Specifiche tecniche per l'utilizzo del sistema di autenticazione centralizzata (allegato 9 alle Linee guida per la governance del sistema informatico regionale)

1 Considerazioni generali

Questo documento ha lo scopo di descrivere l'integrazione delle applicazioni ospitate sulle infrastrutture della Regione Emilia-Romagna con il sistema centralizzato di Identity & Access management (IAM), al fine di implementare l'autenticazione.
Il sistema di IAM è finalizzato alla gestione razionale, scalabile ed omogenea delle utenze del Sistema Informativo della Regione ottemperando al tempo stesso alle normative ed ai requisiti di legge in tema di sicurezza informatica e di protezione dei dati personali.
Il sistema di IAM è composto dalle seguenti componenti:

- un servizio di Directory per la gestione centralizzata delle utenze interne ed esterne, sul quale poggiano le funzioni di "profilatura" e "autenticazione" di sistemi e applicazioni integrati nello IAM;
- una soluzione di Identity Management, che, interfacciandosi a diversi repository utente, consente la gestione dell'intero ciclo di vita delle identità su specifici sistemi e applicazioni, la sincronizzazione delle password degli utenti e la delega ai referenti alla gestione delle loro utenze; consente inoltre l'automatizzazione del processo di provisioning degli account, integrato con i processi organizzativi mediante l'utilizzo di workflow;
- una soluzione di Access Management che permette l'accesso in Single Sign On alle applicazioni web integrate, liberando le applicazioni stesse dalla gestione dell'autenticazione.

2 Gestione degli utenti di un'applicazione con il sistema di Identity Management (IdM)

L'applicazione deve delegare al sistema di Identity la gestione degli utenti. L'identity ha bisogno di oggetti chiamati "connettori" che gli permettano di interagire con il repository degli utenti utilizzato dall'applicazione (es. una tabella utenti su un db). In linea generale esistono due tipologie di connettori:

- connettori standard,
- connettori custom.

I primi sono connettori già implementati dal produttore. In questo caso non è necessaria alcuna fase di sviluppo e si può passare alla fase di integrazione. Per informazioni circa i connettori standard si può richiedere ad idmadmin@regione.emilia-romagna.it.
Nel secondo caso, è invece necessario sviluppare dei metodi **network-enable** (API, Store Procedure, Web Services) che devono essere esposti per l'integrazione con il sistema di IdM.
Tali metodi sono normalmente un sottoinsieme dei seguenti, in funzione delle necessità:

- createUser, crea un account sul target
- deleteUser, cancella un account sul target
- getUser, resituisce la vista di un utente sul target
- getAccountIterator, restituisce un iteratore sull'accoundID degli utenti sul target
- update, aggiorna i dati di un utente sul target (compresa la password)
- enable, abilita l'utente sul target
- disable, disabilita l'utente sul target
- listProfile, effettua la lista di tutti i profili del target
- test, testa il funzionamento del servizio

Per ognuno dei metodi precedenti dovranno inoltre essere definiti i parametri di input e di output in base al target da integrare (alcuni target potranno avere dei parametri differenti).

Particolare attenzione va al campo password, nel caso in cui nel target che si vuole integrare essa sia gestita.
Qualora infatti si debba gestire la password sul target, deve essere indicato al gruppo di IdM l'algoritmo di cifratura utilizzato.

3 Accesso alle web application

Per poter accedere ad una Web Application protetta da un sistema di Web SSO, l'utente deve prima aver effettuato il Logon al sistema di autenticazione.
Effettuato il Logon Primario, l'utente può accedere alle Web Application esposte.

Ad ogni richiesta di accesso vengono ripetuti i seguenti passi:

1. la richiesta viene intercettata dal Reverse Proxy
2. l'Agent del Reverse Proxy verifica la presenza di una sessione autenticata per l'utente. Per il mantenimento della sessione viene utilizzato un cookie volatile sul client. Nel caso non esista una sessione associata alla richiesta, l'utente viene diretto verso il modulo di autenticazione del sistema di Web SSO centralizzato
3. il sistema di Web SSO autentica l'utente e restituisce le informazioni all'Agent che a sua volta trasmette le informazioni necessarie alla web application dell'aderente utilizzando l'header http.

