TFC J2EE: PROVEÏDOR DE SEGURETAT EN UN ENTORN DISTRIBUIT

Estudiant: Joaquim Roca Vergés

Titulació: ETIG

Consultor: Joan Vicent Orenga Serisuelo

Data Lliurament: 25 de Juny de 2008

Universitat Oberta de Catalunya

Dedicat a la memòria del meu pare i a la meva estimada Carme

Agraïments:

A la meva mare, que no ha parat d'alegrar-se i animar-me des que vaig començar a estudiar i a qui li faltava temps per explicar a tothom , orgullosa, qualsevol avanç que feia.

A les meves tietes Enriqueta , Josefina i Roser per creure en els miracles.

A la meva cosina Roser per creure sempre en mi.

A la Sra. Treig i al Sr. Velasco per que en set anys no m'han fet ni un retret per el molt de temps que passava estudiant.

A la Ylenia i al Lander, que s'han portat bé mentre jo era tancat al despatx.

A tots els amics de Global Hard, Ibermática i Bea Systems que m'han ajudat

A tota la comunitat UOC i a tots els consultors amb els que he treballat, els bons i els no tant bons, gracies a tots per haver compartit els seus coneixements i per la seva paciència.

Resum del Treball Final Carrera.

Aquest **TFC-J2EE**, està basat en l'anàlisi, disseny i implementació d'un proveïdor de seguretat en un entorn distribuït.

Aquest entorn distribuït no existia i s'ha hagut d'analitzar, dissenyar i implementar a l'hora que construíem el **proveïdor de seguretat**. Ha consistit en dues aplicacions independents, una consumidora i l'altre servidora que es comunicaven usant el **patró Façana** (una façana amigable amaga el negoci i la complexitat de l'aplicació a la que es vol accedir).

Vistes ambdues aplicacions com un tot, s'ha aplicat el **patró MVC** de manera distribuïda: la aplicació consumidora compté la vista i la aplicació servidora compté el model al que s'accedeix mitjançant l'esmentat patró façana.

A aquestes aplicacions les hem anomenat aplicacions de suport i han consistit en un framework de gestió integral, composat per varies aplicacions, dues de les quals tenien la opció d'accedir a les dades bancàries que gestiona l'altre aplicació.

La implementació de les aplicacions de suport ha consistit en això: simulació d'un accés a un **LDAP** conjuntament amb la simulació del accés a una aplicació securitzada(aplicació amb accés a les dades bancàries) mitjançant una plana inicial, i un manteniment de comptes bancaris. Aquest manteniment , que tenia la part de la vista en la aplicació de Gestió Integral, està realitzat mitjançant peticions a un altre aplicació que es qui serveix les dades. La vista s'ha realitzat partir de la utilització del bastiment **Struts**, el qual s'ha integrat amb el *framework* **Tiles**

El negoci es gestiona vers un **EJB stateless** d'entrada a la aplicació servidora. Totes les operacions de negoci de l'aplicació servidora estan gestionades directament per aquest EJB. Defineix la transaccionalitat i API d'accés a les Dades Bancàries. Els paràmetres d'entrada/sortida són **XMLBeans**. Hem usat el **patró Factoria** per centralitzar tant el lloc on es creen els EJB's com el lloc on es creen les classes desenvolupades segons el **patró DAO**, classes que invoquen les funcions encapsulades que accedeixen a la Base de dades, amb **Hibernate** com a mapejador d'objectes relacional.

Tot i ser un projecte si se'm permet, bastant complex, he buscat en tot moment la simplicitat, la modularitat i la reutilització. El model de dades l'hem encapsulat, de manera que podria ser utilitzat per qualsevol aplicatiu que accedís a les dades bancàries de la nostra aplicació servidora. El model de **missatgeria XML** també ha estat encapsulat i es vàlid per qualsevol LDAP: es crea un bean amb informació d'usuari que tots els LDAPS contenen, com es la de codi d'usuari , perfil o role i grup al que pertany. Per al processament dels formats de missatge XML s'ha escollit la tecnologia de referència **Apache XMLBeans** .XMLBeans permet un mapeig directe entre XML i JavaBeans, la qual cosa simplifica el procés de programació.

Sobre aquestes dues aplicacions, s'ha posat una capa de seguretat que autentica als usuaris i els autoritza a realitzar accions entre dominis.

El projecte usa l'autenticació unificada (**Single Sign-On**) donat que és un procés molt útil per a les aplicacions perquè permet que un usuari només s'identifiqui un cop.

Ell proveïdor d'autenticació responsable de validar l'identitat de l'usuari s'ajusta a les especificacions de seguretat que delimita l'estàndard **JAAS**

L'autorització es basa en estàndards J2EE simplificant molt el procés per a validacions automàtiques. El model més acceptat a J2EE fa servir un sistema de regles d'autorització basat en rols (**role-base authorization**). Aquest sistema d'autorització automàtica passa per utilitzar els descriptors de desplegament J2EE per indicar les regles d'autorització segons diferents rols. **L'autorització automàtica** (**coarse-grained authorization**) s'aplica als **components J2EE**

Els rols són calculats dinàmicament pel proveïdor de seguretat de mapeig de rols del projecte(**Role Mapping Provider).** Les regles de càlcul de rols a partir de la informació de l'usuari i grups disponible al context de seguretat es simulen a la plana inicial esmentada mes dalt en aquest document.

El model de seguretat desenvolupat també permetria d'incorporar validacions d'autorització amb lògica de negoci i amb molt més grau de detall (**fine-grained authorization**) donat que els sistemes estàndard de seguretat J2EE proporcionen un API que permet a les aplicacions executar crides a per autoritzar segons dades dinàmiques que es calculen en temps d'execució i que no es poden incloure en un descriptor de desplegament.

1. Índex.

2. INTRODUCCIÓ.

2.1 Justificació del TFC: punt de partida i aportació.

Qualsevol empresa pot basar el seu model de seguretat en un servei LDAP, un servei Web o qualsevol tipus de servei desenvolupat per aquest us .

Les aplicacions d'interacció humana estan molt integrades amb els sistemes de seguretat estàndard perquè han de conèixer informació sobre els usuaris que hi interaccionen.

Aquest proveïdor de seguretat es una encapsulació que compleix JAAS, arquitectura estàndard J2EE de gestió de seguretat. Tota la feina de gestió d'usuaris, grups i rols es gestiona des de el mòdul que hagi escollit la empresa i no es motiu d'aquest projecte. Aquest projecte tracta de la propagació de seguretat entre dos dominis.

El proveïdor de seguretat es compatible amb qualsevol model de seguretat que s'estigui usant i no afecta pas a altres mòduls que no l'usin: si una aplicació té n components, només es securitzaran aquells que es vulguin, i la seva execució es transparent a les aplicacions que no estan securitzades

Aquest TFC parteix doncs, d'aquesta necessitat de desenvolupar un proveïdor de seguretat entre aplicacions diferents, residents en diferents dominis, integrant qualsevol model de seguretat en el que es basi una aplicació J2EE resident en un entorn WebLogic Server amb un altre entorn WebLogic Server, entorn servidor d'aplicacions J2EE.

Com a qualsevol TFC, el objectiu principal d'aquest és mostrar l'assoliment de l'aprenentatge que s'ha dut a terme al llarg dels estudis d' Enginyeria Tècnica en Informàtica, analitzant un problema complex de tipus pràctic transformant-lo en un projecte informàtic, elaborant un pla de treball i treballant els aspectes formals en el desenvolupament (orientat a objectes en aquest cas), i finalment presentant una solució fiable i eficient.

El punt de partida són doncs els coneixements adquirits durant la carrera. A més , donat que tinc la sort de treballar amb J2EE a la feina, tenia certs coneixements extres com el desenvolupament a partir de patrons de disseny aplicats a aplicacions d'empresa, el bastiment Struts i el Framework Tiles. També havia treballat una mica amb EJB's i Hibernate.

L'aportació d'aquest TFC aplicat a la tecnologia J2EE, com a arquitectura per donar resposta a les necessitats anteriorment exposades, ha estat combinar tots els coneixements teòrics i pràctics amb la fi de crear una implementació distribuïda del patró MVC entre diferents aplicacions i proveir seguretat entre aquestes aplicacions mitjançant una encapsulació JAAS.

2.2 Objectius del TFC.

Els objectius del TFC Proveïdor de Seguretat (PS) en un entorn distribuït han estat:

• **Dissenyar** el proveïdor de seguretat per tal de resoldre els requisits de seguretat de la aplicació servidora

• Fer que **La aplicació consumidora ofereixi al PS les dades sobre usuaris i grups** per tal que el proveïdor de seguretat treballi.

• **Implementar els proveïdors de** seguretat. Les implementacions accedeixen a serveis i informació gestionada pel mòdul de la aplicació consumidora que accedeix al model de seguretat propi (LDAP o altres)

• **Desplegar i fer proves de qualitat** del proveïdor de seguretat sobre les aplicacions.

2.3 Enfocament i mètode seguit.

Com tot projecte de desenvolupament de programari, el procés de construcció s'ha dividit en diverses fases ben diferenciades i amb objectius molt diferents. Les fases utilitzades han estat les habituals: Presa de requisits -que en el nostre cas nosaltres hem fet de clients (hem ideat el projecte)-, anàlisi d'aquests requisits, disseny – etapa , desenvolupament, proves, desplegament i documentació.

Donat el tarannà del projecte, en que el client(nosaltres) sabia exactament el que volia, es a dir que teníem tots i cadascun dels requisits que havia de complir el programari hem seguit el model clàssic de cicle de vida en cascada.
L'especificació del sistema que ha donat el client ha estat fiable al 100% pel que fa a les funcions, que han estat descrites fil per randa. El cost i la durada del projecte s'han calculat sobre una base molt sòlida i tenen amb un marge d'error molt petit. L'únic marge d'error possible era el que donava la implementació: errors de codi i desconeixement tècnic.

No obstant, si que hi ha haguts petites variacions en l'ordre de la execució de les tasques i lleugers canvis tal com s'explica en l'apartat de planificació del projecte .

Finalment dir que la implementació s'ha fet en paral·lel: escrivíem el codi tant de les aplicacions de suport com del proveïdor de seguretat; el fet de poder canviar de projecte sempre que un estava cansat o encallat en un punt, es quelcom que volem recalcar com a molt positiu donat que la sensació de "pèrdua de temps" desapareix i mentre estàs en el segon projecte, moltes vegades t'asserenes i veus clar on radica el problema del primer projecte.

2.4 Planificació del projecte.

L'abast de les especificacions inicials del projecte s'han complert.
No hi ha hagut desajustaments greus. Val a dir per això que si hi ha hagut variacions en l'etapa de implementació, concretament en el començament i durada de les tasques i en les seves dependencies . Finalment , dins d'aquesta mateixa fase, ha aparegut una tasca nova, que no havíem valorat, i dues tasques han desaparegut , donat que ja s'han anat realitzant paulatinament juntament amb les altres.

A continuació adjuntem les dues planificacions i diagrames Gantt inicials i finals i comentem aquets canvis

Planificació inicial

Id	Nombre de tarea	Duración	Comienzo	Fin
1	**PS - Proveidor de Seguretat**	**120 días**	**mié 27/02/08**	**mié 25/06/08**
2	Gestió de projecte	120 días	mié 27/02/08	mié 25/06/08
3	**Planificacio**	**22 días**	**mié 27/02/08**	**mié 19/03/08**
4	Redacció document de Planificac	22 días	mié 27/02/08	mié 19/03/08
5	PS-Fita Lliurament PAC1	0 días	mié 19/03/08	mié 19/03/08
6	**Analisis de requisits**	**10 días**	**jue 20/03/08**	**sáb 29/03/08**
7	Analisi Requisits PS	10 días	jue 20/03/08	sáb 29/03/08
8	**Disseny**	**16 días**	**dom 30/03/08**	**lun 14/04/08**
9	Disseny Tecnic PS	16 días	dom 30/03/08	lun 14/04/08
10	PS-Fita Lliurament PAC2	0 días	lun 14/04/08	lun 14/04/08
11	**Implementació**	**35 días**	**mar 15/04/08**	**lun 19/05/08**
12	Plantilles HTML	4 días	mar 15/04/08	vie 18/04/08
13	Planes JSP	1 día	sáb 19/04/08	sáb 19/04/08
14	Api Accés a Base de dades	3 días	dom 20/04/08	mar 22/04/08
15	EJB execució API	1 día	vie 25/04/08	vie 25/04/08
16	Test EJB	1 día	sáb 26/04/08	sáb 26/04/08
17	Construcció PS	15 días	dom 27/04/08	dom 11/05/08
18	Client PS	2 días	lun 12/05/08	mar 13/05/08
19	Test PS	2 días	mié 14/05/08	jue 15/05/08
20	Integracio JSP/EJB/PS	2 días	vie 16/05/08	sáb 17/05/08
21	Desplegament en dos dominis	2 días	dom 18/05/08	lun 19/05/08
22	PS-Fita Lliurament PAC3	0 días	lun 19/05/08	lun 19/05/08
23	**Documentacio**	**37 días**	**mar 20/05/08**	**mié 25/06/08**
24	Redacció del document	37 días	mar 20/05/08	mié 25/06/08
25	PS-Fita Lliurament FINAL	0 días	mié 25/06/08	mié 25/06/08

Planificació final

	Nombre de tarea	Duración	Comienzo	Fin
1	**PS-Proveidor de Seguretat**	**120 d**	**mié 27/02/08**	**mié 25/06/08**
2	Gestió de projecte	120 d	mié 27/02/08	mié 25/06/08
3	**Planificacio**	**22 d**	**mié 27/02/08**	**mié 19/03/08**
4	Redacció document de planifica	22 d	mié 27/02/08	mié 19/03/08
5	PS-Fita Lliurament PAC1	0 d	mié 19/03/08	mié 19/03/08
6	**Analisis de requisits**	**10 d**	**jue 20/03/08**	**sáb 29/03/08**
7	Analisi Requisits PS	10 d	jue 20/03/08	sáb 29/03/08
8	**Disseny**	**16 d**	**dom 30/03/08**	**lun 14/04/08**
9	Disseny Tecnic PS	16 d	dom 30/03/08	lun 14/04/08
10	PS-Fita Lliurament PAC2	0 d	lun 14/04/08	lun 14/04/08
11	**Implementació**	**35 d**	**mar 15/04/08**	**lun 19/05/08**
12	Plantilles HTML	4 d	mar 15/04/08	vie 18/04/08
13	Planes JSP	1 d	sáb 19/04/08	sáb 19/04/08
14	**Classes Del Controlador**	**2 d**	dom 20/04/08	lun 21/04/08
15	**Api Accés a Base de dades**	**5 d**	dom 20/04/08	jue 24/04/08
16	EJB execució API	1 d	vie 25/04/08	vie 25/04/08
17	Test EJB	1 d	sáb 26/04/08	sáb 26/04/08
18	**Construccio PS**	**21 d**	vie 18/04/08	jue 08/05/08
19	**Client PS**	**7 d**	vie 09/05/08	jue 15/05/08
20	**Test PS**	**4 d**	vie 16/05/08	lun 19/05/08
21	**Integracio JSP/EJB/PS**	**0 d**	lun 19/05/08	lun 19/05/08
22	**Desplegament en dos domi**	**0 h**	lun 19/05/08	lun 19/05/08
23	PS-Fita Lliurament PAC3	0 d	lun 19/05/08	lun 19/05/08
24	**Documentacio**	**37 d**	**mar 20/05/08**	**mié 25/06/08**
25	Redacció del document	37 d	mar 20/05/08	mié 25/06/08
26	PS-Fita Lliurament Final	0 d	mié 25/06/08	mié 25/06/08

Hem remarcat en negreta els canvis:

- Nova tasca **Classes del Controlador**. Es tracta de les classes Action i Form del MVC, que inicialment ens varem descuidar de valorar.

- **Increment** del **Api Accés a Base de dades** en dues jornades, motivat per un problema de desconeixement tècnic d'Hibernate

- Les classes del Controlador es **desenvolupen al mateix temps** que l'Api d'accés a Base de Dades.

- **Increment** de la durada de la **construcció del PS** en sis dies, degut a la seva complexitat. També ens podem fixar que comença abans del previst, fet motivat perquè ens varem donar compte que podíem desenvolupar les aplicacions de suport i el PS en paral·lel. Aquest fet es veu més clar en el gràfic de Gantt final, on podem observar separades les dues implementacions (abans i després de la Construcció PS) .

Podem observar doncs que no es va tenir en compte en la valoració inicial la possibilitat de poder treballar en distintes tasques en paral·lel.

- **Increment** de la durada del **Client PS** en cinc dies i del **Test PS** en quatre, pel mateix motiu que l'anterior. El increment proporcional desmesurat del Client ens indica clarament que estava incorrectament valorat.
  Al estar vinculat Client PS a la construcció del PS, també comença abans, al igual que el Test PS que també comença abans.

- Les tasques **Integració JSP/EJB/PS** i **Desplegament en dos dominis** desapareixen. Les hem conservat en la planificació final per facilitar la comparació entre planificacions. Podem observar que passen de tenir una durada de dos jornades a tenir una durada de cap (0) jornades. El motiu es que s'han desenvolupat a l'hora que es desenvolupaven les altres tasques. El desplegament en dos dominis es pràcticament automàtic quan es fan les proves de test del proveïdor de seguretat No obstant haver desaparegut, no pensem que hagi estat una mala idea de tenir-les en compte en la planificació inicial.

Diagrama de Gantt inicial

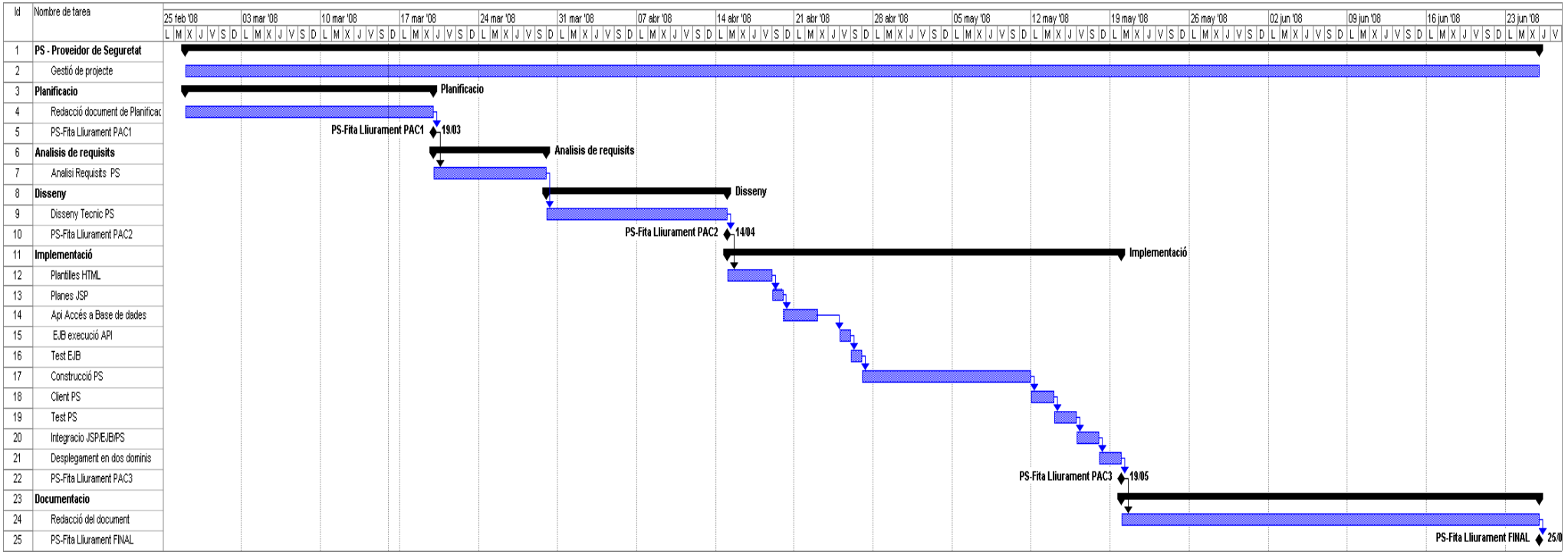
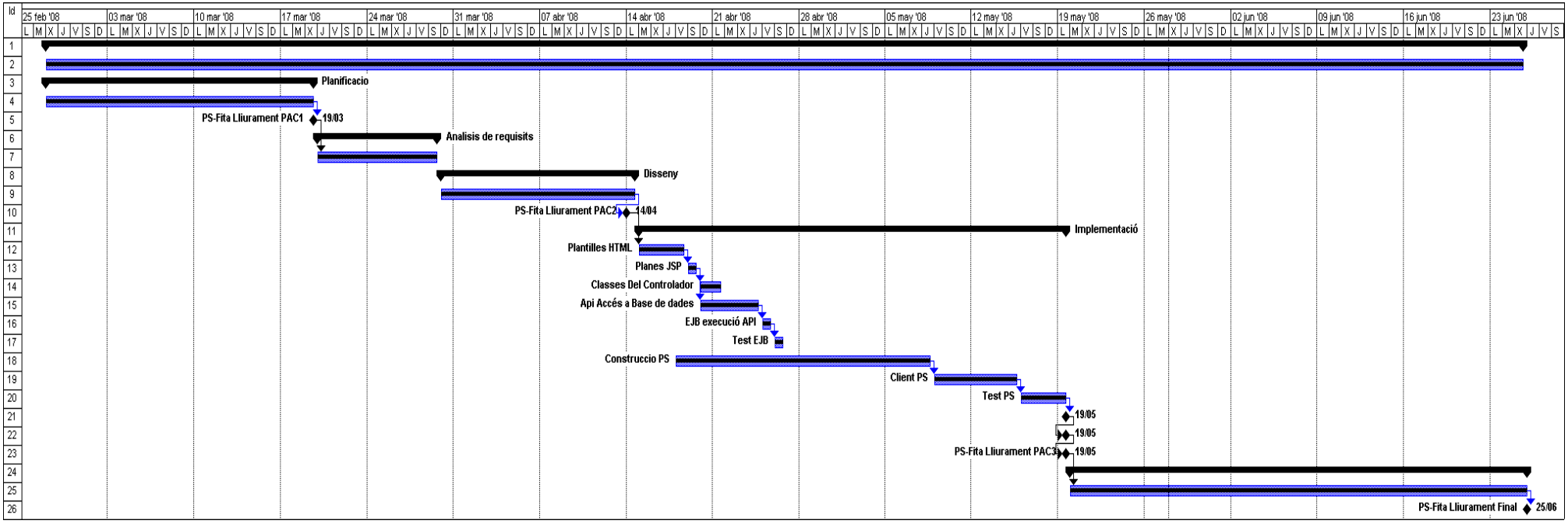


Diagrama de Gantt final



2.5 Productes obtinguts.

Els productes obtingut en la implementació d'aquest projecte han estat dues aplicacions d'empresa J2EE i una peça de seguretat (proveïdor de seguretat) que transforma la informació de seguretat proporcionada pel sistema GEUS(o la de qualsevol LDAP) en l'estàndard de seguretat utilitzat a la plataforma J2EE. Aquest mecanisme s'utilitza per a la protecció de recursos a les aplicacions de Web, components EJB i processos de negoci de BEA WebLogic.

