

Desenvolupament d'una plataforma per a la interacció amb
e-Services

Memòria de Projecte de Final de Carrera realitzar per Jordi Valls i dirigit per Felix Freitag

Índex

| | |
|--|-----------|
| 0. Agraïments | 1 |
| 1. Introducció | 3 |
| 2. Objectius | 7 |
| 2.1 Definició dels objectius | 7 |
| 2.2 Objectius optatius | 8 |
| 3. Estat de l'art | 11 |
| 3.1 Tecnologies per a l'oferiment de serveis..... | 11 |
| 3.1.1 Serveis centralitzats | 11 |
| 3.1.2 Serveis distribuïts..... | 12 |
| 3.1.3 Serveis distribuïts en xarxes col·laboratives..... | 12 |
| 3.2 e-Services..... | 13 |
| 3.2.1 Web Services..... | 13 |
| 3.2.2 Computació Grid i Grid services | 14 |
| 3.2.3 Control del workflow en els grid services | 17 |
| 3.3 e-Services: situació i escenaris actuals | 18 |
| 3.3.1 Escenari <i>marketplace</i> | 18 |
| 3.3.2 Escenari P2P..... | 19 |
| 3.3.3 Escenari <i>brokering</i> | 19 |
| 3.3.4 Situació del projecte | 20 |
| 3.4 Estudi de mercat: projectes existents | 21 |
| 3.4.1 Projecte GridLab | 21 |
| 3.4.2 Catàlegs de serveis | 22 |
| 4. Anàlisi del projecte | 23 |
| 4.1 Anàlisi de requeriments | 24 |
| 4.1.1 Requeriments funcionals | 24 |
| 4.1.2 Requeriments no funcionals..... | 24 |
| 4.2 Planificació del projecte | 26 |
| 4.2.1 Consideracions prèvies..... | 26 |
| 4.2.2 Etapes del projecte..... | 26 |
| 4.2.3 Planificació i diagrama de Gantt | 28 |
| 4.2.4 Anàlisi de la planificació. El camí crític..... | 28 |
| 4.2.5 Anàlisi de la planificació real | 29 |
| 4.3 Anàlisi de mercat i especificació del prototip..... | 29 |
| 5. Especificació | 31 |
| 5.1 Diagrama de casos d'ús | 31 |
| 5.2 Descripció dels casos d'ús | 32 |
| 5.3 Diagrama de classes d'especificació | 39 |
| 6. Disseny | 41 |
| 6.1 Arquitectura del sistema | 41 |
| 6.2 La capa de domini..... | 45 |
| 6.2.1 Diagrama de classes..... | 45 |
| 6.2.2 Selecció dels patrons de disseny | 46 |
| 6.2.3 Relació entre serveis: graf de serveis..... | 48 |
| 6.2.4 Gestió d'errors | 54 |

Al Dani, el meu compi P.T. Amb un company així, les penúries del CAOS sempre passen millor. Uops, el Dani se'ns casa! Moltes felicitats, Dani i Cristina!!

Al Javi, el grid-crack del CAOS, per tot l'ajut que m'ha donat amb els serveis grid, i per passar-se una nit lluitant al meu costat per aconseguir un hostname. A full!

A tots els companys del CAOS, pel seu constant interès, i per intentar no donar-me més feina de la que ja tenia ☺ gràcies! I una menció especial a tots els que en aquestes dates estan acabant tesis i tesines: ànim!!

I finalment, tot i que no pas en últim lloc, a tots els companys de la FIB, per ser com són. Carlos, Nando, Sandra, Fernando, Jose i tants altres, gràcies i ànim!

1

Introducció

En aquest apartat farem una ràpida introducció a l'entorn del projecte, comentant breument en què consisteix, en quin àmbit es trobat acotat i quines són les seves limitacions. Aquests conceptes es tractaran amb més profunditat en els capítols següents, però amb aquestes primeres pinzellades el lector es pot fer una idea general del projecte.

Al llarg dels últims anys, des de l'explosió d'Internet i la *World Wide Web*, la tendència en l'evolució de les tecnologies relacionades amb Internet anuncia un pas de la *web centrada en les persones* cap a la *web centrada en les aplicacions*. Les necessitats de tractar la informació dinàmicament han afavorit la proliferació de tecnologies com *CGIs*, *servlets Java*, *scripts PHP*, etc. que tenien com objectiu actuar d'intermediaris entre l'usuari i la informació.

Aquesta tendència ha arribat al seu màxim exponent amb l'aparició dels *Web Services*. Un *web service* consta d'un conjunt de protocols i estàndards que s'utilitzen per intercanviar dades entre aplicacions. El seu gran avantatge és que són totalment independents de la plataforma i el llenguatge de programació en què estan desenvolupades les aplicacions.

Així doncs, observant que actualment l'ofertament d'un servei és independent de la plataforma hardware i software on es troba, podem definir genèricament un **servei** com **l'ofertament d'un recurs computacional**, el qual pot anar des d'una senzilla operació matemàtica fins al *renderitzat* 3D d'una estructura complexa.

Ara bé, independentment de l'evolució *conceptual* d'una Internet *centrada en les persones* cap a una Internet *centrada en les aplicacions*, també hi ha hagut en els últims anys una dràstica millora de la infraestructura tecnològica disponible: xarxes d'alta velocitat per a Universitats i centres d'investigació, accés a la banda ampla a tots els usuaris, etc. Aquesta millora tecnològica ha comportat una tendència a la *descentralització*: compartició de la informació i, en última instància, compartició de **recursos**.

La compartició de recursos va començar sent utilitzada per determinades aplicacions (per exemple, compartir ample de banda per a l'intercanvi de fitxers en aplicacions P2P com *Bittorrent*, o bé compartir cicles de rellotge no utilitzats per a l'anàlisi de dades, com el projecte SETI@Home). Però ben aviat va aparèixer un estàndard per a la utilització de múltiples recursos computacionals per a oferir serveis complexes que una única màquina no podia resoldre: és l'anomenada *Grid Computing*, i els serveis que ofereix es coneixen com *Grid Services*.

Per englobar les diferents tecnologies existents per oferir serveis, denotarem com **e-Services** el conjunt de tecnologies, tant centralitzades com distribuïdes, que tenen com objectiu l'ofertament de serveis mitjançant protocols estandarditzats.