3.1 Requisiti di progettazione delle interfacce

Il passaggio dei parametri tra il reverse proxy e l'applicazione avviene sfruttando i meccanismi standard del Web e cioè gli header del protocollo HTTP, quindi la Web Application deve essere in grado di gestire gli header HTTP.
Nel caso in cui l'applicazione voglia filtrare ulteriormente gli accessi può prelevare gli attributi dell'utente dall'http header ed effettuare la profilazione applicativa.

3.2 Formato parametri

I parametri passati nell'header HTTP alla web application esposta servono ad identificare ed a caratterizzare l'originatore della richiesta.
Il parametro necessariamente presente nell'header HTTP dovrà essere il seguente:

Parametro	Significato
USERNAME	Identificativo utente (identifica l'originatore della richiesta).
DOMAIN	Identificativo dominio (RERSDM – EXTRARER)
FIRSTNAME	Identificativo del Nome dell'utente
LASTNAME	Identificativo del Cognome dell'utente
MATRICOLA	Identificativo Matricola dell'utente (solo in caso di utenti regionali)

Potranno essere aggiunti ulteriori parametri da passare nell'http header. Questi parametri sono prelevati dagli attributi dell'utente.
I parametri elencati saranno presenti nell'header HTTP di ogni richiesta. La modalità di accesso alle variabili dell'header sono legate al linguaggio utilizzato per la creazione e la gestione delle pagine web. Al termine del presente documento sono riportati esempi di accesso a queste variabili in JSP, PERL, ASP.NET, ASP.

3.3 Vincoli

Sono riportati di seguito i vincoli a cui le Web Application devono attenersi per poter essere protette dal sistema di Web SSO.

3.3.1 Identificazione e Autenticazione dell'utente

La web application esposta non deve richiedere il login agli utenti che accedono, in quanto il processo di identificazione e autenticazione viene già effettuato nella fase di Logon Primario che l'utente effettua sul sistema di Web SSO. In particolare, l'applicazione non deve richiedere l'immissione esplicita da parte dell'utente di un username e di una password, ma può utilizzare le informazioni di identificazione dell'utente contenute nell'header HTTP di ogni richiesta

3.3.2 Autorizzazione dell'utente

Il processo di autorizzazione all'accesso da parte dell'utente alla web application esposta è effettuato dal Reverse Proxy, che abilita o impedisce l'accesso a singole Applicazioni in base a policy prestabilite. E' facoltà dell'applicazione di estendere il processo di autorizzazione svolto dal Reverse-Proxy, utilizzando le informazioni contenute nell'header HTTP di ogni richiesta per realizzare nuove regole di autorizzazione.

3.3.3 Uso di Cookie

Il meccanismo di Logon e le verifiche effettuate dal Reverse Proxy si basano sullo scambio di *cookie* con la Postazione di Lavoro dell'utente che necessariamente deve permettere l'utilizzo di *cookie*.

3.3.4 Convenzioni sui nomi dei domini

Una web application esposta dalla RER potrà essere accessibile, sia da Internet che dalla rete interna, mediante una URL così strutturata:

https://applicazioni.regione.emilia-romagna.it/<*percorso-applicazione*>/

dove <*percorso-applicazione*> è il *path* (al limite costituito da un singolo nome) scelto per identificare la web application (è ammesso il passaggio di parametri nella URL).

