11/06/2018

IW-Studio di fattibilità

# 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 |
|  |  | Simone Ballarin |

# Scopo del documento

Il seguente documento *Studio di fattibilita’* 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.

# 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 (quali ulteriori layer di sicurezza, i.e. 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

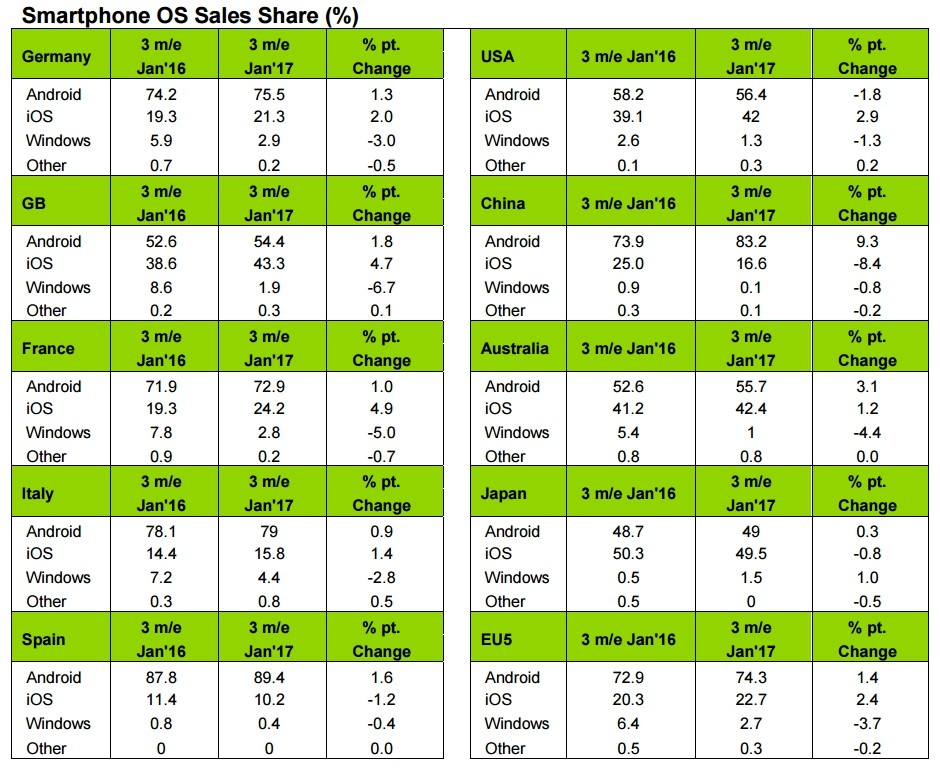
L’applicativo IW dovrà essere usato in un contesto prevalentemente lavorativo. Non si escludono però ulteriori applicazione future in ambito Consumer. In ogni caso il software dovrà poter essere usato da utenti senza specifiche conoscenze informatiche e con minimo training tecnologico. Dato il contesto 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

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. 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.



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 delle varie piattaforme.

**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. Il loro utilizzo presenta diversi vantaggi:

* 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.

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 di 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.

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

# Studio dei costi