Single Sign On sul web

Abstract
Un Sigle Sign On (SSO) è un sistema di autenticazione centralizzata che consente a un utente di fornire le proprie credenziali una sola volta e di accedere a molteplici risorse e applicazioni all'interno di una rete locale o della rete Internet. Un tale meccanismo aumenta l'usabilità delle applicazioni dal punto di vista dell'utente e consente una gestione semplificata degli account. L'articolo affronta le problematiche di Single Sign On nel contesto di applicazioni web e presenta il sistema SSOServer, che permette di realizzare un servizio di Single Sign On sul web.

A Single Sign On (SSO) is a centralized authentication system that allows a user to supply his credentials once and gain access to multiple resources and applications in a local network or the Internet. A mechanism of such kind increases the usability of the applications for the users and allows a simplified management of user accounts. This article discusses the single sign on for web applications and introduces SSOServer, a system that realizes a single sign on service for the web.

Keywords: Single sign on, Autenticazione, SSOServer.

Introduzione al Single Sign On

Nell'approccio tradizionale, i sistemi distribuiti sono costituiti da più componenti, ciascuno con un proprio dominio di sicurezza, da cui la necessità per l'utente di autenticarsi presso ogni componente con cui deve interagire. Considerazioni legate all'usabilità e alla sicurezza suggeriscono di coordinare e integrare i servizi di autenticazione e la gestione degli account degli utenti attraverso un sistema di S*ingle Sign On* (SSO), un meccanismo di autenticazione centralizzata, che consente a un utente di autenticarsi in un sistema informatico una sola volta e di accedere a molteplici risorse. Tale meccanismo trova applicazione in contesti diversi, come l'accesso a risorse (file, stampanti, etc.) all'interno di una intranet, o la fruizione di servizi disponibili sul web. Un sistema di SSO apporta benefici sia lato *client* che lato *server*:

– lato client, l'utente impiega meno tempo nelle operazioni di *login* e non è costretto a ricordare diverse credenziali per accedere a sistemi diversi;
– lato *server*, gli amministratori possono gestire gli *account* utente in modo centralizzato e controllare in modo uniforme e consistente i diritti di accesso (autorizzazioni), rafforzando la sicurezza del sistema.

Autenticazione centralizzata per applicazioni il web

L'autenticazione è la fase durante la quale un utente fornisce la propria identità al sistema e precede la fase di autorizzazione con cui si verifica che l'utente abbia i privilegi necessari per accedere alla risorsa protetta. In ambito web, si associa all'utente un codice unico, per esempio sotto forma di un *ticket*, che identifica l'utente e consente di accedere al suo profilo personale. Solitamente questa funzionalità è implementata inserendo il ticket all'interno di un *cookie*, oppure tramite riscrittura della url. Entrambi i metodi si basano sul presupposto che a ogni richiesta successiva il client http dell'utente ripresenterà il ticket all'applicazione. Il limite di questo meccanismo risiede nel fatto che i ticket sono contestuali alle singole applicazioni e inoltre il protocollo http non consente facilmente lo scambio di informazioni (per esempio tramite cookie) tra applicazioni su domini diversi.

Un sistema per l'autenticazione centralizzata prevede un insieme di applicazioni web che costituiscono un dominio applicativo e condividono lo stesso bacino di utenti. Le applicazioni sono registrate presso un server di SSO attraverso un'opportuna interfaccia di amministrazione. Compito del server di SSO è di estendere

il meccanismo descritto sopra e permettere la condivisione delle informazioni di autenticazione tra applicazioni diverse, in modo trasparente alle applicazioni stesse. Questa funzionalità è offerta dal sistema sviluppato, SSOServer, che sarà presentato nei paragrafi seguenti.

SSOServer e protocollo di comunicazione

SSOServer può essere visto come un servizio in grado di generare, memorizzare e validare i ticket che permettono agli utenti di autenticarsi e avere accesso alle applicazioni registrate. Agli utenti possono essere associati due tipi di ticket (Fig. 1):

- **AuthenticationTicket**: ticket associato all'utente durante la fase di login, utilizzato per identificare l'utente univocamente presso il server;
- **ApplicationTicket**: consente a uno specifico utente di accedere a un'applicazione un numero limitato di volte (solitamente una sola volta) ed entro un certo periodo di tempo (solitamente molto breve).

All'utente provvisto di un AuthenticationTicket valido è possibile assegnare dei ticket di applicazione, che possono essere spesi per accedere alle applicazioni registrate. Una volta che un ticket di applicazione è stato usato viene invalidato e non può garantire ulteriore accesso all'applicazione per il quale era stato generato.