3.3.5 Convenzioni sui nomi delle web application

Qualora un servizio esponga più di una web application, gli URL corrispondenti si differenziano solo relativamente alla componente <*percorso applicazione*>.
Esempio:

https://<dominio>/<percorso applicazione -1>/

https://<dominio>/<percorso applicazione -2>/

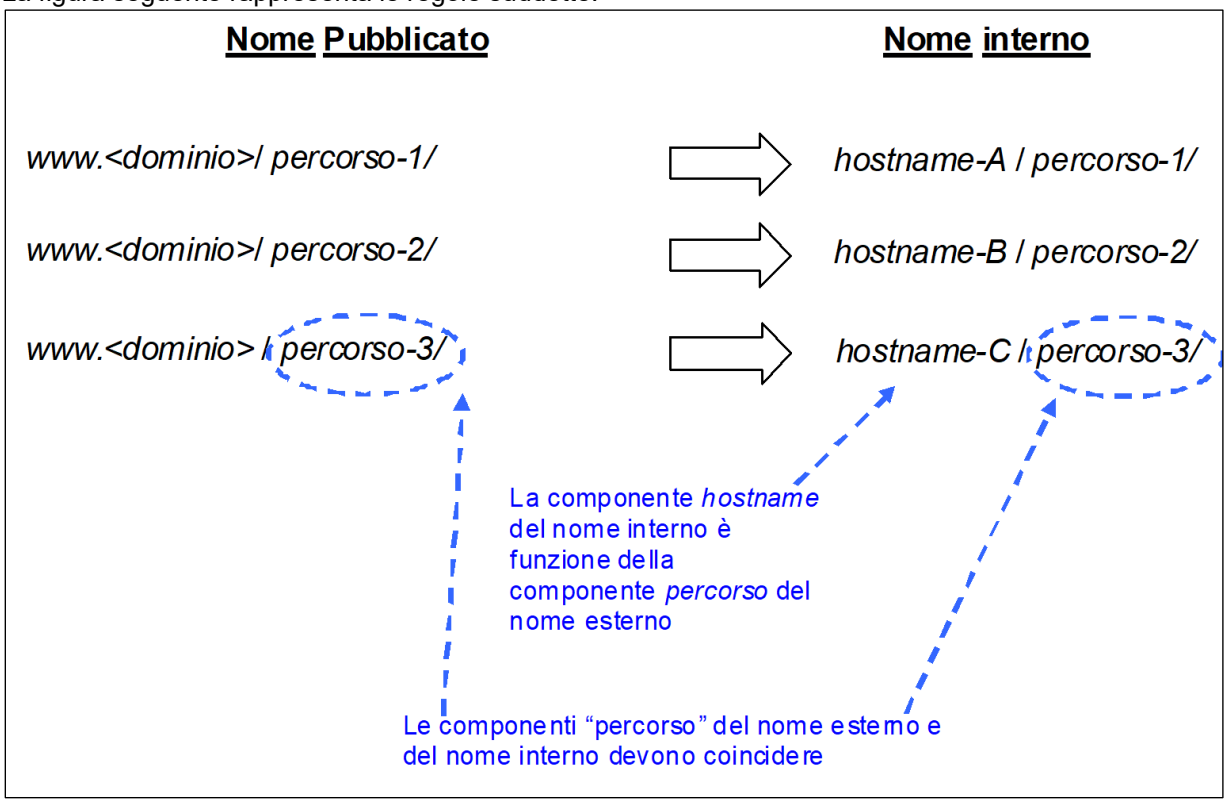
https://<dominio><percorso applicazione -3>/

3.3.6 Regole d'instradamento del Reverse Proxy

La convenzione sui nomi degli URL si riferisce all'esposizione delle web application, ovvero al modo in cui tali applicazioni sono esposte tramite Reverse Proxy e non implica che la medesima convenzione debba necessariamente essere adottata internamente all'applicativo.
Internamente all'applicativo, le web application possono risiedere su uno o più host. La corrispondenza tra il "nome esterno" e il "nome interno" dell'applicazione viene effettuata dal Reverse Proxy tramite le *regole di instradamento*. Tali regole consentono di collocare le proprie web application sui server della rete interna, svincolandosi dall'*hostname* con cui sono visibili.
Nell'instradamento, il path dell'applicazione (cioè la porzione dell'URL che viene dopo l'*hostname*) del

"nome esterno" deve coincidere con il path del "nome interno".

