**MonoKee – Service Provider, analisi e specifica dei requisiti**

Autore: Simone Ballarin

Data: 20/06/18

Destinatari: Athesys

# Diario delle modifiche

|  |  |  |
| --- | --- | --- |
| Data | Descrizione | Autore |
| 20/06/2018 | Creazione documento. Stesura dei capitoli Scopo del documento, Descrizione prodotto. | Simone Ballarin |
| 21/06/2018 | Inseriti capitoli Use Cases, Diagrammi di sequenza, Analisi requisiti, tracciamento requisito-fonti, tracciamento fonte-requisiti, Riepilogo requisiti, Glossario acronimi. | Simone Ballarin |

# Scopo del documento

Questo documento ha lo scopo di fornire una definizione ufficiale dei requisiti individua per la creazione del prodotto Service Provider (SP). Il documento è rivolto agli stakeholders del sistema, al Responsabile, al Progettista, ai Programmatori, Verificatori e futuri manutentori del sistema. Più in particolare il presente documento si prefigge di:

* individuare le fonti per la deduzione dei requisiti;
* dedurre i requisiti dalle fonti;
* descrivere i requisiti individuati;
* catalogare i requisiti individuati;
* prioritizzare i requisiti individuati;

Inoltre ogni modifica, o aggiunta, al presente documento deve essere discussa in riunione e approvata dal Responsabile Athesys Sara Meneghetti e dagli stakeholders.

# Scopo del prodotto

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:

* Identity Wallet (IW)
* Service Provider (SP)
* Identity Trust Fabric (ITF)
* 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 autonomamente tramite l’IW, mandare i propri dati 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 SP, più dettagliatamente, dovrà assolvere ai seguenti compito:

nell’ambito della ricezione dei dati da un Identity Wallet (IW) deve:

* ricevere da parte dell’IW la chiave pubblica (o l’hash di questa);
* ricevere un riferimento alla locazione dell’hash della chiave pubblica all’interno dell’ITF;
* ricevere altre informazioni necessarie da parte dell’IW con relativo riferimento all’interno dell’ITF;
* gestire il trasferimento dei dati tramite codice QR.

Nell’ambito della verifica dei dati provenienti dall’IW deve:

* usare la chiave pubblica dell’IW e il riferimento per verificare l’identità e le varie altre informazione passate dall’Identity Wallet;
* generare e comparare gli hash dei valori ottenuti con quelli presenti nell’ITF;
* verificare che l’identità e le altre informazioni ottenute siamo sufficienti a garantire l’accesso al servizio.

Nell’ambito dell’accesso il SP deve:

* a seguito della verifica comunica il risultato all’organizzazione che fornisce il servizio, in modo tale da garantire l’accesso all’utente dell’IW.

# Riferimenti informativi

1. Ingegneria del Software decima edizione, Sommerville capitolo 4;
2. Blockchain: The Dawn of Decentralized Identity (G00303143), Homan Farahmand per Gartner.
3. MonoKee – WI Studio di fattibilità

# Descrizione generale

## Concept

Si intende sviluppare il componente Service come un’applicazione server che opera in collaborazione con l’ITF al fine di la veridicità dei dati provenienti dall’IW e quindi garantire o meno l’accesso al servizio. Questa connessione deve avvenire tramite il protocollo JSON-RFC descritto nello Yellow Paper. Questi dati vengono presentati sotto forma di codice QR. L’applicazione deve inoltre avere un’interfaccia web accessibile tramite Internet dalla quale il personale del fornitore può configurare il servizio. Si fa notare come con SP non si intenda il servizio o il fornitore dei servizi, ma semplicemente il componente MonoKee che ha lo scopo di interfacciarsi con questi.

## Analisi del dominio

Nel seguente paragrafo verranno fornite le informazioni riguardanti il significato di alcune terminologie riguardanti il dominio rappresentato dalla pratica dell’Identity Management Access (IAM).

**Identity Access Management (IAM)**: è una pratica che permette di gestire gli utenti e le autorizzazioni utente all'interno di un sistema informativo più o meno complesso. Con una soluzione di IAM, è possibile gestire centralmente gli utenti, le credenziali di sicurezza come chiavi di accesso e le autorizzazioni che controllano che gli utenti possono accedere a tutte e solo le risorse di loro competenza.

