

Estat de l'art dels serveis web

Josep Lluís Yuste Castell

jyustecastell@uoc.edu

Estudis d'Enginyeria en Informàtica

Maig del 2010

Resum: Els serveis web són aplicacions de programari que permeten l'intercanvi d'informació entre ordinadors sobre una xarxa. La informació intercanviada normalment està representada mitjançant un xml. Existeixen varis estàndards que donen aquest servei, el més conegut és SOAP, encara que els darrers anys està agafant molta força la filosofia REST. En aquest document analitzarem l'estat actual dels serveis web. L'arquitectura que més ha impulsat els serveis web els darrers anys ha estat l'arquitectura SOA, aplicant les capacitats dels serveis web per tal de millorar els negocis i processos del comerç electrònic a la xarxa. Així gràcies a aquesta nova filosofia veurem la importància que han adquirit els serveis web en els darrers anys i també quina ha estat la seva evolució. Farem una breu introducció al que són els serveis web, analitzarem quines són les recomanacions actuals del W3C en aquest camp, parlarem de quines són les principals tecnologies que actualment s'estan aplicant i quina és l'evolució prevista d'aquestes en el futur. Finalment comentarem l'experiència d'implementar alguns dels serveis web més coneguts.

1 Objectius

L'objectiu principal d'aquest article és estudiar l'estat actual dels serveis web, quins són els estàndards definits pel W3C, quina és la seva evolució i quins són els principals estàndards i arquitectures que s'apliquen avui en dia a la xarxa. Per preparar l'estudi primer veurem breument què són els serveis web i per quina raó han adquirit tanta importància avui en dia. Veurem quines són les recomanacions actuals de la W3C. Posteriorment analitzarem quines són les principals tecnologies emprades a l'actualitat per desenvolupar els serveis web, i estudiarem una tecnologia molt recent d'implementació dels serveis web anomenada RESTful. Un altre objectiu també és l'avaluació de la implementació d'alguns dels serveis web de diferents filosofies. Per acabar s'exposaran les conclusions a les que s'ha arribat després de fer aquest estudi.

2 Els serveis web

Els serveis web són APIs (*Application Programming Interface*) amb els que una entitat concreta publica una funcionalitat que pot ser emprada per un tercer. Normalment aquesta API és una comunicació bidireccional sobre HTTP (*Hypertext Transfer Protocol*). Actualment els serveis web es poden dividir en dos grans grups: els serveis web clàssics, amb comunicació basada en xml, i els serveis web basats en l'arquitectura REST, els quals es basen en la comunicació sobre el protocol de transport HTTP i la filosofia de la web en si mateixa per a dur a terme la comunicació. La informació a REST normalment està codificada en un xml però pot adoptar altres formes de representació com json.

2.1 RPC, els orígens dels serveis web

RPC (*Remote Procedure Call*) va sorgir l'any 1976, quan va ser descrita a la RFC 707. En aquell temps ja aparegué la necessitat d'invocar procediments emmagatzemats a altres màquines remotes, i així no haver de copiar en local aquests procediments si es volien emprar. Així, l'RPC consisteix en una comunicació entre màquines on una invoca una rutina o procediment que està emmagatzemada en una altra màquina, sense que el desenvolupador hagi de conèixer la naturalesa de la funció cridada, invocant-la de forma molt pareguda a si fos en local.

El funcionament del RPC és senzill. El client invoca el mètode remot, enviant un missatge al servidor, indicant el mètode invocat i els paràmetres requerits. El servidor respon al client notificant la seva rebuda de la invocació i que procedeix a processar la petició. El client queda bloquejat esperant la resposta del servidor.

La invocació remota d'un mètode és similar a una local, atorga molts avantatges, però també aporta alguns inconvenients. Els principals d'ells són el retràs inherent a la comunicació que es procedeix, i també els possibles errors de comunicació que es poden produir durant l'execució. El client ha de fer un tractament dels errors acurat per tal d'evitar quedar en estat inconsistent o que es quedi bloquejat.

3 Per què serveis web?

A partir de l'eclosió del món web aquests darrers 10 anys, aprofitant l'arquitectura existent, en quan a xarxes (LAN, VPN, etc) i protocols (tcp/ip, http, etc), les empreses i organitzacions informàtiques varen veure que oferir serveis sobre aquest nou món web que està tan estès i té tant d'èxit podia aportar moltes solucions i també generar molt de negoci. Per això, els darrers anys els serveis web han esdevingut una de les peces més importants del món informàtic en general. Aquests serveis permeten una comunicació entre sistemes de forma fàcil, senzilla i segura. Grans empreses del món web (Google, Amazon, Twitter, Yahoo, eBay, etc) han donat una importància majúscula als serveis webs, considerant-los claus per al desenvolupament de les seves empreses.

Fa uns anys, si una empresa havia de comunicar dos programes d'arquitectures diferents, per exemple, un de generació de factures i un altre de comptabilitat, havia de generar fitxers intermedis a partir del sistema original i carregar-los a mà a l'altre programa. Aquests eren els més avançats, perquè algunes empreses ho havien de repicar a mà. O si una empresa volia oferir la seva informació a una altra empresa, aquesta havia de generar fitxers o bases de dades estàtiques que l'altra havia de carregar i actualitzar a mà o amb fitxers nous periòdicament.

Anys després, varen començar a sorgir tecnologies que facilitaven la comunicació entre programes, com podia ser RMI de Java, o DCE de Microsoft. Però implicava que els sistemes havien de tenir la mateixa arquitectura i els mateixos llenguatges de programació, la qual cosa complicava molt la comunicació.

Finalment, amb l'explosió d'Internet i del desenvolupament de les xarxes de comunicacions sorgiren els serveis web tal com els entenem avui en dia, molt relacionada amb la filosofia Service Oriented Architecture que després comentarem. Aquesta nova filosofia ha fet que els serveis web adoptin noves formes de representació.

Com veurem a continuació, els serveis web aporten el següent: faciliten la comunicació, simplifiquen els processos, i donen valors afegits a les aplicacions.

3.1 Millorar la comunicació

Els serveis web faciliten de forma substancial la comunicació entre sistemes. Així, si un sistema necessita comunicar-se amb un altre perquè necessita una informació o necessita que aquest realitzi una tasca, el desenvolupador pot implementar de forma ràpida i senzilla un servei web que comuniqui els dos sistemes. El desenvolupador la majoria dels casos només s'ha de preocupar de crear els elements de programari necessaris per la comunicació: definir les transaccions, quina informació s'intercanviaran els comunicants, i el nivell de seguretat necessària. No cal que els sistemes siguin de tecnologies idèntiques, ni s'ha de preocupar del canal de comunicació (ja que s'emprarà el protocol http sobre tcp/ip).

3.2 Simplificació de processos

La interconnexió de sistemes sempre suposa la simplificació o l'eliminació d'un procés que abans era més complicat o fins i tot manual. Casos habituals són empreses que publiquen serveis web per tal de modificar i simplificar processos que suposen una pèrdua de temps en la seva cadena de valor. Les empreses poden crear serveis web per a consum propi, entre aplicacions internet, o també a altres empreses. En el cas del món empresarial els serveis web han simplificat nombrosos processos. Per exemple una empresa pot publicar un servei web

en el qual els seus proveïdors poden enviar les factures per tal de què quedin registrades en el sistema.

3.3 Reducció de costos

La simplificació de processos inherentment implica una reducció de costos. Per exemple, en el cas del món turístic, la possibilitat de fer o consultar reserves via web han ocasionat una reducció en els costos de gestió de reserves o de reducció del call center. Fent referència a l'exemple de l'enviament de factures via web, això també ocasiona una reducció de despeses en tasques de gestió de les factures (i una reducció de la possibilitat d'introduir errors).

3.4 El valor afegit a la web

Aquest és el punt que hem deixat el darrer, però no el menys important. Els serveis web han permès que hi hagi sinèrgies d'informació entre sistemes o organitzacions. Dues empreses poden arribar a acords comercials, compartint informació que aporta valor afegit a ambdues parts. També hi ha webs o sistemes que empen serveis webs públics i gratuïts per donar més valor a la seva informació, o donar un servei més ampli i millor als seus clients.

Podem trobar varis exemples clars d'aquests casos:

- Una web de reserva a hotels emprà els serveis de Google Maps per tal d'ubicar els hotels i permetre fer cerques per zones.
- Una web de venda de paquets turístics a destins variats consulta els serveis web de Trip Advisor per tal de mostrar les opinions d'altres viatgers a aquests destins.
- Una web de venda d'aparells electrònics permet la consulta dels serveis webs de Ciao per tal de cercar opinions sobre els aparells que ven i així orientar als seus clients.

