IW-Studio di fattibilità

Autore: Simone Ballarin
Data: 12/06/18
Destinatari: Athesys

Diario delle modifiche

Data	Descrizione	Autore
11/06/2018	Creazione documento e stesura dei capitoli: Scopo del documento, Riferimenti, Descrizione.	Simone Ballarin
12/06/2018	Stesura capitoli: Studio del Dominio.	Simone Ballarin
14/06/2018	Stesura capitoli Tecnologie per lo sviluppo, Breve considerazione sulla comunicazione ITF e IW, e Studio diffusione sistemi operativi mobili.	Simone Ballarin
15/06/2018	Stesura capitoli Breve considerazione sullo sviluppo mobile, Motivazioni. Piccola correzione al capitolo descrizione (autenticazione multi fattore).	Simone Ballarin

Scopo del documento

Il seguente documento *Studio di fattibilità* ha lo scopo di fornire una macro descrizione ed un primo approccio relativo ai benefici ed ai costi di una eventuale progettazione, implementazione e messa in produzione di un Identity Wallet (IW) che dovrà funzionare nel contesto di un'estensione del prodotto MonoKee basata su blockchain.

Sintesi del documento

Il documento inizia descrivendo le caratteristiche del prodotto MonoKee e di come l'IW si cali in questo contesto. Si prosegue analizzando le alternative di sviluppo mobile e Desktop. A seguito di un'analisi dei possibili utenti emerge una netta preferenza per lo sviluppo mobile.
Vengono poi trattati i principali strumenti e librerie disponibili per lo sviluppo. Si ritiene di preferire uno sviluppo multi piattaforma, con target Android e iOS. L'analisi conclude facendo emergere una preferenza per il framework Xamarin. All'interno del documento sono presenti delle considerazioni sull'aspetto della comunicazione tra IW e ITF. Quest'ultima cosa potrebbe porre un grande limite della soluzione mobile da tenere a mente.

Riferimenti

1. Blockchain: The Dawn of Decentralized Identity (G00303143), Homan Farahmand per Gartner

Descrizione

Il progetto ha come scopo la creazione di un Identity Wallet (IW). L'applicativo si colloca nel contesto di un'estensione del servizio Monokee basato su blockchain. L'estensione offre un sistema di Identity Access Management (IAM) composto da quattro principali fattori:

1. Identity Wallet (IW)
2. Service Provider (SP)

3. Identity Trust Fabric (ITF)
4. Trusted Third Party (TTP)

In sintesi l'estensione dovrà operare al fine di fornire la possibilità ad un utente di registrare e gestire la propria identità automatamente tramite l'IW, mandare i propri dati (IPP) all'ITF la quale custodirà la sua identità e farà da garante per le asserzioni proveniente dai TTP. Inoltre il SP dovrà essere in grado con le informazioni provenienti da IW e ITF di garantire o meno l'accesso ai propri servizi.

Il software IW, più dettagliatamente, dovrà assolvere ai seguenti compiti:

nell'ambito della registrazione di un utente il Wallet deve:

- generare e immaganizzare in locale una chiave pubblica;
- generare e immaganizzare in locale una chiave private;
- creare l'hash della chiave la chiave pubblica e inviare l'hash all'ITF;
- incrementare le informazioni (PII) relative alle identità che il Wallet gestisce.

Nell'ambito della presentazione dei dati ad un Service Provider deve:

- invio della chiave pubblica al service provider;
- invio di un puntatore all'hash della chiave pubblica interna al ITF;
- invio di altre informazioni utili presenti nel ITF;
- gestire ulteriori layer di sicurezza, quali impronta digitale, QR code, autentificazione multi fattore)

nell'ambito della richiesta di accesso ad un servizio deve:

- inviare una richiesta di accesso ad un servizio con i dati relativi all'identità al Service Provider;
- attendere la risposta del Service Provider.

Studio del dominio

Dominio applicativo

L'applicativo IW dovrà essere usato in un contesto prevalentemente lavorativo. Non si escludono però ulteriori applicazione future in ambito Consumer. In ogni caso il software dovrà poter essere usato da utenti senza specifiche conoscenze informatiche e con minimo training tecnologico. Dato il contesto applicativo il software dovrà essere il più accessibile e semplice possibile. Per queste ragioni si pensa ad un suo utilizzo prevalentemente in ambito mobile, anche se non si esclude a priori la possibilità di una versione Desktop. L'applicativo mobile deve essere disponibile per la più ampia gamma di utenti possibili.