**Personally, Identifiable Information (PII):** è un’informazione inserita dall’utente attraverso l’Identity Wallet (IW) e poi associata alla propria identità all’interno all’Identity Trust Fabric (ITF). Questa può essere di due tipi certificata o non certificata da parte di un Trusted Third Party (TTP). Un esempio può essere le credenziali di accesso ad un servizio, quali username e password. Le PII sono presenti in chiaro all’interno dell’Identity Wallet (IW) e in forma di hash nella Identity Trust Fabric (ITF).

**Identity Wallet (IW)**: è il componente in analisi in questo documento, esso rappresenta dello strumento lato utente con cui è possibile gestire le informazioni relative ad un’identità digitale.

**Trusted Third Party (TTP):** è un componente dell’infrastruttura che, in qualità di organizzazione affidabile, opera come certificatore delle informazioni provenienti dall’ITF**.** La certificazione viene memorizzata nella Identity Trust Fabric (ITF).

**Service Provider (SP)**: è un componente dell’infrastruttura che si colloca tra il reale fornitore del servizio e il nostro sistema MonoKee. Questo ha il compito di verificare le informazioni provenienti dall’IW e poi comunicare l’esito al reale fornitore.

**Real Service Provider (RSP)**: è il reale fornitore del servizio. Si tratta di un’organizzazione convenzionata e che usufruisce del servizio di IAM MonoKee.

## Specifiche in Linguaggio Naturale

Il linguaggio naturale ha un’enorme potenza espressiva ma, essendo inerentemente ambiguo, può portare ad incomprensioni. È quindi necessario limitarne l’utilizzo e standardizzarlo, in modo da ridurre al minimo le possibili ambiguità. È comunque fondamentale evitare di utilizzare espressioni e acronimi che possano essere fraintendibili dagli stakeholders, a tal proposito in fondo al documento è presente una lista degli acronimi utilizzato.

## Specifiche in Linguaggio Strutturato

Il linguaggio strutturato mantiene gran parte dell’espressività del linguaggio naturale, fornendo però uno standard schematico che permette l’uniformità della descrizione dei vari requisiti. Sebbene l’utilizzo di un linguaggio strutturato permetta di organizzare i requisiti in modo più ordinato e comprensibile, talvolta la ridotta espressività rende difficile la definizione di requisiti complessi. A tal proposito è possibile integrare la specifica in linguaggio strutturato con una descrizione in linguaggio naturale.

## Specifiche in Linguaggio UML Use Case

Per la definizione dei diagrammi UML dei casi d’uso, viene utilizzato lo standard UML 2.0. Nei diagrammi dei casi d’uso vengono mostrati gli attori coinvolti in un’interazione con il sistema in modo schematico, indicando i nomi delle parti coinvolte. Eventuali informazioni aggiuntive possono essere espresse testualmente.

# Use case

## Descrizione attori

I tipi di utente che andranno ad interagire direttamente con il sistema si dividono in due categorie:

* Servizio convenzionato;
* Utente IW.

Tra gli attori precedentemente citati non è però prevista alcuna funzionalità in comune e non emerge quindi la necessità di avere una gerarchia. Di seguito è proposta una visualizzazione grafica di quanto detto:



Non sono stati individuati, invece, attori secondari che partecipano al sistema.

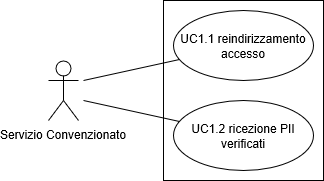
### Attori principali

* **Servizio convenzionato**: l’attore servizio convenzionato è quello che nell’analisi del dominio è stato definito come Real Service Provider (RSP). Si tratta del fornitore reale del servizio.
* **Utente IW**: l’attore utente IW è una persona fisica che utilizza la nostra applicazione mobile al fine di operare l’accesso ad un servizio convenzionato in MonoKee.

### Attori secondari

Non sono presenti attori secondari.