4 Arquitectura SOA - Service Oriented Architecture

L'arquitectura SOA ha sorgit aquests darrers anys per donar resposta a les creixents necessitats del programari en quan a la interconnexió entre sistemes. Amb la creixent globalització, la interconnexió d'organitzacions i empreses, i la ràpida evolució de la tecnologia i del propi negoci, és necessari que el programari segueixi una sèrie de principis que el facilitin la seva adaptació al nou món.

Així, l'Arquitectura Orientada a Serveis (Service Oriented Architecture, SOA) és una arquitectura de fer programari que segueix una sèrie de principis en les fases de desenvolupament i integració per tal d'oferir uns serveis fàcilment integrables i feblement acoblats que poden ser usats en múltiples casos d'ús i de negoci.

Els serveis web són una part important de SOA. Són una de les eines bàsiques i possiblement la més emprada de creació d'interfícies SOA. Però també ho pot ser una web, o una api de qualsevol tipus (ajax, html, o xml). Els desenvolupadors que fa 10 anys desenvolupaven serveis RPC o RMI ja estaven aplicant els principis de l'arquitectura SOA, encara que aquesta encara no havia aparegut en escena.

SOA defineix com integrar aplicacions profundament dispars en un món basat en el web i que emprà múltiples plataformes de desenvolupament. Però SOA no defineix una API, només defineix la interfície d'un servei en termes de protocols i funcionalitats.

Els desenvolupadors de programari SOA han de pensar contínuament en proveir serveis per a altres sistemes de capes middleware o directament facilitar el programari per tal d'implementar els processos de negoci.

L'arquitectura SOA ens proporciona les següents beneficis:

- Millora en els temps de realització de canvis en els processos.
- Facilitat per evolucionar models de negoci basats en tercers.
- Facilitat per abordar models de negoci basats en col·laboració amb tercers.

- Poder reemplaçar elements de la capa aplicativa SOA sense interrompre el procés de negoci.
- Facilitat per comunicar i integrar tecnologies i sistemes distints.

5 Estats dels serveis web actuals segons la W3C

A la web del W3C [19] podem consultar les especificacions dels estàndards definits per la W3C. Aquesta organització és un consorci internacional creat el 1994 per Tim Berners Lee que produeix recomanacions per a la WWW, recomanacions que són considerades estàndards de la xarxa.

En el camp de serveis web, el W3C agrupa les recomanacions en 5 grans àrees: dades, protocols, descriptors del servei, seguretat i internacionalització. A continuació veurem les recomanacions actuals de cada una d'aquestes àrees.

5.1 Dades

En aquesta àrea es troben les especificacions sobre diferents formes de representar la informació intercanviada entre els comunicants del servei web. Per a facilitar aquest intercanvi el W3C defineix una sèrie de vocabularis i regles de representació de la informació.

Les especificacions actuals són les següents:

- **XML**: Extensible Markup Language és un llenguatge de marques que segueixen unes regles per a codificar documents electrònicament. Actualment el W3C defineix les especificacions 1.0.
- **XML Schema**: és un llenguatge d'esquema per a dotar d'una abstracció major a un xml, descrivint-ne una l'estructura i restriccions dels continguts dels documents d'una forma molt precisa, més enllà de les normes sintàctiques del propi XML. Forma part de les recomanacions del W3C des del 2001. És descrit per aquest organisme com el successor dels DTDs, essent la seva potència molt major.
- **XML-binary Optimized Packaging (XOP)**: és un mecanisme per serialitzar documents XML que contenen dades binàries, i també per a retornar un element XOP a l'XML original. Un exemple seria un xml amb la informació en binari d'una imatge. El XOP permet serialitzar aquest XML.
- **RDF (Resource Description Framework)**: és un bastiment per a representar meta dades a la web desenvolupat pel mateix W3C. A l'especificació es defineix la seva sintaxi, els seus objectius, conceptes claus, tipat de dades i normalització de caràcters. Està basat en la idea de convertir les declaracions dels recursos en expressions amb la forma subjecte-predicat-objecte. El subjecte és el recurs (allò que es descriu), el predicat és la propietat o relació del recurs, i l'objecte és el valor o altre recurs amb el que s'estableix la relació. La combinació de RDF amb altres eines com RDF Schema i OWL permet afegir significat a les pàgines, i és una de les tecnologies essencials de la web semàntica.
- **GRDDL (Gleaning Resource Description from Dialects of Languages)**: és un mecanisme per obtenir informació en RDF d'una pàgina. Mitjançant això, una aplicació informàtica pot extreure de forma automàtica informació de pàgines web estructurades per integrar-la a la web semàntica. Ha arribat a recomanació de la W3C el setembre del 2007.

A més, la W3C ha creat grups d'estudi que treballen sobre els següents camps relacionats amb la informació: web semàntica (fòrum per donar suport a desenvolupadors i usuaris de tecnologies de la web semàntica com RDF, OWL, o SPARQL), arquitectura de la WWW, nucli de l'XML, i finalment un grup d'estudi de l'XML SCHEMA.

5.2 Protocols

Sobre els protocols de comunicació a usar pels serveis web, la W3C ofereix les següents recomanacions:

- **HTTP:** actualment en vigència la versió 1.1, és el protocol usat a cada transacció de la web. La seva simplicitat, adaptabilitat i escalabilitat ha estat fonamental pel creixement de la WWW. Fou desenvolupat per la W3C (pel seu fundador Tim Berners Lee) i culminà l'any 1999 amb la publicació d'una sèrie de RFCs, essent el més important el RFC 2616 que especifica la versió 1.1. HTTP és un protocol orientat a transaccions que segueix l'esquema petició-resposta entre un client i un servidor. Defineix la sintaxi i la semàntica que empen els elements de programari de l'arquitectura web per a comunicar-se.
- **SOAP:** posteriorment explicarem més detalladament en què consisteix el protocol SOAP donat que és un dels protocols més importants dels serveis web i que va crear una arquitectura de serveis web pròpia, però a mode de resum podem dir que és un protocol estàndard que defineix com dos elements de programari es poden comunicar intercanviant missatges XML. SOAP fou creat per Microsoft i IBM, entre d'altres, i des de l'any 2000 la W3C defineix les seves especificacions. Actualment es disposa de la versió 1.2 (any 2007).

A més, encara que no siguin protocols concrets, el W3C també ofereix recomanacions sobre l'arquitectura a seguir:

- **Direccionament de serveis web:** aquesta especificació proveeix d'un mecanisme pel qual es poden identificar serveis web i missatges independentment del protocol de transport usat. Defineix un espai de noms que s'empra per a identificar serveis web. Gràcies a aquesta especificació és possible fer que les peticions a serveis web puguin ser transmeses a través de xarxes compostes per nodes que realitzen algun processament sobre el missatge, com tallafocs o passarel·les, de forma que la informació de transport sigui independent del protocol emprat per la transmissió del missatge. La versió actual és la 1.0.
- **Arquitectura de serveis web:** el W3C exposa una sèrie de documents (no estàndards) d'orientació sobre arquitectura de serveis web, com poden ser requeriments i recomanacions per la internacionalització de serveis web, explicació del cicle de vida d'un servei web i del processament d'una petició a un servei web, arquitectura de serveis web (indicant components funcionals i definit les relacions entre els components), etc.

5.3 Descriptors de servei web

La descripció del servei és una part important de l'accés al servei. Molts dels llenguatges de descripció són consumits per programari, creant automàticament esquelets de codi per a facilitar el seu ús. Però també poden ser interpretats per persones, la qual cosa ajuda a l'hora d'aclarir l'ús de paràmetres o la semàntica de les possibles operacions a realitzar.

Les principals especificacions que el W3C ofereix en aquesta àrea són les següents:

- **WSDL (Web Service Description Language):** és un format XML que serveix per descriure serveis web. Actualment està disponible la versió 2.0. El WSDL descriu la interfície pública als serveis web. Està basant en XML i descriu el format dels missatges necessaris per interactuar amb els serveis disponibles. Posteriorment veurem amb més detall que és el WSDL.
- **XML Schema:** com hem comentat al punt anterior on parlàvem de les especificacions de les dades, l'XML Schema és un format molt important de definició de dades i també ens serveix per especificar el format dels missatges intercanviats a la comunicació.
- **SML (Service Modelling Language):** SML i SML-IF (Service Modeling Language Interchange Format) són unes especificacions basades en XML creades per definir relacions entre objectes. Aquestes especificacions són un conjunt d'extensions d'XML Schema que defineixen les relacions entre aquests elements. Foren publicats pel W3C l'any 2009.