Així doncs, podem veure que el concepte de *servei* evoluciona constantment. Degut a aquesta evolució, així com la naturalesa distribuïda i cada cop més descentralitzada de les plataformes orientades a l'oferta de recursos i serveis, apareixen un conjunt de problemes:

- Diferents mecanismes de sol·licitud i negociació.
- Inexistència d'un paràmetre de *qualitat*: a partir d'una necessitat, no hi ha cap mecanisme que ens permeti decidir quina és la millor opció entre totes les ofertes.
- Poca connectivitat entre plataformes diferents.

Aquest projecte tindrà com a objectiu l'estudi i definició d'una plataforma tecnològica que permeti solucionar aquests problemes.

2 Objectius

En aquesta secció definirem formalment l'objectiu del nostre projecte, quin és el seu abast, i diferenciarem entre els objectius que considerem obligatoris per a la realització del projecte i aquells que són optatius, és a dir que no són part fonamental del projecte però aporten un valor afegit prou significatiu com per formar part del projecte.

2.1 Definició dels objectius

L'objectiu del projecte és el disseny i implementació del prototipus d'una plataforma que permeti la interacció entre un client i múltiples serveis, maximitzant el grau d'interacció i qualitat (entenent el concepte de qualitat com una entitat mesurable. Per exemple, si parlem del temps que triga un servei en realitzar-se: mínim temps → màxima eficiència → màxima qualitat).

A grans trets, consta de les següents parts:

- **Interfície d'usuari.**

S'implementarà una interfície d'usuari que permeti visualitzar els serveis oferts, escollir-ne un, introduir les dades o operands necessaris i visualitzar-ne els resultats.

La implementació es realitzarà mitjançant *servlets/portlets*, tenint especial cura en que sigui el més independent possible de la plataforma hardware/software per facilitar la connectivitat al màxim nombre de dispositius mòbils possible (telèfons mòbils, *PDA's*, etc.).

- **Service Access Point (SAP)**

Punt d'entrada a la plataforma, a través del qual els usuaris sol·liciten els serveis i visualitzen els resultats. També integra els mòduls d'administració, mitjançant els quals un usuari *administrador* pot donar d'alta nous serveis i usuaris.

- **Service Engine (SE)**

Motor de la plataforma. És l'encarregat de rebre les sol·licituds iniciades al *SAP* i gestionar-les fins arribar al resultat final. Així doncs, s'encarregarà d'analitzar les sol·licituds, convertir-les en les peticions adequades, enviar-les al Web Service corresponent, recollir el(s) resultat(s) i donar-li el format adequat abans d'enviar-lo de nou al *SAP*.

Adicionalment al disseny i implementació de la plataforma, es realitzaran un conjunt de proves, tant en entorns simulats com en entorns reals i es modelarà un cas d'ús d'exemple, per demostrar la validesa del disseny escollit.

Finalment, es generarà la documentació necessària per permetre l'ampliació de la plataforma.

2.2 Objectius optatius

A continuació detallem algunes ampliacions que completarien la plataforma, i que es realitzaran un cop s'hagin aconseguit els objectius obligatoris, segons les limitacions de temps que hi hagi:

- Implementació d'un mecanisme pseudo-intel·ligent que permeti decidir, a partir d'un conjunt de paràmetres, quin Web Service és el més adequat per una determinada sol·licitud.
- Implementació d'un mecanisme que permeti dividir un servei complex en diferents serveis més senzills. En conjunció amb el mecanisme anterior, es pot aconseguir un sistema que decideixi, per a cada sub-servei, on és millor sol·licitar-lo. Per a completar el mecanisme s'haurà de mantenir en tot moment la coherència de les operacions/serveis realitzats, així com la tolerància a fallades parcials (és a dir, si dividim un servei complex en 3 serveis més petits i d'aquests 3 un falla, com respon el sistema).
- Control del *workflow* (flux) d'un servei.

3 Estat de l'art

En aquest capítol farem un anàlisi de l'estat en què es troba la tecnologia que envolta el projecte. Concretament, estudiarem els protocols i plataformes que tindrem en compte en la realització del projecte, quins frameworks existeixen, quins productes similars al projecte que volem realitzar hi ha al mercat i quina és la tendència cap a on evolucionaran els e-Services.

3.1 Tecnologies per a l'ofertament de serveis

Primer de tot analitzarem quines són les tecnologies que existeixen actualment per a l'ofertament de serveis:

3.1.1 Serveis centralitzats

Els serveis centralitzats han estat les primeres tecnologies que han permès la sol·licitud remota de serveis. Inicialment es tractava d'aplicacions CGI, que podien

ser compilades (per exemple, escrites en llenguatge C) o bé scripts escrits en algun llenguatge de script com, per exemple, Perl.

A partir de l'evolució de Java han aparegut noves tecnologies més avançades i eficients per a l'execució remota i sol·licitud de serveis: servlets, RMI (Remote Method Invocation, invocació remota de mètodes), JSP, etc.

En els últims anys, l'oferta ha crescut enormement, amb la introducció de la tecnologia .NET de Microsoft, així com l'augment de popularitat de les aplicacions orientades a script, com PHP o ASP.

Entre tota aquesta oferta tecnològica, ens centrarem en una en concret: els Web Services. La raó és que aquesta tecnologia té com a objectiu la interconnexió d'aplicacions (per exemple, entre el proveïdor d'un servei i el client) fent que aquesta comunicació sigui totalment independent tant de la plataforma en què es troba com del llenguatge de programació en què s'ha realitzat.

3.1.2 Serveis distribuïts

En aquest conjunt s'introdueixen totes aquelles tecnologies que es basen en el paral·lelisme per incrementar el rendiment dels seus sistemes. La idea principal és trencar un procés complex i que necessitaria molts cicles de rellotge per executar-se en un conjunt de processos més senzills i repartir-los entre un conjunt d'ordinadors que actuarien com un únic "gran ordinador". Algunes d'aquestes tecnologies són PVM i MPI.

En aquest projecte no tindrem en compte aquests sistemes, ja que estan orientats principalment a conjunts homogenis d'ordinadors, normalment agrupats sota la mateixa organització. Per tant no posen un gran èmfasi en la uniformitat del sistemes d'intercanvi d'informació, sinó que ofereixen els seus propis mecanismes (típicament un node Màster o *Scheduler*).

3.1.3 Serveis distribuïts en xarxes col·laboratives

Tal com s'ha comentat en l'apartat anterior, els serveis distribuïts són un dels mecanismes més utilitzats a l'hora d'incrementar la potència de càlcul dins d'una institució, fent que un conjunt d'ordinadors comparteixin els seus recursos amb un mateix objectiu.