## UC1 – Azioni servizio convenzionato



|  |  |
| --- | --- |
| **Descrizione** | Il servizio convenzionato può reindirizzare verso al sistema una richiesta di accesso e ricevere i dati di accesso PII verificati. |
| **Attore primario** | Servizio convenzionato |
| **Attore secondario** | Nessuno |
| **Precondizioni** | Il servizio convenzionato ha richiesto una richiesta di accesso e l’utente che l’ha effettuata a richiesto l’accesso tramite il nostro servizio. |
| **Postcondizioni** | Il servizio ha eseguito le azioni che desiderava compiere in relazione alle sue possibilità |
| **Scenario principale** | * UC1.1 Reindirizzamento accesso * UC1.2 Ricezione PII verificati |
| **Scenari alternativi** | Nessuno |

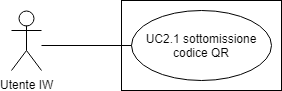
## UC1.1 – Reindirizzamento accesso

|  |  |
| --- | --- |
| **Descrizione** | Un servizio convenzionato può inoltrare al sistema richieste di accesso |
| **Attore primario** | Servizio convenzionato |
| **Attore secondario** | Nessuno |
| **Precondizioni** | Il servizio convenzionato ha ricevuto una richiesta di accesso |
| **Postcondizioni** | Il sistema ha ricevuto la richiesta di accesso e procederà ad eseguirla |
| **Scenario principale** | Il servizio convenzionato inoltra la richiesta di accesso ed il sistema la immagazzina per prendersene carico |
| **Scenari alternativi** | Nessuno |

## UC1.2 – Ricezione PII verificate

|  |  |
| --- | --- |
| **Descrizione** | Il sistema deve, in risposta ad un inoltro di richiesta di accesso, inviare al servizio convenzionato l’esito della verifica e, in caso di successo, le PII in chiaro necessarie per effettuare l’oggetto |
| **Attore primario** | Servizio convenzionato |
| **Attore secondario** | Nessuno |
| **Precondizioni** | Il servizio convenzionato ha precedentemente inoltrato una richiesta di accesso al sistema |
| **Postcondizioni** | Il sistema ha ricevuto l’esito della verificata ed in caso le PII necessarie per l’accesso in chiaro |
| **Scenario principale** | Il sistema riceve l’esito della verificata ed in caso le PII necessarie per l’accesso in chiaro |
| **Scenari alternativi** | Nessuno |

UC2 – Azioni utente IW



|  |  |
| --- | --- |
| **Descrizione** | L’utente IW può eseguire le operazioni per l’accesso |
| **Attore primario** | Utente IW |
| **Attore secondario** | MonoKee |
| **Precondizioni** | Nessuna |
| **Postcondizioni** | L’utente ha eseguito le azioni che desiderava compiere in relazione alla condizione. |
| **Scenario principale** | * UC2.1 Sottomissione codice QR |
| **Scenari alternativi** | Nessuno |