El W3C disposa a més d'altres especificacions sobre aquesta àrea:

- Web Service Choreography Definition Language (WS-CDL): actualment a la versió 1.0, és un llenguatge basat en XML que serveix per definir el comportament d'un conjunt d'elements de programari que treballen amb un comportament similar i de forma conjunta per assolir uns objectius comuns.
- Web Service Policy: (actualment a la versió 1.5) proposa un model de propòsit general i la seva sintaxi de definició d'elements a un sistema de serveis web.
- Semantic Annotations for WSDL and XML Schema: és un estàndard de l'any 2007 que consisteix en un conjunt d'extensions pel WSDL i l'XML SCHEMA per tal d'afegir a aquests llenguatges la capacitat de descriure elements semàntics.

5.4 Seguretat

La seguretat és molt important als serveis web. La comunicació entre programari requereix sovint unes condicions d'integritat, autenticitat i privacitat. La signatura d'XML, encriptació i XKMS ens poden assegurar aquestes característiques.

Aquestes són les especificacions que ofereix el W3C:

- **Signatura d'XML:** aquesta especificació defineix les regles i la sintaxi de la signatura digital d'un XML. Aquesta proveeix integritat i autenticació del missatge. Està molt relacionat amb el punt que a continuació veurem, l'encriptació d'xml.
- **Encriptació d'XML:** és una recomanació del W3C que especifica un procés per a xifrar dades (no únicament documents XML) i representar aquesta informació xifrada a la seva vegada en XML per a que viatgi pels mitjans de transmissió. Aquest any 2010 ha sortit la versió 1.1 de la recomanació sobre sintaxi i processat d'encriptació XML.
- **XKMS (Key Management Specifications):** defineix una arquitectura per a facilitar l'ús de les claus públiques a una comunicació de serveis web entre aplicacions. És un protocol desenvolupat íntegrament pel W3C i descriu la distribució i registre de les claus públiques als serveis web. Aquests poden accedir a un servidor XKMS per rebre informació actualitzada de les claus per encriptar i autenticar missatges. Actualment està a la versió 2.0 (publicada l'any 2005).

5.5 Internacionalització

Donat l'abast mundial de la web i la recent globalització de qualsevol servei, és evident la importància del factor d'internacionalització de qualsevol web o servei web disponible a la xarxa. L'accés a la web per a tothom ha estat un dels objectius principals del W3C des del principi. El problema és la varietat de cultures, llenguatges i sistemes d'escriptura que hi ha arreu del món. El W3C disposa de grups de dissenyadors i desenvolupadors que treballen en aquesta àrea, participant a tots els altres grups de treball aportant la seva visió per tal de què les especificacions que es vagin creant siguin de l'abast de tothom i no tinguin barreres culturals. Actualment es disposa de diverses recomanacions sobre internacionalització, i són aplicables a molts de camps de la xarxa, tant a serveis web com a la web en si mateixa.

Centrant-nos en serveis web, el W3C disposa de notes (no estàndards) sobre internacionalització, com són les IRIs esteses per identificació d'XML, recomanacions per treballar amb zones horàries, o consells i patrons per usar a escenaris on tenim requeriments d'internacionalització.

També existeixen uns esboços que pot ser pròximament es converteixin en estàndards: la internacionalització de serveis web (WS-I18N) que descriu ampliacions als missatges SOAP per a proveir una internacionalització major a les operacions, o també Tags de Llenguatge i Identificadors Locals per la WWW, que descriu mecanismes per identificar o seleccionar l'idioma del contingut o la configuració local necessària per a processar la informació a l'ordinador client.

6 Serveis web basats en XML

6.1 SOAP

Aquesta tipologia de serveis web són els més estesos actualment. La seva arquitectura deriva dels RPC (Remote Procedure Call). De fet, el SOAP prové d'un protocol anomenat XML-RPC. La seva aparició fou una revolució en el món dels serveis web, i fou adoptat ràpidament com un estàndard comú per les principals empreses de programari. En el seu moment, en el mercat hi havia nombroses tecnologies d'intercanvi de transaccions, cada una impulsada per una companyia concreta o una comunitat de desenvolupadors. Així, teníem DCE de Microsoft, RMI en Java, RPC de Sun, o ORPC de Corba. SOAP posà d'acord a tots aquests grups de persones i s'adoptà ràpidament com un estàndard de facto com a protocol per a l'intercanvi d'informació. A continuació veurem una llista d'avantatges de SOAP que motivaren aquest canvi de filosofia.

6.2 Funcionament de SOAP

El servidor accepta peticions d'informació dels clients, i retorna un missatge en un format similar. L'acció realitzada i la informació enviada resideixen dins el missatge. Aquest missatge és un XML (Extensible Markup Language). Recordem que l'XML és una forma d'estructurar informació mitjançant un llenguatge d'etiquetes. Aquesta comunicació es fa normalment sobre http, així, el missatge XML emprà el protocol http per ser enviat. Com veurem més tard, aquest és un factor important per veure les diferències entre serveis web xml i Restful.

El format dels missatges intercanviats entre client i servidor més estès és el SOAP (Simple Object Access Protocol). Aquest protocol fou creat per IBM i Microsoft, entre d'altres, i deriva d'un altre protocol anomenat XML-RPC, creat el 1998. Actualment SOAP està controlat per el W3C.

Exemple de missatge SOAP:

```
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>
```

Fixem-nos que la primera part del missatge és l'encapçalament de HTTP, no és propi de SOAP. Així, aquest missatge té dos embolcalls, el de HTTP i el de SOAP.

6.3 WSDL

WSDL són les sigles de Web Service Description Language, i com el seu nom indica, és un format XML que serveix per descriure serveis web. Està controlat per la W3C i actualment es disposa de la versió 2.0.

WSDL descriu la interfície pública d'un servei web. Està basat en XML i descriu la forma de comunicació: protocol i formats dels missatges que participen a la comunicació, així com un catàleg del serveis proveïts. WSDL s'empra amb combinació amb SOAP. Així un servei web SOAP ofereix la seva descripció amb WSDL. Això és molt important per el desenvolupador perquè pot conèixer el serveis que pot consumir i els formats dels mateixos. Una vegada vist el missatge, el desenvolupador implementa la cridada en SOAP.

Existeixen APIs de desenvolupament capaces de generar classes de consulta d'un servei web donat el seu WSDL.

6.4 Avantatges i inconvenients de SOAP

6.4.1 Avantatges de SOAP

Les principals avantatges de SOAP són les següents:

- SOAP no implica lligar-se a un llenguatge de programació concret. El desenvolupador pot consumir un servei web SOAP en qualsevol llenguatge. L'existència d'una API de consulta en aquest llenguatge facilitarà la seva tasca, però res més.
- SOAP no es lliga a cap protocol de transport. Donada la seva naturalesa XML, aquest es pot enviar sobre qualsevol protocol de transport, encara que donada l'eclosió d'Internet i del món web, HTTP ha esdevingut el protocol més emprat pels desenvolupadors. Encara que es podria emprar SOAP sobre SMTP (un correu electrònic podria ser enviat mitjançant SOAP) o qualsevol altre protocol.
- Permet interoperabilitat entre entorns, independentment de la seva naturalesa (aplicacions client a un petit ordinador portàtil o un mainframe d'una gran empresa) i de la seva tecnologia (.Net amb Java, Ruby, Python, etc).
- Empra XML com a representació de la informació, que és un meta llenguatge molt potent, extensible, llegible, conegut i amb el suport de la comunitat informàtica.
- L'ús de l'http com a protocol de transport permet mecanismes per evitar firewalls, travessar proxies, etc.

6.4.2 Inconvenients de SOAP

- La seva implementació sense eines adients com una bona interfície d'usuari pot ser molt complicada.
- Com veurem en el cas dels Restful, SOAP no és que tingui inconvenients en si mateix, sinó que altres com RESTful empen de forma més eficient el protocol HTTP, ja que aquest ha esdevingut el principal protocol de transport que empra SOAP.

7 Serveis web basats en HTTP: REST.

7.1 La filosofia REST

REST (Representational State Transfer) és una arquitectura de programari per a continguts hipermèdia distribuïts sobre Internet, seguint la filosofia de la WWW. El terme es va originar l'any 2000, apareixent per primera vegada a una tesi doctoral sobre la web escrita per Roy Fielding, un dels principals autors de l'especificació de l'HTTP.

