

Università degli Studi di Padova

DIPARTIMENTO DI MATEMATICA "TULLIO LEVI-CIVITA"

CORSO DI LAUREA IN INFORMATICA



**Progettazione e sviluppo di un servizio di
Identity Access Managment basato su
blockchain**

Tesi di laurea triennale

Relatore

Prof. Gilberto Filè

Laureando

Simone Ballarin

Lorem ipsum dolor sit amet, consectetur adipiscing elit.

— Oscar Wilde

Dedicato a ...

Sommario

Il presente documento riassume il lavoro svolto durante il periodo di stage, della durata di 320 ore, presso l'azienda iVoxIT S.r.l. di Padova.

Lo scopo principale del prodotto sviluppato è quello di integrare all'interno dell'applicativo Monokee, un sistema di creazione e verifica dell'identità basato su tecnologia blockchain compatibile con lo standard SAML. Nel corso del documento verranno anche esposte le basi teoriche del prodotto, come la gestione delle identità e il Single Sign-On (SSO).

In una prima fase mi sono concentrato sullo studio di vari documenti forniti dall'azienda inerenti al progetto e a come questo si doveva integrare con l'attuale sistema, una volta capiti gli aspetti fondamentali mi sono dedicato alla ricerca e all'apprendimento di vari strumenti tecnologici confacenti ad un corretto sviluppo. Dopo aver identificato le principali funzionalità richieste e compreso il funzionamento di come erano definiti i flussi di lavoro ho iniziato la progettazione del servizio e quindi alla definizione di un'architettura che fosse estendibile, manutenibile e integrabile con quella esistente. Questo lavoro ha richiesto molti sforzi, ma il risultato, seppure all'altezza della aspettative dell'azienda, ha fatto emergere come un'approccio basato su blockchain risulti essere non adatto nella maggior parte dei contesti.

“Life is really simple, but we insist on making it complicated”

— Confucius

Ringraziamenti

Innanzitutto, vorrei esprimere la mia gratitudine al Prof. NomeDelProfessore, relatore della mia tesi, per l'aiuto e il sostegno fornitomi durante la stesura del lavoro.

Desidero ringraziare con affetto i miei genitori per il sostegno, il grande aiuto e per essermi stati vicini in ogni momento durante gli anni di studio.

Ho desiderio di ringraziare poi i miei amici per tutti i bellissimi anni passati insieme e le mille avventure vissute.

Padova, Agosto 2018

Simone Ballarin

Indice

1	Introduzione	1
1.1	L'azienda	1
1.2	L'idea	2
1.3	Organizzazione del testo	2
2	Processi e metodologie	3
2.1	Processo sviluppo prodotto	3
2.1.1	Metodologie di sviluppo Agile	3
2.1.2	Scrum	4
3	Descrizione dello stage	5
3.1	Introduzione al progetto	5
3.2	Analisi preventiva dei rischi	5
3.3	Requisiti e obiettivi	5
3.4	Pianificazione	5
4	Analisi dei requisiti	7
4.1	Casi d'uso	7
4.2	Tracciamento dei requisiti	8
5	Progettazione e codifica	11
5.1	Tecnologie e strumenti	11
5.2	Ciclo di vita del software	11
5.3	Progettazione	11
5.4	Design Pattern utilizzati	11
5.5	Codifica	11
6	Verifica e validazione	13
7	Conclusioni	15
7.1	Consuntivo finale	15
7.2	Raggiungimento degli obiettivi	15
7.3	Conoscenze acquisite	15
7.4	Valutazione personale	15
A	Appendice A	17
	Bibliografia	21

Elenco delle figure

1.1	Logo aziendale iVoIT	1
2.1	Diagramma flusso Scrum	4
4.1	Use Case - UC0: Scenario principale	7

Elenco delle tabelle

4.1	Tabella del tracciamento dei requisiti funzionali	9
4.2	Tabella del tracciamento dei requisiti qualitativi	9
4.3	Tabella del tracciamento dei requisiti di vincolo	9

Capitolo 1

Introduzione

Introduzione al contesto applicativo.

Esempio di utilizzo di un termine nel glossario
[Application Program Interface \(API\)](#).

