UNIVERSITÀ DELLA CALABRIA

Facoltà di Scienze Matematiche Fisiche e Naturali

Corso di laurea in Informatica

TESI DI LAUREA

Titolo

Private Cloud SaaS con Software Libero

Relatori

Prof. Salvatore Di Gregorio

Dott. Vincenzo Bruno

Candidato

Giuseppe Leone

Anno Accademico 2011/2012

L’OPERA (COME SOTTO DEFINITA) È MESSA A DISPOSIZIONE SULLA BASE DEI TERMINI DELLA PRESENTE LICENZA “CREATIVE COMMONS PUBLIC LICENCE” ("CCPL" O "LICENZA"). L’OPERA È PROTETTA DAL DIRITTO D’AUTORE, DAGLI ALTRI DIRITTI ATTRIBUITI DALLA LEGGE SUL DIRITTO D'AUTORE (DIRITTI CONNESSI, DIRITTI SULLE BANCHE DATI, ECC.) E/O DALLE ALTRE LEGGI APPLICABILI. OGNI UTILIZZAZIONE DELL’OPERA CHE NON SIA AUTORIZZATA AI SENSI DELLA PRESENTE LICENZA E/O DELLE ALTRE LEGGI APPLICABILI È PROIBITA.

CON IL SEMPLICE ESERCIZIO SULL’OPERA DI UNO QUALUNQUE DEI DIRITTI QUI DI SEGUITO ELENCATI, TU ACCETTI E TI OBBLIGHI A RISPETTARE INTEGRALMENTE I TERMINI DELLA PRESENTE LICENZA. IL LICENZIANTE CONCEDE A TE I DIRITTI QUI DI SEGUITO ELENCATI A CONDIZIONE CHE TU ACCETTI DI RISPETTARE I TERMINI E LE CONDIZIONI DI CUI ALLA PRESENTE LICENZA.

Art. 1 - Definizioni

Ai fini e per gli effetti della presente licenza, si intende per

a. **"Collezione di Opere"**, un’opera, come un numero di un periodico, un’antologia o un’enciclopedia, nella quale l’Opera nella sua interezza e forma originale, unitamente ad altri contributi costituenti loro stessi opere distinte ed autonome, sono raccolti in un’unità collettiva. Un’opera che costituisce Collezione di Opere non verrà considerata Opera Derivata (come sotto definita) ai fini della presente Licenza;

b. **"Opera Derivata"**, un’opera basata sull’Opera ovvero sull’Opera insieme con altre opere preesistenti, come una traduzione, un arrangiamento musicale, un adattamento teatrale, narrativo, cinematografico, una registrazione di suoni, una riproduzione d’arte, un digesto, una sintesi, o ogni altra forma in cui l’Opera possa essere riproposta, trasformata o adattata. Nel caso in cui un’Opera tra quelle qui descritte costituisca già Collezione di Opere, essa non sarà considerata Opera Derivata ai fini della presente Licenza. Al fine di evitare dubbi è inteso che, quando l’Opera sia una composizione musicale o registrazione di suoni, la sincronizzazione dell’Opera in relazione con un’immagine in movimento (“synching”) sarà considerata Opera Derivata ai fini di questa Licenza;

c. **"Licenziante"**, l’individuo, gli individui, l’ente o gli enti che offre o offrono l’Opera secondo i termini e le condizioni della presente Licenza;

d. **"Autore Originario"**, il soggetto o i soggetti che ha o hanno creato l'Opera;

e. **"Opera"**, l’opera dell’ingegno o, comunque, qualsiasi bene o prestazione suscettibile di protezione in forza delle leggi sul diritto d’autore (diritto d'autore, diritti connessi, diritto sui generis sulle banche dati, ecc.), la cui utilizzazione è offerta nel rispetto dei termini della presente Licenza;

f. **"Tu"/"Te"**, l’individuo o l’ente che esercita i diritti derivanti dalla presente Licenza e che non abbia precedentemente violato i termini della presente Licenza relativi all’Opera o che, nonostante una precedente violazione degli stessi, abbia ricevuto espressa autorizzazione dal Licenziante all’esercizio dei diritti derivanti dalla presente Licenza;

g. **"Elementi della Licenza"**, gli attributi fondamentali della Licenza scelti dal Licenziante ed indicati nel titolo della Licenza: Attribuzione, Non commerciale, Condividi allo stesso modo.

Art. 2 - Libere Utilizzazioni

La presente Licenza non intende in alcun modo ridurre, limitare o restringere alcuna utilizzazione non protetta dal diritto d'autore o alcun diritto di libera utilizzazione o l’operare della regola dell’esaurimento del diritto o altre limitazioni dei diritti sull’Opera derivanti dalle leggi applicabili.

Art. 3 - Concessione della Licenza

Nel rispetto dei termini e delle condizioni contenute nella presente Licenza, il Licenziante concede a Te una licenza per tutto il mondo, gratuita, non esclusiva e perpetua (per la durata del diritto d’autore applicabile) che autorizza ad esercitare i diritti sull’Opera qui di seguito elencati:

a. riproduzione dell’Opera, incorporazione dell’Opera in una o più Collezioni di Opere e riproduzione dell’Opera come incorporata nelle Collezioni di Opere;
b. creazione e riproduzione di un’Opera Derivata, a condizione che l'Opera Derivata (ivi incluse le traduzioni, con qualsiasi mezzo esse siano realizzate) contenga, nei modi appropriati alla forma dell'Opera Derivata, una chiara indicazione del fatto che sono state effettuate delle modifiche rispetto all'Opera originaria. Per esempio, una traduzione potrebbe contenere l'indicazione "questa è la traduzione in spagnolo dell'opera originaria, scritta in inglese"; una modifica potrebbe contenere l'indicazione "l'opera originaria è stata modificata";
c. distribuzione di copie dell’Opera o di supporti fonografici su cui l’Opera è registrata, prestito di copie dell’Opera o di supporti fonografici su cui l’Opera è registrata, comunicazione al pubblico, rappresentazione, esecuzione, recitazione o esposizione in pubblico, ivi inclusa la trasmissione audio digitale dell’Opera, e ciò anche quando l’Opera sia incorporata in Collezioni di Opere;
d. distribuzione di copie dell’Opera Derivata o di supporti fonografici su cui l’Opera Derivata è registrata, prestito di copie dell’Opera Derivata o di supporti fonografici su cui l’Opera Derivata è registrata, comunicazione al pubblico, rappresentazione, esecuzione, recitazione o esposizione in pubblico, ivi inclusa la trasmissione audio digitale di Opere Derivate.

I diritti sopra descritti potranno essere esercitati con ogni mezzo di comunicazione e in tutti i formati. Tra i diritti di cui sopra si intende compreso il diritto di apportare all’Opera le modifiche che si rendessero tecnicamente necessarie per l’esercizio di detti diritti tramite altri mezzi di comunicazione o su altri formati. Tutti i diritti non espressamente concessi dal Licenziante rimangono riservati, ivi inclusi quelli di cui ai punti 4.f e 4.g. Tutti i diritti morali irrinunciabili riconosciuti dalla legge applicabile rimangono riservati.

Qualora l'Opera concessa in licenza includa una o più banche dati sulle quali il Licenziante è titolare di un diritto sui generis ai sensi delle norme nazionali di attuazione della Direttiva 96/9/CE sulle banche dati, il Licenziante rinuncia a far valere il diritto corrispondente.

Art. 4 - Restrizioni

La Licenza concessa in conformità al precedente punto 3 è espressamente assoggettata a, e limitata da, le seguenti restrizioni:

a. Tu puoi distribuire, comunicare al pubblico, rappresentare, eseguire, recitare o esporre in pubblico l'Opera, anche in forma digitale, solo alle condizioni della presente Licenza e, insieme ad ogni copia dell'Opera (o supporto fonografico su cui è registrata l'Opera) che distribuisci, comunichi al pubblico o rappresenti, esegui, reciti o esponi in pubblico, anche in forma digitale, devi includere una copia della presente Licenza o il suo Uniform Resource Identifier. Non puoi proporre o imporre alcuna condizione relativa all'Opera che restringa i termini della presente Licenza o la capacità da parte di chi riceve l'Opera di esercitare gli stessi diritti concessi a Te con la presente Licenza. Non puoi concedere l'Opera in sublicenza. Devi mantenere intatte tutte le informative che si riferiscono alla presente Licenza ed all'esclusione delle garanzie. Quando distribuisci, comunichi al pubblico, rappresenti, esegui, reciti o esponi in pubblico l'Opera, non puoi utilizzare alcuna misura tecnologica tale da limitare la capacità di chiunque riceva l'Opera da Te di esercitare gli stessi diritti concessi a Te con la presente licenza. Questo art. 4.a si applica all'Opera anche quando questa faccia parte di una Collezione di Opere, anche se ciò non comporta che la Collezione di Opere di per sé ed indipendentemente dall'Opera stessa debba essere soggetta ai termini ed alle condizioni della presente Licenza. Qualora Tu crei una Collezione di Opere, su richiesta di qualsiasi Licenziante, devi rimuovere dalla Collezione di Opere stessa, ove materialmente possibile, ogni riferimento in accordo con quanto previsto dall'art. 4.d, come da richiesta. Qualora Tu crei un'Opera Derivata, su richiesta di qualsiasi Licenziante devi rimuovere dall'Opera Derivata stessa, nella misura in cui ciò sia possibile, ogni riferimento in accordo con quanto previsto dall'art. 4.d, come da richiesta.
b. Tu puoi distribuire, comunicare al pubblico, rappresentare, eseguire, recitare o esporre in pubblico un'Opera Derivata, anche in forma digitale, solo alle condizioni (i) della presente Licenza, (ii) di una versione successiva di questa Licenza dotata degli stessi Elementi della Licenza di questa Licenza, (iii) della licenza Creative Commons "Unported" (non adattata) o di una licenza Creative Commons di un'altra giurisdizione (sia la presente versione 3.0 che una successiva) che contenga gli stessi Elementi della Licenza di questa Licenza (ad es. Attribuzione-NonCommerciale-Condividi allo stesso modo 3.0 "Unported") (la "Licenza Applicabile"). Insieme ad ogni copia dell'Opera Derivata (o supporto fonografico su cui è registrata l'Opera Derivata) che distribuisci, comunichi al pubblico o rappresenti, esegui, reciti o esponi in pubblico, anche in forma digitale, Tu devi includere una copia della Licenza Applicabile o il suo Uniform Resource Identifier. Non puoi proporre o imporre alcuna condizione relativa all'Opera Derivata che restringa i termini della Licenza Applicabile o la capacità di chiunque riceva l'Opera Derivata da Te di esercitare gli stessi diritti concessi a Te con la Licenza Applicabile. Devi mantenere intatte tutte le informative che si riferiscono alla Licenza Applicabile ed all'esclusione delle garanzie. Quando Tu distribuisci, comunichi al pubblico, rappresenti, esegui, reciti o esponi in pubblico l'Opera Derivata, non puoi utilizzare sull'Opera Derivata alcuna misura tecnologica tale da limitare la capacità di chiunque riceva l'Opera Derivata da Te di esercitare i diritti concessi a tale soggetto in forza della Licenza Applicabile. Questo art. 4.b si applica all'Opera Derivata anche quando questa faccia parte di una Collezione di Opere, ma ciò non comporta che la Collezione di Opere di per sé ed indipendentemente dall'Opera Derivata debba esser soggetta ai termini ed alle condizioni della Licenza Applicabile.
c. Tu non puoi esercitare alcuno dei diritti a Te concessi al precedente punto 3 in una maniera tale che sia prevalentemente intesa o diretta al perseguimento di un vantaggio commerciale o di un compenso monetario privato. Lo scambio dell'Opera con altre opere protette dal diritto d'autore, per mezzo della condivisione di file digitali (c.d. filesharing) o altrimenti, non è considerato inteso o diretto a perseguire un vantaggio commerciale o un compenso monetario privato, a patto che non ci sia alcun pagamento di alcun compenso monetario in connessione allo scambio di opere coperte da diritto d'autore.
d. Qualora Tu distribuisca, comunichi al pubblico, rappresenti, esegua, reciti o esponga in pubblico, anche in forma digitale, l'Opera (come definita dal succitato art. 1) o qualsiasi Opera Derivata (come definita dal succitato art. 1) o Collezione di Opere (come definita dal succitato art. 1), a meno che sia stata avanzata una richiesta ai sensi dell'art. 4.a, devi mantenere intatte tutte le informative sul diritto d'autore sull'Opera. Devi riconoscere una menzione adeguata rispetto al mezzo di comunicazione o supporto che utilizzi: (i) all'Autore Originario citando il suo nome (o lo pseudonimo, se del caso), ove fornito; e/o (ii) alle terze parti designate, se l'Autore Originario e/o il Licenziante hanno designato una o più terze parti (ad esempio, una istituzione finanziatrice, un ente editoriale, un giornale) ("Parti Designate") perché siano citate nell'informativa sul diritto d'autore del Licenziante o nei termini di servizio o con altri mezzi ragionevoli; (iii) il titolo dell'Opera, se indicato; (iv) nella misura in cui sia ragionevolmente possibile, l'Uniform Resource Identifier, che il Licenziante specifichi dover essere associato con l'Opera, salvo che tale URI non faccia alcun riferimento alla informativa sul diritto d'autore o non dia informazioni sulla licenza dell'Opera; (v) inoltre, in conformità a quanto previsto dall'art. 3.b, in caso di Opera Derivata, devi menzionare l'uso dell'Opera nell'Opera Derivata (ad esempio, "traduzione francese dell'Opera dell'Autore Originario", o "sceneggiatura basata sull'Opera originaria dell'Autore Originario"). La menzione richiesta dal presente art. 4.d può essere realizzata in qualsiasi maniera ragionevole possibile; in ogni caso, in ipotesi di Opera Derivata o Collezione di Opere, qualora compaia una menzione di tutti i coautori dell'Opera Derivata o della Collezione di Opere, allora essa deve essere parte di tale menzione e deve apparire con lo stesso risalto concesso alla menzione degli altri coautori. Al fine di evitare dubbi, è inteso che la menzione di cui al presente articolo ha lo scopo di riconoscere la paternità dell'Opera nei modi sopra indicati e che, esercitando i Tuoi diritti ai sensi della presente Licenza, Tu non puoi implicitamente o esplicitamente affermare o fare intendere un qualsiasi collegamento con l'Autore Originario, il Licenziante e/o le Parti Designate, o che l'Autore Originario, il Licenziante e/o le Parti Designate sponsorizzino o avallino Te o il Tuo utilizzo dell'Opera, a meno che non sussista un apposito, espresso e preventivo consenso scritto dell'Autore Originario, del Licenziante e/o delle Parti Designate.
e. Al fine di evitare dubbi, è inteso che le restrizioni di cui ai precedenti punti 4.a, 4.b, 4.c e 4.d non si applicano a quelle parti dell'opera che siano da considerarsi Opera ai sensi della presente Licenza solo in quanto protette dal diritto sui generis su banca dati ai sensi delle norme nazionali di attuazione della Direttiva 96/9/CE sulle banche dati.
f. Al fine di evitare dubbi è inteso che, se l'Opera sia di tipo musicale:

i. **Compensi per la comunicazione al pubblico o la rappresentazione o esecuzione di opere incluse in repertori.** Il Licenziante si riserva il diritto esclusivo di riscuotere compensi, personalmente o per il tramite di un ente di gestione collettiva (ad es. SIAE), per la comunicazione al

pubblico o la rappresentazione o esecuzione, anche in forma digitale (ad es. tramite webcast) dell'Opera, se tale utilizzazione sia prevalentemente intesa o diretta a perseguire un vantaggio commerciale o un compenso monetario privato.

ii. **Compensi per versioni cover.** Il Licenziante si riserva il diritto esclusivo di riscuotere compensi, personalmente o per il tramite di un ente di gestione collettiva (ad es. SIAE), per ogni disco che Tu crei e distribuisci a partire dall'Opera (versione cover), nel caso in cui la Tua distribuzione di detta versione cover sia prevalentemente intesa o diretta a perseguire un vantaggio commerciale o un compenso monetario privato.

g. **Compensi per la comunicazione al pubblico dell'Opera mediante fonogrammi.** Al fine di evitare dubbi, è inteso che se l'Opera è una registrazione di suoni, il Licenziante si riserva il diritto esclusivo di riscuotere compensi, personalmente o per il tramite di un ente di gestione collettiva (ad es. IMAIE), per la comunicazione al pubblico dell'Opera, anche in forma digitale, nel caso in cui la Tua comunicazione al pubblico sia prevalentemente intesa o diretta a perseguire un vantaggio commerciale o un compenso monetario privato.

h. **Altri compensi previsti dalla legge italiana.** Al fine di evitare dubbi, è inteso che il Licenziante si riserva il diritto esclusivo di riscuotere i compensi a lui attribuiti dalla legge italiana sul diritto d'autore (ad es. per l'inserimento dell'Opera in un'antologia ad uso scolastico ex art. 70 l. 633/1941), personalmente o per tramite di un ente di gestione collettiva (ad es. SIAE, IMAIE), se l'utilizzazione dell'Opera sia prevalentemente intesa o diretta a perseguire un vantaggio commerciale o un compenso monetario privato. Al Licenziante spettano in ogni caso i compensi irrinunciabili a lui attribuiti dalla medesima legge (ad es. l'equo compenso spettante all'autore di opere musicali, cinematografiche, audiovisive o di sequenze di immagini in movimento nel caso di noleggio ai sensi dell'art. 18-bis l. 633/1941).

Art. 5 - Dichiarazioni, Garanzie ed Esonero da responsabilità

SALVO CHE SIA ESPRESSAMENTE CONVENUTO ALTRIMENTI PER ISCRITTO FRA LE PARTI, IL LICENZIANTE OFFRE L'OPERA IN LICENZA "COSÌ COM'È" E NON FORNISCE ALCUNA DICHIARAZIONE O GARANZIA DI QUALSIASI TIPO CON RIGUARDO ALL'OPERA, SIA ESSA ESPRESSA OD IMPLICITA, DI FONTE LEGALE O DI ALTRO TIPO, ESSENDO QUINDI ESCLUSE, FRA LE ALTRE, LE GARANZIE RELATIVE AL TITOLO, ALLA COMMERCIABILITÀ, ALL'IDONEITÀ PER UN FINE SPECIFICO E ALLA NON VIOLAZIONE DI DIRITTI DI TERZI O ALLA MANCANZA DI DIFETTI LATENTI O DI ALTRO TIPO, ALL'ESATTEZZA OD ALLA PRESENZA DI ERRORI, SIANO ESSI ACCERTABILI O MENO. ALCUNE GIURISDIZIONI NON CONSENTONO L'ESCLUSIONE DI GARANZIE IMPLICITE E QUINDI TALE ESCLUSIONE PUÒ NON APPLICARSI A TE.

Art. 6 - Limitazione di Responsabilità

SALVI I LIMITI STABILITI DALLA LEGGE APPLICABILE, IL LICENZIANTE NON SARÀ IN ALCUN CASO RESPONSABILE NEI TUOI CONFRONTI A QUALUNQUE TITOLO PER ALCUN TIPO DI DANNO, SIA ESSO SPECIALE, INCIDENTALE, CONSEQUENZIALE, PUNITIVO OD ESEMPLARE, DERIVANTE DALLA PRESENTE LICENZA O DALL'USO DELL'OPERA, ANCHE NEL CASO IN CUI IL LICENZIANTE SIA STATO EDOTTO SULLA POSSIBILITÀ DI TALI DANNI. NESSUNA CLAUSOLA DI QUESTA LICENZA ESCLUDE O LIMITA LA RESPONSABILITÀ NEL CASO IN CUI QUESTA DIPENDA DA DOLO O COLPA GRAVE.

Art. 7 - Risoluzione

a. La presente Licenza si intenderà risolta di diritto e i diritti con essa concessi cesseranno automaticamente, senza necessità di alcuna comunicazione in tal senso da parte del Licenziante, in caso di qualsivoglia inadempimento dei termini della presente Licenza da parte Tua, ed in particolare delle disposizioni di cui ai punti 4.a, 4.b, 4.c e/o 4.d, essendo la presente Licenza condizionata risolutivamente al verificarsi di tali inadempimenti. In ogni caso, la risoluzione della presente Licenza non pregiudicherà i diritti acquistati da individui o enti che abbiano acquistato da Te Opere Derivate o Collezioni di Opere, ai sensi della presente Licenza, a condizione che tali individui o enti continuino a rispettare integralmente le licenze di cui sono parte. Le sezioni 1, 2, 5, 6, 7 e 8 rimangono valide in presenza di qualsiasi risoluzione della presente Licenza.
b. Sempre che vengano rispettati i termini e le condizioni di cui sopra, la presente Licenza è perpetua (e concessa per tutta la durata del diritto d'autore applicabile sull'Opera). Nonostante ciò, il Licenziante si riserva il diritto di rilasciare l'Opera sulla base dei termini di una differente licenza o di cessare la distribuzione dell'Opera in qualsiasi momento; fermo restando che, in ogni caso, tali decisioni non comporteranno recesso dalla presente Licenza (o da qualsiasi altra licenza che sia stata concessa, o che sia richiesto che venga concessa, ai termini della presente Licenza), e la presente Licenza continuerà ad avere piena efficacia, salvo che vi sia risoluzione come sopra indicato.