El terme REST es refereix originalment a un conjunt de principis per al disseny d'arquitectures en xarxa. Aquests principis resumeixen com els recursos són definits. El terme freqüentment és emprat en el sentit de descriure qualsevol interfície que transmet dades específiques d'un domini sobre HTTP sense una capa addicional, com fa SOAP. Aquests dos significats poden xocar i fins i tot solapar-se. És possible dissenyar un sistema

de programari de gran volum, d'acord amb l'arquitectura proposada per Fielding sense emprar HTTP o sense interactuar amb la Web. O també es pot dissenyar una interfície XML sobre HTTP que no segueixi els principis REST, i en canvi seguir un model RPC.

Cal destacar que REST no és un estàndard, és una arquitectura. Encara que REST no sigui un estàndard, està basat en la utilització dels següents estàndards com HTTP, URL, representació de recursos (XML/HTML/GIF/JPEG/...) i tipus MIME (text/xml, text/html, ...).

La motivació de REST és de seguir les mateixes característiques de la web que l'han feta esdevenir una vertadera revolució a nivell mundial. La web és la única aplicació distribuïda que ha aconseguit ser escalable a nivell de tot internet. L'èxit és degut a l'ús de formats de missatges extensibles i estàndards, amb un adreçament global (estàndard i extensible a la vegada).

El concepte central de la web és l'espai de URIs unificat. Les URIs permeten la densa xarxa d'enllaços on es poden publicar tot tipus de webs. Aquesta és la clau que els serveis REST segueixen, publicar serveis web sobre URIs talment com ho fa la web, sense implementar capes superposades a sobre que no ens aporten res, ja que amb els protocols i estàndards de la web, en tenim més que suficient.

7.2 Els principis de REST

Ja hem comentat que l'arquitectura de referència de REST és la web. Els objectius d'aquest estil d'arquitectura són els següents:

- Escalabilitat de la interacció amb els components. La web ha crescut exponencialment sense degradar el seu rendiment. Una prova d'això és la varietat de clients que poden accedir a la web: portàtils, servidors, telèfons mòbils, estacions de treball, etc.
- Generalitat d'interfícies. Gràcies al protocol HTTP qualsevol client pot interactuar amb qualsevol servidor HTTP sense cap configuració especial.
- Posada en funcionament independent. Cada dia nous servidors entren a la web, servint noves aplicacions i serveis, sense cap efecte colateral a cap servei existent. A més, els servidors nous són capaços d'entendre's amb els vells, gràcies a que l'HTTP permet l'extensibilitat mitjançant l'ús de capçaleres, mitjançant URIs, i també creant nous mètodes i tipus de contingut.
- Compatibilitat amb components intermedis. Els més populars intermediaris són els proxys, agrupant tota la tipologia de proxys que podem trobar: proxy cau, proxys de seguretat (firewall), proxys intermediaris (gateway), etc. Així, aquesta compatibilitat amb intermediaris ens permet implementar millores a la xarxa sense afectar el seu rendiment: reduir la latència d'interacció, reforçar la seguretat i encapsular altres sistemes.

7.3 Restriccions de REST

REST aconsegueix els anteriors objectius aplicant les següents 4 restriccions:

- Identificació de recursos i manipulació d'ells a partir de representacions URI. HTTP és un protocol centrat en URIs. Els recursos són objectes lògics als que s'envien missatges, indicant si se sol·licita informació de l'objecte o bé s'envia informació per modificar l'estat d'aquest objecte.
- Missatges autodescriptius. REST dicta que els missatges HTTP han de ser tan descriptius com sigui possible.
- Emprar els mètodes estàndards adients a cada acció. Així, per obtenir informació d'un servei emprarem el mètode GET, el qual es pot guardar a la memòria cau. En canvi, si volem modificar dades d'un objecte, emprarem el mètode POST, que no es guarda a la memòria cau. HTTP és un protocol sense estat i quan s'empra correctament, és possible interpretar cada missatge sense cap coneixement dels missatges precedents. Per exemple, HTTP envia informació d'un login anterior a cada missatge

(un identificador de sessió per exemple), així el client no s'ha d'identificar cada vegada.

- Hipermedia com un mecanisme de l'estat de l'aplicació. L'estat actual d'una aplicació web ha de ser capturada en un o més documents d'hipertext, residint tant en el client com en el servidor. El servidor coneix l'estat dels seus recursos, encara que no intenti seguir-li la pista a les sessions individuals dels clients. Aquesta és la missió del navegador, ell sap com navegar de recurs en recurs, recollint informació que el necessita o canviar l'estat que necessita canviar.

Els defensors de REST han cregut que aquestes idees són tan aplicables als problemes d'integració d'aplicacions com els problemes d'integració d'hipertext. Fielding assegura que REST no és la solució a tot, ja que alguns d'aquests principis es poden no adequar a les necessitats d'un servei web concret. Però adoptar REST com a servei web al manco segueixen una filosofia de disseny forta, de provat èxit, documentada, i amb debilitats conegudes.

7.4 Pautes a seguir per dissenyar un servei web REST

Les pautes a seguir per dissenyar un servei web REST són les següents:

- Identificar totes les entitats conceptuals que es vulguin exposar al servei.
- Crear una URL per a cada entitat. El nom de la URL ha de ser un nom identificatiu de l'entitat, no pot ser un verb.

Per exemple, no seria correcte fer:

<http://www.servei.com/entitats/getEntitat?id=001>

El correcte seria:

<http://www.servei.com/entitats/001>

- Catalogar cada un dels recursos als que poden accedir els clients per tal de definir si un recurs és només de consulta o bé es pot modificar o eliminar. La consulta es fa per mètode GET, mentre que la modificació o eliminació es fa mitjançant el mètode POST, PUT o DELETE.
- Especificar els formats de resposta de cada un dels recursos accedits mitjançant un DTD (per exemple). També s'ha d'especificar les respostes als mètodes PUT, POST o DELETE.
- Especificar quins codis de resposta HTTP pot retornar cada invocació d'un recurs. Recordem els codis de resposta HTTP: 200, 201, 301, 400, 410, etc.
- Totes les consultes GET únicament obtenen representacions dels elements, no tenen cap efecte sobre les dades emmagatzemades al servidor.
- Descriure en un document com ha de ser invocat el nostre servei, ja sigui en un html o un WSDL.

8 Per què sorgeix el debat entre serveis web REST i SOAP?

El problema principal és que SOAP és massa complex per molts dels casos d'ús als que dona solució. La idea original de SOAP era que fos una versió de DCOM o CORBA sobre Internet. Així ho demostra el seu predecessor, l'XML-RPC. Aquestes arquitectures, basades en models RPC són més adequades en entorns aïllats, controlats, i amb un nombre determinat de serveis i de clients. En aquests casos és molt senzill evolucionar aquests sistemes, ja que donada la localitat del problema, és fàcil modificar el servei i els clients. Però quan el nombre d'usuaris és molt gran, aquesta estratègia és errònia. El problema amb l'eclosió d'Internet, és que aquest entorn passa a ser global, sense un control estricte (tecnologies molt diverses que proveeixen i consumeixen serveis) i amb un nombre il·limitat de clients que poden consumir un determinat servei. Així, s'han de menester bastiments que permetin evolucionar, tant pel costat del client com del servidor i que tinguin interoperabilitat entre sistemes.

La solució a això és la web. A primera vista es pot pensar que SOAP ho fa, ja que emprà HTTP com a mitjà de transport. Però Fielding argumenta que la Web funciona millor quan s'empra a l'estil que ho fa REST. Emprar HTTP com a mitjà de transport per a protocols d'aplicació a través de firewalls és una idea equivocada. Això redueix l'efectivitat de tenir un firewall, la qual cosa augmenta les possibilitats de nous forats de seguretat.

Per contra, els partidaris de SOAP argumenten que gràcies a la tecnologia existent que permet als dissenyadors encapsular la complexitat del sistema, donant lloc a interfícies generades automàticament que permeten facilitar el disseny del sistema. Segons argumenta Spolsky [3], el que és interessant de SOAP és que permet emprar llenguatges d'alt nivell per cridar i per implementar el servei. Afegeix que "al·legar que SOAP és dolent perquè el format de connexió és horrible i pesat és com al·legar que ningú hauria d'emprar Pentiums perquè el seu joc d'instruccions és molt més complicat que el joc d'instruccions del 8086. Això és cert, i per això existeixen els compiladors".

8.1 Resum de característiques REST i SOAP

Característiques de SOAP:

- Múltiples instàncies d'un procés comparteixen la mateixa operació.
- Una única adreça per a totes les operacions.
- Components fortament acoblats.
- Operacions i dades definides al WSDL.

Característiques REST:

- Direcció única per a cada instància del procés.
- Components feblement acoblats.
- Les operacions es defineixen en el missatge (mètode i direcció).
- Els objectes poden acceptar les operacions definides a l'estàndard.