També s'ha entregat una aplicació de test, per tal de facilitar la implantació del producte així com per a validar-ne la seva correctesa.

Aplicacions de suport

- Desplegables d'ambdós projectes en dos ears descomprimits en carpetes , i les llibreries necessàries per el seu funcionament.
- Codi font de ambdós projectes , i les llibreries necessàries per el seu funcionament.
- Documentació detallada i normalitzada en format javadoc de ambdós projectes
- Manual d'instal·lació , que inclou el script de creació de la taula i la seqüència necessàries per mantenir els comptes bancaris i les instruccions de navegació per la aplicació.

Proveïdor de seguretat

- Els dos jars que conformen el projecte, i que s'han de desplegar a nivell de Classpath l'un i de domini l'altre
- El jar del client del proveïdor de seguretat
- Aplicació de Test en un ear. Amb el jar del client inserit dins.
- Codi Font dels tres projectes:
  - Proveïdor de seguretat
  - Client del proveïdor de seguretat
  - Test del proveïdor de seguretat
- Documentació detallada i normalitzada en format javadoc del tres projectes
- Manual d'instal·lació , que inclou la explicació de la estructura organitzativa , la estructura dels packages i codi del proveïdor de seguretat i el seu client i la relació de components clau del proveïdor de seguretat. Finalment s'inclouen les instruccions per la execució de la aplicació de test.

2.6 Descripció de la resta de capítols de la memòria.

En la resta de capítols de la memòria es comenten la funcionalitat, disseny i implementació del TFC.

3. FUNCIONALITAT

3.1 Introducció

Un proveïdor de seguretat és un subsistema de BEA WebLogic Server de baix nivell i del qual depenen molts altres subsistemes. La solució de proveïdors de seguretat que presentem amb delegació de funcions a GEUS(mòdul fictici que representa la capa entre qualsevol aplicació i un LDAP) ha d'assegurar una màxima disponibilitat perquè les aplicacions que s'hi executen funcionin amb estabilitat.

El nostre projecte analitza aquest escenari i planteja la solució mitjançant la descripció general del sistema i dels casos d'ús principals, que son la autenticació i la autorització.

Per utilitzar l'autorització estàndard J2EE en aplicacions d'empresa J2EE és necessari una peça de seguretat (proveïdor de seguretat) que transformi la informació de seguretat proporcionada pel sistema GEUS en l'estàndard de seguretat utilitzat a la plataforma J2EE. Aquest mecanisme serà utilitzat per a la protecció de recursos a les aplicacions de Web i els components EJB .

Actualment per tal de que una aplicació es comuniqui amb un altre aplicació, estigui o no en el mateix domini, es necessari que es dupliquin(o tripliquin si la comunicació es amb tres aplicacions etc.) els accessos al LDAP, donat que una primera validació no es valida per la segona aplicació. També s'han buscat solucions de consulta de identitat constants durant la navegació.

Aquest projecte tracta del disseny i implementació d'un proveïdor de seguretat per a dos dominis BEA WebLogic Server segons les directrius dels estàndards de seguretat J2EE que comunicarà la seguretat a les aplicacions d'un domini a les aplicacions del altre domini i la integrarà de forma que no s'hagi de duplicar accessos a LDAPS ni gestions d'autenticació i comprovació d'identitats durant la navegació.

El proveïdor de seguretat és una peça que encapsula les dades proporciones per GEUS, és a dir, la gestió d'aquesta informació és responsabilitat única de GEUS i no del proveïdor de seguretat.
EL nous proveïdors de seguretat són compatibles amb el model actual de seguretat GEUS i no afecten pas els mòduls ja existents de GestioIntegral.cat.
Les primeres aplicacions que han de funcionar amb el proveïdor de seguretat són FACT i CONTA. Donat que son les dues que accedeixen a les dades de l'aplicació BANC .

Nota: Els mòduls on establirem el proveïdor de seguretat son mòduls que han de existir. Ara bé per tal de desenvolupar el proveïdor de seguretat, donat que no disposem d'aquets mòduls, els haurem de construir mínimament.
Necessitem simular la aplicació consumidora a la que hem anomenat GestioIntegral.cat, i la aplicació servidora, a la que hem anomenat EdadesBanc. No era objecte del nostre projecte aquestes aplicacions. No obstant, en el capítol de disseny si que s'inclou un apartat on es descriu com hem fet la construcció d'aquets mòduls i quines tecnologies hem usat per cadascun d'ells.

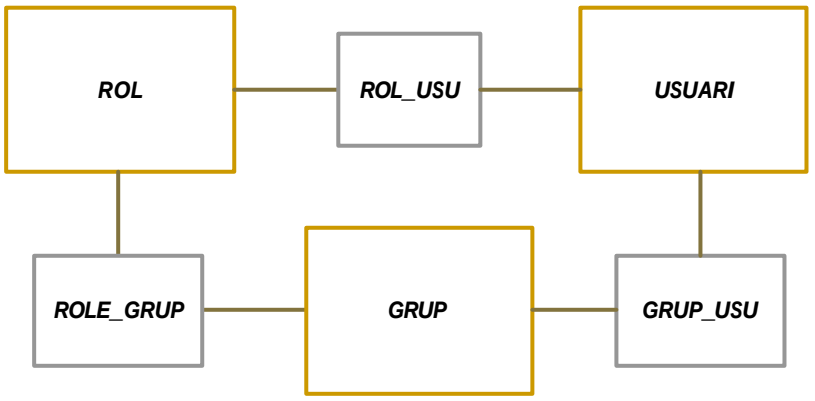
3.2 Relació d'usuaris implicats

- Usuaris que usin el mòdul de Facturació(FACT)
- Usuaris que usin el mòdul de Comptabilitat(CONTA)

3.3 Abast del sistema

3.3.1 Descripció

El security provider consisteix en una peça d'infraestructura responsable de l'autentificació i autorització d'usuaris per a controlar l'accés a recursos del tipus URL's d'aplicacions de web i mètodes d'EJB's.
El funcionament d'aquest esquema està basat en tres entitats que són: usuaris, roles i grups. Un usuari pot estar associat amb un role i/o grup. A més a més, un usuari pot estar associat a un grup. Per tant, els rols d'un usuari són aquells que té assignats directament més aquells que es troben assignats als grups als que pertany l'usuari.

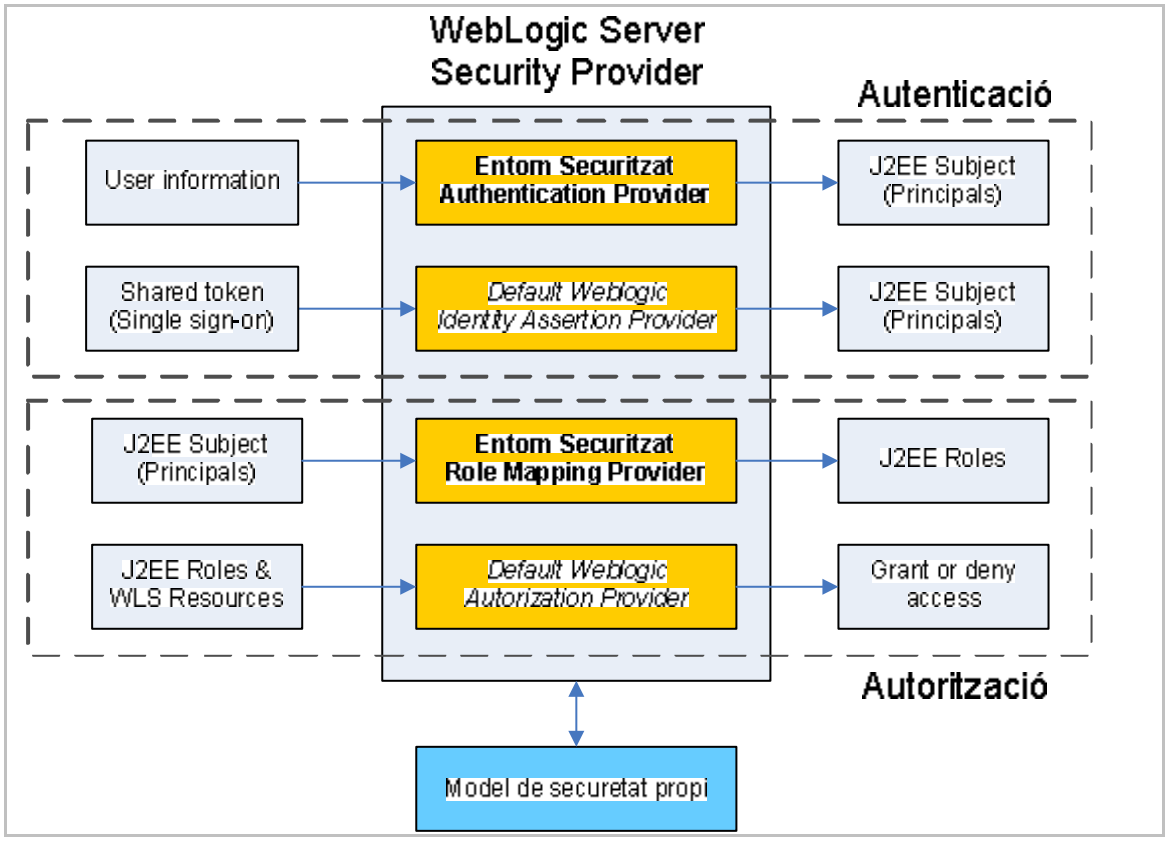


Esquema de seguretat

La informació de l'usuari I els seus grups es construeix en el moment que l'usuari fa el login contra el sistema. L'usuari més els seus grups constitueixen els principals de l'usuari. Els principals constitueixen la identitat de l'usuari anomenada subject de l'usuari.

Subject de l'usuari

Els roles s'utilitzen en els descriptors de les aplicacions J2EE per tal de protegir els recursos. L'usuari i els grups al que pertany constitueixen els principals de l'usuari que formen el que s'anomena el subject de l'usuari. El proveïdor de seguretat utilitza aquesta informació per tal d'autoritzar o denegar l'accés al recurs sol·licitat.



Proveïdor de seguretat WLS

Com podem veure a la imatge anterior, el proveïdor de seguretat està format diferents components dos dels quals s'han de desenvolupar per a les necessitats específiques del sistema com són:

- El proveïdor d'autentificació responsable de validar la identitat de l'usuari. Aquest mòdul es desenvolupa conforme a l'estàndard JAAS.
- El proveïdor de mapeig de roles responsable de transmetre els roles de l'usuari cap al proveïdor d'autorització de WebLogic perquè decideixi.

3.4 Model lògic del sistema

3.4.1 Model de casos d'ús

Els casos d'us del projecte son dos:

- **Autenticació**

Cas d'ús que dona resposta a la pregunta Qui ets.

- **Autorització**

Cas d'ús que dona resposta a la pregunta Que pots fer.

El proveïdor de seguretat ha d'implementar dos casos d'ús: l'autentificació de l'usuari I l'autorització d'execució o accés a un recurs ubicat en una aplicació Web o component EJB.

3.4.1.a Descripció dels actors

Actor Usuari del sistema

Es el principal usuari del sistema, es ell qui activa el procés de creació del context de seguretat quan es loga a la aplicació

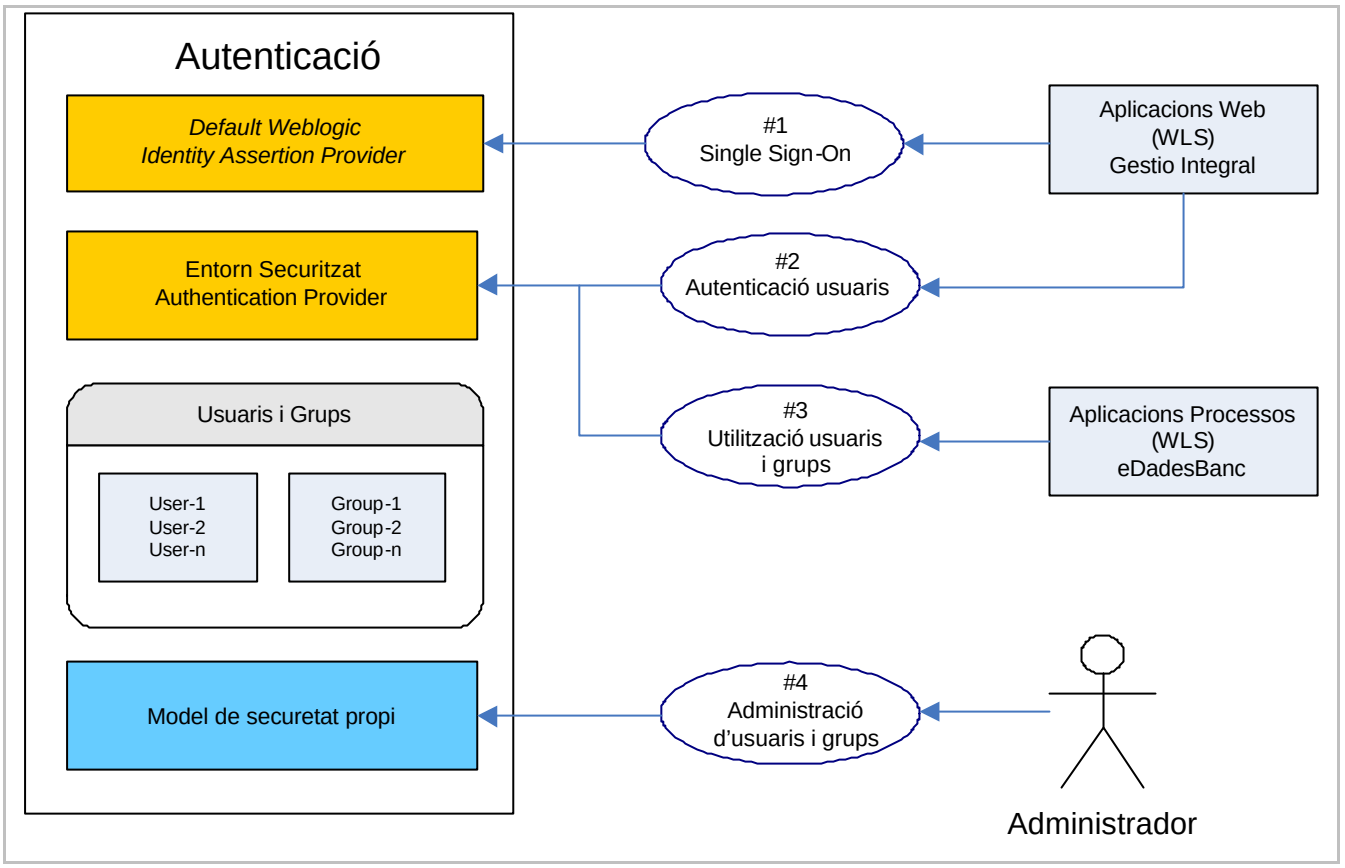
Actor Administrador de GEUS

Es l'usuari que administra els usuaris i els grups. Es qui assigna un role al usuari, per una determinada aplicació. Suposant els roles {Master, Admin, Usuari}, un usuari pot ser administrador a FACT, master A CONTA , i usuari a CRM.

3.4.2 Descripció dels casos d'ús

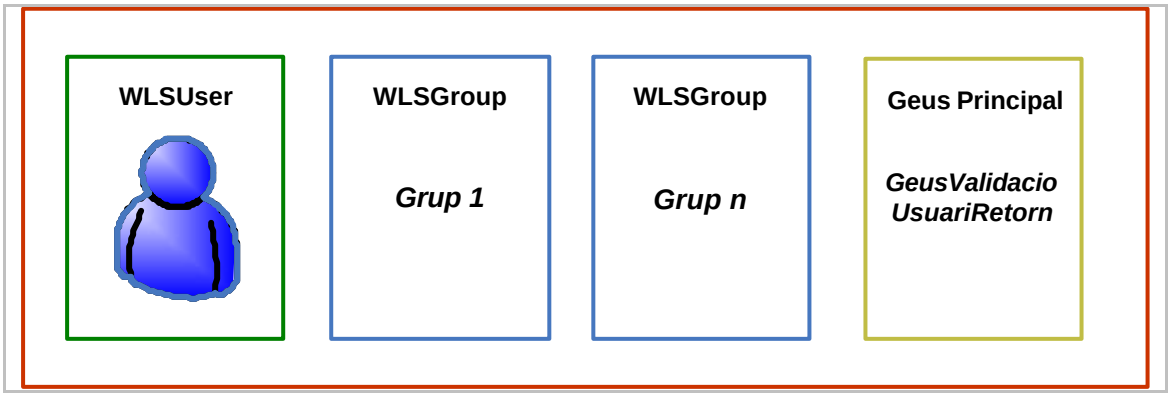
3.4.2.a Cas d'ús: Autenticació

Aquest és el procés mitjançant el qual s'identifica la identitat de l'usuari, és a dir, es valida que l'usuari existeixi en el sistema així com les seves credencials. Aquest procés es executat pel proveïdor d'autentificació del proveïdor de seguretat.



Cas d'ús autenticació

Si el procés d'autenticació s'executa amb èxit es construeix el subject de l'usuari que constitueix la seva identitat:



Subject de l'usuari

Com podem veure a la imatge, el subject d'un usuari està format per:

- L'identificador de l'usuari.
- Els grups associats a l'usuari

- Un principal especial que s'utilitza per emmagatzemar les dades retornades pel sistema GEUS. Aquestes dades contenen els roles de l'usuari agrupats en grups. Donat que els roles de l'usuari depenen del grup al que pertanyen, aquest principal disposa del grup seleccionat per l'usuari que pot canviar en funció de l'aplicació seleccionada per l'usuari. Un usuari potser es administrador al mòdul de facturació (FACT) però es un usuari normal al mòdul de comptabilitat(CONTA)

La següent taula mostrar la descripció del cas d'ús d'autenticació:

Descripció del cas d'ús d'autenticació	
ID	UC-Autenticació
Descripció	Assegurar l'existència i comprovació de les credencials d'un usuari en el sistema.
Actor	Usuari del sistema
Precondicions	L'usuari no es troba autenticat en el sistema.
Fluxe principal	▪ L'usuari es dirigeix a l'aplicació PORT per tal d'autenticar-se. ▪ L'aplicació PORT invoca a GEUS per tal de realitzar la validació de l'usuari. ▪ GEUS retorna totes les dades agrupades de l'usuari: o Informació de l'usuari. o Grups als que pertany. o Roles per grups. ▪ L'usuari es dirigirà a l'aplicació CONTA , FACT, CRM … en funció del que ha escollit a PORT. ▪ Si l'usuari ha escollit CONTA O FACT s' invoca al proveïdor de seguretat amb les dades retornades per GEUS. ▪ El proveïdor de seguretat autentica a l'usuari creant el seu corresponent subject que serà utilitzat en el procés d'autorització. Addicionalment, crea un token que serà utilitzant pel mecanisme de SSO.
Postcondicions	L'usuari es troba autenticat en el sistema i, per tant, pot accedir a recursos Web i d'EJB en funció dels seus principals i les regles de validació determinada pels seus roles associats.

