LINEE GUIDA PER L'INTEGRAZIONE CON ORACLE LOGIN SERVER

PARTE 1: APPLICAZIONI WEB JAVA

Ver. 1.0

Data	Versione	Descrizione	Cap. /Sez. modificati
novembre 2003	1.0	Nascita del documento	tutti

Indice

1 Introduzione

Scopo di questo documento è fornire una serie di semplici linee guida per l'integrazione nativa di applicazioni web in ambito MEF (sia internet che intranet) con l'infrastruttura centralizzata che costituisce il punto unico di accesso alle applicazioni ed ha in carico l'autenticazione degli utenti e la gestione dei profili.

Tale infrastruttura, si basa sui componenti PORTAL e LOGIN SERVER della suite Oracle 9iAS .

Si precisa che il termine SSO usato nel seguito indica la funzionalità di logon unico, metre i termini "Login server" e "Single Sign On (SSO) Server" individuano i prodotti software Oracle nei 2 diversi release che realizzano la funzionalità di logon unico.

Completano questo documento alcuni esempi di codice java. Si precisa che tale codice costituisce un esempio e viene qui allegato "as is".

2 Descrizione del Contesto

L'infrastruttura prevede, come riportato schematicamente nella figura sottostante, l'adozione di due Portal e di un unico Login Server.

Il Portal esterno costituisce il punto unico di ingresso per tutti gli utenti che vogliano accedere ad applicazioni abilitate a fornire servizi ad utenze internet. Il Portal interno è il punto di raccordo delle applicazioni presenti nella Intranet e prevede, a differenza di quello esterno, la presenza di un area di contenuti pubblici.

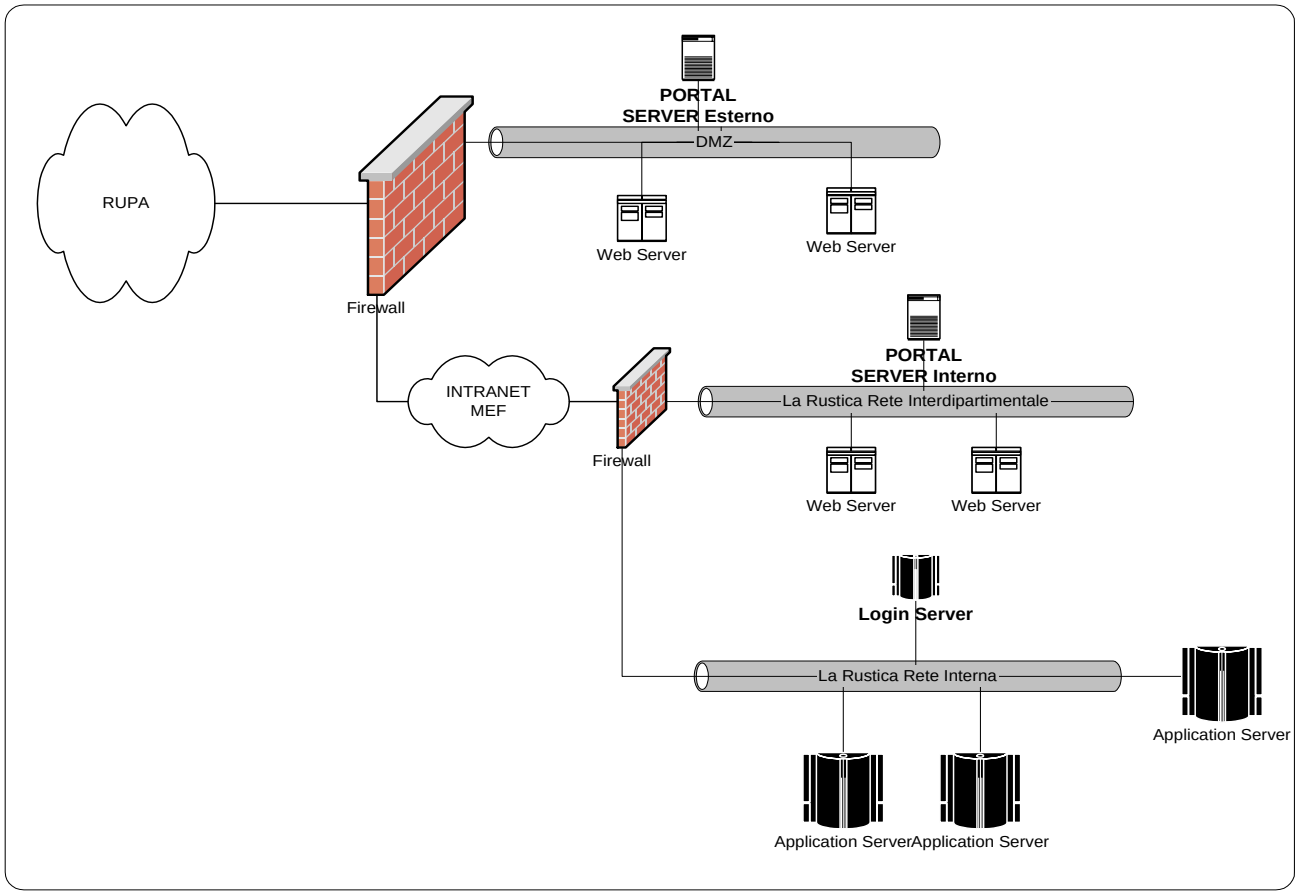
Il Portal esterno è accessibile sia direttamente (con un proprio URL) sia come link dal sito del Ministero dell'Economia e Finanze, ed espone come prima pagina la maschera che richiede all'utente di inserire le proprie credenziali. Questa scelta deriva dalla volontà di non esporre alcuna informazione prima che l'utente si sia identificato. A fronte di una corretta identificazione (lo scambio di credenziali avverrà in una connessione sicura) verrà proposto all'utente un menù delle applicazioni cui è autorizzato.