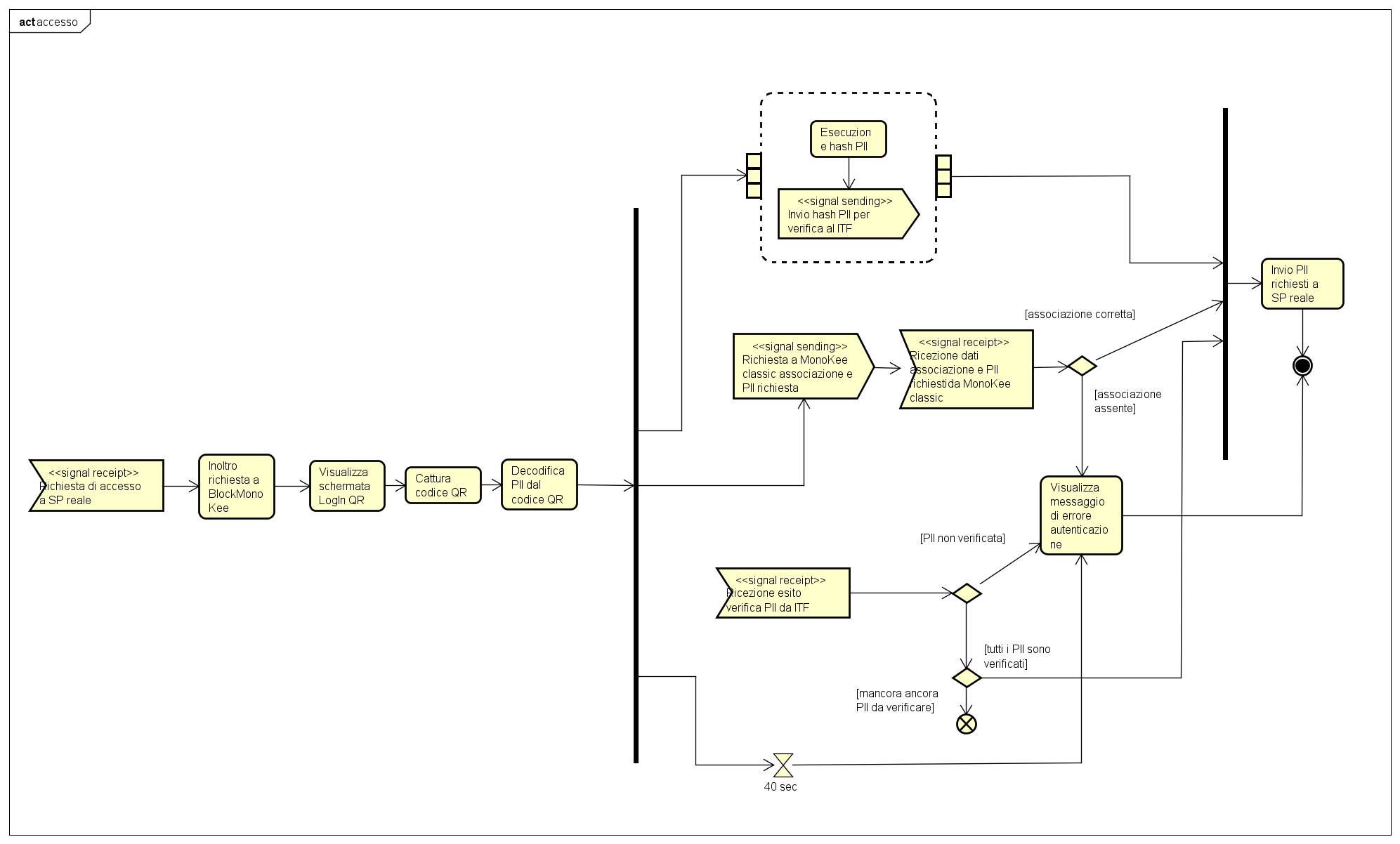
## UC2.1 Sottomissione codice QR

|  |  |
| --- | --- |
| **Descrizione** | L’utente IW può eseguire l’operazione di sottomissione di codice QR |
| **Attore primario** | Utente IW |
| **Attore secondario** | Nessuno |
| **Precondizioni** | Il servizio convenzionato ha inoltrato l’utente al nostro sistema di accesso e l’utente ha generato il codice QR dall’IW |
| **Postcondizioni** | Il sistema ha catturato il codice QR |
| **Scenario principale** | Il sistema accende la webcam del computer e cattura il codice QR che presenta l’utente. |
| **Scenari alternativi** | Nessuno |

# Diagramma delle attività

Al fine di descrivere il corretto flusso che il componente deve utilizzare viene utilizzato un diagramma di attività. L’unica operazione che il componente dovrà gestire al fine di garantire gli scopi che si prefigge è la gestione di un inoltro accesso da parte di un RSP.

## DA1 – Gestione richiesta accesso



Ora si procederà ad una breve descrizione del diagramma sopra proposto.

Il flusso parte con l’arrivo di una richiesta di accesso da parte di un RSP, questo le seguenti operazioni in maniera sequenziale:

* inoltro della richiesta verso il nostro sistema SP;
* il sistema visualizza una schermata dove richiede la sottomissione del codice QR;
* cattura del codice QR;
* decodifica le PII in chiaro dal codice QR.

Poi il flusso si divide in tre operazioni differenti:

* la prima con il compito di inviare una richiesta di verifica all’ITF per ogni PII decodificato dal codice QR;
* la seconda con il compito di interfacciarsi al sistema MonoKee per ottenere l’associazione tra account e servizio e la lista dei PII necessari;
* il terzo con il compito di aspettare gli esiti delle verifiche dall’ITF.

In caso l’associazione sia presente e corretta e tutte le PII necessarie sono verificate allora si procede con la comunicazione dei dati verso il reale fornitore del servizio.

In caso, o si riceva un esito negativo di una PII necessaria, o non tutte quelle necessarie siano state presentate tramite il codice QR, si procede alla comunicazione dell’errore di autentificazione ed alla conclusione del flusso.

In ogni caso se dopo 40 secondi dalla decodifica del codice QR il sistema non ha effettuato l’accesso viene visualizzato un messaggio di errore ed il flusso termina.

