Infrastruttura di autenticazione @ UNIVPM

Giuliano Latini, Daniele Ripanti, Sandro Tumini

Università Politecnica delle Marche

Conferenza GARR 2011— Centro Congressi CNR - Area della Ricerca di Bologna, 8 -10 Novembre 2011

Abstract

Il numero crescente di richieste di accessi a tipologie di servizi di rete autenticati, ha reso necessario riorganizzare la base dati utenti nell' implementazione di un repository centralizzato in grado di garantire un sistema a "credenziali uniche" per tutte le applicazioni di Ateneo.

L'infrastruttura realizzata è basata su standard diffusi e comprende:

1. Repository LDAP/MS Active Directory distribuito geograficamente;
2. DBMS Oracle;
3. Tecnologie di virtualizzazione per la realizzazione di tutti i server;
4. Software per l'implementazione di cluster simmetrici per l'alta affidabilità/disponibilità;
5. WebServices in standard SOAP/REST;
6. Shibboleth-IDP per l'adesione alla federazione GARR-IDEM.

Progetto

Nella prima fase della realizzazione dell'infrastruttura di autenticazione, sono stati popolati i domain controller dell'Active Directory con i profili utente e successivamente sono stati realizzati i WebServices per l'interfacciamento della base dati.

Particolare attenzione è stata posta nella creazione di un algoritmo di selezione dei Domain Controller all'interno dei WebServices che avesse come obiettivo l'affidabilità.

Per mezzo di un meccanismo di distribuzione random delle richieste nella infrastruttura geografica di server, è stato possibile implementare un sistema scalabile e fault-tolerant.

Il sistema di autenticazione integra, per l'esposizione dei WebServices, la soluzione opensource HAPROXY in uno scenario di cluster simmetrico e tecnologie di virtualizzazione VMWare per il deploy della server farm basata su sistemi operativi Windows, Linux e *BSD.

La soluzione realizzata ha applicazione nella validazione delle credenziali nel captive-portal ottenendo un sistema "uniforme" di accesso ad Internet, nell'interfacciamento ad altri sistemi quali "UGOV-Ricerca" e nella creazione di applicazioni per la gestione di modulistica interna di Ateneo e di supporto alla docenza.

Da circa un anno è stato integrato nell'infrastruttura un identity provider Shibboleth-based aderendo alla federazione IDEM-GARR.

Software

Software	Funzione	Descrizione
MS Windows 2008	Domain Controller	Sistema operativo utilizzato per la creazione del repository LDAP
FreeBSD / Linux	Server Web	Utilizzati per la distribuzione dei WebServices
	Shibboleth-IDP	Piattaforma per l'implementazione dell'IDP IDEM
VMWare ESXi	Piattaforma di Virtualizzazione	Sistemi host dell'infrastruttura di virtualizzazione
HAProxy	Altà disponibilità dei servizi web	Proxy per la distribuzione delle connessioni web ai servizi (WS, U-Gov)
AMP	Apache + MySQL + PHP	Piattaforma più diffusa in UNIVPM per la realizzazione delle applicazioni
MemCached	Cache per le applicazioni	Integrato in tutti i WS e in U-GOV pubblicazioni

Sviluppi futuri

Attualmente sono in corso le seguenti attività:

1. Sviluppo di un modulo Drupal per l'interfacciamento del CMS con i WS SOAP. La piattaforma è utilizzata per la realizzazione dei siti dipartimentali, dei progetti e dei gruppi di ricerca. E' impiegata con il principale obiettivo di uniformare i siti web presenti in Ateneo e pertanto è integrata con il repository degli utenti, ad oggi attraverso il modulo LDAP. Si prevede nei prossimi mesi il completamento del modulo e il test del pacchetto connettore Shibboleth;
2. Test e sostituzione dell'attuale metodo di autenticazione LDAP con Shibboleth per la piattaforma Moodle impiegata per la gestione dei corsi di profitto in modalità elearning;
3. Completamento della migrazione dell'infrastruttura MS Active Directory a Windows Server 2008. Attualmente la foresta AD è popolata da Domain Controller basati su sistema operativo MS Windows 2003. Entro la prima metà del 2012 tutti i sistemi saranno migrati alla versione più recente del sistema operativo;
4. Aggiornamento dell'algoritmo di selezione dei DC all'interno dei WS con l'introduzione di un sistema di interrogazione dei server che valuti alcuni parametri statistici sull'efficienza e performance delle risposte alle richieste di autenticazione. Questo aggiornamento modificherà l'attuale selezione random integrando una classificazione dei DC in "classi di server" mediante un sistema di check in background per la memorizzazione degli "indici".

