

Elemental Battles



Welcome, zehaoli@hotmail.com | [SIGN OUT](#)

Build a blockchain game. Learn EOSIO.

[GET STARTED](#)

Current capabilities

- What blockchain **can** do
 - Turn based
 - Digital assets (i.e. CryptoKitties, dGoods)
 - Pseudo random
- What blockchain **can't** do (yet)
 - Real-time interaction
 - True randomization
 - Fetch external data (oracle)

Components

- Signup & login
- Rules of the game
 - Game state - *Contract tables*
 - Gameplay - *Contract actions*
- Randomization
- Game AI

Dependencies

- React - frontend framework
 - <https://github.com/facebook/react>
- Redux - frontend state machine
 - <https://github.com/reduxjs/redux>
- eosjs
 - <https://github.com/EOSIO/eosjs>

Project structure

<https://github.com/eos-studio/elemental-battles/tree/lesson-3>

- Frontend

```
└── frontend
    ├── node_modules
    └── public
        └── ...
    └── src
        ├── actions
        ├── components
        ├── const
        ├── reducers
        └── services
            ├── ApiService.js
            ├── index.js
            ├── store
            └── index.js
        └── ...
    └── ...
└── package-lock.json
└── package.json
```

- Smart contract

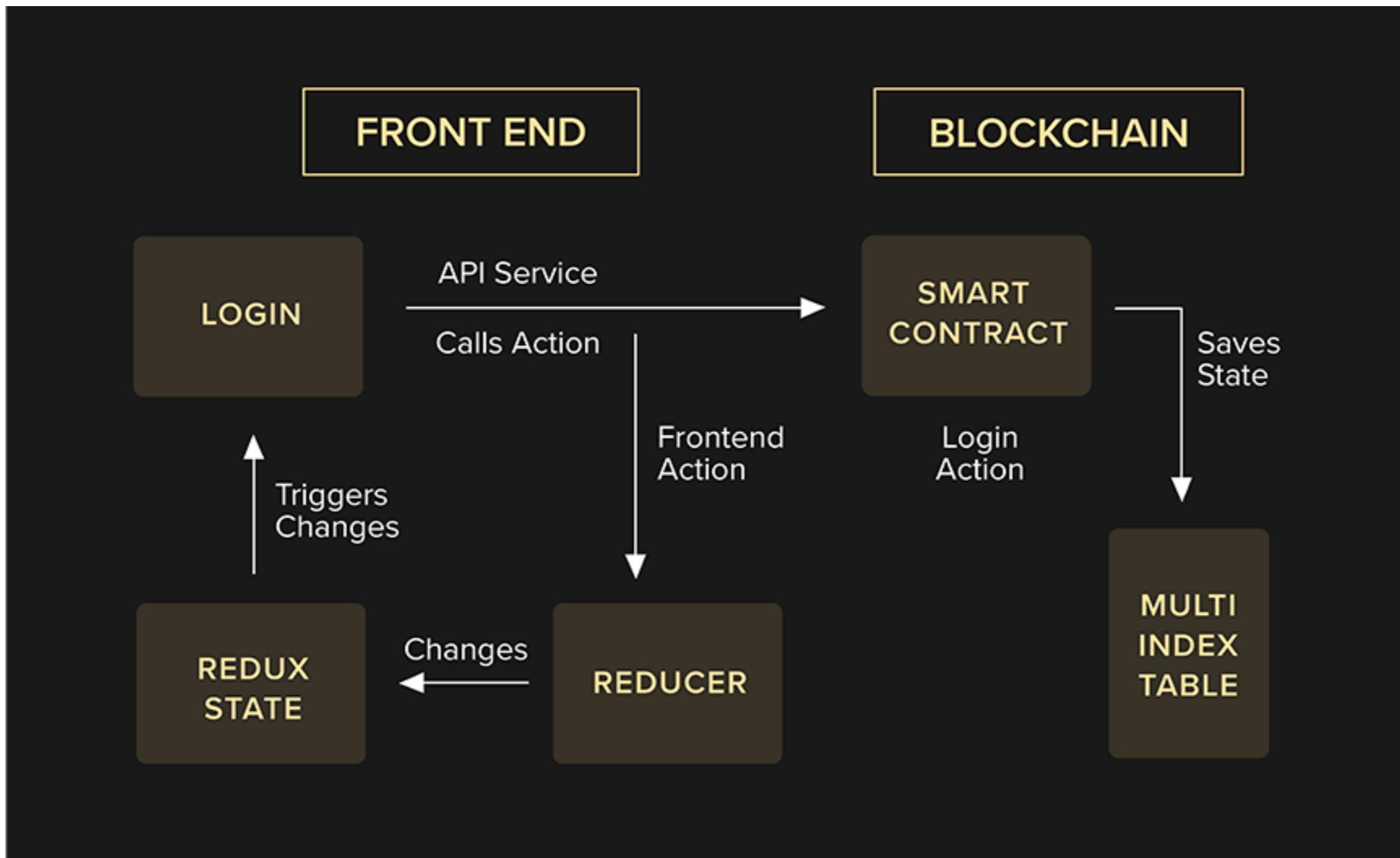
```
└── cardgame
    ├── cardgame.cpp
    ├── cardgame.hpp
    └── gameplay.cpp
```

React

Redux

eosjs

Project structure - Signup



Rules of the game

<https://github.com/eos-studio/elemental-battles/tree/lesson-6>

Game State

```
50  struct game {
51      int8_t          life_player = 5;
52      int8_t          life_ai = 5;
53      vector<uint8_t> deck_player = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17};
54      vector<uint8_t> deck_ai = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17};
55      vector<uint8_t> hand_player = {0, 0, 0, 0};
56      vector<uint8_t> hand_ai = {0, 0, 0, 0};
57      uint8_t         selected_card_player = 0;
58      uint8_t         selected_card_ai = 0;
59      uint8_t         life_lost_player = 0;
60      uint8_t         life_lost_ai = 0;
61      int8_t          status = ONGOING;
62  };
```

Rules of the game

Actions

```
static startGame() {
| return takeAction("startgame", { username: localStorage.getItem("cardgame_account") });
}

static playCard(cardIdx) {
| return takeAction("playcard", { username: localStorage.getItem("cardgame_account"), player_card_idx: cardIdx });
}

static nextRound() {
| return takeAction("nextround", { username: localStorage.getItem("cardgame_account") });
}

static endGame() {
| return takeAction("endgame", { username: localStorage.getItem("cardgame_account") });
}
```

Randomization

```
int cardgame::random(const int range) {
    // Find the existing seed
    auto seed_iterator = _seed.begin();

    // Initialize the seed with default value if it is not found
    if (seed_iterator == _seed.end()) {
        seed_iterator = _seed.emplace( _self, [&]( auto& seed ) { });
    }

    // Generate new seed value using the existing seed value
    int prime = 65537;
    auto new_seed_value = (seed_iterator->value + now()) % prime;

    // Store the updated seed value in the table
    _seed.modify( seed_iterator, _self, [&]( auto& s ) {
        s.value = new_seed_value;
    });

    // Get the random result in desired range
    int random_result = new_seed_value % range;
    return random_result;
}
```

Game AI

- AI doesn't know which card you are going to use.
- AI will randomly pick a strategy so player doesn't know which card AI is going to use.

Maximize win

Minimize loss

Balanced

Loss prevention

Thank you

- EOS Studio: <https://www.eosstudio.io>
- Documentations: <https://docs.eosstudio.io>
- GitHub: <https://github.com/ObsidianLabs>