El següent diagrama de seqüència mostra la interacció temporals entre els diferents sistemes i/o components durant el procés d'autenticació:

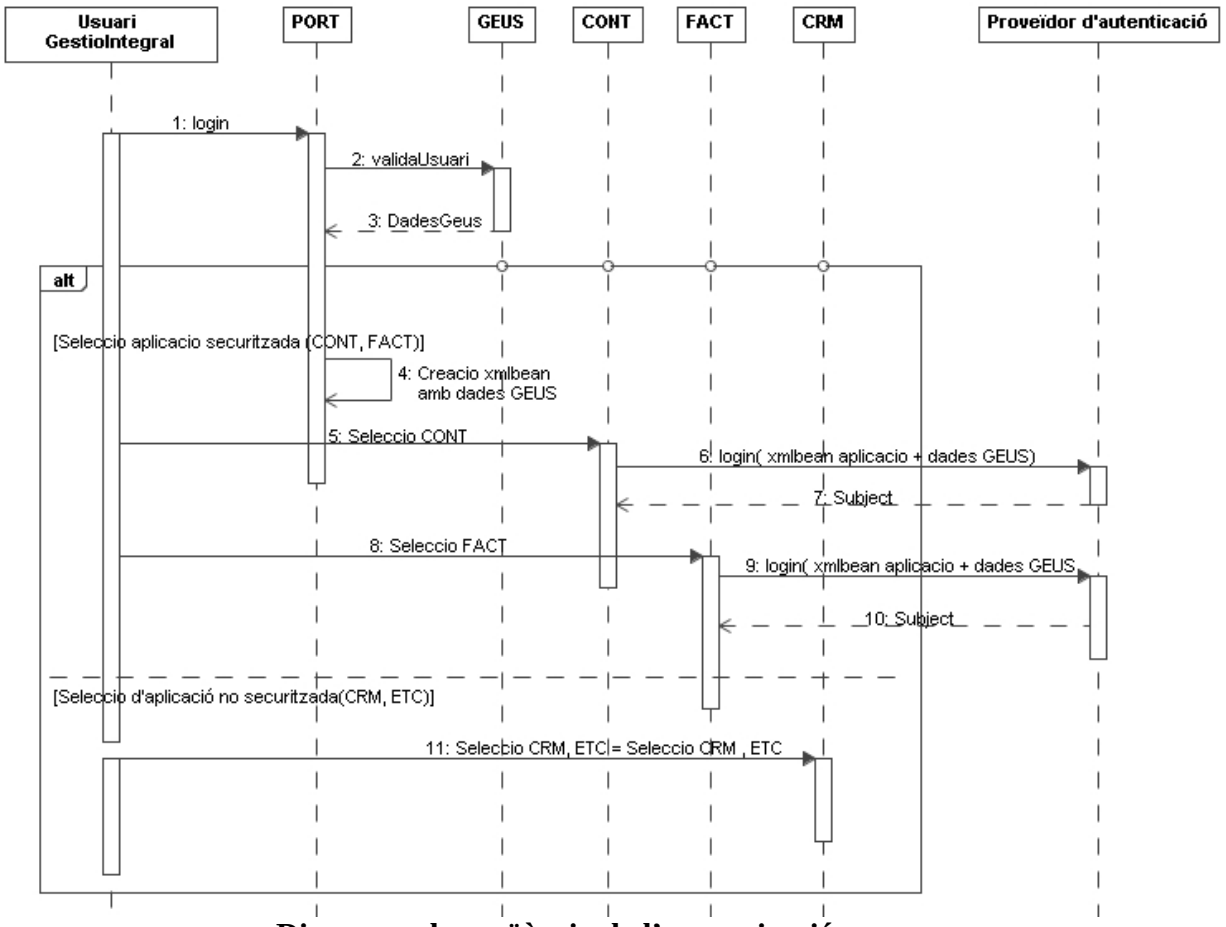
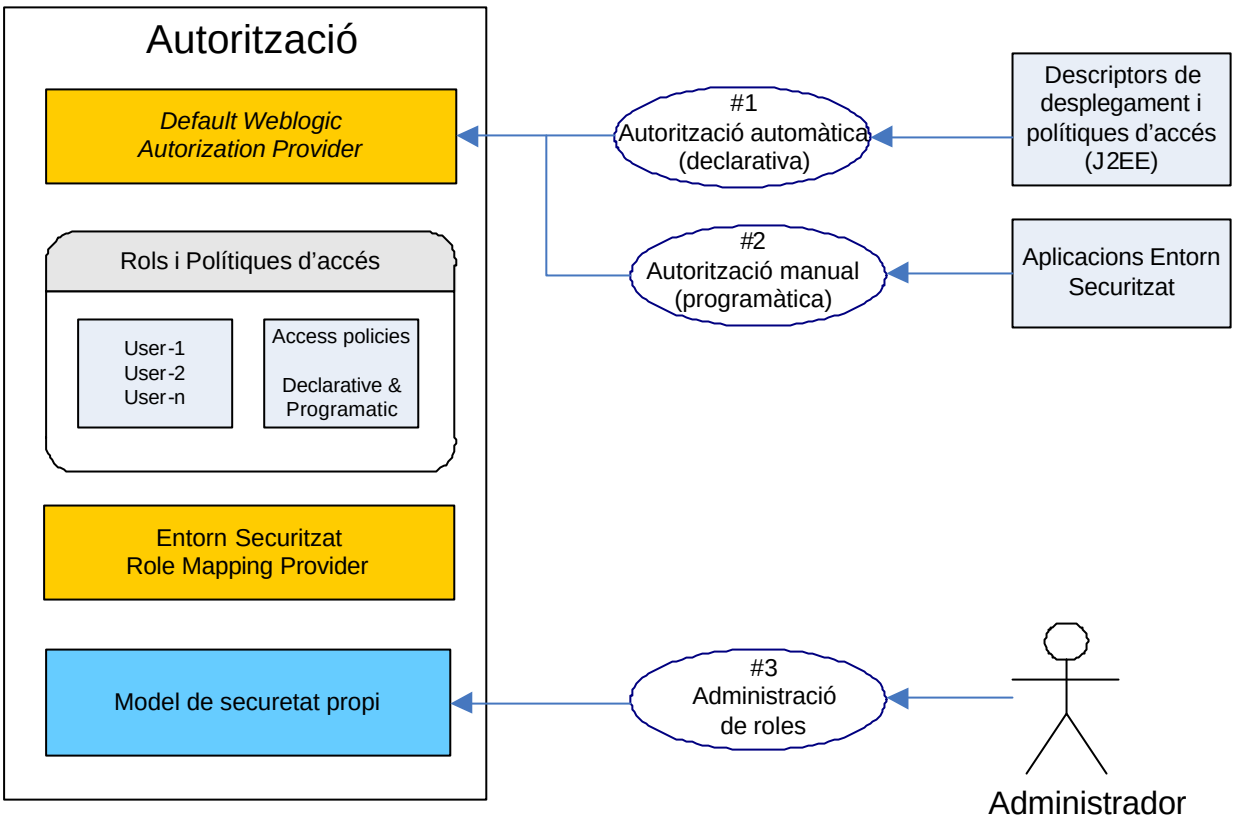


Diagrama de seqüència de l' autenticació

3.4.2.b Cas d'ús: Autorització

És el procés mitjançant el qual WebLogic server autoritza o denega l'accés a un recurs Web o EJB (associat a un role de seguretat) a un usuari en funció de les dades d'identitat de l'usuari i els seus roles associats.



Cas d'ús autorització

Tant a les aplicacions Web com als components EJB, podem protegir recursos creant associacions entre els recursos i els roles que poden accedir a aquests recursos. A la següent imatge tenim un exemple de protecció d'un recurs Web, en aquest cas, es tracten de tot el conjunt d'URL's que comencin per /hello:

```
<security-constraint>
    <display-name>security constrain</display-name>
    <web-resource-collection>
        <web-resource-name>say hello</web-resource-name>
        <url-pattern>/hello/*</url-pattern>
        <http-method>GET</http-method>
        <http-method>POST</http-method>
    </web-resource-collection>
    <auth-constraint>
        <role-name>role1</role-name>
    </auth-constraint>
    <user-data-constraint>
        <description/>
        <transport-guarantee>NONE</transport-guarantee>
    </user-data-constraint>
</security-constraint>
```

Com podem veure, la política de seguretat estableix que aquesta restricció de seguretat està associada a tots aquells usuaris que tinguin el role role1.

```
<security-role>
    <role-name>role1</role-name>
</security-role>
<!--  end role definition -->
```

El proveïdor de mapeig de roles s'encarrega de retornar els roles associats a l'usuari en funció de la aplicació seleccionada.

La següent taula mostrar la descripció del cas d'ús d'autorització:

Descripció del cas d'ús d'autorització	
ID	UC-Autorització
Descripció	Atorgar o denegar l'accés a un recurs Web o EJB protegits amb el mecanisme estàndard basat en roles de seguretat
Actor	Usuari del sistema
Precondicions	L'usuari es troba autenticat en el sistema.
Fluxe principal	▪ L'usuari vol accedir a un recurs protegit. En el nostre cas a un mètode de EDadesBanc ▪ WLS sap el mapa de polítiques de seguretat perquè les ha llegides dels descriptors de les aplicacions. ▪ El proveïdor de mapeig de roles retorna el rol per defecte o el role associats al grup treball. ▪ El proveïdor d'autorització aprova l'accés si el role de la política de seguretat del recurs protegit és igual o es troba dintre de la llista de roles retornat pel proveïdor de roles.
Postcondicions	L'usuari accedeix al recurs protegit si el proveïdor d'autorització li autoritza.

El següent diagrama de seqüència mostra la interacció temporal entre els diferents sistemes i/o components durant el procés d'autorització:

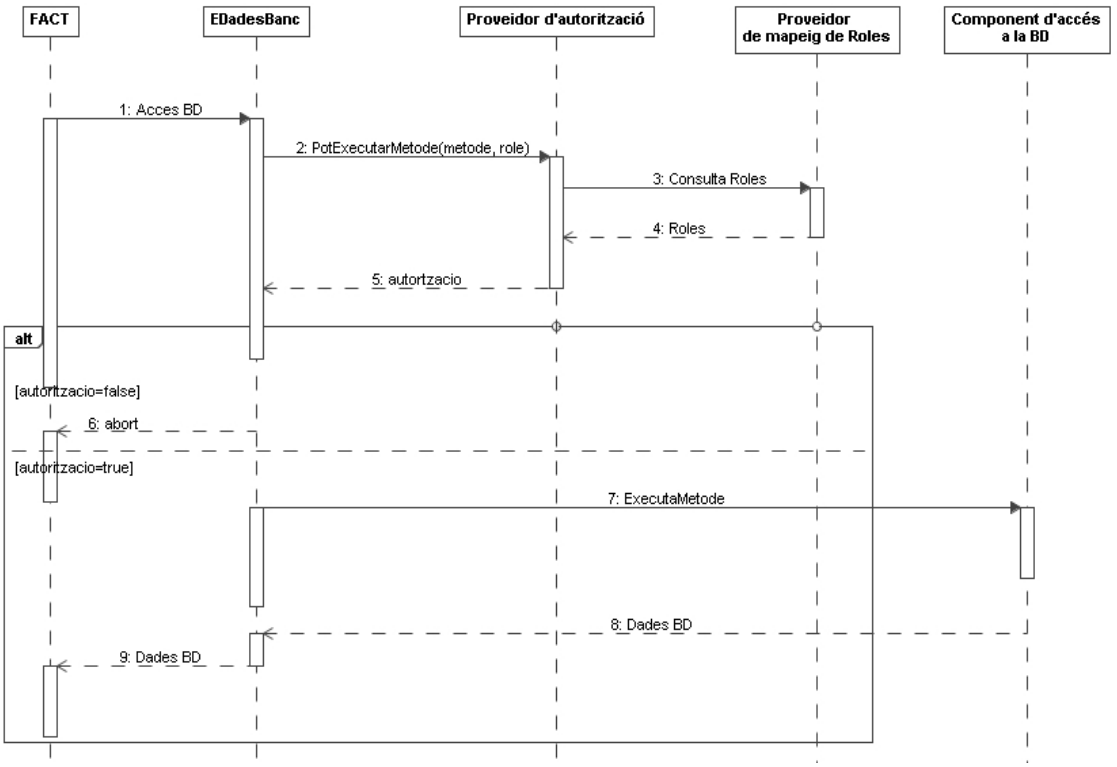


Diagrama de seqüència de l' autorització

3.4.2.c Canvis relacionats amb la autorització

Amb relació al manteniment de la autorització dels objectes, es a dir en respecte als seus possibles canvis tenim que:

1. Canvis dels roles en els descriptors: La informació de seguretat s'especifica a nivell dels descriptors de l'aplicació d'empresa. Si es vol fer alguna modificació en aquests descriptors l'aplicació ha de ser redesplegada perquè aquests canvis siguin aplicats.

2. Nivell de canvis: El nivell de canvis no és elevat, és a dir, per securitzar les aplicacions només s'han de modificar els seus descriptors. Inicialment, s'afegiran tots els roles possibles i, més endavant, quan es completi el mapa final de la seguretat es poden tornar a modificar per tal de fer-les més restrictives

4. DISSENY.

4.1 Introducció

El projecte del proveïdor de seguretat es un embolcall de seguretat JAAS que posarem per sobre de dos dominis diferents WebLogic.

Es tracta de crear un context de seguretat vàlid per aquestes aplicacions residents en aquets dominis.

El projecte del proveïdor de seguretat es aquest embolcall de seguretat, no pas aquestes aplicacions. No obstant donat que els mòduls on establirem el proveïdor de seguretat son mòduls que han de existir i donat que no disposem d'aquets mòduls, els haurem de construir mínimament.

Necessitem simular la aplicació GestioIntegral.cat que es la aplicació consumidora , i EdadesBanc que es la aplicació Servidora.

La aplicació GestioIntegral té exclusivament la capa de presentació, mentre que la capa de negoci resideix a EdadesBanc; es tracta per tant de l'ús del patró MVC entre dominis.

Al apartat 4.2 tractem de la construcció d'aquets mòduls de suport i als posteriors de la solució tècnica per securitzar aquest escenari tot plantejant la solució mitjançant la definició de la arquitectura del sistema i de la realització dels casos d'ús principals, que son la autenticació i la autorització

4.2 Aplicacions de suport

4.2.1 GestioIntegral .cat : Aplicació del Core Business

4.2.1.a Descripció:

Es tracta del nom genèric que té el conjunt de aplicacions de la empresa. GestioIntegral.cat consta de n aplicacions, entre les que destaquem:

- GEUS :Aplicació que gestiona els usuaris i centralitza el accés al LDAP
- PORT :Aplicació de punt de entrada a GestioIntegral.cat
- FACT :Aplicació de facturació que usa EDadesBanc. Les planes de accés a EDadesBanc son les mateixes de CONTA.
- CONTA: Aplicació de comptabilitat que usa EDadesBanc. Les planes de accés a EDadesBanc son les mateixes de CONTA.
- CRM: Aplicació de seguiment de clients , que no usa EDadesBanc

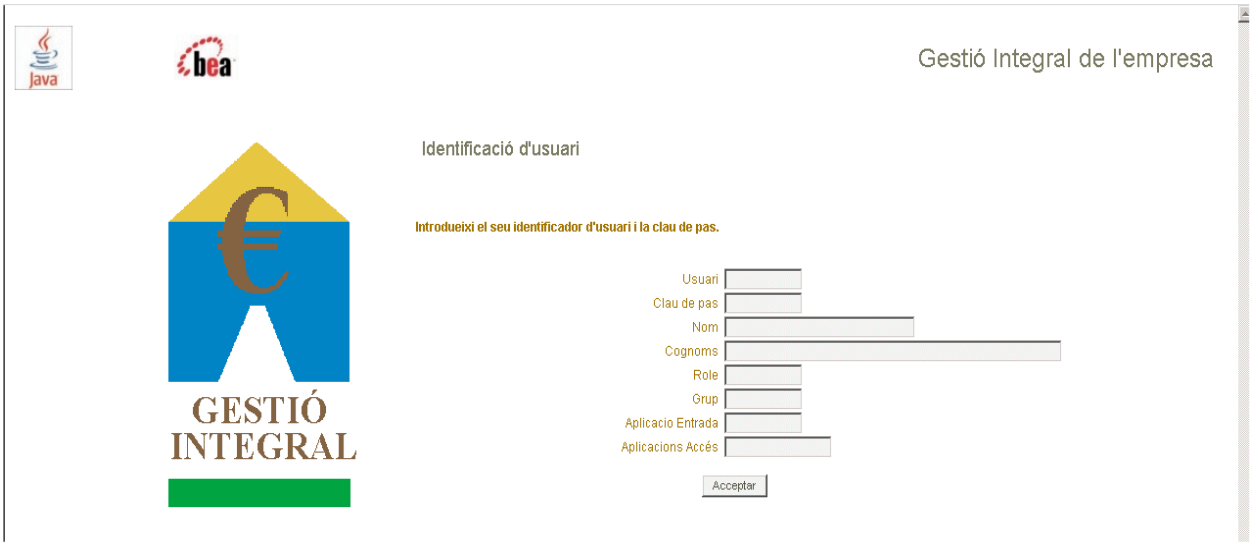
A) Accés a GestioIntegral.cat

Per tal de simular la aplicació construirem una plana d'accés per inserir les dades al xmlbean GusValidacioUsuariRetornDocument que contindrà les dades que necessita el proveïdor de seguretat per crear el context.

En aquesta plana podrem informar els següents camps:

- Usuari → Codi de l'usuari
- Clau de pas → Clau de pas de l'usuari
- Nom → Nom de l'usuari
- Cognoms → Cognoms de l'usuari
- Role → Role del usuari per el grup. Usarem els següents roles:
  - Master
  - Admin
  - User
- Grup → Un dels Grups al que pertany l'usuari
- Aplicació Entrada → Aplicació a la que es vol accedir
- Aplicacions Accés→ Aplicacions a les que l'usuari té accés. Si en té més de una les separem amb comes:
  - 1 Aplicació = FACT
  - 2 o més aplicacions = FACT,CONTA,CRM,APP4....

Tal com ho podem veure a la imatge:



Accés a GestioIntegral.cat

B) Manteniment del compte bancari

Una vegada hem entrat a la aplicació i s'ha inserit la informació del usuari a GeusValidacioUsuariRetornDocument , se'ns mostra la plana inicial del manteniment del compte bancari del banc.
Aquesta plana inicial, es una plana de recerca, des don podrem veure la informació dels diferents comptes bancaris segons els criteris de recerca establers, crear un nou compte bancari, editar-lo o esborrar-lo.

Els criteris de recerca que podem informar son:

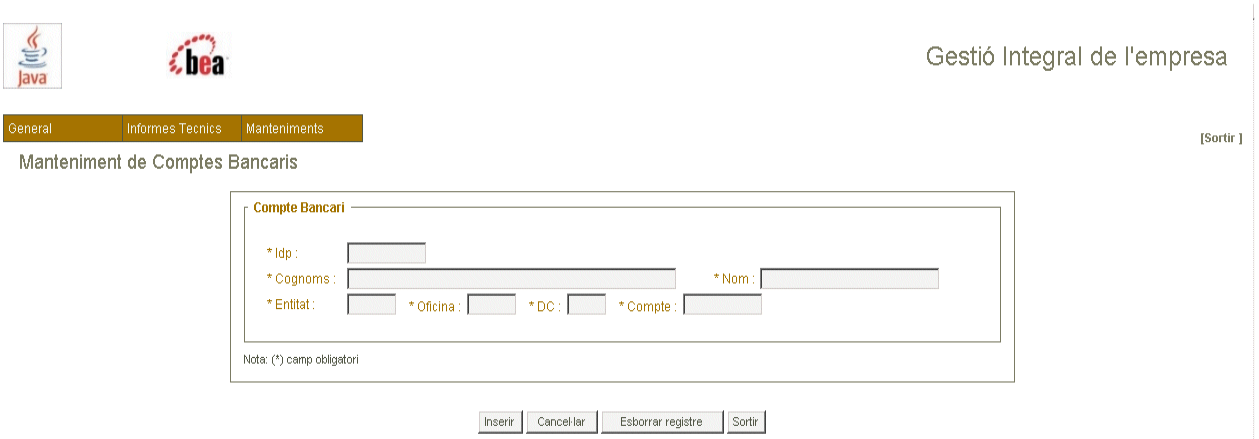
- Cognoms → Cognoms del usuari
- Entitat → Núm. d'entitat a la que pertany el compte
- Compte → Núm. de Compte bancari

Ho podem veure a la següent imatge :

Recerca dels comptes bancaris

Si la recerca torna algun registre, el podrem seleccionar i anar a la plana de manteniment per tal de canviar les seves dades o bé per esborrar el compte
Si l'usuari prem el botó "Nou" anirem a la plana de manteniment, per tal de inserir un nou compte.