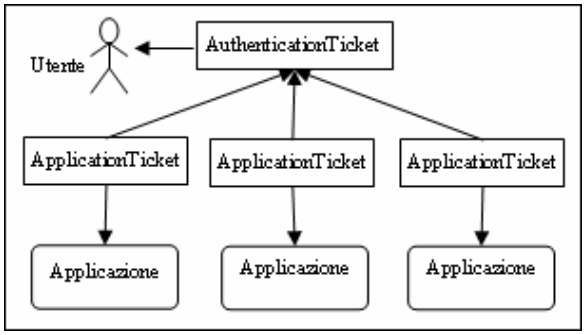

Fig. 1 – Relazioni tra utenti, AuthenticationTicket, ApplicationTicket e applicazioni.

Il protocollo di comunicazione tra il server e le applicazioni registrate può essere schematizzato nei seguenti passi (Fig. 2):

1. Un utente cerca di accedere per la prima volta a una pagina protetta delle applicazioni registrate; poiché non è riconosciuto viene reindirizzato al server di SSO, che gli presenta il modulo per l'inserimento delle proprie credenziali.
2. Se le credenziali sono verificate con successo, vengono generati un AuthenticationTicket, associato all'utente, e un ApplicationTicket, associato all'applicazione richiesta. Entrambi i ticket sono consegnati all'utente, che viene reindirizzato nuovamente all'applicazione richiesta.

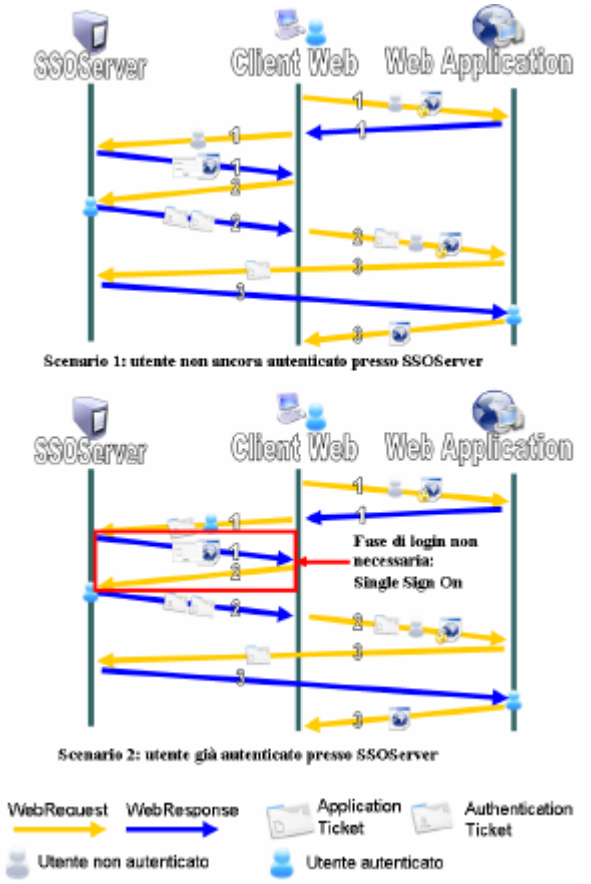

Fig. 2 – Interazione tra utente (web client), SSOServer e applicazioni web durante la fase di autenticazione secondo il protocollo di comunicazione descritto. Nel primo scenario l'utente non è ancora autenticato e deve effettuare la fase di login presso SSOServer; nel secondo scenario l'utente è automaticamente riconosciuto, per cui si realizza un meccanismo di single sign on.

3. L'utente può presentare l'ApplicationTicket all'applicazione, che ne controlla la validità mediante una richiesta al server di SSO. Se il ticket presentato dall'utente è valido, il server risponde con un identificativo unico, che permette di recuperare l'identità dell'utente da un *datastore* locale all'applicazione. A questo punto l'applicazione può mettere in atto un proprio meccanismo per riconoscere automaticamente l'utente, senza richiedere l'intervento del server di SSO per le richieste future. Per esempio l'appli-

cazione può assegnare all'utente un proprio cookie di autenticazione.

4. Quando l'utente accede per la prima volta a un'altra applicazione registrata, viene nuovamente reindirizzato al server di SSO, in quanto non è riconosciuto. Se l'AuthenticationTicket assegnatogli precedentemente risulta ancora valido, l'utente è automaticamente autenticato dal server, che gli assegna un nuovo ApplicationTicket per la nuova applicazione richiesta. In questo modo l'utente riesce ad accedere a tutte le applicazioni registrate, senza che debba ripetere la fase di login sul server di SSO, realizzando di fatto un meccanismo di Single Sign On.