Dominio tecnologico

Un eventuale applicativo Desktop ha un'elevata probabilità di non rientrare dentro i tempi dello stage, per questo si ritiene di non tenerlo in considerazione. L'applicativo quindi dovrà essere

fruibile tramite un'applicazione mobile multipiattaforma sviluppabile entro i limiti temporali della durata dello stage.

Per queste ragioni lo studio del dominio tecnologico si incentrerà principalmente su tecnologie multi piattaforma mobili. Verrà comunque tenuto in considerazione anche lo sviluppo nativo.

Studio diffusione sistemi operativi mobili

Procedo di seguito ad un'analisi sulla diffusione dei vari sistemi operativi mobili. I dati di seguito riportati provengono da Kantar società di analisi inglese e fanno riferimento al trimestre che va da novembre 2016 a gennaio 2017.

Smartphone OS Sales Share (%)

Germany	3 m/e Jan'16	3 m/e Jan'17	% pt. Change
Android	74.2	75.5	1.3
iOS	19.3	21.3	2.0
Windows	5.9	2.9	-3.0
Other	0.7	0.2	-0.5

USA	3 m/e Jan'16	3 m/e Jan'17	% pt. Change
Android	58.2	56.4	-1.8
iOS	39.1	42	2.9
Windows	2.6	1.3	-1.3
Other	0.1	0.3	0.2

GB	3 m/e Jan'16	3 m/e Jan'17	% pt. Change
Android	52.6	54.4	1.8
iOS	38.6	43.3	4.7
Windows	8.6	1.9	-6.7
Other	0.2	0.3	0.1

China	3 m/e Jan'16	3 m/e Jan'17	% pt. Change
Android	73.9	83.2	9.3
iOS	25.0	16.6	-8.4
Windows	0.9	0.1	-0.8
Other	0.3	0.1	-0.2

France	3 m/e Jan'16	3 m/e Jan'17	% pt. Change
Android	71.9	72.9	1.0
iOS	19.3	24.2	4.9
Windows	7.8	2.8	-5.0
Other	0.9	0.2	-0.7

Australia	3 m/e Jan'16	3 m/e Jan'17	% pt. Change
Android	52.6	55.7	3.1
iOS	41.2	42.4	1.2
Windows	5.4	1	-4.4
Other	0.8	0.8	0.0

Italy	3 m/e Jan'16	3 m/e Jan'17	% pt. Change
Android	78.1	79	0.9
iOS	14.4	15.8	1.4
Windows	7.2	4.4	-2.8
Other	0.3	0.8	0.5

Japan	3 m/e Jan'16	3 m/e Jan'17	% pt. Change
Android	48.7	49	0.3
iOS	50.3	49.5	-0.8
Windows	0.5	1.5	1.0
Other	0.5	0	-0.5

Spain	3 m/e Jan'16	3 m/e Jan'17	% pt. Change
Android	87.8	89.4	1.6
iOS	11.4	10.2	-1.2
Windows	0.8	0.4	-0.4
Other	0	0	0.0

EU5	3 m/e Jan'16	3 m/e Jan'17	% pt. Change
Android	72.9	74.3	1.4
iOS	20.3	22.7	2.4
Windows	6.4	2.7	-3.7
Other	0.5	0.3	-0.2

Da come si può evincere dai dati, Android, iOS, Windows Phone rappresentano, in questa sequenza ed in ogni mercato, i sistemi più diffusi. I restanti sistemi non raggiungono cifre significative. Ponendo maggiore attenzione ai primi tre sistemi si nota come Android nell'area EU5 rappresenti i ¾ del mercato. In Giappone, Stati Uniti, Australia e Gran Bretagna invece la situazione risulta più bilanciata con una sostanziale parità. Windows Phone in ogni mercato si pone in terza posizione con percentuali che non superano mai l'otto percento. Individuando nell'area EU5 il principale mercato per MonoKee si ritiene che il prodotto IW debba essere sviluppato per i sistemi Android e iOS, dando la precedenza al primo. Non si ritiene necessario lo sviluppo di un'applicazione Windows Phone in quanto difficilmente attuabile nei tempi dello stage.

Tecnologie per lo sviluppo

Segue un approfondimento relativo alle potenziali tecnologie con cui sviluppare l'IW. Data la necessità di sviluppare sia per Android, che per iOS l'analisi si concentrerà principalmente su framework multi piattaforma senza comunque ignorare la possibilità di sviluppi nativi.