Questo menù è costituito da una "content area" gestita dal Portal che costruisce, in modo dinamico, in base ai gruppi cui appartiene l'utente, l'elenco delle applicazioni cui l'utente ha accesso.

Il servizio di identificazione ed autorizzazione dell'utente è fornito del Login Server che è l'*access manager* fornito dalla suite Oracle 9iAS. Questo componente centralizza le funzioni di gestione delle utenze e dei gruppi di utenze e quelle di impostazione delle politiche sulle password (lunghezza, formato, scadenza etc) e si prevede di richiamarlo indifferentemente sia quando un'applicazione viene raggiunta tramite il Portal sia quando è richiamata direttamente.

Si vuole ribadire infatti che uno degli obiettivi che si vuole perseguire tramite questa infrastruttura è che le nuove applicazioni siano realizzate in modo da demandare ad una entità esterna il compito dell'autenticazione dell'utente e che si introduca, nello sviluppo dell'applicazione, il concetto di ruolo, come attributo dell'utente e che questo sia lo strumento con il quale l'applicazione controlla l'accesso alle funzioni ed ai dati (es funzionario, direttore, controller etc.).

Il Portal gestisce l'associazione tra le applicazioni ed i gruppi di utenze ad esse autorizzate.



Per maggior chiarezza, nel prosieguo del documento si adotta la stessa suddivisione delle applicazioni come prevista da Oracle. Rispetto ad essa le applicazioni si distinguono in:

- *partner application* - applicazioni che si integrano compiutamente con il Portal, demandando l'identificazione al Login server e che fruiscono delle funzionalità di Single Sign On fornite dal Portal stesso;
- *external application* - applicazioni, di solito preesistenti, che gestiscono un proprio "user repository" e che sono integrabili con il Portal ma non con le funzioni di Single Sign On.

Le applicazioni *external* sono destinate gradualmente ad essere rimpiazzate da applicazioni *partner*, per sostituzione o per manutenzione evolutiva.

Il documento si interessa dunque di tracciare le linee guida per la realizzazione di nuove *partner applications* e per l'aggiornamento da *external* a *partner* di quelle esistenti.

3 Architettura - Elementi Costitutivi

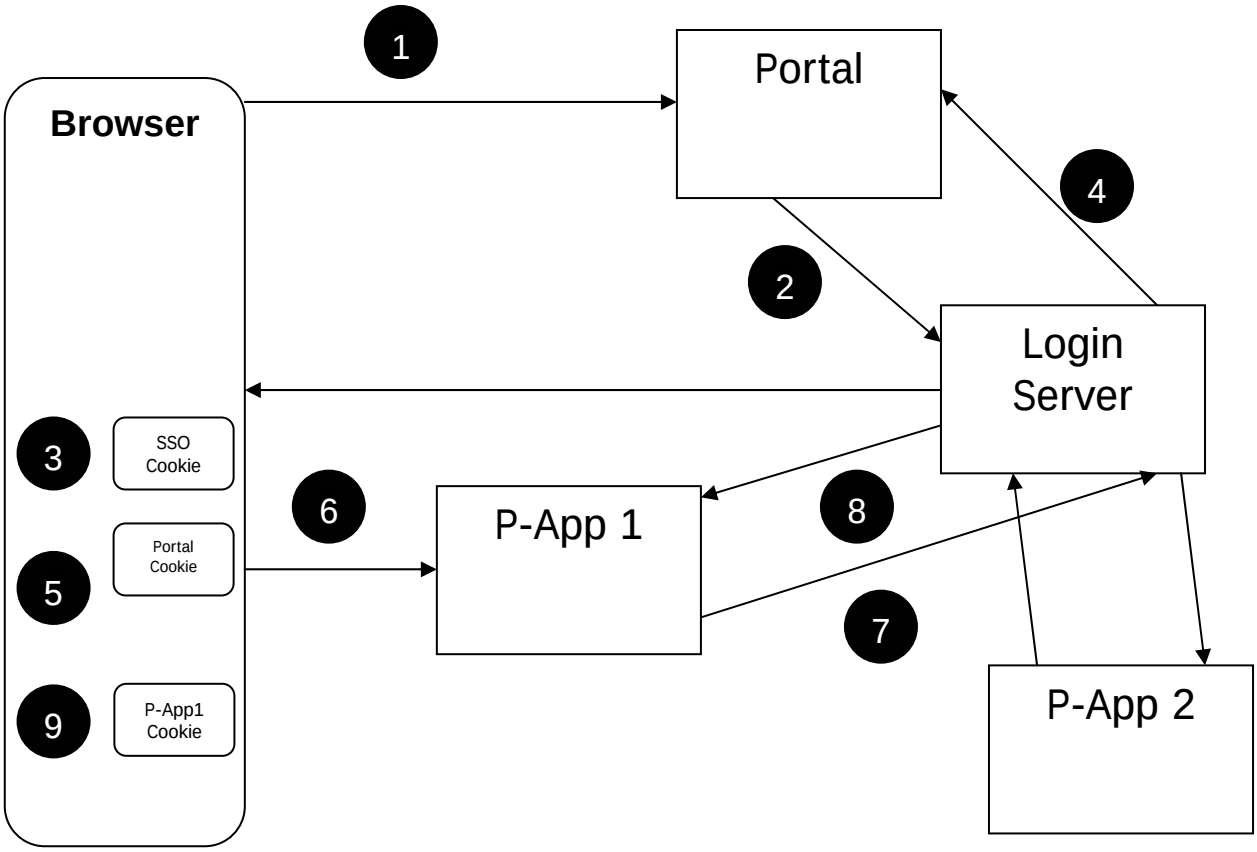
In questo paragrafo viene descritto in maniera dettagliata il flusso logico e funzionale della procedura di identificazione ed autenticazione degli utenti sia nel caso di accesso diretto all'applicazione sia di accesso tramite Portal Server.