Veiem la plana de manteniment a la imatge:

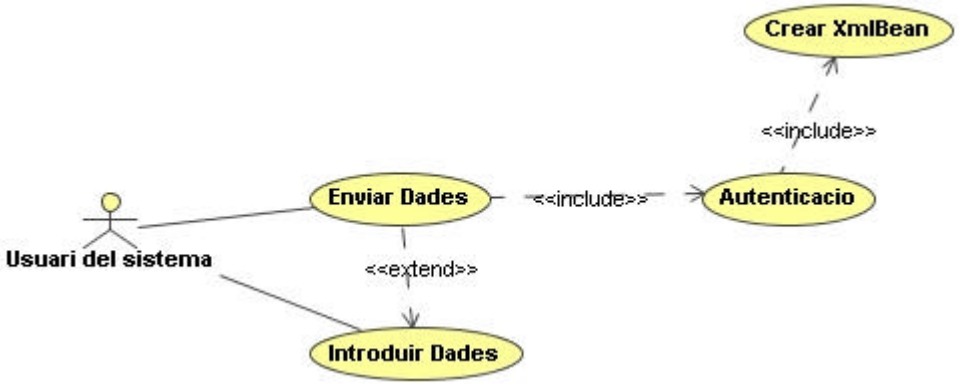


Manteniment dels comptes bancaris

4.2.1.b Realització dels casos d'ús

A) Realització del Cas d'ús d'accés a la aplicació

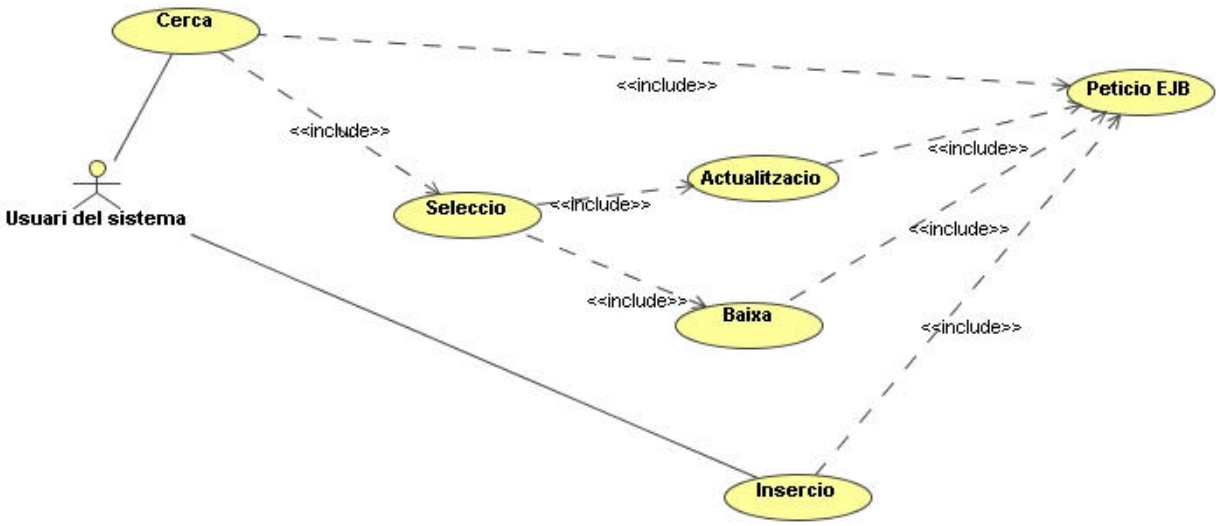
S'introdueixen les dades necessàries per construir el xmlbean i cridem al procés de login del proveïdor de seguretat



Cas d'us Accés a la aplicació

B) Realització del Cas d'ús d'operacions contra la base de dades

Qualsevol operació contra la base de dades, es connecta amb Edades banc que es qui serveix les dades i realitza les operacions. El usuari introdueix les dades als formularis a la vista i fa la petició mitjançant el client del EJB a EdadesBanc que es qui retorna les dades i els resultats de les operacions.



Cas d'us Operacions contra la base de dades

4.2.2 EdadesBanc: Aplicació servidora

4.2.2.a Descripció

Aplicació d'empresa que rep peticions d'accions a base de dades i retorna els resultats a les aplicacions que consumeixen els seus serveis.

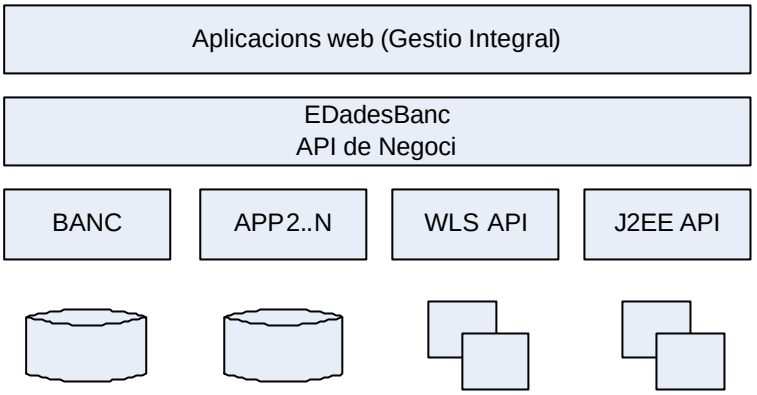
4.2.2.b Model de Interacció per a clients

Per construir el sistema d'interacció amb EdadesBanc es vol dissenyar un model d'interacció per a client basat en un API de negoci. Aquesta capa permetrà aïllar de forma efectiva els detalls tècnics de EDadesBanc de la utilització que se'n faci al client.

La imatge representa l'API de negoci de EDadesBanc. EDadesBanc un sistema complex que accedeix a diferents subsistemes :

- BANC: model de comptes del banc
- APP2..APPN: model de altres dades del sistema

Les aplicacions de client de GestioIntegral.cat podran executar totes les operacions de negoci de EdadesBanc i no hauran de interaccionar directament amb tots aquests sistemes.



API de negoci de EdadesBanc

EDadesBanc ha de considerar-se un servei més de GestioIntegral.CAT. El disseny de EdadesBanc l'ha estructurat per integrar-se dins una arquitectura complexa orientada a serveis.

4.2.2.c Missatgeria XML

A) Missatges XML

Tota la comunicació i intercanvi d'informació entre client i EDadesBanc es farà mitjançant missatgeria XML. Aquest format de missatge permet la màxima flexibilitat i és el més acceptat per a arquitectura orientades a serveis.

Com a avantatges de la missatgeria XML:

- Flexibilitat de processament i enviament d'informació – Productivitat
- Definició tipus de missatge amb esquemes XML (XSD) – Reutilització

Per al processament dels formats de missatge XML s'ha escollit la tecnologia de referència Apache XMLBeans (http://xmlbeans.apache.org). XMLBeans permet un mapeig directe entre XML i JavaBeans, la qual cosa simplifica el procés de programació.

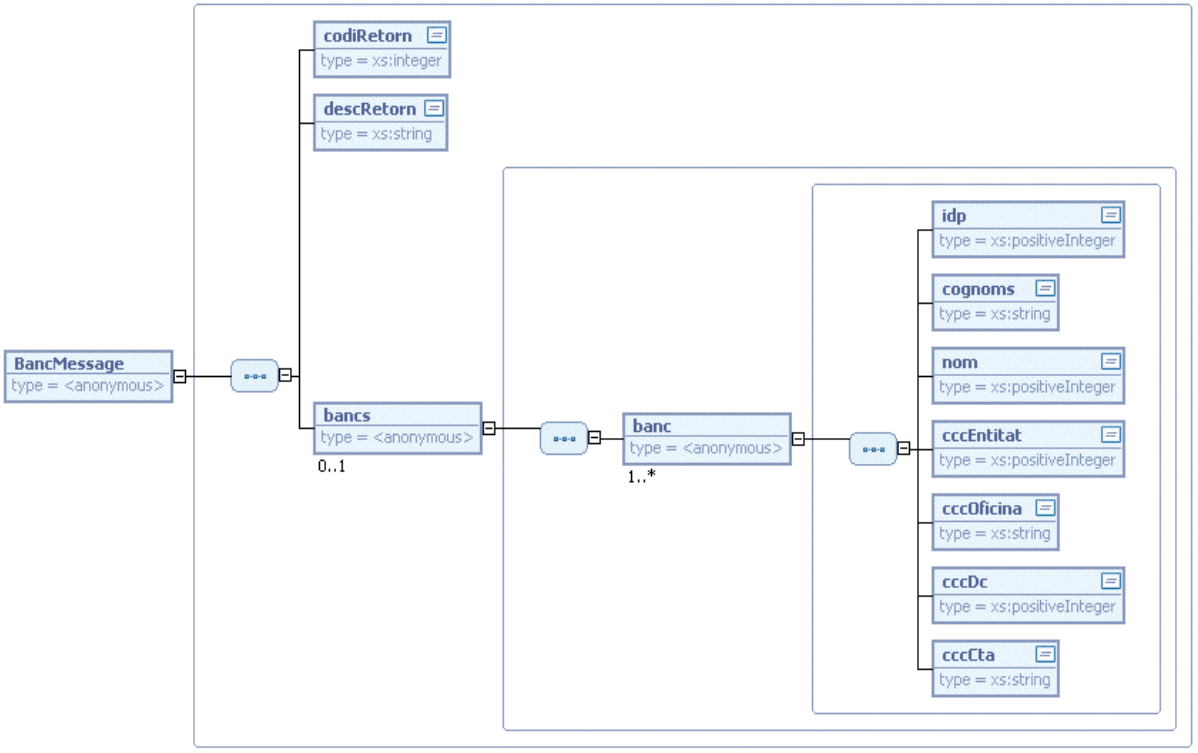
Tot l'intercanvi de missatges definit a l'API de negoci de EDadesBanc fa servir JavaBeans generats amb tecnologia XMLBeans. No obstant això, en qualsevol moment es podria redefinir la signatura de l'API per intercanviar només text XML i que el client utilitzés una altra tecnologia.

B) Format de missatgeria

La part més important de la missatgeria és la definició del format (XML esquema – XSD). Aquesta representació és bàsica per a l'èxit d'aquesta aplicació d'empresa ja que ha de permetre la flexibilitat suficient perquè pugui evolucionar de forma senzilla.

S'ha definit missatgeria per a la taula de comptes bancàries:
Principals entitats:

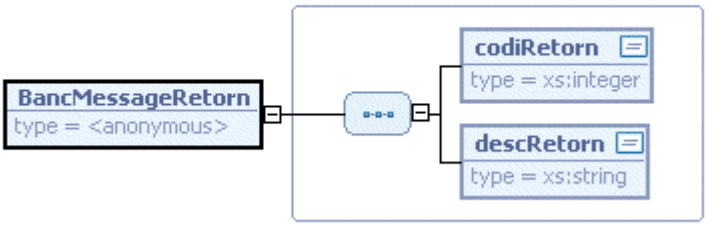
- **BancMessage** , que compté un o més comptes bancaris (una celest torna



Format BancMessage

més d'un compte) i un filag amb el codi de resultat de la operació i una descripció, per les operacions de selecció.

- **BancMessageRetorn**, flag amb el codi de resultat de la operació i una descripció



Format BancMessageRetorn

4.2.2.d API de Client

La Següent taula descriu els mètodes de negoci dissenyats per a l'API client de EDadesBanc. Els mètodes més importants són els que permeten accedir a la base de dades per a consultar les dades o bé per a realitza-hi accions.

El disseny de l'API assegura que qualsevol necessitar posterior pugui ser integrada de forma molt senzilla.

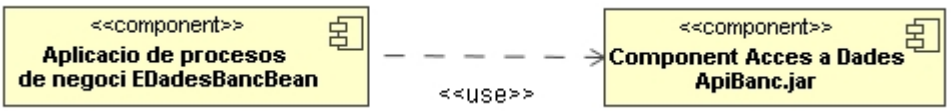
Operació	Paràmetres	Resultat	Descripció
getCompteBancari	BancMessage	BancMessage	Consulta un o més comptes bancaris en funció dels paràmetres enviats a BancMessage, retorna un document BancMessageRetorn
setCompteBancari	BancMessage	BancMessage Retorn	Insereix un nou compte Bancari i retorna un document BancMessageRetorn
updateCompteBancari	BancMessage	BancMessage Retorn	Actualitza un compte Bancari existent i retorna un document BancMessageRetorn
delCompteBancari	BancMessage	BancMessage Retorn	Esborra un compte Bancari existent i retorna un document BancMessageRetorn

Operacions de negoci disponibles a l'API client de EDadesBanc

4.2.2.e Arquitectura

A) Diagrama de components

La imatge representa el diagrama dels principals components definits a EdadesBanc:



Descripció dels components:

- **EdadesBancBean**

  EJB stateless d'entrada a EDadesBanc. Totes les operacions de negoci de EDadesBanc estan gestionades directament per aquest EJB.
  Defineix la transaccionalitat i API d'accés a EDadesBanc. Els paràmetres d'entrada/sortida són XMLBeans.
  Accedeix a base de dades i hi realitza accions mitjançant l'ApiBanc.jar

- **ApiBanc.jar**

  Component d'infraestructura que permet consultar informació de base de dades i realitzar-hi insercions, actualitzacions i esborrats de registres. Aquest component està desenvolupat amb Hibernate, i consta del negoci que l'EJB ofereix al seu consumidor.

B) Diagrama classes client J2EE

La Imatge representa el diagrama de classes que farà servir el client J2EE de EDadesBanc. Com que l'API de negoci s'ha implementat amb un EJB stateless, el client serà un RemoteHome estàndard de J2EE. Com a classe d'utilitat s'hi ha afegit un localitzador d'interfícies EDadesBancLocator.

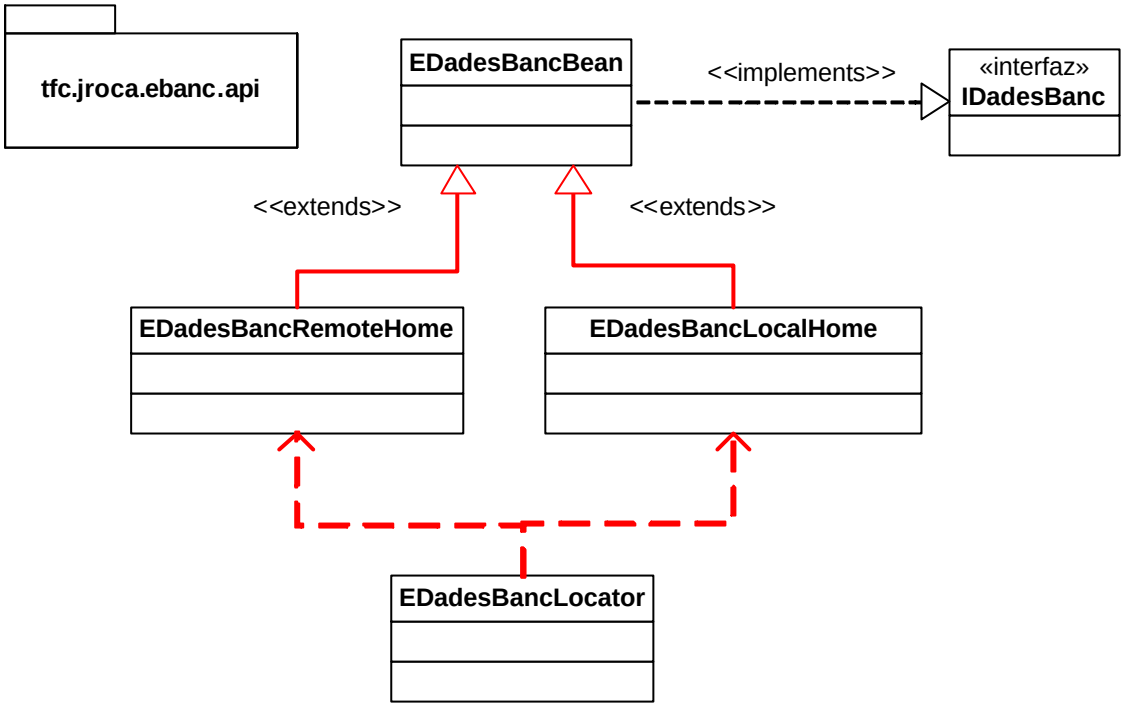


Diagrama de classes client J2EE de EdadesBanc

C) Diagrama classes missatgeria XMLBeans

La missatgeria de client i servidor és XML i està encapsulada amb XMLBeans. La imatge representa la jerarquia de classes existent per a les principals entitats definides.

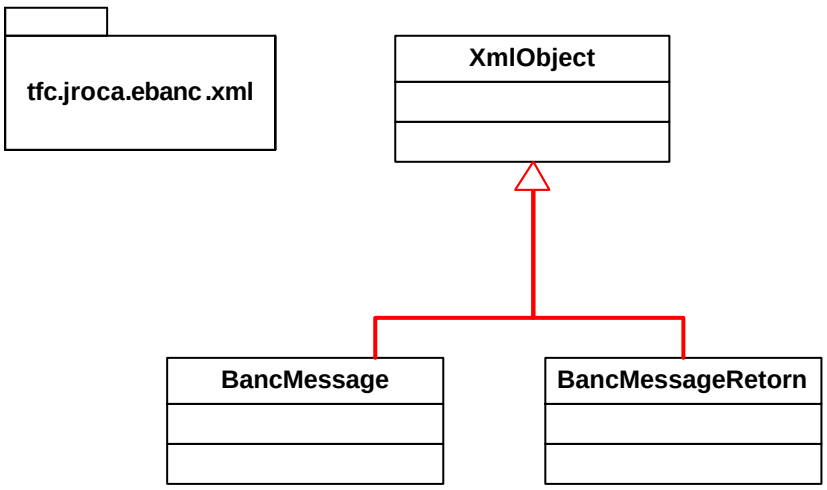
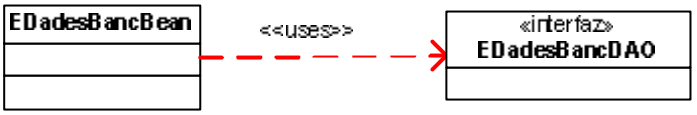


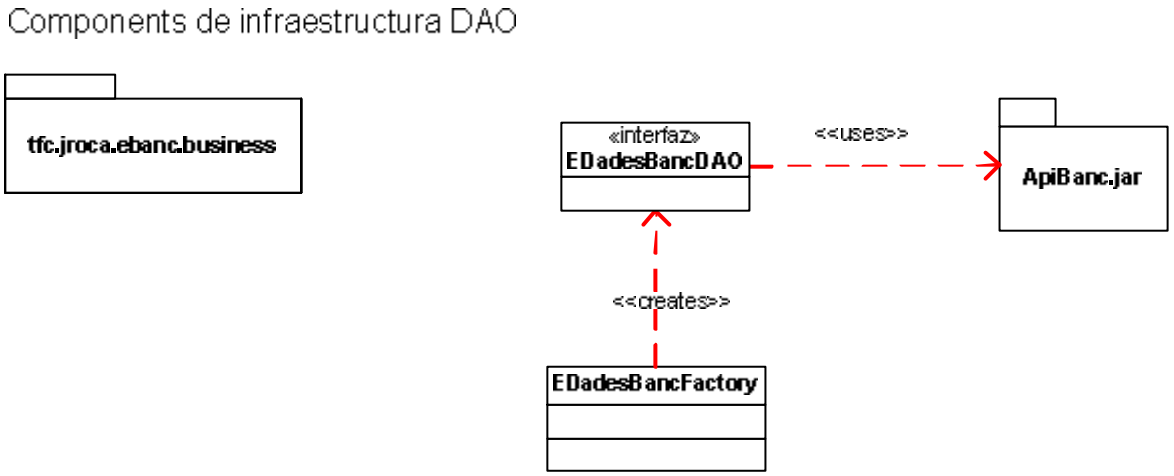
Diagrama de classes de missatgeria XMLBeans

D) Diagrama classes Servidor

L'Ejb EdadesBancBean, utilitza EDadesBancDAO per tal de accedir a la base de dades.

EdadesBancBean crea una instància de EdadesBancDAO mitjançant EdadesBancFactory, una vegada creada la classe DAO, s'invoquen les funcions encapsulades al jar ApiBanc, que accedeix a la Base de dades. El codi de ApiBanc.jar es crea de manera automàtica amb Hibernate Tools.

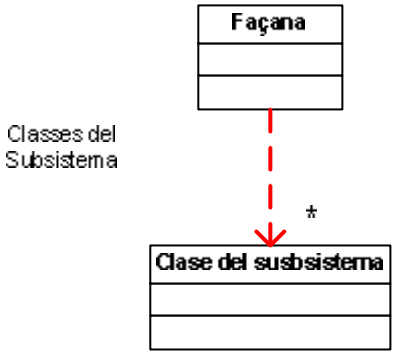




E) Patrons de disseny utilitzats

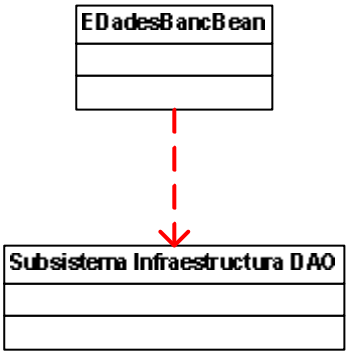
Patró Façana

Per tal de simplificar la interface entre dos sistemes , s'amaga la complexitat dels subsistemes darrera de una classe que fa de façana, de manera que només hi ha un punt de entrada al sistema tapat per la façana, aïllant els canvis que es puguin produir en el subsistema:



Patró Façana

Al nostre projecte, s'ofereix com a punt de entrada el client , amagant el subsistema de components de infraestructura DAO:



Patró Façana aplicat al nostre projecte

Patró Factoria

Centralitzem el lloc on es creen els objectes tornant una instancia de un objecte o un altre, que implementaran la mateixa interface.