Art. 8 - Varie

a. Ogni volta che Tu distribuisci, o rappresenti, esegui o reciti pubblicamente in forma digitale l'Opera o una Collezione di Opere, il Licenziante offre al destinatario una licenza per l'Opera nei medesimi termini e condizioni che a Te sono stati concessi tramite la presente Licenza.
b. Ogni volta che Tu distribuisci, o rappresenti, esegui o reciti pubblicamente in forma digitale un'Opera Derivata, il Licenziante offre al destinatario una licenza per l'Opera originaria nei medesimi termini e condizioni che a Te sono stati concessi tramite la presente Licenza.
c. L'invalidità o l'inefficacia, secondo la legge applicabile, di una o più fra le disposizioni della presente Licenza, non comporterà l'invalidità o l'inefficacia dei restanti termini e, senza bisogno di ulteriori azioni delle parti, le disposizioni invalide o inefficaci saranno da intendersi rettificate nei limiti della misura che sia indispensabile per renderle valide ed efficaci.
d. In nessun caso i termini e le disposizioni di cui alla presente Licenza possono essere considerati rinunciati, né alcuna violazione può essere considerata consentita, salvo che tale rinuncia o consenso risultino per iscritto da una dichiarazione firmata dalla parte contro cui operi tale rinuncia o consenso.
e. La presente Licenza costituisce l'intero accordo tra le parti relativamente all'Opera qui data in licenza. Non esistono altre intese, accordi o dichiarazioni relative all'Opera che non siano quelle qui specificate. Il Licenziante non sarà vincolato ad alcuna altra disposizione addizionale che possa apparire in alcuna comunicazione da Te proveniente. La presente Licenza non può essere modificata senza il mutuo consenso scritto del Licenziante e Tuo.

f. La presente licenza è stata redatta sulla base della legge italiana, in particolare del Codice Civile del 1942 e della legge 22 Aprile 1941, n. 633 e successive modificazioni sulla protezione del diritto d'autore e di altri diritti connessi al suo esercizio.

A mio Padre

INDICE

INTRODUZIONE....1
1 COS'È IL CLOUD COMPUTING....3
1.1 Software as a Service....6
1.1.1 Architettura SaaS....7
1.1.2 Integrazione e protocolli aperti....7
1.2 Private cloud....8
2 PUBLIC E PRIVATE CLOUD SAAS....10
2.1 Il public cloud SaaS secondo Google....10
2.1.1 Single-Sign-On....11
2.1.2 Analisi del software....13
2.1.2.1 Messaggistica....13
2.1.2.2 Collaborazione....15
2.1.2.3 Altro....16
2.2 Il private cloud SaaS con Software Libero....17
2.2.1 Single-Sign-On....17
2.2.2 Analisi del software....19
2.2.2.1 Messaggistica....19
2.2.2.2 Collaborazione....20
2.2.2.3 Altro....20
2.3 Critiche e sicurezza....21
2.3.1 Critiche sul modello Public....21
2.3.2 Critiche sul modello Private....21
2.3.3 Aspetti di sicurezza....22
3 SERVIZIO DI AUTENTICAZIONE CENTRALIZZATO....24
3.1 Analisi di LDAP....25
3.1.1 Organizzazione della directory....26
3.1.2 openLDAP: installazione, configurazione e gestione....27
3.1.3 phpLDAPadmin....29
3.2 OpenID: Integrazione con LDAP....30
4 INSTALLAZIONE E CONFIGURAZIONE PIATTAFORMA SAAS....32
4.1 Analisi, installazione e configurazione del software per la messaggistica...32
4.1.1 Postfix....32
4.1.2 Courier Mail Server....34
4.1.3 Horde Groupware Suite....35
4.1.4 Groupserver....38
4.1.5 ejabberd....40
4.2 Analisi, installazione e configurazione del software per la collaborazione...42
4.2.1 Alfresco Community Edition....42
4.2.2 Etherpad Lite....44
4.2.3 DAViCAL....46
4.2.4 Kronolith....49
4.3 ownCloud....51
4.4 Funambol....52
5 RISULTATI OTTENUTI E CONCLUSIONI....55

INTRODUZIONE

L'ultimo decennio informatico ha segnato un importante cambiamento nell'abitudinario digitale della vita di ognuno di noi. La disponibilità di nuovi componenti hardware sempre più potenti, meno costosi, di dimensioni ridotte ed efficienti da un punto di vista energetico, ha permesso il proliferare di nuovi dispositvi mobili (smartphone e tablet) che hanno ormai eguagliato in termini di capacità di calcolo anche i notebook di ultima generazione, e attraverso questi dispositivi coordiniamo ormai gran parte delle nostre vite digitali.

Internet, con la sua distribuzione dell'informazione alla velocità della luce e con una rete ormai capillarizzata su tutto il pianeta ci ha abituato ad avere una percezione dell'informazione sempre disponibile, veloce e sintetica. Tramite una serie ridotta di operazioni possiamo acquistare un libro e farcelo recapitare a qualsiasi indirizzo desideriamo, prenotare un biglietto aereo, intrattenere video conferenze con più persone, utilizzare strumenti collaborativi per la scrittura di documenti, accedere e commentare articoli di diverse testate giornalistiche e altro ancora.

Negli ultimi anni, sembra quasi avessimo stretto un grande rapporto di fiducia con Internet, siamo sempre meno scettici nell'adozione di nuovi servizi e sempre più connessi. Ogni giorno affidiamo a questi servizi un numero sempre maggiore delle nostre informazioni personali: amici, interessi, hobbies, gusti, datori di lavoro, livello di istruzione, scuole e università frequentate, fotografie, video, posizioni geografiche, appunti vari, files e tanto altro ancora.

Ormai siamo abituati ad interfacciarci con tutti questi servizi da qualunque tipo di dispositivo (PC, Notebook, Smartphone e Tablet), quasi se le nostre vite dipendessero da loro, così tanto che l'adozione di questo nuovo approccio alla vita digitale, insieme all'incremento di strumenti per la virtualizzazione di sistemi operativi, ha incentivato lo sviluppo ed il definirsi di un nuovo insieme e paradigma di tecnologie informatiche che prende il nome di "Cloud Computing".

Un numero sempre più crescente di aziende, organizzazioni, istituzioni, governi e privati hanno deciso di demandare a terzi - aziende Internet - la fornitura di una serie di servizi e risorse di calcolo accessibili tramite un comune browser web. Applicazioni di contabilità, CRM, ERP, strumenti collaborativi, posta elettronica e altri ancora, sempre più integrati tra loro e in produzione su infrastrutture Cloud, gestiscono e controllano le dinamiche di intere organizzazioni.

L'opportunità che offre il Cloud Computing è costituita dall'apparente alta disponibilità dei servizi, dai bassi - se non nulli - costi di attivazione e di gestione, e dalla rapidità con cui quest'ultimi possono essere erogati.

Oggi le aziende possono abbattare i costi di gestione dell'infrastruttura software demandando il tutto a terzi, ma a che prezzo? Le insidie, i dubbi e le critiche sollevate su questa nuova tendenza vertono principalmente sulla privacy: riservatezza dei dati, proprietà dei dati, continuità dei servizi, disponibilità dei servizi, sicurezza, formato dei dati e utilizzo di tecnologie proprietarie. Sono tutti aspetti assai importanti che spesso si perdono nell'opportunità di risolvere in un batter d'occhio una grande quantità di problemi relativi al mantenimento interno di tutta una infrastruttura dedita alla gestione del reparto software aziendale.

La soluzione a tutti questi aspetti è rappresentata dal Private Cloud, ovvero la

realizzazione di una infrastruttura Cloud Computing, privata e interna all'azienda che cerchi il più possibile di fornire gli stessi servizi offerti di infrastrutture Cloud Pubbliche (in gergo chiamate Public Cloud).

La motivazione e lo scopo principale di questo lavoro consiste nello studio e nella realizzazione di una soluzione Cloud Computing basata su un modello di distribuzione di tipo Privato. Nell'ambito del Cloud Computing, Coopyleft è un'azienda informatica di Cosenza che opera già da diversi anni nel settore ed in particolare con sole tecnologie libere attraverso l'ausilio di Software Libero.

Analizzeremo e configureremo una serie di applicazioni Software Libero per la costruzione di un sistema Software as a Service base che potrà in futuro esser preso in considerazione per eventuali ampliamenti, miglioramenti e particolari verticalizzazioni aziendali.

Si fornirà qualche cenno storico e una definizione ben più chiara del termine "Cloud Computing" che è oggi oggetto di forte discussione e criticità; si approfondiranno nello specifico il "Software as a Service" e il "Private Cloud", che rappresentano rispettivamente un service model ed un deployment model di un sistema Cloud Computing.

1 COS'È IL CLOUD COMPUTING

Il concetto chiave che sta alla base del cloud computing risale al 1960, quando John McCarthy affermò che probabilmente un giorno le risorse di calcolo saranno organizzate sotto forma di risorsa pubblica.

Dal 1960 ad oggi la crescita esponenziale della capacità di calcolo degli elaboratori, la rapida diffusione e diversificazione - per utilizzo, dimensione e potenza - degli stessi, unita all'avvento di nuovi strumenti, tecnologie e infrastrutture nel campo dell'informatica (come linguaggi di programmazione, sistemi operativi, protocolli di comunicazione, la rete Internet ed i sistemi di virtualizzazione delle risorse di calcolo) hanno permesso la convergenza di un numero sempre crescente di servizi, attività ed elaborazione dati verso grandi centri di calcolo (o data center) specifici per determinate mansioni.

Oggi giorno, appare sempre meno necessario possedere un PC o un Laptop per svolgere le proprie attività quotidiane, questo perché la maggior parte di esse possono essere comodamente gestite tramite thin clients, come ad esempio dispositivi mobili smart phones o tablets; e dal momento che la quasi totalità della fase di elaborazione e di archiviazione dei dati è affidata a servizi esterni, raggiungibili attraverso la rete Internet e da una gran parte di dispositivi, la capacità di calcolo di quest'ultimi può essere drasticamente ridotta. Possiamo comunicare con i nostri amici, guardare video, leggere la posta elettronica, scrivere documenti di testo, fogli di calcolo e tanto altro, senza avere installato uno specifico software per ciascuna di queste attività e senza preoccupazioni riguardo l'integrità e disponibilità dei dati stessi.

Ed è a questo punto che il confine tra l'utilizzatore e l'elaborazione dei dati assume una certa importanza. Per questo motivo, nel 1997, Ramnath K. Chellappa (professore presso l'Emory University di Atlanta) coniò il termine Cloud Computing e ne diede una prima definizione accademica [1]:

> *"Un paradigma informatico dove i confini dell'elaborazione dei dati vengono determinati più da ragioni economiche che da limiti tecnici."*

La definizione in sé fa capire ben poco su cosa sia un sistema di Cloud Computing e di come ciò possa essere organizzato. Difatti, stando ad una definizione ben più recente, organizzata e maggiormente dettagliata, fornita dal NIST [2] (National Institute of Technology and Standards) nel 2011, possiamo meglio descrivere il Cloud Computing come un modello - un insieme di strumenti informatici - per consentire un accesso perpetuo, conveniente e su richiesta verso una rete di risorse di calcolo programmabili (come ad esempio: server, storage, applicazioni, servizi e altro) che possono essere rapidamente configurate ed erogate con un minimo sforzo di gestione, o di interazione con il fornitore del servizio stesso.

Questo modello si compone da cinque caratteristiche essenziali, tre modelli di servizio (service models) e quattro modelli di distribuzione (deployment models).

Caratteristiche essenziali:

On-demand self-service: Un utilizzatore del servizio può unilateralmente alterare la fornitura della capacità di calcolo, come tempo macchina e storage, in modo automatico secondo le proprie necessità, senza richiedere l'interazione dell'uomo per ciascun servizio.

Broad network access: Le funzionalità sono disponibili sulla rete e accedute

tramite meccanismi standard per promuovere un utilizzo eterogeneo di piattaforme client thin oppure thick (ad esempio: smart phone, tablet, laptop oppure PC.)

Resource pooling: Le risorse di calcolo disponibili vengono raggruppate per servire più consumatori attraverso un modello multi-tenant, con risorse fisiche e virtuali che vengono assegnate e riassegnate in modo dinamico in base alla domanda. C'è un senso di indipendenza dal luogo per cui un utilizzatore non ha alcun controllo e conoscenza riguardo l'esatta localizzazione delle risorse messe a disposizione, ma in alcuni casi è possibile specificare la posizione ad un livello di astrazione superiore (ad esempio: paese, stato oppure datacenter). Esempi di risorse includono storage, elaborazione, memoria e larghezza di banda della rete.

Rapid elasticity: Le funzionalità possono essere fornite e rilasciate in modo elastico, in alcuni casi automaticamente, in base alla domanda degli utilizzatori stessi. Al cliente, le funzionalità fornite appaiono illimitate, come se potessero essere accedute in ogni momento e in qualsiasi quantità.

Measured service: I sistemi cloud controllano e ottimizzano in modo automatico l'uso delle risorse, sfruttando degli indici di misurazione - secondo certi criteri di astrazione - per ciascun tipo di servizio (ad esempio: storage, elaborazione, larghezza di banda e account utenti attivi). L'uso delle risorse può essere monitorato e controllato in modo trasparente sia per il fornitore che per l'utilizzatore.

Service Models:

Software as a Service (SaaS): La funzionalità fornita al cliente consiste nell'utilizzare un'applicazione di un fornitore che è in esecuzione su una infrastruttura cloud. Le applicazioni sono accessibili da diversi dispositivi client attraverso interfacce leggere, come un browser web (ad esempio: una email web-based). L'utilizzatore non gestisce e non ha il controllo sull'infrastruttura cloud sottostante, come la rete, i server, i sistemi operativi, l'archiviazione oppure alcune funzionalità specifiche dell'applicazione. Qualche eccezione potrebbe essere relativa alla configurazione di una qualche impostazione dell'applicazione.

Platform as a Service (Paas): La funzionalità offerta consiste nel permettere all'utilizzatore di mettere in produzione una propria applicazione o un'altra acquisita da terzi, su una infrastruttura cloud, utilizzando linguaggi di programmazione, librerie, servizi e strumenti supportati dal fornitore. L'utilizzatore non ha il controllo e non gestisce l'infrastruttura cloud sottostante, ma ha un controllo delle impostazioni di configurazione della sua applicazione e dell'ambiente di hosting.

Infrastructure as a Service (IaaS): La funzionalità consiste di mettere a disposizione dell'utilizzatore risorse di calcolo fondamentali come potenza di elaborazione, archiviazione e rete, in cui lo stesso utilizzatore deciderà di utilizzare in modo arbitrario (ad esempio: installare un sistema operativo e relative applicazioni). Anche in questo caso, l'utilizzatore non ha il controllo dell'infrastruttura cloud sottostante, ma può gestire il sistema operativo, il sistema di archiviazione e le applicazione in produzione.

Deployment Models:

Private cloud: L'infrastruttura cloud è fornita per un uso esclusivo da parte di una singola organizzazione comprendente una serie di utilizzatori (es. dipendenti). Essa può essere di proprietà, gestita e utilizzata dalla stessa organizzazione, da terzi, oppure in combinazione.

Community cloud: L'infrastruttura cloud è fornita per un uso esclusivo da parte di una comunità di utilizzatori di varie organizzazioni che condividono uguali interessi. Essa può essere di proprietà, gestita e utilizzata da una o più organizzazioni della comunità, da terzi, oppure in combinazione.

Public cloud: L'infrastruttura cloud è fornita per un uso aperto da parte del pubblico. Essa può essere di proprietà, gestita e utilizzata da un'azienda, da una università, da una organizzazione governativa, oppure da una loro combinazione.

Hybrid cloud: L'infrastruttura cloud è una composizione di due o più infrastrutture cloud distinte (private, community, oppure public) che mantiene le entità uniche, ma le associa a livello di portabilità di dati e di applicazioni attraverso la condivisione di una qualche tecnologia proprietaria o standard. (un esempio di questa infrastruttura potrebbe essere relativa al load balancing tra infrastrutture cloud).

Più in generale, oggi le risorse di calcolo (siano esse applicazioni e/o server) vengono rilasciate da un fornitore, un Cloud Provider, in forma pubblica e in certi casi, anche in pay-per-use. Le risorse possono essere rilasciate a diversi livelli a seconda delle esigenze del fruitore, a partire dall'applicazione che il cliente può utilizzare all'istante, al server su cui si ha completo controllo e capacità di personalizzazione. la definizione fa pensare molto a risorse "virtuali". In effetti è proprio quello che sono, la tecnologia centrale del Cloud Computing è appunto la virtualizzazione.

Ma un uso sempre più crescente di soluzioni basate su Cloud Computing, cosa può comportare? Quali sono i possibili utilizzi? I vantaggi? Innanzitutto scompare quella fase in cui l'utente acquisisce i propri mezzi, in un certo senzo l'informatica del Cloud è l'informatica senza computer, tutto risiede in modo virtualizzato, su di una infrastruttura distribuita, accessibile via Internet. Ora l'utente non ha bisogno di utilizzare il proprio PC, può usufruire delle sue applicazioni da qualunque postazione. La piccola azienda che ha bisogno di una piattaforma online non deve porsi più il problema di investire sui propri server, pagare un team di programmatori per lo sviluppo di un servizio ad-hoc o cercare un servizio con le caratteristiche adeguate, generalmente una soluzione Cloud fornisce tutto il necessario ad un costo ridotto; non solo, mentre spesso i servizi standard risultano inadeguati, le risorse insufficienti o i costi di gestione troppo alti, la promessa del Cloud è di avere sempre e comunque il tipo di risorsa adatto, con tempi di cambiamento strettissimi. Ciò che prima richiedeva settimane (acquisire nuove macchine, acquisire personale, configurare il software e fornire personalizzazioni) ora si è ridotto a pochi minuti, e quelli che erano i costi più vari (hardware, tecnici, elettricità, raffreddamento, consulenze, outsourcing) ora si traducono in una sola formula, "si paga quello che si usa".

Dal punto di vista del provider, invece, i benefici non sono minori. Dotarsi di una infrastruttura Cloud vuol dire riuscire a soddisfare varie tipologie di esigenze. Con gli stessi server a disposizione il modello Cloud permette di tenere pronti diversi tipi di ambienti sempre preparati a soddisfare le richieste dei clienti.

Il Cloud Computing rappresenta un radicale cambiamento delle modalità di approvvigionamento delle risorse. Negli ultimi anni ha avuto una crescita incredibile, dovuto soprattutto a grossi investimenti da parte di grosse aziende. Oggi esistono diverse conferenze sul tema, la più importante è la Cloud Computing Expo [3], la prima tenutasi nel 2007 a New York con 450 delegati, la prossima si terrà nuovamente a New York e conta già 10.000 delegati e oltre 1.000 sponsor.

1.1 Software as a Service

Con il termine Software as a Service (più brevemente chiamato SaaS) si indica un modello di distribuzione del software su richiesta - "on-demand software" - in cui sia l'applicazione che i dati sono centralizzati in un'unica infrastruttura cloud. Come detto nel paragrafo precedente, l'accesso da parte dell'utilizzatore a questo genere di software avviene per mezzo di thin client usando un comune browser web.

Oggi, il modello Software as a Service rappresenta la soluzione comune e largamente usata per molti software di carattere aziendale, come ad esempio: Customer Relationship Management (CRM), Collaborazione, Management Information Systems (MIS), Enterprise Resource Planning (ERP), Human Resource Management (HRM) e Content Management (CM). Ed è fortemente tenuto in considerazione in tutte le strategie di mercato delle più grandi compagnie di software.

La fornitura di applicazioni aziendali attraverso una infrastruttura centralizzata non è un concetto nuovo. Già dal 1960 in poi, IBM insieme ad altri produttori di mainframe portarono avanti un progetto per delle compagnie di servizi finanziari. Lo scopo di questo progetto consisteva di mettere a disposizione delle banche e di altre grosse organizzazioni, la potenza di calcolo e l'archiviazione dati attraverso dei Data Center distribuiti a livello mondiale.

L'espansione di Internet durante gli anni '90 diede vita ad una nuova classe di Calcolo Centralizzato, chiamato Application Service Providers (ASP) [4]. Il mercato degli ASP si basava sulla fornitura di hosting e di gestione di applicazioni aziedali specifiche, con l'obiettivo di ridurre notevolmente i costi di gestione tramite l'esistenza di una amministrazione centralizzata e anche attraverso la verticalizzazione di diverse applicazioni, realizzate dallo stesso fornitore del servizio.

Essenzialmente, il concetto di Software as a Service, tende ad estendere quello dell'ASP. In particolare, il termine SaaS viene usato per distinguere alcune caratteristiche fondamentali che lo separano dal modello ASP:

- Gli ASP si limitavano a gestire e ospitare del software sviluppato da terze parti. Nel 2012 i produttori di Software as a Service puntano a sviluppare e gestire loro stessi, sia i servizi che il Software.
- Gli ASP fornivano un più tradizionale modello di utilizzo Client-Server che richiedeva l'installazione di un software client su ogni computer dell'utilizzatore, il modello SaaS si affida in via predominante sul web e richiede un browser web per poter essere utilizzato.
- Nel modello ASP venivano installate e configurate tante instanze dello stesso Software per quanti fossero i clienti, nel modello SaaS si utilizza normalmente un'architettura di tipo multi-tenant, in cui esiste un'unica infrastruttura SaaS che viene utilizzata da più gruppi e utenti.

La prima forma di servizi Cloud fornita attraverso il modello Software as a Service è stata realizzata da Salesforce.com che nel 1999 aveva reso possibile utilizzare la propria soluzione CRM tramite web.

1.1.1 Architettura SaaS

La maggior parte delle soluzioni SaaS si basano su una architettura di tipo multi-tenant. Con l'espressione Multi-tenant facciamo riferimento ad un principio dell'architettura software che prevede l'esecuzione su un server di una singola istanza di un software, utilizzata per gestire le richieste di più organizzazioni e clienti (tenants). Questo tipo di architettura è in netto contrasto con la multi-instance, in cui vengono configurate più istanze software per gestire le richieste di più organizzazioni.