Architettura di SSOServer

L'architettura di SSOServer è stata progettata cercando di favorire i seguenti aspetti:

- interfaccia applicativa: le funzionalità del server devono poter essere accedute dalle applicazioni attraverso un'interfaccia pubblica semplice da utilizzare e di alto livello;
- interoperabilità: il servizio deve essere il più possibile indipendente dalla piattaforma di sviluppo delle applicazioni registrate che utilizzano le sue funzionalità;
- estensibilità: il server deve comprendere punti di estensione, in modo che possa essere utilizzato in contesti diversi.

Per rendere il server parametrico rispetto alle componenti che svolgono le varie funzionalità, si è fatto ricorso al *design pattern Provider*[5]. Qui di seguito vengono presentate le componenti principali di SSOServer (Fig 3).

AuthenticationProvider: è il componente in grado di validare le credenziali di un utente astraendo sia dalla forma delle credenziali (per esempio username/password), sia dal modo in cui esse sono memorizzate (su un database, in un file di configurazione, in chiaro o criptate, etc.) e verificate.

SSOProvider: è il modulo che si occupa di generare e memorizzare effettivamente i ticket, eliminare quelli scaduti e fornire le informazioni sulle applicazioni registrate. I ticket possono essere memorizzati in una struttura in memoria in un database, a seconda del tipo di implementazione effettiva.

SSOManager: è l'interfaccia del server di SSO, visto come sistema in grado di generare, memorizzare e validare i ticket. Rappresenta l'interfaccia verso il livello che si occupa di processare le richieste di accesso al servizio (che possono essere richieste web/http o di altra natura) e si serve di un AuthenticationProvider per verificare le credenziali utente e di un SSOProvider per gestire i ticket e le applicazioni registrate.

Controller: i controller gestiscono una richiesta decidendo il flusso da seguire e le azioni da intraprendere, in base al tipo di richiesta e alle risposte dell'SSOManager. Sono definiti quattro tipi di controller, che corrispondono ai quattro *entrypoint* del servizio di SSO: LoginController, ValidationController, LogoutController e AuthenticationServiceController. Per entrypoint intendiamo un punto di ingresso al server di SSO, associato a una specifica funzionalità. In un contesto web, un entrypoint corrisponde a una url del servizio di SSO che permette di accedere a una funzionalità servizio stesso.

ControllerSupport: è il componente che fa da supporto al controller, permettendo di astrarre dal contesto in cui arrivano le richieste e gestendo il flusso di informazione dal server verso l'esterno e viceversa. Per esempio, in un contesto web è il ControllerSupport che si occupa di recuperare il ticket presentato da un utente da un cookie, piuttosto che da un parametro inviato con la richiesta, o che effettua un reindirizzamento alla pagina di login quando il server richiede che l'utente fornisca le sue credenziali.

L'architettura descritta rende il server indipendente dal contesto in cui è utilizzato, in modo tale che possa essere adattato a esigenze diverse installando diverse implementazioni dei vari componenti. Per esempio, è sufficiente installare un adeguato modulo di ControllerSupport per personalizzare la risposta del server o modificare il meccanismo di recupero dei ticket presentati dall'utente. In questa prima versione è fornito un ControllerSupport per il protocollo http, che permette di usare i servizi di SSO nell'ambito di applicazioni web mediante l'uso di cookie e del meccanismo di redirecting, mentre l'AuthenticationProvider sviluppato permette di verificare le credenziali attraverso un database Oracle.

SSOServer può essere usato anche come servizio di validazione delle credenziali utente, senza la funzionalità di Sigle Sign On, attraverso l'entrypoint AuthenticationService.

Questa funzionalità è utile nell'ambito di quelle applicazioni che vogliano gestire direttamente la fase di autenticazione, sfruttando

comunque il servizio di gestione centralizzata degli account.

I vari moduli del server, così come le impostazioni della libreria di integrazione, possono essere facilmente configurati tramite un file XML. Per quanto riguarda il server, oltre alla possibilità di configurare l'AuthenticationProvider e l'SSOProvider, è possibile impostare anche i seguenti parametri:

- tempo di validità degli AuthenticationTicket;
- tempo di validità degli ApplicationTicket;
- rinnovo automatico degli AuthenticationTicket a ogni accesso dell'utente (*sliding expiration*);
- tempo di attivazione del TicketCleaner, ovvero il componente che elimina periodicamente i ticket scaduti.

Inoltre, l'attività del server viene monitorata attraverso opportuni file di *logging*, che permettono di tracciare gli eventi di autenticazione ed eventuali situazioni anomale che si siano presentate.