Partint d'aquesta base, una tendència que ha obert moltes línies d'investigació en els últims anys consisteix a la col·laboració entre diferents institucions, les quals comparteixen els seus *clusters* (que internament utilitzen serveis paral·lels com MPI, PVM i d'altres).

Alguns d'aquests sistemes són per exemple Condor, de la Universitat de Wisconsin, o bé el projecte Grid, el qual engloba molt projectes relacionats. És precisament aquesta última l'altra tecnologia que tindrem en compte dins el nostre projecte. Aquesta selecció s'ha donat per 3 factors:

- **Alta projecció.** El projecte Grid està avalat per la Unió Europea, la qual en financia un alt nombre de línies d'investigació en diferents universitats, i també per grans multinacionals com IBM.
- **Alt grau de normalització.** Al estar en gran part finançat per fons públics, el projecte Grid té un alt grau d'uniformitat: tots els protocols i tècniques d'intercanvi d'informació són adequadament publicats i sotmesos a un procés per esdevenir estàndards.
- **Comptabilitat amb Web Services.** Dels diferents frameworks que existeixen en el mercat per a interactuar amb la Grid, un dels més utilitzats és Globus. I des de l'última versió d'aquest *framework* es suporta Web Services com a sistema per a sol·licitar serveis.

3.2 e-Services

3.2.1 Web Services

Amb l'augment i popularització dels sistemes de serveis distribuïts ha començat a aparèixer un gran nombre de línies de recerca i projectes que tenen l'objectiu de millorar la interacció entre el client i el sistema i afegir-ne funcionalitats.

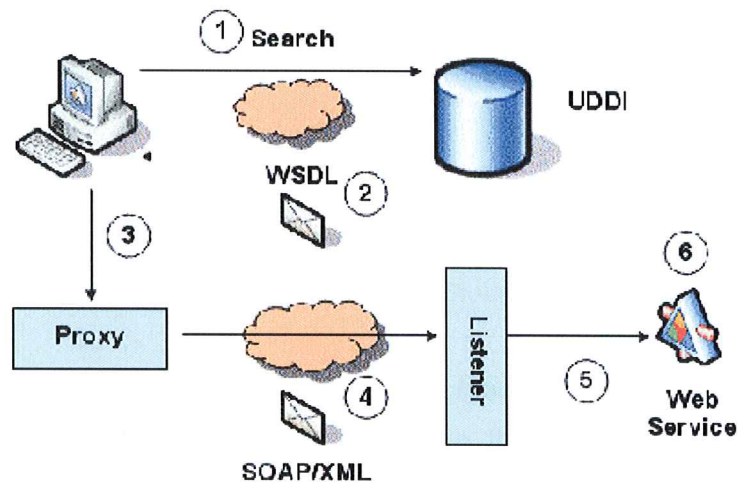
[Notació]

Degut a l'elevat nombre d'arquitectures que existeixen en l'actualitat, les englobarem en 2 conjunts per a poder diferenciar-los:

- Web services: serveis centralitzats
- Grid services: serveis distribuïts

[Definició]

Un **web service** és una interfície accessible a través d'una xarxa de comunicacions a les funcionalitats d'una aplicació, construïda utilitzant protocols estàndard d'Internet. [Tid01]

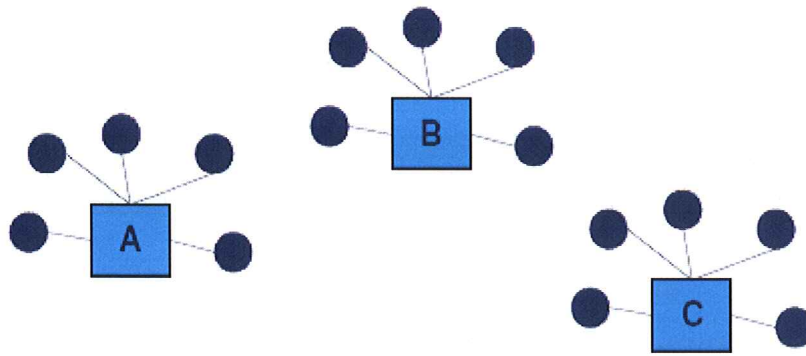


Il·lustració 1 - Protocol web service [Tid01]

Així doncs, els **web services** ofereixen un mecanisme per accedir a un servei mitjançant protocols estàndard, independentment del llenguatge de programació, plataforma software o hardware, etc.

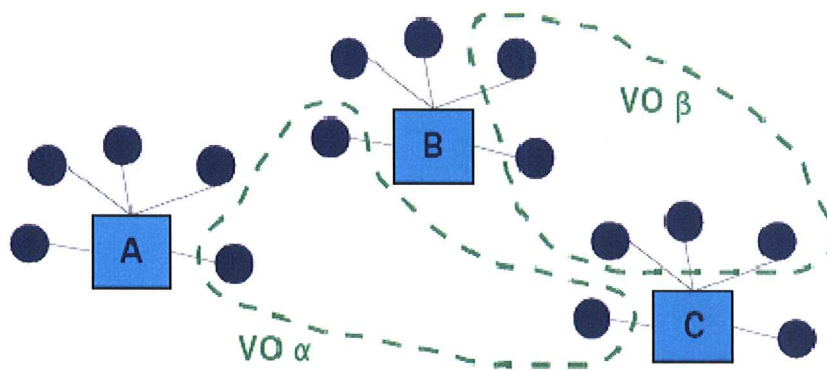
3.2.2 Computació Grid i Grid services

La computació Grid neix com a solució al problema del tractament de processos que necessiten una capacitat de processament que excedeix, en molt, la d'un ordinador. En front a la solució tradicional d'utilitzar sistemes paral·lels i distribuïts, la computació Grid proposa la col·laboració entre diferents organitzacions, que comparteixen els seus recursos.



II·lustració 2 - Organitzacions i recursos. Referència: [Sot04]

El principi en què es fonamenta la computació Grid és la creació d'**organitzacions virtuals (VO)**. Les organitzacions virtuals són agrupacions dinàmiques de recursos que tenen l'objectiu de resoldre un problema computacional.



II·lustració 3 - Organitzacions virtuals [Sot04]

[Definició]

"Una grid computacional és una infraestructura hardware i software que proveeix un accés fiable, consistent, persistent i econòmic a recursos computacionals d'alta qualitat." [Fos03]

La computació Grid té com a objectiu arribar a una plataforma que resolgui els problemes que apareixen en aquest paradigma de computació: planificació, paral·lisme, balanceig de càrrega, seguretat.

