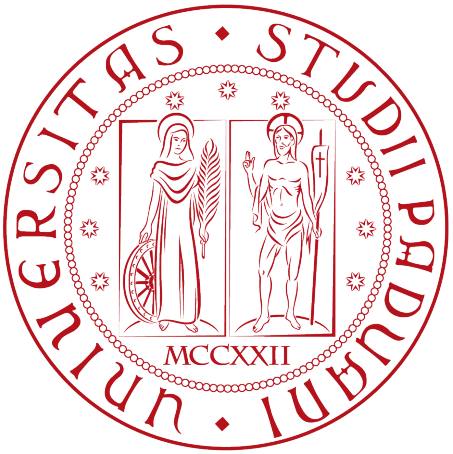
Università degli Studi di Padova

Dipartimento di Matematica "Tullio Levi-Civita"

Corso di Laurea in Informatica



Autenticazione per Zimbra Collaboration Suite (ZCS) tramite protocollo SAML

Tesi di laurea triennale

Relatore

Prof.Tullio Vardanega

Laureando

Francesco De Filippis

Anno Accademico 2019-2020

Dedicato a ...

Sommario

Il presente documento descrive il lavoro svolto durante il periodo di stage, della durata di 304 ore, dal laureando Francesco De Filippis presso l'azienda Zextras S.r.l di Torri di Quartesolo (VI). Gli obiettivi da raggiungere erano molteplici.
La prima funzionalità richiesta dall'azienda era l'autenticazione di un utente presente su Zimbra attraverso l'identity provider Okta, il quale supporta il protocollo SAML. Oltre all'autenticazione per gli utenti già esistenti su Zimbra era richiesto anche il Provisioning. In particolare si trattava di creare un nuovo account su Zimbra al primo tentativo di login dell'utente con conseguente autenticazione. I dati per la creazione dell'utente venivano forniti da Okta attraverso una SAML Assertion. Utilizzando questi dati era richiesta la configurazione dell'account creato.
Il documento è così suddiviso:

- Il primo capitolo descrive l'azienda presso cui ho svolto lo stage. In particolare viene illustrata la sua storia, i suoi prodotti e il modo in cui opera;
- Il secondo capitolo descrive gli obiettivi dello stage in relazione alle aspettative aziendali e personali;
- Il terzo capitolo descrive la scelte progettuali che ho compiuto al fine di proporre una soluzione per soddisfare gli obiettivi prefissati dallo stage;
- Il quarto capitolo presenta una valutazione dello stage in relazione agli obiettivi dell'azienda e all'esperienza da me acquisita nel corso del suo svolgimento.

Indice

Elenco delle figure

Elenco delle tabelle

Capitolo 1

L'azienda

1.1 Profilo aziendale

Zextras s.r.l. nasce a Torri di Quartesolo (VI) nel 2011 come estensione di **Studio Storti s.r.l.** che opera, dal 1997, nel campo delle soluzioni open source. Sin dall'inizio, l'obiettivo principale di questa società era quello di estendere Zimbra *Open Source Edition*, uno dei più diffusi strumenti collaborativi per aziende e pubbliche amministrazioni, aggiungendo nuove funzionalità.
Nel corso degli anni è nata e cresciuta **Zextras Suite**, una raccolta di estensioni che permettono di arricchire Zimbra con nuove funzionalità utili nel suo utilizzo in ambito professionale. Le soluzioni proposte da **Zextras** con i sui prodotti vengono da subito apprezzate da **Synacor**, l'azienda che sviluppa e mantiene Zimbra, la quale decide di includere parte del suo codice nella versione open source. Attualmente i prodotti sviluppati dall'azienda vengono utilizzati da più di 100 milioni di utenti in tutto il mondo.

1.2 Dominio applicativo

1.2.1 Zimbra Open Source Edition

Zextras, come già accennato, è nata con l'obiettivo di creare nuovi contenuti per Zimbra facendone quindi il suo *core business*. Zimbra è un software collaborativo di gruppo adatto a coordinare e supportare l'attività lavorativa di aziende, pubbliche amministrazioni e altri enti. I principali servizi offerti da questo *software* sono i seguenti:

- posta elettronica;
- gestione calendari condivisi e organizzazione eventi;
- interfaccia amministratore;
- supporto dei servizi su dispositivi mobili.

Per estendere l'applicativo con ulteriori funzionalità, sviluppate anche da terze parti, è possibile installare un plug-in che in ambiente Zimbra viene chiamato **Zimlet**.
Esistono due versioni di Zimbra:

- **Zimbra Open Source Edition**: è la versione su cui lavora **Zextras** e offre i servizi elecanti in precedenza;
- **Zimbra Network Edition**: è una versione a pagamento che offre alcune funzionalità closed source tra cui un protocollo per la sincronizzazione di calendario e contatti e maggiori funzionalità per gli amministratori.