8.2 Avantatges i inconvenients entre REST i SOAP

Avantatges de REST:

- Baix consum de recursos.
- No és necessària la informació d'enrutament, únicament necessitem la URI.
- Aprofitament de l'estàndard HTTP.
- Segueix el disseny i filosofia de la web.
- No subjecta a cap llenguatge ni plataforma determinat.
- Més senzill que SOAP.
- Corba d'aprenentatge més reduïda, ja que depèn menys de l'experiència del desenvolupador en unes eines determinades.
- Generalment fàcil de construir i adoptar.

Avantatges de SOAP:

- No subjecta a cap llenguatge, plataforma ni protocol de transport determinat.
- Generalment fàcil d'emprar si es disposen de les IDEs adients, ja que faciliten aquesta tasca.
- La publicació d'una APIs existents és senzill.
- Les operacions complexes es poden amagar sota una interfície senzilla.

- És l'estàndard actual per a serveis web, el qual fa que els proveïdors actuals proporcionin eines per ell.

Inconvenients de REST:

- Poques eines de desenvolupament adaptades a aquesta filosofia.
- Manejar les URIs pot ser embullós.
- Lligat al protocol de transport HTTP.
- Mancança d'estàndards per a la seguretat, polítiques de reenviament de missatges, models de comunicació amb intermediaris, etc.

Inconvenients de SOAP:

- Conceptualment més complicat.
- Major consum de recursos.
- Dedicació de ports a operacions.
- Molt més difícil de programar si no tens les eines adients.

9 Noves tipologies d'APIs

Fa uns anys quan es parlava d'una API web pràcticament tothom del món de la informàtica entenia que s'estava parlant de transaccions xml, normalment en SOAP. La gran majoria de les APIs d'accés a una aplicació estaven fetes amb aquesta arquitectura.

Degut a l'explosió de la web els darrers anys, la tipologia d'APIs també ha sofert grans canvis. Així, els APIs web han deixat de ser simplement xml intercanviats entre servidors per tenir altres formes de representació.

Aquests nous tipus de serveis poden ser imatges que es descarreguen en el navegador client (sense passar pel servidor de la web que allotja la pàgina que conté la imatge), dades codificades en json, codis javascript que es descarreguen també en el client i realitzen una tasca concreta, etc.

Personalment penso que els serveis que a continuació veurem també són serveis web, encara que la seva forma de ser publicats i la seva resposta no sigui un xml. La novetat de la majoria d'aquestes noves API és que la comunicació va des del navegador de la màquina del client al servidor, i no de servidor a servidor.

9.1 Ajax

Google és un dels precursors d'aquesta tecnologia de serveis web. Realment no és una nova tipologia de servei web, sinó una utilització conjunta de la filosofia ajax més un servei web REST. Així, el client consumidor d'un servei web no ha de programar res a la seva part de negoci al servidor. Únicament ha d'incloure una sèrie de llibreries javascript i una sèrie de línies de codi (javascript i html) per tal d'inserir dins la seva aplicació una eina de consulta d'un servei web determinat. Aquestes llibreries javascript el que fan és emprar Ajax per fer una consulta a un determinat servei http. Realment el que fan és consumir un servei web REST i pintar el resultat a la plana web. Normalment aquestes llibreries ofereixen moltes possibilitats de ser parametritzades, poden adaptar tant el funcionament com el look&feel a les necessitats de la plana que els adopta. El funcionament es parametritza mitjançant valors de paràmetres a la creació de l'objecte javascript que maneja tot l'objecte. Mentre que l'apariència es modifica mitjançant fulls d'estils propis a la plana, ja que el resultat no deixa de ser codi html amb una apareïència modificable a partir de fulls d'estils.

Les avantatges d'aquesta filosofia són vàries:

- L'aplicació que consumeix els serveis web descarrega la part del consum del servei sobre el client, és a dir, el servidor de l'aplicació no suporta cap càrrega, no fa cap transacció. És el client que fa directament les peticions al servei web remot.

- Són generalment molt fàcils d'implementar i no suposen cap complexitat. En un temps ràpid pots tenir el servei a la teva web.

Els inconvenients són els següents:

- La presentació i el control de la informació mostrada és poc modificable. Únicament les possibilitats que el propi servei proveeix i fulls d'estil propis. La llibertat de poder mostrar la informació de qualsevol manera o de poder filtrar-ne, no és possible.

9.2 Processos web

Aquest és un servei web que també s'està començant a emprar molt els darrers anys, sobretot a entorns empresarials. No he trobat informació a la web però per experiència personal crec que és important remarcar-lo. Consisteix en que una web A vol emprar un servei de la web B, però no vol desenvolupar l'API, ja sigui per qüestions de complexitat, de pressupost o de no disponibilitat de recursos. Així B li proveeix a A una entrada a un cert punt de la web B (entrada mitjançant un salt de web a web), passant una sèrie de paràmetres acordats amb anterioritat. El client desenvolupa el procediment que sigui sobre la web de B i finalment aquesta retorna a la web A el control mitjançant un salt de web a web informant d'una sèrie de paràmetres acordats, en el que hi hauria d'haver el resultat de les accions del client sobre la web B.

Els avantatges d'aquesta filosofia són varis. És senzill d'implementar, ja que únicament implementant els salts d'entrada i sortida als processos que es vulguin ja es disposa d'ells. Aquests salts es poden fer simplement amb un pas de paràmetres mitjançant forms. Per webs d'empreses amb recursos limitats, és una bona solució per poder emprar serveis web d'altres empreses, sense haver d'implementar de nou l'api duplicant el procés.

Un dels inconvenients d'aquesta filosofia és que el disseny de la web B pot ser molt diferent de la web A. Això provoca que el salt pot quedar molt malament des del punt de vista estètic a més de què la idea és que A cridi a B i el client no en tengui constància del salt. Si el client veu clarament el canvi de web, pot tenir una sensació d'inseguretat i tancar el navegador. Hi ha molts mecanismes per dissimular aquest salt de web a web. La web B pot oferir aquest servei oferint diverses possibilitats de modificació del seu look&feel per tal d'adaptar-se el més possible a la web A que el consumeix. Aquestes adaptacions es poden fer de diverses formes:

- La web A configura a la web de B els seus colors corporatius i B quan es crida per A mostra aquests colors (modificant d'alguna forma el seu codi, amb fulls d'estils o modificacions del codi).
- La web A quan salta a B informa dels seus fulls d'estils a la web B, i aquesta aplicarà aquests fulls d'estils en lloc dels seus. Prèviament la web A ha d'haver desenvolupat aquest full d'estil amb els elements de B per tal de què s'apliqui correctament a B.
- La web A pot cridar a la web B dins un iframe, inserint a fora de l'iframe les capçaleres i altres elements característiques de la web A

Un aspecte a tenir en compte és la seguretat. L'arquitectura està basada únicament en salts. Així, inspeccionant el codi de la web client podríem veure els paràmetres passats i copiar el salt. Per això s'han d'implementar mesures per tal de verificar que els clients dels serveis web estan autoritzats. Això es pot fer de diverses formes. Per exemple, obligant a que les webs que emprin aquests serveis abans del salt a la web hagin de fer una transacció d'identificació i validació del salt. Aquesta transacció retornarà un id que s'haurà d'informar als salts que es faran en aquella sessió. També s'han de validar els Referers de les capçaleres HTTP per tal de verificar que la web que crida el servei està autoritzada.

Exemples d'aquestes arquitectures de serveis les podem trobar a webs d'agències de viatges. Per exemple, si una agència de viatges vol oferir lloguer de cotxes a la web, però té recursos limitats per implementar l'API d'una empresa de lloguer de cotxes determinada, pot decidir emprar un servei web d'aquests comentats. Això redueix molt el temps de desenvolupament. I si aquest servei web té el disseny parametrizable, el client no es donarà compte del canvi de web.

9.3 Imatges

Aquest servei és el més senzill d'emprar i donada la seva senzillesa també ha irromput molt fort els darrers anys. La idea consisteix en què una web determinada ofereix un servei web que consisteix en una imatge on representa o es mostra informació referent al servei que es vol consumir. La web client només ha d'incloure la imatge al seu html, a la posició i dimensions que vulgui (depenent de les opcions que ofereixi el servei). L'empresa creadora del servei el que fa és crear la imatge dinàmicament i mostrar la informació sol·licitada.