A l'actualitat hi ha moltes aplicacions que utilitzen la computació Grid de manera eficient. La majoria, però, acotades a l'àmbit acadèmic. Tot i això no existeix una plataforma universal de computació Grid, sinó que n'hi ha múltiples, cadascuna orientada a tipus específics d'aplicació.

Proyectos Grid

| EEUU | Europa |
|-----------------------------------|---|
| •NASA Information Power Grid | •EGEE (CERN, ...) |
| •DOE Science Grid | •DataGrid (CERN, ...) |
| •NSF National Virtual Observatory | •EuroGrid (Unicore) |
| •NSF GriPhyN | •DataTag (CERN,...) |
| •DOE Particle Physics Data Grid | •Astrophysical Virtual Observatory |
| •NSF TeraGrid | •GRIP (Globus/Unicore) |
| •DOE ASCI Grid | •GRIA (Aplicaciones industriales) |
| •DOE Earth Systems Grid | •GridLab (Cactus Toolkit) |
| •DARPA CoABS Grid | •CrossGrid (Componentes de infraestructura) |
| •NEESGrid | •EGSO (Física Solar) |
| •DOH BIRN | •UK e-Science Grid |
| •NSF IVGDL | |

Introducción a la Computación Grid

II-lustració 4 - Projectes Grid [Sot04]

Un dels problemes que té actualment la computació Grid és precisament aquest: l'existència múltiples "grids" independents. Com a proposta per solucionar aquest problema es crea l'associació **Global Grid Forum**, que té com a objectiu el disseny d'un **protocol de comunicació InterGrid** i una plataforma estàndard d'accés als seus serveis. La implementació d'aquest disseny es realitza sota el projecte de **Globus Toolkit**.

Amb aquests objectius neix la **OGSA: Open Grid Services Architecture**, una plataforma que modernitza i amplia el **Globus Toolkit**. Addicionalment, s'ha acordat la utilització dels **web services** com a mecanisme de sol·licitud i intercanvi d'informació entre els clients i la plataforma **Globus**.

És de gran importància el fet que les companyies multinacionals que treballen en computació Grid, com IBM, Microsoft o Sun Microsystems, han donat el seu suport a l'estàndard OGSA. Aquest és un primer pas per a esdevenir un estàndard *de facto*.

Així doncs, definirem els **grid services** com aquells serveis que pot oferir una plataforma Grid, com per exemple la OGSA.

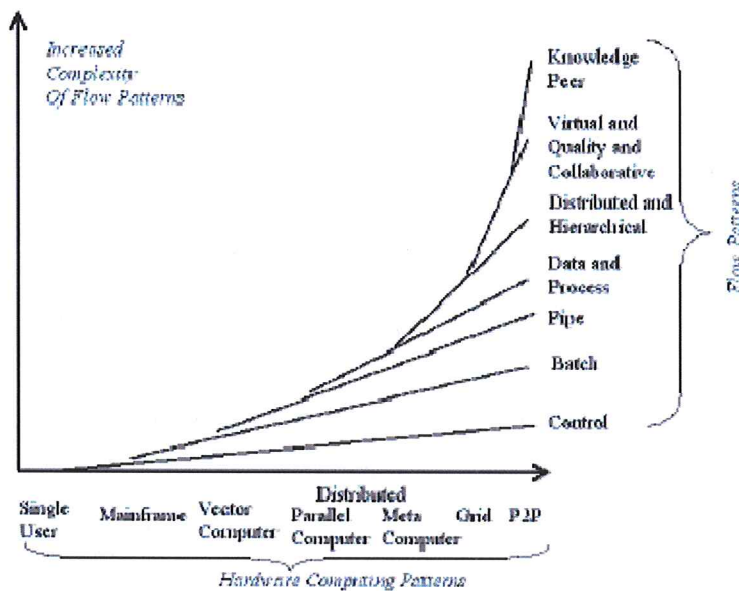
Aquesta manca d'un protocol o mecanisme estàndard per a la sol·licitud i comunicació entre Grids, així com el fet que és encara una tecnologia molt complexa en fase de recerca, és un dels factors que ha impedit el salt d'aquest tipus de tecnologies al món industrial i empresarial. En canvi observem com els **web services** ja s'utilitzen habitualment al mercat, com és el cas de molts portals i altres llocs web d'Internet.

3.2.3 Control del workflow en els grid services

[Definició]

Definim el *workflow*, o flux d'execució, com la composició de **grid services** que s'executen en recursos heterogenis i distribuïts, en un ordre ben definit per aconseguir un objectiu específic.

En un sistema Grid, compost per un conjunt heterogeni de recursos, totalment dinàmic, on les organitzacions virtuals es creen dinàmicament (si executem un servei 2 cops, segurament no s'executarà en els mateixos recursos, ni en el mateix ordre), el control del *workflow* esdevé un element clau i d'una elevada complexitat.



Il·lustració 5 - Evolució de la complexitat del flux d'execució [Fos03]

3.3 e-Services: situació i escenaris actuals

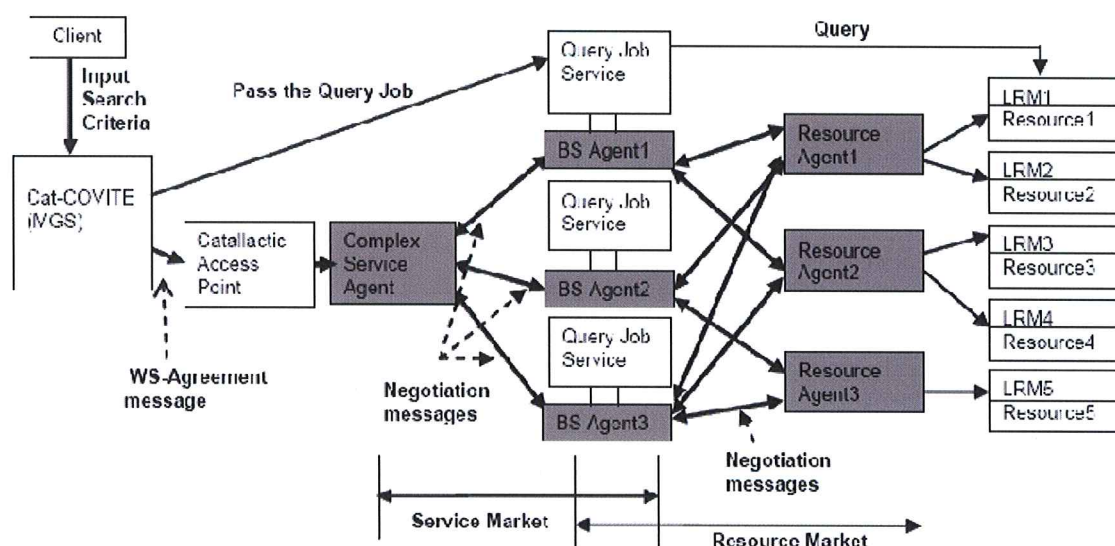
Tal com hem comentat darrerament, a l'entorn dels grid services s'estan desenvolupant un gran nombre d'arquitectures que es troben en fase de recerca o desenvolupament. A continuació descriurem diferents escenaris que il·lustren les línies en què evolucionen aquests projectes, per tal de situar el lector en el marc dels projectes d'investigació sobre Grid Services, que tindrà una gran importància en el nostre projecte.