3.1.1 Schema logico della procedura di login

Il principio di funzionamento della procedura è abbastanza complesso, dunque per maggiore semplicità e chiarezza si ritiene opportuno trattare separatamente i due casi principali che possono presentarsi.

3.1.2 L'utente accede tramite il portale alla Partner Application 1

Nella figura che segue è rappresentato lo schema logico del flusso in questa prima situazione.



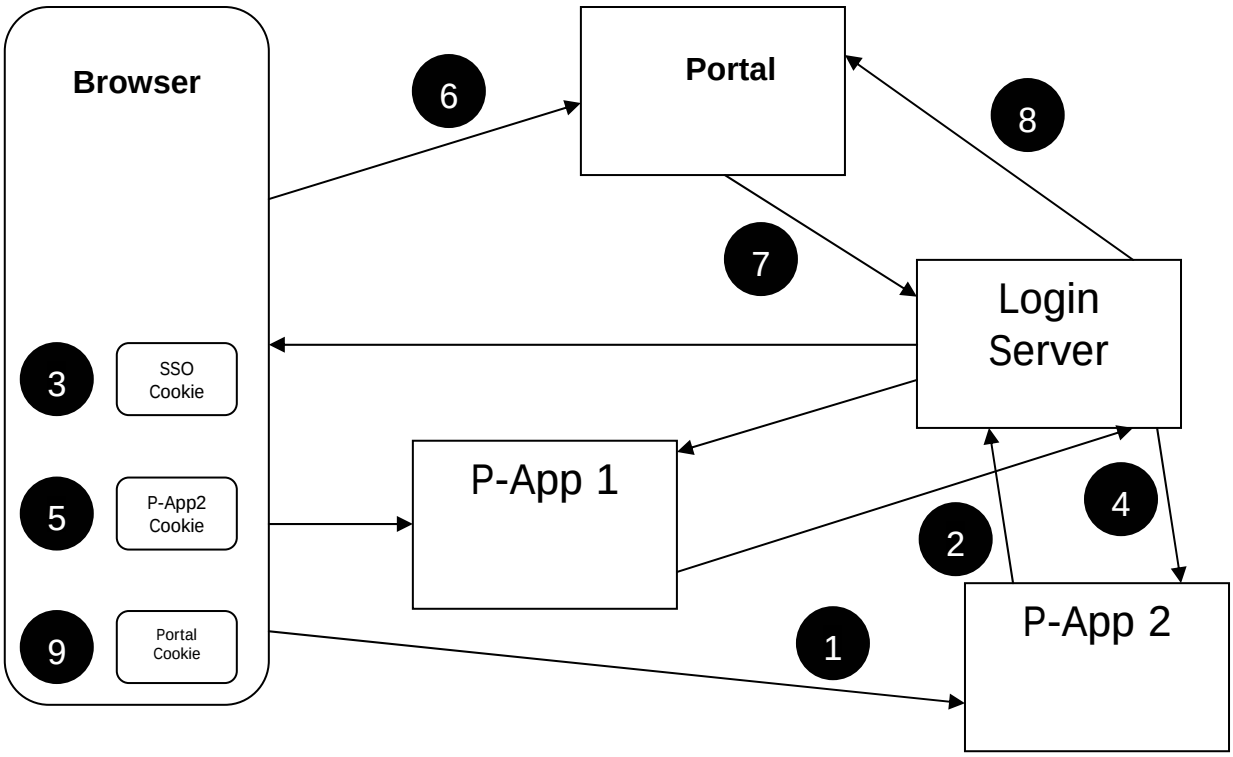
Il processo può essere riassunto come segue:

1. L'utente, sfruttando un generico browser, richiede al portale di essere riconosciuto e autenticato dal sistema per usufruire dei servizi cui è abilitato.
2. Il portale verifica che l'utente non sia già autenticato (verificando, tramite una funzione che convenzionalmente definiremo "A" che sul browser dell'utente non sia presente il cookie di sessione del portal) e quindi redirige la richiesta di autenticazione al login server. Per effettuare questa redirezione la funzione "A" interroga l'apposito schema ottenendo la locazione del Login Server e la chiave per crittografare la comincazione.
3. Il login server (eventualmente facendo ricorso ad un LDAP server) identifica l'utente e invia al browser client dell'utente un cookie di sessione (SSO cookie).
4. Il login server redirige l'utente verso la pagina di esito positivo del portal, quella che normalmente conterrà i link verso le applicazioni che l'utente ha il diritto di fruire includendo in un token eventualmente crittografato le informazioni sull'utente.
5. Il portale (tramite una funzione che convenzionalmente definiremo "B") invia un proprio cookie di sessione al browser client (quello che di solito contiene il time stamp).
6. l'utente chiede di utilizzare la Partner application 1 (P-App1) cliccando su uno dei link che gli viene proposto dal portale.
7. La P-App1 verifica se l'utente è già autenticato (verificando l'esistenza di un proprio cockie di sessione –funzione A-), quindi chiede al Login Server se l'utente è riconosciuto.
8. Il Login server verifica che il client abbia il cookie di sessione e risponde positivamente alla P-App1, inviando la userid dentro un token crittografato.
9. La P-App1 decrtiita il token e invia il proprio cookie di sessione (P-App1 cookie –funzione B-) al browser client. Ogni volta che l'utente, lavorando sull'applicazione 1, lancia una procedura viene eseguita la funzione A che verifica che la sessione non sia scaduta.