Alguns exemples clars d'ús d'aquesta arquitectura serien els següents:

- Serveis de meteorologia: pàgines com www.meteoclimatic.com ofereixen aquest servei. Així, pots posar fàcilment a la teva pàgina web o al teu blog una imatge on es mostrarà el pronòstic meteorològic de la ciutat que vulguis. I la imatge s'actualitzarà automàticament i mostrarà sempre la darrera informació sense que ens haguem de preocupar de canviar-la.
- Servei d'estadístiques de visites: un exemple d'aquest servei seria www.sitemeter.com. Mitjançant la inclusió d'una imatge determinada d'aquest site a la teva pàgina, obtens un servei d'estadístiques web molt útil. Cada vegada que es carrega la imatge, la web del servei genera capta estadístiques per generar aquesta informació tan important per una web. A més, la imatge mostra quants usuaris han visitat la teva pàgina (es pot configurar si no es vol mostrar).

9.3.1 RSS

Un servei que cal destacar per la seva gran difusió últimament ha estat el servei RSS. Es pot catalogar com un servei web, i de fet algunes empreses ja l'empren com a servei web disponible on ofereixen els seus serveis, o fan publicitat dels seus productes.

RSS són dades codificades en un xml mitjançant un format determinat que serveix per a oferir informació actualitzada a les persones que s'hagin subscrit a aquest RSS. Aquesta subscripció és anomenada sindicació. Últimament tenen molt d'èxit aplicacions web agregadores de serveis RSS com és el Google Reader [4] o el Netvibes [5]. Aquestes aplicacions permeten gestionar als usuaris les seves sindicacions, i mostren la informació actualitzada dels llocs webs als que estan subscrits els usuaris. Això permet als usuaris estar al corrent de les novetats de les webs sense haver-les de visitar periòdicament.

Donades la potència d'aquesta eina, moltes empreses l'empren per publicar les seves novetats o ofertes. També altres com Yahoo Weather l'empren per donar informació meteorològica periòdicament actualitzada sobre el lloc que es demani.

10 Exemples actuals de serveis web

10.1 Delicious

Arquitectura: REST.

Delicious [6] és una coneguda aplicació web en la que els usuaris es poden crear un compte personal i emmagatzemar urls de llocs web que vulguin recordar. Els llocs web es guarden baixa un nom, una descripció, i es poden etiquetar per a facilitar la seva organització. Resulta molt útil, ja que qui no ha perdut adreces de webs guardades a la carpeta de favorits d'algun pc. També es poden consultar quines són les webs més desades pels usuaris, navegar per etiquetes, etc.

Delicious proveeix una senzilla API basada en REST molt senzilla d'emprar i amb la que s'han pogut fer proves satisfactòries. S'ha de dir que no és RESTful, però és molt senzilla i cobreix les necessitats de forma simple i correcta.

Primer de tot s'ha fet un registre a la web de Delicious com a usuari i s'han guardat una sèria de urls de llocs web per provar la funcionalitat. Posteriorment s'ha estudiat la documentació de l'API, per cert, molt clara i senzilla, i s'ha implementat una consulta de llocs web i d'etiquetes d'un usuari donat. S'ha fet una pàgina web on es demana un usuari i password de Delicious i es mostra els posts i etiquetes que té emmagatzemats aquest usuari a la web.

Cal comentar que si es vol emprar l'API de Delicious mitjançant una id de yahoo es pot fer mitjançant OAuth [7]. Això és interessant perquè s'ha trobat que molts APIs estan passant a implementar la seva identificació mitjançant OAuth. Aquest és un protocol obert que proveeix una forma segura i simple d'autenticació entre aplicacions, ja siguin web o d'escriptori.

Possible aplicació: a una web a la qual es disposa d'una pàgina amb enllaços a altres pàgines, es podria fer que aquests enllaços fossin carregats des de Delicious, carregant enllaços d'una determinada etiqueta, o bé enllaços d'un determinat usuari que crearíem prèviament. Així, si volguéssim afegir enllaços a la pàgina, no hauríem de canviar el codi, sinó que simplement afegir-lo a la compta Delicious de l'usuari consultat.

10.2 Ebay

Arquitectura: SOAP.

Ebay [8] a primera vista pareix que dóna moltes facilitats per utilitzar el seu API. Té un portal per a desenvolupadors (<http://developer.ebay.com/>) on es pot accedir a tota una bateria de recursos, documentació, codis d'exemple, etc. Però després comencen els problemes. En aquest web hi ha un excés d'informació i està sensiblement desordenada, la qual cosa complica la recerca d'informació o de solucions. Per començar, Ebay proveeix tres tipus diferents d'API: SOAP, xml sobre HTTP i REST. D'entrada es fa complicat decidir quina es vol emprar. En el nostre cas s'han provat les tres, sense gaire èxit en les proves fetes.

Primer de tot s'ha fet el registre al portal per tal de poder accedir als recursos disponible. El registre proveeix una sèrie de claus necessàries per emprar l'API, encara que segons s'emprí una API o una altra es necessitaran unes o unes altres. Després es va baixar l'SDK per .Net, però està compilat amb VS2003, i no s'ha pogut obrir en VS2008 perquè donava problemes de compatibilitat. De totes formes es pot obrir el codi font per observar com està implementat.

S'ha pogut implementar un únic mètode de l'API SOAP (el GeteBayOfficialTime), perquè inexplicablement la resta de mètodes no retornen cap valor. No s'ha arribat a saber si la falta de resultats és degut a un error en el codi o bé degut a ser l'entorn de proves. També s'ha trobat una falta de resposta lògica a les transaccions, és a dir, la transacció no retornava res i no informava de cap error. Comentar que el WSDL és molt pesat, 4 MB, la qual cosa fa que per carregar-se al VS tarda molt, arribant en algun moment a penjar la IDE.

En el cas de les proves a l'API REST, s'ha de dir que l'API respecte poc les bases de REST. L'operació a fer és un paràmetre de la consulta HTTP, i l'autenticació es fa en el cos del missatge SOAP. A més, per a utilitzar-lo s'ha d'emprar un nou paràmetre anomenat token que s'ha d'obtenir a la web, i és una cadena de caràcters de varis centenars de lletres. Totes les proves que s'han fet inexplicablement han donat un error de token. S'han obtingut varis tokens, s'han passat per POST, GET, s'han codificat amb urlEncode, etc. Però no s'ha aconseguit que la resposta a la transacció fos positiva.

En resum, s'ha invertit molt de temps en l'API d'Ebay sense obtenir un resultat gaire satisfactori. Es milloraria el servei de l'API d'Ebay si mostrés més clara i ordenada la informació al web. Hauria de ser més clar sobre quina és la millor forma d'implementar la autenticació a cada un dels tipus d'API. A uns demana unes claus dins el cos del missatge SOAP. A l'API REST segons la seva documentació es pot passar el token per paràmetre Get, la qual cosa és estranya donat que el token és una paraula alfanumèrica de varis centenars de caràcters. També s'ha trobat a faltar una ajuda als desenvolupadors en cas d'error. Les proves mitjançant prova i error poden ser desesperants. A la xarxa tampoc s'ha trobat gaire informació sobre el desenvolupament de l'API d'Ebay.

Possible aplicació: si una persona és un usuari d'Ebay i ven productes, podria montar una pàgina web que fos una tenda, consultant els articles que té a la venda a l'eBay i mostrar-los a la seva pàgina, vinculant-nos mitjançant un enllaç a eBay si els clients estan interessats en comprar-ho.

10.3 Yahoo Weather

Arquitectura: RSS.

Yahoo [9] és una de les empreses a la xarxa que proveeix més serveis web per a poder emprar gratuïtament. Entre ells podem trobar aquest Yahoo Weather. Aquest és un servei d'informació meteorològica (<http://developer.yahoo.com/weather/>) que consisteix en proveir un RSS amb informació actualitzada. Aquest RSS porta un paràmetre w en el que s'ha d'indicar un nombre anomenat WOEID. Aquest nombre significa Where on Earth Identifier, i és un identificador únic creat per Yahoo! Per identificar qualsevol lloc a la terra.

En el nostre cas s'ha cercat el WOEID de Palma de Mallorca i es mostra la informació meteorològica per aquesta ciutat. S'ha aconseguit mirant el codi font de la pàgina de Yahoo!, perquè no s'ha trobat cap lloc web on poder trobar fàcilment aquests codis WOEID.

El servei consisteix en un xml amb format RSS. Així es pot emprar per ser agregat a qualsevol lector de RSS com el Google Reader, i tenir actualitzada la informació del temps del lloc elegit. El codi desenvolupat consisteix en obtenir l'xml i llegir el resultat, mostrant la informació més rellevant. No cal identificació de cap tipus, únicament s'ha de carregar l'xml de la url donada.

