



OBERON

SPECIFICA ARCHITETTURALE V. 1.0.0

A.A. 2021-2022

Componenti del gruppo:

Casazza Domenico, matr. 1201136

Casonato Matteo, matr. 1227270

Chen Xida, matr. 1217780

Pavin Nicola, matr. 1193215

Poloni Alessandro, matr. 1224444

Scudeler Letizia, matr. 1193546

Stojkovic Danilo, matr. 1222399

Indirizzo repository GitHub:

<https://github.com/TeamOberon07/ShopChain>



UNIVERSITÀ
DEGLI STUDI
DI PADOVA

Indice

1	Registro delle modifiche	2
2	Introduzione	3
2.1	Scopo del documento	3
2.2	Obiettivi del prodotto	3
2.3	Riferimenti	3
2.3.1	Riferimenti informativi	3
2.3.2	Riferimenti tecnici	3
3	Architettura	4
3.1	Pattern architetturale	4
4	Punti possibili di estensione	6

1 Registro delle modifiche

v	Data	Nominativo	Ruolo	Descrizione
0.2.0	01/05/2022	Casonato Matteo	Verificatore	Verifica del documento
0.1.1	08/04/2022	Casazza Domenico	Amministratore	Ampliamento Pattern Architettuale §(3.1)
0.1.0	05/04/2022	Casazza Domenico	Verificatore	Verifica del documento
0.0.1	05/04/2022	Chen Xida	Progettista	Creazione bozza documento §(1), §(2), §(3.1)

2 Introduzione

2.1 Scopo del documento

In questo documento si possono trovare i pattern architeturali sfruttati per lo sviluppo del prodotto. Nello specifico faremo riferimento a dei paper pubblicati negli ultimi anni, dato che dopo un breve periodo di ricerca il team ha constatato che non ci sono ancora dei veri e propri design pattern per le cosiddette DApp (Decentralized App).

2.2 Obiettivi del prodotto

Al giorno d'oggi, numerosi sono gli e-commerce che non hanno un sistema affinché l'acquirente e il venditore possano creare transizioni sicure. Difatti, l'acquirente può venire truffato dal venditore se dopo il pagamento non gli viene consegnato il prodotto o viceversa.

ShopChain è un applicativo in grado di affiancare un e-commerce nelle fasi di pagamento fino alla consegna usando la tecnologia delle blockchain. La blockchain è incaricata di ricevere l'ammontare speso dall'acquirente in criptovaluta, consegnandola al venditore solo quando il pacco gli viene recapitato.

Nel momento della consegna del pacco l'acquirente dovrà necessariamente inquadrare il QR code applicato sul collo che ne certifica l'avvenuta consegna. Quindi verrà effettuato il passaggio della criptovaluta dal wallet della piattaforma al wallet del venditore.

2.3 Riferimenti

2.3.1 Riferimenti informativi

- È stato creato il documento *Glossario_1.0.0.pdf* per chiarire il significato dei termini tecnici che possono creare dubbi e perplessità.
- La pianificazione è divisa in sprint, seguendo la metodologia agile. Le modalità e il modello di sviluppo sono riportate nel documento *NormeDiProgetto_1.0.0.pdf*

2.3.2 Riferimenti tecnici

- Pattern architeturali 1: <https://medium.com/hexamount/architecting-modern-decentralized-applications-52b3ac3baa5a>
- Pattern architeturali 2: <https://ieeexplore.ieee.org/abstract/document/8432174>

3 Architettura

3.1 Pattern architetturale

Il team dopo un periodo approfondito di ricerca ha individuato l'architettura più opportuna tra quelle dedicate per le DApp basate su blockchain (sistema distribuito).

L'architettura di ShopChain è di tipo Fully Decentralized (Pure DApp).

In una Pure DApp l'utente, dopo essersi connesso al proprio wallet, dal front-end può chiamare direttamente i metodi dello SmartContract senza dover passare per un intermediario (con tutti i vantaggi e svantaggi che ne conseguono). Questo è possibile se il frontend viene hostato su servizio distribuito come ad esempio IPFS.