1.2.2 Zextras Suite

Zextras Suite è un'insieme di funzionalità che permettono di aggiungere delle funzionalità a Zimbra *Open Source Edition* in modo indipendente da quest'ultimo. Ciò permette una configurazione altamente modulare e personalizzabile in base alle necessità dell'utilizzatore.
Questa *suite* offre i seguenti prodotti:

- **Powerstore**: sistema di ottimizzazione dei dati che permette il risparmio di memoria sui server Zimbra;
- **Backup**: motore di backup in real-time;
- **Admin**: strumenti dedicati agli amministratori per la gestione e il monitoraggio dei servizi attivi sull'istanza di Zimbra;
- **Mobile**: gestione e sincronizzazione di posta elettronica, contatti, eventi e calendario su dispositivi mobili tramite protocolli *Exchange* e *EAS 16.0 (ActiveSync)*;
- **Chat**: piattaforma di messaggistica istantanea nativamente integrata in Zimbra, che permette scambio di messaggi e videochiamate;
- **Drive**: piattaforma per la condivisione di file e l'utilizzo di fogli di lavoro condivisi.

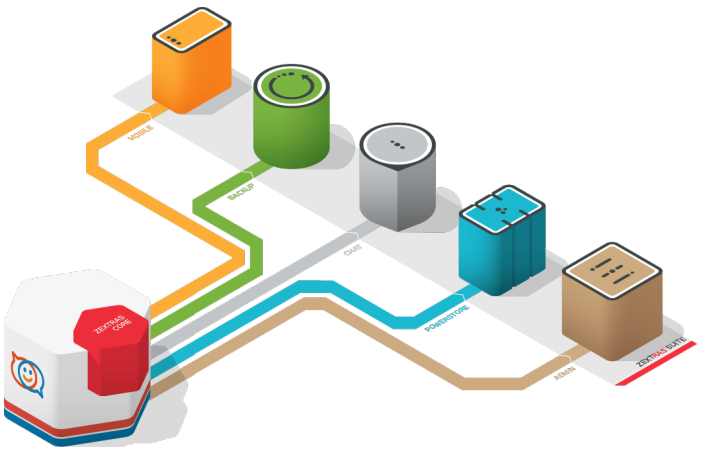


Figura 1.1: Zextras *suite*
Fonte: zimbra-zextras.pl

1.3 Struttura interna

L'azienda è suddivisa in diversi settori specializzati, ciò permette di avere un organico fornito di tutte le competenze necessarie per raggiungere gli obiettivi prefissati. Di seguito un elenco che illustra i diversi reparti:

- **Commercio**: questo reparto, facente parte di **Studio Storti**, si occupa di gestire tutto ciò che riguarda la parte commerciale e *marketing* dei prodotti proposti dall'azienda;
- **System administration**: questo team svolge l'attività di gestione e manutenzione dell'infrastruttura interna all'azienda che ospita numerosi *server*, i quali erogano i servizi offerti da Zimbra ai clienti;
- **Team di sviluppo**: il team di sviluppo è suddiviso in più aree, ognuna delle quali lavora su un aspetto speficifico dei prodotti di **Zextras**:
  - **Front-end**: questa divisione progetta e sviluppa tutto ciò che riguarda l'interfaccia grafica delle applicazioni web;
  - **Back-end**: questo team si occupa di progettare e implementare l'architettura di tutti i servizi presenti nei prodotti dell'azienda. É ulteriormente suddiviso in due team, ciascuno responsabile di un insieme di prodotti diverso;
  - **UI/UX Design**: si occupa di progettare l'interfaccia grafica e di studiare l'user experience delle applicazioni web. Lavora a stretto contatto con la divisione front-end;
  - **Mobile**: progettazione e sviluppo delle applicazioni per dispositivi mobili.

1.4 Processi aziendali

I processi sono un tassello di estrema importanza all'interno di un'azienda che si pone degli obiettivi ben definiti, soprattutto quando questi sono particolarmente ambiziosi e raggiungibili tramite lo sviluppo di prodotti complessi e di grandi dimensioni. Sono inoltre fondamentali qualóra l'azienda avesse al suo interno dei reparti che svolgono mansioni molto differenti tra loro.
Al fine di orchestrare e mettere in funzione tutti i settori dell'azienda affinché questi lavorino in modo coeso, è necessario ricorrere alla definizione e istanziazione di processi, i quali permettono di stabilire con precisione quali sono le attività da svolgere e il modo più strategico per portarle a termine, al fine di raggiungere gli obiettivi prefissati.

1.4.1 Fornitura

La grande diffusione di Zimbra in molteplici ambiti, contribuisce alla necessità di nuove funzionalità e al miglioramento di quelle già esistenti. Per questo motivo **Zextras** ottiene spesso nuovi incarichi e nuovi progetti da sviluppare.
La ricerca di nuovi progetti avviene tramite il *CEO* dell'azienda e il *Project Manager*. Oltre all'esigenze dei clienti, nascono dei progetti anche interni dell'azienda stessa che derivano, ad esempio, dalla necessità di migliorare i processi interni con dei nuovi strumenti.

Una volta individuato un possibile progetto, viene fissata una riunione alla quale partecipano i responsabili e il team di sviluppo (formato da gruppi appartenenti a più reparti) che dovrà poi occuparsi del ciclo di vita del nuovo prodotto. Per progetti di grandi dimensioni o particolarmente complessi, la fase di studio di fattibilità dura più tempo e prevede un numero maggiore di riunioni, durante le quali si cerca fin da subito di individuare possibili soluzioni ad alto livello per valutarne la fattibilità.
Dalle riunioni effettuate vengono generati dei verbali, dai quali si estraggono le informazioni più rilevanti che andranno a contribure ai primi contenuti della documentazione. Quando la soluzione viene approvata da azienda e cliente, si procede con le fasi successive.