3.3.1 Escenari *marketplace*

En aquest primer escenari s'engloben diferents línies d'investigació que tenen com objectiu arribar a un sistema de decisió que permeti a un client i un proveïdor decidir, en base a una sol·licitud, el servei a utilitzar i el cost d'aquest en base a un protocol de negociació.

Aquest paradigma és essencialment econòmic: es tracta d'aplicar l'estratègia de lliure mercat a l'hora de relacionar client i proveïdor, oferta i demanda.

Com a exemple d'aquest escenari podem citar el projecte *Catallactic Grid Middleware*, en l'entorn del qual hi investiguen professors de la FIB.



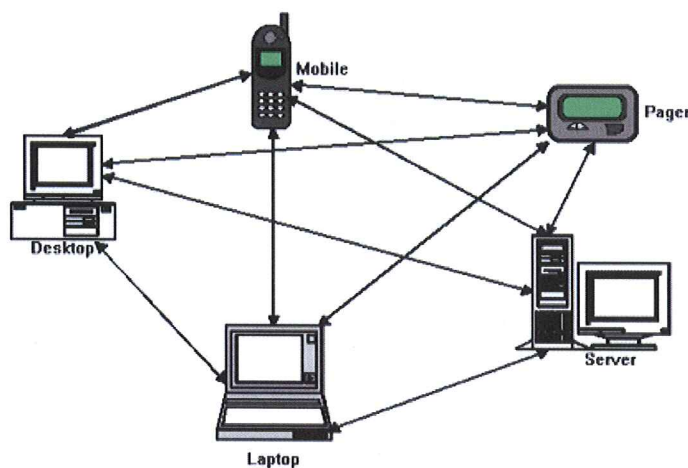
Il·lustració 6 - Arquitectura de Catallactic Grid Middleware [Joi]

3.3.2 Escenari P2P

Aquest escenari apareix amb l'objectiu de descentralitzar l'execució final dels serveis, de manera que els propis clients puguin actuar com a proveïdors de servei. D'aquesta manera s'aconsegueix reduir la utilització de recursos computacionals i d'ample de banda de xarxa en la infraestructura dels proveïdors de servei, mentre s'aprofiten aquests recursos directament dels clients.

En els sistemes inclosos en aquest escenari l'objectiu principal és la gestió dels recursos de què disposa el sistema (serveis), de manera que quan un client sol·licita un servei, aquest és enviat i executat en el mateix client, convertint-se en un nou proveïdor del servei.

Els sistemes enfocats en aquest paradigma han de tenir especial cura en l'aspecte de la seguretat i les comunicacions, ja que els nous proveïdors de servei són totalment heterogenis: poden ser sistemes amb *firewalls*, NAT (*Network Address Translation*, traducció d'adreces d'Internet, típicament entre rangs de xarxa privada i rangs de xarxa pública i viceversa), etc.

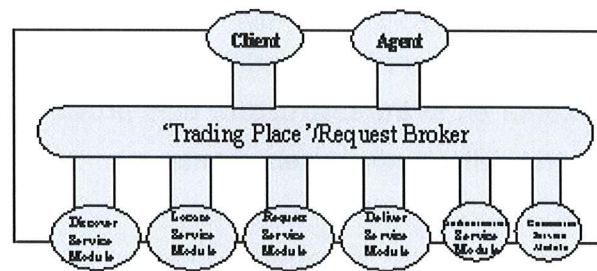


II-lustració 7 - Escenari P2P [Gra]

3.3.3 Escenari *brokering*

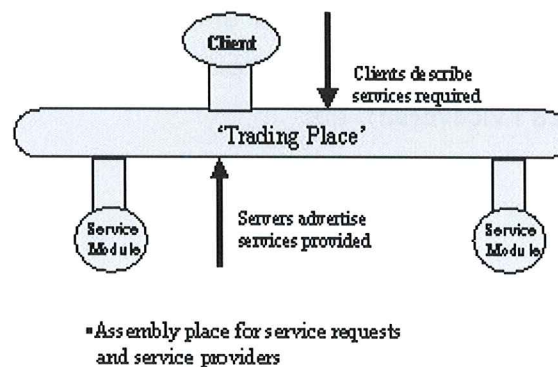
En aquest escenari, el sistema fa les funcions d'intermediari, associant les sol·licituds dels clients amb les ofertes dels proveïdors de serveis. A partir d'un conjunt de paràmetres especificats per l'usuari (cert nombre de CPUs, certa capacitat d'ample de banda, ...) el sistema presenta una llista de proveïdors

disponibles que entren dins d'un barem acceptable. Després d'un refinament per part de l'usuari, finalment es posa en contacte la sol·licitud amb el servei proveït.



Il·lustració 8 - Esquema de l'escenari brokering [Dem99]

En aquests casos, el sistema que actua d'intermediari típicament es beneficia d'una comissió, ja sigui un percentatge del que el client paga per l'ús del servei o bé una quantitat fixa com a intermediari pagada pel proveïdor.