Esempio di citazione in linea
site:agile-manifesto

Esempio di citazione nel pie' di pagina
citazione¹

1.1 L'azienda

L'attività di stage è stata svolta presso l'azienda iVoIT S.r.l. (logo in figura 1.1) con sede a Pavoda presso il centro direzionale La Cittadella. iVoxIT S.r.l. con Athesys



Figura 1.1: Logo aziendale iVoIT

S.r.l. e Monokee S.r.l. fa parte di un gruppo di aziende fondato nel 2010 dall'unione di professionisti dell' [Information Technology](#)^[8] (IT) con l'obiettivo di fornire consulenza ad alto livello tecnologico e progettuale. Tra le altre cose, Athesys S.r.l fornisce supporto

¹womak:lean-thinking.

nell'istanziamento del processo di [Identity Access Management](#)^[gl] (IAM) , con particolare attenzione alla sicurezza nella conservazione e nell'esposizione dei dati sensibili gestiti. L'azienda opera in tutto il territorio nazionale, prevalentemente nel Nord Italia, e vanta esperienze a livello europeo in paesi quali Olanda, Regno Unito e Svizzera. Grazie all'adozione delle [best practise](#)^[gl] definite dalle linee guida [Information Technology Infrastructure Library](#)^[gl] (ITIL) e alla certificazione ISO 9001 il gruppo è in grado di assicurare un'alta qualità professionale.

1.2 L'idea

Nell'ottica di estendere le funzionalità del prodotto Monokee di [Identity Access Management](#) basato su Cloud, lo stage ha visto lo sviluppo di due moduli applicativi in ambito Blockchain. Il primo modulo è una applicazione mobile (Wallet) contenente l'identità digitale dell'utente finale mentre il secondo modulo è un layer applicativo (Service Provider) per gestire gli accessi alle applicazioni di terze parti. In figura ?? un'immagine dei moduli da implementare e il loro posizionamento in un tipico scenario di accesso ai servizi. La tipologia di Blockchain da integrare è stata individuata in una

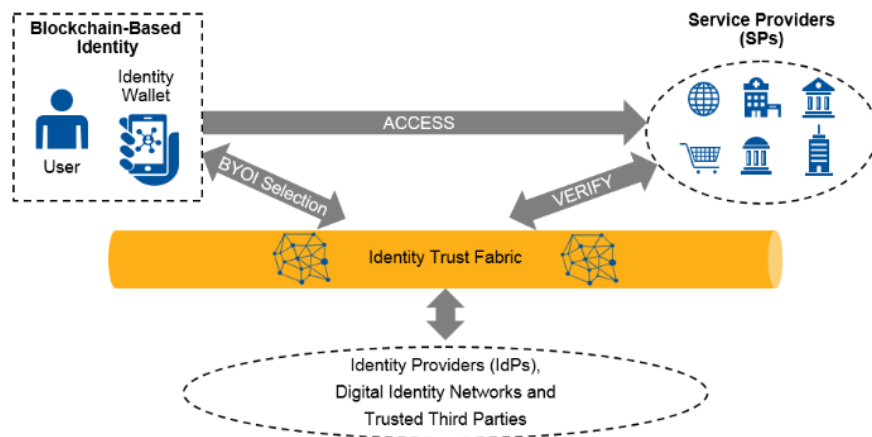


Figura 1.2: Diagramma Moduli

prima prima fase di analisi.

1.3 Organizzazione del testo

[Il secondo capitolo](#) descrive ...

[Il terzo capitolo](#) approfondisce ...

[Il quarto capitolo](#) approfondisce ...

[Il quinto capitolo](#) approfondisce ...

[Il sesto capitolo](#) approfondisce ...

[Nel settimo capitolo](#) descrive ...

Riguardo la stesura del testo, relativamente al documento sono state adottate le seguenti convenzioni tipografiche:

- * gli acronimi, le abbreviazioni e i termini ambigui o di uso non comune menzionati vengono definiti nel glossario, situato alla fine del presente documento;
- * per la prima occorrenza dei termini riportati nel glossario viene utilizzata la seguente nomenclatura: *parola*^[g];
- * i termini in lingua straniera o facenti parti del gergo tecnico sono evidenziati con il carattere *corsivo*.

Capitolo 2

Processi e metodologie

Brevissima introduzione al capitolo

2.1 Processo sviluppo prodotto

Durante ogni attività è stata seguita una metodologia di sviluppo [agile](#). L'azienda iVoxIT S.r.l. per la precisione attua in ogni suo progetto il metodo [Scrum](#). Le attività sono state descritte in task secondo la modalità [Scrum](#); ogni task veniva esposto e discusso in riunioni giornaliere con il tutor aziendale Dott. Sara Meneghetti. Inoltre erano previste riunioni settimanali con il responsabile del progetto Ing. Roberto Griggio.