1.4.2 Comunicazione

Una comunicazione efficiente all'interno del nucleo aziendale è fondamentale per poter essere tutti allineati e coordinati, soprattutto quando è necessario interagire con membri di altri team collocati in altre aree dell'azienda.

Strumenti utilizzati

- **Posta elettronica**: servizio offerto tramite la posta elettronica di Zimbra;
- **Team**: piattaforma di messaggistica istantanea integrata in Zimbra sviluppata da **Zextras**;
- **Drive**: piattaforma per la condivisione di documenti su *server* sviluppata dall'azienda;
- **Riunioni**: per discussioni con tutti i membri (o una parte) del team, è possibile indire delle brevi riunioni.

1.4.3 Metodologia di sviluppo

Per sostenere lo sviluppo di software complessi e dalle dimensioni sostanziose, è necessario avere a disposizione una certa flessibilità ed essere pronti a possibili cambiamenti in termini di requisiti, data anche la tiplogia di servizi che l'azienda sviluppa. Per questo motivo **Zextras** adotta una filosofia *agile*, preferita rispetto a modelli più rigorosi con l'istanziazione di processi molto forti che portano ad una minore capacità di reagire a cambiamenti imminenti. In particolare, la metodologia di sviluppo adottata in azienda è *Scrum*. In seguito descriverò *Scrum* in relazione a come l'azienda lo attua e da quello che concerne la mia esperienza di stage.
Lo sviluppo di un progetto viene suddiviso in fasi, chiamati *Sprint*, la cui durata può variare da una a quattro settimane. In questo caso, la durata dello *Sprint* era di due settimane. Ciascuno *Sprint* ha uno o più obiettivi da portare a termine entro la fine di esso. Il *framework Scrum* prevede lo svolgimento di alcuni eventi e di seguito saranno descritti quelli effettivamente svolti:

- ***Sprint Planning***: si tratta di una riunione che viene indetta all'inizio di ogni *Sprint*, durante la quale vegono discussi gli obiettivi da raggiungere (*Sprint Goal*). Nello specifico, lo *Scrum Master* ovvero colui che coordina il team sceglie, insieme ai colleghi, i *task* da svolgere nel corso dello *Sprint* e li inserisce nello *Sprint Backlog*. I *task* vengono scelti dal *Product Backlog*, ovvero dalla lista completa di

funzionalità da implementare nel prodotto. Vengono inoltre stabilite le ore che ciascun membro del team potrà e dovrà dedicare allo sviluppo, alle riunioni e ad altre attività durante il prossimo *Sprint*;

- ***Daily Scrum***: è una riunione della durata di circa 15 minuti che viene svolta ogni inizio giornata con il team di sviluppo (ed eventuali altre persone coinvolte) chiamato anche *Stand-up meeting*. Duranta questa riunione ciascun membro del team parla di ciò che ha svolto il giorno precedente, se ha incontrato ostacoli nello svolgere i suoi *task* e su quali lavorerà durante la giornata. Per più della metà dello stage, i *daily scrum* del mio team si sono svolti in lingua inglese al fine di migliorare le abilità linguistiche;

- ***Sprint Retrospective***: questo evento prevede una riunione in cui il team esegue una valutazione retrospettiva sull'andamento dello *Sprint* appena concluso. In particolare vengono analizzate le metriche riguardanti il numero di *task* portati a termine e gli obiettivi raggiunti. Durante questa riunione viene inoltre svolta un'altra importante attività, ovvero la discussione delle abitudini del team e la ricerca di nuove *best practice*. Per fare ciò, ricorre allo *Starfish Retrospective*, uno schema in cui è possibile individuare i seguenti punti:

  - ***More of***: elenco delle attività che svolte con più frequenza gioverebbero alla crescrita del team e allo sviluppo del prodotto;
  - ***Less of***: elenco delle attività da ridurre;
  - ***Start doing***: nuove attività individuate di recente che potrebbero migliorare il team e il prodotto;
  - ***Keep doing***: attività che è utile continuare a svolgere;
  - ***Stop doing***: attività il cui svolgimento va interrotto poiché non più necessario oppure perché si tratta di *bad practice*.

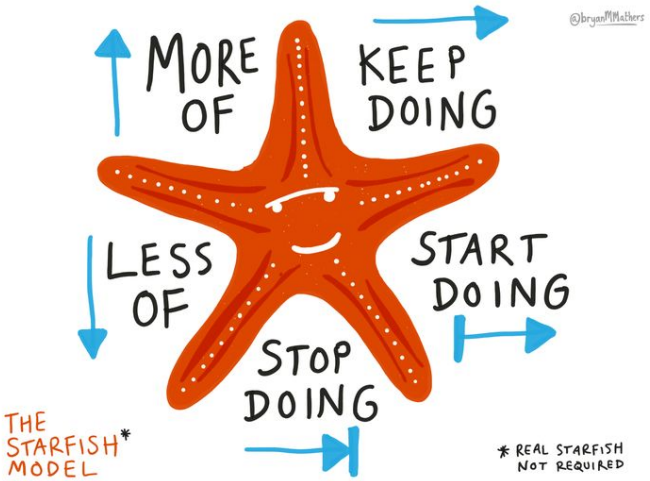


Figura 1.2: *Starfish retrospective*
Fonte: bryanmathers.com

- ***Backlog Refinement***: questa attività, svolta durante lo *Sprint Planning*, consiste nel controllare che tutti gli *item* presenti nel *Product Backlog* siano pronti per essere selezionati e inseriti nello *Sprint Backlog*. Ne vengono rivisti i contenuti, le priorità, le persone assegnate per il loro svolgimento e se necessario, vengono suddivisi in *task* più piccoli o inclusi in altri già esistenti.