3.1.3 L'utente accede direttamente alla Partner Application 2

Nella seguente figura è rappresentato lo schema logico del flusso in questa diversa situazione.
Il processo può essere riassunto come segue:

1. L'utente, sfruttando un generico browser, richiede di accedere alla applicazione partner 2.
2. L'applicazione verifica che l'utente non sia già autenticato (tramite una funzione che convenzionalmente definiremo "A") e quindi redirige la richiesta al login server. Per effettuare questa redirezione la funzione "A" interroga l'apposito schema ottenendo la locazione del Login Server e la chiave per crittografare la comincazione.
3. Il login server (eventualmente facendo ricorso ad un LDAP server) identifica l'utente e invia al browser client dell'utente un cookie di sessione (SSO cookie).
4. Il login server redirige l'utente verso la pagina di esito positivo dell'applicazione partner 2, inviando nel contempo la userid dell'utente sotto forma di token crittografato.
5. L'applicazione partner 2 (tramite una funzione che convenzionalmente definiremo "B") decritta il token invia un proprio cookie di sessione al browser client (quello che di solito contiene il time stamp). Ogni volta che l'utente, lavorando sull'applicazione 2, lancia una procedura viene eseguita la funzione A che verifica che la sessione non sia scaduta.



Si fa notare che l'utente, pur non essendo passato per il portale ha comunque sul suo browser un il cookie di Single Sign On, e che se volesse collegarsi ad una altra applicazione partner o al portale, nessuna ulteriore autenticazione verrebbe richiesta, ma verrebbe fatto direttamente accedere all'applicazione o alla lista delle applicazioni autorizzate. Di seguito si descrivono i passi logici secondo i quali si comporta il sistema nel caso in cui l'utente voglia connettersi al Portale.

6. l'utente chiede di accedere al Portale cliccando su un link opportuno o digitando la relativa URL.
7. Il portale verifica se l'utente è già autenticato (verificando l'esistenza di un proprio cookie di sessione), quindi chiede al Login Server se l'utente è riconosciuto.
8. Il Login server verifica che il client abbia il cookie di sessione e risponde positivamente alla al portale.
9. Il portale invia il proprio cookie di sessione (Portal cookie) al browser client e presenta la pagina con l'elenco delle applicazioni autorizzate per l'utente. Da questo momento tutto prosiegue come indicato nel caso precedente.

Come si nota le due procedure sono esattamente analoghe. A tutti gli effetti il Portal si comporta come una partner application, e tale va considerato.
Questo meccanismo, apparentemente complesso, realizza una vera e propria funzionalità di single sign on, grazie alla quale all'utente viene richiesto di autenticarsi una sola volta per tutte.

In definitiva quello che caratterizza le partner applications sono le due funzioni che convenzionalmente abbiamo indicato come “A” e “B”, che vanno incorporate opportunamente nel codice applicativo e per le quali ORACLE fornisce i sample.

Schematicamente i compiti svolti dalle due funzioni sono i seguenti:
Funzione “A”:

- cerca il cookie dell`applicazione
- se il cookie viene trovato si procede con il rendering relativo all`utente collegato
- se il cookie non esiste si ha una redirect al Login Server
- questo a sua volta autentichera` l`utente e chiamera` la "procedura B" con un token utente, che impostera` il cookie ed eseguira` una redirect alla "procedura A", che a quel punto si trovera` il cookie di sessione applicativa impostato e potra` procedere quindi con il rendering.

Funzione “B”:

- Questa procedura si limita a settare tale cookie ed eseguire una redirect alla "procedura A", che si trovera` quindi tale cookie settato correttamente.

La potenziale presenza di diversi cookie sul client è completamente trasparente per l’utente e permette di realizzare controlli di sessione (e quindi, ad esempio, gestione dei timeout) distinti per applicazione.

Il cookie di single sign on (SSO) certifica l’avvenuta identificazione dell’utente e contiene le informazioni relative alla sessione con il login server. I cookie di applicazione contengono le informazioni di sessione relative all’applicazione.

4 Sviluppo di nuove applicazioni (Partner Application)

Brevemente, si ricorda che con il termine di partner application si intende una applicazione che delega al Login Server la funzione di autenticazione. L'applicazione contiene al suo interno codice che verifica se l'utente è già autenticato e, nel caso contrario, produce una re-direct alla "login form" del Portale, funzioni che convenzionalmente abbiamo indicato come "A" e "B".

La gestione della "login form", del codice che implementa la logica di autenticazione e del repository di utenti e password NON è compito della partner application che delega interamente tali funzioni al Login server.

La Oracle fornisce una libreria di classi Java, detta "Single Sign-On developer's kit" (SSO SDK) che semplifica l'integrazione di applicazioni sviluppate in tecnologia Java.

La versione iniziale del Single Sign On SDK è stata resa disponibile dalla Oracle tramite "Oracle technology network" (OTN): il sito ufficiale attraverso cui la Oracle pubblica gli strumenti resi disponibili agli sviluppatori.
Release successive, prodotte ad esempio per correggere eventuali problemi segnalati, sono disponibili attraverso il canale istituzionale del supporto, come parte del meccanismo generale di distribuzione delle patch release, o possono, su richiesta, essere fornite dalla Consulting.