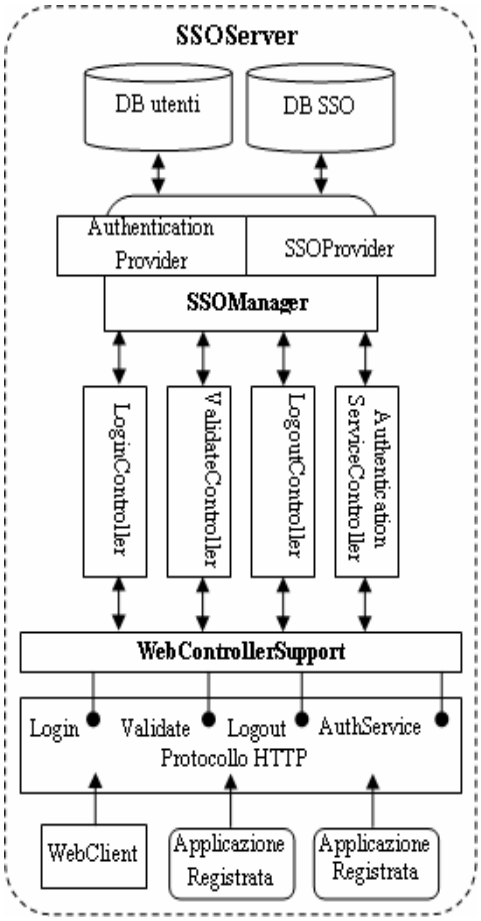

Fig. 3 – Architettura di SSOServer in un contesto web.

Integrazione client

L'applicazione SSOServer è stata sviluppata su piattaforma Microsoft con tecnologia .NET 2.0[6], ma le funzionalità del server possono essere integrate in qualsiasi piattaforma web (Asp.NET, J2EE, PHP, etc.), in quanto il protocollo di comunicazione è basato su http. Accanto al server è stata sviluppata una libreria di integrazione client, SSOWebClient, di cui in questo primo rilascio è disponibile la versione per la piattaforma Asp.NET. La libreria gestisce tutto il protocollo di comunicazione con SSOServer ed espone una API pubblica per accedere direttamente alle sue funzionalità. In particolare, la libreria effettua in modo trasparente le seguenti operazioni:

- intercetta le richieste degli utenti non autenticati ed effettua il *redirecting* al server, impostando gli opportuni parametri;
- si occupa di validare gli ApplicationTicket presentati dagli utenti comunicando direttamente con il server tramite connessione di rete e interpretando la risposta ricevuta;
- notifica, tramite eventi, l'avvenuta autenticazione dell'utente o un'operazione di *logout*;
- può gestire l'autenticazione automatica generando e fornendo all'applicazione il relativo cookie locale per un utente di cui siano state riconosciute le credenziali;
- permette di accedere al servizio di autenticazione per la validazione delle credenziali attraverso una api semplice.

La libreria integra ed estende il meccanismo di *Forms Authentication* fornito dal Framework .NET[7].

Conclusioni

Nell'articolo è stato presentato il sistema SSOServer, che offre un servizio configurabile ed estendibile di Single Sign On per applicazioni web. Il server è stato sviluppato su piattaforma Microsoft.NET, ma i suoi servizi sono accessibili da qualsiasi piattaforma web. Per la piattaforma Asp.Net è disponibile una libreria che consente di sfruttare il servizio di SSO in modo trasparente all'applicazione.

Futuri sviluppi possono riguardare:

- l'utilizzo del protocollo https, per rafforzare la sicurezza del protocollo di comunicazione
- lo sviluppo di un'interfaccia di amministrazione del server;
- l'implementazione del meccanismo di *single sign off*;

– la gestione di più gruppi di applicazioni (domini applicativi), separati tramite la stessa istanza del server.

Bibliografia

[1] The Open Group: Enterprise Architecture Standards, Certification and Services.
URL: http://www.opengroup.org/

[2] Microosft MSDN, Single Sign-On Enterprise Security for Web Applications.
URL: http://msdn2.microsoft.com/en-us/library/ms972971.aspx

[3] JA-SIG Central Authentication Service (CAS).
URL: http://www.ja-sig.org/products/cas/

[4] Cosign: web single sign-on.
URL: http://www.umich.edu/~umweb/software/cosign/

[5] Provider Design Pattern.
URL: http://msdn2.microsoft.com/en-us/library/ms972319.aspx

[6] Microsoft .Net Framework.
URL: http://msdn2.microsoft.com/en-us/netframework/default.aspx

[7] Microsoft .Net Framework Forms Authentication.
URL: http://msdn2.microsoft.com/en-us/library/aa302397.aspx