In una architettura di tipo multi-tenant, il partizionamento dei dati, il confine tra un dato di una organizzazione con quello di un'altra, viene definito attraverso un'ulteriore astrazione software.

Un'architettura multi-tenant deve essere anche scalabile, per questo motivo la stessa applicazione viene installata su più server (parliamo quindi di scalabilità orizzontale). In alcuni casi, specialmente durante le fasi di collaudo, una seconda versione dell'applicazione viene configurata in modo tale da offrire ad un insieme ristretto di utilizzatori, la possibilità di accedere ad una versione in fase di testing o di pre-release.

Sebbene un'architettura di tipo multi-tenant è un componente necessario per una infrastruttura cloud SaaS, è ancora oggetto di discussione e di controversie.

Un'alternativa al multi-tenant, per gestire un grande numero di clienti e abbattere i costi di gestire e la stessa complessità del software, è rappresentata dalla virtualizzazione. In uno scenario simile, vengono installate più macchine virtuali con diverse installazioni SaaS, ognuna per un numero prestabilito di clienti.

1.1.2 Integrazione e protocolli aperti

Uno dei fattori chiave che ha prodotto un aumento dell'adozione di applicazioni Software as a Service (SaaS) da parte di grosse aziende e istituzioni, è certamente rappresentato dall'incredibile riduzione dei costi di gestione delle vecchie soluzioni, le quali costituivano costi per l'acquisto e la configurazione dell'intera infrastruttura, acquisto di licenze d'uso, costi di personalizzazione, aggiornamento e ordinaria manutenzione. Grossi investimenti inziali, combinati con ritorni imprevedibili sullo stesso investimento, hanno convinto le aziende a cercare alternative meno costose e a basso rischio.

L'altenativa, in questo caso, è rappresentata dalle soluzioni SaaS, che richiedono costi minimi o nulli per la manutenzione dell'intera infrastruttura, possono essere abilitate e messe in produzione molto velocemente e nei casi in cui il modello viene fornito in "pay-per-use", forniscono un modello di costo prevedibile, a basso rischio e ritorno sull'investimento più rapido.

La domanda sempre più crescente da parte di aziende e organizzazioni nell'adozione di soluzioni Cloud ha portato un'ulteriore innovazione nelle applicazioni SaaS e in particolar modo negli strumenti di integrazione. Ancora oggi, molte aziende hanno l'esigenza di integrare le applicazioni SaaS con altre soluzioni cloud e anche con quelle di back-office ancora in funzione. Senza degli strumenti di integrazione opportuni, le soluzioni SaaS hanno poco o nessun valore. L'integrazione, e dunque l'interoperabilità di più soluzioni SaaS tra loro, rappresenta oggi uno degli argomenti di sviluppo ancora aperti ed una sfida accettata da parte delle più grosse aziende che sono nel mercato del Cloud Computing.

In genere, una o più applicazioni SaaS dovrebbero prevedere alcuni o tutti dei livelli di

integrazione elencati qui sotto:

Autenticazione

L'autenticazione dell'utente verso le applicazioni SaaS deve avvenire nel modo più semplice possibile. L'utilizzo di più applicazioni SaaS da parte di uno stesso utente o di una stessa organizzazione, dovrebbe garantire un sistema di autenticazione basato su una base dati utente unica o meglio, garantire l'autenticazione per mezzo di un Identity Provider (un provider esterno che conferma l'identità di un utente). In questo modo, l'utente non dovrà creare e abilitare tanti account per ogni applicazione.

Accessibilità

Il servizio deve poter essere acceduto da parte di diversi dispositivi (PC, Notebook, Tablets o Smartphones) e per mezzo di un thin client (ad esempio un browser web).

Spesso, molte soluzioni SaaS sviluppano dei thin client specifici e ottimizzati per diversi supporti. In genere, l'accesso alle informazioni avviene sempre via HTTP.

Interoperabilità

I vari servizi SaaS devono fornire opportuni strumenti per l'importazione ed esportazione dei dati gestiti. In genere, il processo di importazione e di esportazione avviene processando formati dati aperti e specialmente human readable (quali ad esempio: Plain text ASCII, CSV, XML e Open Document Format). Ciò fa sì che diverse soluzioni SaaS possano gestire dati provenienti da altre soluzioni.

D'altra parte, permette all'utilizzatore di estrarre le informazioni per una eventuale migrazione su altri servizi e/o elaborazioni interne da parte di vecchie applicazioni.

Definizione di opportune API

Applicazioni SaaS di ultima generazione forniscono definizioni e implementazione di API per consentire l'accesso agli stessi servizi da parte di software personalizzato o per garantire a vecchie soluzioni software di poter essere adeguatamente aggiornate per comunicare con i servizi SaaS.

1.2 Private cloud

Nel caso del Private Cloud, facciamo riferimento ad un'azienda oppure ad una organizzazione che si munisce di una infrastruttura Cloud (personale) per erogare servizi sia interni che esterni alla propria attività. Questo modello è utile per tutte quelle organizzazioni che necessitano o preferiscono avere maggiore controllo sui propri dati e di modalità di servizio personalizzate.

Il modello Private Cloud permette all'organizzazione di centralizzare il controllo sui propri server, implementando i servizi in un unico o più data center. Ciò richiede un numero di personale ridotto, un minore dispendio di energia e attrezzatura per il raffreddamento.

Il Private Cloud diventa vantaggioso per un'organizzazione che dispone di un elevato numero di server (da qualche decina in su) che erogano diversi servizi, magari su macchine ridondate per garantire efficienza e continuità del servizio.

Una infrastruttura di questo tipo genera molti sprechi se c'è disparità di utilizzo dei vari servizi o discontinuità di utilizzo nel tempo. La virtualizzazione di molte macchine su un numero limitato di server permette di ottimizzare le risorse, riducendo gli sprechi. Il paradigma del Cloud Computing consente di spostare dinamicamente le risorse laddove è necessaio, con una elasticità che su grandi numeri paga in termini di efficienza, riduzione dei costi ed anche dei consumi energetici.

In genere, una soluzione Private Cloud è consigliabile per tutte quelle istituzioni pubbliche e governative che scelgono di adottare applicazioni SaaS per la gestione di tutte le attività riguradanti l'informatizzazione di diverse mansioni (dalla gestione delle anagrafiche a tutta una serie di servizi a disposizione dei cittadini). In questa ottica, affidare a terzi, la gestione dell'intero sistema informativo di un governo o di una grossa istituzione potrebbe rappresentare un aspetto per la sicurezza e la disponibilità dei dati, troppo delicato.

2 PUBLIC E PRIVATE CLOUD SAAS

Nella prima parte di questo capitolo descriveremo le componenti fondamentali di una infrastruttura Public Cloud SaaS, ovvero, un caso pratico di una implementazione di un service model di tipo Software as a Service distribuito con un modello Public Cloud.

Nella seconda parte ci occuperemo di fornire un'analisi dettaliata di una possibile infrastruttura Private Cloud basata esclusivamente su Software Libero come alternativa all'infrastruttura Public Cloud descritta in precedenza. Cercheremo di analizzare tutte le varie opzioni SaaS Software Libero, suddividendole per dominio applicativo, e di scegliere quelle più idonee per la realizzazione di una infrastruttura Private Cloud SaaS con Software Libero.

In ultimo, metteremo in evidenza sia i pregi che i difetti che entrambe le soluzioni Public e Private possono presentare.

2.1 Il public cloud SaaS secondo Google

Il modello Cloud più diffuso è senz'altro il Public Cloud. In questo contesto il servizio viene distribuito da provider tramite internet in forma pubblica. I servizi Public Cloud possono essere offerti gratuitamente o in forma pay-per-use.

Questa tipologia è una forma di Utility Computing, ovvero la possibilità di usufruire di risorse computazionali in modo simile alle risorse pubbliche (elettricità, acqua, gase, etc), in modo trasparente (per utilizzare l'energia elettrica non è necessario conoscere la rete, allo stesso modo, per utilizzare le risorse computazionali non è necessario conoscere l'infrastruttura), a consumo e con minima o nessuna competenza tecnica.

Il modello Public è particolarmente adatto per enti privati e no-profit che necessitano di risorse per periodi di tempo limitati, o di potenza di calcolo scalabile. Utilizzando risorse Cloud si hanno abbattimento dei costi di acquisizione hardware, di reperimento personale e significativa riduzione dei tempi.

A livello realizzativo, per erogare servizi Public richiede enormi investimenti e soltanto le grosse aziende del campo informatico come Amazon, Microsoft e Google sono entrate nel campo. Il Public Cloud è implementato su migliaia di server (virtuali e non) distribuiti in centinaia di data center dislocati in varie parti del mondo.

Google, oggi rappresenta una delle più grandi aziende nel settore dell'informatica, nel Cloud Computing e in particolar modo nell'erogazione di servizi Public Cloud sia in forma gratuita che in pay-per-use. In questi ultimi anni, ha specializzato e potenziato molte delle sue attività nel Software as a Service e nel Platform as a Service, introducendo nuovi servizi e tecniche di integrazione tra le varie soluzioni proposte.

Ogni giorno, milioni di utenti (tra cui aziende, enti no-profit, istituzioni, università e scuole) accedono e fanno regolarmente uso di molti servizi Google. Tutti questi servizi sono in esecuzione e gestiti attraverso una infrastruttura di tipo Public Cloud.

Affinchè una infrastruttura così grande possa funzionare, esiste dunque un'implementazione completa su tutti i livelli di gerarchia dei Service Models che sono stati discussi nel primo capitolo: Software as a Service, Platform as a Service e Infrastructure as a Service, che sono distribuiti su tutta la rete di data centers Google dislocati in varie parti del mondo.

L'insieme di tutte le soluzioni SaaS di Google viene denominato con il termine "Google Apps" [5], da non confondersi con "Google App Engine" [6], che rappresenta la soluzione PaaS di Google.

Come spiegato nel paragrafo 1.3.2 ("*Integrazione e protocolli aperti*"), uno degli aspetti più importanti di una infrastruttura cloud che intende includere una serie di soluzioni SaaS, è rappresentato dal livello di integrazione che quest'ultime hanno l'una rispetto all'altra, dalla facilità da parte degli utilizzatori di autenticarsi e accedervi con estrema facilità, dalle modalità di autenticazione e dalle possibilità di interazione con gli stessi servizi attraverso opportune API.

Il primo aspetto che approfondiremo riguarda la gestione centralizzata delle utenze, dei metodi e dei protocolli utilizzati per realizzare un'autenticazione unica e cross-domain.

2.1.1 Single-Sign-On

Il Single-Sign-On è una funzionalità per il controllo dell'autenticazione utente su diversi sistemi software completamente indipendenti tra loro ma correlati comunque per un qualche motivo (ad esempio: software di uno stesso ente). Il concetto chiave di questa funzionalità è rappresentato dal fatto che una volta ottenuto l'accesso ad un particolare software, si è automaticamente autenticati su tuttli gli altri software senza dover ripetere più volte la procedura di autenticazione.

Il caso di studio che abbiamo preso in considerazione, le Google App, rappresenta un ottimo esempio laddove la funzionalità di Single-sign-on si rivela di fondamentale importanza. Google ha unito un ventaglio abbastanza ampio di software, apparentemente indipendenti tra loro, fornendo una base dati utenti comune. Infatti, autenticandosi ad una sola applicazione di Google si abilita automaticamente l'accesso a tutte le altre.

Il sistema di autenticazione Single-Sign-On di Google [7] è basato su una implementazione del protocollo OpenID [8], uno standard aperto che descrive la modalità con cui gli utenti possono essere autenticati in una maniera decentralizzata, eliminando la necessità da parte del servizio offerto (come una stessa applicazione), di mantenere un sistema ad hoc proprietario per la verifica delle credenziali di accesso. Gli utenti, in questo modo, creano il proprio account su un Identiy Provider OpenID arbitrario (in questo caso Google stesso), e successivamente usano l'account appena creato per effettuare l'accesso verso tutti i servizi/applicazioni web/SaaS che accettano un'autenticazione OpenID.

Google sfrutta questo meccanismo sia per autenticare lo stesso account utente su tutte le varie soluzioni SaaS presenti nelle Google App, e sia per agire come un Identity Provider per terze parti che accettano di autenticare i propri utenti attraverso un account Google.

Difatti, è facile fare confusione quando parliamo di utenze centralizzate e utenze decentralizzate. Se facciamo riferimento alle applicazioni messe a disposizione da Google, possiamo parlare di utenze centralizzate dal momento che esiste una sola base dati comune per il sistema di autenticazione usato da tutte le applicazioni.

D'altra parte, parliamo di utenze decentralizzate quando facciamo riferimento ad un'applicazione sviluppata da terzi, che non ha nulla a che vedere con le Google App, ma che utilizza comunque la base dati utente di Google per autenticare i propri utenti.

In questo caso, Google sta implementando un tipo di funzionalità Single-sign-on basato su un'architettura ad approccio federativo.

Per fornire un'idea più chiara di come questo sistema di autenticazione funzioni e di

quali siano tutti i passaggi e gli attori che vi partecipano, di seguito viene visualizzato il diagramma di interazione tra una web applicaction (o qualsivoglia servizo che necessita di autenticarsi) e il Google Login Authentication (che rappresenta il servizio di autenticazione di Google):

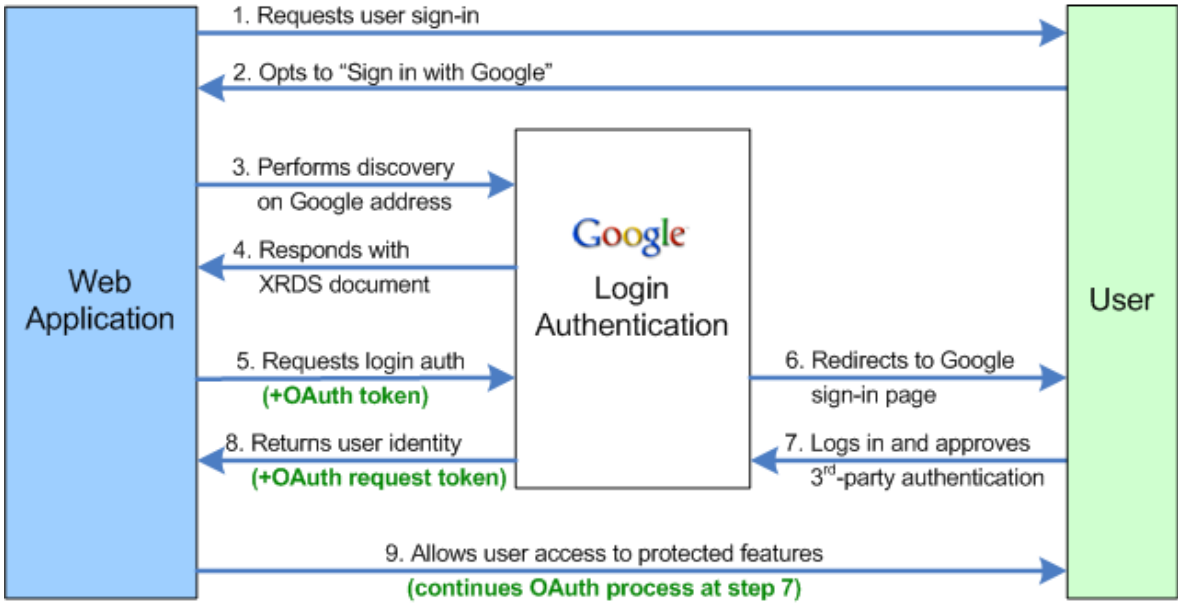

Fig. 2.1 - Diagramma di interazione tra una web application e il Google Login Authentication service.

Secondo le specifiche ufficiali del protocollo OpenID, i termini tecnici con cui si identificano i tre attori principali, User, Web Application e Google Login Authentication prendono rispettivamente il nome di: User-Agent (individua il web-browser usato per accedere al servizio), Relying Part (abbreviato RP) e OpenID Provider (abbreviato OP, individua il server preposto all'autenticazione dell'utente).

La fase di autenticazione (decentralizzata) viene descritta e portata a termine attraverso nove passaggi [9]:

1. Il Relying Part richiede all'utente la modalità di accesso che si vuole usare per autenticarsi nel sistema
2. L'utente sceglie di autenticarsi tramite l'OP di Google
3. Il Relying Part effettua una fase cosiddetta di discovery (cerca di risolvere l'indirizzo corretto, URL, a cui inviare la richiesta di autenticazione)
4. L'OP, una volta contattato, risponde con un documento in cui sono presenti diverse informazioni sul tipo di richiesta di autenticazione che è possibile realizzare e quale sia il reale URL a cui sottoporre la richiesta di autenticazione
5. Il Relying Part effettua una richiesta di autenticazione presso l'OP di Google
6. Se l'utente non ha effettuato l'accesso e/o l'applicazione in questione non è stata autorizzata dall'utente stesso, l'OP reindirizza lo User-Agent verso una pagina in cui vengono richieste le credenziali di accesso dell'utente e/o l'autorizzazione esplicita dell'utente ad accettare - come fidato - il Relying Part
7. Lo User-Agent invia una richiesta di autorizzazione (inserisce le proprie credenziali di accesso e/o approva l'applicazione)

8. L'OP, una volta ottenuta l'autorizzazione, restituisce l'esito dell'operazione di autenticazione al Relying Part
9. Il Relying Part, in accordo all'esito ricevuto, decide se permettere o rifiutare l'accesso allo User-Agent.

2.1.2 Analisi del software

Come accennato nel primo capitolo, l'obiettivo principale di questo lavoro consiste nel redigere una prima analisi ed uno studio di fattibilità sulla realizzazione di una soluzione Private Cloud SaaS con Software Libero, che possa essere facilmente estesa per supportare una varietà di software e anche un certo livello di integrazione.

La quantità di software disponibile nelle Google App e il complesso livello di integrazione che essi presentano, impediscono - almeno in prima analisi - di realizzare un sistema completo Private Cloud SaaS paragonabile alle Google App.

In questa prima fase di analisi prenderemo in considerazione solo le principali categorie ed i software più utilizzati delle Google App, e sulla base di questi, più avanti cercheremo di discutere le possibili alternative Software Libero, realizzando una controparte Private Cloud.

Le categorie prese in analisi sono quella di messaggistica e quella di collaborazione, rispettivamente si occupano di fornire funzionalità che consentono la comunicazione istantanea e non, di più utenti, attraverso software di posta elettronica, newsgroups e di chat, e di fornire strumenti collaborativi per la creazione di documenti online e l'organizzazione delle attività per più utenti.

In ultima analisi discuteremo di un servizio esterno alle Google App, Dropbox, che permette la sincronizzazione in tempo reale di files e cartelle presenti su diversi tipi di dispositvi (PC, Notebook, Tablets o Smartphones) con un file server remoto.

2.1.2.1 Messaggistica

- **Google Talk**

  È un servizio di Voice over IP e di messaggistica istantanea che usa come protocollo di comunicazione XMPP [10] (Extensible Messaging and Presence Protocol), un insieme di protocolli basati su XML che definiscono gli standard di comunicazioni di Instant Messaging.

  Basandosi su uno standard aperto, il servizio di messaggeria istantanea di Google può essere acceduto con qualsiasi client che implementi lo stesso protocollo.

  Allo stato attuale, Google Talk è integrato in diversi servizi di Google, come GMail e Google+ oppure, come software in versione stand-alone disponibile per diversi sistemi operativi e device (tra cui smartphone e tablet).

Fig. 2.2 - Logo del servizio Google Talk

- **Gmail (Google Mail)**

  È un servizio di posta elettronica che può essere acceduto attraverso diversi protocolli: via HTTPS per mezzo di una interfaccia web, via IMAP4 oppure via POP3 (per accedere tramite questi due ultimi protocolli è necessario utilizzare un qualche client di posta elettronica installato sul computer o dispositivo mobile dell'utilizzatore).

  Il livello di integrazione di questo servizio con altri della suite Google App è abbastanza alto, difatti è possibile sfruttare il servizio di messaggeria istantanea Google Talk direttamente dalla stessa interfaccia web, è possibile avviare chiamate vocali attraverso Google Voice ed esiste anche l'equivalente software per dispositivi mobili, quali smartphones o tablets.

  Stando alle statistiche del 2012, il numero degli utenti gestiti da questo servizio si aggira attorno ai 350 milioni.