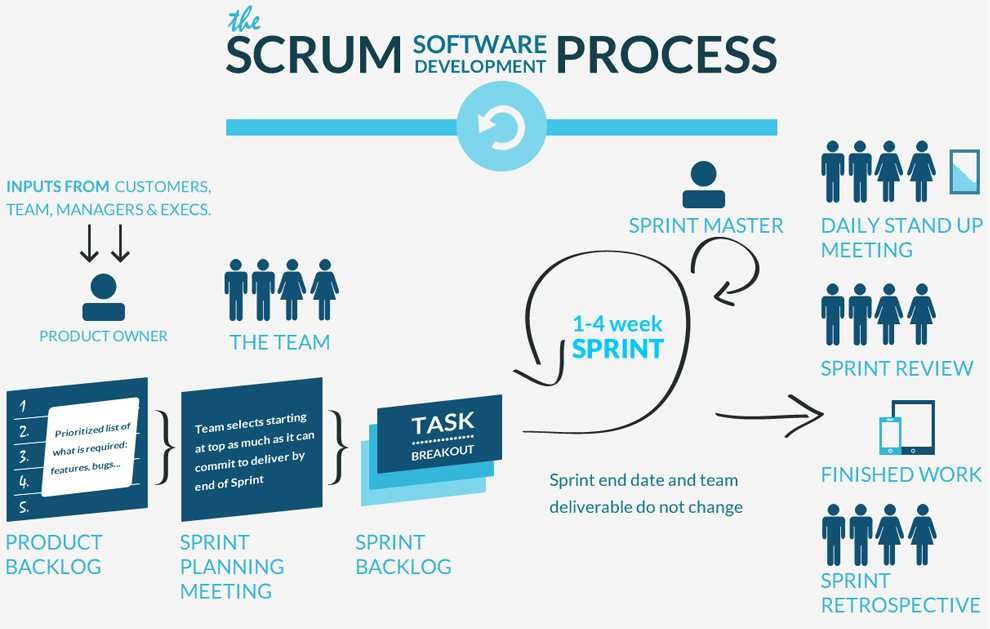


Figura 1.3: Scrum *flow*
Fonte: ICS

1.4.4 Gestione di progetto

La gestione del progetto viene supportata principalmente da due strumenti:

- **Atlassian Jira**: è un *issue tracking system* molto avanzato che offre la possibilità di gestire tutto ciò che riguarda il *framework Scrum*, quindi gestione degli *Sprint*, dei *task* e molto altro. Da **Jira** è inoltre possibile tracciare le ore impiegate per portare a termine ciascun *task* fornendo, al termine dello *Sprint*, delle serie storiche con i tempi impiegati dai membri del team e il numero di *task* portati a termine rispetto a quelli previsti nello *Sprint Backlog*;

- **Zimbra**: offre, tra i tanti servizi, il calendario per la gestione degli eventi interni all'azienda, per esempio le riunioni e la possibilità di lavorare su documenti condivisi, per esempio i fogli di calcolo utilizzati per il tracciamento delle ore durante lo *Sprint Planning* come descritto nella sezione precedente.

1.4.5 Documentazione

La documentazione di tutti i team di sviluppo è raccolta in un unico punto accessibile a tutti. Per ottenere ciò viene utilizzato il *software* di collaborazione **Confluence** sviluppato da **Atlassian**, il quale offre un editor di tipo *WYSIWYG* che permette di scrivere documenti completi e ben formattati oltre a fornire un sistema di catalogazione molto flessibile e personalizzabile.

1.4.6 Configurazione

Gli strumenti per il versionamento del codice utilizzati sono i seguenti:

- **Git**: sistema di versionamento;
- **Bitbucket**: servizio che ospita i *repository* di tutti i team di sviluppo. Il vantaggio di usare questo strumento è la sua completa integrazione con tutti gli altri strumenti di **Atlassian** impiegati dall'azienda;
- **GitHub**: servizio che ospita i repository *open source* dell'azienda.

Per lo sviluppo di nuove funzionalità viene utilizzato un *workflow* molto simile a ***Gitflow***[1]. Il funzionamento è il seguente:

1. Creazione di un nuovo *branch* sul quale sviluppare una nuova funzionalità;
2. Implementazione nuova funzionalità;
3. *Pull request* per effettuare il *merge* del codice sul *branch master*, nella quale vengono inseriti alcuni membri del team come revisori, al fine di effettuare una *code review* la quale permette di verificare che il codice sia in linea con gli standard di qualità.
4. Dopo aver messo a punto le correzioni segnalate nella *code review* e aver ricevuto l'approvazione dai revisori, avviene il *merge* sul *branch master*.

