item>50</item>
</numbers>
```





# WS-I

- Web Service Interoperability
  - Componente cheie: SOAP, WSDL, UDDI
- Definește un set de reguli pentru asigurarea interoperabilității serviciilor Web
  - e.g. folosirea a stilurilor SOAP fie document, fie cel literal
  - Reguli de folosire a wsdl:import vs. xsd:import
- Instrumentele produc astăzi servicii Web gata pregătite a fi interoperabile



# UDDI

- Universal Description, Discovery and Integration (a serviciilor Web :)
- Furnizează un model pentru organizarea serviciilor web
- Furnizează un WebService API pentru interacțiunea cu regiștrii de servicii web; operații precum:
  - publishing,
  - searching,
  - retrieving,
  - replicating,
  - Subscribing



# Continuarea...

- WSDL 2.0
- WS-Addressing: Mecanisme de adresare a serviciilor Web
- WS-N, WS-Events, WS-Evening: Mecanisme de tip publish/subscribe
- BPEL: Workflow/Orchestrații de servicii Web
- WSRF: Reprezentarea resurselor folosind servicii Web
- WS-Security: SAML, XACML, etc. pentru criptare, semnare/validare și delegarea autorității
- ebXML-BP, OWL-S, Semantic Framework etc. pentru negocieri de politici/contracte
  - etc...

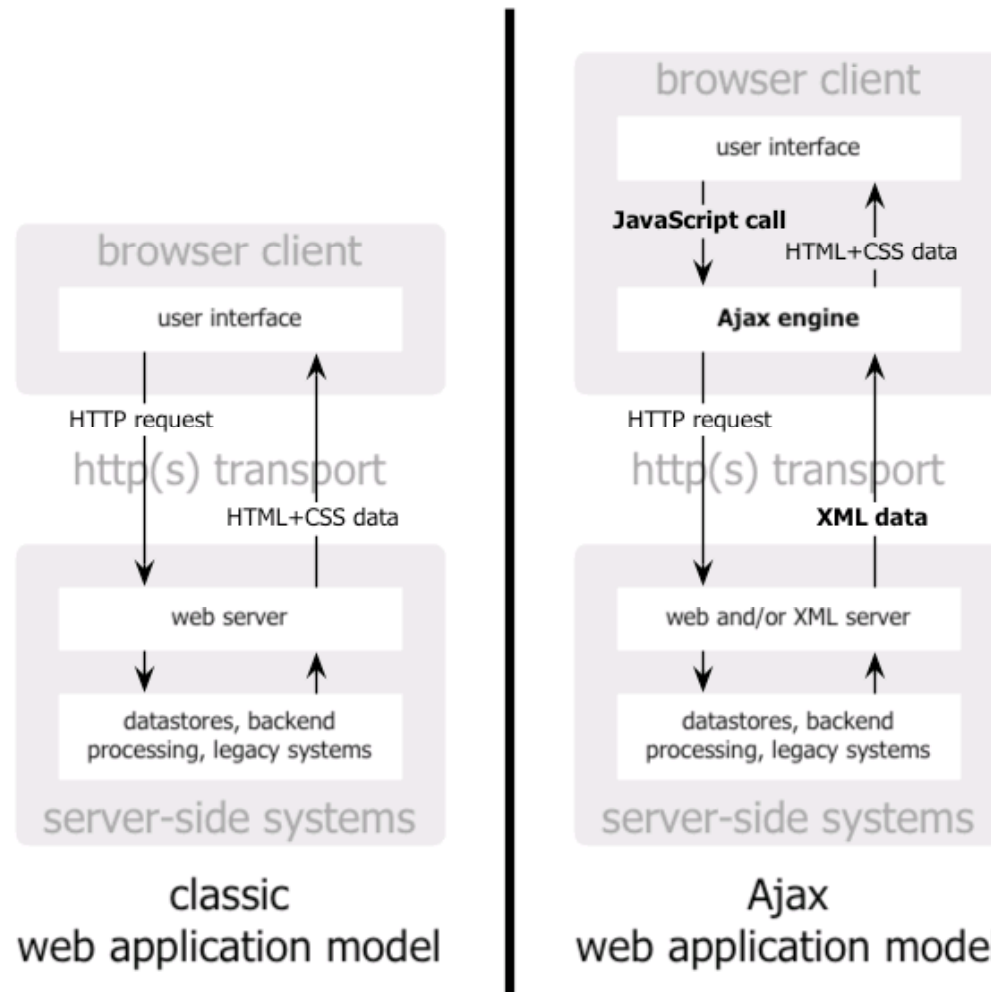


# Rich Interactivity Web - AJAX

- AJAX: Asynchronous JavaScript + XML
- AJAX încorporează:
  - Prezentare bazată pe standarde folosind *XHTML* și *CSS*;
  - Afișare și interacțiune dinamică folosind *Document Object Model*;
  - Schimb și manipulare de date folosind *XML* și *XSLT*;
  - Regăsirea asincronă a datelor folosind *XMLHttpRequest*;
  - Și *JavaScript* care leagă toate aceste lucruri împreună.
- Exemple:
  - <http://www.gmail.com>
  - <http://www.kiko.com>



# AJAX Application Model





# Amazon Web Services (AWS)

- Amazon E-Commerce Service
  - Căutare catalog, regăsire de informații despre produse, imagini și review-uri ale clienților
  - Regăsire listă de doleanțe, registru de nuntă...
  - Căutare vânzători și oferte
- Alexa Services
  - Regăsirea informațiilor precum site rank, traffic rank, thumbnail, site-uri înrudite pornind de la un URL
- Amazon Historical Pricing
  - Acces programatic la peste trei ani de date reale despre vânzări
- Amazon Simple Queue and Storage Service
  - Un manager de resurse distribuit pentru stocarea rezultatelor serviciilor web
- Amazon Elastic Compute Cloud
  - Vânzare de capacitate computațională



# Google Web APIs

- Google are o listă lungă de API-uri
  - <http://code.google.com/apis/>
- Google Search
  - AJAX Search API
  - Motor de căutare customizat folosind Google Co-op
- Google Map API
- Google Data API (GData)
  - Blogger, Google Base, Calendar, Gmail, Spreadsheets și altele
- Google Talk XMPP pentru comunicație și IM
- Google Translation
- Multe API-uri nedocumentate/nelistate ce pot fi descoperite în Google Blog



# eBay API

- Cumpărători:
  - Obținerea listei curente de categorii eBay
  - Vizualizarea informațiilor referitoare la lucrurile listate pe eBay
  - Afișarea listing-urilor eBay pe alte site-uri
  - Inserarea de feedback despre alți utilizatori la încheierea unei tranzacții comerciale
- Vândători:
  - Submiterea de lucruri pe eBay
  - Obținerea informațiilor despre licitațiile organizate
  - Regăsirea elementelor postate de un utilizator pe eBay





# Alți Provideri de Servicii/API

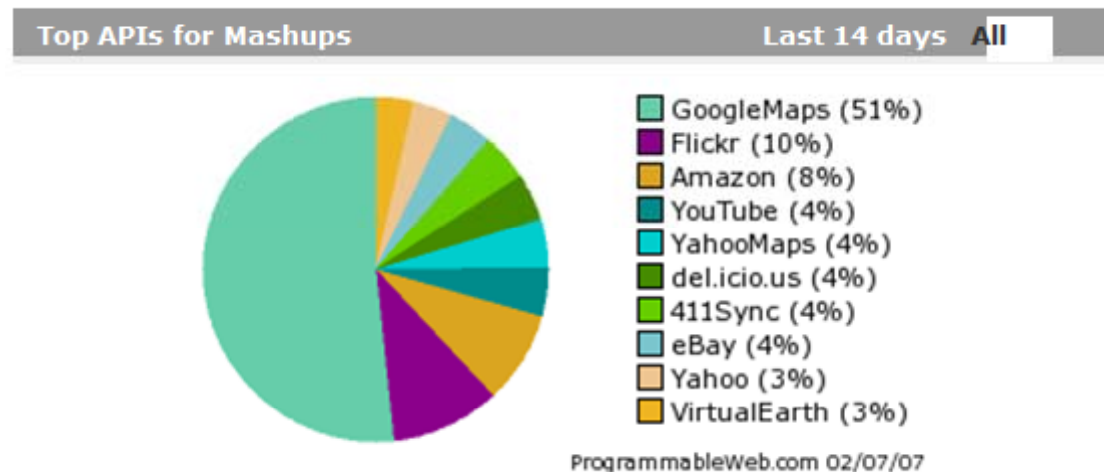
- **Yahoo!** <http://developer.yahoo.com/>
  - Căutare (web, news, video, audio, image...)
  - Flickr, del.icio.us, MyWeb, Answers API
- **Windows Live** <http://msdn2.microsoft.com/en-us/live/default.aspx>
  - Search (SOAP, REST)
  - Spaces (blog), Virtual Earth, Live ID
- **Wikipedia**
  - Downloadable database

[http://en.wikipedia.org/wiki/Wikipedia:Technical\\_FAQ#Is\\_it\\_possible\\_to\\_download\\_the\\_contents\\_of\\_Wikipedia.3F](http://en.wikipedia.org/wiki/Wikipedia:Technical_FAQ#Is_it_possible_to_download_the_contents_of_Wikipedia.3F)
- **Mai mulți la Programmableweb.com**
  - <http://www.programmableweb.com/apis>

# Mashup

## O nouă formă de Web Reuse

- “A mashup is a website or application that combines content from more than one source into an integrated experience.” – Wikipedia
- API X + API Y = mashup Z
- Business model: Advertisement





# Mashup: Weather Bonk

<http://www.weatherbonk.com/weather/index.jsp>

API-uri: Google AdWords + Google Maps + hostip.info + MS Virtual Earth + NASA + NOAA Weather Service + WeatherBug + Yahoo Geocoding + Yahoo Maps + Yahoo Traffic

**Weather Bonk - Live Weather, Forecasts, WebCams, and more on a Google Map - Mozilla Firefox**

File Edit View History Bookmarks Tools Help

http://www.weatherbonk.com/weather/index.jsp

Most Visited Customize Links Free Hotmail Windows Marketplace Windows Media Windows CIS 630 Compiler Desi...

Curs: Instrumente pentru dezvoltarea pr... Google Docs - Person - florinpop@gmail.c... "Register Amazon subscription ID" - Căut... Weather Bonk - Live Weather, Fore...

Weather Bonk | SkiBonk | GolfBonk | CampBonk

Local Weather Webcams Trip Planner Traffic Monthly Averages Power Map Add Your Webcam Quick Forecast go [english | metric]

**Weather Search**

Where:   
  
(set default) [+ faves]

**Forecast**

Provider:  [set default]

**The Weather Channel® Forecast**  
 Rimmicu Vilcea, Romania

**Tuesday May 19**  
 P. Cloudy, 10% chance of precipitation. **13.8 °C**  
 Winds 5.4 kph from E. Humidity 70%  
 Sunset: 8:50 PM

**Wednesday May 20**  
 P. Cloudy, 20% chance of precipitation. **27.2 °C**  
 Winds 4.8 kph from S. Humidity 54%  
 Sunrise: 5:48 AM  
 P. Cloudy, 20% chance of precipitation. **15.5 °C**  
 Winds 4.8 kph from N. Humidity 72%  
 Sunset: 8:51 PM

**Thursday May 21**  
 PM T-Shews, 60% chance of precipitation. **25.5 °C**  
