



Marvin - Gruppo 353

Raccolta verbali

Versione 0.1 in data 22-11-2017

Gruppo 353

Informazioni sul documento

Responsabili	Elena Mattiazzo
Redazione	Mirco Cailotto
Verifica	Parwinder Singh
	Davide Stocco
	Gianluca Marraffa
	Valentina Marcon
	Riccardo E. Giorato
Stato	In corso
Uso	Interno
Destinato a	Prof. Tullio Vardanega
	Prof. Riccardo Cardin
	Gruppo 353
Email di contatto	353swe@gmail.com



Diario delle modifiche

Versione	Data	Descrizione	Autore	Ruolo
0.0.1	22-11-2017	Inserita la discussione avvenuta su Slack il 22 ottobre 2017	Mirco Cailotto	-

Indice

1	Prima discussione	1
1.1	Informazioni generali	1
1.1.1	Informazioni discussione	1
1.1.2	Argomenti	1
1.2	Domande e risposte	1
1.3	Riepilogo tracciamento decisioni	3

1. Prima discussione

1.1 Informazioni generali

1.1.1 Informazioni discussione

- **Mezzo di comunicazione:** Slack
- **Data:** 22 ottobre 2017
- **Ora:** 11:00
- **Partecipanti del gruppo:** Mirco Cailotto, Riccardo E. Giorato, Valentina Marcon, Gianluca Marraffa, Elena Mattiazzo, Parwinder Singh, Davide Stocco
- **Partecipanti esterni:** Alessandro Maccagnan (alemhnan), Milo Ertola (milo)

1.1.2 Argomenti

Gli argomenti vertono principalmente su chiarimenti in merito alla implementazione del progetto ed allo stile da seguire per la scrittura del codice. Il proponente ha esposto alcune precisazioni sui requisiti e indicato le parti sulle quali impiegare maggior impegno.

1.2 Domande e risposte

1. Possiamo scrivere la documentazione in italiano?

Alessandro Maccagnan: La documentazione 'interna' in italiano. Quella 'esterna' in inglese. Per esterna quella che viene letta da chi usa quello che fate. Per esempio se dovete dare la repo a qualcuno, le istruzioni per installare e lanciare o estendere le funzionalità, in questi casi deve

essere in inglese.

2. **Il login può essere implementato in modo trasparente attraverso Metamask? Cioè l'utente, nel momento che accede al sito, sottopone il suo indirizzo pubblico a una analisi e, senza necessità di ulteriori login con email o password, accede alla sua area privata. Ovviamente in caso di errori (indirizzo non registrato, Metamask non installato o bloccato da password) verranno presentate all'utente delle informazioni su come procedere.**

Alessandro Maccagnan: L'identità in Ethereum si prova avendo una coppia di chiavi pubbliche/private. Ogni volta che si parla con la rete Ethereum (qualsiasi rete) dovete firmare le transazioni sottomesse con la vostra chiave. Questa procedura non e' proprio lineare da fare per un utente da browser. Quindi si usa Metamask. Metamask fa' due cose: vi gestisce le vostre chiavi e mette a disposizione delle pagine web un oggetto Web3. Questo oggetto si occupa di firmare le transazioni con le chiavi gestite da Metamask. Qui c'e' un tutorial che spiega in pratica come fare:

<http://truffleframework.com/docs/advanced/truffle-with-metamask>

3. **La registrazione può avvenire in due fasi? Pensavamo ad una prima fase nel quale l'utente immette il suo nome e cognome nel sistema, e questi assieme alla sua chiave pubblica viene inserito in una lista "pending", per essere confermato deve presentarsi all'università con i documenti ed accertare la propria chiave, questo per evitare registrazioni fasulle.**

Alessandro Maccagnan: Si va bene. State attenti all'implementazione.

4. **Attualmente non sembra esistere nessun standard UML per quanto riguarda Solidity, è accettabile nella documentazione rappresentare un contratto come se fosse una classe nella programmazione ad oggetti e ridefinire gli aspetti che si discostano da quelli standard, come ad esempio la visibilità?**

Alessandro Maccagnan: UML e' uno strumento che viene usato con profitto in un insieme specifico di casi e industrie. Per quanto riguarda la domanda specifica vi rimando a Vardanega o Cardin. Per quanto riguarda me e Milo non abbiamo interesse ad indagare la parte di UML.

Milo Ertola: Per estendere il punto 3 di Ale. La documentazione non

ci fornisce alcun valore. Se avrete la necessità di spiegare, chiarire, formalizzare un concetto per poter proseguire nell'implementazione del progetto vi invitiamo caldamente di fare un prototipo che investiga l'idea/il concetto in questione

5. **Promise centric approach: cosa vuol dire promise centric approach per noi durante il progetto?**

Alessandro Maccagnan: Non usate le callbacks.

6. **The usage of callbacks must be limited and thoroughly justified ossia dobbiamo preferire le promise alle callback lavorando con redux sagas?**

Alessandro Maccagnan: Quella riga fa letta come: 'non usate le callback' quindi, a meno di casi specifici (in nodejs ci sono alcuni casi in cui non si puo' fare a meno dell callback), usate solo promesse.

7. **React: possiamo usare Arc.js come base di partenza frontend per permette uno sviluppo con basi aggiornate rispetto a Redux-minimal?**

Alessandro Maccagnan: Risposta breve, no. Arc.js usa concetti (Atomic design) troppo articolati per lo scopo del progetto. La scelta di uno starter kit minimal e' stata appunto quella di mettervi nella situazione di usare le componenti principali dell'architettura react (React+Redux+Integrazione tra le 2), ma senza troppe complicazioni, non necessarie in questo progetto specifico.

1.3 Riepilogo tracciamento decisioni

TODO