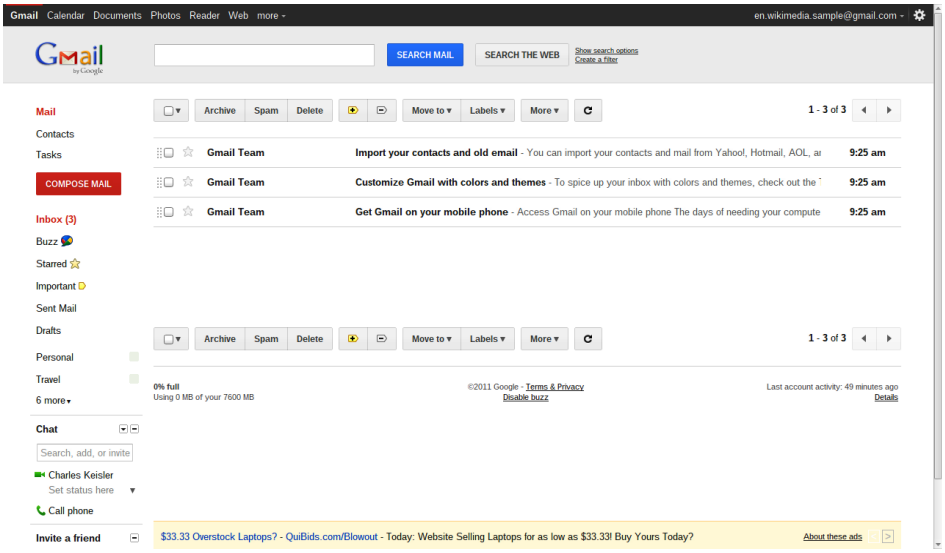

Fig. 2.3 - Interfaccia web del servizio GMAIL

- **Google Groups**

  È un servizio che mette a disposizione degli utenti la possibilità di creare gruppi di discussione, sottoscrivere un'iscrizione a quelli già esistenti e parteciparvi. L'interazione con il servizio può avvenire tramite una interfaccia web accessibile anche in versione mobile per smartphone e table, oppure, attraverso un client di posta elettronica è anche possibile inviare nuovi messaggi, come fosse un Newsgroup oppure una Mailing List.

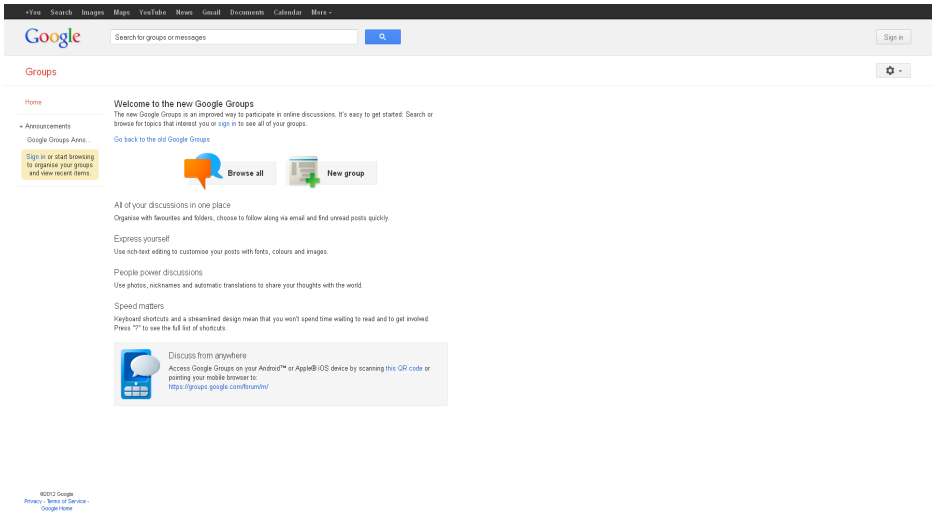

Fig. 2.4 - Interfaccia web del servizio Google Groups

2.1.2.2 Collaborazione

- **Google Docs**

  È una suite office "Software as a Service". Consente la creazione, l'importazione, l'esportazione e l'invio tramite email dei maggiori tipi di documento, come fogli di testo, di calcolo e di presentazione. I documenti possono essere modificati in tempo reale e in maniera collaborativa (l'accesso in contemporanea sullo stesso documento da parte di più utenti) direttamente via web, attraverso una interfaccia scritta prevalentemente in Javascript. I documenti possono essere salvati sul computer locale dell'utente in diversi tipi di formato (ODF, HTML, PDF, RTF, Text e Microsoft Office).

  Il servizio offre un buon livello di integrazione, è disponibile anche una versione per dispositivi mobili, quali smartphones e tablets. Nella versione a pagamento è presente anche un software aggiuntivo per Microsoft Office che consente l'archiviazione e la sincronizzazione dei documenti Word, fogli di calcolo Excel e presentazioni Power Point direttamente con Google Docs.

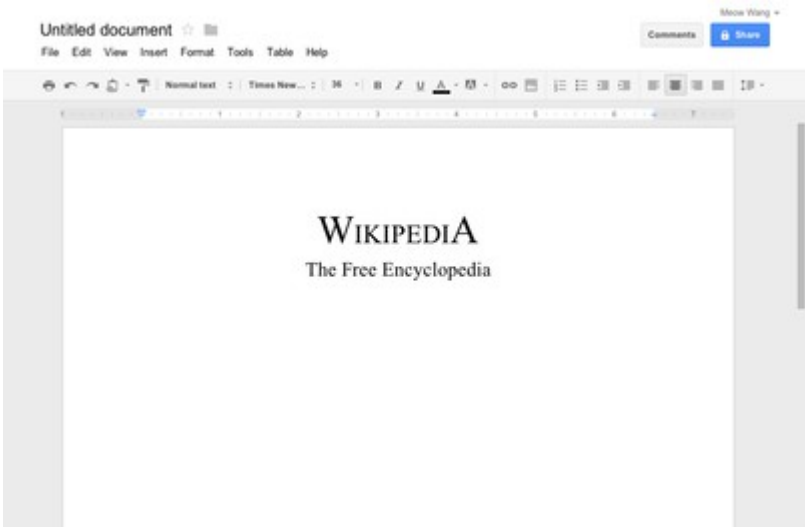

Fig. 2.5 - Un foglio di testo di esempio aperto sull'interfaccia web di Google Docs.

- **Google Calendar**

  È una piattaforma per l'organizzazione delle attività attraverso la funzionalità di un agenda online. Vi si accede per messo di una interfaccia web oppure tramite uno specifico software disponibile per smartphone e tablet. Consente la creazione di più calendari, così da differenziare ciascuna attività per tipologia (festività, lavoro, tempo libero, e così via) e lascia all'utente la possibilità di condividere ciascun calendario con altri utenti, di impostare i permessi di visibilità (pubblico o privato) e di notificare automaticamente - via email - la scadenza di qualche impegno.

  L'integrazione offerta da questo servizio è alta. Google Calendar implementa infatti **CalDAV** [11], un protocollo aperto basato su HTTP per l'accesso e la modifica dei dati presenti sul server, ed utilizza **iCalendar** [12] come formato per la presentazione dei dati. In questo modo, ogni software che è in grado di sfruttare questo protocollo e di supportare il formato di presentazione dei dati, può interagire con Google Calendar in modo trasparente, sfruttando il servizio Google Calendar come un Calendar Server.

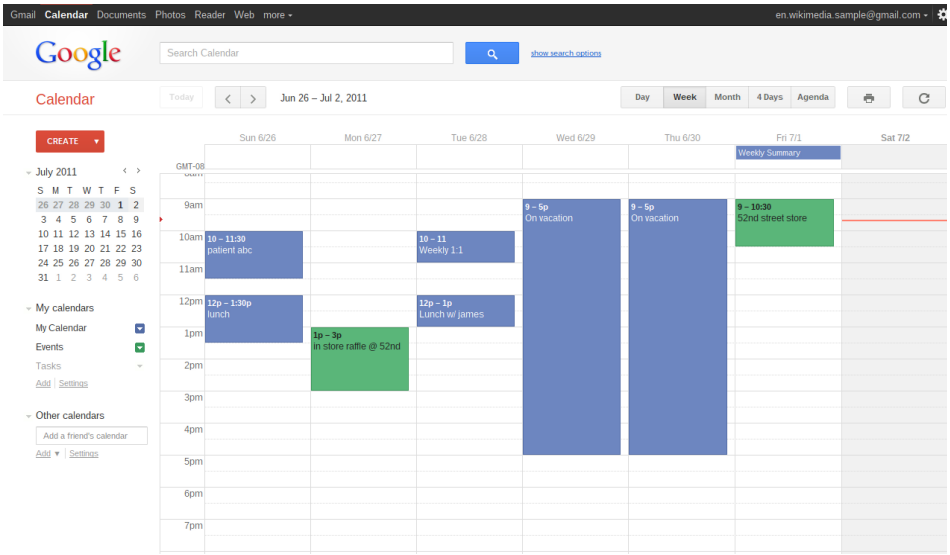

Fig. 2.6 - Interfaccia web di esempio di Google Calendar.

2.1.2.3 Altro

- **Dropbox**

  Il servizio mette a disposizione dell'utente, uno storage di rete virtuale, accessibile attraverso un software multipiattaforma disponibile per diversi dispositivi e sistemi operativi, L'utente può facilmente sincronizzare e condividere con altri utenti, file e cartelle presenti sui suoi dispositivi.

Fig. 2.7 - Logo di Dropbox.

2.2 Il private cloud SaaS con Software Libero

Il modello Software as a Service ha ragion d'esistere solo se vi è uno stack cloud completo, ovvero, esiste già una infrastruttura cloud Infrastructure as a Service e Platform as a Service su cui far poggiare il modello SaaS. Questo è imposto dalle caratteristiche essenziali del Cloud Computing: il sistema deve essere scalabile e costantemente monitorato da opportuni strumenti.

Qui discuteremo solo una possibile realizzazione private cloud SaaS senza approfondire i livelli cloud sottostanti, quali PaaS e IaaS. Daremo per certo che esiste già una infrastruttura cloud su cui operare e ci soffermeremo esclusivamente sul software.

La grande quantità di software e di soluzioni disponibili nel mondo del Software Libero ci consente di trovare con una certa facilità un insieme di potenziali applicazioni, che ben si prestano per la realizzazione di un sistema Private Cloud SaaS.

Un aspetto che invece assume ben altra difficoltà, è rappresentato dalle politiche di autenticazione per la realizzazione della funzionalità di Single-sign-on, o per lo meno, di garantire un'autenticazione che faccia uso di una sola base dati utente centralizzata affinché tutte le applicazioni scelte possano usarla per autenticare gli accessi.

2.2.1 Single-Sign-On

Come meccanismo per la gestione centralizzata dei dati utente si è optato per il protocollo LDAP [13] ed una sua implementazione server opensource openLDAP [14], che viene spesso adottato dalle maggiori distribuzioni GNU/Linux per l'autenticazione in ambienti di calcolo distribuito.

La gestione delle informazioni in LDAP è basata sul concetto di directory e di entry. Una directory è una struttura dati ad albero ottimizzata per l'accesso e la ricerca delle informazioni, utilizzata per la memorizzazione delle informazioni. Una entry è una raccolta di attributi che fanno riferimento ad un Distinguished Name (DN). Il DN è unico ed è usato per identificare in modo univoco una certa entry.

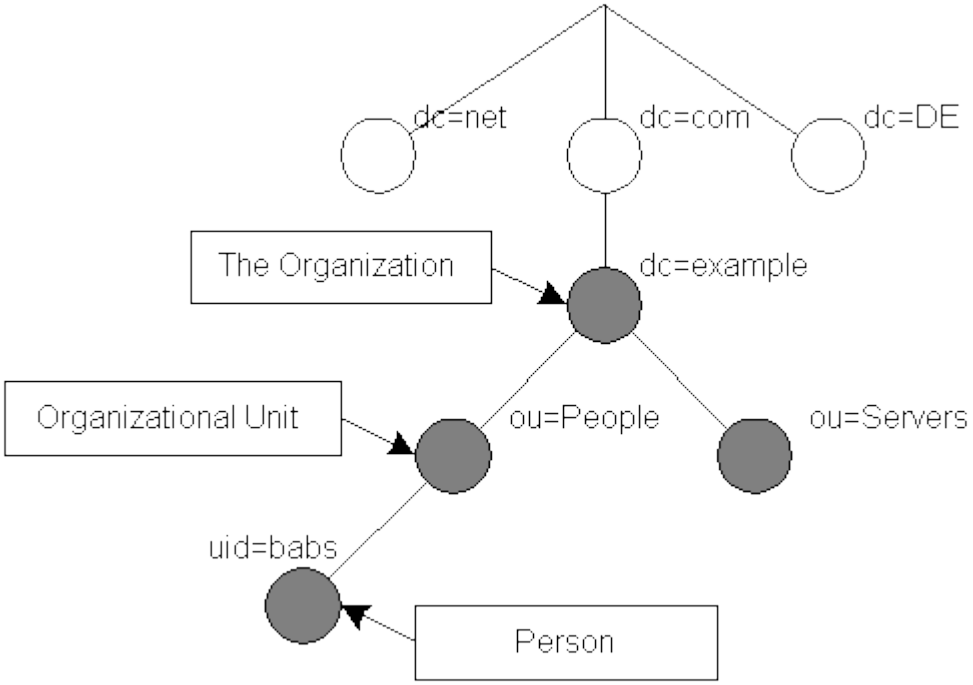

Fig. 2.8 - Esempio di una directory, della sua rappresentazione ad albero e di ciascuna entry.

Nella Figura 2.8 è mostrato un esempio di come sono organizzate le informazioni all'interno di una Directory (l'intero albero) e di ciascuna entry (ogni nodo). In questo scenario il DN dell'utente "babs" può essere costruito a partire dal nodo stesso e risalendo fino alla radice: [uid=babs,ou=People,dc=example,dc=com].

Tra le applicazioni principali della suite openLDAP troviamo `slapd` e `slurpd`, rispettivamente usate per la gestione delle informazioni contenute nella directory e per la replica delle stesse.

Per il nostro scopo ci occuperemo solo di `slapd`.

Per realizzare il servizio di Single-sign-on e anche di OpenID Identity Provider, di cui abbiamo già discusso nel paragrafo 1.1 di questo capitolo, è necessario utilizzare una implementazione OpenID che sfrutti come backend un server LDAP per la gestione delle informazioni utente. In questo caso, si è optato per l'utilizzo di `openid-ldap`. L'utilizzo di una configurazione OpenID e LDAP ci permette di raggiungere due risultati importati:

1. Centralizzare tutta la gestione delle informazioni attraverso un unico servizio
2. Fornire a terzi la possibilità di verificare l'attendibilità degli utenti attraverso la consultazione del nostro Identity Provider.

Come vedremo più avanti, durante l'analisi del software, allo stato attuale non siamo in grado di fornire una piena integrazione del servizio di Single-sign-on su tutti i software scelti, poiché diversi di questi non supportano l'autenticazione OpenID. D'altra parte, possiamo garantire un primo di livello di integrazione basato su autenticazione utente centralizzata, forzando tutti i software ad una interrogazione sul server LDAP per completare la fase di accesso utente.

2.2.2 Analisi del software

Qui cercheremo di prendere in considerazione tutte le alternative Software Libero alle soluzioni proprietarie elencate in precedenza e lo stato di integrazione che siamo riusciti ad ottenere.

Dal momento che il primo livello di integrazione che dobbiamo raggiungere è rappresentato dall'autenticazione centralizzata di tutti i software attraverso un unico sistema per la gestione delle informazioni utente, le soluzioni elencate di seguito presentano tutte un supporto LDAP.

Il sistema operativo preposto come base di tutta la configurazione, è una distribuzione GNU/Linux Debian [15] Unstable.

2.2.2.1 Messaggistica

- **ejabberd** [16]

  Server di Instant Messaging basato sul protocollo XMPP. Derivato dal progetto jabberd1, è stato completamente riscritto in Erlang/OTP, ottimizzato e potenziato con il supporto di nuove caratteristiche XMPP.

- **Postfix** [17]

  

  È un MTA (Mail Transfer Agent) libero e opensource, rilasciato sotto licenza IBM Public License 1.0 (compatibile con la definizione di Software Libero), che si occupa dell'instradamento della posta elettronica. Tra le sue caratteristiche principali, riconosciamo: delega delle politiche di SMTP a processi esterni, delega della consegna della posta attraverso processi esterni, supporto a diversi Database per la mappatura delle informazioni gestite e supporto ai virtual hosts. Per i nostri scopi, Postfix agirà da server SMTP.

- **Courier Mail Server** [18]

  È un server MTA che integra funzionalità di groupware/mail attraverso l'implementazione di diversi protocolli ESMTP, IMAP, POP3, LDAP, SSL, e HTTP. Noi ci occuperemo solo della sua configurazione IMAP.

- **Horde Groupware Suite** [19]

  È una suite completa per la messaggistica. Consente all'utente finale di leggere, inviare e organizzare la posta elettronica, gestire e condividere calendari, rubriche e note. Tutto il servizio è accessibile tramite un browser web e/o un thin client presente su smartphone o tablet.

- **GroupServer** [20]

  È un servizio basato sul web per la gestione di Mailing List/Newsgroups. Strumenti come le mailing list sono utili poiché permettono alle persone di collaborare e comunicare in gruppi, sfruttando la posta elettronica. GroupServer, oltre alla gestione del server per mailing list, mette a disposizione anche una interfaccia web, simile a Google Groups, per l'interazione via browser con i gruppi.

2.2.2.2 Collaborazione

- **Alfresco Community Edition** [21]

  È l'edizione open source di Alfresco: un'applicazione web che integra una serie di strumenti per l'ottimizzazione della gestione documentale in azienda.

- **Etherpad Lite** [22]

  È un editor di testi che consente una scrittura in via collaborativa e in tempo reale su uno stesso documento. È stato derivato direttamente da Etherpad, riscritto in gran parte, più leggero e ottimizzato rispetto al progetto iniziale, può anche essere integrato in diverse applicazioni già esistenti attraverso l'uso di opportune API basate su HTTP.

- **DAViCAL** [23]

  

  È un server per la condivisione di calendari. Fornisce una implementazione del protocollo CalDAV, il quale è stato progettato per l'archivizione e la gestione di risorse (nel formato iCalendar) su un server remoto. Le risorse gestite dal server possono essere accedute da qualsiasi dispositivo e client che supportino il protocollo CalDAV ed il formato dati iCalendar.

- **Kronolith** [24]

  È un client basato sul web per la gestione di calendari e per l'accesso a risorse remote (CalDAV e iCalendar). Viene integrato in Horde Groupware Suite.

2.2.2.3 Altro

- **ownCloud** [25]

  

  È una piattaforma web per la gestione di uno storage di rete virtuale, accessibile da qualsiasi dispositivo e sistema operativo che supporti i protocolli usati per l'interfacciamento ai dati archiviati da ownCloud. Consente di caricare, scaricare, condividere i file con diversi utenti e di sincronizzare cartelle locali del nostro computer con lo spazio virtuale.

- **Funambol** [26]

  

  È un server per la sincronizzazione di dati provenienti da più fonti (mail server, file server, calendar & contacts server, LDAP e così via) su diversi dispositivi e software (smartphone, tablets, PIM, etc). Viene utilizzato per sincronizzare in tempo reale la rubrica, i messaggi ed i file di dispositivi mobili, con dati provenienti da fonti esterne.

2.3 Critiche e sicurezza

2.3.1 Critiche sul modello Public

Il modello Public è molto diffuso ma anche molto criticato, le principali critiche riguardano:

Riservatezza

I dati personali vengono affidati a terzi, essi risultano in possesso dell'azienda e ciò comporterebbe, almeno in via ipotetica, l'accesso ed il controllo dei nostri dati da parte della stessa azienda fornitrice del servizio, per scopi di indagini di mercato e di profilazione dell'utente.

Continuità del servizio

Delegando a un servizio esterno la gestione dei dati e la loro elaborazione, l'utente di trova fortemente limitato nel caso in cui i suddetti servizi non siano più operativi, per una qualche ragione. Un eventuale malfunzionamento, inoltre, colpirebbe contemporaneamente un numero elevato di utenze dal momento che questi servizi sono condivisi.

Dipendenza da un singolo fornitore

Dal momento che i dati sono in possesso di un determinato provider, potrebbe essere complicato o addirittura impossibile migrare ad un altro fornitore, a causa di politiche interne del servizio, di formati chiusi o dell'impossibilità di fare backup. In gergo, questo fenomeno è chiamato "Lock-In".

Cambiamento improvviso del servizio

Nel caso in cui l'utente utilizzi un'applicazione sul proprio pc, può decidere se aggiornare il software o meno, o se installare un'applicazione con le medesime funzionalità. Per i servizi Cloud ciò non è valido, l'utente potrebbe trovarsi improvvisamento ad utilizzare un servizio o un'interfaccia diversa.

Cambiamento dei termini di servizio

Può capitare che un servizio gratuito diventi a pagamento, o che venga effettuato un aumento dei prezzi, in questo caso l'utente potrebbe trovarsi in condizioni spiacevoli, specie se ha assoluto bisogno del servizio.

Chiusura del codice

Nella maggior parte dei casi ci si trova ad utilizzare codice proprietario e formati chiusi.

2.3.2 Critiche sul modello Private

Negli ultimi tempi, le critiche sollevate riguardo problematiche relative la riservatezza dei dati e l'adozione di eventuali politiche di lock-in da parte di fornitori di servizi Cloud di tipo Public, hanno incentivato il finanziamento e lo sviluppo di soluzioni Cloud di tipo privato; In questo caso, un'organizzazione o un'azienda decide di realizzare privatamente, all'interno della stessa sede oppure all'esterna, un'infrastruttura Cloud completa ad uso privato.

Il modello Private Cloud risolve molte dei problemi che riguardano la sicurezza e la riservatezza dei dati; d'altra parte comunque, ha sollevato diverse critiche:

Costi

Implementare una completa infrastruttura di Cloud Computing su tutti e tre i livelli: SaaS, PaaS e IaaS comporta dei costi elevati che comprendono: acquisto di tutto l'hardware necessario, la relativa installazione, configurazione e gestione di tutto l'apparato (sostituzione di eventuali componenti difettosi, sviluppo di piattaforme o applicazioni ad-hoc, ed altro).

Rottura del modello economico del Cloud

Il modello economico del Cloud Computing si basa principalmente sull'alta disponibilità a basso costo di risorse e servizi informatici, che possono essere attivate su richiesta, senza alcuno sforzo e in ogni momento. Si considera un servizio Cloud come una qualsiasi altra risorsa pubblica, come l'energia elettrica, l'acqua e così via. In questo scenario, immaginiamo grandi data center che concentrano al loro interno una quantità elevata di servizi.