La figura seguente rappresenta le regole suddette:



E' ammesso il passaggio di parametri nell'URL, se previsto dalla web application, mentre non è ammesso utilizzare i parametri per identificare la web application esposta.

Esempio di utilizzo di URL non è ammesso:
https:// www.<dominio>/entrypoint?applicazione=applicazione1

3.3.7 Accessibilità dell'applicazione tramite proxy

Si descrivono di seguito i vincoli che la web application dell'Aderente deve rispettare per poter essere esposta attraverso il Reverse Proxy (requisiti di *proxability*).
La web application da esporre non deve contenere riferimenti assoluti alle proprie risorse, ma solo puntamenti relativi.
In altri termini le eventuali risorse referenziate all'interno dell'applicazione (quali., ad esempio, immagini o link ad altre pagine) devono essere indirizzate tramite **URL relativi**, ovvero URL in cui viene esplicitata solamente la componente *path* senza le componenti *protocollo* ed *hostname*. Oltre agli URL anche **i PATH devono essere relativi**, ovvero non devono iniziare con il carattere "*/*".

Ad esempio:

URL ASSOLUTI (NON UTILIZZABILI)	URL RELATIVI (DA UTILIZZARE)
http(s)://hostname/logo.gif	logo.gif
http(s)://hostname/subdir1/index.html	subdir1/index.html **NOTA :** non è possibile utilizzare URL relativi del tipo /subdir1/index.html, i quali, pur essendo URL relativi (non vengono infatti indicati protocollo ed hostname), sono comunque PATH assoluti.

Inoltre, ogni singola Web Application deve prevedere un unico punto di ingresso da cui si diramano i diversi sottoservizi.
Non sono quindi consentiti collegamenti a sottoservizi non appartenenti all'albero che ha come radice la URL di ingresso della Web Application: ad esempio, supponendo che l'URL di "ingresso" della Web Application *sia* ***http(s)://hostnameX/directoryY***, non è consentito il collegamento a pagine che non risiedano sotto il path *directoryY* quali ***http(s)://hostnameX/directoryZ/pageJ.jsp*** (sempre che l'applicazione che ha come "ingresso" della WebApplication l'URL ***http(s)://hostnameX/directoryZ*** non sia stata esposta a sua volta).

3.3.8 Sicurezza

Ogni applicazione dovrà accettare solo le richieste pervenute dal reverse proxy. Il controllo dovrà essere effettuato mediante un meccanismo di autenticazione con certificato client. Il certificato pubblico del reverse proxy verrà installato sui server sulle quali è installata l'applicazione e l'applicazione, l'application server o il web server dovranno essere configurati in modo da concedere l'accesso all'applicazione solo se il certificato client viene riconosciuto essere quello del reverse proxy. Nel caso questo non fosse possibile occorrerà verificare, a livello applicativo e, ove possibile, a livello sistemistico (con un firewall o sul web server) che le richieste provengano dall'ip del reverse proxy. L'applicazione dovrà riconoscere l'utente, tramite i parametri passati nell'http Header, solo nel caso la richiesta gli venga inoltrata dal *reverse proxy*, in tutti gli altri casi l'applicazione non dovrà permettere l'accesso all'utente.

3.3.9 Gestione del logout applicativo

Nel caso in cui l'applicazione abbia creato una sessione al momento dell'autenticazione dell'utente, il logout dall'applicazione deve invalidare la sessione applicativa.
In ogni caso l'utente deve essere reindirizzato alla lista delle applicazioni al seguente link relativo:

- /index.php

3.3.10 Definizione delle regole di autorizzazione

La regola implicita di autorizzazione degli accessi a una web application esposta sul dominio RER prevede la negazione di ogni accesso. Pertanto, ogni abilitazione deve essere espressamente dichiarata al sottosistema di controllo accessi tramite la formulazione di opportune regole.
Ogni singola regola prevede la specifica di:

1. il nome della risorsa oggetto della regola di abilitazione, dove per 'risorsa' si intende una web application o una sua porzione (sottoalbero o singola pagina);
2. l'elenco dei gruppi applicativi (nell'ambito di quelli definiti dalla RER) abilitati ad accedere alla risorsa di cui al punto precedente;
3. le operazioni ammesse (get e post).