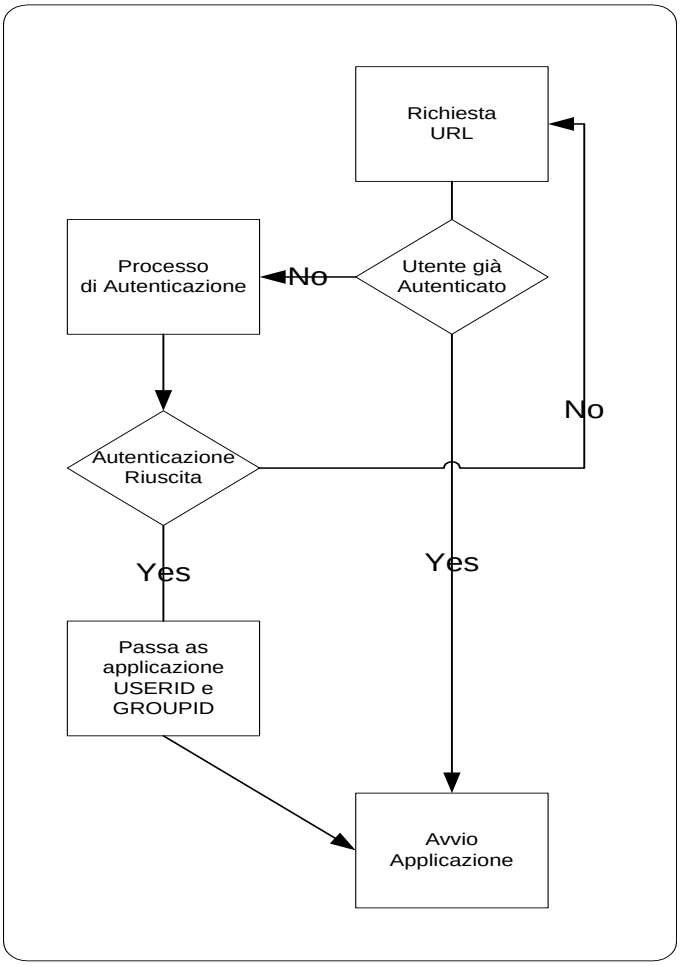
Nel seguito del documento sono riportate le linee guida per la programmazione di queste funzioni, comprensive di esempi di codifica (in allegato).

L'utilizzo del SSO SDK richiede che:

- Il database Oracle sia almeno aggiornato alla release 8.1.7.1.
- La release di Oracle Portal sia la 3.0.9.8
- La release di iAS sia la 1.0.2.2.

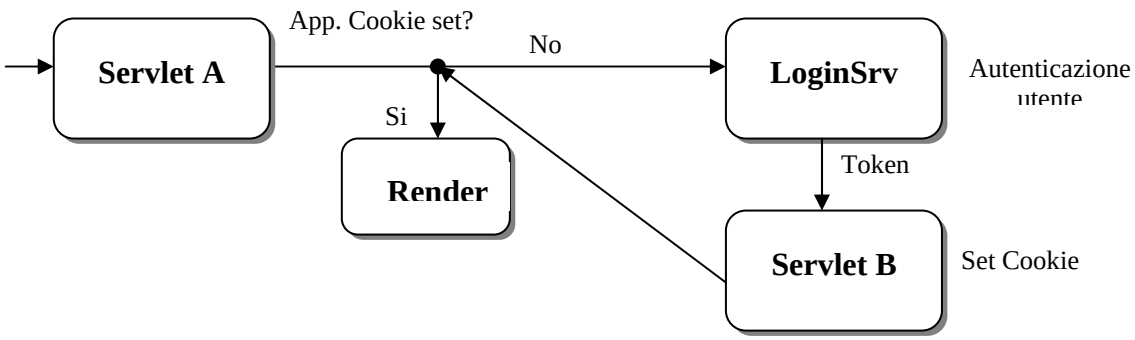
Si noti che trattandosi di esempi sarà cura del programmatore mettere in atto tutte le parametrizzazioni del caso; in particolare quelle relative a nomi logici, indirizzi IP e stringhe per la connessione al DB, per le quali, trattandosi di elementi legati all'ambiente run-time e come tali suscettibili di variazioni, si suggerisce l'utilizzo di variabili esterne impostate tramite gli init-parameter delle web application.

E' fortemente auspicabile che, fatte salve le personalizzazioni che di volta in volta potrebbero rendersi necessarie, l'impianto, che viene sinteticamente rappresentato da questo diagramma a blocchi, non venga stravolto.



4.1 Linee Guida di programmazione

Il flusso operativo del sistema di autenticazione di una partner application Oracle è il seguente:



L'utente si collega alla Servlet A, la quale provvede a controllare la presenza del Cookie di sessione applicativa, a fronte della mancanza di tale Cookie il controllo viene rediretto al Login Server per

procedere con l'autenticazione dell'utente da parte di quest'ultimo. In caso contrario la servlet provvede ad eseguire il rendering opportuno in base alla tipologia di utenza connessa.

Il Login Server, se chiamato, provvede ad autenticare l'utente e redirige il controllo alla Servlet B, passando a quest'ultima un Token identificativo, attraverso il quale quest'ultima imposta il Cookie di sessione applicativa prima di redirigere il controllo alla Servlet A, che a questo punto, trovando il Cookie richiesto, provvede a procedere con il rendering applicativo idoneo.

I passi da compiere per implementare l'integrazione sono:

- Il file ssosdk307.jar deve essere collocato nel CLASSPATH del servlet engine.
- La classe Test_SSO (che implementa le funzioni della servlet "A"), fornita con il presente documento, deve essere accessibile dai servlet dell'applicazione.
- Analogo requisito per le classi SSOEnablerBean, SSOSignOnServlet, fornite con il presente documento.
- Nella classe Test_SSO, nella sezione compresa tra "/** Start configuration parameters" e "/** End of configuration parameters */" si devono customizzare ed adattare alla situazione i valori dei parametri ivi presenti.

L'utilizzo della classe Test_SSO è modificato come segue:

Nel punto (o nei punti) dell'applicazione in cui si deve verificare se l'utente è già autenticato si deve inserire il seguente codice:

```
Test_SSO l_ssobean = new Test_SSO();
String l_userInfo = l_ssobean.getSSOUserInfo(p_request,
p_response);

if(l_userInfo != null)
{
  /* utente autenticato */
 /* se si vuole la lista dei gruppi a cui l'utente appartiene */
/* added for CONSIP */
 String sGruppi = l_ssobean.getUserGroups(l_userInfo);
prosegue con le elaborazioni.....
}
else
{
  /* NON si deve scrivere codice per la redirect !!!!*/
 /* è fatta da getSSOUserInfo */
}
```

Infine, per il settaggio dell'application cookie (la funzione B), si suggerisce di utilizzare il SSOSignOnServlet, fornito con il presente documento.