Aquest és un servei molt senzill i interessant, ja que és molt fàcil d'implementar. El tema del WOEID també resulta molt interessant, ja que està associat a un servei de geolocalització proveït per Yahoo!. Aquest disposa d'un API mitjançant la qual es poden obtenir tant el WOEID com les coordenades de latitud i longitud d'una adreça física donada. És similar a la Geolocation API de Google, ja que ofereix un servei molt similar.

Possible aplicació: la informació meteorològica és molt útil per a molts d'usos. Un exemple senzill seria posar la informació meteorològica de la teva zona geogràfica a la teva web. Un altre exemple més complet seria que una web d'una agència de viatges online, podria implementar l'enviament d'un correu electrònic als clients un o dos dies abans de la data de sortida de la seva reserva amb la informació meteorològica dels propers dies de la zona on han de viatjar. Així el client rebria quin temps farà allà on ha de viatjar, una informació molt útil segons el destí triat i a l'hora de fer la maleta.

10.4 Twitter

Arquitectura: REST.

Un dels serveis que han irromput és fort a la web els darrers anys és el Twitter [10]. És el sistema de microblogging per excel·lència. Els seus usuaris envien microentrades basades en texts, denominats tweets, d'una longitud màxima de 140 caràcters. Aquestes entrades es mostren a la pàgina del perfil de l'usuari, i s'envien automàticament a altres usuaris que han elegit l'opció de rebre-les.

L'API de twitter ha esdevingut també un element important per al seu creixement. De fet, segons Isaac Stone (cofundador de twitter) més del 50% del tràfic de twitter arriba via API [11]. La seva importància radica en què ha permès que moltes de les entrades i altres interaccions no siguin fetes directament sobre la web de twitter, sinó que s'han creat altres serveis i altres webs que permeten fer-ho, la qual cosa suposa que la seva web ha de suportar menys tràfic i permet el sistema ser més escalable.

La documentació de l'API és molt clara. Té una arquitectura REST, i és molt senzilla d'entendre i d'implementar. Té tots els mètodes necessaris per interactuar amb Twitter: consultar tweets, generar-ne, modificar dades de compte d'usuari, etc. La implementació ha estat molt senzilla i no té cap complicació. Es fan consultes al servei REST i s'obté un xml.

Possible aplicació: twitter és una bona font d'informació de darrer moment. Una web podria consultar el que es diu sobre un tema en concret. Per exemple, una web de tecnologia podria consultar què es diu sobre el windows 7 o sobre la ipad. O una web d'esports podria consultar el que es comenta sobre el campionat del món. També es pot fer un seguiment d'un usuari concret. Així, una web d'un mitjà de comunicació pot mostrar a una secció el que escriuen els seus periodistes al twitter.

10.5 Google Ajax Search API

Arquitectura: AJAX en el navegador del client.

Google [12] és el principal cercador d'informació a la web arreu del món, i com a empresa és una de les principals fonts d'innovació en el món informàtic. Són pioners en molts de camps, i el que fan es mira amb lupa per la resta del món. I en aquest cas també foren pioners en el món dels serveis web. Començaren fent el servei web principal de cerques amb SOAP, i a partir de l'any 2006 l'abandonaren i ofereixen el mateix servei basat en Ajax i REST. Això és indicatiu de la migració paulatina de serveis basats en SOAP cap a arquitectures REST. Com podem veure al blog de Google Code [1], la SOAP Search API fou jubilada i declarada com a desfassada l'any 2006, encara que donaren 3 anys més de manteniment per a les aplicacions que encara l'empraven.

Com hem comentat, aquest nou servei de Google està basat en Ajax i REST, i s'implementa directament sobre l'html, mitjançant javascript. Això fa que el servidor web de la nostra plana web no faci cap interacció amb el servidor de Google, sinó que ho fa directament el navegador del client. Això allibera d'una càrrega considerable al nostre servidor, però per contra perdem control sobre la informació mostrada i sobre la seva presentació.

La seva implementació és molt senzilla. Primer s'ha d'obtenir una clau per a poder emprar l'API. Es pot obtenir fàcilment al mateix lloc de Google Code. Una vegada obtingut la clau es pot inserir el codi html i javascript necessari a la nostra plana web per tal de mostrar la informació de Google.

Possible aplicació: aquesta API ens permet emprar el motor de cerca dins la nostra web. Seria afegir una utilitat més a la nostra web, si decidíssim emprar la cerca general. Però una bona utilitat seria emprar el motor de cerca restringint-lo al nostre web. Així podem habilitar als usuaris una cerca de termes dins les pàgines del nostre web, i no l'hem d'implementar nosaltres.

10.6 Google REST Api

Arquitectura: REST.

Així com s'ha implementat en html la Google Search Ajax Api, també s'ha volgut implementar l'API Rest de Google. No hi ha gaire documentació ni Google la publicita gaire, pot ser perquè volen que s'empli mitjançant gadgets propis i la pròpia Api Ajax. La seva utilització és molt senzilla. No necessita identificació ni Google key. És una cridada http a una url i retorna la respota. La única peculiaritat és que aquesta API retorna la informació en format json, expressament per ser emprada en un entorn client mitjançant javascript. Llavors en entorn servidor s'ha de convertir a classes manejables. Això, després de molt cercar, s'ha aconseguit mitjançant uns mètodes propis de .Net.

Possible aplicació: aquesta API permet fer cerques sobre Google des del servidor, i així poder controlar molt més els resultats mostrar. Per la qual cosa es podrien habilitar cerques sobre la nostra web i filtrar el que vulguem segons els nostres requeriments.

10.7 Google Maps

Arquitectura: javascript i imatge en el navegador client.

Google Maps [13] és un altre dels serveis més coneguts de Google. És una eina molt útil per cercar carrers i ubicar direccions de llocs que no coneixem. A més, gràcies a la seva API s'ha estès per molts de llocs web, com a referent per a ubicació de llocs. Per exemple, és normal trobar a webs de viatges la ubicació dels hotels mitjançant l'API de Google Maps.

L'API consisteix en codi html i javascript, que inserit a la teva plana web ja s'encarrega de gestionar l'obtenció de la imatge del mapa. Té una sèrie de paràmetres per poder variar la configuració del mapa.

Com a la majoria de serveis de Google, la seva utilització és molt senzilla. S'ha de disposar d'una clau de Google, fàcil d'obtenir al lloc web de Google Code. Una vegada s'obté la clau, es pot copiar el codi html i javascript.

Possible aplicació: les aplicacions de l'API de Google Maps són gairebé infinites, i actualment en podem trobar moltes a la xarxa. Des de poder ubicar hotels a una web d'una cadena hotelera o d'una agència de viatges online, fins a indicar la situació de la petita tenda que té una web a internet amb la informació de la mateixa.

10.8 Bing

Arquitectura: SOAP

El nou cercador Bing de Microsoft ha nascut com un intent de restar quota de mercat a Google. És un potent cercador de l'estil mateix de Google. Disposa d'API de consulta. Està disponible en REST i en SOAP. En aquest cas s'ha optat per implementar SOAP donat que de REST ja teníem altres exemples.

La implementació ha resultat senzilla. L'únic problema trobat és que hi ha dues versions de l'API (1.1 i 2.0), i has d'anar en compte quin estàs emprant tot el temps per no equivocar-te, ja que varia bastant la seva implementació d'una a l'altra. Per a poder emprar l'API s'ha de disposar d'una clau. Aquesta clau s'obté a la web de Bing, on també podem trobar documentació i codi d'exemple.

Possible aplicació: les possibles aplicacions de Bing són les mateixes que les de Google, és a dir, les possibilitats que et dona disposar d'un cercador i poder fer cerques controlades sobre la xarxa des de la teva web. Es pot emprar per cercar sobre la teva web, sobre un determinat tema, etc.

10.9 Altres serveis

S'ha provat d'implementar els serveis que podem veure a continuació.

10.9.1 Aemet

Es va demanar informació Aemet.es [14] (Agència Estatal de Meteorologia, Unitat d'Informació Meteorològica) sobre quins serveis proveïen per a ser consultades de forma online les prediccions. Després de varies setmanes es va rebre contesta que no hi havia inconvenient en emprar un XML actualitzat que podria emprar a la meua pàgina, però que havia de sol·licitar autorització d'ús i emplenar i firmar un imprès determinat. Finalment vaig abandonar la intenció d'implementar el servei per falta de temps i que es podia obtenir un bon resultat per exemple amb Yahoo Weather sense haver d'emplenar impresos.