2.1.1 Metodologie di sviluppo Agile

le metodologie di sviluppo agile si basano su quattro principi cardine:

- * il software funzionante prima dei documenti;
- * il rapporto con il cliente;
- * i rapporti interno al team;
- * rispondere al cambiamento;

Dal momento che i processi di pianificazione necessari ad uno sviluppo a cascata sono molto costosi e che se questi non vengono rispettati tutta la pianificazione deve essere completamente rivista, queste metodologie si basano su modelli incrementali. Gli approcci agile tendono a progettare il minimo indispensabile in modo tale da essere sempre più reattivi e proattivi possibili agli inevitabili. Inoltre, scrivere del software incrementale permette una progettazione iniziale molto snella, la quale con l'accrescere della conoscenza sul dominio può diventare sempre affinata. Questo in conclusione rende meno costoso il refactoring del codice e anche l'inserimento di nuove funzionalità. Questi metodi sono adatti per piccoli team, molto coesi e uniti, non dislocati in regioni diverse in quanto la comunicazione di persona è fondamentale. Il team in cui ero inserito difatti si componeva di tre membri. Un altro punto critico è che essendo l'approccio alla comprensione dei requisiti e alla progettazione poco concentrato all'inizio e molto diluito in tutto il progetto questo rende necessario un costante interesse da parte degli

[stakeholders](#), cosa che in un nuovo progetto è ipotizzabile, mentre in un progetto in fase manutentiva no.

2.1.2 Scrum

Scrum è un metodo iterativo che divide il progetto in blocchi rapidi di lavoro chiamati Sprint, della durata massima di quattro settimane. Alla fine di ogni Sprint si ottiene un incremento del prodotto. Durante ogni fase dello stage si è seguito questo modello. Scrum prevede una prima fase di analisi e progettazione di massima, poi il lavoro viene suddiviso in unità creabili e implementabili in un unico sprint e messi in un backlog, Prima di ogni sprint si sceglie una selezione di lavori in base alle priorità e si inseriscono nel backlog dello sprint. Gli sprint durano da 1 a 4 settimane e se il lavoro non viene portato a termine non viene prolungato lo sprint, ma rimesso nel backlog principale. Emerge come sia importante dare il giusto quantitativo di lavoro. È compito dello Scrum Master assicurarsi del rispetto dei tempi: quotidianamente effettua una breve riunione, della durata di circa quindici minuti, per valutare i progressi fatti (Daily Scrum). I rapporti tra i membri del progetto sono stati importanti e per questo Scrum prevede riunioni giornaliere per rimanere aggiornati sullo stato dei lavori di ogni componente. In Figura 2.1 è mostrato un tipico ciclo Scrum.

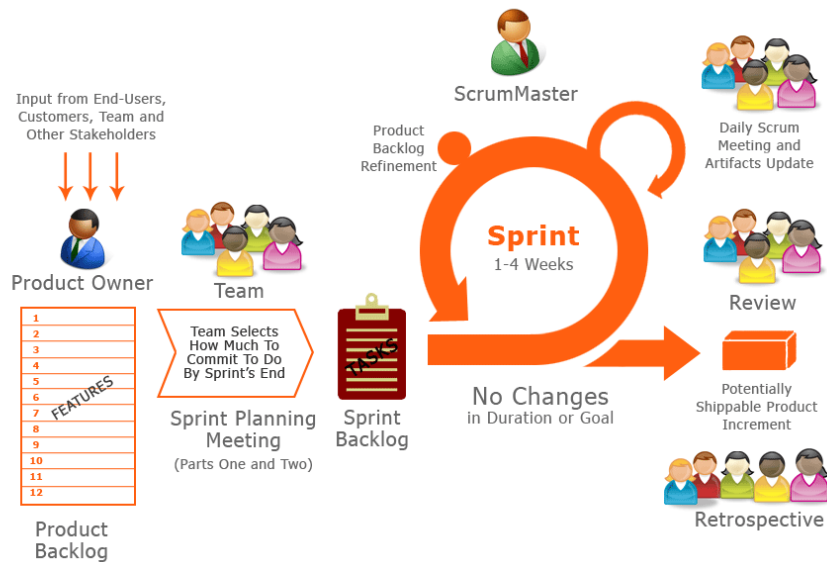


Figura 2.1: Diagramma flusso Scrum

Capitolo 3

Descrizione dello stage

Breve introduzione al capitolo

3.1 Introduzione al progetto

3.2 Analisi preventiva dei rischi

Durante la fase di analisi iniziale sono stati individuati alcuni possibili rischi a cui si potrà andare incontro. Si è quindi proceduto a elaborare delle possibili soluzioni per far fronte a tali rischi.

1. Performance del simulatore hardware

Descrizione: le performance del simulatore hardware e la comunicazione con questo potrebbero risultare lenti o non abbastanza buoni da causare il fallimento dei test.

Soluzione: coinvolgimento del responsabile a capo del progetto relativo il simulatore hardware.