Si precisa che:

- In linea di principio gli sviluppatori delle singole applicazioni dovrebbero modificare solo la classe Test_SSO, nella sezione compresa tra i due commenti prima indicati.
- Il nome del servlet SSOSignOnServlet può essere modificato. Ovviamente è il nome selezionato che deve essere specificato nella SUCCESS_URL.
- La classe SSOEnablerBean è parte integrante del SSO SDK Oracle e come tale non andrebbe modificata. IL sorgente è fornito dalla Oracle allo scopo di documentare i meccanismi di integrazione e renderli più chiari agli sviluppatori.

La classe Test_SSO va istanziata fornendo come parametro la stringa "I" o "E" che scinde la provenienza della richiesta interna/esterna per poter caricare dai file di properties i valori necessari all'accesso del LS corretto.
Es.

Test_SSO t = new sso.Test_SSO("I"); //interno
Test_SSO t = new sso.Test_SSO("E"); //esterno

Nella classe Test_SSO la funzione getUserGroups fornisce la lista dei gruppi di Portal a cui l'utente appartiene.

Per il funzionamento della classe Test_SSO devono essere previste due distinte connessioni Oracle, per due database che in linea di principio sono distinti[1]:

- Il database in cui sono registrate le informazioni necessarie alla partner application per agganciarsi al Login server;
- Il database di Portal, da cui sono letti i gruppi.

Le informazioni relative alle connessioni verso i suddetti db sono contenuti in due distinti file di properties nel formato <*chiave = valore*>.
Di seguito un esmpio dei due file di properties:

login.properties

username=<username dello schema ls per le partner application>
password=<username dello schema ls per le partner application>
host=<ip/hostname>
port=<porta>
sid=<sid>
cookie_name=<nome del cookie>
cookie_name_ext=<nome del cookie per accesso esterno>
cookie_domain=<hostname[domain]>
cookie_path=/
request_url=<url applicazione per accesso interno>
cancel_url=<url per il redirect dell'applicazione in caso di utente non autorizzato per accesso interno>

[1] Ma che potrebbero, per scelta implementativa coincidere.

request_url_ext=<url applicazione per accesso esterno >
cancel_url_ext=<url per il redirect dell'applicazione in caso di utente non autorizzato per accesso esterno >
listener_token=<nomeprogetto.hotname[domain]>
listener_token_ext=<nomeprogetto.hotname[domain] per accesso esterno >
logout_url=<url di logout per login server interno>
logout_url_ext=< url di logout per login server esterno>

portal.properties

username=<username dello schema portal per le partner application>
password=<username dello schema portal per le partner application>
host=<ip/hostname>
port=<porta>
sid=<sid>

I file di properties verranno compilati direttamente dal supporto sistemistico, le uniche informazioni che dovranno essere fornite dagli applicativi saranno la request_url e la cancel_url relative al file login.properties.

Infine la funzione getUserGroups è stata implementata con:

- Una singola query, con JOIN tra le tre tabelle coinvolte;
- Utilizzando BIND variable.

Le modifiche sono suggerite allo scopo di migliorare le performance ottenibili e ridurre l'uso delle risorse macchina.

N.B.
Per un problema legato al dominio nei cookie non gestito correttamente dai browser e' necessario che l'hostname dei server sia del formato full qualified (host.domain.ext) .
Nel caso non sia disponibile un ambiente di questo tipo si rende necessaria una modifica nell' SSOEnablerBean commentando la seguente riga nelle funzioni setPartnerAppCookie / remAppCookie dell' SSOEnablerBean:

l_AppCookie.setDomain(m_pappCookieDomain);

5 Integrazione di applicazioni esistenti (External Application)

Come si è detto, secondo la terminologia Oracle si definiscono "External Application" quelle applicazioni che eseguono un'autenticazione con un proprio form HTML di login gestendo un proprio "user repository". Queste applicazioni per poter demandare questa gestione al Login Server dovrebbero essere modificate e l'opportunità di questa operazione va valutata caso per caso.

Ciò nonostante è possibile integrare, con un minimo impatto, queste applicazioni nell'Oracle Portal registrandole come "External Application" mantenendo così il punto unico d'ingresso per le applicazioni. La necessità di intervenire su queste applicazioni è determinata dalle caratteristiche della pagina di Login, se questa è una form standard che gestisce i campi utenza e password statici non sono necessarie modifiche, se invece i campi sono dinamici o se la form di login è costituita da più frame sarà necessario intervenire.

6 Modalità di definizione delle Applicazioni

Le applicazioni di tipo external vengono dichiarate al Login Server in maniera estremamente semplice e lineare, tramite le apposite forms web a disposizione dell'amministratore.

Come accennato, tuttavia, le applicazioni external proprio per il fatto di non subire alcun tipo di intervento di modifica o adattamento, non possono essere integrate nel meccanismo di Single Sign On. Esse possiedono un proprio Repository interno per la gestione degli utenti e dei profili, ove quasi sempre sono riportate, in chiaro, le password.

Viceversa il Single Sign On prevede, come si intuisce dal nome, un punto singolo di identificazione e autenticazione degli utenti, il Login Server, appunto: le applicazioni devono sapere dove sia collocato sulla rete e conoscerne il protocollo di comunicazione.

Per questi motivi la procedura per dichiarare le applicazioni Partner al Login Server è più complessa. Occorre infatti che luogo le applicazioni siano predisposte, ovvero che conoscano il protocollo di comunicazione e l'ubicazione del Login Server, oltre ad una serie di informazioni aggiuntive (identificativi di applicazione, chiavi per la crittografia delle informazioni che vengono scambiate, etc.).

Non è buona regola cablare questo genere di informazioni nel codice, dunque Oracle ha impiantato un meccanismo alternativo per la loro condivisione, che fa ricorso ad uno schema appositamente predisposto. Le informazioni possono dunque essere lette dalle applicazioni su di un generico database.