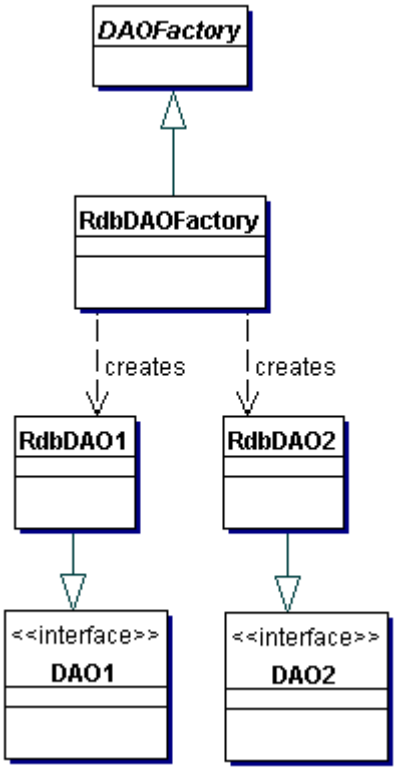
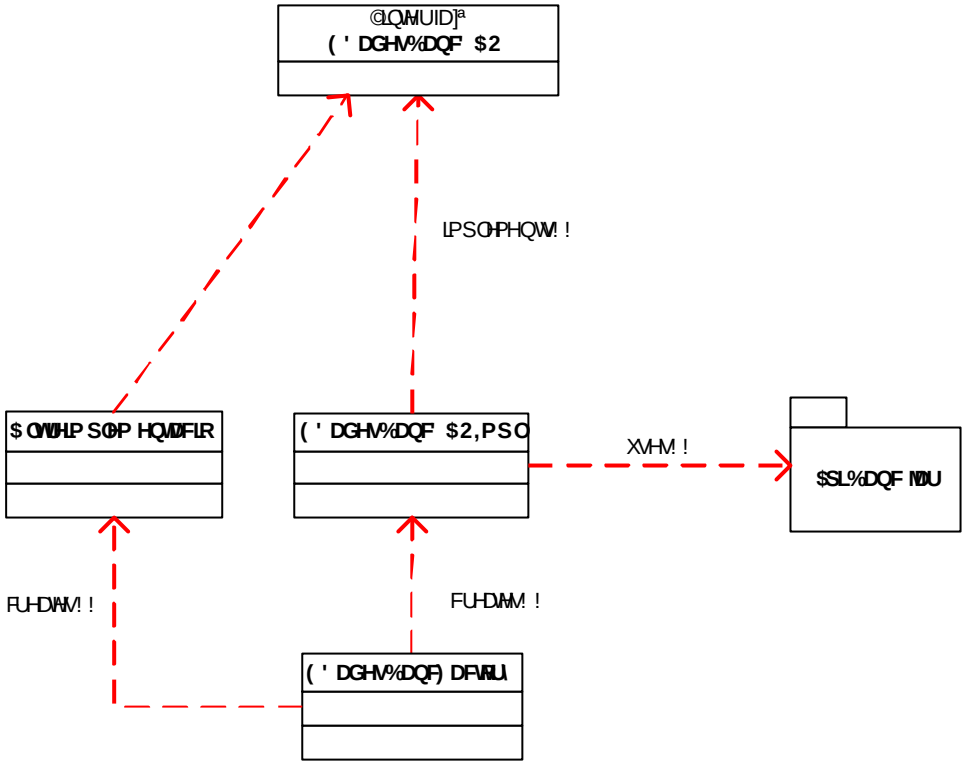


Figure 9.3 Factory for Data Access Object strategy using Factory Method

(font : http://java.sun.com/blueprints/corej2eepatterns/Patterns/DataAccessObject.html)

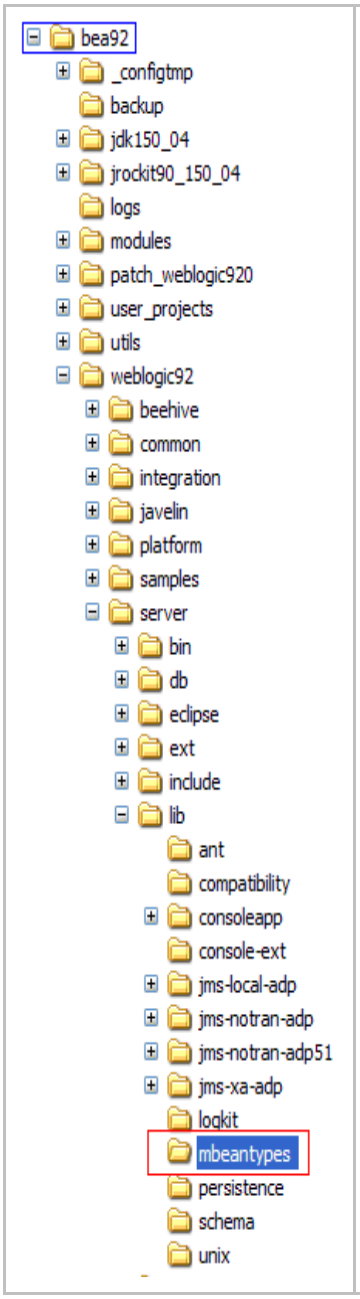
Al nostre projecte, obrim la possibilitat de canvi o millora del accés a base de dades, centralitzant i potser auditant el procés de creació de la classe EdadesBancDAO



Patró Factoria aplicat al nostre projecte

4.3 Definició de l'arquitectura del sistema

4.3.1 Model de desplegament



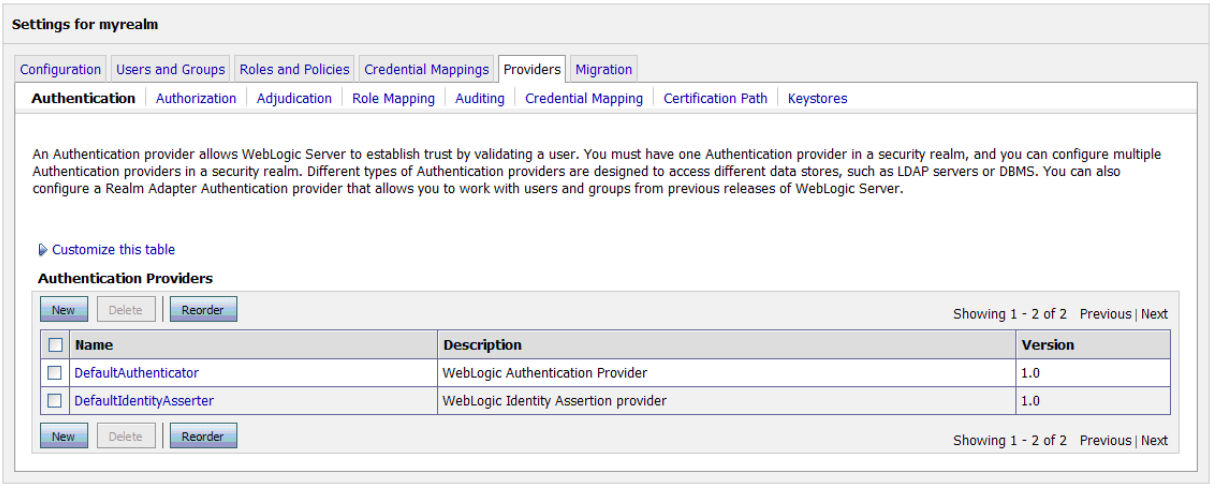
El proveïdor de seguretat no es desplega com qualsevol altre aplicació J2EE sobre WLS. Tots els components del proveïdor de seguretat s'empaqueten en un jar que s'anomena ***bancsecurityprovider.jar***.

Aquest jar s'ha de copiar físicament a:

<BEA_HOME>/weblogic92/server/lib/mbeantypes

Una vegada copiat aquest jar i, després de reiniciar les instàncies de tot el domini afectat, es podrà configurar el proveïdor de seguretat. Tots els dominis que es trobin definits sobre la instal·lació de WLS1 / WLS2 on afegim aquest jar poden utilitzar el proveïdor de seguretat.

Una vegada copiat aquest jar ja es pot configurar el proveïdor d'autentificació en el realm de seguretat del domini així com el proveïdor del mapa de roles.



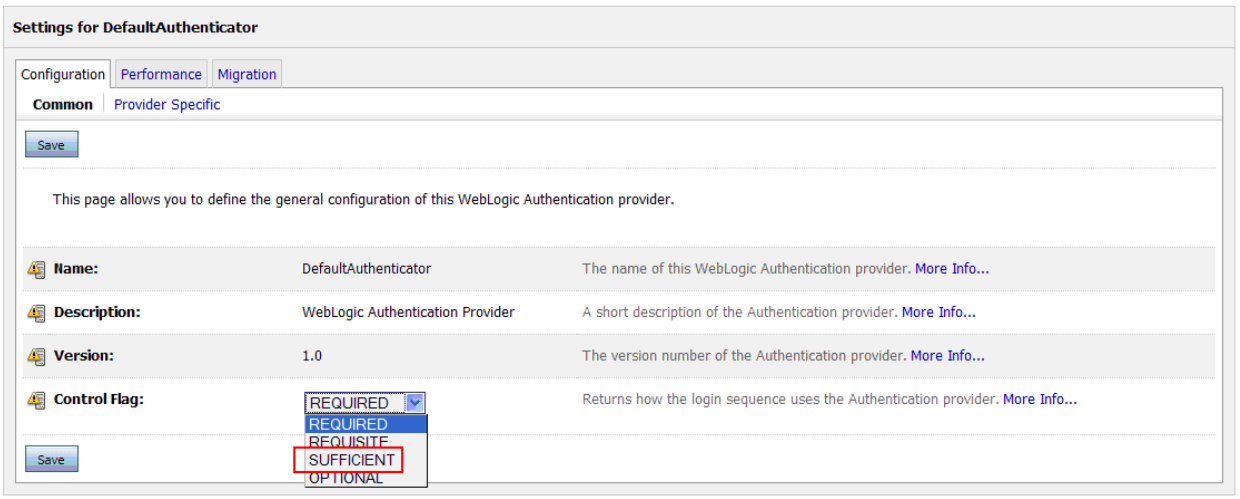
Creació del proveïdor d'autenticació

Una vegada configurat el proveïdor de seguretat s'ha de configurar un flag de control.

Aquest flag de control s'utilitza per indicar el tipus d'utilització del proveïdor de seguretat que pot ser del següent tipus:

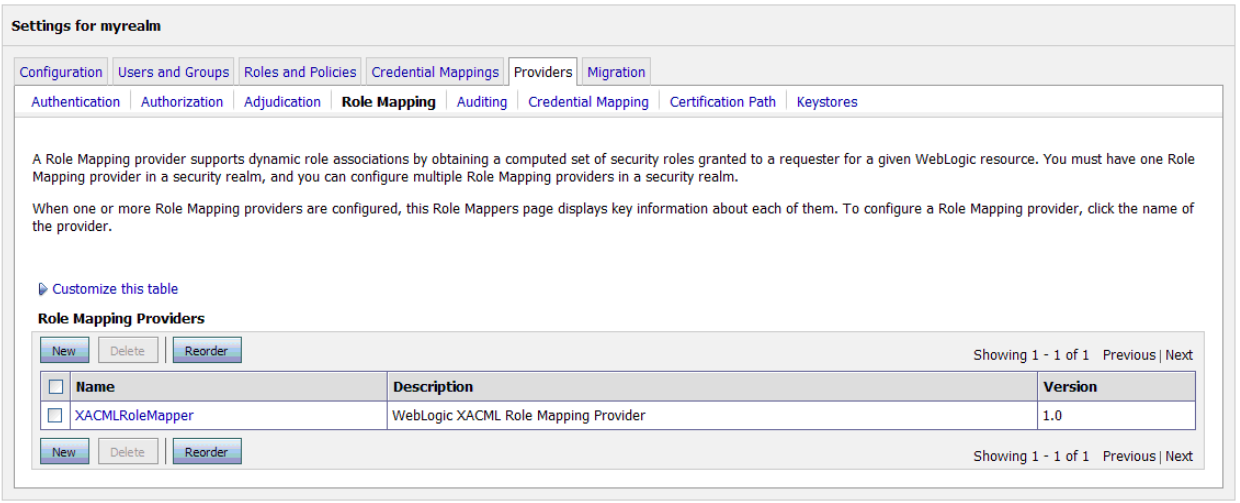
Flag de control del proveïdor de seguretat	Descripció
REQUIRED	Per a que l'usuari s'autentifiqui és imprescindible que l'autentificació es realitzi correctament en aquest proveïdor d'autentificació.
REQUISITE	Per a que l'usuari s'autentifiqui és imprescindible que l'autentificació es realitzi correctament en aquest proveïdor d'autentificació. Si no és així, el control és retornat a l'aplicació.
SUFFICIENT	No fa falta que l'autenticació es produeixi amb èxit. Si l'autenticació es produeix amb èxit el control és retornat a l'aplicació i en cas contrari WLS continua amb el següent proveïdor de la llista.
OPTIONAL	No fa falta que l'autenticació es produeixi amb èxit. En qualsevol cas WLS continua amb el següent proveïdor de la llista.

Per aquest motiu és necessari modificar el flag de control del proveïdor de seguretat per defecte de Weblogic i especificar-lo a SUFFICIENT així com al proveïdor de seguretat de GestioIntegral.Cat.



Modificació del flag de control del proveïdor de seguretat de WLS

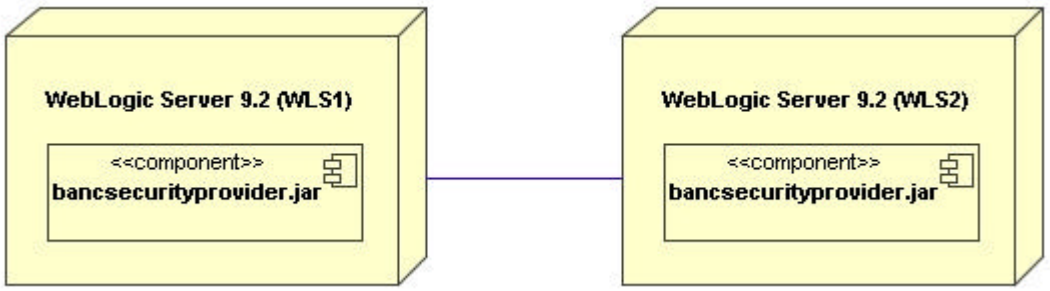
D'aquesta manera poden conviure aplicacions de GestioIntegral amb seguretat amb proveïdor de seguretat i sense.



Creació del proveïdor de mapes de roles

4.3.1.a Diagrames de desplegament

Tal i com hem vist anteriorment, el proveïdor de seguretat es troba empaquetat en un jar que s'ha de copiar al directori *<BEA_HOME>/weblogic92/server/lib/mbeantypes* de cadascuna de les instal·lacions de WLS1 i WLS2.



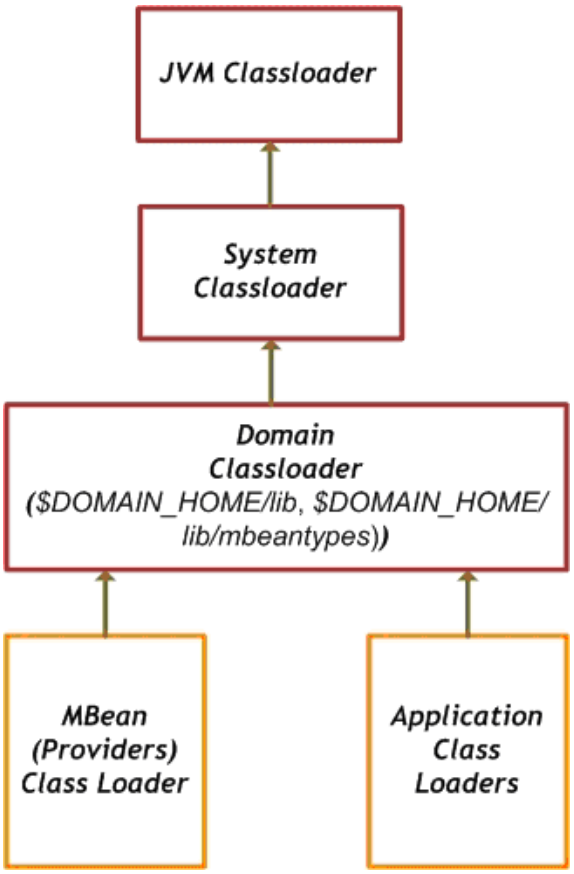
Desplegament del security provider

Sobre els dominis es despleguen les aplicacions que han de ser protegides amb el security provider.

4.3.1.b Descripció de nodes

El proveïdor de seguretat es pot instal·lar sobre WLS 9.2.

La següent imatge mostra la jerarquia de **classloader** de WebLogic Server on podem veure la ubicació del jar del proveïdor de seguretat:



Jerarquia de classloader de WLS

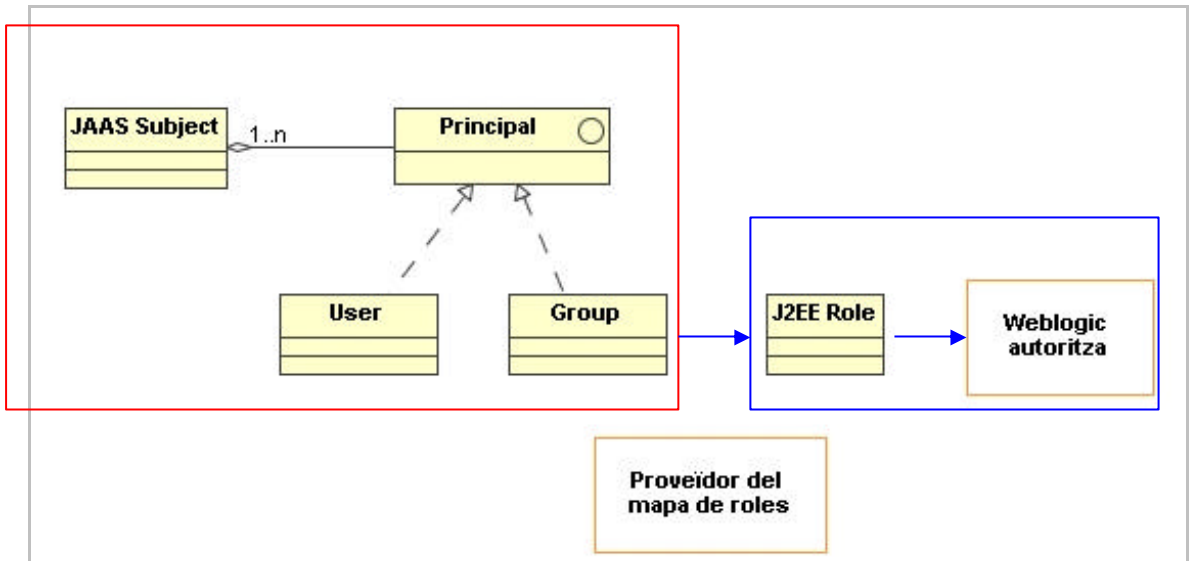
4.3.1.c Descripció de les comunicacions

El proveïdor de seguretat agafa les dades de l'usuari validat des de l'aplicació des don s'hagi d'invocar el login JAAS. En el nostre cas, invocarem aquest login des l'aplicació PORT ,que es la que invoca a la seva vegada a GEUS i recolleix la resposta d'aquesta aplicació .

4.3.2 Model lògic

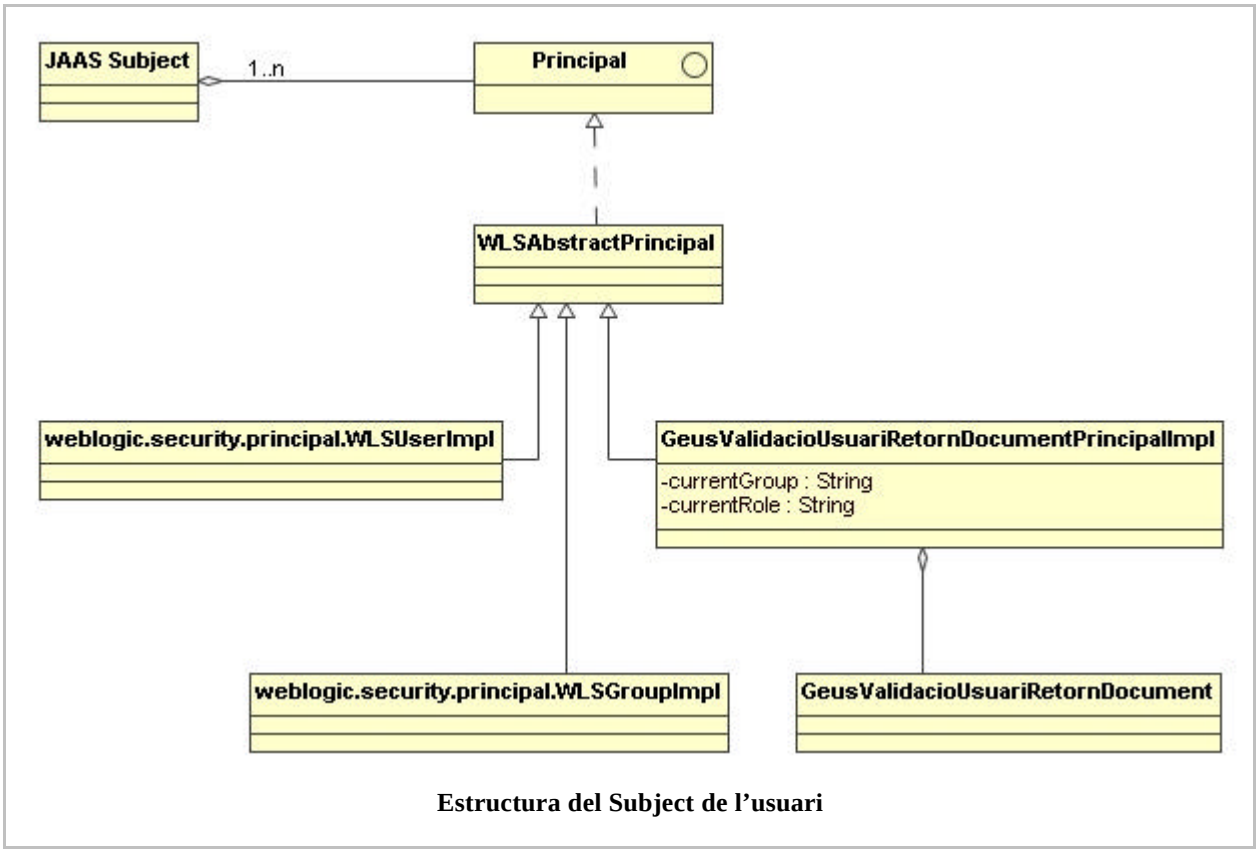
4.3.2.a Descripció general

Tal i com podem veure a la següent imatge, els components que es troben en el requadre són aquells que es troben involucrats en el procés d'autenticació realitzat pel proveïdor de seguretat mentre que els del requadre blau són aquells relacionats amb l'autorització.



