An end-to-end voting-system based on ethereum

Federico Ginosa

Requisiti richiesti e soddisfatti

Requisiti richiesti e soddisfatti

Ammissibilità e autenticazione
in un sistema di votazione elettronico deve poter votare solo chi è autorizzato

Verificabilità e Auditabilità
è possibile verificare che tutti i voti siano stati correttamente contabilizzati nel conteggio finale

Unicità
nessun elettore può votare più di una volta

Precisione
i sistemi elettorali dovrebbero registrare correttamente i voti, con una tolleranza agli errori estremamente ridotta

Requisiti richiesti e soddisfatti

Integrità
i voti non devono poter essere modificati, falsificati o eliminati senza rilevamento

Votare in maniera anonima
né le autorità elettorali né nessuno dovrebbe essere in grado di determinare il modo in cui un individuo ha votato. L'anonimato del voto deve essere garantito anche nel caso in cui l'elettore stesso volesse infrangerlo, questo per evitare la vendita di voti

Conteggio e riconteggio
il risultato dell'elezione deve poter essere accessibile solo al termine delle votazioni.

Risultato solo a urne chiuse
il risultato dell'elezione deve poter essere accessibile solo al termine delle votazioni.

Cos'è Ethereum?

Solidity : Java = JVM : ETH VM

Esempio di Smart Contract

```
contract HelloSmartContract {
    uint256 a;
    function HelloSmartContract() {
      a = 1;
    }
}
```

Esempio di Smart Contract

```
solc - -bin - -asm HelloSmartContract.sol
```

```
60606040523415600e57600080fd5b5b 60016000819055505b5b603680602
6
6000396000f30060606040525b600080fd00a165627a7a72305820af3193f
6
fd31031a0e0d2de1ad2c27352b1ce081b4f3c92b5650ca4dd542bb770029
```

Esempio di Smart Contract

60 x: pusha il valore x sullo stack

81: duplica il secondo elemento dello stack

90: swappa i primi due elementi affioranti sullo stack

55: prende i primi due elementi e effettua lo store del secondo elemento nella posizione indicata nel primo elemento e poi cancella dallo stack tutti e due valori 50: fa il pop del valore affiorante

Esempio di Smart Contract

60 01	Pusha 1 sullo stack	[1]
60 00	Pusha 0 sullo stack	[0 1]
81	Copia il secondo elemento dello stack (1) e lo pusha.	[1 0 1]
90	Swappa i primi due elementi affioranti dello stack.	[0 1 1]
55	Effettua lo store del secondo elemento dello stack (1) nella posizione indicata dal primo elemento dello stack (0) e poi cancella tutti e due i valori.	[1] Store Blockchain: 0 => 1
50	Fa il pop dell'elemento affiorante.	[ ]

Paperopoli alle urne

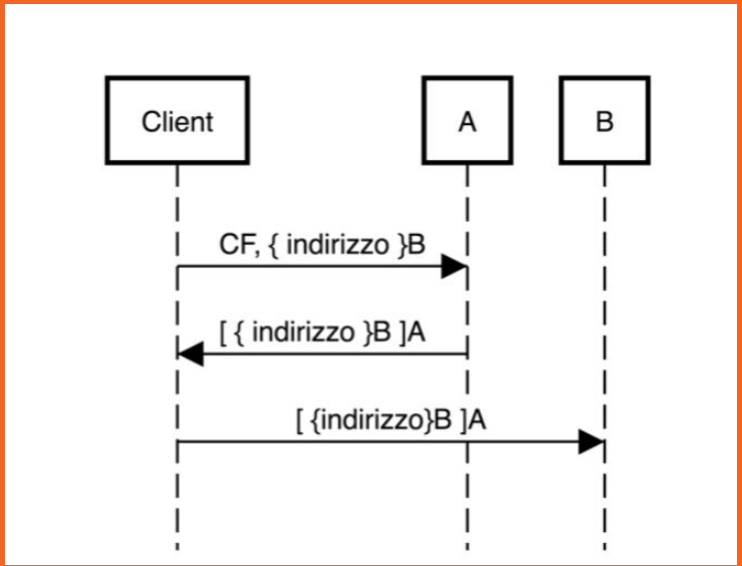
Smart Contract Elezioni

```
contract ElezioniAPaperopoli {

    mapping(address => address[]) candidati;

    function vota(address candidato) public {
        require(!contains(getElettori(candidati), msg.sender);
        candidati[candidato].push(msg.sender);
      }

    function votiByCandidato(address candidato) public returns(address[]) {
        return candidati[candidato];
      }

}
```

Client

A: Verificatore credenziali

B: Registratore dell’indirizzo



DEMO

http://localhost:3000/