



Winning Software Solution

winningsoftwaresolution@gmail.com

## Analisi delle tecnologie

### *Informazioni*

<i>Redattori</i>	Federico Marchi Matteo Galvagni
<i>Revisori</i>	Giovanni Cocco
<i>Responsabili</i>	Elia Scandaletti
<i>Versione</i>	1.0.0
<i>Uso</i>	Esterno

### Descrizione

---

Analisi delle tecnologie.

Versione	Data	Persona	Attività	Descrizione
1.0.0	11/02/2022	Elia Scandaletti	Approvazione	Approvazione per RTB.
0.0.4	11/02/2022	Marchi Federico	Redazione	Correzione indice di Gulpease.
0.0.3	10/02/2022	Matteo Galvagni	Redazione	Aggiornamento di alcune tecnologie.
0.0.2	05/01/2022	Matteo Galvagni	Redazione	Stesura iniziale sezione Scansione QR code.
0.0.1	28/12/2021	Federico Marchi	Redazione	Stesura iniziale sezione Backend, Pagamento e Database.
0.0.0	24/12/2021	Matteo Galvagni	Redazione	Stesura iniziale sezione Introduzione e Blockchain.

# Introduzione

L'infrastruttura designata per il progetto verte sulla comunicazione diretta tra attori e blockchain. Bisognerà dunque sviluppare:

- un contratto digitale per lo smistamento dei fondi;
- una landing page per facilitare il pagamento;
- un servizio di visualizzazione di transazioni per venditori e clienti;
- uno script per automatizzare la messa in vendita;
- un server per fornire le pagine di consultazione e landing page;
- un database per memorizzare gli ID delle transazioni;
- una web app per il cliente finale per lo sblocco dei fondi e relativo controllo.

Per richiamare funzioni nel contratto sarà necessario utilizzare più librerie. Grazie a queste sarà possibile richiamare le funzioni sia lato server che lato client.

## Blockchain

La blockchain sulla quale sviluppare il contratto digitale dovrà garantire alcuni requisiti; tra questi:

- una elevata velocità di transazione;
- un costo di transazione quanto più basso possibile;
- un buon livello di sicurezza.

Questi requisiti sono richiesti al fine di garantire una buona usabilità per tutti gli attori. Sarà inoltre cruciale la velocità e il costo ridotto per lo sblocco dei fondi. Diverse blockchain sono state prese in esame, ma la scelta è ricaduta su Polygon; infatti:

- i costi di transazione sono molto bassi;
- la velocità di transazione è ottima;
- eredita parte della sicurezza di Ethereum grazie al meccanismo di salvataggio a "checkpoint".

Polygon è compatibile con la Ethereum Virtual Machine. Il suo linguaggio nativo è Solidity. La documentazione esistente per questo linguaggio è esaustiva e facilmente reperibile. Verrà utilizzata la comoda e gratuita testnet di Polygon "Mumbai" per lo sviluppo del contratto.

## Backend

Sono state valutate diverse opzioni per la scelta del framework e del linguaggio per lo sviluppo lato backend. Tra questi sono stati considerati Java e Javascript; infatti ci consentono di utilizzare le librerie Web3j e Web3js. Queste librerie permettono di emettere transazioni in blockchain; inoltre permettono la comoda gestione di oggetti (transazione, contratto, ecc). La scelta del framework è ricaduta su NodeJS e sul linguaggio Typescript; infatti ci consente di usufruire della comodità di Javascript integrando la tipizzazione; Per l'ascolto degli eventi emessi lato contratto sarà necessario un nodo presso il quale "ascoltare". La scelta è ricaduta sul servizio di API per Polygon "Moralis.io"; infatti è uno dei pochi nodi che offre un punto di accesso WebSocket e non solo in HTTPS. Questo sarà necessario per la "sottoscrizione" ad eventi in JS.

## Pagamento automatizzato alla messa in vendita

Per automatizzare il pagamento di un oggetto con ShopChain sarà necessario integrare uno script. In particolare durante la fase di vendita. Questo script dovrà dialogare con la blockchain per creare un'istanza per il pagamento; verrà poi generato un ID univoco. Inoltre dovrà essere conservato il prezzo; infatti dovrà essere poi confrontato con il denaro inviato dal cliente all'acquisto. Tale script verrà sviluppato in Python; per comodità di sviluppo e facilità di integrazione. Sarà necessario utilizzare la libreria "Web3py". Non utilizzando Metamask, il nodo presso la quale emettere le transazioni sarà fornito dal servizio di API per Polygon "Moralis.io".

## Database

Per la memorizzazione dei dati sulle transazioni è necessario l'utilizzo di un database. La scelta è ricaduta su MariaDB; infatti:

- il gruppo lo ha già utilizzato;
- Postgres è stato escluso poiché richiede un downgrade di alcuni pacchetti Python;
- non sono emerse necessità particolari per l'utilizzo di un database specifico.

## Scansione QR code

Il wallet Metamask su mobile non fornisce un'estensione per browser; infatti si tratta di un'applicazione con relativo browser integrato. Verrà dunque sviluppata una webapp per lo scansionamento del QR code; tale webapp dovrà permettere un comodo accesso dal browser di Metamask.