Model general de funcionament

El procés d'autenticació construeix el *Subject* de l'usuari format per l'identificador de l'usuari i els seus grups. Aquesta informació s'utilitza per determinar el rol associat a aquell usuari. En el cas del proveïdor de seguretat de GestioIntegral.Cat tenim que el *Subject* de l'usuari està format pels següents components:



Estructura del Subject de l'usuari

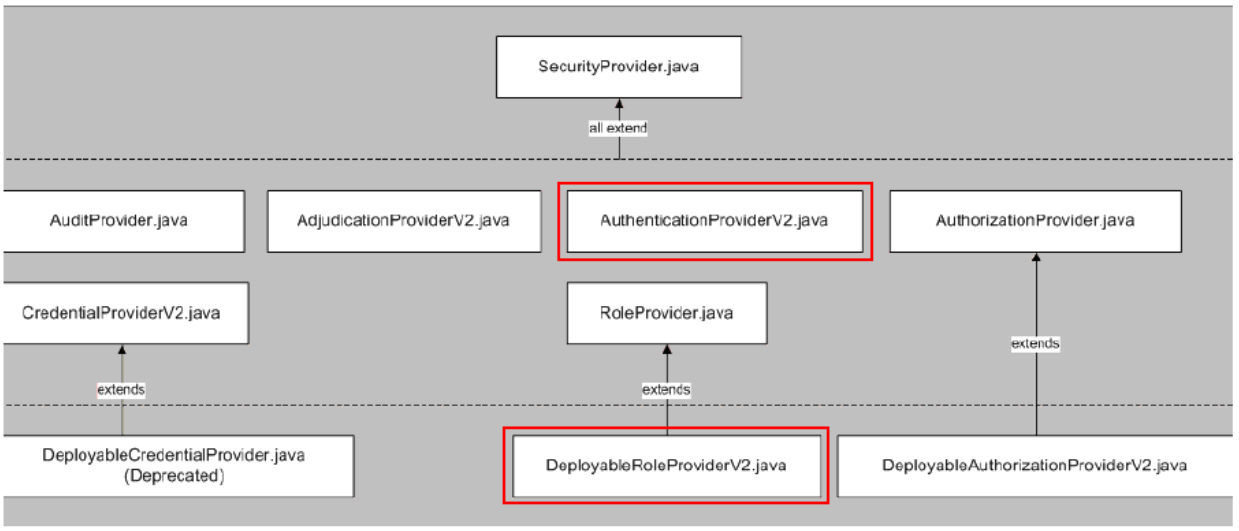
L'usuari tindrà com a grup el grup tornat per GEUS . A més , es defineix un custom principal on es guarda:

- Les dades retornades de l'autenticació realitza realitzada per GEUS i que PORT ha enviat a CONT o FACT (que son les aplicacions securitzades) mitjançant un xmlbean , tals com el codi del usuari per exemple.

- El grup de treball de l'usuari i el seu role de treball

La informació emmagatzemada al Subject de l'usuari és utilitzada pel proveïdor de mapes de roles per tal de retornar a la infraestructura de seguretat de WLS el rol de l'usuari.

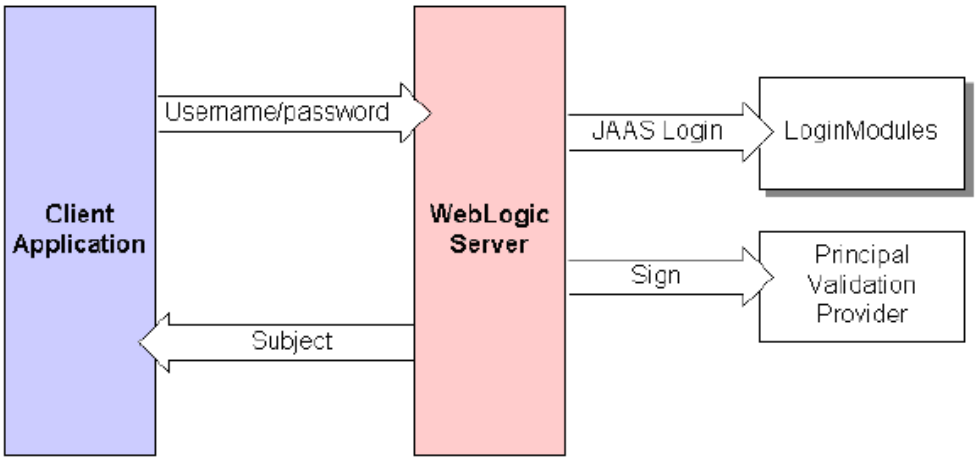
Com podem veure més endavant, el proveïdor de seguretat conté dos tipus de components que són el proveïdor d'autenticació i el proveïdor de mapes de rols. Ambdós components es recolzen sobre els SSPI's de WebLogic Server que són els components que comuniquen els custom providers amb el WebLogic Security FrameWork:



Estructura dels SSPI's

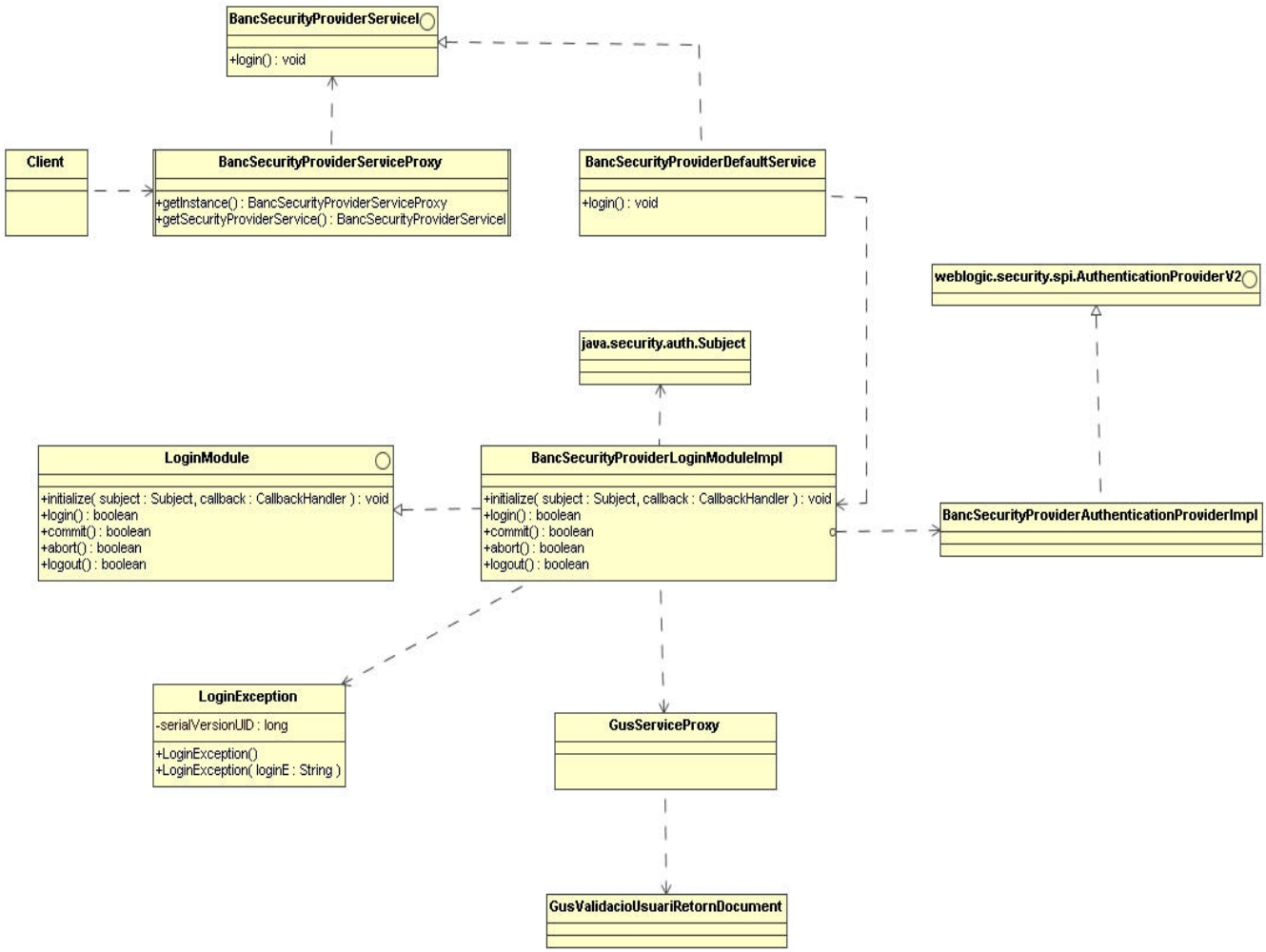
4.3.2.b Descripció capa per al cas d'ús d'autenticació

La següent imatge mostra el procés d'autenticació que es produeix a nivell de WLS i l'aplicació client:



Procés d'autenticació

WLS estableix l'autoritat de l'usuari comprovant les credencials de l'usuari creant el Subject de l'usuari que està format pels principals de l'usuari. Aquesta tasca és realitzada pel mòdul de login de JAAS. La següent imatge mostra els principals components que intervenen en el proveïdor de seguretat:

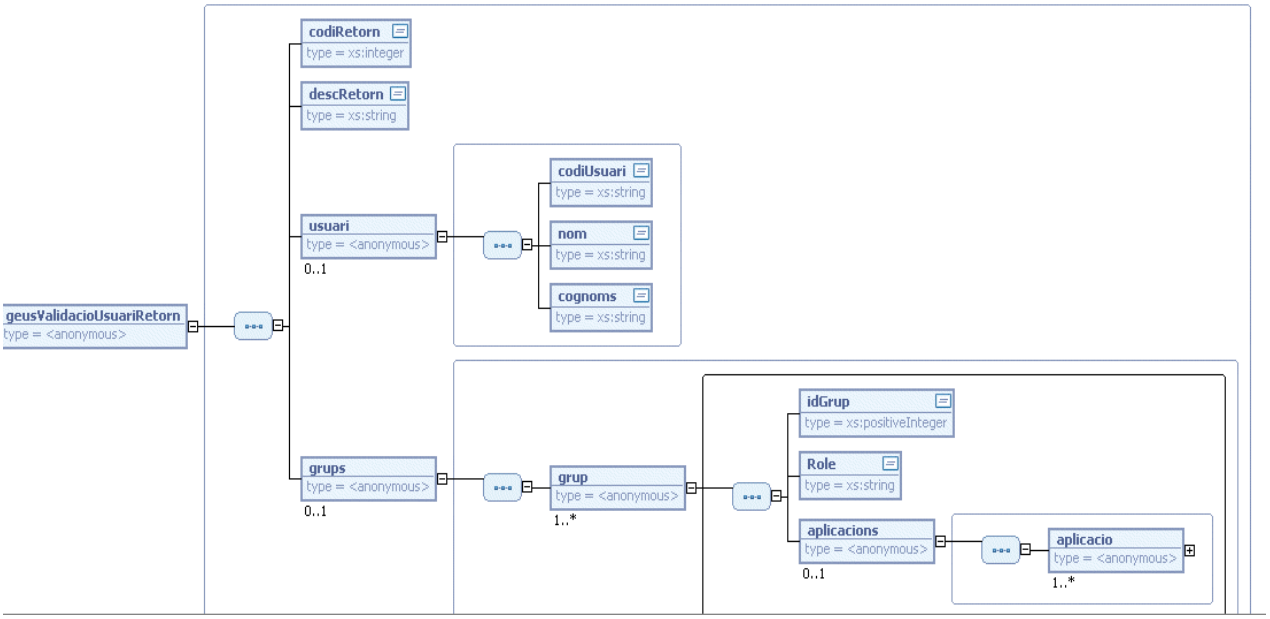


Components per a l'autenticació

Com podem veure a la imatge, existeixen dos proxies d'accés als serveis d'autenticació: del proveïdor de seguretat i de GEUS:

- BancCatSecurityProviderServiceProxy: Proxy d'accés al servei de login per al proveïdor de seguretat.

- GEUSServiceProxy (que simularem en una funció que agafarà les dades de la plana de login i construirà el bean): Proxy d'accés al servei de GEUS que realitza l'autenticació de l'usuari des del PORTAL. Com podem veure, el servei GEUS té associat com a paràmetres de sortida dels seus mètodes de negoci implementats com a XMLBeans. La seva estructura la podem veure en la següent imatge.



Xsd base per al XmlBean GeusValidacioUsuariRetornDocument

El proxy d'accés al BancSecurityProviderServiceProxy s'implementa amb el patró Factory Pattern. Aquest proxy permet desacoblar la interfície de la implementació del servei. Això resulta de utilitat a l'hora de canviar d'implementacions o durant la fase de proves on es poden incloure implementacions dummy. El procés d'autenticació es recolza en els components:

- **BancSecurityProviderAuthenticationProviderImpl**: Aquest és el mòdul que connecta el proveïdor d'autenticació amb els serveis del framework de seguretat de WebLogic.

- **BancSecurityProviderLoginModuleImpl**: En aquest component es realitza el procés de login pròpiament dit. En aquest cas, aquest mòdul recull les dades de l'XMLBean retornat per GEUS en el mètode gusPeticioValidacioUsuari invocat per l'aplicació PORT. El mòdul de login JAAS és invocat en dues fases. En la primera fase s'invoquen els mètodes initialize i login. En la segona fase es criden als mètodes commit o abort:

    - **initialize**: Estableix les referències al Subject i al CallbackHandler. Aquest últim component s'utilitza per passar paràmetres de credencials de l'usuari al mòdul. En aquest cas, es defineix un callbackhandler per tal de passar per paràmetre les dades retornades per GEUS a través de PORTAL.

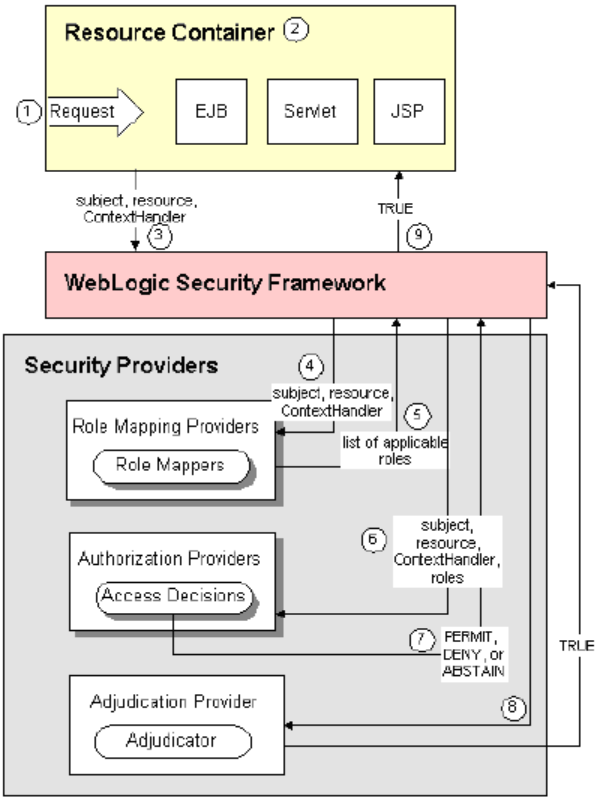
    - **login**: Realitza el login pròpiament dir, és a dir, crea el Subject i crea els principals. Si tot va bé retorna true i en cas de que es produeixi algun error llença una LoginException.

    - **commit**: Assigna els principals calculats anteriorment al Subject i retorna true.

    - **abort** / **logout**: Esborren els principals del Subject.

4.3.2.c Descripció capa per al cas d'ús d'autorització

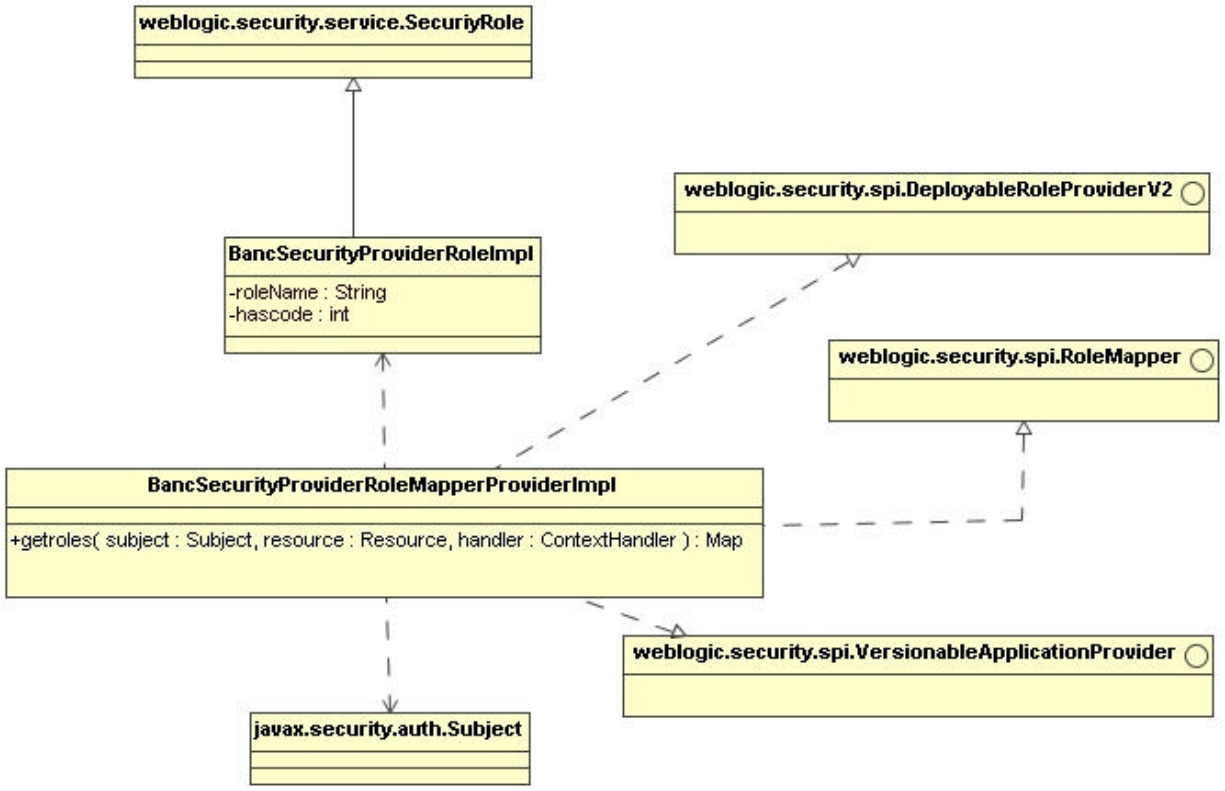
La següent imatge mostra el procés d'autorització.



Procés d'autorització

Bàsicament es tracta de que el proveïdor d'autorització de WebLogic Server decideix si la petició d'accés al recurs (EJB / Aplicació de Web) ha de ser autoritzada o no. Per tal de respondre a aquesta qüestió, el proveïdor d'autorització utilitza el Subject que ha estat calculat pel proveïdor d'autenticació i els rols associats a l'usuari que són calculats pel proveïdor de mapes de rols.

La següent imatge mostra els principals components del proveïdor de mapes de rols. Bàsicament la funcionalitat d'aquest component és retornar els rols associats a l'usuari que es troben emmagatzemats en el principal anomenat GeusValidacioUsuariRetornDocumentPrincipalImpl.