 Winds 4.8 kph from E. Humidity 65%  
 Sunrise: 5:47 AM  
 P. Cloudy, 20% chance of precipitation. **13.8 °C**  
 Winds 1.6 kph from W. Humidity 72%  
 Sunset: 8:52 PM

**Friday May 22**  
 M Sunny, 10% chance of precipitation. **28.3 °C**  
 Winds 4.8 kph from WSW. Humidity 52%  
 Sunrise: 5:46 AM

**Live Conditions Tue 8:55 PM** [set default] [refresh]

VIEW: Map Satellite Hybrid OVERLAY: Radar Clouds Temp

Map data ©2009 Tele Atlas. Map data ©2009 Tele Atlas. Map data ©2009 Tele Atlas. Map data ©2009 Tele Atlas.

**Nearby Webcams** Add your webcam to WeatherBonk!

More local cams | Popular cams | My favorites

**Sibiu Piata Mare webcam6** 83.7 km away  
**Sibiu Piata Mare webcam3** 83.8 km away

Now: Sunny, 23 °C Wed: 27 °C Thu: 29 °C



# (Câteva) Resurse suplimentare

- Java Web Services Developer Pack
  - <http://java.sun.com/webservices/downloads/webservicespack.html>
- Consuming Web Services with NetBeans
  - <http://www.javapassion.com/planning/handsonbyol/netbeanswebservices/>



# Introducere în Google API...

# Ce este Google API?



- API-ul Google este un serviciu web ce permite dezvoltatorilor regăsirea și manipularea rapidă a informațiilor pe web.
- Google Web API-urile sunt special create pentru dezvoltatorii și cercetătorii interesați de folosirea Google ca resursă în cadrul propriilor aplicații.
- Google Web API-urile permite dezvoltatorilor interogarea a mai mult de 4 miliarde de documente web direct din cadrul propriilor programe software.
- Google folosește standardele SOAP și WSDL pentru a acționa ca o interfață între programul utilizatorului și Google API.
- Medii de programare precum Java, Perl, Visual Studio .NET sunt toate compatibile cu Google API.

\*Definiții pe baza [http:// www.google.com/apis/](http://www.google.com/apis/)

# Începutul folosirii API-ului

- Trebuie să,
  - Descărcați API Package de la <http://www.google.com/apis/>
  - Creați un cont și obțineți cheia de licențiere
  - Instalați kit-ul
  - Și să aveți Soap::Lite

Da, vom lucra în Perl :)



# Conținutul pachetului

- **googleapi.jar** – bibliotecă Java pentru accesarea serviciului API-ului Google Web.
- **GoogleAPIDemo.java** – Program exemplu ce folosește googleapi.jar.
- **Example .NET** – programe ce folosesc API-ul Google Web.
- **APIs\_Reference.html** - Document ce descrie semantica apelurilor și câmpurilor disponibile.
- **Javadoc** – Documentația exemplelor din bibliotecile exemple Java.
- **Licenses** - Licențe pentru codul Java distribuit în cadrul pachetului.
- **GoogleSearch.wsdl** – descrierea WSDL a Google SOAP API.
- **soap-samples/**





# Google API pentru Perl

## SOAP:Lite

SOAP:Lite pentru Perl reprezintă o colecție de module Perl ce furnizează o interfață la comunicația SOAP atât pe client, cât și pe server.



# Interogare Google?

