



[projectorigin2023@gmail.com](mailto:projectorigin2023@gmail.com)

## *Verbale* esterno del 26 maggio 2023

<b>Versione</b>	0.1.0
<b>Responsabile</b>	Corbu Teodor Mihail
<b>Redattori</b>	Ibra Elton
<b>Verificatori</b>	Lotto Riccardo
<b>Uso</b>	Esterno
<b>Destinatari</b>	<i>Project Origin</i> Prof. Vardanega Tullio Prof. Cardin Riccardo

### **Descrizione**

*Verbale* riguardante il meeting tenuto il 26 maggio 2023 con l'azienda *InfoCert S.p.A*

## Registro delle modifiche

Vers.	Data	Autore	Ruolo	Descrizione
0.1.0	2023-05-29	Lotto Riccardo	Verificatore	Verifica documento
0.0.1	2023-05-28	Ibra Elton Andreetto Alessio	Analista	Redazione documento

## Indice

<b>1</b>	<b>Informazioni generali</b>	<b>3</b>
1.1	Dettagli sull'incontro . . . . .	3
<b>2</b>	<b>Ordine del giorno</b>	<b>4</b>
2.1	Libreria . . . . .	4
2.2	Entità . . . . .	4
2.3	Database . . . . .	4
2.4	Framework . . . . .	5
2.5	Credenziali . . . . .	5
2.6	Link . . . . .	5

# 1 Informazioni generali

## 1.1 Dettagli sull'incontro

- **Luogo:** Incontro telematico tramite piattaforma *Microsoft Teams*;
- **Data:** 26-05-2023;
- **Ora di inizio:** 11:00;
- **Ora di fine:** 12:00;
- **Presenze:**
  - *InfoCert:*
    - \* Boldrin Luca
    - \* Manzi Paolo
  - *Project Origin:*
    - \* Andreetto Alessio
    - \* Beschin Michele
    - \* Bobirica Andrei Cristian
    - \* Corbu Teodor Mihail
    - \* Ibra Elton
    - \* Lotto Riccardo

## 2 Ordine del giorno

1. Libreria
2. Entità
3. Database
4. Framework
5. Credenziali
6. Link

### 2.1 Libreria

Durante le prime fasi di consultazione con l'azienda, è stata sollevata una domanda riguardante l'utilizzo delle librerie REST API. La scelta di utilizzare REST API come metodologia di interazione per la nostra web app è considerata ragionevole. Trattandosi di una libreria ampiamente adottata, inclusa l'implementazione da parte di Infocert in progetti validi, si può affermare che si tratta di una soluzione matura e affidabile. Approviamo quindi l'utilizzo di REST API per la nostra applicazione, trasformando il server in un'entità che espone un'interfaccia basata su questa tecnologia.

Per quanto riguarda la libreria *Walt.id* risponde in modo completo allo standard OpenID.

Pertanto, abbiamo deciso di utilizzarla senza la necessità di implementare una soluzione personalizzata. Non c'è nulla di sbagliato in questa scelta, in quanto ha valore didattico: non abbiamo l'obiettivo di sviluppare tutto da zero. Infatti, *InfoCert* è più interessata all'implementazione delle parti più rilevanti, piuttosto che alle componenti già disponibili come questa libreria. REST API e la libreria *Walt.id* sono state approvate.

### 2.2 Entità

Durante la riunione è emersa la discussione sulla scelta di come implementare la libreria *Walt.id*, tra creare tre entità distinte all'interno dello stesso container o effettuare un unico deploy del server con gli attori che comunicano all'interno dello stesso container. Tale scelta implica una valutazione del concetto di decentralizzazione. Secondo il parere di *InfoCert*, la soluzione più logica è quella di avere tre agenti diversi con tre endpoint diversi, utilizzando *Docker Compose* per orchestrare l'ambiente di sviluppo.

Questo approccio offre una maggiore chiarezza e coerenza nella rappresentazione delle diverse entità e dei relativi endpoint.

Sebbene sia possibile emulare il flusso di comunicazione con un unico server, si ritiene che non sia conveniente adottare questa soluzione. L'approccio di avere entità distinte sembra essere più appropriato in termini di gestione, manutenzione e comprensione del sistema.

### 2.3 Database

Durante la discussione, è stata considerata la possibilità di utilizzare due web app separate per la parte del Issuer, una per la gestione delle richieste e l'altra per fornire le risposte. Inoltre, si è discusso dell'utilizzo di un database comune tra la web app BOC (Back-office component) e IUIC (Issuer User Interface Component).

Dal punto di vista concettuale, l'idea di avere un database comune sembra essere adeguata. Tuttavia, dal punto di vista pratico, potrebbe essere più conveniente e efficiente collassare tutto in un unico front-end, semplificando così l'architettura e semplificando la gestione dei dati. È importante notare che, nonostante l'uso di un unico database, le credenziali e i tipi di accesso tra la web app BOC e IUIC possono essere diversi. Ciò significa che le autorizzazioni e le restrizioni di accesso devono essere attentamente gestite per garantire la sicurezza e la privacy dei dati. Dato che il database non è una componente strategica, si consiglia di optare per la soluzione più semplice, utilizzando un DBMS SQL.

## 2.4 Framework

Durante la discussione, è stato valutato il framework da utilizzare e si è giunti alla conclusione che React rappresenta una buona scelta intermedia, ed è stata approvata come tale. Si è deciso anche di evitare l'utilizzo di TypeScript. Inoltre, è stato menzionato l'utilizzo di React Native come una soluzione vantaggiosa, in quanto permette di simulare l'applicazione su diverse piattaforme, è stato suggerito di utilizzarlo per la componente del wallet.

Il gruppo ha considerato l'adozione di React Native per lo sviluppo dell'applicazione mobile wallet proposta dall'azienda.

## 2.5 Credenziali

Durante le discussioni, si è affrontato il tema del salvataggio del wallet, ovvero delle credenziali, in locale. Sono state prese in considerazione due opzioni:

- memorizzare le credenziali nel local storage del browser all'interno della web app;
- salvare le credenziali nel cloud utilizzando un server fornito dal sistema.

Considerando che la web app in realtà è una web view, è stato ritenuto più realistico utilizzare un server per memorizzare le credenziali anziché affidarsi esclusivamente al local storage del browser. In questo scenario, si è proposto di utilizzare *Walt.id* come server anziché come libreria, sfruttando i suoi servizi REST API e di memorizzazione delle credenziali. Tuttavia, è stata riconosciuta la necessità di considerare attentamente i tempi e le risorse necessarie per implementare questa soluzione.

Sebbene sia importante mantenere la proprietà delle credenziali, l'archiviazione delle stesse su un server può risultare utile in caso di smarrimento o problemi vari. Inoltre, è importante valutare le difficoltà tecniche associate, come il backup, il ripristino, l'importazione e l'esportazione dei dati. Pertanto, è stata suggerita l'adozione di un'architettura "ibrida" che combinasse entrambe le opzioni. Sebbene questa soluzione non sia perfetta, offre un compromesso più utilizzabile, consentendo il salvataggio delle credenziali nel server.

## 2.6 Link

Durante la riunione, l'azienda ha suggerito di guardare la demo di *Walt.id*, in cui si può simulare holder, issuer e verifier.