10.9.2 Meteocat

Es va enviar un correu electrònic a l'adreça de contacte del personal de Meteocat per tal de sol·licitar informació sobre els serveis de consulta online mitjançant serveis web que s'anuncien a la web [2], però mai s'ha rebut resposta, per la qual cosa es va abandonar la iniciativa d'implementar aquest servei web.

10.9.3 Amazon

Un cas molt atractiu que s'ha de tenir en compte i s'ha començat a investigar és el d'Amazon [16]. El que va començar essent una web de venda de llibres a Internet ha esdevingut un proveïdor de serveis web de darrera generació importantíssim. És possiblement l'empresa capdavantera en oferta de serveis sobre el núvol, concepte sorgit darrerament per definir tots aquells serveis que s'ofereixen sobre internet i la implementació física dels quals és totalment desconeguda pels usuaris d'aquests serveis (hosting, bases de dades, emmagatzement, etc). L'èxit dels serveis web d'Amazon és tal que ha desbancat a Google en aquest camp (Google App Engine) [17], i l'ús d'aquests serveis web ja és major que el propi tràfic de la tenda [18].

Amazon ofereix accés a aquests serveis web mitjançant REST i SOAP. El preu dels serveis depèn de la magnitud de l'ús que se'n faci, i el preu depèn també del servei triat.

11 Conclusions

En aquest article hem vist com els serveis web han evolucionat molt en els darrers anys, així com han anat augmentant la seva importància en el món de la informàtica. I també podem constatar que encara estan en plena evolució. La irrupció d'Internet i de la web ha estat l'empenta definitiva per a l'eclosió dels serveis web. Hem vist com els serveis web clàssics estan evolucionant de SOAP a arquitectura REST, més adaptat a Internet i més fàcil d'implementar. També han sofert grans canvis les APIs web. Han sorgit noves tipologies

d'APIs sobretot adaptades al món web: llibreries javascript que implementen cridades Ajax, o imatges generades dinàmicament, arxius flash, etc.

Hem pogut comprovar com Google ha passat d'un servei web basat en SOAP a una API basada en Ajax. Aquest és un exemple clar de què les empreses estan tendint a fer serveis web senzills, orientats totalment a la web i fàcils de consumir. La idea de les empreses és facilitar l'adopció i la utilització dels seus serveis web, per així poder expandir el més possible el seu negoci i oferir els seus serveis a la major quantitat de possibles clients, o cercar sinèrgies de negoci amb altres empreses, compartint informació cercant una simbiosi que doni profits a ambdues parts.

És clara la migració de serveis web SOAP a serveis web basats en REST. Aquests aprofiten millor la filosofia de la xarxa, i són més fàcils d'implementar. Com hem pogut comprovar, programar un client d'un servei web en SOAP és més complicat que en REST, sobretot si no es tenen les eines adients. En canvi, amb REST és més senzilla la seva implementació, siguin quina sigui la plataforma tecnològica que s'empri.

Segons la meua opinió, els serveis també estan passant de ser únicament en format xml a adoptar nous tipus de format. Això fa que el terme servei web es redefineixi, i el concepte de servei web possiblement hagi canviat. Un servei web actualment és un servei sobre la web. Pot ser un xml sobre http, un component ajax o una imatge, però tots segueixen essent serveis web.

En el món actual, on els canvis són molt ràpids i les necessitats són molt immediates, és imprescindible fer serveis web senzills per tal de què siguin fàcilment consumibles, sense dependre de tecnologies concretes i sense lligar-se a un entorn de programació. La facilitat d'ús i la capacitat de ser implementat per diverses plataformes són algunes de les claus de l'èxit d'un servei web.

12 Treballs futurs

El treball futur sobre l'estat dels serveis web és bàsicament seguir observant cap on evolucionen aquests serveis web. La necessitat d'aquests per part de les empreses i organitzacions segueix augmentant, tant la necessitat d'emprar-los com de produir-los.

Tal com hem comentat a les conclusions, la necessitat d'establir cooperacions i sinèrgies entre empreses, de compartir sistemes i connectar aplicacions està a l'ordre del dia. Això fa que en els propers anys l'evolució dels serveis web sigui un camp molt atractiu d'estudi. Sorgirà molta oferta de serveis, així com noves formes de comunicació de sistemes.

Una filosofia que està augmentant considerablement la seva importància és la web semàntica. La idea d'afegir metadades semàntiques i ontològiques a la web i als propis serveis web per fer agents intel·ligents que donin una informació molt més acurada obre un gran camp de treball i de millora als serveis web.

Haurem d'estar atents a quines tecnologies impulsen els principals actors del món informàtic, com són les grans empreses com Microsoft o Google. Seguiran amb SOAP, optaran per REST i noves filosofies tipus Ajax com ja ha fet Google o bé inventaran noves arquitectures de serveis web i APIs. Especial atenció hem de dedicar al cas d'Amazon Web Services, degut a l'augment de l'ús dels serveis en el núvol. Però tampoc ens hem d'oblidar de les petites empreses, o programadors que sovint aporten noves solucions i que també innoven constantment. Segur que des d'aquest darrer grup sorgeixen noves idees molt interessants.

13 Referències

- [1] Eric Tholomé, Google Code, A well earned retirement for the SOAP Search API (<http://googlecode.blogspot.com/2009/08/well-earned-retirement-for-soap-search.html>)
- [2] Serveis que ofereix Meteocat (http://www.meteocat.com/mediamb_xemec/servmet/marcs/marc_produc.html)
- [3] Joel on Software, Joel Spolsky (<http://www.joelonsoftware.com/news/20020425.html>)
- [4] Google Reader (<http://reader.google.com>)

- [5] NetVibes (<http://www.netvibes.com/es>)
- [6] Delicious (<http://delicious.com/>)
- [7] OAuth.net (<http://oauth.net/>)
- [8] eBay (<http://www.ebay.es/clasico/>)
- [9] Yahoo! (<http://es.yahoo.com/>)
- [10] Twitter (<http://twitter.com/>)
- [11] Isaac Stone Interview, (<http://d7.allthingsd.com/20090526/biz-stone-and-evan-williams/>)
- [12] Google (<http://www.google.es/>)
- [13] Google maps (<http://maps.google.es/>)
- [14] Aemet (<http://www.aemet.es/>)
- [15] Bing (<http://www.bing.com/>)
- [16] Amazon Web Services (<http://aws.amazon.com/>)
- [17] Por qué Amazon Web Services sí tiene éxito y Google App Engine no (<http://www.nubeblog.com/2009/01/23/por-que-amazon-web-services-si-tiene-exito-y-google-app-engine-no/>)
- [18] Los web services de Amazon superan a la tienda (<http://www.maestrosdelweb.com/actualidad/los-webservices-de-amazon-superan-a-la-tienda/>)
- [19] World Wide Web Consortium (<http://www.w3.org/>)

14 Bibliografia

- Restful Web Services, web services for the real world. Leonard Richardson, Sam Ruby. Editorial O'Reilly. Edició del maig del 2007.
- Web Service (http://en.wikipedia.org/wiki/Web_service)
- Service Oriented Architecture (http://en.wikipedia.org/wiki/Service_Oriented_Architecture)
- Restful Web Services: The Basics (<http://www.ibm.com/developerworks/webservices/library/ws-restful/>)
- List of Web Service Specifications (http://en.wikipedia.org/wiki/List_of_Web_service_specifications)
- Meteocat (http://www.meteocat.com/mediamb_xemec/servmet/index.html)
- RESTful web services (http://en.wikipedia.org/wiki/Representational_State_Transfer)
- RPC (http://en.wikipedia.org/wiki/Remote_procedure_call)
- DCE (Distributed Component Object Model) de Microsoft (http://es.wikipedia.org/wiki/Distributed_Component_Object_Model)
- Java RMI (<http://java.sun.com/javase/technologies/core/basic/rmi/index.jsp>)
- Magnus Lassi, Taking a look at the Ebay API's (<http://ubiquitous-tech.blogspot.com/2009/02/taking-look-at-ebay-apis.html>)
- John Mueller, The New Era of Public Web Services (<http://www.devsource.com/c/a/Techniques/The-New-Era-of-Public-Web-Services/>)
- SOAP (<http://en.wikipedia.org/wiki/SOAP>)

- Shannon Horn, Integrating the Google Web Service into ASP.NET (<http://www.devx.com/codemag/Article/20644>)
- Api Directory in Programmable Web (<http://www.programmableweb.com/apis/directory>)
- Ajax Composite Apps – The Last Mile between Your Users and Your SOA, by Chris Warner (<http://opensource.sys-con.com/node/219091>)
- TERMCAT, Centre de terminologia (<http://www.termcat.cat>)
- Gran diccionari de la llengua Catalana.. Diversos autors (1998).