3.3 Requisiti e obiettivi

3.4 Pianificazione

Capitolo 4

Analisi dei requisiti

Breve introduzione al capitolo

4.1 Casi d'uso

Per lo studio dei casi di utilizzo del prodotto sono stati creati dei diagrammi. I diagrammi dei casi d'uso (in inglese *Use Case Diagram*) sono diagrammi di tipo [Unified Modeling Language \(UML\)](#) dedicati alla descrizione delle funzioni o servizi offerti da un sistema, così come sono percepiti e utilizzati dagli attori che interagiscono col sistema stesso. Essendo il progetto finalizzato alla creazione di un tool per l'automazione di un processo, le interazioni da parte dell'utilizzatore devono essere ovviamente ridotte allo stretto necessario. Per questo motivo i diagrammi d'uso risultano semplici e in numero ridotto.

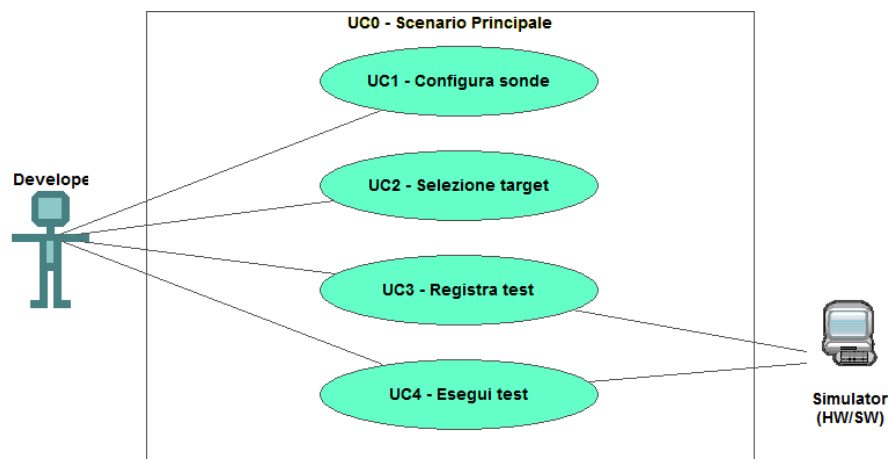


Figura 4.1: Use Case - UC0: Scenario principale

UC0: Scenario principale

Attori Principali: Sviluppatore applicativi.

Precondizioni: Lo sviluppatore è entrato nel plug-in di simulazione all'interno dell'IDE.

Descrizione: La finestra di simulazione mette a disposizione i comandi per configurare, registrare o eseguire un test.

Postcondizioni: Il sistema è pronto per permettere una nuova interazione.

4.2 Tracciamento dei requisiti

Da un'attenta analisi dei requisiti e degli use case effettuata sul progetto è stata stilata la tabella che traccia i requisiti in rapporto agli use case.

Sono stati individuati diversi tipi di requisiti e si è quindi fatto utilizzo di un codice identificativo per distinguerli.

Il codice dei requisiti è così strutturato $R(F/Q/V)(N/D/O)$ dove:

R = requisito

F = funzionale

Q = qualitativo

V = di vincolo

N = obbligatorio (necessario)

D = desiderabile

Z = opzionale

Nelle tabelle 4.1, 4.2 e 4.3 sono riassunti i requisiti e il loro tracciamento con gli use case delineati in fase di analisi.

Tabella 4.1: Tabella del tracciamento dei requisiti funzionali

Requisito	Descrizione	Use Case
RFN-1	L'interfaccia permette di configurare il tipo di sonde del test	UC1

Tabella 4.2: Tabella del tracciamento dei requisiti qualitativi

Requisito	Descrizione	Use Case
RQD-1	Le prestazioni del simulatore hardware deve garantire la giusta esecuzione dei test e non la generazione di falsi negativi	-

Tabella 4.3: Tabella del tracciamento dei requisiti di vincolo

Requisito	Descrizione	Use Case
RVO-1	La libreria per l'esecuzione dei test automatici deve essere riutilizzabile	-

Capitolo 5

Progettazione e codifica

Breve introduzione al capitolo

5.1 Tecnologie e strumenti

Di seguito viene data una panoramica delle tecnologie e strumenti utilizzati.

Tecnologia 1

Descrizione Tecnologia 1.

Tecnologia 2

Descrizione Tecnologia 2

5.2 Ciclo di vita del software

5.3 Progettazione

Namespace 1

Descrizione namespace 1.

Classe 1: Descrizione classe 1

Classe 2: Descrizione classe 2

5.4 Design Pattern utilizzati

5.5 Codifica

Capitolo 6

Verifica e validazione

Capitolo 7

Conclusioni

7.1 Consuntivo finale

7.2 Raggiungimento degli obiettivi

7.3 Conoscenze acquisite

7.4 Valutazione personale

Appendice A

Appendice A

Citazione

Autore della citazione

Bibliografia