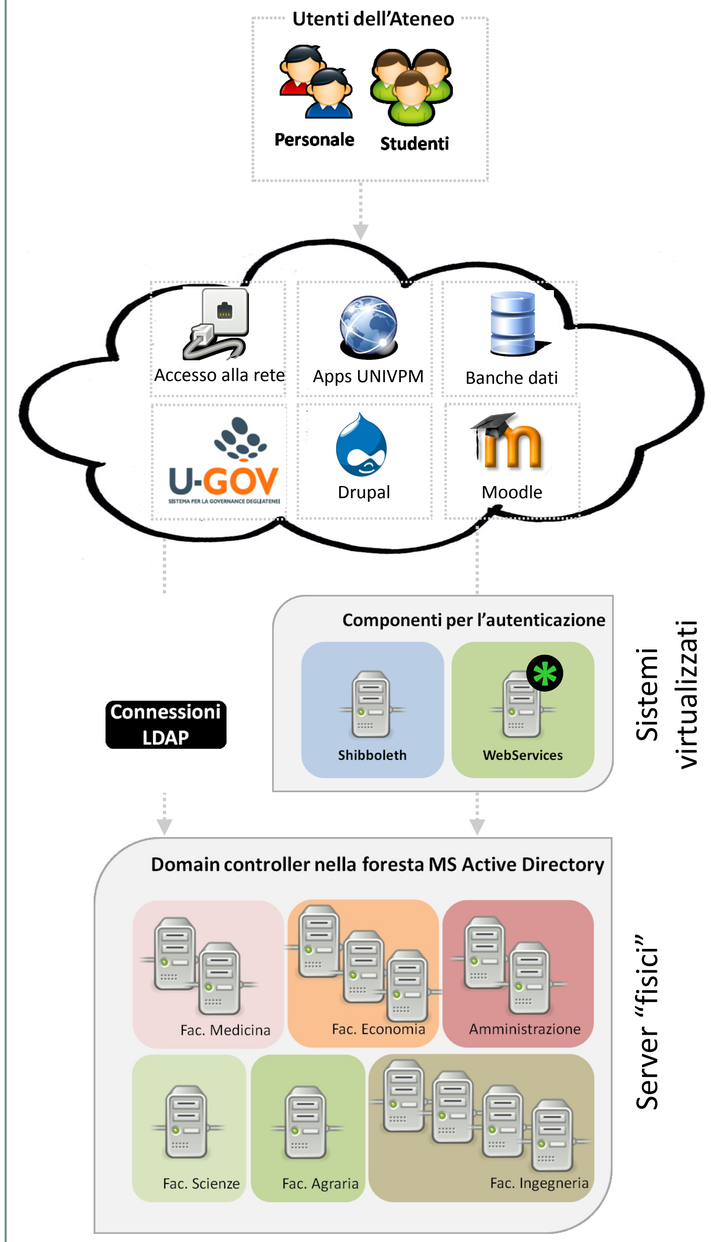
Conclusioni

L'implementazione di WebServices come connettori tra l'infrastruttura MS Active Directory e le applicazioni web based, ottimizza l'aspetto dell'alta disponibilità del servizio di validazione delle credenziali. Il sistema così organizzato migliora l'accesso ai servizi di Ateneo e rende indipendenti gli sviluppatori dallo scrivere codice vincolato al repository.

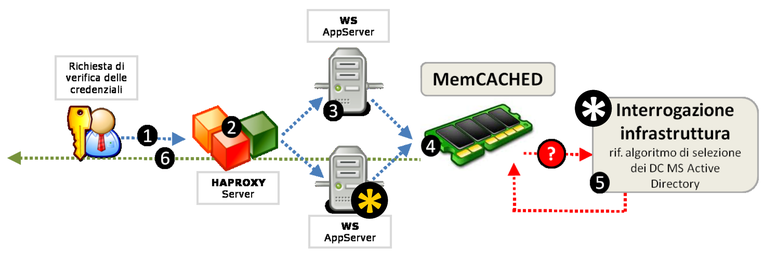
Il potenziamento dell'infrastruttura con nuovi Domain Controller aumenta la robustezza dell'infrastruttura che, combinata all'uso dei WS, garantisce scalabilità e flessibilità.

