

roundabout

Roundabout - Etherless

Verbale esterno 2020-03-27

Versione	0.0.1
Approvazione	
Redazione	Feim Jakupi
Verifica	
Stato	Non Approvato
Uso	Esterno
Destinato a	<i>Roundabout</i> Prof. Tullio Vardanega Prof. Riccardo Cardin

Descrizione

Riassunto dell'incontro tra il gruppo Roundabout ed il Proponente tenutosi il 2020-03-27.

team.roundabout.13@gmail.com

Diario delle modifiche

Versione	Data	Nominativo	Ruolo	Descrizione
0.0.1	2020-03-27	Feim Jakupi	<i>Progettista</i>	Stesura del documento.

Indice

1	Informazioni generali	3
1.1	Informazioni incontro	3
1.2	Ordine del giorno	3
2	Verbale	4
2.1	Richiesta di chiarimenti riguardo il capitolato C2 - <i>Etherless</i>	4
2.2	Tecnologie individuate	4
3	Riepilogo delle decisioni	5

1 Informazioni generali

1.1 Informazioni incontro

- **Luogo:** Chiamata tramite Zoom;
- **Data:** 2020-03-27;
- **Ora di inizio:** 11.30;
- **Ora di fine:** 12.00;
- **Partecipanti:**

Tutti i membri di *Roundabout*:

- Veronica Barbieri;
- Luca Benetazzo;
- Nicoletta Fabro;
- Egon Galvani;
- Feim Jakupi;
- Marco Positello;
- Alessandro Sgreva;
- Antonio Zlatkovski.

Referenti di *RedBabel*:

- Alessandro Maccagnan;
- Milo Ertola.

- **Segretario:** Feim Jakupi.

1.2 Ordine del giorno

- Richiesta di chiarimenti riguardo il capitolato C2 - *Etherless*;

2 Verbale

2.1 Richiesta di chiarimenti riguardo il capitolato C2 - *Etherless*

I membri del gruppo hanno esposto una serie di domande volte a chiarire dei dubbi su parti del capitolato e varie tecnologie da utilizzare. Di seguito si trova un riassunto delle informazioni estrapolate dalla discussione:

Ambiente di sviluppo: la scelta dell'ambiente di sviluppo per la programmazione degli smart contract in Solidity è stata lasciata al team. E' stato consigliato l'uso di un programma di scripting invece di un IDE. Alcune valide opzioni possono essere Visual Studio Code e Truffle;

Identificazione delle funzioni: Le funzioni possono essere identificate con il nome;

Libreria per comunicazione con nodi Ethereum: La libreria proposta è Web3 ma non è vincolante. Possono essere valutate delle alternative come ethers.js, la scelta necessita di una attenta valutazione;

Costo delle funzioni: Il costo della funzione può essere implementato in diversi modi: da una stima calcolata dallo sviluppatore, pensato come un sistema di crediti, oppure come escrow. La possibilità di stabilire un costo differente per la modifica e la rimozione delle funzioni è un dettaglio implementativo che deve essere valutato. Lo stesso vale anche per il deploy delle funzioni;

Caricamento delle funzioni sulla piattaforma: il comportamento della funzione caricata dallo sviluppatore non deve inficiare il funzionamento del sistema. Non è necessario applicare l'analisi statica sulla funzione ma è invece consigliato esercitare controlli sulla infrastruttura e sugli effetti che il codice vi può provocare applicando delle limitazioni;

Gestione del codice di deploy: valutare la possibilità di non usare in questo caso gli smart contract, ma effettuare una scelta implementativa alternativa;

2.2 Tecnologie individuate

Sono state individuate una serie di tecnologie da utilizzare per lo sviluppo:

- **Solidity:** come linguaggio per l'implementazione degli smart contract;
- **Airbnb Javascript:** come standard che delinea il modo in cui il codice sarà scritto;

3 Riepilogo delle decisioni

Tabella 3.1: Decisioni della riunione esterna del 2020-03-27

Codice	Decisione
VE_2.1	Scelto di usare Solidity per lo sviluppo degli smart contract.
VE_2.2	Scelto di usare Airbnb Javascript style guidance.