Il nome della risorsa (*Resource Name*) è espresso mediante una *regular expression* che può contenere *wildcards*.

Esempi di nomi risorsa:

https:// www.<dominio>/applicazione1/*	Tutta l'applicazione: 'applicazione1'
https://www.<dominio>/applicazione1/home.htm	La sola pagina 'home.htm' dell'applicazione 'applicazione1'
https://www.<dominio>/applicazione1/consult/*	Tutte le risorse all'interno del sottoalbero 'consult' dell'applicazione 'applicazione1'

Ogni volta che un utente accede ad una *web application* esposta sul dominio RER tramite il *Reverse Proxy*, le URL delle pagine chiamate vengono confrontate con queste *regular expression*. In caso di corrispondenza, viene consentito l'accesso solo se il gruppo dell'originatore della richiesta è stato abilitato per la risorsa identificata e se l'operazione (get o post) è ammessa.
Nel caso vi siano applicazioni con redirezione implicita su una pagina di default è necessario aggiungere una nuova regola dedicata. (ad esempio https:// www.<dominio>/applicazione2).

4 Modulo di raccolta informazioni

Concordemente con quanto espresso sopra, al fine di impostare le configurazioni del *Reverse Proxy*, le regole di autorizzazione, ecc., è necessario per il sottosistema di *Access Management* conosca un insieme di informazioni che dovranno essere fornite dai responsabili applicativi, che verranno raccolte mediante il modulo riportato di seguito.
Sulla base dei dati in esso raccolti verranno configurati il *reverse proxy* (interno ed esterno) e le regole di autorizzazione centralizzate del sistema di *Access Management*.

DATO	VALORE
Nome applicativo	
Referente tecnico	
Path relativo	
Path assoluto	
Note	

Indicazioni per la compilazione:

- **Nome applicativo:** nome dell'applicativo per esteso (esempio: Vetrina della sostenibilità)
- **Referente tecnico:** riferimenti della persona indicata come referente tecnico per l'integrazione delle web application (esempio: Marco Rossi, telefono 051-999999, mail xxx@xx.xx)
- **Path relativo:** nome breve dell'applicativo, utilizzato per formare l'URL di accesso alla web application sul portale delle applicazioni[1] (esempio: VetrinaSostenibilita)
- **Path assoluto:** URL completo della web application da mappare sull'Access Management (esempio: https://amservizi.regione.emilia-romagna.it/VetrinaSostenibilita).
- **Note:** eventuali note

[1] https://applicazioni.regione.emilia-romagna.it

5 Esempi di utilizzo degli attributi dell'http Header

Esempio di JSP:

```
<html>
<head><title>Recupero Username e Dominio </title></head>
<body>
<%
String userid = request.getHeader("username");
String gruppo = request.getHeader("domain");

out.println("<BR> UserID  ->  " + username);
out.println("<BR> Dominio  ->  " + domain);

%>
</body>
</html>
```

Esempio di PERL:

```
#!/usr/bin/env perl
#
# Programma di test
#

print "Content-type: text/html\n\n" ;
print "<html><head><title>Recupero Username e Dominio></title></head>\n";

print "<table border=1 bordercolor=black>";
foreach $e (%ENV)
{
	if ($e eq "HTTP_USERNAME" || $e eq "HTTP_DOMAIN")
	{
		print "<tr>";
		print "<td>$e </td><td>$ENV{$e}</td>";
		print "</tr>";
	}
}
print "</table></body></html>\n";
```

Esempio di ASP.NET:

```
string username = string.Format(@"{0}\{1}",
                    Request.Headers["Domain"],
                    Request.Headers["Username"]
                );

Response.Write(username);
```

Esempio di ASP:

```
username = Request.ServerVariables("HTTP_Domain") & "\" &
        Request.ServerVariables("HTTP_Username")

Response.Write username
```