La realizzazione delle componenti software di autenticazione ha avuto come obiettivo principale quello di avere un sistema"fault-tolerant" e in grado di bilanciare i carichi degli accessi "al meglio" su tutti i server disponibili.

Schema delle componenti presenti nella rete UNIVPM

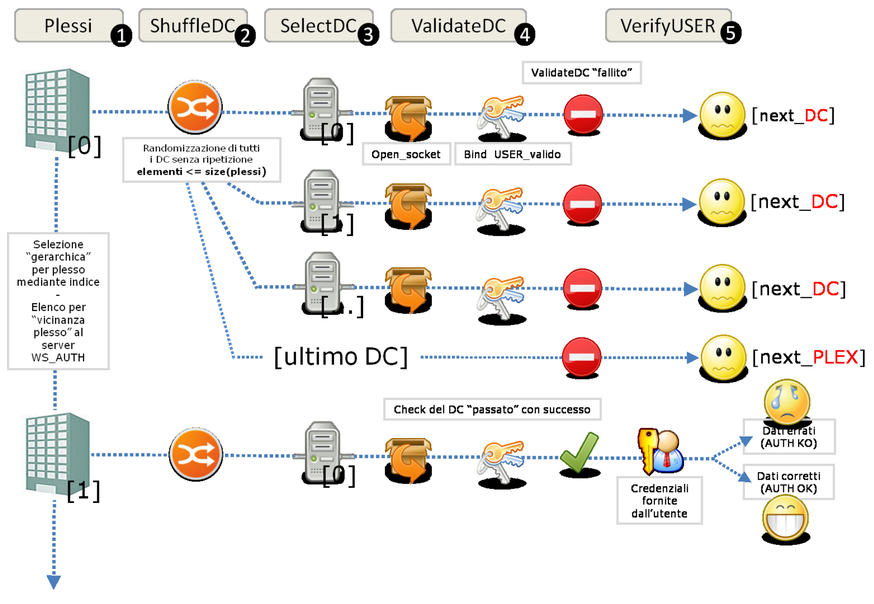


HAPROXY e MemCached



WS SOAP e Rest disponibili per gli sviluppatori

Algoritmo di selezione del DC controller nella foresta Active Directory



Alta flessibilità e affidabilità del servizio di autenticazione

L'infrastruttura **MS Active Directory** è organizzata in **Domain Controller** divisi per **plesso fisico** (sedi delle Facoltà, Amministrazione centrale). Ciascuna sede ha **2** o più server disponibili per il servizio di validazione e interrogabili mediante il protocollo **LDAP**.

Nella **figura** è schematizzato l'algoritmo impiegato dai WebServices per la verifica delle credenziali utente ad ogni richiesta da parte di una applicazione web e può essere sintetizzato nelle seguenti operazioni:

1. La componente software ordina i **plessi** in modo **gerarchico** in base alla **vicinanza geografica** attraverso una struttura predefinita contenente i riferimenti dei server disponibili;
2. **Selezione del primo plesso** definito nell'ordine gerarchico;
3. **Generazione casuale** della lista dei Domain Controller presenti all'interno del plesso selezionato;
4. **Verifica della disponibilità** del server estratto casualmente attraverso alcuni controlli tra i quali una **bind** valida;
5. **Verifica delle credenziali utente inviate** dall'applicazione che richiede il servizio;
6. Nel caso in cui il **Domain Controller non sia disponibile**, la richiesta passerà alla selezione del secondo elemento (ed eventualmente del successivo) dell'elenco generato casualmente;
7. Nel caso in cui tutti i **Domain Controller del plesso non siano disponibili**, si passerà al **plesso successivo** definito nella gerarchia.

Il sistema non è in grado di fornire il servizio esposto solo nel caso di **fault di tutti i Domain Controller** dell'infrastruttura.