Al contrario, il Private Cloud rompe questo principio, come se producesse di suo la propria energia elettrica o gestisse privatamente la rete idrica. Il concetto di risorsa pubblica e la centralizzazione viene meno.

2.3.3 Aspetti di sicurezza

La sicurezza, nel contesto del Cloud Computing, si basa sulla fiducia. Fiducia verso il produttore dell'hardware, verso il produttore del software e verso l'Internet Service Provider. Nel paradigma Cloud bisogna fidarsi anche del fornitore del servizio. Tuttavia c'è una certa differenza, se si ha accesso fisico alla macchina, si ha maggiore controllo sulla sicurezza, configurando firewall e controllando direttamente il traffico, la fiducia verso i produttori non è quindi del tutto necessaria. Se la risorsa è in qualche modo fornita in via vrtualizzata da una infrastruttura Cloud, resta di fidarsi completamente del fornitore del servizio.

In genere, all'utente - utilizzatore finale del servizio - dovrebbero essere garantiti i seguenti aspetti:

Protezione dei dati

I dati devono essere archiviati in modo sicuro, nessun altro deve avere la possibilità di accedervi, non devono esserci perdite di dati e il trasferimento da una locazione ad un'altra deve essere altrettanto sicuro.

Controllo dell'identità

Devono essere forniti sistemi di autenticazione sicuri.

Sicurezza fisica

L'accesso alle macchine fisiche deve essere ristretto e controllato, nonchè l'accesso ai dati degli utenti.

Disponibilità

Deve essere garantito l'accesso ai dati e alle applicazioni in ogni momento.

Privacy

I dati sensibili (come numeri di carte di credito, e altro) devono essere opportunamente mascherati e l'accesso ad essi deve essere riservato ai soli autorizzati. L'identità digitale e le credenziali, nonchè le attività degli utenti devono essere altrettanto protette.

Continuità del servizio e recupero dei dati

Il servizio e l'accesso ai dati devono essere garantiti anche a seguito di problemi tecnici.

Recupero dei dati in caso di termine di servizio

Qualora il servizio venisse terminato - per un qualsiasi motivo - deve essere garantito il recupero dei dati anche dopo tale evento.

Mantenimento dei log

I log devono essere mantenuti e devono rimanere accessibili anche dopo lunghi periodi, sia per il controllo da parte degli utenti che per eventuali investigazioni giudiziarie.

3 SERVIZIO DI AUTENTICAZIONE CENTRALIZZATO

Il modello del Software as a Service rappresenta lo strato più esterno di uno stack Cloud Computing completo. In un contesto sia Public che Private Cloud è sempre necessario che le caratteristiche fondamentali che contraddistinguono una infrastruttuta cloud non vengano meno e siano sempre rispettate: *on-demand self-service*, *broad network access*, *resource pooling*, *rapid elasticity* e *measured service*. In un'ottica del genere possiamo pensare che lo strato SaaS sia configurato come un ambiente di calcolo distribuito, ovvero, esistano più istanze dello stesso strato, accessibili all'utente finale. Ciò serve a garantire una certa continuità di servizio e distribuzione del carico di lavoro per i vari software messi a disposizione.

In questo caso, il servizio di autenticazione utente deve necessariamente essere centralizzato affinché le varie istanze SaaS possano sincronizzarsi su una base dati utente comune, ed evitare di replicare il medesimo meccanismo di autenticazione e dati utente per ciascuna installazione SaaS.

Per far fronte a questa problematica deve esistere un qualche servizio (*opportunamente ridondato e ad alta disponibilità*) che garantisca una gestione dei dati utente centralizzata a cui tutte le applicazioni SaaS possano far riferimento durante la fase di autenticazione.

Una comune soluzione è rappresentata nell'adottare un meccanismo di autenticazione che sia flessibile, sufficientemente robusto, collaudato e accessibile dalla gran parte delle applicazioni SaaS scelte. Già nel paragrafo 2.2.1, durante l'analisi dei meccanismi di Single-Sign-On di Google abbiamo fatto riferimento a LDAP.

In questo lavoro abbiamo analizzato e scelto di usare il protocollo LDAP - *Lightweight Directory Access Protocol* ed una sua implementazione opensource `openLDAP` per l'archiviazione, l'interrogazione e la gestione degli account utente.

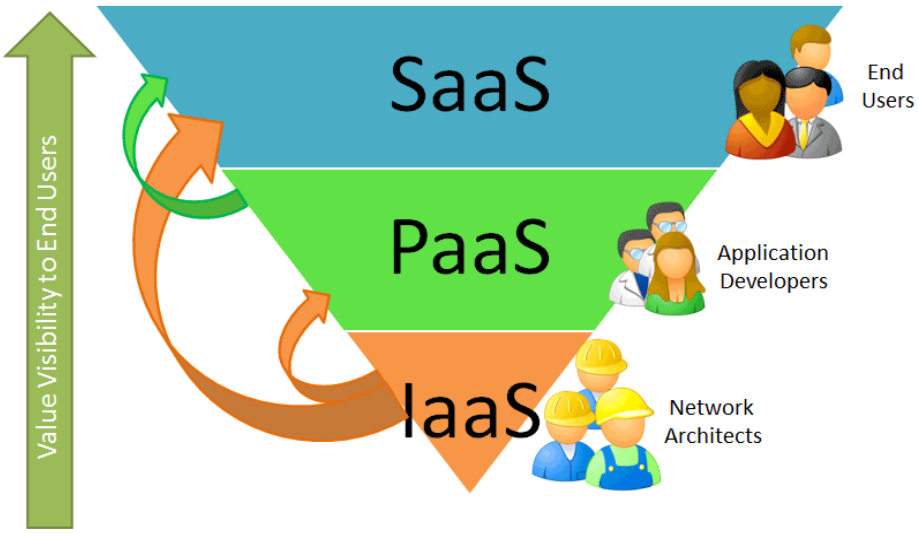

Fig. 3.1 - Lo stack completo del Cloud Computing.

3.1 Analisi di LDAP

L'acronimo LDAP sta per *Lightweight Directory Access Protocol* ed è un protocollo per l'interrogazione di una directory (database) mediante una connessione TCP/IP. Comunque, lo stesso termine è spesso usato per far riferimento sia al database che al protocollo.

Un database di LDAP memorizza le informazioni mediante una struttura dati ad albero, in cui ciascun nodo rappresenta una entry (o anche oggetto). Ciascun nodo eredita l'insieme ed i tipi di informazioni che può memorizzare, da una classe. Per ricavare l'informazione completa (chiamata Distinguished Name, o più brevemente DN) contenuta in LDAP è sufficiente ricostruire e concatenere tutti gli attribuiti presenti dal nodo foglia sino alla radice. Nel paragrafo 2.2.1 è visualizzato un esempio.

Le richieste di interrogazione a LDAP avvengono via TCP/IP. Un client instaura una sessione LDAP connettendosi ad un server che implementa tale protocollo, chiamato Directory System Agent (DSA) che da configurazione standard è in ascolto sulla porta TCP 389. Successivamente il client avvia una richiesta di operazione, e il server fornisce delle risposte. Con alcune eccezioni, il client non deve necessariamente attendere che il server fornisca una risposta per effettuare una seconda richiesta, d'altra parte, il server potrebbe inviare le risposte in vario ordine. Ciò che si istaura è una connessione asincrona.

Il client può richiedere le seguenti operazioni:

- StartTLS: Usa l'estensione TLS (Transport Layer Security) definita nel protocollo LDAPv3 per instaurare una connessione sicura
- Bind: Effettua un'autenticazione specifica la versione del protocollo da usare
- Search: Ricerca e/o richiede una determinata entry
- Compare: Controlla che il nome di una entry contiene un dato valore
- Aggiungere una nuova entry
- Cancellare una entry
- Modificare una entry
- Modificare un DN: Muovere o rinominare una entry
- Abandon: Interrompe una richiesta
- Extended Operation: Una operazione generica per definire altre operazioni
- Unbind: Chiude la connessione (da non intenere come operazione inversa a Bind)

Le directory (o database) acceduti tramite LDAP mantengono il modello di un protocollo già definito nel 1993, l'X.500:

- Una entry consiste in un insieme di attributi
- Ciascun attributo ha un nome (un tipo di attributo oppure la descrizione dell'attributo) e uno o più valori. Gli attributi sono definiti per mezzo di uno schema
- Ciascuna entry può essere identificata per mezzo di un identificatore univoco: il Distinguished Name (DN), che consiste del Relative Distinguished Name (RDN) contenuto nel nodo foglia completato con tutti i dati contenuti nei nodi padre sino a raggiungere il nodo radice.

Un esempio di una entry è così come segue:

```
dn: cn=John Doe,dc=example,dc=com
cn: John Doe
givenName: John
sn: Doe
telephoneNumber: +1 888 555 6789
telephoneNumber: +1 888 555 1232
mail: john@example.com
manager: cn=Barbara Doe,dc=example,dc=com
objectClass: inetOrgPerson
objectClass: organizationalPerson
objectClass: person
objectClass: top
```

Questo formato è conosciuto come LDIF (LDAP Data Interchange Format).

"dn" è il Distinguished Name della entry; non rappresenta nè un attributo e neanche parte dell'entry. "cn=John Doe" è la Relative Distinguished Name (RDN), e "dc=example,dc=com" è il DN entry padre dove "dn" sta per "Domain Component". Tutte le altre righe mostrano una serie di attributi nella entry. I nomi degli attributi tipicamente sono stringhe mnemoniche, come ad esempio "cn" per common name, "dc" per domain component, "mail" per e-mail address e "sn" per surname.

Un server LDAP mantiene un sotto-albero a partire da una specifica entry, ad esempio "dc=example,dc=com" e tutti i suoi nodi figli.

3.1.1 Organizzazione della directory

Nel paragrafo 2.2.1 abbiamo detto che LDAP viene utilizzato per mantenere e ben organizzare - attraverso una disposizione gerarchica ad albero - le informazioni di un ambiente di calcolo distribuito, laddove le risorse e le rispettive tipologie possono essere molteplici e risulta quindi necessario affidarsi ad uno strumento che possa meglio catalogare e rendere efficiente la ricerca di queste informazioni.

In un contesto relativo l'autenticazione utente e più in generale lo studio di fattibilità di un sistema SaaS con solo Software Libero, non sarà necessario scendere così tanto nel dettaglio e definire dunque un'organizzazione troppo articolata delle informazioni.

Per il nostro scopo forniremo un'organizzazione della directory abbastanza orizzontale e uniforme nel tipo di dati da essa gestita.

L'albero che manterrà tutte le informazioni sugli utenti disporrà di un nodo radice "dc=example,dc=com" e tanti nodi figli a partire da essa per quanti sono gli utenti da gestire.

Nel nostro caso, come vedremo durante la fase di configurazione di LDAP, ciascuna entry dovrà rispettare uno schema specifico per memorizzare particolari informazioni sull'utente.

3.1.2 openLDAP: installazione, configurazione e gestione

OpenLDAP è una raccolta di software open source che implementano un server LDAP ed uno strumento per la replica delle directory, rispettivamente `slapd` e `slurpd`.

In questo lavoro ci preoccuperemo di fornire una installazione base di `slapd` e di fornire una prima configurazione, utile per la realizzazione di un sistema di autenticazione centralizzato funzionante.

Installiamo il pacchetto `slapd`:

```
# aptitude install slapd ldap-utils
```

Indichiamo una password di amministratore e procediamo con la configurazione del server e la successiva predisposizione della directory:

```
# dpkg-reconfigure -plow slapd
```

Nel nostro esempio indicheremo come DN il valore **example.org** e forzeremo LDAP ad utilizzare solo il protocollo LDAPv3.

La struttura della directory che vogliamo realizzare sarà siffatta:

```
|- dc=example,dc=org
|-- user1
|-- user2
|-- ...
|-- userN
```

In cui ciascun nodo `user` identifica un utente del sistema e tutte le rispettive informazioni ad esso associato. Lo schema da applicare a ciascuna di questy entry è specifico per il mantenimento delle informazioni relative alla password, alla casella di posta elettronica, al percorso assoluto della directory dove vengono salvate tutte le email, e altre informazioni.

Lo schema in questione è presente nella directory `/usr/share/doc/courier-authlib-ldap/` e si chiama `authldap.schema`. Sarà necessario copiarlo in `/etc/ldap/schema/` e istruire LDAP di utilizzare anche questo schema.

Aggiungere questa linea:

```
include /etc/ldap/schema/authldap.schema
```

nel file `/etc/ldap/slapd.conf`.

Riavviato il servizio `slapd`, non resta che aggiungere gli account inserendo tutto le informazioni del caso all'interno di un file `ldap_user.ldif` e successivamente avviare la modifica sul server LDAP.

File: ldap_user.ldif

```
dn: uid=user1, dc=example, dc=org
mailbox: user1/
sn: user1
userPassword: {SSHA}15Uc91sxWmx5pF3aThfCt2DDYlI7f2oJ
```

```
uidNumber: 1001
gidNumber: 1001
mail: user1@example.org
objectClass: top
objectClass: CourierMailAccount
objectClass: person
uid: user1
cn: user1
homeDirectory: /home/vmail

dn: uid=user2, dc=example, dc=org
mailbox: user2/
sn: user2
userPassword: {SSHA}15Uc91sxWmx5pF3aThfCt2DDYlI7f2oJ
uidNumber: 1001
gidNumber: 1001
mail: user2@example.org
objectClass: top
objectClass: CourierMailAccount
objectClass: person
uid: user2
cn: user2
homeDirectory: /home/vmail
```

Aggiunta delle informazioni in LDAP:

```
# ldapadd -c -x -D "cn=admin,dc=example,dc=org" -W -f
ldap_user.ldif
```

3.1.3 phpLDAPadmin

Per migliorare la gestione delle risorse contenute in LDAP, uno strumento che può essere molto utile all'interno di una completa soluzione SaaS è rappresentato da un gestore grafico della directory LDAP, accessibile anche da un browser web. In questo caso, esiste un progetto open source - phpLDAPadmin [27] - che fornisce, appunto, questo genere di funzionalità.

phpLDAPadmin può mantenere attive diverse sessioni verso più server LDAP. Supporta pienamente OpenLDAP su tutte le richieste di operazione. Per altri server LDAP come Microsoft Active Directory e Fedora Directory Server sono supportate solo le richieste di operazioni in sola lettura.

Tra le varie funzionalità, phpLDAPadmin permette di creare, a partire da una serie di template pre-impostati una serie di entry e di completare diverse operazioni di manutenzione ordinaria.

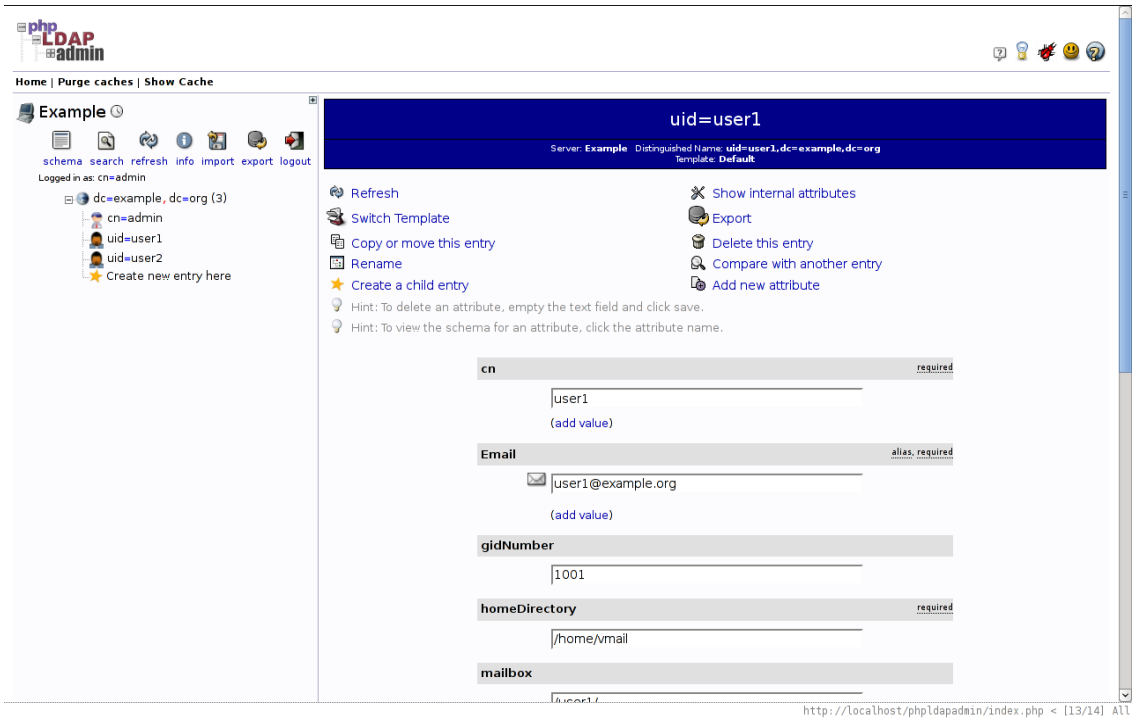

Fig. 3.2 - Schermata di phpLDAPadmin che visualizza le informazioni sull'utente user1.

3.2 OpenID: Integrazione con LDAP

Allo stesso modo di come il set di soluzioni SaaS di Google realizza il Single-sign-on, e consente ad applicazioni sviluppate da terzi di sfruttare il proprio servizio di autenticazione per validare le identità degli utenti, anche una soluzione Private Cloud SaaS dovrebbe garantire simili funzionalità. Uno scenario che potrebbe verificarsi potrebbe comprendere la necessità di voler integrare all'interno dell'infrastruttura Private Cloud, un'applicazione sviluppata da terzi e volerla collegare allo stesso sistema di autenticazione.

Oppure, ancora, utilizzare la propria infrastruttura Cloud come un Identity Provider per certificare le nostre utente su applicazioni esterne al Private Cloud.

Per la realizzazione di un sistema del genere è necessario possedere una implementazione di un protocollo di Single-sign-on come quello discusso nel paragrafo 2.1.1, ovvero di OpenID.

Dal momento che la nostra configurazione SaaS utilizza LDAP per la gestione della base dati utente, è necessario individuare un software open source che sia in grado fornire una implementazione di OpenID basata su LDAP.

Il progetto specifico esiste, è stato scritto in PHP, e si chiama OpenID-LDAP [28].

Fig. 3.3 - Logo di OpenID-LDAP

Per ottenere una configurazione funzionante di OpenID-LDAP è necessario abilitare diversi moduli in Apache2, quali: mod_rewrite, mod_ssl, mod_proxy e mod_proxy_http, e inoltre impostare correttamente alcuni settaggi di configurazione dello stesso servizio, ad esempio sarà necessario indicare il server LDAP su cui effettuare le varie richieste di operazione, i dati di accesso di amministratore a LDAP e gli attributi delle entry su cui effettuare la ricerca.

Nel nostro caso, l'URL necessario ad identificare uno dei due utenti precedentemente creati, avrà una forma del genere: `http://example.org/openid-ldap/user1`.

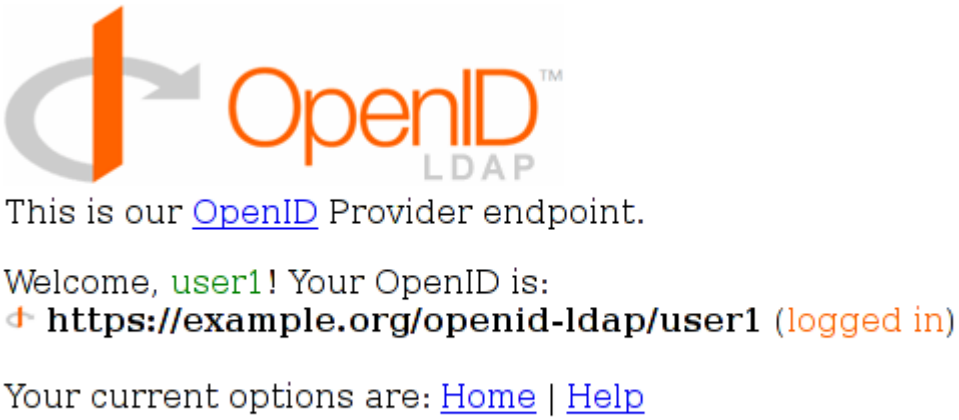

Fig. 3.4 - Schermata di OpenID-LDAP che conferma l'autenticazione dell'utente user1.

4 INSTALLAZIONE E CONFIGURAZIONE PIATTAFORMA SAAS

Descrivere nel dettaglio le fasi di installazione e configurazione di tutte le alternative Software Libero per la costruzione di un sistema Saas, non può trovare spazio in un elaborato del genere, probabilmente sarebbe più idoneo rivolgersi a testi completi su ciascun argomento; per cui, in questa parte del lavoro ci soffermeremo nel descrivere un po' più approfonditamente ciascun singolo software, fornendo una descrizione introduttiva, i protocolli utilizzati, i formati dati gestiti, qualche riferimento all'architettura software, alle configurazioni adottate, alle tecnologie e linguaggi di programmazione adoperati, alle licenze, al supporto verso LDAP e a sistemi di Single-sign-on, e in particolar modo ci soffermeremo sul livello di integrazione che siamo riusciti a raggiungere.