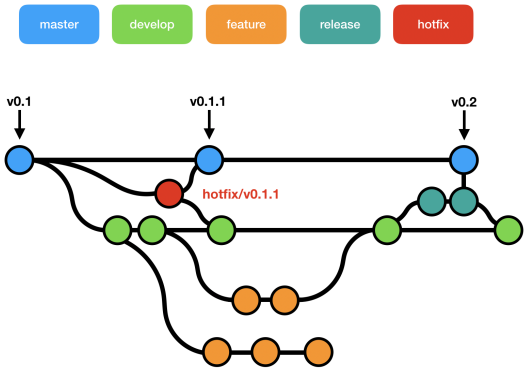


Figura 1.4: *Gitflow*
Fonte: codewall.co.uk

[1] https://danielkummer.github.io/git-flow-cheatsheet/

1.4.7 Verifica

La verifiche che permettono di perseguire la qualità di prodotto avvengono tramite **Jenkins**, un *automation server* che supporta la *continuous integration* e la *continuous delivery*. In questo modo è possibile automatizzare il processo di build ed esecuzione delle varie tipologie di test ad ogni commit sul *repository* e rilasciare quest'ultima in ambiente di produzione. **Jenkins** permette di aggiungere altre funzionalità al processo di build, per esempio l'analisi statica del codice e la rilevazione di alcune tipologie di *bug*, *code coverage* rendendolo quindi uno strumento molto flessibile e personalizzabile. Prima della verifica automatizzata vengono effettuate le *code review* come descritto nella sezione precedente.

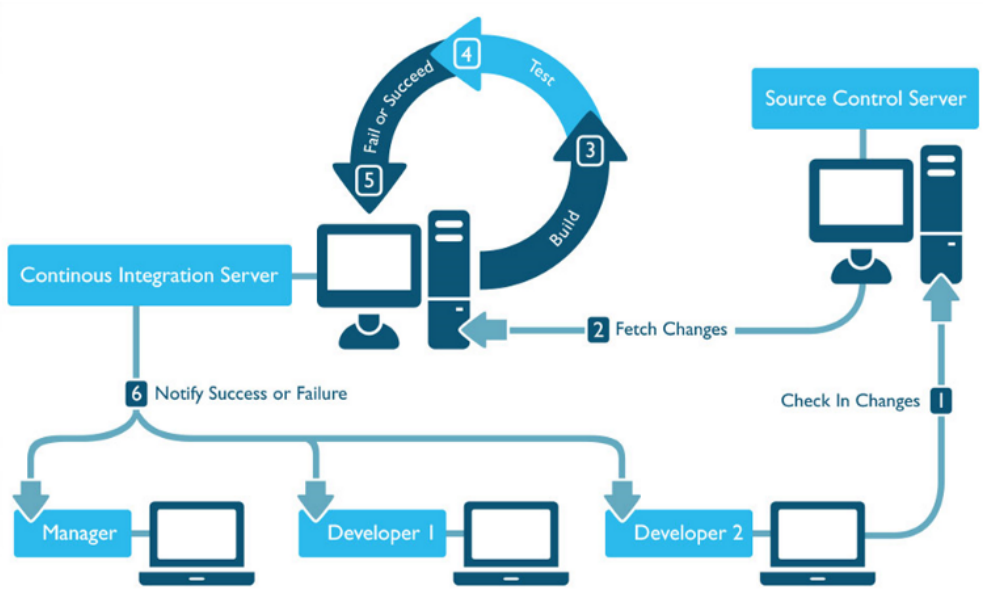


Figura 1.5: *Continuous Integration*
Fonte: developers.redhat.com

Capitolo 2

Obiettivi dello stage

2.1 Presentazione del progetto

Questo progetto nasce dall'esigenza di avere un sistema di autenticazione per la *webmail* di Zimbra che fornisse il giusto connubio tra sicurezza e facilità d'uso. **Zextras** utilizza molti strumenti e servizi al suo interno, sia per lo sviluppo sia per l'amministrazione. Tutti questi servizi utilizzano Okta per la gestione dell'autenticazione, ciò significa che è sufficiente effettuare una sola autenticazione con il proprio account su Okta per poter poi accedere a tutti i servizi ad esso collegati.
Questa tipologia di autenticazione si chiama Single Sign-On (SSO) e consiste nell'utilizzo di una singola credenziale che permette di accedere a più servizi diversi.
Zimbra era l'unico serivizio, uno dei più estensivamente utilizzati in azienda, che non beneficiava di questa tecnica di autenticazione. Per questo motivo **Zextras** prende la decisione di esplorare la possibilità di svilupparne una personalizzata che, oltre all'autenticazione base, avesse le seguenti caratteristiche:

- quando un utente che possiede un *account* su Okta ma non su Zimbra, accede per la prima volta su quest'ultima, viene creato un nuovo account su Zimbra, associato a quello di Okta;
- importazione su Zimbra delle informazioni utente presenti su Okta;
- sincronizzazione dei gruppi di Okta ai quali un utente appartiene, con liste di distribuzione e classi di servizio di Zimbra.

Il fine di questo progetto era quello di uniformare il metodo di autenticazione di Zimbra rispetto a tutti gli servizi utilizzati dall'azienda e soprattutto avere un sistema personalizzato e configurabile che permette di automatizzare alcune operazioni ripetitive e dispendiose in termini di tempo. Per poter proporre una soluzione conforme alle esigenze emerse, ho dovuto per prima cosa condurre uno studio e un'analisi dei protocolli di autenticazione presenti sul mercato. La seconda parte invece consisteva nella progettazione e implementazione del sistema utilizzando il protocollo più idoneo.