Sviluppo multi piattaforma

Tra le principali alternati multi piattaforma si ritengono particolarmente interessanti le seguenti:

- React Native;
- Cordova;
- Xamarin;

Segue un'analitica descrizione dei vari framework.

React Native: è un framework di sviluppo mobile derivato da React. Il progetto e sviluppato e mantenuto da Facebook. React Native si focalizza nello sviluppo di UI tramite componenti scritti in JavaScript, su un approccio funzionale e flusso di dati unidirezionale. A differenza di React, React Native non manipola il DOM del browser, ma una struttura diversa. I componenti non vengono scritti a partire da elementi HTML o simili (i.e. Bootstrap o Grommet), ma bensì a partire da un set di componenti base presenti nella libreria. La libreria permette di sviluppare applicazioni per iOS e Android.

Cordova: è un framework open-source per lo sviluppo di applicazione mobili che propone un approccio ibrido e non nativo. Permette di usare tecnologie web ampiamente utilizzate, quali HTML5, CSS3, Javascript, per la codifica. Il software così prodotto verrà eseguito in appositi wrapper diversi per ogni piattaforma, quindi in maniera non nativa. Il framework è sviluppato da Apache ed ormai ha raggiunto un elevato grado di maturità. Rappresenta uno dei primi framework per lo sviluppo multi piattaforma.

Xamarin: è un framework per lo sviluppo di applicazioni native e multi piattaforma con C#. Il framework si basa sul progetto open source Mono e offre pieno supporto non solo alle piattaforme Android e iOS ma anche a Windows Phone. La possibilità di sviluppare anche per Windows Phone potrebbe risultare un punto a favore rispetto agli altri framework. Xamarin si compone di tre componenti principali: Xamarin.Android, Xamarin.iOS, Xamarin.Forms. L'ultimo componente si pone come strumento completamento neutro rispetto alla piattaforma. Grazie a queste componenti è possibile gestire in C# tutte le caratteristiche di Android, iOS e Windows Phone.

Framework	**Approccio**	**Piattaforme supportate**	**Linguaggio**
React Native	Nativo	iOS, Android	Javascript
Cordova	Ibrido	iOS, Android	HTML5, CSS3, Javascript
Xamarin	Nativo	iOS, Android, Windows Phone	C#

Data l'impossibilità degli approcci ibridi, quali Cordova, di sfruttare a pieno le caratteristiche tipiche delle diverse piattaforme mobili si ritiene di scartare questo tipo di soluzioni.
Inoltre si evidenziano difetti come una mancata o incompleta integrazione dell'aspetto grafico con la specifica piattaforma e una maggiore lentezza nell'esecuzione e accesso alle risorse locali.
Per queste ragione si ritiene più opportuno l'utilizzo di un framework che permetta di scrivere applicazioni in maniera nativa.
Richiudendo la visione ai soli approcci nativi, Xamarin rispetto a React Native lascia aperte le porte ad una eventuale applicativo Windows Phone. Inoltre utilizza C# un linguaggio che rispetto a Javascript fornisce una tipizzazione forte e caratteristiche più orientate agli oggetti. Per queste ragioni si consiglia l'utilizzo di Xamarin o React Native con la preferenza per il primo.

Sviluppo nativo

Un'applicazione nativa è un'applicazione mobile sviluppata interamente nel linguaggio del dispositivo sul quale vengono eseguite. Quindi stiamo parlando di Java per Android e Swift o Object-C per iOS. Il loro utilizzo presenta diversi vantaggi rispetto allo sviluppo multi piattaforma:

- interazione con tutte le caratteristiche del dispositivo consentendo l'utilizzo al 100%;
- maggiore velocità offrendo quindi una User Experience di più alto livello;
- facilità di integrazione di terze parti tramite utilizzo di SDK ufficiali.

Il primo punto non dovrebbe rappresentare un plus in quanto non si ritiene che l'IW debba usufruire di feature particolari dei dispositivi.
È da notare che uno sviluppo nativo richiede il doppio delle risorse necessarie in quanto prevede lo sviluppo di due applicazione completamente diverse (Android e iOS), con framework e quindi architetture potenzialmente diverse.
Riassumendo, dato che:

- l'applicativo che si dovrà sviluppare non prevede particolari requisiti prestazionali;
- l'alto costo in termini orari di sviluppare soluzioni differenti ha una forte probabilità di non rientrare nei tempi previsti dall'attività di stage;

si ritiene non conveniente lo sviluppo parallelo di più applicazioni native.