```
#!/usr/local/bin/perl -w  
use SOAP::Lite;
```

```
# Configurare
```

```
$key = "Your Key Goes Here";
```

```
# Inițializare cu fișierul SOAP::Lite local
```

```
$service = SOAP::Lite
```


```
    -> service('file:GoogleSearch.wsdl');
```

```
$query= "duluth";
```



# Search Continuare...

```
$result = $service  
-> doGoogleSearch(  
    $key,                # cheia  
    $query,              # query de căutare  
    0,                   # început rezultate  
    10,                  # nr. max de rezultate  
    "false",             # filtru: boolean  
    "",                  # restrict (string)  
    "false",             # safeSearch: boolean  
    "",                  # lr (limbă)  
    "",                  # ie (input encoding)  
    "",                  # oe (output ...)  
);
```

Name	Description
key	 <p>Provided by Google, this is required for you to access the Google service. Google uses the key for authentication and logging.</p>
q	Query Phrase.
start	Zero-based index of the first desired result.
maxResults	<p>Number of results desired per query. The maximum value per query is 10.</p> <p><b>Note:</b> If you do a query that doesn't have many matches, the actual number of results you get may be smaller than what you request.</p>
filter	<p>Activates or deactivates automatic results filtering, which hides very similar results and results that all come from the same Web host.</p> <p><a href="http://diveintopython.org/soap_web_services/google.html">http://diveintopython.org/soap_web_services/google.html</a></p>



<b>restrict</b>	Restricts the search to a subset of the Google Web index, such as a country like "Ukraine" or a topic like "Linux."
<b>safeSearch</b>	A Boolean value which enables filtering of adult content in the search results.
<b>lr</b>	<i>Language Restrict</i> - Restricts the search to documents within one or more languages.
<b>ie</b>	<i>Input Encoding</i> - this parameter has been deprecated and is ignored. All requests to the APIs should be made with UTF-8 encoding.
<b>oe</b>	<i>Output Encoding</i> - this parameter has been deprecated and is ignored. All requests to the APIs should be made with UTF-8 encoding.



# Regăsirea rezultatelor căutării

```
if(defined($result->{resultElements})) {  
    print join "\n",  
    "Found:",  
    $result->{resultElements}->[0]->{title},  
    $result->{resultElements}->[0]->{URL},  
    $result->{resultElements}->[0]->{snippet} . "\n"  
}
```

```
print "\n The search took ";  
print $result->{searchTime};  
print "\n\n";  
print "The estimated Number of results for your query is: ";  
print $result->{estimatedTotalResultsCount};  
print "\n\n";
```



# Exemplu de output

Found:

University of Minnesota **Duluth** Welcomes You

<http://www.d.umn.edu/>

The University of Minnesota **Duluth** Homepage: an overview of academic programs, campus life, resources, news and events, with extensive links to other web sites **...**

The search took 0.159791

The estimated Number of results for your query is: 881000



## Sau, pentru afișarea tuturor elementelor:

```
foreach $temp (@{$result->{resultElements}}) {  
    print $temp->{snippet};  
}
```

```
foreach $temp (@{$result->{resultElements}}) {  
    print $temp->{URL};  
}
```

```
foreach $temp (@{$result->{resultElements}}) {  
    $title_array[$count++]=$temp->{title};  
}
```





# Cum folosiți o sugestie de ortografie

```
#!/usr/local/bin/perl -w
```

```
use SOAP::Lite;
```

```
# Configuration
```

```
$key = "Your Key Goes Here";
```

```
# Initialize with local SOAP::Lite file
```

```
$service = SOAP::Lite
```

```
    -> service('file:GoogleSearch.wsdl');
```

```
$correction = $service-
```

```
    >doSpellingSuggestion($key,$searchString);
```



# Cum obținem rezultatele?

- Variabila Correction va conține sugestia de ortografie, dacă Google are o astfel de sugestie de făcut, sau va fi empty
- Regăsirea rezultatelor deci se face:

```
print "\n The suggested spelling for  
$searchString is $correction \n\n";
```



# Output:

Enter a word

dulut

The suggested spelling for “Duluth” is:

duluth



# Cum puteți obține o pagină web din cache?

- Google are acest feature: pornind de la URL va încerca regăsirea paginii web din propriul “cache”.
  - Conținutul paginii poate fi vechi, relativ la momentul când crawler-ii au făcut un update
  - Exemplu pe slide-ul următor...



# Exemplu Continuat...

```
#!/usr/local/bin/perl -w  
use SOAP::Lite;
```

```
# Configuration  
$key = "Your Key Goes Here";
```

```
# Initialize with local SOAP::Lite file  
$service = SOAP::Lite  
  -> service('file:GoogleSearch.wsdl');
```

```
$url="http://www.d.umn.edu";
```

```
$cachedPage=$service->doGetCachedPage($key,$url);
```



# Regăsirea rezultatelor?

- La fel ca în cazul ortografiei,
  - Dacă pagina web există veți avea conținutul acesteia în variabila “cachedWebpage”.
  - Altfel veți primi un mesaj din partea Google care spune
  - “ This web page has not been updated...blah...blah...blah “



# Căutarea folosind alte opțiuni:

Google are patru restricții legate de topică:

Topic<restrict>	value
US. Government	unclesam
Linux	linux
Macintosh	mac
FreeBSD	bsd

<http://codeidol.com/other/google-hack/Programming-Google/Understanding-the-Google-API-Query/>



# Căutarea utilizând restricții:

```
$result = $service  
-> doGoogleSearch(  
    $key,           # key  
    $query,        # search query  
    0,             # start results  
    10,            # max results  
    "false",       # filter: boolean  
    "linux",       # restrict (string)  
    "false",       # safeSearch: boolean  
    "",            # lr  
    "",            # ie  
    "",            # oe  
);
```





# Căutarea folosind restricții de limbă

```
$result = $service
-> doGoogleSearch(
    $key,          # key
    $query,       # search query
    0,            # start results
    10,          # max results
    "false",     # filter: boolean
    "",          # restrict (string)
    "false",     # safeSearch: boolean
    "lang_de",   # lr
    "",          # ie
    ""           # oe
);
```

```
print "\n The search took ";
print $result->{searchTime};
print "\n\n";
print "The estimated Number of results for your query is: ";
print $result->{estimatedTotalResultsCount};
print "\n\n";
```

lang\_de = German

```
if(defined($result->{resultElements})) {
    print join "\n",
    "Found:",
    $result->{resultElements}->[0]->{title}, $result->{resultElements}->[0]->{URL},
    $result->{resultElements}->[0]->{snippet} . "\n"
}
```



# Output...

Please Enter Search Item  
der sturm

The search took 0.309039

The estimated Number of results for your query is: 206000

Found:

SK **STURM** GRAZ - Willkommen beim Sk **Sturm**

<http://www.sksturm.at/>

Eintreten. Puntigamer das bierige Bier, Steiermark.com, Puma,  
Tipp3, **Autohaus Jakob Prügger, Graz - Hausmannstätten.** ©  
2003 SkSturm **...**



# Interesant...

- Căutați “yahoo” și verificați numărul estimat de rezultate întoarse.
- Așteptați în jur de un minut.
- Căutați din nou “yahoo” și verificați numărul de rezultate întoarse.
- Rezultatul, 5 din 10 ori, va fi diferit...