Figura 2.1: Single Sign-On
Fonte: developer.fourth.com

2.1.1 Analisi stato dell'arte protocolli di autenticazione

La prima attività che ho dovuto svolgere era la ricerca e lo studio dei protocolli di autenticazione più diffusi e utilizzati sul mercato. L'azienda conosceva già alcuni di questi, ma voleva avere un'analisi approfondita e degli scenari pratici. Inoltre poteva rivelarsi una valida occasione per scoprire nuovi possibili protocolli adatti all'implementazione del sistema di autenticazione.
L'azienda ha quindi richiesto di redigere un documento che riportasse tutti i risultati delle mie ricerche, in particolare l'analisi dei pro e dei contro di ciascun protocollo e se questo potesse essere idoneo al nostro caso d'uso.

2.1.2 Progettazione di un sistema di autenticazione personalizzato

In seguito all'analisi svolta, il secondo obiettivo dello stage era la progettazione, seguita dall'implementazione, di un sistema di autenticazione per Zimbra utilizzando il protocollo ritenuto più idoneo. Il sistema di autenticazione doveva essere in grado di:

- utilizzare Okta come identity provider;
- supportare altri identity provider che utilizzato lo stesso protocollo;
- supportare Zimbra tramite il Single Sign-On di Okta rendendo opzionale l'accesso tramite *email* e *password*;
- garantire sicurezza.

Figura 2.2: *Research & Development*
Fonte: youteam.io

2.2 Vantaggi aziendali

Zextras ospita stage di questa tipologia da ormai diversi anni nonostante sia un'attività che richiede un quantità di tempo e impegno non indifferente poiché è costretta a sottrarre risorse dai progetti e dalle attività in corso che, molto spesso, sono vincolate da scadenze. Tuttavia i vantaggi portati da uno strumento come lo stage sono molteplici. Per prima cosa l'azienda ha la possibilità di condurre dei progetti di ricerca che molto spesso coinvolgono l'utilizzo di nuove tecnologie, senza essere costretta a rallentare lo sviluppo dei prodotti ordinari e soprattutto riducendo il rischio di investire troppe risorse in progetti che potenzialmente potrebbero non proseguire.
Un altro aspetto interessante dal punto di vista aziendale è il poter entrare in contatto con studenti che, seppur privi di esperienza lavorativa, sono spesso in grado di proporre soluzioni alternative e creative a problemi comuni. I vantaggi elencati si sono effettivamente concretizzati nel tempo, infatti negli utlimi anni il team di sviluppo ha integrato nel suo organico alcuni studenti che, una volta completato lo stage, sono rimasti a contribuire alle sfide tecnologiche che l'azienda affronta giornalmente.
Ciò significa che uno stage ben organizzato e seguito si rivela essere un vero e proprio investimento, non solo per l'azienda che lo ospita, la quale beneficerà degli *output* di questo strumento, ma anche per la crescita degli studenti, i quali saranno le figure professionali che nel futuro faranno parte dell'intera industria che al giorno d'oggi necessita di molte risorse vista la sua dinamicità.

2.3 Vincoli

2.3.1 Vincoli metodologici

Per lo svolgimento dell'attività di stage è stato deciso, in comune accordo con il *tutor*, che il modo più efficace per portarla a termine fosse lavorare presso la sede aziendale. La prima motivazione deriva dalla metodologia di sviluppo adottata, descritta nella sezione §1.4.3, la quale è fortemente incentrata sulla comunicazione e sul confronto con tutti i membri nel *team*. Inoltre, poiché sarei stato affiancato da alcuni *senior developer* si trattava di un'ottima occasione per apprendere il più possibile da professionisti nel settore.
Oltre agli incontri di allineamento giornalieri, il *Project Manager* ha stabilito che ci sarebbero stati degli incontri formali, insieme al *tutor* aziendale, per fare il punto della situazione. Nella fase conclusiva del progetto era inoltre prevista una *demo* a cui avrebbero presenziato:

- il *CEO* dell'azienda;
- il *Project Manager*;
- il responsabile tecnico dell'azienda;
- il *team* di sviluppo con il *tutor* aziendale;
- il responsabile di un altro *team*;

La presenza di figure appartenti a diversi settori e *team* sarebbe state utile a discutere il prodotto sviluppato, sia dal punto di vista funzionale sia da quello infrastrutturale. Inoltre, ricevere un *feedback* da punti di vista diversi è certamente utile per il miglioramento dell'applicazione nel suo insieme.

Figura 2.3: *Teamwork*
Fonte: sandler.com

2.3.2 Vincoli temporali

L'attività di stage prevista aveva una durata di 304 ore, da svolgere nell'arco di due mesi, suddivise in 8 settimane della durata di circa 40 ore ciascuna. L'orario di lavoro accordato con l'azienda era dal Lunedì al Venerdì dalle ore 9.00 alle 18.00.
Prima dell'inizio dello stage ho pianificato, insieme al *tutor* le attività per ciascuna settimana di lavoro, facendone una stima orario per il completamento. Sin dal primo giorno io e il *tutor* aziendale abbiamo rispettato il piano di lavoro stabilito. Tuttavia, in seguito all'attività di analisi svolta, come descritto nella sezione §2.1.1, abbiamo dovuto dedicare più tempo per ottenere un prototipo base funzionante, poiché non era ancora chiaro come utilizzare il protocollo di autenticazione scelto. Nonostante un rallentamento a metà percorso, il resto dello stage ha avuto un andamento lineare che mi ha permesso di terminare il lavoro senza fretta.