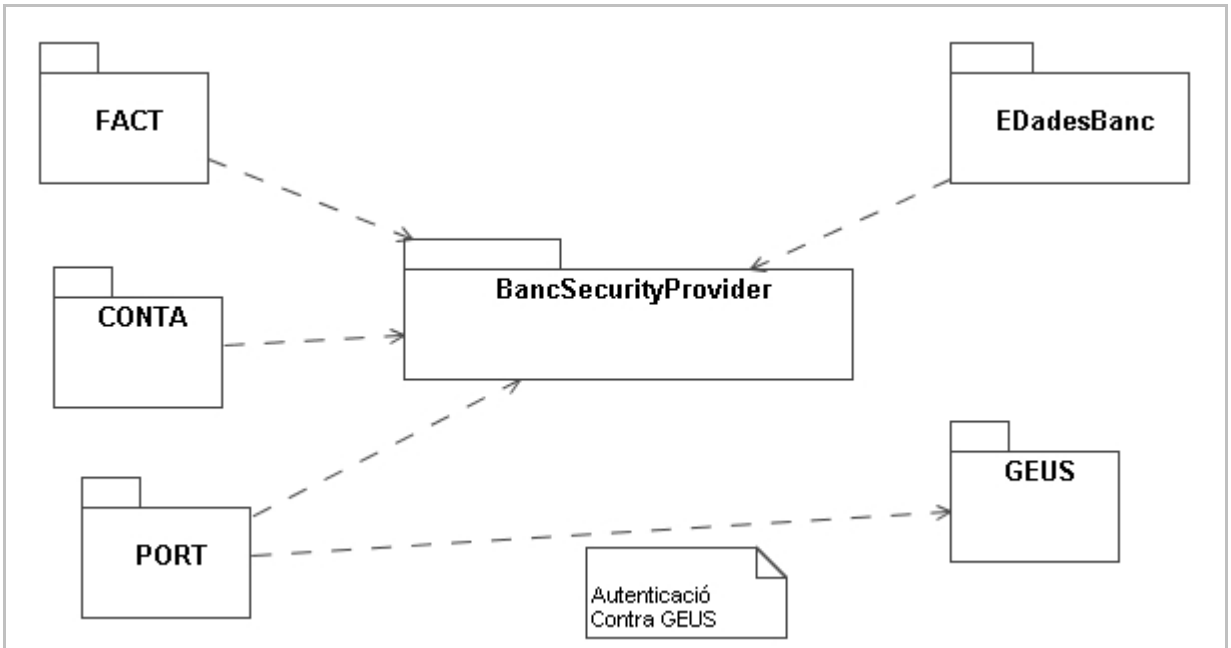
Components per al proveïdor de mapes de rols

El procés de resolució del mapa de roles es recolza sobre els següents components:

- **BancSecurityProviderRoleImpl**: Aquest component modela un role.
- **BancCatSecurityProviderRoleMapperProviderImpl**: Aquest és el mòdul que connecta el proveïdor d'autenticació amb els serveis del framework de seguretat de WebLogic i realitza el mapa dels rols:
  - **getRoles**: Retorna a la infraestructura de seguretat de WebLogic els rols associats a l'usuari:
    - **subject**: El Subject que conté els principals de l'usuari
    - **resource:** El recurs al que es vol accedir
    - **handler:** El component que dona accés al WebLogic Security Framework

4.3.3 Model de subsistemes de disseny (model de paquets)

En el següent diagrama podem veure la relació entre els diferents sistemes i el proveïdor de seguretat:



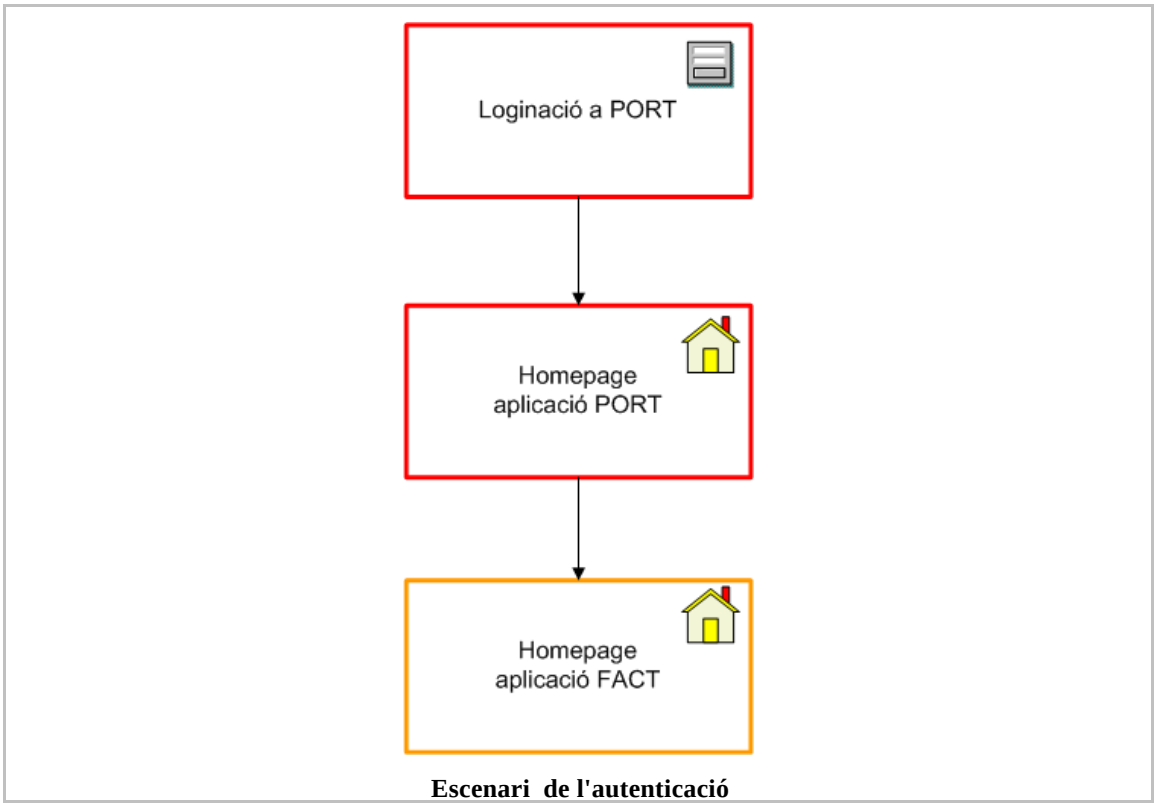
Model de subsistemes

4.4 Realització dels casos d'ús

4.4.1 Realització del Cas d'ús autenticació

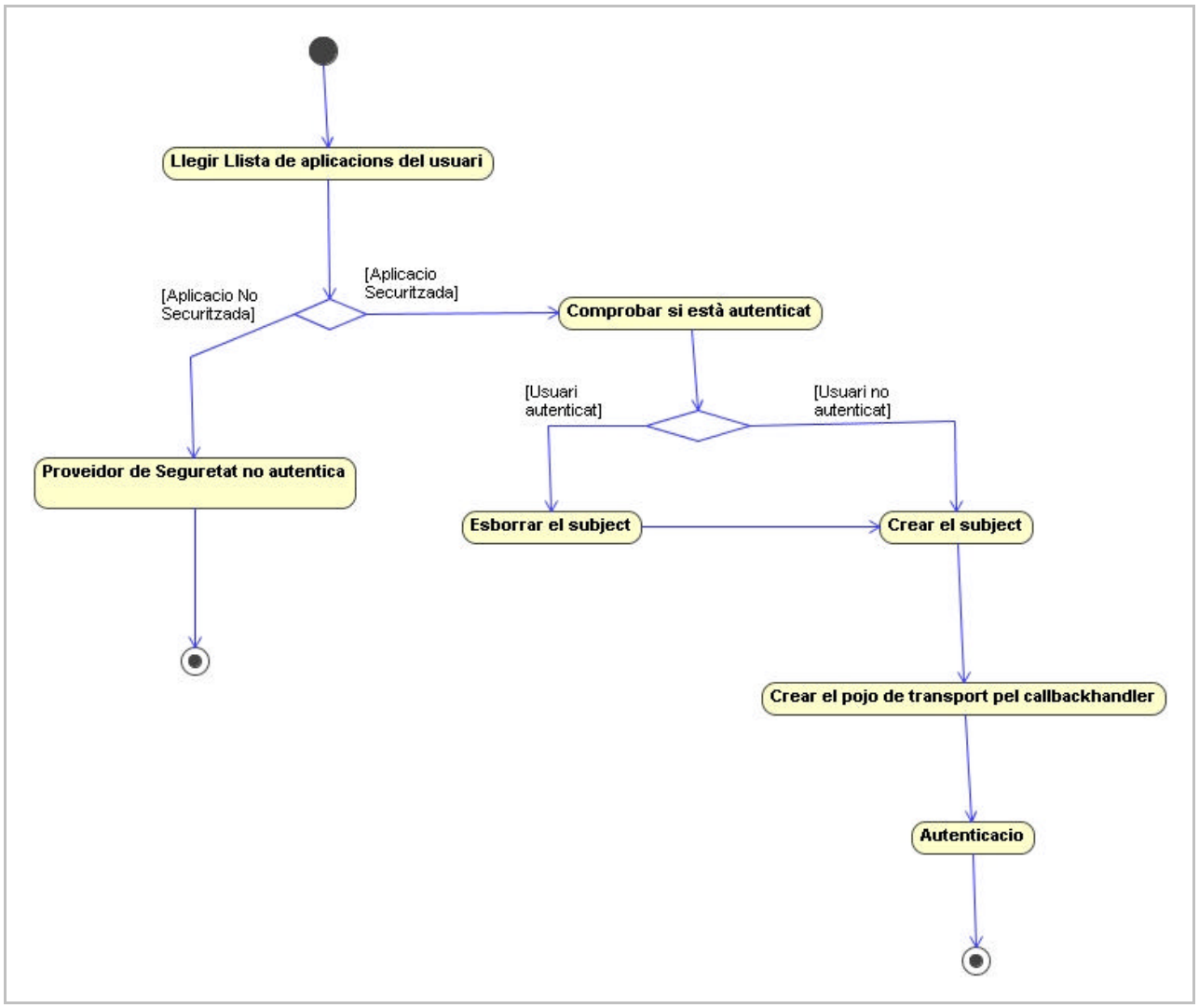
Hi ha un sol escenari per a l'autenticació JAAS en el proveïdor de seguretat. L'autenticació JAAS del proveïdor de seguretat es produeix posteriorment a que PORT cridi a GEUS per tal d'autenticar l'usuari. GEUS retorn un XMLBean () que conté tota la informació de l'usuari. Desde PORT es selecciona la aplicació a que es vol accedir (CONTA, FACT, CRM...) , que es passa per paràmetre al procés de autenticació, conjuntament amb l'XMLBean

La invocació al login del proveïdor es produeix en l'acció de login de PORT.

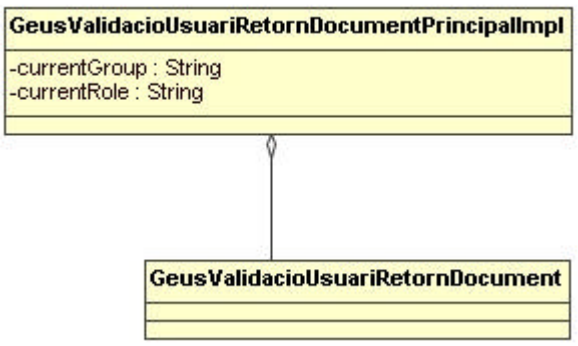


Escenari de l'autenticació

. La següent imatge mostra l'algorisme de l'autenticació:



El rol de l'usuari així com el grup es guarden a la custom principal GeusValidacioUsuariRetornDocumentPrincipalImpl. El grup es guarda a l'atribut currentGroup, el rol a l'atribut currentRole



Custom principal per al Subject de l'usuari

A més a més del custom principal anterior, es crea la principal associat a l'usuari anomenat WLSUserImpl. A aquest principal s'associa el codi de l'usuari que es troba a GeusValidacioUsuariRetornDocument al qual s'ha de accedir utilitzant els corresponents mètodes accessors. A més a mes, s'han de crear els grups. El principal corresponent al grup s'anomena WLSGroupImpl. A aquest principal s'associa el grup

de l'usuari que ha seleccionat en l'aplicació i que, a més a més, es guarda a la custom principal GeusValidacioUsuariRetornDocumentPrincipalImpl.

4.4.2 Realització del Cas d'ús d'autorització

L'autorització la proveeix automàticament WLS1 / WLS2 en funció de les dades del Subject de l'usuari i els roles que calcula el proveïdor de mapes de roles. El càlcul del mapa és simple, és a dir, el mètode de getRoles del proveïdor de mapes de rols ha de retornar el currentRole emmagatzemat a la custom principal GeusValidacioUsuariRetornDocumentPrincipalImpl. Per tant, fa la cerca d'aquest principal al Subject, és a dir, Subject.getPrincipals() i buscar la custom principal per retornar el role associat.

Tant a les aplicacions Web com als components EJB, podem protegir recursos creant associacions entre els recursos i els roles que poden accedir a aquests recursos. A la següent imatge tenim un exemple de protecció d'un recurs Web, en aquest cas, es tracten de tot el conjunt d'URL's que comencin per /hello:

```
<security-constraint>
    <display-name>security constrain</display-name>
    <web-resource-collection>
        <web-resource-name>say hello</web-resource-name>
        <url-pattern>/hello/*</url-pattern>
        <http-method>GET</http-method>
        <http-method>POST</http-method>
    </web-resource-collection>
    <auth-constraint>
        <role-name>role1</role-name>
    </auth-constraint>
    <user-data-constraint>
        <description/>
        <transport-guarantee>NONE</transport-guarantee>
    </user-data-constraint>
</security-constraint>
```

Com podem veure, la política de seguretat estableix que aquesta restricció de seguretat està associada a tots aquells usuaris que tinguin el role role1.

```
<security-role>
    <role-name>role1</role-name>
</security-role>
<!--  end role definition -->
```

En el cas dels EJB's tenim la definició de la seguretat declarativa es realitza fixant els següents tipus de valors en els descriptors ejb-jar.xml i weblogic-ejb-jar.xml:

```
<assembly-descriptor>
    <security-role>
        <role-name>manger</role-name>
    </security-role>


    <security-role>
        <role-name>east</role-name>
    </security-role>

    <method-permission>
        <role-name>manager</role-name>
        <role-name>east</role-name>
        <method>
            <ejb-name>accountsPayable</ejb-name>
            <method-name>getReceipts</method-name>
        </method>
    </method-permission>
...

</assembly-descriptor>
```

Configuració seguretat declarativa a l'ejb-jar.xml

```
<security-role-assignment>
    <role-name>manager</role-name>
    ...
</security-role-assignment>
```

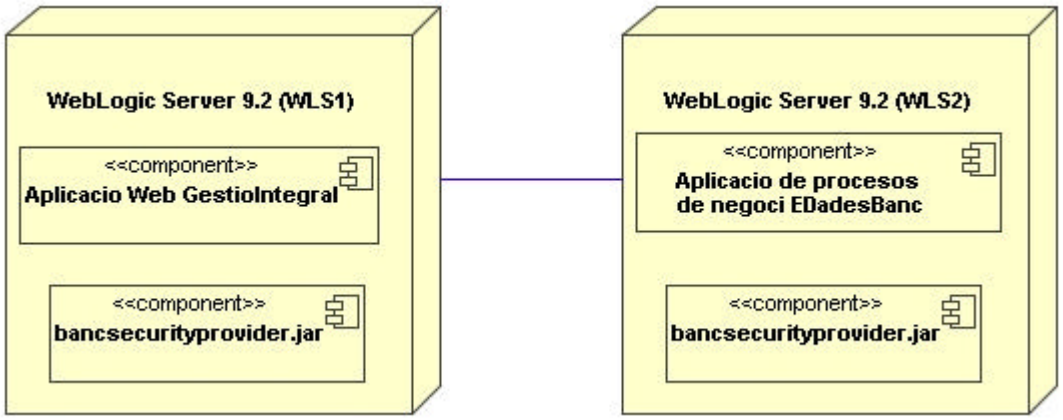
Configuració seguretat declarativa al weblogic-ejb-jar.xml

4.5 Especificacions organitzatives

4.5.1 Especificació d'operació i seguretat

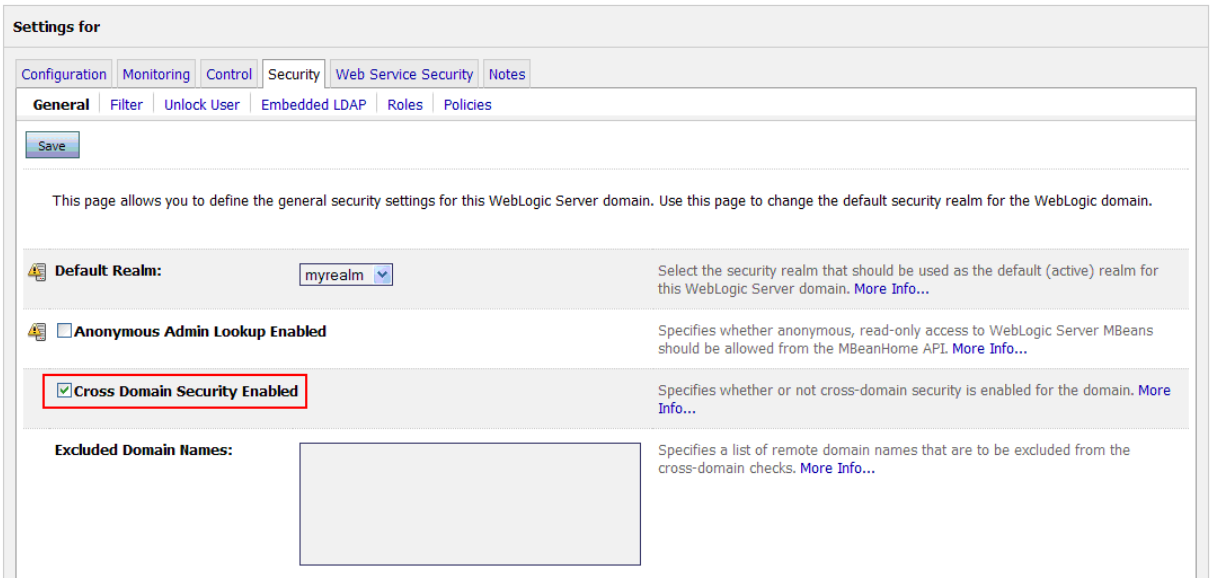
4.5.1.a Confiança entre els dominis WLS1 / WLS2

Hi ha aplicacions de GestioIntegral.Cat que tenen dos parts diferenciades. Per una banda es despleguen les aplicacions de Web sobre un domini WLS1 i per altre banda, es despleguen les aplicacions de processos de negoci sobre un altre domini WLS2. L'usuari s'autentica a l'aplicació de Portal que corre sobre WLS1. Una vegada l'usuari es troba autenticat, la seva informació de seguretat, és a dir el seu *Subject* serà propagat en les crides cap als components que es troben desplegats en el domini WLS2.

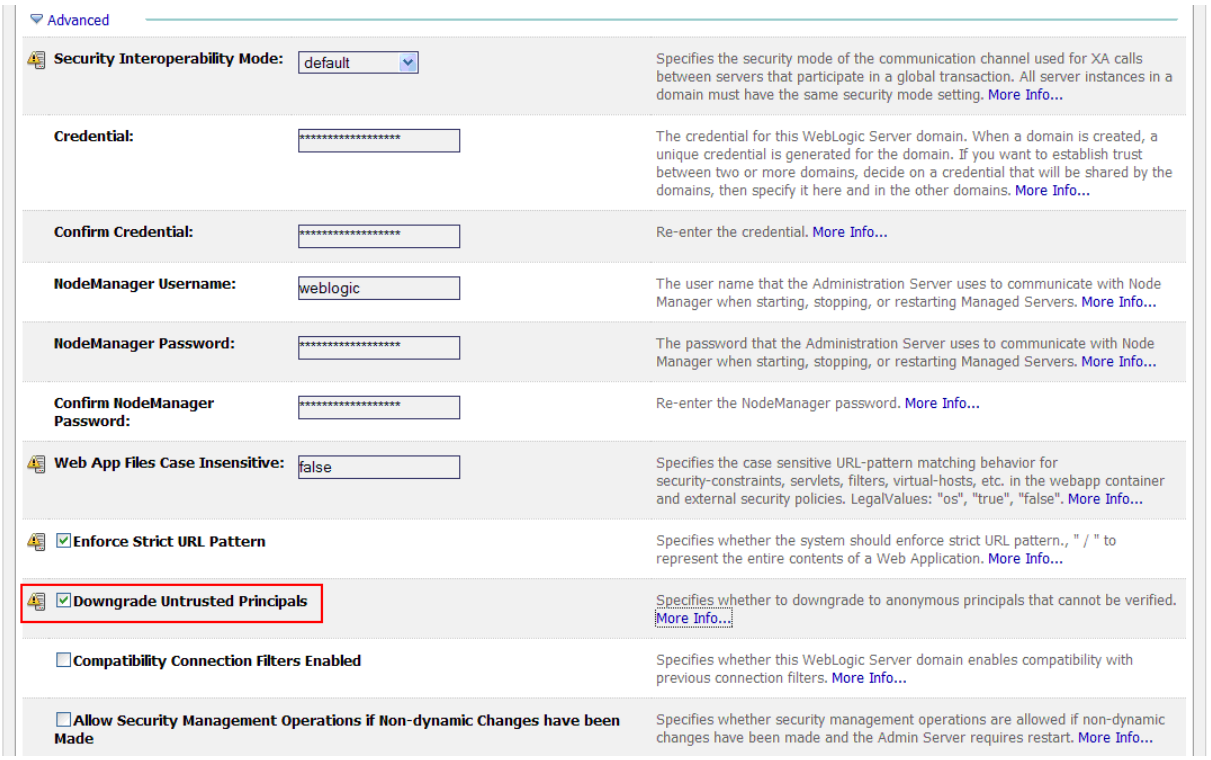


Desplegament del security provider

Les següents imatges mostren com activar la propagació de seguretat entre dominis. Tots aquells que tinguin la mateixa credencial de domini seran confidents entre tots ells.