4.1 Analisi, installazione e configurazione del software per la messaggistica

4.1.1 Postfix

Postfix è un servizio di posta elettronica SMTP (comunemente chiamato MTA, Mail Transfer Agent). È rilasciato sotto licenza IBM Public License e ciò lo rende compatibile secondo le definizioni di Software Libero. Largamente supportato dalla comunità di sviluppatori e compatibile con vari sistemi operativi Unix-like dotati di un compilatore ANSI-C, librerie di sviluppo compatibili POSIX e librerie per il supporto di socket BSD, Postfix nasce inizialmente per rimpiazzare l'alternativa Sendmail (programma storico in ambienti Unix-like per l'invio di email), al fine di fornire maggiori garanzie di funzionamento, di sicurezza e di flessibilità.

È noto per tollerare una grande quantità di messaggi di posta elettronica ed è ormai presente nella gran parte delle distribuzioni GNU/Linux e scelto come MTA di default su molte di queste.

Per il nostro scopo, Postfix dovrà essere opportunamente configurato al fine di supportare l'autenticazione utente tramite LDAP, gestione di indirizzi di posta virtuali ed ottenere le informazioni necessarie per ciascun utente.

Dal momento che l'argomento di questa tesi non comprende gli aspetti più specifici riguardo la configurazione di questo servizio, eviteremo di soffermarci su aspetti relativi la sicurezza, filtri anti-spam, filtri anti-virus, processi di mail delivery sofisticati e così via. Ci limiteremo semplicemente a fornire le configurazioni di Postfix, affinché quest'ultimo possa autenticare gli utenti tramite LDAP.

Per installare Postfix sul nostro sistema:

```
# aptitude install postfix-ldap
```

Nel file di configurazione principale del servizio `/etc/postfix/main.cf` è necessario inserire:

```
virtual_mailbox_domains = example.org
virtual_mailbox_base = /home/vmail
virtual_mailbox_maps = ldap:/etc/postfix/ldap-virtual-
```

mbox.cf

```
virtual_minimum_uid = 134
virtual_uid_maps = static:134
virtual_gid_maps = static:65534
virtual_alias_maps = hash:/etc/postfix/virtual
```

Rispettivamente, le seguenti righe di configurazione identificano: il nome del dominio a cui tutti gli indirizzi di posta elettronica fanno riferimento, la directory base in cui sono presenti tutte le Maildir (è il tipo di formato per l'archiviazione di email ormai più diffuso e versatile utilizzato dalla gran parte dei server IMAP) di ciascun utente, la mappatura tra indirizzi costruiti durante le richieste SMTP e account utente memorizzati in LDAP, lo UserID e il GroupID proprietario della directory base ed eventuali alias di indirizzi di posta elettronica.

Il file `/etc/postfix/ldap-virtual-mbox.cf` , invece, contiene le informazioni di configurazione necessarie per l'interrogazione della directory LDAP:

```
version = 3
search_base = dc=example,dc=org
result_attribute = mailbox
query_filter = (mail=%s)
timeout = 10
host = localhost
server_port = 389
```

Ad operazione ultimato sarà necessario riavviare il servizio per rendere le modifiche permanenti:

```
/etc/init.d/postfix restart
```

Quadro riepilogativo	
Linguaggio di programmazione:	C, C++
Supporto Database	MySQL, PostgreSQL, Berkeley DB, LDAP, Memcached, CDB, DBM, SQLite
Protocolli utilizzati	ESMTP
Architettura software	Multi-tenant
Riferimento internet	http://www.postfix.org
Licenza	IBM Public License

4.1.2 Courier Mail Server

Courier Mail Server è un Mail Transfer Agent (MTA) che fornisce una collezione di software individuali che implementano vari tipi di protocolli e servizi, tra questi: ESMTP, IMAP, POP3, SMAP, webmail e mailing list. Il suo sviluppo originario era particolarmente orientato verso il supporto IMAP piuttosto che a tutti gli altri servizi e protocolli, per questo motivo, ancora oggi è spesso scelto come soluzione principale per implementare un server IMAP.

Un server IMAP permette la gestione e l'organizzazione delle email con più directory direttamente sul server stesso, senza che l'utente finale - al contrario del protocollo POP3 - sia costretto a scaricare l'intera posta elettronica sul proprio computer o dispositivo mobile. Attraverso l'utilizzo di thin client disponibili per dispositivi mobile, ogni utente potrà accedere alla propria casella di posta.

In questo lavoro utilizzeremo solo il componente IMAP di Courier Mail Server, che sarà interfacciato alla base dati utente LDAP per autenticare ed estrapolare le informazioni relative alla localizzazione delle directory email presenti sul server.

L'installazione di Courier sul nostro sistema si completa con il seguente comando:

```
# aptitude install courier-imap courier-ldap gamin
```

I due file di configurazione che necessitano una modifica per abilitare il protocollo di autenticazione LDAP e per fornire le indicazioni necessarie all'interrogazione della directory sono: `/etc/courier/authdaemonrc` e `/etc/courier/authldaprc`.

Di seguito si motrano le configurazioni nel dettaglio:

/etc/courier/authdaemonrc

```
authmodulelist="authldap"
```

e /etc/courier/authldaprc

```
LDAP_URI          ldap://example.org
LDAP_PROTOCOL_VERSION  3
LDAP_BASEDN       dc=example, dc=org
LDAP_AUTHBIND     1
LDAP_GLOB_UID     vmail
LDAP_GLOB_GID     nogroup
LDAP_HOMEDIR homeDirectory
LDAP_MAILDIR      mailbox
```

Quadro riepilogativo	
Linguaggio di programmazione:	C, C++, Perl
Supporto Database	MySQL, PostgreSQL, Berkeley DB, LDAP
Protocolli utilizzati	IMAP, POP3, ESMTP, HTTP, SMAP
Architettura software	Multi-tenant

Riferimento internet	http://www.courier-mta.org
Licenza	GNU GPL

4.1.3 Horde Groupware Suite

È una applicazione web di groupware open source, rilasciata con licenza GPL. Tutti i componenti che la compongono fanno parte dell'omonimo framework web Horde. È un framework per lo sviluppo di applicazioni web, interamente scritto in PHP.

L'edizione Groupware Suite è una collezione di applicazioni web utili alla gestione di un groupware, e comprende: posta elettronica, calendario, rubrica dei contatti, note, appuntamenti e un file manager minimale.

Il componente senz'altro più importante dell'intera collezione è rappresentano da IMP (Internet Messaging Program), è una interfaccia avanzata, ricca di molte funzionalità, per la gestione della posta elettronica.

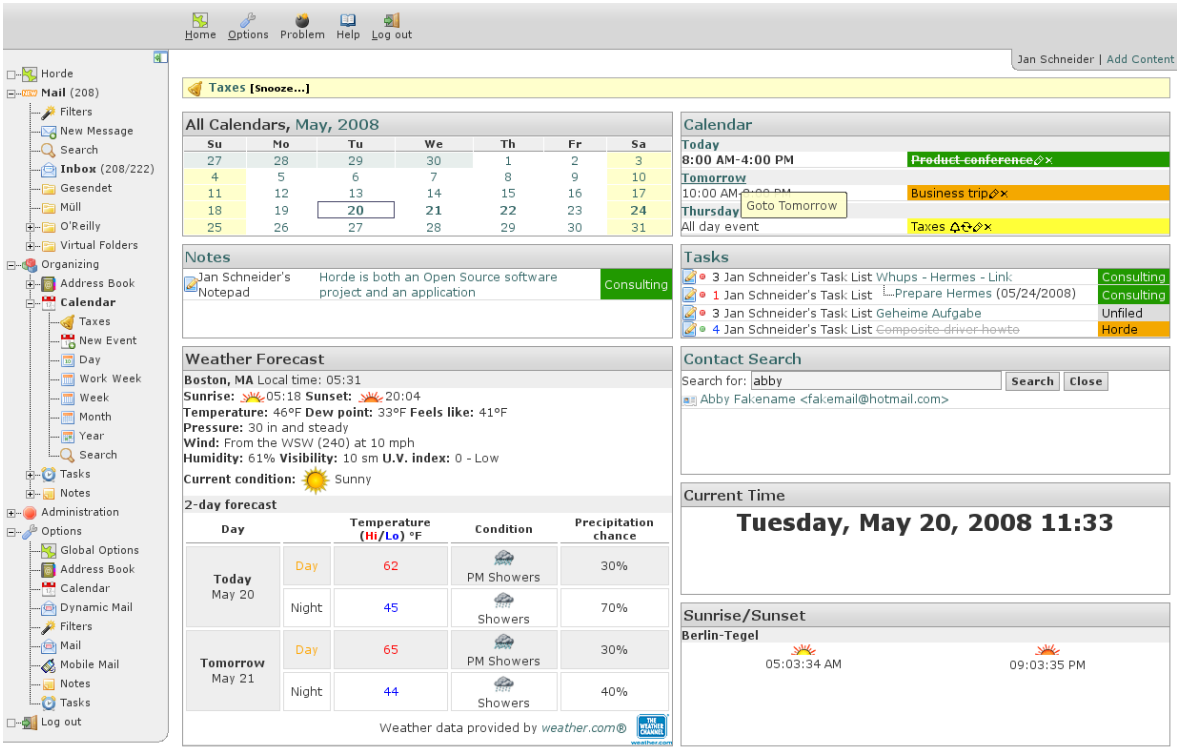

Fig. 4.1 - Schermata principale di Horde Groupware suite

L'ultima versione disponibile di questo Groupware, adatta l'interfaccia grafica anche per dispositivi mobile e permette la consultazione del servizio anche da smartphone e tablets.

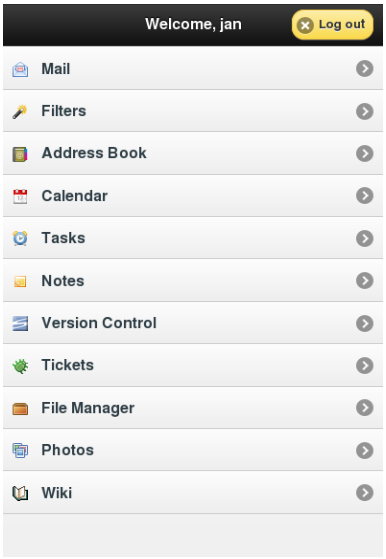

Fig. 4.2 - Interfaccia Horde visualizzata da dispositivi mobili.

L'installazione e la configurazione di Horde è basata su PEAR (PHP Extension and Application Repository), un framework e un sistema per la distribuzione di codice scritto in PHP. La comunità di Horde ha suddiviso l'intero framework in centinaia di componenti, caricandoli in ultimo su questo repository al fine di renderlo il più versatile e utilizzato possibile. Così facendo, ciascun componente può essere recuperato dal repository e usato singolarmente per altre applicazioni, non necessariamente integrate in Horde.

Per poter installare Horde Groupware Suite sul sistema è necessario recuperare il software necessario per la gestione dei repository PEAR:

```
# aptitude install php-pear
```

Si prosegue con la configurazione del repository e successiva installazione dell'intera applicazione:

```
# pear channel-discover pear.horde.org
# pear install horde/horde_role
# pear run-scripts horde/horde_role
# pear install -a -B horde/webmail
# webmail-install
```

Per completare l'installazione e rendere l'applicazione web funzionante e interfacciata con i servizi Postfix e Courier Mail Server, e quindi collegato indirettamente ad LDAP, è necessario modificare il file di configurazione del componente IMP (Internet Messaging Protocoll) e indicare i dati relativi al server IMAP attraverso cui eseguire l'autenticazione:

/var/www/horde/imp/config/backends.local.php

```
<?php
$servers['imap'] = array(
    'disabled' => false,
    'name' => 'IMAP Server',
    'hostspec' => 'example.org',
    'hordeauth' => false,
```

```
    'protocol' => 'imap',
    'port' => 143,
    'secure' => false,
    'maildomain' => '',
    'cache' => false,
);
?>
```

Horde Groupware Suite si poggia (opzionalmente) su Database esterni, quali MySQL, PostgreSQL o SQLite per la memorizzazione delle impostazioni e delle preferenze definite dagli utenti, come ad esempio: filtri per la posta elettronica, calendari, contatti, sincronizzazione con dispositivi mobili, preferenze per la consultazione delle email e così via.

Quadro riepilogativo	
Linguaggio di programmazione:	PHP 5.2.x o superiore
Supporto Database	MySQL, PostgreSQL, SQLite, LDAP
Protocolli utilizzati	HTTP, HTTPS, IMAP, POP3
Architettura software	Multi-tenant
Riferimento internet	http://www.horde.org
Licenza	GNU Lesser GPL

4.1.4 Groupserver

È software open source, rilasciato con licenza GPL, sviluppato principalemte in ZopeFive [29] e in Python [30] , implementa una interfaccia web per la gestione di mailing list elettroniche, di grandi dimensioni. Fornisce un'esperienza utente simile alle tradizionali mailing list ma in più supporta la lettura, la ricerca e l'invio di messaggi direttamente via web. Gli utenti registrati in GroupServer possono gestire il proprio profilo e preferenze attraverso un pannello di amministrazione messo a disposizione nella stessa interfaccia web. Tra le altre caratteristiche, vi è la possibilità di personalizzare l'interfaccia web in modo evidente e senza troppa difficoltà.

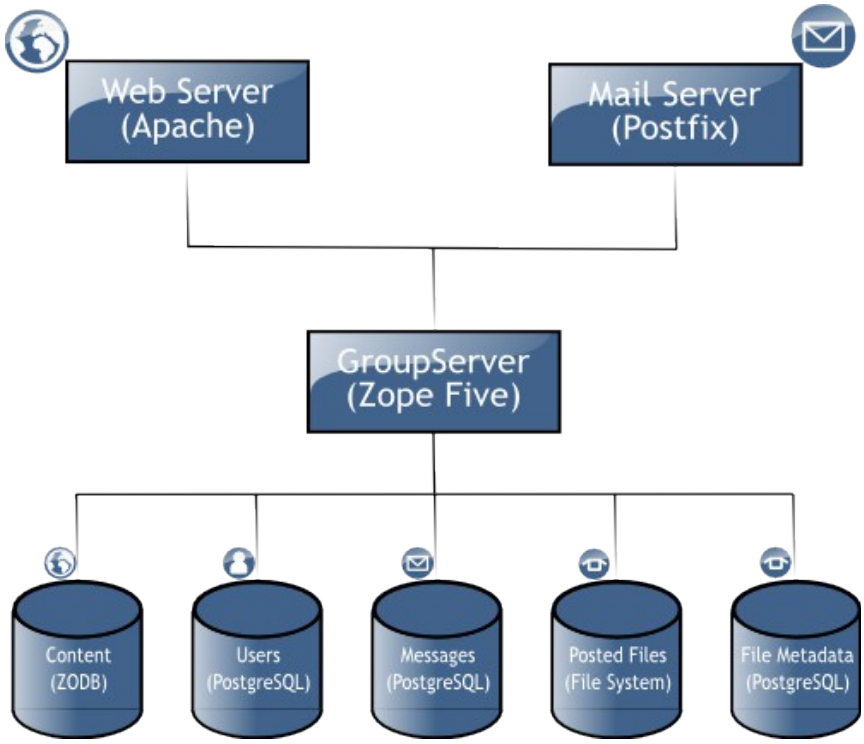

Fig. 4.3 - Architettura tecnica di GroupServer.

GroupServer combina web server e mail server, collegaondoli a diversi meccanismi per la memorizzazione dei dati. In questo modo, tutti i messaggi gestiti possono essere facilmente recuperabili e integrabili in piattaforme esterne.

L'architettura si base quindi su quattro livelli principali:

1. Apache è utilizzato per mediare le connessioni tra il browser web ed il software stesso.
2. Postfix è il Mail Transfer Agent che si occupa di inviare e ricevere i messaggi in formato posta elettronica.
3. GroupServer rappresenta la parte principale, gestisce, organizza e visualizza le informazioni delle mailing list
4. Dati:
    - Zope Object Database: Tutte le pagine statiche gestite da GroupServer sono archiviate via ZODB, che è un componente del framework Zope.

- PostgreSQL: I messaggi pubblicati, le informazioni relative all'utente e tutte le altre informazioni relative ai gruppi sono archiviate e gestite via PostgreSQL. La connessione tra Zope e PostgreSQL è realizzata via SQLAlchemy, è una libreria Python per il supporto ORM (Object Relational Mapper).
- File System: In ultimo, tutti i file caricati nei gruppi, in forma di allegato, vengono salvati attraverso i meccanismi standard forniti dal sistema operativo.

Dal momento che ciascuno livello può essere configurato, singolarmente, su macchine diverse, GroupServer si presta in modo particolare ad una soluzione SaaS in fatto di scalabilità.

Per quanto concerne l'installazione e la configurazione di questo software [mettere riferimento alla guida di installazione (http://groupserver.org/downloads/install)], non esiste un package distribuito con le distribuzioni GNU/Linux, quindi è necessario installare e configurare manualmente ogni singolo componente:

```
# apt-get install python2.6 python2.6-dev python-virtualenv
postgresql-server-dev-8.4 libpq-dev postfix-dev libzzip-dev
```

Installato l'occorrente, seguirà la configurazione di PostgreSQL, di Zope ed infine del Mail Server Postfix.

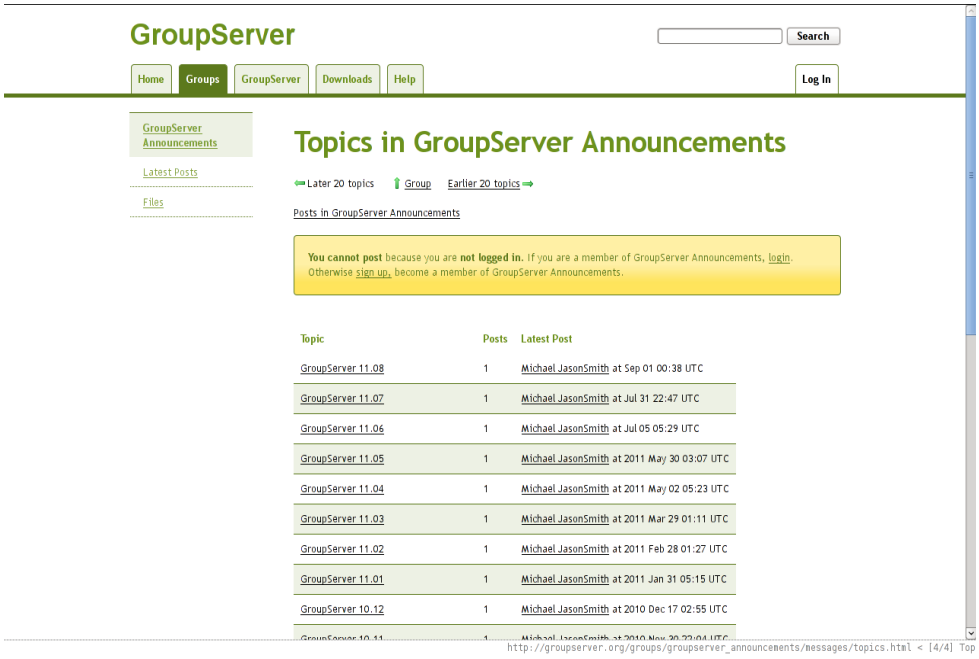

Fig. 4.4 - Schermata della lista dei messaggi in un gruppo creato su GroupServer

La versione attuale di GroupServer, la 11.08, non ha il supporto con LDAP, per questo motivo l'autenticazione utente non può essere centralizzata. Ogni utente facente parte dell'infrastruttura Private Cloud SaaS dovrà registrarsi nuovamente le proprie credenziali di accesso su GroupServer.

Quadro riepilogativo

Linguaggio di programmazione:	Python 2.6
Dipendenze	Postfix, Apache
Supporto Database	PostgreSQL
Protocolli utilizzati	HTTP, HTTPS
Architettura software	Multi-tenant
Riferimento internet	http://groupserver.org
Licenza	GNU GPL

4.1.5 ejabberd

ejabberd è un server che implementa il protocollo XMPP. Il software è scritto completamente in C ed è rilasciato sotto licenza GPL.

XMPP sta per Extensible Messaging and Presence Protocol, ed è un protocollo di comunicazione a standard aperto per lo scambio di messaggi tra sistemi distribuiti nella rete. Il formato dei messaggi scambiati tra i server XMPP è in XML.

È il protocollo più diffuso e utilizzato dalla maggior parte di servizi di Instant Messaging. Sono servizi che permettono a più utenti di poter intrattenere conversazioni testuali e non, in tempo reale.

Al contrario di altri protocolli - proprietari - di Messaggeria Instantanea, XMPP, basandosi su uno standard aperto consente lo sviluppo di applicazioni interoperabili tra diverse organizzazioni. Esistono infatti diverse implementazioni server e client che possono essere scambiate a piacere, mantenendo comunque la compatibilità di comunicazione tra le varie soluzioni.

Non esiste difatti, in una rete che presenta diversi server XMPP distribuiti, un server centrale autoritativo. I messaggi all'interno della rete vengono instradati a partire dal nome di dominio a cui appartiene un certo utente. Ad esempio, l'indirizzo `user1@example.org` rappresenterà l'identificativo XMPP dell'utente `user1` sul server `example.org`. Ciò vuol dire che è possibile comunicare, in tempo reale, con un altro utente, anche se quest'ultimo non possiede un identificativo sullo stesso server XMPP utilizzato da un altro utente.

Questo protocollo è stato formalizzato nel 2002 dalla Internet Engineering Task Force (IETF) come "IETF instant messaging and presence technology".

Lo scopo di integrare, all'interno di una infrastruttura Private Cloud SaaS, un sistema di Instant Messaging è quello di permettere ad utenti di una stessa di organizzazione di poter scambiare messaggi in tempo reale, da qualsiasi thin client e dispositivo che supporti il protocollo XMPP.