Figura 2.4: *Planning*
Fonte: sysaid.com

2.3.3 Vincoli tecnologici

Le tecnologie e i linguaggi utilizzati duarante questo progetto dello *stack* tecnologico dell'azienda e sono le seguenti:

- **Java**: essendo il linguaggio con cui è scritto Zimbra, ne consegue che tutta la tecnologia di **Zextras** si sia adeguata, compreso il mio progetto;
- **Git**: sistema di versionamento, come descritto nella sezione §1.4.6;
- **Docker**: è una tecnologia che permette di creare, rilasciare ed eseguire delle applicazioni utilizzando i container. In questo modo il container potrà essere eseguito, tramite *docker*, su macchine diverse che solitamente sono configurate in diversi modi. Questo comportamente è simile alla virtualizzazione e permette di rendere gli ambienti di esecuzione solidi e deterministici. In particolare l'azienda lo utilizza per creare delle istanze di Zimbra utilizzate come ambienti di test in fase di sviluppo. In azienda i container di *docker* sono gestiti tramite un servizio di nome ***Portainer***[1].

[1] https://www.portainer.io/

2.4 Aspettative aziendali

Al termine delle 304 ore previste per il completamento dello stage, l'azienda si aspettava di avere almeno un prototipo funzionante che facesse uso di un protocollo di autenticazione, al fine di permettere l'autenticazione su Zimbra tramite l'identity provider Okta. Come già descritto in precedenza però, la parte interessante di questo progetto era l'aggiunta di alcune funzionalità personalizzate ad un sistema di autenticazione standard. Per questo motivo dopo aver concluso l'attività di ricerca sui protocolli, descritta nel paragrafo §2.1.1, ho discusso, insieme al *team* e al *Project Manager*, i requsiti specifici da implementare. Ciò è stato utile perché al momento della stesura del piano di lavoro, alcuni di essi non potevano essere a me chiari, causa mancanza di contesto riguardante l'ambiente Zimbra. Gli obiettivi stabiliti erano suddivisi, in ordine di importanza, in **obbligatori** e **desiderabili**.

Obiettivi obbligatori

- Analisi stato dell'arte dei protocolli di autenticazione più diffusi;
- Implemenazione di un sistema di autenticazione per Zimbra tramite il protocollo scelto;

Obiettivi desiderabili

- *Provisioning*;
- Importazione dei dati dell'utente da Okta a Zimbra;
- Flusso di autenticazione a partire sia dall'identity provider sia da Zimbra;
- Controllo delle classi di servizio di Zimbra tramite l'identity provider;
- Gestione delle liste di distribuzione di Zimbra tramite l'identity provider;
- Autenticazione a due fattori;
- Integrazione con *WebAuthn*[2]

2.5 Aspettative personali

Descrizione di ciò che io mi aspettavo al momento della scelta di questo stage

[2] https://www.w3.org/TR/webauthn-2/

Capitolo 3

Resoconto dello stage

3.1 Descrizione del progetto

Descrizione dettagliata del progetto

3.2 Analisi

Descrizione dell'analisi dei requisiti e descrizione di alcuni dei requisiti

3.3 Pianificazione

Descrizione di come ho pianificato lo sviluppo del progetto rispettando i vincoli descritti in precedenza e le esigenze dell'azienda

3.4 Scelta del protocollo di autenticazione

Descrizione delle motivazioni che hanno portato alla scelta del protocollo SAML in seguito all'analisi svolta

3.4.1 Confronto SAML e OpenID

Breve confronto di due protocolli rispetto al problema da risolvere

3.4.2 SAML

Descrizione del protocollo SAML

3.5 Progettazione

Progettazione della soluzione proposta per l'implementazione del sistema di autenticazione

3.5.1 Configurazione applicazione Okta

3.5.2 Progettazione handler HTTP

3.5.3 Configurazione Zimbra

3.6 Sviluppo

Alcuni dettagli di rilievo per quanto riguarda la parte implementativa. In particolare ciò che ho dovuto attuare per superare alcuni ostacoli

3.6.1 Parsing SAML assertion

3.6.2 Adattamento libreria HTTP

3.7 Documentazione

Documentazione del codice e del funzionamento del prodotto sviluppato

3.7.1 Codice

3.7.2 Manutenzione

3.8 Verifica e Validazione

Descrizione dell'attività di verifica svolta e della validazione

3.8.1 Verifica

3.8.2 Validazione

Capitolo 4

Valutazione retrospettiva

4.1 Soddisfacimento degli obiettivi

Illustrazione degli obiettivi raggiunti rispetto a quelli attesi da me e dall'azienda

4.2 Conoscenze e abilità acquisite

Descrizione di tutte le conoscenze e competenze acquisite sul piano tecnico e sul mondo del lavoro

4.3 Valutazione personale

Valutazione personale sullo di stage in relazione all'esperienza accademica

Glossario

Agile Approccio allo sviluppo software che pone il focus sul consegnare al cliente un *software* completo, funzionante e di qualità in tempi brevi. 4, 18

Backup Quando si parla di *backup*, si fa riferimento al processo di duplicazione di dati su più supporti (fisici o cloud) al fine di poterli recuperare in caso di perdita inattesa. 2, 18

Bug In informatica si tratta di un errore *software* che produce risultati inattesi. 8, 18