# Analisi requisiti

## Fonti

Per la deduzione dei requisiti utente e di sistema, che verranno presentati nelle sezioni a seguire, sono stati usate le seguenti fonti:

* studio Gartner in nota 2;
* documento MonoKee – SP Studio di fattibilità;
* Use Case presentati nella sezione Use Case;
* Il diagramma di attività presentato nella sezione Diagrammi di attività.

La struttura e le convenzioni usate sono ispirate dal capitolo 4 del libro “Ingegneria del Software” in nota 1. In seguito vengono riportate le categorie che vengono usate per la catalogazione:

* F: requisito funzionale;
* V: requisito di vincolo;
* Q: requisito di qualità.

Per l’attribuzione della priorità viene usata la tecnica MoSCoW, quindi gli indici usati sono i seguenti:

* M: must;
* S: should;
* C: could;
* W: will.

## Requisiti Funzionali

|  |  |  |
| --- | --- | --- |
| Codice | Descrizione | Fonte |
| R[F][M]0001 | Il sistema deve permettere ad un servizio convenzionato di inoltrare le richieste di accesso ricevute al nostro sistema | UC1, UC1.1, DA1 |
| R[F][M]0002 | Il sistema deve inviare l’esito della verifica al reale fornitore del servizio | UC1, UC1.2, DA1 |
| R[F][M]0003 | Il sistema deve inviare i PII in chiaro in caso di verifica positiva al reale fornitore del servizio | UC1, UC1.2, DA1 |
| R[F][M]0004 | Il sistema deve permettere ad un utente dell’IW di sottomettere un codice QR generato dall’applicazione IW. | UC2, UC2.1, DA1 |
| R[F][M]0005 | Il sistema deve visualizzare una schermata di accesso | DA1 |
| R[F][M]0006 | Il sistema deve catturare nella schermata di accesso il codice QR attraverso l’uso della webcam | DA1 |
| R[F][M]0007 | Il sistema deve essere in grado di decodificare le informazioni contenute in un codice QR | DA1 |
| R[F][M]0008 | Il sistema deve essere in grado di fare l’hash di una PII | DA1 |
| R[F][M]0009 | Il sistema deve essere in grado di inviare una richiesta di verifica per un particolare PII | DA1 |
| R[F][M]0010 | Il sistema deve essere in grado di eseguire l’operazione di hash e invio richiesta verifica per ogni PII presenta in un codice QR | DA1 |
| R[F][M]0011 | Il sistema deve inviare una richiesta dell’associazione utente-servizio a Monokee classico | DA1 |
| R[F][M]0012 | Il sistema deve essere in grado di ricevere le informazioni richiesta dell’associazione utente servizio da Monokee classico | DA1 |
| R[F][M] 0013 | Il sistema deve visualizzare un messaggio di errore in caso cui l’associazione utente-servizio non sia presente per il servizio richiesto | DA1 |
| R[F][M] 0014 | Il sistema deve essere in grado di ricevere l’esito della verifica di un singolo PII proveniente dall’ITF | DA1 |
| R[F][M] 0015 | Il sistema deve visualizzare un messaggio di errore in caso cui la verifica di una PII richiesta sia negativa | DA1 |
| R[F][M] 0016 | Il sistema deve visualizzare un messaggio di errore in caso non tutte le verifiche delle PII necessarie tornino in 40 secondi. | DA1 |
| R[F][M] 0017 | Il sistema deve in caso di presenza dell’associazione e del ritorno positivo di tutte le verifiche necessarie inviare i dati PII al SP reale | DA1 |

## Requisiti di vincolo

|  |  |  |
| --- | --- | --- |
| R[V][M] 0018 | Il sistema deve offrire le proprie funzionalità come applicazione server centralizzata | MonoKee - SP Studio di fattibilità |
| R[V][M] 0019 | Il sistema è implementato tramite in linguaggi .NET | MonoKee - SP Studio di fattibilità |
| R[V][M] 0020 | Il progetto prevede almeno i seguenti quattro ambienti di sviluppo: Local, Test, Staging, Production | MonoKee - SP Studio di fattibilità |
| R[V][M] 0021 | Il prodotto è sviluppato utilizzando uno strumento di linting | MonoKee - SP Studio di fattibilità |
| R[V][C] 0022 | Il sistema deve comunicare con la rete blockchain tramite un client Ethereum. | MonoKee - SP Studio di fattibilità |

