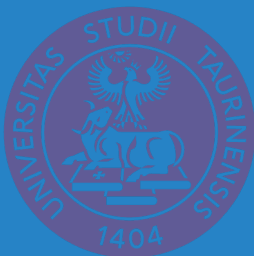


INTEGRAZIONE TRA SERVIZI DI AUTENTICAZIONE E BLOCKCHAIN: COME GESTIRE WALLET CRITTOGRAFICI

Università degli Studi di Torino
Corso di Laurea in Informatica
Anno Accademico 2021/2022

Tesista: Rondinella Raoul

Relatore: Schifanella Claudio



ALTEN

Alten è leader europeo nel settore della consulenza per le tecnologie avanzate in ambito ingegneristico e ICT, è quotata sulla Borsa di Parigi.

Alten ha creato la piattaforma **Andromeda** che estende le funzionalità della blockchain Ethereum, permettendo l'integrazione della tecnologia in soluzioni aziendali, facilitando l'installazione, la gestione e il monitoraggio.



REQUISITI PROGETTO

PROBLEMA

In un sistema Blockchain tradizionale, il wallet è memorizzato all'interno del dispositivo dell'utente.

Inoltre, richiede un'interazione diretta dell'utente quando è necessario compiere una transazione.

Come può un app interagire con la blockchain, interfacciarsi con l'utente comune e non essere troppo complessa da usare?



SOLUZIONE

Creare un sistema di autenticazione e gestione del wallet basato su metafore che l'utente comune conosce ed usa agevolmente



LA BLOCKCHAIN

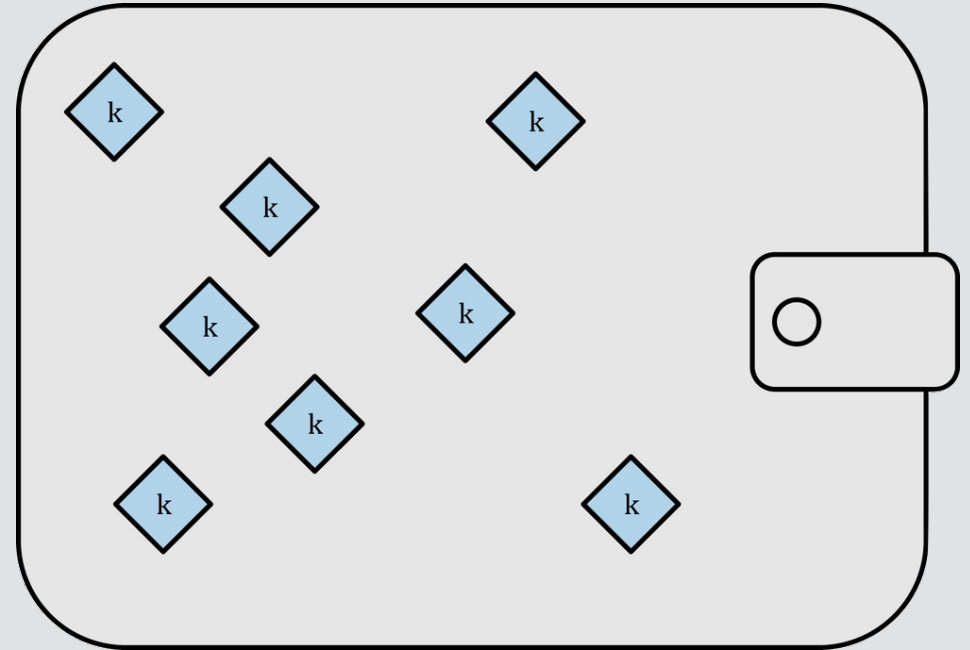
WALLET

Per firmare e verificare le transazioni è necessario possedere una chiave privata ed una corrispondente chiave pubblica