Conclusioni

A seguito di quanto detto nelle sezioni "Sviluppo multipiattaforma" e "Sviluppo nativo" si ritiene quindi più conveniente lo sviluppo di un'applicazione multi piattaforma. Nello specifico si consiglia l'utilizzo di framework quali React Native e Xamarin con la preferenza di quest'ultimo.

Breve considerazione sulla comunicazione tra IW e ITF

L'architettura di MonoKee presenta un componente chiamato ITF il quale comunica con l'IW. Lo sviluppo dell'IW come applicazione mobile potrebbe rappresentare un problema in termini di Fiducia tra l'ITF e l'IW, infatti l'unica soluzione attualmente presente per operare direttamente sulla blockchain da mobile è Status. Status è un progetto che propone una serie di API che permettono di sviluppare un'applicazione mobile nativa operante direttamente su blockchain senza la necessità di possedere un intero nodo. Attualmente però non sembrerebbe rappresentare una soluzione utilizzabile in quanto troppo acerba e poco utilizzata. L'unica opzione rimanente risulta quella di gestire le comunicazioni tra ITF e IW tramite API REST. Quest'ultima soluzione renderebbe l'IW totalmente incapace di verificare i dati provenienti dall'ITF rendendo così assente l'aspetto fiducia.

Questo renderebbe il componente all'interno dell'ITF che gestisce le comunicazioni REST non distribuito, quindi di fatto snaturerebbe la scelta di utilizzare una blockchain come base dell'ITF.

Un'eventuale futuro uso di Status potrebbe comportare forti modifiche all'applicazione.

Breve considerazione sullo sviluppo mobile

Dall'analisi del dominio applicativo emerge come un'applicazione di tipo mobile sia la scelta più adatta. Questa scelta è basato sulla tipologia di utenti e sul tipico uso ipotizzato per l'applicazione. Tuttavia come precedentemente detto l'obbligatorietà di comunicare con l'ITF tramite API REST snaturerebbe il concetto stesso di blockchain, in quanto l'IW vedrebbe il componente REST come fonte centralizzata e non verificabile delle informazioni. Per queste ragioni si vuole far notare come un'applicazione Desktop risulterebbe notevolmente più appropriata dal punto di vista tecnologico, ma fuori contesto dal punto di vista dell'uso previsto.

Al fine di effettuare una scelta finale bisogna tenere sempre in menti questi due fattori e considerare cosa si vuole ottenere. Ritengo che l'utente dell'IW non sia in grado di apprezzare questo concetto di fiducia e che potrebbe apprezzare maggiormente il fatto che l'applicativo sia mobile.

Motivazioni

Aspetti positivi

A seguito dell'analisi sopra proposta sono stati individuati i seguenti aspetti positivi:

- lo sviluppo di un'applicazione mobile Android e iOS porterebbe MonoKee alla portata della quasi totalità dei possibili utenti;
- uno sviluppo con un framework multi piattaforma abbatterebbe i costi di produzione dell'applicazione, pur garantendo risultati accettabili;
- i framework multi piattaforma portati in esame (Xamarin e React Native) sono ampiamente utilizzati e supportati da grandi aziende IT. Questo garantisce un elevato grado di affidabilità e una ampia documentazione;
- un eventuale uso di Xamarin potrebbe facilitare una successiva implementazione di un'applicazione Windows Phone.
- seppur MonoKee utilizza una soluzione basata su blockchain, l'IW non risulta colpito da questa ulteriore complessità.

Fattori di rischio

Invece per quanto riguarda i fattori di rischio, sono emersi i seguenti punti:

- la comunicazione tra IW e ITF dovrebbe avvenire attraverso chiamate REST. Questo renderebbe l'IW totalmente incapace di verificare i dati provenienti dall'IW;
- un'applicazione mobile di questo tipo per l'IW potrebbe non essere considerata come strumento di IAM distribuito, ma potrebbe essere vista come centralizzata.

Conclusioni

Da questo primo studio di fattibilità emerge come, da un punto di vista dell'utente, lo sviluppo di un'applicazione mobile sia maggiormente adatto. Invece, da un punto di vista tecnologico, risulta come ci siano delle problematiche inerenti la verificabilità dei dati. Riguardo questo si ritiene che lo sviluppo di un applicativo Desktop risulterebbe più adatto, ma molto probabilmente mal visto dalla maggioranza degli utenti finali.

Per quanto detto si conclude ribadendo la fattibilità del progetto come applicazione mobile sviluppata con un framework multi piattaforma. Per la scelta del framework si consiglia Xamarin.