L'installazione e la configurazione del server `ejabberd` su un sistema operativo GNU/Linux è relativamente semplice:

```
# aptitude install ejabberd
```

E successivamente non resta che configurare opportunamente il server, abilitando l'autenticazione utente basata su LDAP (`/etc/ejabberd/ejabberd.cfg`):

```
%% Authentication using LDAP
{auth_method, ldap}.
%% List of LDAP servers:
{ldap_servers, ["example.org"]}.
%% Encryption of connection to LDAP servers (LDAPS):
{ldap_encrypt, none}.
%% Port connect to LDAP server:
{ldap_port, 389}.
%% Search base of LDAP directory:
{ldap_base, "dc=example,dc=org"}.
%% LDAP attribute that holds user ID:
{ldap_uids, [{"mail", "%u@example.org"}]}.
{ldap_filter, "(objectClass=CourierMailAccount)"}.
```

Fig. 4.5 - Client IM "Pidgin" connesso al server ejabberd

Quadro riepilogativo	
Linguaggio di programmazione:	Erlang/OTP
Supporto Database	MySQL, PostgreSQL, LDAP
Protocolli utilizzati	XMPP
Architettura software	Multi-tenant

Riferimento internet	http://www.ejabberd.im
Licenza	GNU GPL v.2

4.2 Analisi, installazione e configurazione del software per la collaborazione

4.2.1 Alfresco Community Edition

È un'applicazione sviluppata completamente in Java, per la gestione dei contenuti aziendali, il suo design è orientato alle utenze che richiedono un alto grado di modularità e prestazioni scalabili. Alfresco include un repository per l'accesso ai contenuti in esso archiviati, una interfaccia grafica accessibile via web per la gestione e l'utilizzo dei vari contenuti, una interfaccia CIFS (Common Interface File System) per fornire un livello di integrazione a livello di file system con tutti i sistemi operativi che supportano tale protocollo, un sistema per la gestione di contenuti web capace di virtualizzare applicazioni web esterne oppure siti statici attraverso Apache Tomcat, un motore di indicizzazione dei contenuti basato su Apache Lucene.

L'utilizzo di Alfresco è particolarmente indicato per tutte quelle organizzazioni che necessitano di un software scalabile per la gestione di documenti, componenti web, records, immagini e lo sviluppo collaborativo dei contenuti.

Tra le principali funzionalità, Alfresco presenta:

- Gestione documentale
- Gestione contenuti web
- Versioning di tutti i contenuti in archivio
- Gestione delle immagini
- Accesso dell'archivio via CIFS/SMB, FTP, WebDAV, NFS e CMIS
- Flusso delle attività
- Motore di ricerca Lucene
- Autenticazione NTLM, LDAP, Kerberos, CAS
- Supporto a più DBMS

Sotto l'aspetto tecnico, Alfresco sfrutta le maggiori tecnologie Open Source legate a Java, come Spring, Hibernate, Lucene, Web Services, Java Server Faces e altri progetti nati dalla comunità Open Source.

In questo nostro lavoro ci siamo limitati a fornire una installazione e configurazione funzionante di questa soluzione, agganciando il sistema di autenticazione interno con la base dati utenti centralizzata gestita dal server LDAP. In questo caso, Alfresco è in grado di importare le utenze contenute in LDAP, definire per ognuna l'elenco dei privilegi e completare le informazioni che andranno a caratterizzare il profilo utente all'interno dell'applicazione. D'altra parte, è in grado di sincronizzare in modo automatico, le utenze presenti in LDAP, già nel momento in cui avviene un'autenticazione.

Le funzionalità a cui abbiamo prestato maggiore attenzione hanno riguardato la gestione documentale e l'accesso all'archivio via CIFS/SMB.

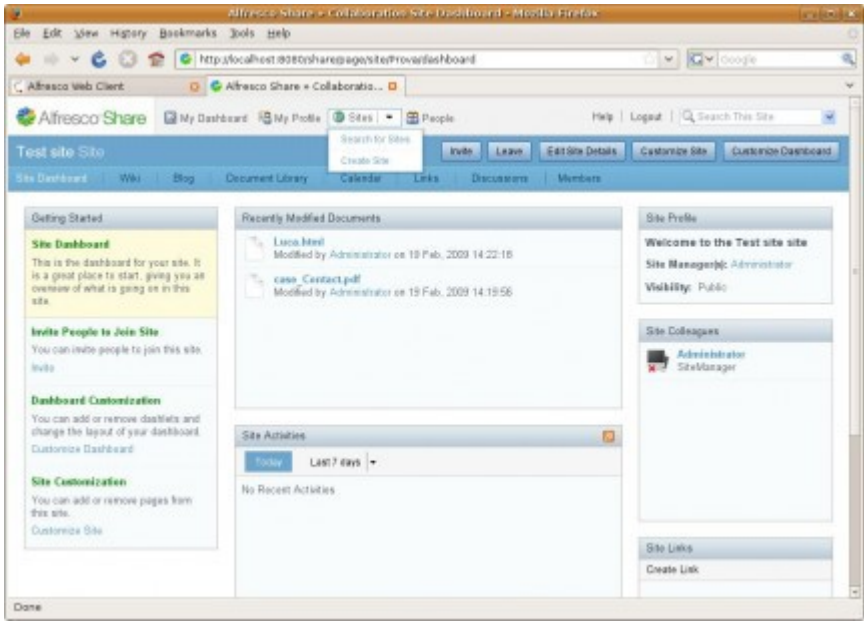

Fig. 4.6 - Schermata principale di Alfresco

Una piattaforma che include così tante funzionalità e che poggia su tecnologie quali Java e relativi framework, richiede ingenti risorse hardware e d'altra parte, l'attuale versione pronta per il download, direttamente dal sito della comunità, è disponibile solo per sistemi operativi a 64bit e l'hardware minimo richiesto è una CPU con una frequenza di calcolo superiore ai 2GHz e 4GB di RAM.

Quadro riepilogativo	
Linguaggio di programmazione:	Java
Tecnologie e strumenti usati:	Apache Tomcat, Spring, Hibernate, Lucene, Web Services, Java Server Faces
Supporto Database	MySQL, PostgreSQL, Oracle, IBM DB2, Microsoft SQL Server
Protocolli utilizzati	HTTP, HTTPS, CIFS
Architettura software	Multi-tenant
Riferimento internet	http://www.alfresco.com/community/
Licenza	GNU GPL v.2

4.2.2 Etherpad Lite

Etherpad lite è un editor collaborativo e in real time di documenti di testo, scritto completamente in Javascript, il progetto è stato derivato e quasi totalmente riscritto, dal più noto Etherpad. Sfrutta la libreria, oramai ben collaudata e matura per garantire l'editing veramente in tempo reale. Basandoti su node.js è molto più leggero stabile del suo predecessore.

È ottimizzato per essere facilmente integrabile in molte piattaforme, difatto, piuttosto che rappresentare uno servizio a sé stante, Etherpad lite è da considerarsi come uno strumento da poter essere integrato in diverse applicazioni.

Il livello di integrazione fornito da questo strumento è rappresentato da una specifica di una API basata su HTTP che permetterebbe alle applicazioni di gestire documenti, utenti e gruppi. Attualmente, di questa API esistono diverse implementazioni per diversi linguaggi di programmazioni, quali: PHP, .Net, Node.js, Ruby, Python e anche per jQuery.

Per ottenere una installazione funzionante di Etherpad lite è necessario procurarsi `nodejs`, compilarlo e installarlo nel sistema:

```
$ wget http://nodejs.org/dist/v0.6.16/node-v0.6.16.tar.gz
$ tar zxf node*
$ cd node-v0.6.16
$ ./configure && make
$ sudo make install
```

Infine si procede con l'installazione di Etherpad Lite:

```
$ cd ..
$ git clone 'git://github.com/Pita/etherpad-lite.git'
$ cd etherpad-lite
$ ./bin/installDeps.sh
```

Configurazione di alcune impostazioni via `settings.json`

Avvio di Etherpad Lite:

```
$ ./bin/run.sh
```

Il servizio resterà in ascolto sulla porta 9001.

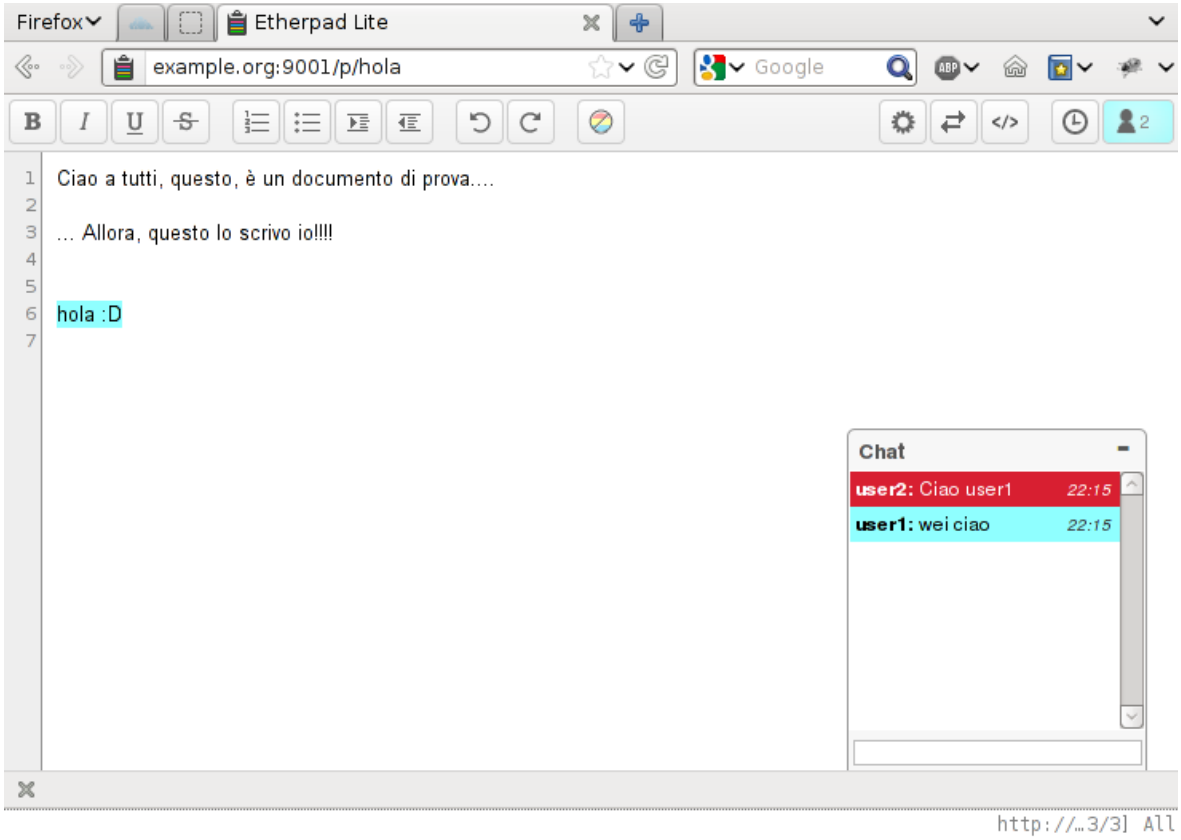

Fig. 4.7 - Schermata di Etherpad con due sessioni utente contemporanee sullo stesso documento.

Quadro riepilogativo	
Linguaggio di programmazione:	Javascript
Protocolli utilizzati	HTTP, HTTPS
Riferimento internet	https://github.com/Pita/etherpad-lite
Licenza	Apache License v.2.0

4.2.3 DAViCAL

DAViCal è un server per la condivsione e gestione di calendari e contatti. Implementa il protocollo CalDAV e CardDAV [31] per l'archivizione delle risorse calendario (nel formato iCalendar) e contatti, su un server remoto accessibile da qualsiasi software che supporti questo tipo di formato e protocollo.

Il numero di client (sia per PC che per dispositivi mobili) per la gestione e visualizzazione di questo genere di servizi, è in costante aumento. Tra i software disponibili troviamo Mozilla Calendar, Evolution, Mulberry, Chandler, e altri prodotti a codice chiuso come Apple iCal, iPhone e Google Calendar

Le caratteristiche chiave del server DAViCal sono:

- È un Software Libero rilasciato con licenza GPL
- Utilizza un database SQL per la memorizzazione dei dati
- Supporta una retro-compatibilità per l'accesso tramite WebDAV in modalità sola-lettura
- Può essere utilizzato con la maggior parte del client software che supportano il protocollo CalDAV.

I requisiti necessari per poter installare DAViCal sono:

- PostgreSQL
- Un webserver con supporto a PHP 5

Il supporto a LDAP di DAViCal è abbastanza maturo e implementa inoltre delle procedure di sincronizzazione tra gli utenti disponibili in LDAP e gli utenti ed i calendari ad essi associati, memorizzati in PostgreSQL. Di seguito vengono illustrate le politiche di sincronizzazione tra LDAP e DAViCAL:

- Quando avviene l'autenticazione di un utente LDAP e non esiste ancora il corrispondente utente in DAViCal, allora ne viene creato uno nuovo in PostgreSQL. Se invece esistono entrambi gli utenti, DAViCal aggiorna le informazioni del rispettivo utente in base ad un confronto di timestamp.
- Possibilità di sincronizzare una Directory LDAP verso DAViCAL. Questa operazione viene così descritta:
    1. Si controlla la validità dell'utente nella directory LDAP
    2. Si controlla il rispettivo utente in DAViCAL

       e allora
    3. Se un utente è presente in DAViCal ma non in LDAP, allora viene marcato come inattivo in DAViCal
    4. Se un utente è presente in LDAP ma non in DAViCal, viene automaticamente creato in DAViCal
    5. Se un utente è presente sia in LDAP ch ein DAViCal, vengono aggiornate le informazioni in DAViCal

Una volta che DAViCal è stato installato, non resta che abilitare il supporto LDAP e

integrarlo con il sistema di autenticazione centralizzato della nostra infrastruttura SaaS.

Nello specifico il file di configurazione di DAViCal (`/etc/davical/config.php`) sarà modificato come segue:

```
<?php
$c->pg_connect[] = "dbname=davical user=davical_app password=app";
$c->system_name = "DAViCal CalDAV Server";
$c->hide_TODO = false;
$c->admin_email ='user1@example.org';
$c->enable_row_linking = true;
$c->collections_always_exist = false;
$c->home_calendar_name = 'home';
$c->enable_scheduling = true;

$c->authenticate_hook['call'] = 'LDAP_check';
$c->authenticate_hook['config'] = array(
    'host' => 'example.org',
    'port' => '389',
    'protocolVersion' => '3',
    'baseDNUsers'=> 'dc=example,dc=org',
    'filterUsers' => 'objectClass=CourierMailAccount',
    'mapping_field' => array("username" => "uid",
                             "updated" => "modifyTimestamp",
                             "fullname" => "cn" ,
                             "email" => "mail"
                             ),
    'default_value' => array("date_format_type" => "E",
                    "locale" => "it_IT"
                             ),
    'format_updated'=> array('Y' => array(0,4),
                    'm' => array(4,2),
                    'd'=> array(6,2),
                    'H' => array(8,2),
                    'M'=>array(10,2),
                             'S' => array(12,2)
                             ),
    'scope' => 'subtree',
    );
$c->do_not_sync_from_ldap = array( 'admin' => true );
include('drivers_ldap.php');
$c->default_locale = "it_IT";
```

?>

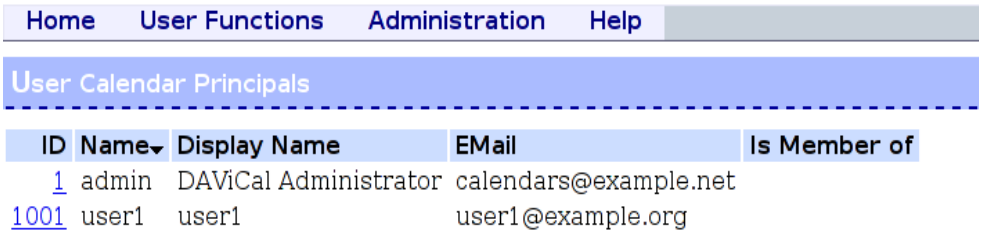

Fig. 4.8 - Lista degli utenti DAViCal sincronizzati da LDAP a seguito di una autenticazione e successiva creazione di un calendario.

Tra i vari software client (disponibili sia per dispositivi mobili che per computer) che si interfacciano correttamente con DAViCal possiamo elencare:

- Chandler
- Evolution
- Mozilla Lightning
- iCal
- iPhone

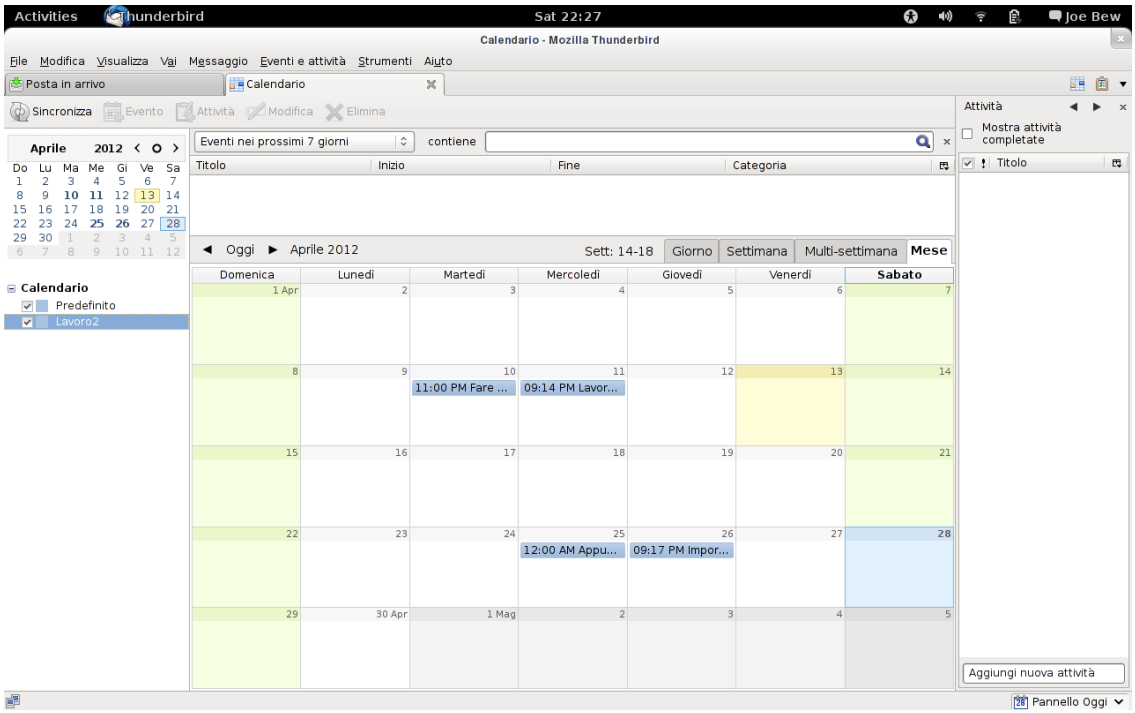

Fig. 4.9 - Mozilla Thunderbird interfacciato con DAViCal.

Quadro riepilogativo

Linguaggio di programmazione:	PHP 5.x o superiore
Supporto Database	PostgreSQL, LDAP
Protocolli utilizzati	HTTP, HTTPS, CalDAV
Architettura software	Multi-tenant
Riferimento internet	http://davical.org
Licenza	GNU GPL

4.2.4 Kronolith

Kronolith è un'applicazione web che fa parte del progetto Horde. Essa fornisce un sistema ricco di funzionalità, abbastanza stabile e maturo, per la gestione di calendari individuali per ciascun utente, che possono essere integrati tra loro attraverso meccanismi di collaborazione e scheduling.

Questa applicazione implementa una serie di funzionalità integrate nel calendario che consentono di gestire eventi ripetibili, eventi per tutto il giorno, campi personalizzabili, parole chiave, calendari condivisi, supporto al formato iCalendar e gestione di più utenti tramite il meccanismo di autenticazione disponibile nel framework Horde.

L'API di Kronolith per la gestione dei calendari fornisce un'astrazione che permette di lavorare e interfacciarsi con backend di vario genere, queli: MySQL, PostgreSQL, Oracle e Microsoft SQL Server. Può anche integrarsi con altri software groupware, come ad esempio le librerie di backend di Kolab.

Tra le varie funzionalità che tornano particolarmente utili ai nostri scopi, vi è la possibilità di interrogare Calendar Server remoti, come ad esempio DAViCal, e gestirli internamente nell'applicazione.

La configurazione di un calendario remoto è abbastanza banale: sarà sufficiente inserire l'URL che identifica il nostro calendario gestito da DAViCal e successivamente immettere i dati di autenticazione. Nel nostro esempio, prendendo in considerazione l'utente `user1` che intende collegarsi al proprio calendario, dovrà indicare come URL, un indirizzo siffatto:

```
http//example.org/davical/caldav.php/user1/calendar
```

Kronolith avvierà il processo di autenticazione, sincronizzazione e visualizzazione dei dati contenuti nel server remoto, per visualizzarli nella propria interfaccia grafica.