Closed Source Con il termine ***closed source*** si fa riferimento ad un software proprietario, quindi non disponibile pubblicamente. 2, 18

Code Review Attività di revisione del codice effttuata da persone diverse dagli autori, al fine di correggere errori, migliorarne la qualità ed eventualmente proporre soluzioni alternative. 7, 8, 18

Container un *Docker container* è un'unità *software* che contiene un ambiente di esecuzione completo di librerie, dipendenze e configurazioni che è in grado di essere eseguito in modo sicuro e deterministico in altri ambienti *Docker* ospitati su diverse macchine. 13, 18

Continuous Delivery Pratica nell'ambito dell'ingeneria del *software* che consiste nel rilasciare la *build* di un *software* pronta per l'ambiente di produzione. 8, 18

Continuous Integration Pratica nell'ambito dell'ingeneria del *software* che consiste nell'integrazione frequente del lavoro svolto negli ambienti in locale degli sviluppatori verso l'ambiente condiviso, ovvero il *repository* in remoto. 8, 18

CRUD Questo acronimo viene spesso usato in ambito di ***database management*** e indica:

- ***Create***: creazione di un utente;
- ***Read***: richiesta attributi di un utente;
- ***Update***: aggiornamento attributi di un utente;
- ***Delete***: non si parla di una cancellazione vera e propria di un utente ma di ***deprovisioning***, ovvero una disabilitazione dell'*account* di quest'ultimo o di un cambio di permessi

. 19

Demo dimostrazione di in tempo reale di un prodotto, in questo caso di un *software*. 12, 19

Framework Insieme di strumenti che definiscono la struttura di un sistema a livello concettuale. Nel caso del *software* si può intedendere come un'architettura sulla quale basare lo sviluppo di un prodotto. Quando si parla di modello di sviluppo si intende l'insieme di strumenti teorici che permettono di mettere in atto un concetto specifico. 4, 6, 19

Identity Provider Un *identity provider* è un sistema che crea, mantiene e gestisce le informazioni sull'identità di un utente. Si occupa di fornire il servizio di autenticazione ai service provider. v, 10, 14, 19

Issue Tracing System Software che permette di gestire in maniera ordinata un insieme di issue, ovvero dei *task* da svolgere. Tipicamente viene utilizzato in ambito collaborativo in quanto permette di tenere traccia delle *issue* risolte da tutti i membri del *team*. 6, 19

Okta Okta è una società di gestione di identità e di accessi, quindi un identity provider. v, 9, 10, 14, 19

Open Source Con il termine ***open source*** si fa riferimento ad un *software* la cui lincenza permette di utilizzarlo, modificarlo e redistribuirlo. 1, 7, 19

Plug-in Componente *software* che aggiunge funzionalità all'applicazione su cui viene installato. 1, 19

Provisioning Con il termine ***provisioning*** si intende, generalmente, la gestione degli utenti. Questo termine include un insieme di funzionalità riassunte dall'acronimo CRUD. v, 14, 19

Real-time *Software* che opera sotto condizioni temporali ben definite. 2, 19

Repository Nell'ambito dello sviluppo *software* rappresenta un contenitore di codice sorgente, gestito da un sistema di versionamento. 7, 8, 18, 19

SAML (Secuity Assertion Markup Language) É un protocollo bastato su XML che permette lo scambio di messaggi per effettuare autenticazione e autorizzazione tra domini distinti. Tipicamente gli attori del procollo sono un identity provider che fornisce l'identità dell'utente da autenticare e un service provider che fornisce il servizio a cui l'utente ha richiesto l'accesso. . v, 19

SAML Assertion Una asserzione SAML è un documento in formato XML che contiene le informazioni sull'autenticazione e/o autorizzazione di un utente. Tale documento è solitamente generato dall'identity provider e inviato al service provider. v, 19

Service Provider Un *service provider* ed è un sistema che fornisce un servizio a degli utenti. Lo si può identificare come un sito web che eroga un certo servizio. 19

SSO (Single Sign-On) Si tratta di un sistema di autenticazione che permette ad un utente di effettuare un'unica autenticazione, valida per più servizi e/o risorse ai quali è abilitato. Questo permette all'utente di avere un'unica credenziale valida per più servizi indipendenti. 10

Task Incarico di piccole dimensione assegnato ad un soggetto che dovrà portarlo a termine. 4–6, 19, 20

User experience L'esperienza che manifesta l'utente, nell'interagire con un certo prodotto, sistema o servizio. 3, 20

Workflow Flusso di esecuzione di un insieme di attività. 7, 20

WYSIWYG Sta per "*What You See Is What You Get*" e si riferisce ad una tipologia di *editor* di testo in grado di mostrare in tempo reale, durante la scrittura, quale sarà l'aspetto finale del documento. 7, 20

XML Linguaggio di ***Markup*** che consente la definizione di **metadati**. 19

Zimbra Collaboration software collaborativo che offre servizi come la posta elettronica e calendario condiviso. É possibile estenderlo con nuove funzionalità tramite meccanismi chiamati **Zimlet**. v, 1–4, 6, 9, 10, 13, 14, 20

Bibliografia

Siti web consultati

Manifesto Agile. URL: https://agilemanifesto.org/.

Opensource. URL: https://opensource.com/resources/what-docker.

Wikipedia. URL: https://www.wikipedia.org/.