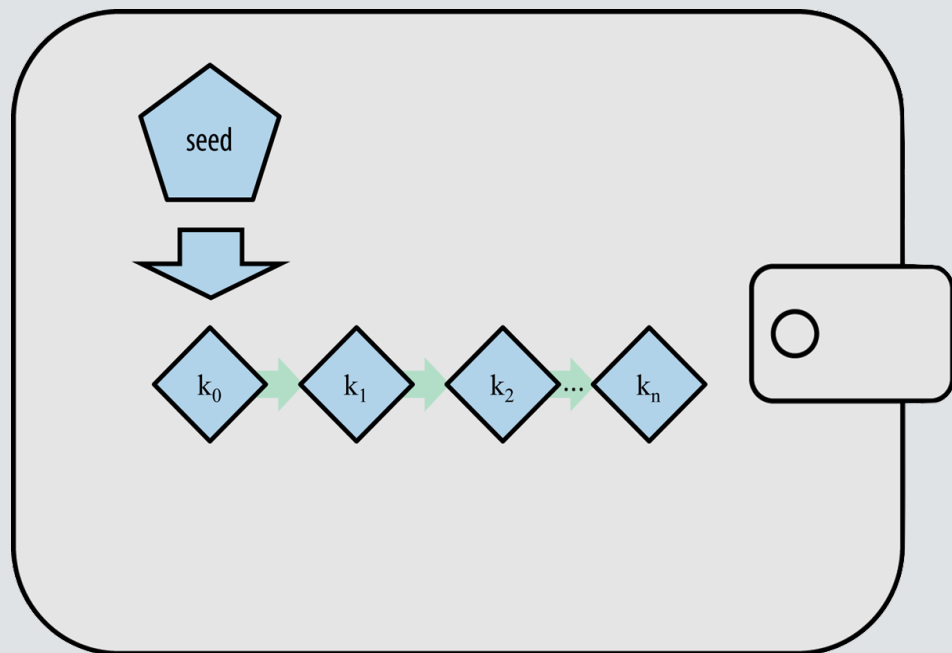


NON DETERMINISTICO

- Ogni account del wallet è «a sé»
- Generazione completamente casuale



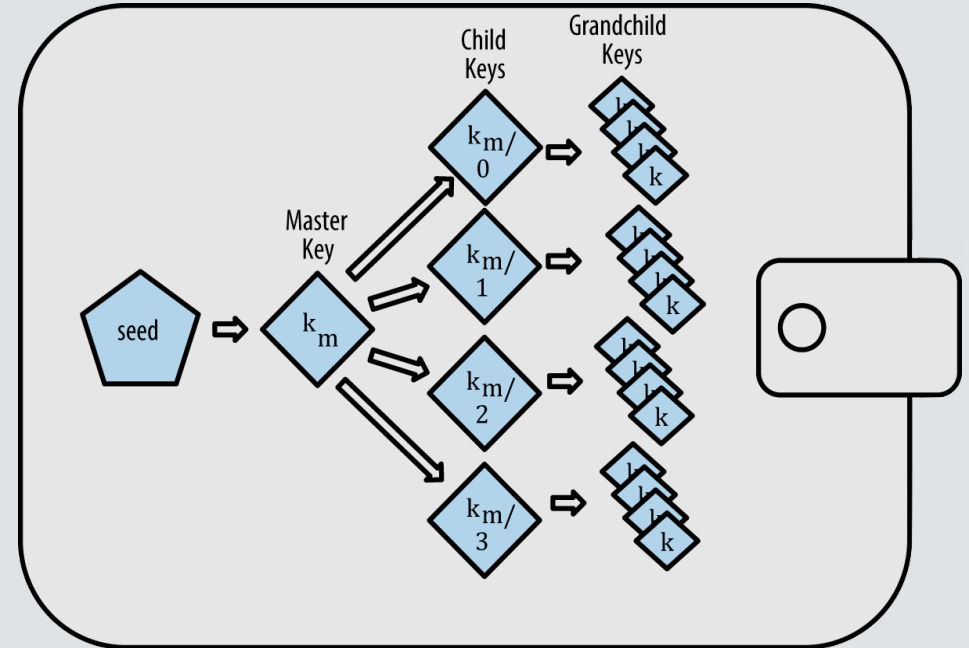
DETERMINISTICO



- Generazione dei wallet a partire da un «seed»
- Wallet generati «a cascata»

GERARCHICO

- Miglioramento del wallet deterministico
- Possibilità di organizzare account



OAuth2
OIDC

COSA SONO

OAuth2:

- Moderno sistema di autorizzazione
- Permette accesso con username e password

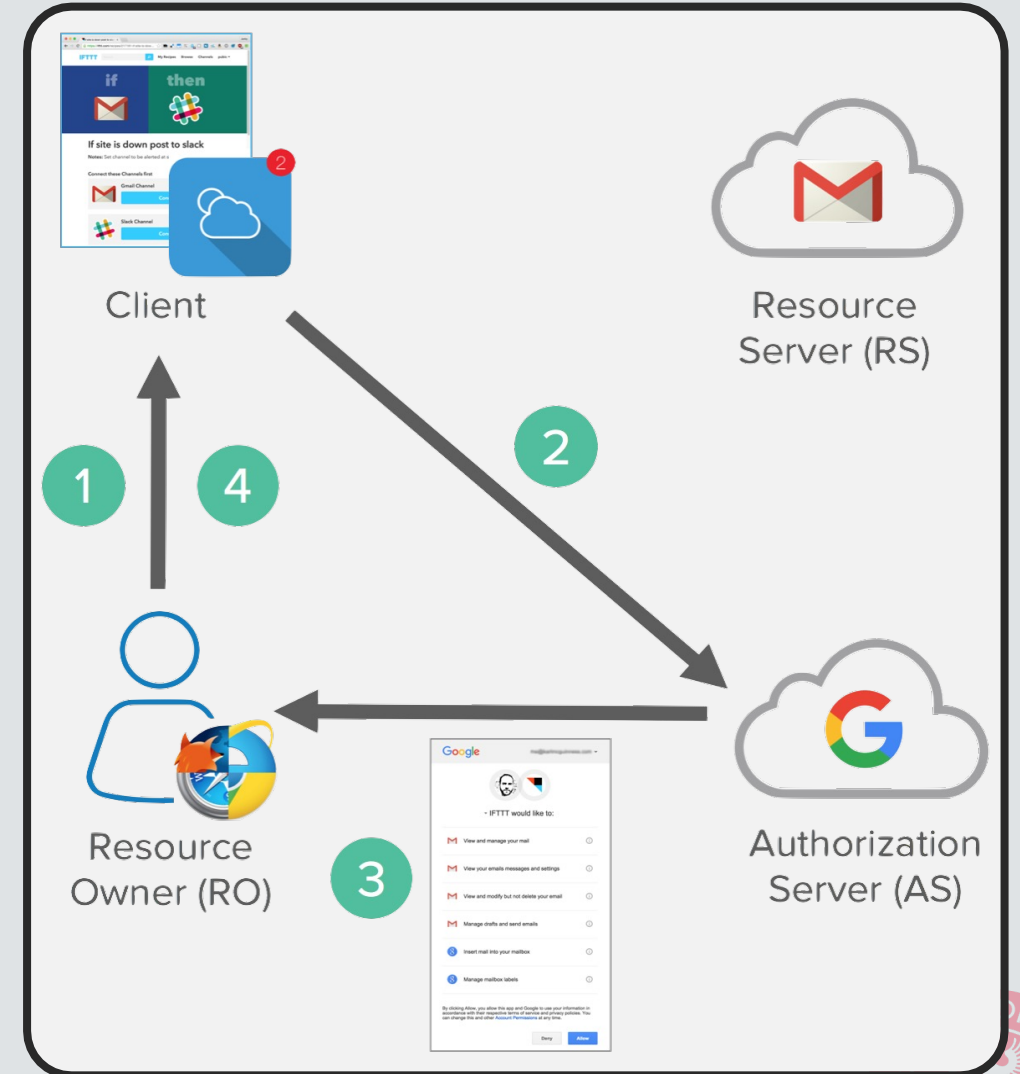
OIDC:

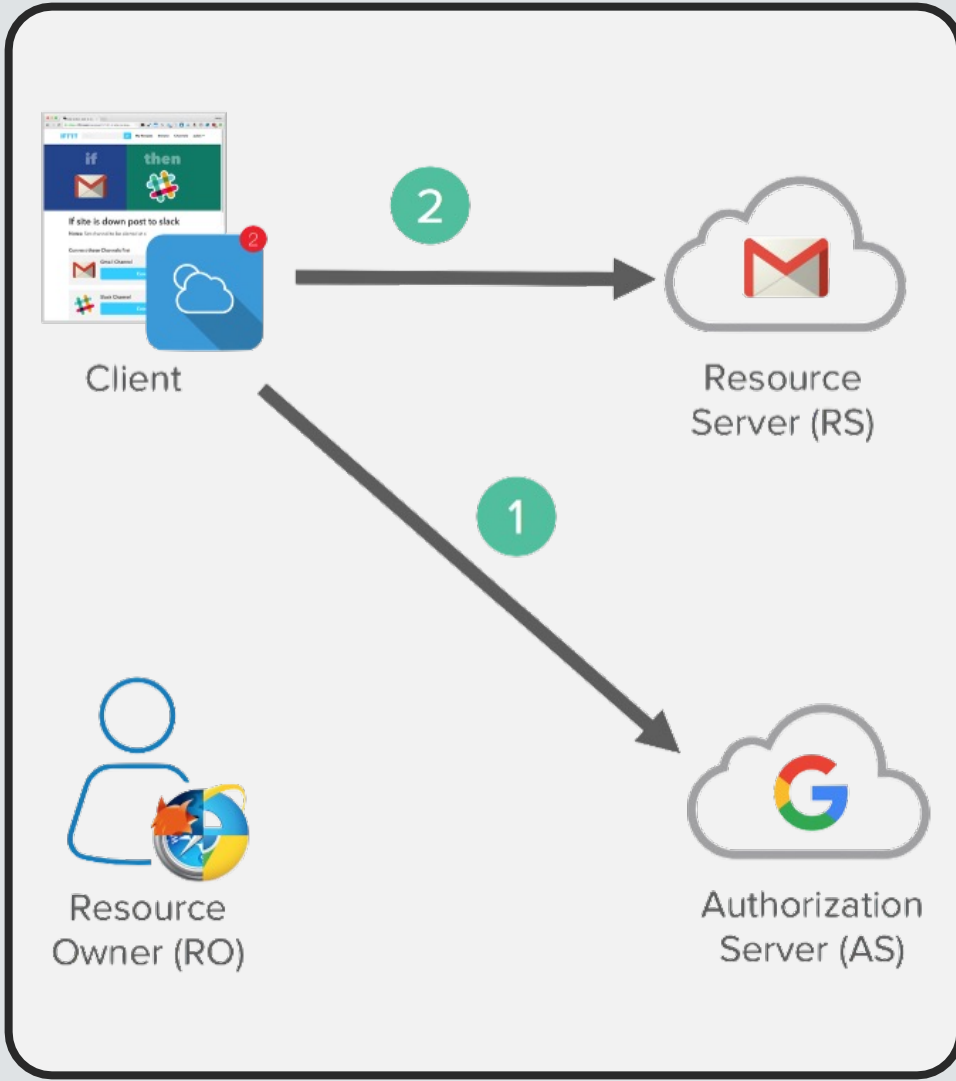
- Aggiunge autenticazione ad OAuth2
- Permette di aggiungere informazioni, **claims**, all'utente come nome, indirizzo, ecc...



FLUSSO DI AUTENTICAZIONE

1. Il resource owner delega la gestione dell'accesso al client
2. Il client richiede all'autorization server il token d'accesso e gli scopes desiderati.
3. L'autorization server comunica all'utente la volontà del client di accedere ai dati del resource owner. Per procedere l'utente deve accettare
4. L'autorizzazione viene quindi confermata al client





SCAMBIO DEI TOKEN

1. Il client manda all'autorization server il token di accesso chiedendo di utilizzare e/o accedere ad una determinata risorsa
2. Se il token è valido per quella richiesta, il resource server garantirà l'accesso