Il team ha scelto in particolare questo pattern per questi vantaggi:

- Maggiore decentralizzazione (assenza di un server centralizzato)
- Maggiore sicurezza (non ci sono intermediari tra l'utente e la blockchain)

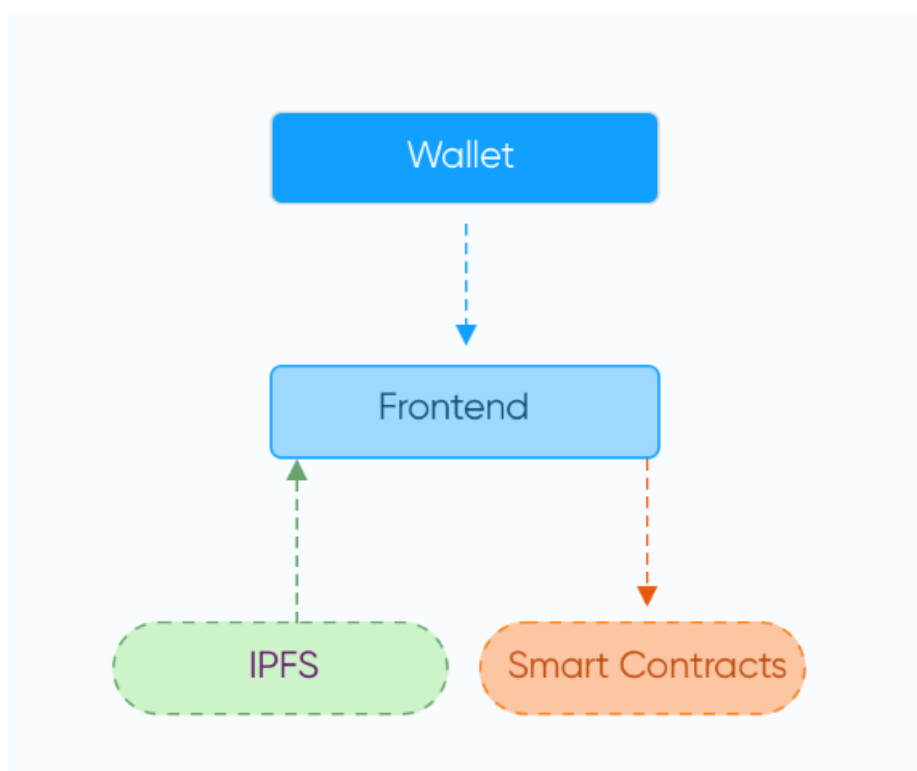


Figura 1: Interazione tra Wallet - Frontend - IPFS - Smart Contract

Fonti:

- <https://medium.com/hexmount/architecting-modern-decentralized-applications-52b3ac3baa5a>
- <https://ipfs.io/>

L'architettura di ShopChain fa riferimento al *"Pattern B – Self-Confirmed Transactions"* descritto nel paper *"Engineering Software Architectures of Blockchain-Oriented Applications"* di F. Wessling e V. Gruhn dell'Università di Duisburg-Essen.

In questo paper il Pattern B consiste nell'interazione dell'utente solo con un'applicazione web e/o un gestore di wallet (es. MetaMask) per creare transazioni su una blockchain: le transazioni non vengono create direttamente dall'utente ma vengono generate dall'applicazione web e poi mandate manualmente al nodo della blockchain a cui l'utente è collegato.

Questo pattern bilancia sicurezza e facilità nell'interazione con la webApp perché creare transazioni manualmente è un'operazione difficile e realizzabile solo da utenti esperti, ma questo implica che chi interagisce con l'applicativo si fidi degli sviluppatori dato che la generazione di una transazione non è completamente trasparente.

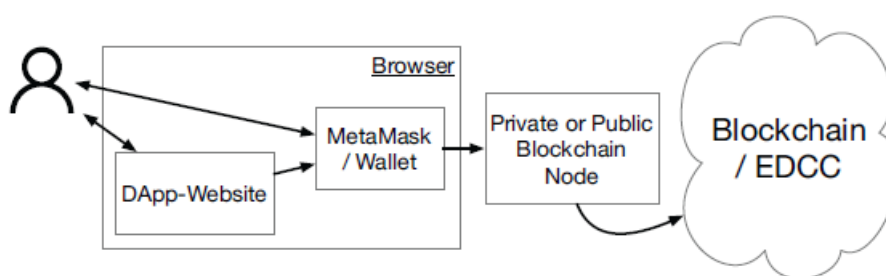


Figura 2: DApp Pattern B - Self-Confirmed Transactions

Fonte:

- <https://ieeexplore.ieee.org/abstract/document/8432174>

4 Punti possibili di estensione