Il·lustració 9 - Interacció dels actors a l'escenari brokering [Dem99]

3.3.4 Situació del projecte

Després de veure les anteriors catalogacions dels sistemes Grid, estem en situació d'ubicar més específicament l'àmbit del nostre projecte. El projecte a realitzar serà típicament un sistema englobat en l'àmbit dels proveïdors de serveis (escenari *brokering*). També podria, tot i que s'escapa de l'abast d'aquest projecte, incorporar mecanismes de negociació tal com es descriuen en l'escenari *marketplace*. En canvi, l'escenari P2P queda lluny de les intencions d'aquest projecte, ja que un sistema d'aquest tipus implica utilitzar el client com a proveïdor de servei (és a dir, ha d'oferir un recurs computacional), i això és precisament el que intentem evitar, ja que els dispositius mòbils presenten una limitació en la capacitat de processament.

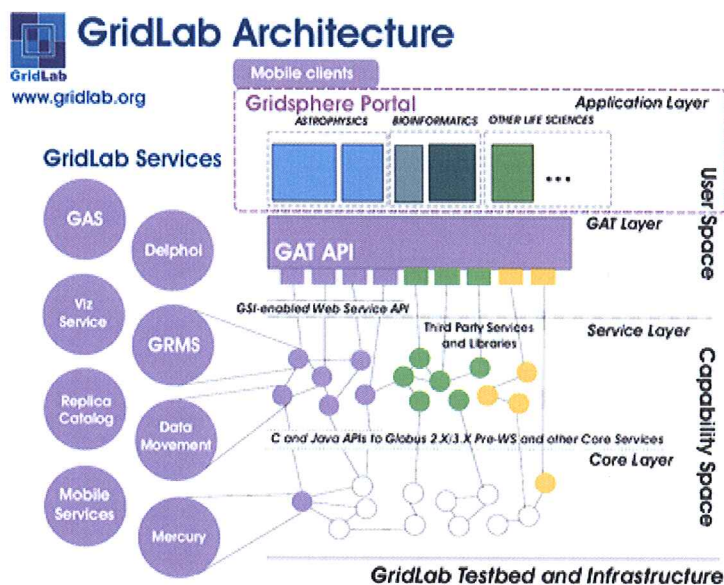
3.4 Estudi de mercat: projectes existents

En aquest apartat analitzarem algunes plataformes que existeixen actualment en el mercat i que treballen en l'àmbit del nostre projecte. L'objectiu és estudiar la "competència", per no fer una feina que ja està feta i per identificar mancances que pot omplir el nostre projecte.

3.4.1 Projecte GridLab

Al començar la recerca de documentació per al projecte, ben aviat vam tenir coneixement del projecte GridLab. GridLab és una plataforma que ofereix un conjunt d'eines per a interactuar amb serveis Grid. El seu objectiu és que un usuari final sigui capaç d'interactuar amb la Grid sense tenir coneixements de l'entorn amb què treballa.

GridLab està dissenyat en una arquitectura de mòduls que interactuen entre ells. L'objectiu final és oferir tot allò que un usuari final pot necessitar per accedir a un servei Grid: localització de recursos, monitorització, tractament de dades, seguretat, informació, accés a serveis, etc.



II-lustració 10 - Arquitectura del projecte GridLab [Gri05]

Conceptualment, GridLab és un projecte força similar al nostre. Té el mateix objectiu: desenvolupar una plataforma que permeti l'accés i la interacció amb

serveis remots, oferint una transparència total a l'usuari entre la seva sol·licitud i la recepció del resultat.

Adicionalment, GridLab és un projecte internacional de gran envergadura, finançat per importants associacions, i forma part de la *Globus Alliance*. És un projecte que porta molts anys de desenvolupament i és utilitzat per un conjunt de projectes internacionals amb resultats molt satisfactoris. Per tant, prendrem el seu projecte com un exemple.

3.4.2 Catàlegs de serveis

Un altre tipus de plataforma per a l'oferiment de serveis remots. En aquest cas, però, ens trobem davant d'un catàleg de web services orientats als desenvolupadors d'aplicacions. Per tant els seus objectius són diferents dels que té el nostre projecte.

Alguns exemples d'aquest tipus de catàlegs són l'Amazon Web Service Solutions Catalog o l'API de Google.

4 Anàlisi del projecte

En aquesta secció iniciarem la tasca d'anàlisi a nivell de projecte. La dividirem en 4 apartats que resumim breument:

Anàlisi de requeriments

Definirem els requeriments funcionals i no funcionals del nostre projecte, extrets a partir dels objectius definits i de l'estat de l'art.

Planificació del projecte

Realitzarem l'anàlisi temporal del projecte, on exposarem les diferents fases en què es divideix la seva realització, veurem quina dependència hi ha entre elles, i mostrarem tant l'estimació en temps de cadascuna com l'organització entre les tasques amb l'objectiu de finalitzar el projecte en l'espai de temps establert.

Anàlisi de mercat i especificació del prototip

En aquest últim punt analitzarem algunes tecnologies existents en el mercat per veure'n, a grans trets, les mancances que tenen i que el nostre projecte permet omplir. Així mateix, comentarem com definim

el nostre projecte com a prototip i quines diferències hi ha amb un producte final.

4.1 Anàlisi de requeriments

4.1.1 Requeriments funcionals

Els requeriments funcionals són aquells que indiquen exactament quines són les necessitats que ha cobrir el projecte. Hem definit els següents:

- El sistema ha de proporcionar un mecanisme a l'usuari per visualitzar una llista dels serveis disponibles, seleccionar-ne un, introduir les dades amb les quals haurà d'operar el servei sol·licitat i visualitzar-ne el resultat.
- Les dades es poden introduir *in situ* mitjançant quadres de text i elements similars, o bé poden trobar-se emmagatzemades en un fitxer remot. En aquest cas es demanarà a l'usuari que introdueixi la URL per permetre al sistema localitzar el fitxer i obtenir-ne les dades.
- Els serveis poden ser de 2 tipus: *online* (el resultat s'obté immediatament passats uns instants) o *batch* (el resultat triga un període de temps variable). Si el servei sol·licitat és de tipus *online*, es mostrarà el resultat un cop s'obtingui. Si el servei és de tipus *batch*, s'oferirà a l'usuari un mecanisme per consultar en quin estat es troba l'execució i, un cop hagi finalitzat, consultar el resultat.
- El sistema ha d'incorporar una aplicació de gestió que permeti a un administrador donar d'alta, modificar i eliminar serveis. En cas que es considerés oportú, aquesta aplicació també s'utilitzaria per donar d'alta, modificar i eliminar usuaris del sistema.

4.1.2 Requeriments no funcionals

A continuació descriurem el conjunt de requeriments no funcionals que, degut al tipus de projecte, als objectius i als requeriments funcionals, hauria d'assolir el projecte.