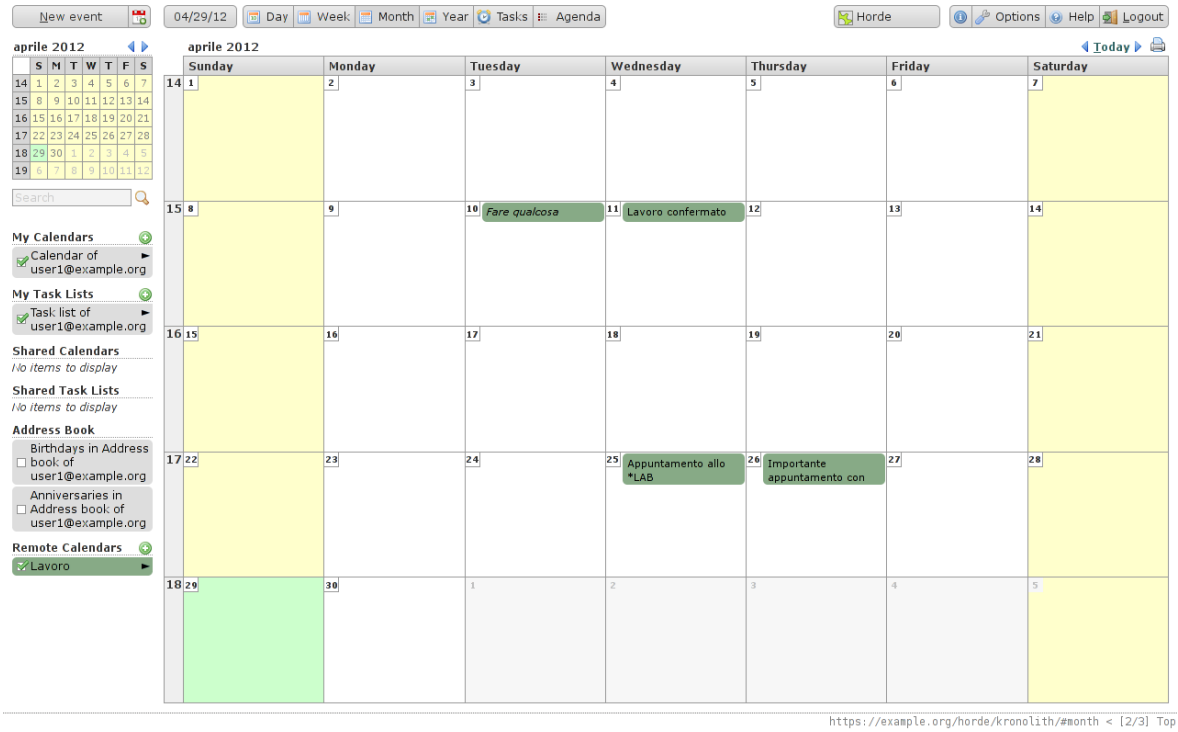

Fig. 4.10 - Schermata di Kronolith che visualizza i dati provenienti da un Calendar Server remoto, DAViCal.

Quadro riepilogativo	
Linguaggio di programmazione:	PHP 5.2.x o superiore
Supporto Database	MySQL, PostgreSQL, SQLite, LDAP
Protocolli utilizzati	HTTP, HTTPS, IMAP, POP3
Architettura software	Multi-tenant
Riferimento internet	http://www.horde.org
Licenza	GNU Lesser GPL
Dipendenze	Horde Framework

4.3 ownCloud

ownCloud punta a diventare il punto di riferimento open source di soluzioni per Cloud Storage. Basato su PHP, implementa una serie di servizi che realizzano un vero e proprio storage di rete virtuale indipendente dal dispositivo locale e che sia raggiungibile da più tipologie di software client, attraverso il protocollo WebDAV, via HTTP.

I servizi offerti da questa piattaforma possono essere acceduti direttamente via web, con un browser, e ciò lo rende completamente indipendente dal sistema operativo dell'utilizzatore. Inoltre, diverse applicazioni, già native in alcuni sistemi operativi possono accedere ai file gestiti da ownCloud tramite un apposito interfacciamento che prevede l'uso del protocollo WebDAV.

Le caratteristiche principali della piattaforma:

- I file vengono archiviati secondo una struttura ad albero
- Streaming di file audio
- Interfacciamento a LDAP
- Condivisione di file tra utenti del sistema, oppure resi pubblici
- Editor di testi via web
- Galleria fotografica

ownCloud supporto molto bene sisteme di autenticazione esterni, basati su LDAP oppure su Single-sign-on OpenID. Per questo motivo, rientra in un contesto di Private Cloud SaaS.

Inoltre, esistono già diversi software, che consentono la sincronizzazione in tempo reale di file e cartelle locali, presenti sui nostri dispositivi, con l'archivio dati ownCloud.

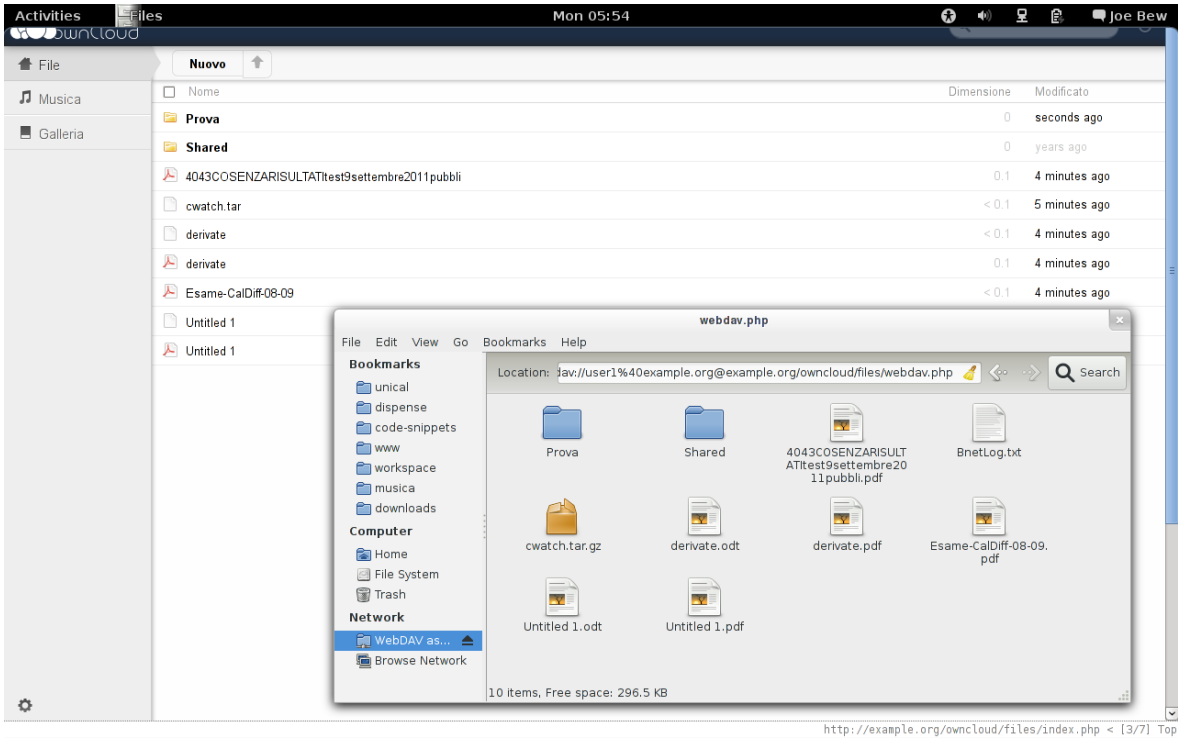

Fig. 4.11 - Sessione utente LDAP. Sullo sfondo: interfaccia grafica ownCloud per la gestione di file e

cartelle. In primo piano: storage ownCloud acceduto via WebDAV dal file manager di GNOME3.

Quadro riepilogativo	
Linguaggio di programmazione:	PHP 5.2.x o superiore
Supporto Database	MySQL, PostgreSQL
Protocolli utilizzati	HTTP, HTTPS, WebDAV, CalDAV
Architettura software	Multi-tenant
Riferimento internet	http://www.owncloud.org
Licenza	GNU Affero GPL

4.4 Funambol

È un server che fornisce funzionalità di sincronizzazione per Push Mail, PIM (rubrica contatti e calendario) e dati multimediali (immagini, video, musica), per dispositivi mobili Android, iPhone, Windows e GNU/Linux.

In un'ottica di Cloud Computing, laddove l'informazione è centralizzata e deve poter essere acceduta da più tipi di dispositivi, un server come Funambol consente di fornire un punto di accesso unico a tutti i servizi esposti in precedenza, come ad esempio rubrica contatti, appuntamenti del calendario, email e altre informazioni. In questo modo, il thin client o dispositivo che intende sincronizzarsi a tutta questa serie di informazioni, dovrà indicare solo i dati del server Funambol a cui collegarsi per ricevere gli aggiornamenti in tempo reale, piuttosto che configurare singolarmente ciascun software.

Inoltre, Funambol, una volta configurato per collegarsi a più tipi di server: email, calendar, contact o altri, sincronizzerà in tempo reale tutti i dispositivi associati ad un singolo utente.

Il sistema è basato su SyncML (Synchronization Markup Language) che è uno standard per la sincronizzazione dell'informazioni indipendente dalla piattaforma.

Funambol è anche una piattaforma di sviluppo di applicazioni per disposisitivi mobili. Fornisce delle API scritte in C++ e Java per facilitare lo sviluppo, la produzione e la gestione dei progetti.

I componenti fondamentali che costituiscono Funambol sono:

- Data Synchronization Server: il componente server per la sincronizzazione dei dati tra dispositivi mobili e sorgenti esterne.
- Device Management: il componente server per la gestione dei dispositivi mobili associati al server di sincronizzazione.
- Connectors: sono dei gateway per accedere a funzionalità del filesystem, database, sistemi di posta elettronica e altre applicazioni sviluppate da terzi per realizzare modelli di sincronizzazione a due vie con dati già esistenti.
- Client Plugin: Estensioni per la maggior parte di software esistente, come Microsoft Office Outlook, Mozilla Thunderbird, Windows Mobile, Palm, iPhone e Android per la sincronizzazione della rubrica e del calendario, con il server.

Per installare il server è necessario procurarsi l'ultima versione disponibile:

```
$ wget
http://sourceforge.net/projects/funambol/files/bundle
%2Fv10%2Ffunambol-10.0.3.bin
```

```
$ sudo ./funambol-10.0.3.bin
```

Seguiranno una serie di domande per faciltarci la configurazione. In genere il server verrà installato in `/opt` e metterà a disposizione un web server in ascolto sulla porta 8080 per varie operazioni di manutenzione.

Per avviare funambol:

```
$ sudo ./opt/Funambol/bin/funambol start
```

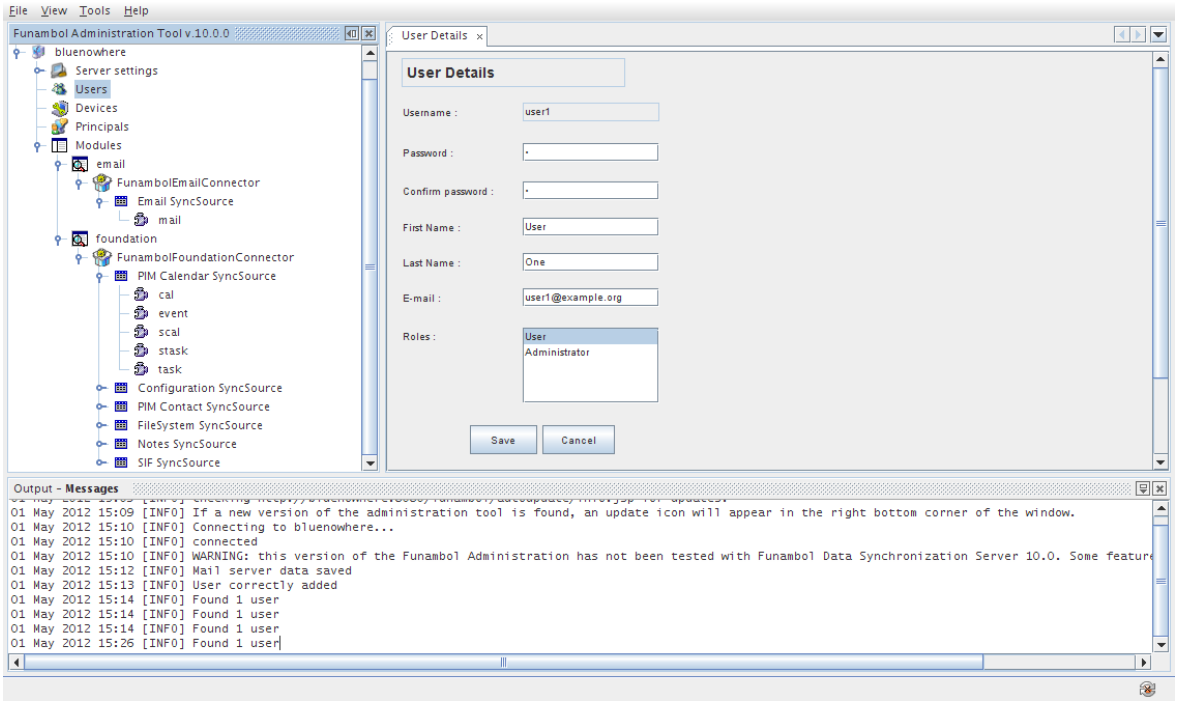

Fig. 4.12 - Schermata di Funambol relativa la gestione dell'utente.

L'autenticazione tramite LDAP e quindi il mantenimento di una base dati centralizzata, è affidata ad un modulo esterno a Funambol, chiamato LUPO (LdapUserProvisioningOfficer). Questo modulo non sostituisce il gestore utenze di Funambol, piuttosto fornisce un sistema ausiliare per l'interrogazione di LDAP durante la fase di autenticazione dell'utente.

I passaggi che vengono eseguiti durante la verifica dell'utente sono i seguenti:

1. Quando un utente avvia il processo di sincronizzazione, Funambol controllo le sue credenziali via LDAP
2. Se l'utente esiste in LDAP allora Funambol lo inserisce nella propria base dati

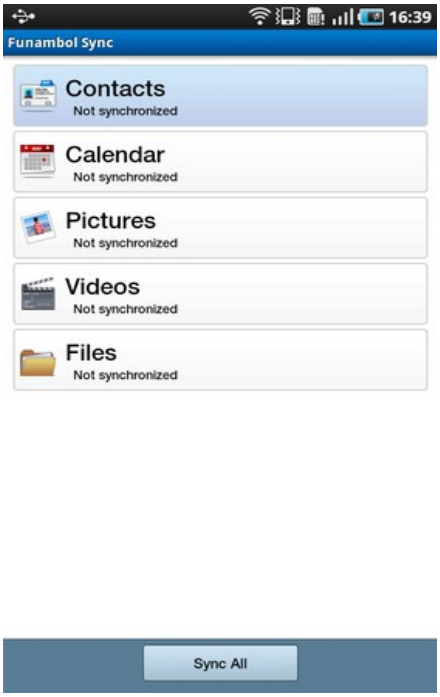

Fig. 4.13 - Client Android per Funambol (sincronizzazione di contatti, calendario, immagini, video e files).

Quadro riepilogativo	
Linguaggio di programmazione:	Java
Supporto Database	MySQL
Protocolli utilizzati	SyncML
Riferimento internet	https://www.forge.funambol.org
Licenza	GNU Affero GPL

5 RISULTATI OTTENUTI E CONCLUSIONI

Come è stato introdotto nella parte iniziale di questo lavoro, tutto ciò che è esposto in questo elaborato è da considerarsi come uno studio di fattibilità e di analisi sullo stato attuale del Software Libero, nel far fronte ad una possibile realizzazione abbastanza completa di una infrastruttura Private Cloud "Software as a Service" basata esclusivamente su soluzioni libere, sia nelle licenze d'uso che nei formati e protocolli adottati.

Si è fornito uno studio sullo stato dell'arte del Cloud Computing SaaS, analizzando la controparte commerciale e proprietaria di Google, "Google Apps". Abbiamo effettuato un'analisi ed una scelta di vari software liberi alternativi accomunandoli tramite un sistema di autenticazione utente centralizzato, basato su LDAP/OpenID.

Un primo livello di integrazione, quindi, su tutte le soluzioni proposte si è potuto raggiungere fornendo un meccanismo funzionante e collaudato per realizzare un sistema di autenticazione centralizzato per tutti gli utenti che accedono ai vari servizi erogati.

Tra i vari servizi messi a disposizione dell'utente, riassumiamo: Posta Elettronica, Instant Messaging, Calendario, Rubrica Contatti, Gestione Documentale, Editor collaborativo e in realtime di documenti di testo, Gruppi di discussione e storage di rete virtuale. Un utente appena creato può accedere a tutti questi servizi, anche da più interfacce e dispositivi, e iniziare a utilizzarli e condividerli con altri utenti interni o esterni all'infrastruttura Cloud.

In parte, si è visto come alcuni dei software disponibili possono essere interfacciati da dispositivi mobili Android e iPhone, come Tablet e Smartphone, cosicché l'accesso all'informazione sia fruibile anche da più punti, e non necessariamente localizzato in specifici dispositivi e software.

La quantità e la complessità relativamente alta delle soluzioni libere, qui analizzate, rispetto alla controparte proprietara e commerciale, ci permette di concludere che la realizzazione pratica di una infrastruttura Private Cloud SaaS "Libera", è possibile.

Molto del lavoro che resta ancora da svolgere, e che la comunità non ha ancora affrontato direttamente, è rappresentato dallo scarso livello di integrazione che più applicazioni server, di diverso tipo e per determinati domini applicativi debbano necessariamente fornire come garanzia per una certa facilità e rapidità di utilizzo dell'intera infrastruttura analizzata. Certamente, l'uso di protocolli e di formati aperti, facilita parecchio lo sviluppo di componenti che siano in grado di collegare diverse applicazioni tra loro in determinate funzionalità. Ad esempio un caso pratico potrebbe riguardare la condivisione di uno storage di rete virtuale, che sia accessibile e utilizzato da tutte le soluzioni descritte.

Aumentare il livello di integrazione tra i vari software, rappresenta uno sfida per tutta la comunità. Il lavoro di questa tesi ha dimostrato quanto sia possibile preparare una base di partenza per la realizzazione di una distribuzione GNU/Linux orientata al Cloud Computing e in particolare, al Software as a Service.

L'Hacking Laboratory Catanzaro [32], con la cooperazione dell'Hacking Laboratory Cosenza [33] e di Coopyleft [34], un'azienda con sede a Cosenza che lavora già da diversi anni nel settore del Software Libero e in particolare nelle soluzioni Cloud Computing, stanno già lavorando alla realizzazione di un progetto SaaS con Software Libero.

In futuro si spera di poter rendere disponibile una collezione di Software as a Service,

che sia pronta all'uso e in piena linea con le caratteristiche richieste dal Cloud Computing. Così facendo, probabilmente, tutte le critiche sollevate sul Public Cloud non rappresenteranno più un problema di cui avere preoccupazione.

Nel frattempo un abstract di questo lavoro è stato sottoposto per essere discusso alla VI Conferenza Italiana sul Software [35], che quest'anno si terrà il 22 e 23 Giugno presso il Dipartimento di Ingegneria dell'Informazione dell'Università Politecnica delle Marche.

Indice bibliografico

[1] "Tre domande all'inventore del "cloud computing"", http://tech4green.it/2010/08/tre-domande-allinventore-del-cloud-computing/

[2] "NIST SP 800-145, The NIST Definition of Cloud Computing", http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf

[3] "Cloud Computing Expo", http://cloudcomputingexpo.com/

[4] "Application Service Providers", http://it.wikipedia.org/wiki/Application_service_provider

[5] "Google Apps for business", http://www.google.com/apps/intl/it/business/index.html

[6] "Google App Engine", https://developers.google.com/appengine/

[7] "Google Accounts Authentication and Authorization", https://developers.google.com/accounts/

[8] "OpenID specification", http://openid.net/developers/specs/

[9] "OpenID authentication interaction sequence", https://developers.google.com/accounts/docs/OpenID#Interaction

[10] "XMPP", http://xmpp.org

[11] "CalDAV Resources", http://caldav.calconnect.org/

[12] "Internet Calendaring and Scheduling Core Object Specification", http://tools.ietf.org/html/rfc5545

[13] "Lightweight Directory Access Protocol", it.wikipedia.org/wiki/Lightweight_Directory_Access_Protocol

[14] "openLDAP", http://www.openldap.org/

[15] "Debian", http://www.debian.org/

[16] "ejabberd", http://www.ejabberd.im/

[17] "Postfix MTA", http://www.postfix.org/

[18] "Courier Mail Server", http://www.courier-mta.org/

[19] "The Horde Project", http://www.horde.org/

[20] "GroupServer", http://groupserver.org/

[21] "Alfresco community", http://www.alfresco.com/community/

[22] "Etherpad lite", https://github.com/Pita/etherpad-lite

[23] "DAViCal", http://www.davical.org/

[24] "Horde/Kronolith", http://www.horde.org/apps/kronolith

[25] "ownCloud", http://owncloud.org

[26] "Funambol forge", https://www.forge.funambol.org

[27] "phpLDAPadmin", http://phpldapadmin.sourceforge.net

[28] "OpenID-LDAP", http://www.openid-ldap.org

[29] "ZopeFive", http://codespeak.net/z3/five/

[30] "Python Programming Language", http://www.python.org/

[31] "CardDAV Resources", http://carddav.calconnect.org/

[32] "Hacking Laboratory Catanzaro", http://hacklab.cz

[33] "Hacking Laboratory Cosenza", http://hacklab.cosenzainrete.it/

[34] "Coopyleft", http://www.vinsoft.it/Coopyleft

[35] "VI Conferenza Italiana sul Software Libero", http://confsl.org/confsl12/