Activar la seguretat entre els dominis WLS / WLI



Activar la seguretat entre els dominis WLS / WLI

4.5.1.b Restricció de codificació en les classes SSPI's

En les classes d'implementació del proveïdor d'autenticació o del proveïdor de mapes de roles no es pot escriure codi que requereixi a la seva vegada cap tipus de check de seguretat. En aquest cas es produeix una situació recursiva entre aquests components i la infraestructura de seguretat de WebLogic.

5. IMPLEMENTACIÓ.

5.1 Requeriments de maquinari

- Per tal de executar o compilar les dues aplicacions, es recomana utilitzar com a mínim 2GB de memòria.

5.2 Requeriments de programari.

- Java 1.5 o superior
- Servidor d'aplicacions i contenidor web WebLogic Server 9.2.1 o 9.2 amb el patch identificat pel CR CR294824 mitjançant l'eina anomenada BEA SmartUpdate que es troba al menú d'instal·lació de WLS 9.2. En qualsevol de les dues versions s'usa Java EE 5
- Les aplicacions de suport securitzades s'han provat en Windows 2000 , XP i en Red Hat
- Internet Explorer 6.0 sp1 i Firefox 2.0.0.9
- Jakarta Ant. Per automatitzar el procés de desplegament i generar els fitxers jar i ear de l'aplicació del proveïdor de seguretat, el seu client i l'aplicació de Test.
- EJB 3.0
- Framework Apache struts (MVC) per al desenvolupament de la capa web. Struts 1.2.4 (inclou Tiles)
- Planes jsp, amb fulles d'estil css i execució de javascript. Per tant els navegadors que s'usin han de permetre la seva execució.
- Hibernate Tools per generar les classes hibernate, i els seus arxius de configuració
- SGBD, Oracle Express v10.2.0.1

5.3 Eines de desenvolupament.

5.3.1 Aplicacions de Suport

- Les aplicacions GestioIntegral i EdadesBanc s'han construït usant Workshop 9.2 de Bea Systems sota Windows 2000. Es tracta de un entorn de desenvolupament gratuït, que va dins del WebLogic Plattform

- La generació automàtica de Hibernate s'ha fet amb Hibernate Tools Plugin

- La generació automàtica del codi dels xml beans, s'ha fet usant WebLogic Workshop

5.3.2 Proveïdor de seguretat

- BancSecurityProvider (aplicació de seguretat) , BancSecurityProviderClient (client) i BancSecurityProviderTestCase (aplicació de test) s'han construït usant Eclipse 3.1. sota Windows XP

- La compilació dels tres projectes s'ha fet amb Jakarta Ant versió 1.6.5

5.4 Configuració de WebLogic Server

- Abans de crear un Data Source, s'ha de crear la taula DADES_BANCARIES (veure annex) i la seqüència SEQ_COMPTEBANCARI (veure annex)

- S'ha de crear un domini per les aplicacions. Suggerim dir-li edadesbanc_9001 o bé gestiointegral_7001. Gestió integral surt per el port 7001 i edadesbanc per el port 9001.

- Abans de desplegar les aplicacions, s'ha de crear un Data Source:

    - Nom del data source: jdbc/PoolEDADESBANC

- Driver : Oracle Driver Thin versions 9.0.1, 9.2.0, 9.10

5.5 Aplicacions de Test

Existeixen dos projectes importants per al control de qualitat:

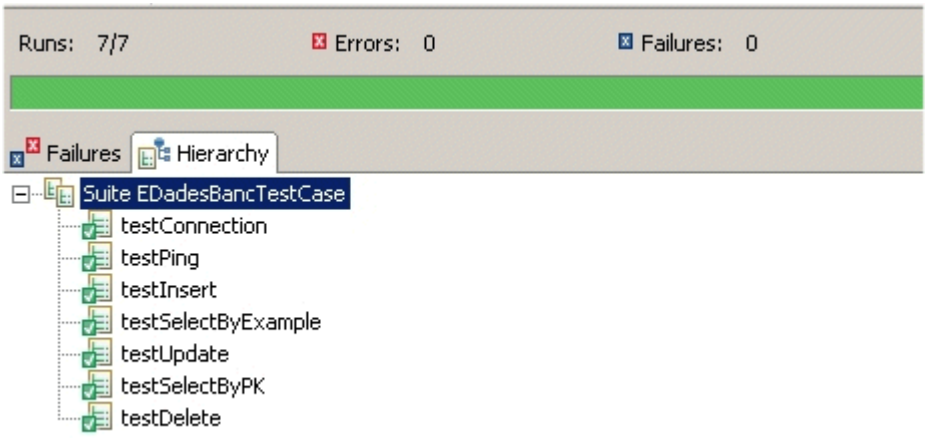
5.5.1 EdadesTest

Classes Java i JUnit amb codi de proves del accés a base de dades amb Hibernate.

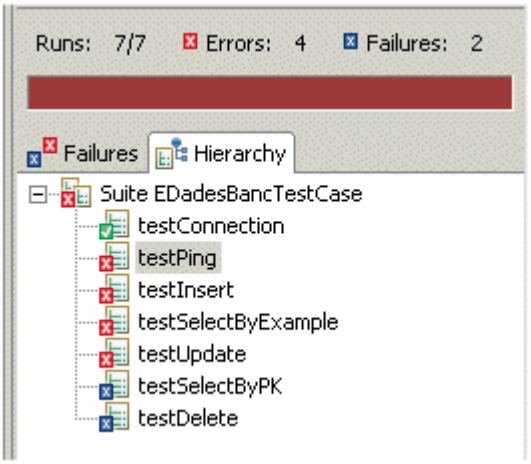
***EDadesBancBeanTestCase**:* Classe de proves unitàries que executa una bateria de funcions disponibles al DAO de EDadesBanc.
Executa la suite EdadesBancTestCase que prova el manteniment de la taula dades_bancàries:
Es connecta, insereix un registre, fa una select d'aquest registre, un update, una select per primary key i finalment, l'esborra

Aquest test s'ha de fer amb anterioritat a la securització dels entorns. Una vegada securitzats, l'execució de suite EdadesBancTestCase, no funcionarà donat que intentarà accedir al EJB securitzat sense estar creat el context de seguretat.

Execució del Junit abans de la securització:



Execució del Junit després de la securització



Motiu:

```
java.rmi.AccessException: [EJB:010160]Security Violation: User:
'<anonymous>' has insufficient permission to access EJB: type=<ejb>,
application=EDadesBanc, module=EDadesBanc, ejb=EDadesBancBean,
method=ping, methodInterface=Remote, signature={java.lang.String}.
```

5.5.2 BancSecurityProviderTestCase

5.5.2.a Descripció

El projecte BancSecurityProviderTestCase correspon a un test case per provar el proveïdor de seguretat.

La compilació del projecte BancSecurityProviderTestCase.ear ha de ser desplegada sobre dos dominis diferents configurats amb el proveïdor de seguretat i amb la relació de confiança entre ells dos.

5.5.2.b Configuració

A les dues instàncies de Weblogic (els dos dominis) :

Desplegar la aplicació: Seleccionar el directori de windows on tenim el ear i desplegar-lo . Assabentar-nos de que resta Active (fixem-nos que tenim una aplicació al domini on hi ha desplegat Gestió Integral i l'altre desplegat on hi ha EdadesBanc):

Per fer-la active tenim que dir que prémer el botó start i seleccionar Servicing all request:

5.5.2.c Test

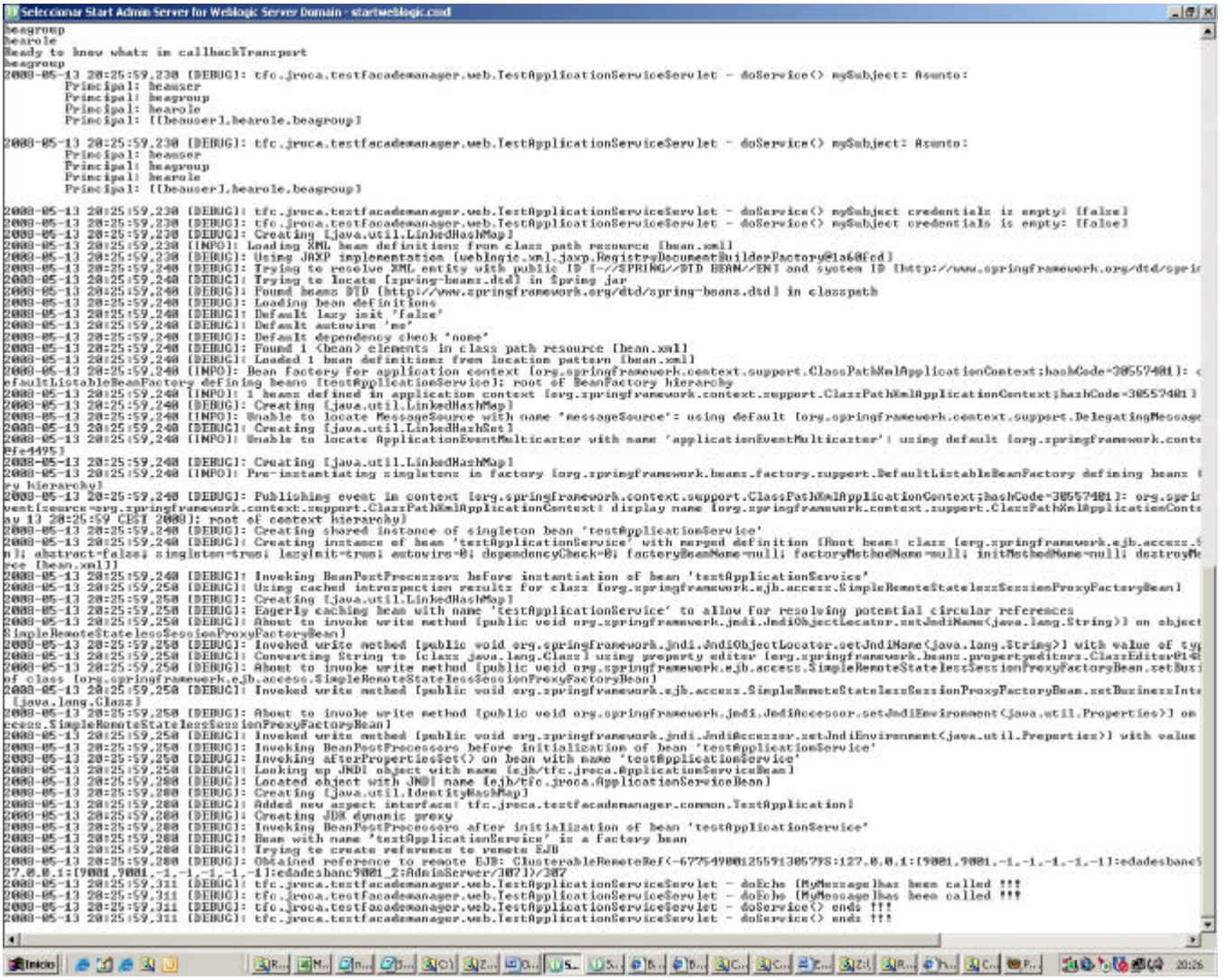
Desde el domini gestiointegral7001.

Primer provem de fer una **petició sense autenticació**:
http://127.0.0.1:7001/BancSecurityProviderTestCase/aps

Resultat:

2008-05-17 20:18:23,257 [DEBUG]: Remote service [**ejb/tfc.jroca.ApplicationServiceBean**] threw exception java.rmi.AccessException: [EJB:010160]**Security Violation: User: '<anonymous>' has insufficient permission to access EJB:** type=<ejb>, application=BancSecurityProviderTestCase#1.0, module=BancSecurityProviderTestCase-ejb.jar, ejb=tfc.jroca.testfacademanager.ejb.TestApplicationServiceBean, method=create, Etc.

A continuació fem una **petició amb autenticació**
http://127.0.0.1:7001/BancSecurityProviderTestCase/aps?auth=true
Resultat correcte!!!



6. Conclusions.

El model actual de seguretat que presentem compleix els estàndards J2EE de seguretat sense afectar a la resta d'aplicacions que s'estiguin executant dins el framework GestioIntegral.cat

Amb la utilització de la peça de seguretat que composa aquest projecte resulta molt més senzill autenticar i autoritzar l'accés als diferents mòduls de qualsevol projecte.

Aquest proveïdor de seguretat es només una encapsulació per complir l'estàndard. J2EE. Tota la feina de gestió d'usuaris, grups i rols de es gestiona desde l'aplicació al ús per aquesta feina,que en el nostre projecte hem anomenat GEUS.

El sistema tecnològic que presentem , resol la problemàtica de negoci que es planteja des del punt de vista de seguretat.

La seguretat que presentem és un servei d'infraestructura que pot ser comú a qualsevol mòdul perquè la problemàtica bàsica a resoldre (els casos d'ús del model de seguretat) sempre és la mateixa.

Totes les aplicacions que s'executen sobre BEA WebLogic Server es poden beneficiar directament del proveïdor de seguretat perquè poden disposar de la informació d'usuaris i grups.

7. Glossari.

Concepte	Descripció
Aplicació Servidora	Al pla de projecte explicàvem que es tractava de l' aplicació que ofereix serveis d'accés a comptes bancaris En endavant anomenarem a aquesta aplicació EDadesBanc
Coarse-grained authorization	Autorització automàtica que s'aplica als components J2EE
CONT	Aplicació de Comptabilitat de GestioIntegral.cat que es vol que es vol securitzar, donat que té accés al manteniment de Bancs de EDadesBanc
Core Business	Al pla de projecte explicàvem que es tractava de l'aplicació que basa la seva seguretat en un determinat model de seguretat (LDAP etc). Consumeix serveis de la aplicació servidora. En endavant anomenarem a aquesta aplicació Gestió Integral
CRM	Aplicació de Seguiment de clients de GestioIntegral.cat no securitzada
EDadesBanc	Veure Aplicació Servidora
FACT	Aplicació de Facturació de GestioIntegral.cat que es vol securitzar, donat que té accés al manteniment de Bancs de EDadesBanc
Fine-grained authorization	Validacions d'autorització amb lògica de negoci i amb molt més grau de detall que l'autorització automàtica
Gestió Integral	Veure Core Business
GEUS	Mòdul de gestió d'usuaris. Controla tots els usuaris de GestioIntegral.cat i és l'únic mòdul de seguretat disponible i centralitza el accés al LDAP de la empresa
Grups	Son les agrupacions, a les que pertany el usuari. Un usuari pot pertànyer a 0 o a n grups
Identity	Identificació d'un usuari amb el servidor
JAAS	Java Authentication and Authorization Service: Consisteix en una API per tal de que les aplicacions Java utilitzin serveis d'autenticació i autorització de manera estàndard. http://java.sun.com/products/jaas/
PORT	Aplicació de Portal de GestioIntegral.cat. Punt de entrada, que nosaltres simularem per tal de poder desenvolupar el proveïdor de seguretat.
Principal	Nom associat a un Subject. Un subject pot estar format per un conjunt de principals. Només s'associen al subject si s'ha

	autenticat amb èxit
Recursos	Diferents mòduls funcionals relacionats entre si, que composen el Core Business(GestioIntegral.CAT).
Roles	Cada persona dins un grup, actua segons el rol que té assignat. Nivell de permís que té un determinat usuari. Perfil d'un usuari
Role-base authorization	Sistema J2EE de regles d'autorització basat en rols que utilitza els descriptors de desplegament J2EE per indicar les regles d'autorització segons diferents rols.
Role Mapping Provider	Proveïdor de seguretat de mapeig de rols del projecte
Single Sign-on	Autenticació unificada que permet que un usuari només s'identifiqui un cop.
SSO	Mecanisme d'autenticació unificada.
SSPI's	Components de WebLogic Server que comuniquen els custom providers amb el WebLogic Security Framework
Subject	Identificador JAAS de una petició. Una o més identities (veure identity) d'un usuari poden estar relacionats amb un mateix subject.
WLS1	Domini WebLogic on es desplegat GestioIntegral
WLS2	Domini WebLogic on es desplegat EdadesBanc

8. Bibliografia.

Llibres:

- **J2EE Applications and BEA WebLogic** Server; Girdley,Woolen i Emerson. Ed. Prentice Hall
- **J2EE Design Considerations for Weblogic Server**; Bea Systems White Paper
- **Java 2 Enterprise Edition 1.4 Bible**; James McGovern, Rahim Adatia, Yakov Fain,Jason Gordon, Ethan Henry, Walter Hurst,Ashish Jain, Mark Little, Vaidyanathan Nagarajan,Harshad Oak, Lee Anne Phillips. Ed. Wiley
- **J2EE Tutorial**; Stephanie Bodoff, Dale Green, Kim Haase, Eric Jendrock, Monica Pawlan, Beth Stearns. Ed. Addison Wesley
- **Weblogic Definitive Guide**; Avinash Chugh i Jon Mountjoy. Ed. Oreilly
- **J2EE Best Practices;** Darren Broemmer. Ed.Wiley
- **Programming Jakarta Struts** ; Chuck Cavaness. Ed. Prentice Hall
- **Struts in Action**; Ted Husted, Cedric Dumoulin, George Franciscus i David Winterfeldt. Ed.Manning
- **Patrones de diseño aplicados a Java**; Olaf Maasen . Ed: Prentice Hall
- **Java 2 Enterprise Edition 1.4 Bible**; James McGovern, Rahim Adatia, Yakov Fain,Jason Gordon, Ethan Henry, Walter Hurst,Ashish Jain, Mark Little, Vaidyanathan Nagarajan,Harshad Oak, Lee Anne Phillips. Ed. Wiley

- **Designing Enterprise Applications with J2EE Plattofrm;** Inderjeet Singh, Beth Stearns, Mark Johnson, and the Enterprise Team. Ed: Addison-Wesley
- **Java 2EE and XML Development;** Kurt A. Gabrick I David B. Weiss. Ed:Manning
- **Programming WebLogic Enterprise JavaBeans;** Bea White Paper

Internet:

- http://java.sun.com/products/jaas
- http://edocs.bea.com/wls/docs92/security.html
- http://java.sun.com/blueprints/corej2eepatterns
- http://java.sun.com/j2ee/tutorial/1_3-fcs/doc/Security.html
- http://edocs.bea.com/wls/docs92/dvspisec/index.html
- http://e-docs.bea.com/wls/docs92/secintro/model.html
- http://e-docs.bea.com/wls/docs92/secintro/concepts.html#wp1123184
- http://www.coresecuritypatterns.com/patterns.htm

9. ANNEXOS

9.1 Annex 1. Script de base de dades.

El manteniment dels comptes bancaris es fa mitjançant una única taula i una seqüència

Taula:

```
create table DADES_BANCARIES
(
  IDP         NUMBER(10),
  COGNOMS   VARCHAR2(60),
  NOM      VARCHAR2(30),
  CCC_ENTITAT VARCHAR2(4),
  CCC_OFICINA VARCHAR2(4),
  CCC_DC      VARCHAR2(2),
  CCC_CTA     VARCHAR2(10)
);

-- Add comments to the table
comment on table DADES_BANCARIES
  is 'Dades bancàries de les persones: compte bancari de
domiciliació.';
-- Add comments to the columns
comment on column DADES_BANCARIES.IDP
  is 'Identificador de la persona o empresa';
comment on column DADES_BANCARIES.COGNOMS
  is 'Cognoms  de la persona';
comment on column DADES_BANCARIES.NOM
  is 'Nom de la persona';
comment on column DADES_BANCARIES.CCC_ENTITAT
  is 'Identificador de l'entitat bancaria, compte domiciliació ';
comment on column DADES_BANCARIES.CCC_OFICINA
  is 'Identificador oficina bancaria compte domiciliació ';
comment on column DADES_BANCARIES.CCC_DC
  is 'Dígits de control compte domiciliació ';
comment on column DADES_BANCARIES.CCC_CTA
  is 'Núm de compte domiciliació ';
-- Create/Recreate primary, unique and foreign key constraints
alter table DADES_BANCARIES
  add constraint PK_DATBAN primary key (IDP);
```

Seqüència:

```
CREATE SEQUENCE SEQ_COMPTEBANCARI
  START WITH 1
  MAXVALUE 999999999999999999999999999
  MINVALUE 0
  NOCYCLE
  NOCACHE
 NOORDER;
================================================================
```