## Requisiti di qualità

|  |  |  |
| --- | --- | --- |
| R[Q][S] 0023 | Il progetto prevede un ragionevole set di test di unità e di test di integrazione | Nota 1 |
| R[Q][S] 0024 | I test possono essere eseguiti localmente o come parte di integrazione continua | Nota 1 |
| R[Q][S] 0025 | Il sistema solo alla fine sarà testato in un server di prova | Nota 1 |
| R[Q][S] 0026 | Il codice sorgente del prodotto e la documentazione necessaria per l’utilizzo sono versionati in repository pubblici usando GitHub, BitBucket o GitLab | Nota 1 |
| R[Q][C] 0027 | Lo sviluppo si eseguirà utilizzano un approccio incrementale | MonoKee - SP Studio di fattibilità |
| R[Q][C]0028 | Il sistema potrebbe essere testato con l’ITF migrato nella rete di prova Ropsten | MonoKee - ITF Studio tecnologico |

## Tracciamento requisito – fonti

|  |  |
| --- | --- |
| Codice | Fonte |
| R[F][M]0001 | UC1, UC1.1, DA1 |
| R[F][M]0002 | UC1, UC1.2, DA1 |
| R[F][M]0003 | UC1, UC1.2, DA1 |
| R[F][M]0004 | UC2, UC2.1, DA1 |
| R[F][M]0005 | DA1 |
| R[F][M]0006 | DA1 |
| R[F][M]0007 | DA1 |
| R[F][M]0008 | DA1 |
| R[F][M]0009 | DA1 |
| R[F][M]0010 | DA1 |
| R[F][M]0011 | DA1 |
| R[F][M]0012 | DA1 |
| R[F][M] 0013 | DA1 |
| R[F][M] 0014 | DA1 |
| R[F][M] 0015 | DA1 |
| R[F][M] 0016 | DA1 |
| R[F][M] 0017 | DA1 |
| R[V][M] 0018 | MonoKee - SP Studio di fattibilità |
| R[V][M] 0019 | MonoKee - SP Studio di fattibilità |
| R[V][M] 0020 | MonoKee - SP Studio di fattibilità |
| R[V][M] 0021 | MonoKee - SP Studio di fattibilità |
| R[V][C] 0022 | MonoKee - SP Studio di fattibilità |
| R[Q][S] 0023 | Nota 1 |
| R[Q][S] 0024 | Nota 1 |
| R[Q][S] 0025 | Nota 1 |
| R[Q][S] 0026 | Nota 1 |
| R[Q][C] 0027 | MonoKee - SP Studio di fattibilità |
| R[Q][C]0028 | MonoKee - ITF Studio tecnologico |

## Tracciamento fonte – requisiti

|  |  |
| --- | --- |
| Fonte | Codice |
| UC1 | R[F][M]0001  R[F][M]0002  [F][M]0003 |
| UC2 | R[F][M]0004 |
| UC1.1 | R[F][M]0001 |
| UC1.2 | R[F][M]0002  R[F][M]0003 |
| UC2.1 | R[F][M]0004 |
| DA1 | R[F][M]0001  R[F][M]0002  R[F][M]0003  R[F][M]0004  R[F][M]0005  R[F][M]0006  R[F][M]0007  R[F][M]0008  R[F][M]0009  R[F][M]0010  R[F][M]0011  R[F][M]0012  R[F][M]0013  R[F][M]0014  R[F][M]0015  R[F][M]0016  R[F][M]0017 |
| MonoKee - SP Studio di fattibilità | R[V][M] 0018  R[V][M] 0019  R[V][M] 0020  R[V][M] 0021  R[V][M] 0022  R[Q][C] 0027 |
| Nota 1 | R[Q][S] 0023  R[Q][S] 0024  R[Q][S] 0025  R[Q][S] 0026 |
| MonoKee - ITF Studio tecnologico | R[Q][C]0028 |

## Riepilogo requisiti

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Categoria** | **Must** | **Should** | **Could** | **Will** |
| **Funzionale** | 17 | 0 | 0 | 0 |
| **Di vincolo** | 4 | 0 | 1 | 0 |
| **Di qualità** | 0 | 4 | 2 | 0 |

# Lista acronimi

|  |  |
| --- | --- |
| **IW** | Identity Wallet |
| **ITC** | Identity Trust Fabric |
| **TTP** | Trusted Third Party |
| **SP** | Service Provider |
| **IAM** | Identity Access Management |
| **UC** | Use Case |
| **PII** | Personally, identifiable information |
| **RSP** | Real Service Provider |