TECNOLOGIE UTILIZZATE

FRONT-END

- Node.js
- Vue.js
- Web3.js

BACK-END

- Java Spring
- Jakarta Persistence



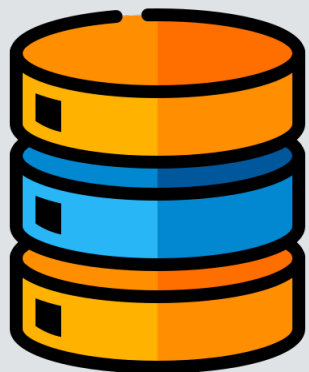
PROGETTAZIONE

COMPONENTI

Client



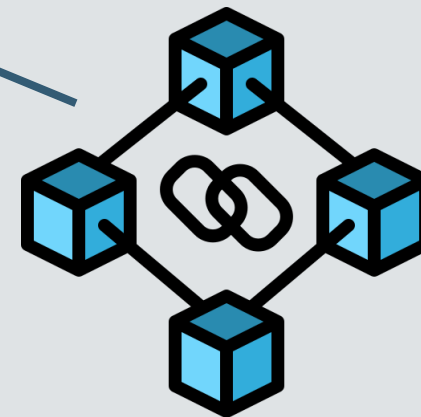
Server OAuth/OIDC



Database



Utente



Blockchain



GESTIONE UTENTE

- Persistenza utente utilizza il database
- Utilizzo di Jakarta Hibernate per la gestione del database
- Il wallet sarà immagazzinato nel database, associandolo all'utente



GESTIONE WALLET

- Web3.js cifra e decifra il wallet
- L'utente OIDC avrà un **claim** personalizzato «wallet»
- Il server non vedrà mai il wallet decifrato
- Sincronizzazione tra password account e password wallet (OAuth2/Web3.js)