Al momento della registrazione di una applicazione partner occorre inserire, oltre al nome dell'applicazione (che nel codice applicativo deve essere referenziato come "Application Token", una Home URL ed una Success URL; il Login Server gli attribuisce un codice identificativo, un token ed una chiave di crittografia. Queste tre informazioni devono essere inserite nello schema di

cui sopra per mezzo di una procedura PL/SQL (regapp.sql) che viene fornita, in modo che siano disponibili all'applicazione. Questo per ciascuna Partner Application.

Questo meccanismo permette di centralizzare la gestione delle informazioni, agevolando eventuali operazioni di manutenzione o di gestione dei server.

7 Linee guida per la definizione dei gruppi e degli utenti

In questo capitolo vogliono introdurre i concetti di gruppo e utente ed indicare come questi debbano essere gestiti tramite il Portal (per i gruppi) e il Login Server (per le utenze e password).

Per default il Login Server utilizza tabelle interne per memorizzare le utenze e le password degli utenti ma è anche possibile configurarlo per utilizzare un LDAP server esterno quale ad esempio l'Oracle Internet Directory che fa parte della suite di Oracle 9iAS.

L'Oracle Portal mette a disposizione dell'amministratore un'interfaccia grafica verso il Login Server per la creazione, modifica e cancellazione
e dei gruppi e per l'associazione applicazione gruppi abilitati. Per le regole di nomenclatura delle utenze si rimanda agli standard esistenti.

7.1 Organizzazione dei Gruppi

Portale Interno

Per ogni applicazione deve essere creato il gruppo corrispondente ed eventuali sottogruppi.

Il gruppo principale deve essere identificato con un nome di massimo otto digit alfanumerici, i nomi dei sottogruppi, associati a questa applicazione (identificativi del ruolo), devono avere il seguente formato XXX-YYYY dove XXX è un acronimo per l'applicazione e YYYY è il ruolo.

A titolo di esempio:

applicazione ----> SPESE
sottogruppo 1 ---> SPS-840 (indica la ragioneria di appartenenza) (ufficio)
sottogruppo 2 ---> SPS-DIR (indica il direttore) (ruolo)

Si ritiene opportuno che vengano definiti, all'atto della creazione di questo Portal Server, 6 gruppi istituzionali, uno per ciascun dipartimento, ed uno per tutti gli utenti che accedono da Internet. Questi gruppi, che hanno l'unico obiettivo di raggruppare logicamente gli utenti in base alle unità organizzative di appartenenza, non hanno valenza nei confronti delle applicazioni, ma sono da prevedere solo per facilitare eventuali operazioni massive di autorizzazione ad applicazioni e/o trasferimento dei dati utenti su macchine e/o prodotti diversi.

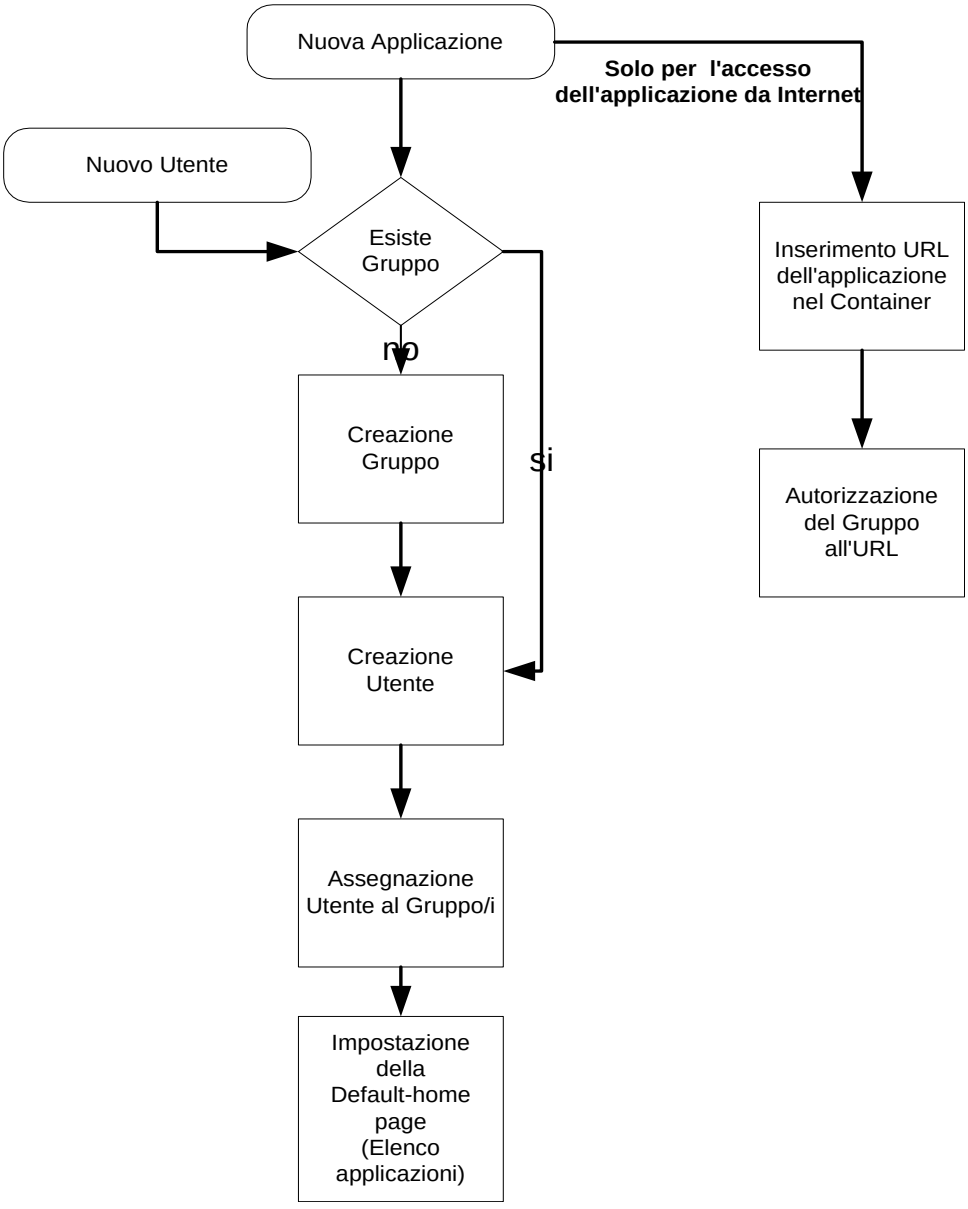
A titolo di esempio i suddetti gruppi potrebbero essere:
DIPI, DIPII, DIPIII, DIPIV, DIPV ed ESTERNI