Prèviament es pot esmentar que l'objectiu final del projecte és actuar d'intermediari entre clients i proveïdors de serveis en un entorn altament dinàmic i canviant. Per tant, els requeriments no funcionals que s'intentarà assolir seran aquells relacionats amb la facilitat per adaptar el software a canvis hardware/software i amb el manteniment de la integritat de les dades des de la sol·licitud d'un servei fins la recepció del resultat. En canvi, seran menys importants aquells requeriments relacionats amb l'eficiència, la precisió o la usabilitat.

- **Flexibilitat.**

Una propietat desitjable en el nostre projecte és una alta flexibilitat, ja que amb el nostre sistema interactuen un conjunt de tecnologies que estan en constant evolució, i per tant subjectes a canvis. És important que el sistema no sigui rígid, sinó que tingui un alt grau de flexibilitat i canviabilitat per fer front a possibles modificacions.

- **Mantenibilitat.**

Per la mateixa raó que la flexibilitat és desitjable, també volem que el nostre sistema sigui fàcilment mantenible. D'aquesta manera podrem identificar ràpidament fallades en el sistema, podrem fer canvis que afectin a diferents parts del sistema, etc.

- **Portabilitat.**

Una altra característica molt desitjable és la portabilitat: no volem que hi hagi dependència de la plataforma hardware o software, sinó tot el contrari. L'àmbit en què s'engloba el projecte (Web Services, estàndards) fa que, indirectament, s'aposti per la portabilitat i la interconnectivitat entre sistemes totalment diferents.

- **Fiabilitat i integritat.**

Degut a la naturalesa distribuïda del projecte, un dels requeriments en què s'ha de prendre més cura és la fiabilitat i integritat de les dades. Des del moment en què el client sol·licita un servei fins que recull el resultat, les dades seran analitzades, tractades, dividides, s'enviaran cap a sistemes externs, se'n recuperarà un resultat parcial, el qual serà reformatejat per donar lloc al resultat final. En tot aquest camí s'ha de mantenir la coherència de les dades, i per tant el sistema haurà d'estar preparat per respondre a violacions de restriccions d'integritat (per exemple, l'usuari introdueix text

quan se li demana un número), a fallades parcials (per exemple, un Web Service no és capaç de resoldre la nostra sol·licitud), etc.

- **Seguretat.**

Finalment, no podem oblidar que ens trobem en un sistema que connecta usuaris amb recursos a través de la xarxa, i això d'alguna manera ens obliga a mantenir certs criteris de seguretat, tant a nivell d'autenticació (un usuari és qui diu ser), a nivell de validació (un proveïdor de servei és qui diu ser, i el seu servei fa el que diu que fa) com a nivell de transmissió (seguretat en la transmissió de les dades).

4.2 Planificació del projecte

4.2.1 Consideracions prèvies

A l'hora de fer la planificació del projecte és convenient fer algunes consideracions que afecten a l'organització. Aquest projecte es planteja com un prototipus d'un conjunt de tecnologies emergents, desenvolupat a partir d'una tasca d'estudi i investigació. L'objectiu a nivell de projecte no és aconseguir un producte, sinó obtenir una plataforma el suficientment àmplia i funcional com per ser utilitzada i ampliada per a l'estudi de les tecnologies orientades a serveis.

Degut a l'enfocament que s'ha donat al projecte, no es tindran en compte els aspectes econòmics del projecte, com podria ser el cost aproximat de la producció de la plataforma o els recursos necessaris.

4.2.2 Etapes del projecte

A continuació es descriu la planificació global que s'ha estimat per a la realització del projecte a nivell temporal. Per fer-ho s'ha dividit la realització del projecte en un conjunt de tasques, les quals s'han d'organitzar adequadament en el temps per a poder fer una estimació de la durada del projecte.

A continuació presentem les tasques que han format el projecte, així com una breu descripció de cadascuna.

- **Documentació teòrica.** Tasca que dóna inici al projecte. Primer de tot hi ha hagut un procés de documentació sobre l'àmbit del projecte: quin és l'estat actual de les tecnologies orientades a servei, quins productes existeixen, quines línies d'investigació hi ha, etc.
- **Selecció de la plataforma tecnològica.** A partir de la fase de documentació s'ha escollit la plataforma sobre la qual es desenvoluparà el projecte. La plataforma escollida ha estat J2EE + Web Services.
- **Documentació tècnica.** Un cop decidida la base del projecte hi ha hagut una fase de documentació tècnica per conèixer com funciona la plataforma escollida.
- **Instal·lació de la plataforma de desenvolupament.** Pas previ necessari abans de poder començar la implementació. Un cop s'ha seleccionat la plataforma tecnològica s'ha instal·lat tot el software necessari per a poder iniciar la implementació: Java2 SDK, Apache Tomcat, JDom i Apache Axis.
- **Instal·lació d'una plataforma de proves.** Per poder provar la implementació del projecte s'ha preparat un entorn que simula la situació en què es trobaria el projecte en el món real. Per això s'ha preparat un altre sistema amb Apache Tomcat, Apache Axis i GT4 (Globus Toolkit 4).
- **Especificació.** Fase d'anàlisi del projecte per obtenir-ne el model conceptual.
- **Disseny.** Disseny del software a partir de l'especificació, per obtenir-ne l'arquitectura, el diagrama de classes de disseny i els contractes de les operacions.
- **Implementació.** Un cop finalitzada la fase de disseny, la documentació tècnica i la preparació de l'entorn de desenvolupament es pot iniciar la implementació del prototipus.
- **Proves.** Quan ja s'ha finalitzat la implementació del prototipus es realitzen un conjunt de proves per verificar la correctesa del programa.

- **Generació de la documentació.** Tot i que es troba en últim lloc, la generació de la documentació (de la qual forma part aquesta memòria) s'inicia amb la primera fase del projecte i va avançant amb el projecte, però és en la finalització quan tenim tota la documentació necessària per poder documentar completament el projecte.