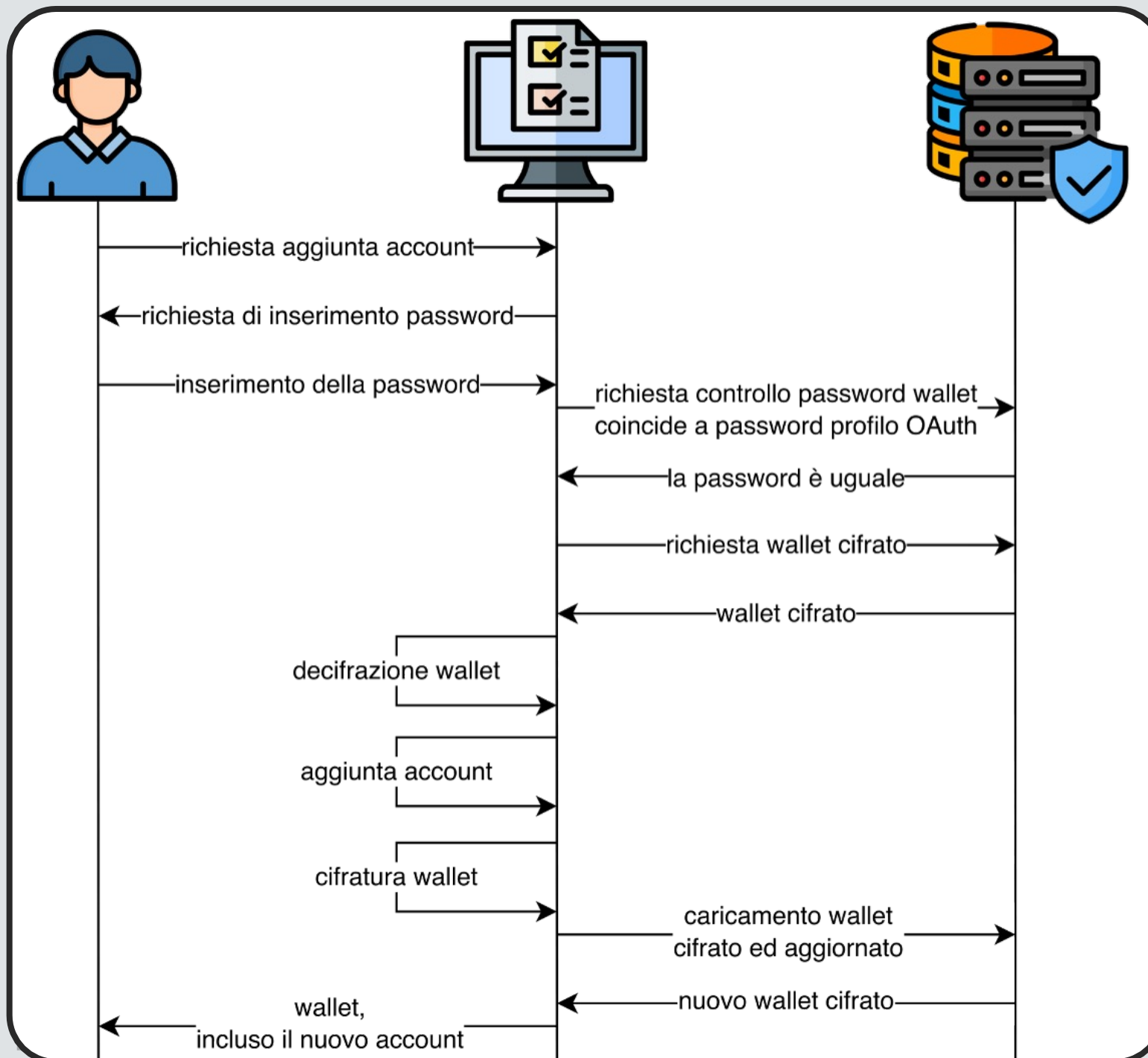


GESTIONE PASSWORD

- La password del profilo utente e del wallet sono sempre uguali. (OAuth2/Web3.js)
- Implementazione di un sistema di controllo per mantenere la coerenza



Caso d'uso aggiunta account



IMPLEMENTAZIONE



Registrati!

SIGN UP

Please sign in

Sign in

Consent required

crypto-client wants to access your account **root**

The following permissions are requested by the above app.
Please review these and consent if you approve.

☐ wallet

☐ profile

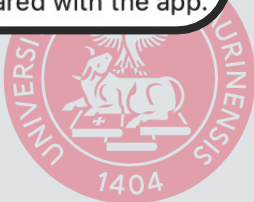
☐ email

Submit Consent

Cancel

Your consent to provide access is required.

If you do not approve, click Cancel, in which case no information will be shared with the app.



Il mio wallet

In uso	Nome	Chiave pubblica	Bilancio	Mostra chiave privata	Elimina
<input type="radio"/>	Account 1	0xDD1C8F10204FBBC76755BD912A15B3401AA7EE87	0.0000000000000000		
<input type="radio"/>	Account 2	0xED0ECCE1901BA361B85BA57FF0A55E4B1BD2BE4F	0.0000000000000000		
<input type="radio"/>	Progetto	0xE45ABCD91D3790C21B2476CDECC83E2D44B9A105	0.0000000000000000		
<input type="radio"/>	Il mio account	0x91AC8494EDC1E5DCA59A8F34DCA5C0151A17B360	0.0000000000000000		
<input type="radio"/>	Personale	0xFD79A5327F8CCBFB280785AFF2F745FEDAFAEA9F	0.0000000000000000		

CONFERMA MODIFICA

GENERA NUOVO ACCOUNT

Importa chiave privata

Chiave privata

IMPORTA

Importa da frase mnemonica

Chiave mnemonica

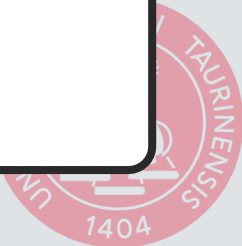
IMPORTA

Sei sicuro di voler mostrare la chiave privata!?

Password

CONFERMA

ANNULLA



 ACCOUNT

 SECURITY

Account Details

First Name

Raoul


Last Name

Rondinella

E-mail

r2000raoul@gmail.com

Phone Number

 ACCOUNT

 SECURITY

Change Password

Current Password



New Password



Confirm New Password



CONFERMA

RESET



CONCLUSIONI

CONCLUSIONI

È stato possibile trovare un buon compromesso tra sicurezza e facilità di utilizzo

L'integrazione di OAuth 2.0 con Ethereum ha permesso di fornire un'autenticazione sicura e semplificata, facilitando la creazione e l'importazione del wallet per gli utenti.

Il progetto riesce ad abbattere le barriere tecniche rendendo possibile l'utilizzo della blockchain anche ad utenti comuni.



INTEGRAZIONE TRA SERVIZI DI
AUTENTICAZIONE E BLOCKCHAIN:
COME GESTIRE WALLET CRITTOGRAFICI

GRAZIE

Università degli Studi di Torino
Corso di Laurea in Informatica
Anno Accademico 2021/2022