ed ogni utente, all'atto della creazione, dovrebbe essere associato ad uno dei suddetti gruppi.

Portale Esterno

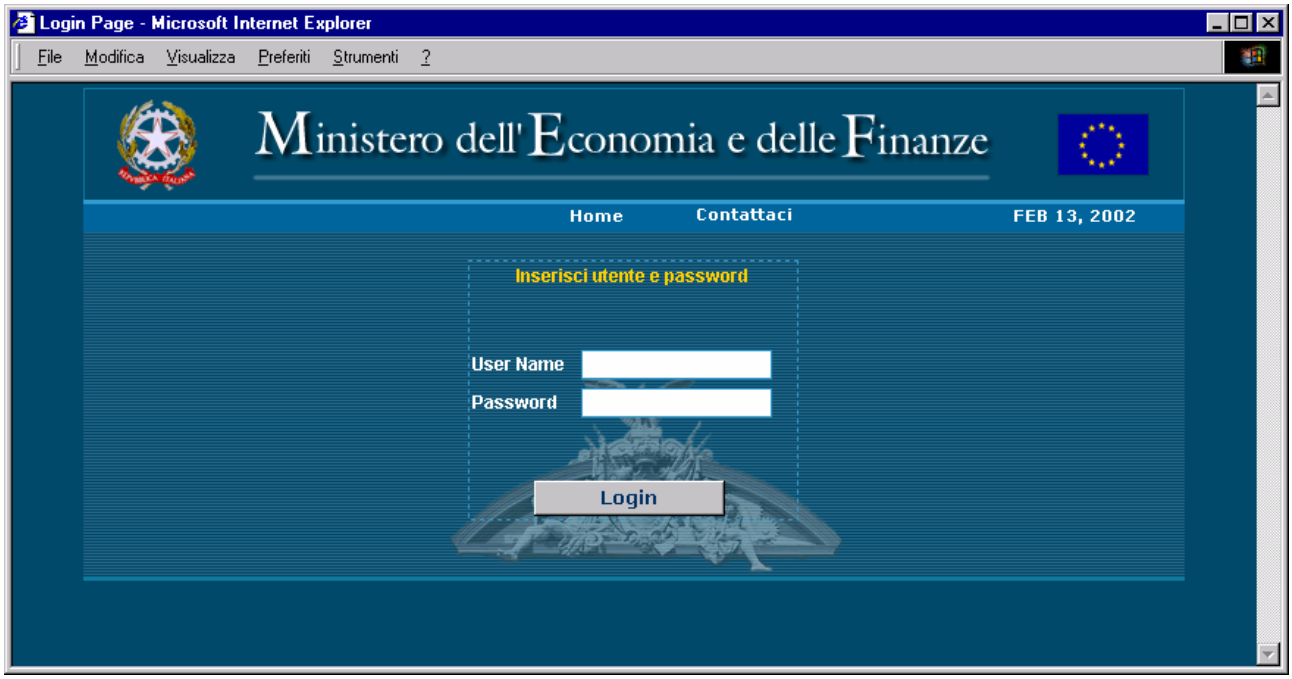
Per ogni applicazione che deve essere accessibile da internet è necessario, sul portale esterno, un gruppo apposito la cui nomenclatura deve essere <nome applicazione>-EXT.
A questo gruppo dovranno essere associati gli utenti precedentemente definiti.

Il seguente diagramma rappresenta il flusso per la creazione di nuovi gruppi, utenti e di come viene creata l'associazione gruppi applicazione (a livello di Oracle Portal).



8 Portal Esterno

Il Portale esterno, come si è detto, non prevede la presenza di contenuti pubblici, sono state invece predisposte sia la pagina iniziale (Login Page) sia la pagina che contiene l'elenco delle applicazioni (Default-Home Page).



L'elenco delle applicazioni che viene presentato in questa seconda pagina è funzione dei gruppi applicativi cui appartiene l'utente.
In particolare per ciascun link verso le applicazioni che compare in questa pagina saranno date le grant di visualizzazione al solo corrispondente gruppo applicativo.

Ogni utente esterno sarà associato a tutti e soli i gruppi corrispondenti alle applicazioni cui avrà il diritto di accedere, in modo che ad ognuno saranno presentati i soli link di interesse.

Resta a carico dei gruppi di sviluppo la personalizzazione delle procedure "A" e "B" per far si che un utente dopo essersi autenticato al portale esterno non possa avere accesso ad una applicazione non di sua pertinenza tramite una chiamata ad una URL diretta.

Potrebbe succedere, infatti, che un utente potrebbe provare ad accedere ad altre applicazioni oltre a quelle che gli vengono presentate dal portale, invocandone direttamente la URL. Tale pratica deve essere impedita dalle funzioni "A" e "B" tramite verifica, questa volta sul portale interno, che l'utente sia abilitato ad accedere all'applicazione in questione, facendo parte del gruppo relativo.

In altre parole la associazione di un utente ai gruppi applicativi non è soltanto funzionale all'individuazione del suo profilo, ma costituisce una vera e propria misura di sicurezza volta ad escludere intromissioni indebite o, peggio, dolose.