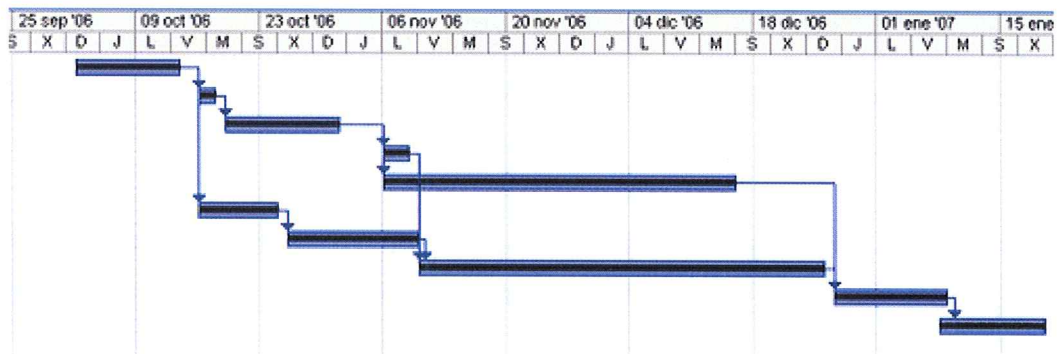
4.2.3 Planificació i diagrama de Gantt

Data d'inici del projecte: 01/03/2006

Data límit d'entrega: 29/06/2006

Taula 1 - Fases del desenvolupament del projecte

| Nombre | Duración | Comienzo | Fin | Predecessoras |
|--|----------|--------------|--------------|---------------|
| Documentació teòrica | 10 días | mié 01/03/06 | mar 14/03/06 | |
| Selecció de la plataforma tecnològica | 2 días | mié 15/03/06 | jue 16/03/06 | 1 |
| Documentació tècnica | 9 días | vie 17/03/06 | mié 29/03/06 | 2 |
| Instal·lació de la plataforma de desenvolupament | 3 días | jue 30/03/06 | lun 03/04/06 | 3 |
| Instal·lació de la plataforma de proves | 30 días | jue 30/03/06 | mié 10/05/06 | 3 |
| Especificació | 5 días | mar 14/03/06 | mar 21/03/06 | 1 |
| Disseny | 9 días | mié 22/03/06 | lun 03/04/06 | 6 |
| Implementació | 35 días | mar 04/04/06 | lun 22/05/06 | 4;7 |
| Proves | 5 días | mar 23/05/06 | lun 29/05/06 | 5;8 |
| Generació de la documentació | 7 días | mar 30/05/06 | mié 07/06/06 | 9 |



Il·lustració 11 - Planificació del projecte: diagrama de Gantt

4.2.4 Anàlisi de la planificació. El camí crític.

Com es pot observar en el diagrama de Gantt hi ha 2 seqüències de tasques que es realitzen pràcticament en paral·lel: d'una banda la documentació sobre les tecnologies i la preparació de l'entorn de treball i de l'altra l'especificació, disseny i implementació.

El camí crític (seqüència de tasques que no poden ser reordenades i que provoquen la màxima expansió del projecte al llarg del temps) ve donat per la seqüència de les tasques *especificació, disseny, implementació, prova i documentació*, principalment a causa de la gran durada de la implementació.

Aixa doncs, com a conclusió, veiem que en les tasques de preparació de l'entorn i de documentació tenim més marge, i si es produeix alguna desviació en el temps possiblement no afectarà a la durada del projecte. En canvi amb les tasques que formen part del camí crític haurem d'anar amb molt de compte de mantenir-nos dins de la planificació establerta, ja que qualsevol desviació afectarà directament a la durada del projecte, i si la desviació fos important podríem excedir la data límit.

4.2.5 Anàlisi de la planificació real

Si observem les dates amb què hem treballat al fer la planificació del projecte, podem adonar-nos que no s'han complert les dates fixades per un marge molt important: gairebé 1 any de retard.

L'explicació d'aquesta demora es troba en un cúmul de circumstàncies externes al projecte, tant professionals com personals, que han afectat de manera significativa en la dedicació que s'ha pogut tenir en el mateix.

Així doncs, es demostra que és molt important, a l'hora de fer una planificació, tenir en compte factors externs al projecte que poden afectar en el seu desenvolupament. D'aquesta manera s'aconseguirà una planificació molt més acurada a la situació real dels recursos (en aquest cas, temps de dedicació).

4.3 Anàlisi de mercat i especificació del prototip

En les seccions anteriors hem especificat els objectius del nostre projecte i hem fet un balanç de les tecnologies existents a l'actualitat. En aquest apartat acotarem definitivament l'àmbit del projecte, a partir dels requeriments, els objectius, i tenint en compte l'estat de l'art de la tecnologia.

Com ja s'ha comentat anteriorment, l'objectiu d'aquest projecte és la realització d'un prototip. En comptes d'un sistema que realitzi una tasca de forma completa, es desitja un prototip que englobi moltes tasques, tot i que les realitzi de manera

superficial. Tenint això en compte, és evident que la destinació d'aquest projecte, un cop finalitzat, no és el mercat ni l'entorn empresarial, sinó l'àmbit educatiu universitari i d'investigació.

A partir d'aquesta premissa, no té sentit realitzar un estudi econòmic ni un estudi de viabilitat, ja que en la realització d'aquest projecte no entren en joc els mecanismes típics dels projectes empresarials: beneficis, cost, mercat.

D'altra banda, entre tots els tipus de plataformes que s'han vist en el capítol anterior, escollirem 2 que siguin prou representatives a l'hora d'implementar-ne la seva gestió i incorporar-les a l'arquitectura del projecte. Concretament:

- Web services
- Grid services, mitjançant la plataforma Globus/OGSA

Aquesta elecció s'ha realitzat en base a les següents raons:

- Les 2 plataformes utilitzen mecanismes de sol·licitud basats en web services.
- En quant als grid services, d'entre totes les plataformes existents, el Globus Toolkit és considerat un estàndard *de facto*, donat que ja té 6 anys d'experiència i refinament del seu codi. En base a Globus Toolkit es va crear, posteriorment, la Open Grid Service Architecture (OGSA), la qual modernitza i estén les funcionalitats de Globus Toolkit. Entre aquestes noves funcionalitats s'inclou la utilització de Web Services com a sistema de sol·licitud.

Com a nota final per acabar aquesta secció, comentar que s'han escollit 4 exemples o serveis públics als quals es recorrerà per il·lustrar diferents aspectes del projecte al llarg d'aquesta documentació. Cadascun d'aquests 4 models modelitza un tipus de problema diferent, de manera que amb aquests serveis podrem il·lustrar el comportament del projecte en diferents situacions en què es podria trobar en el món real.

Els serveis model són:

- o Suma
- o Doble suma
- o Mitjana
- o Producte de matrius