ClojureScript in the decentralized world

Alexander Pantiukhov
github.com/alwx

Is it possible to build reliable and truly decentralized web applications?

What ClojureScript tools can we use?

Is it worth the effort?

What is a decentralized application?

Architectural—how many physical computers is a system made up of? How many of those computers can it tolerate breaking down at any single time?

Political—how many individuals or organizations ultimately control the computers that the system is made up of?

Logical—does the interface and data structures that the system presents and maintains look more like a single monolithic object, or an amorphous swarm?

Vitalik Buterin
Ethereum founder

Blockchain

The simplified version of “Bullshit Bingo” game
for startup conferences in 2017

Ethereum

decentralised, immutable “database” with built-in currency support, where you can not only store your app’s data, but also API for accessing & changing those data.

web3.js
solidity *(smart contracts)*
swarm *(storage)*
moon *(code-interchange format)*

web3 + ClojureScript

reagent + re-frame

https://github.com/district0x/cljs-web3

https://github.com/district0x/re-frame-web3-fx

```clojure
(defn account-view [account]
  [:div {:on-click #(rf/dispatch [:log-in account])}
   [:h4 "Unknown account"]
   [:small
    account]])

(defn scene-view []
  [:div.centered-root
   [:div.container
    (if (exists? js/web3)
      (let [accounts (cljs-web3.eth/accounts js/web3)]
        [:div
         [:h1 "Choose your account:"]
         [:div.accounts
          (for [account accounts]
            ^{:key (str "account-" account)}
            [account-view account])]])
      [:div
       [:h1 "Not connected"]
       "You can use Metamask, Mist Browser or Status.im."])]])
```

```clojure
(defn account-view [account]
  [:div {:on-click #(rf/dispatch [:log-in account])}
   [:h4 "Unknown account"]
   [:small
    account]])

(defn scene-view []
  [:div.centered-root
   [:div.container
    (if (exists? js/web3)
      (let [accounts (cljs-web3.eth/accounts js/web3)]
        [:div
         [:h1 "Choose your account:"]
         [:div.accounts
          (for [account accounts]
            ^{:key (str "account-" account)}
            [account-view account])]])
      [:div
       [:h1 "Not connected"]
       "You can use Metamask, Mist Browser or Status.im."])]])
```

```clojure
(defn account-view [account]
  [:div {:on-click #(rf/dispatch [:log-in account])}
   [:h4 "Unknown account"]
   [:small
    account]])

(defn scene-view []
  [:div.centered-root
   [:div.container
    (if (exists? js/web3)
      (let [accounts (cljs-web3.eth/accounts js/web3)]
        [:div
         [:h1 "Choose your account:"]
         [:div.accounts
          (for [account accounts]
            ^{:key (str "account-" account)}
            [account-view account])]])
      [:div
       [:h1 "Not connected"]
       "You can use Metamask, Mist Browser or Status.im."])]])
```

```
(:require [cljsjs.web3]
...)
```

MetaMask
Mist Browser
Brave *(contains MetaMask)*
Status.im

Saving data to blockchain

```
pragma solidity ^0.4.0;

contract SimpleStorage {
    uint storedData;

    function set(uint x) {
        storedData = x;
    }

    function get() constant returns (uint) {
        return storedData;
    }
}
```

```
(def contract-address “0x56Cc236D21334d97EFF2643c350E2e285413Ef0c")

(def abi (.parse js/JSON “[{\”constant\”:true,\"inputs\":
[{\"name\":\"entityAddress\",\"type\":\"address\"}],\"name\":
\"isEntity\",\"outputs\":[{\"name\":\"isIndeed\",..."))

(def contract-object
  (web3-eth/contract-at js/web3 abi contact-address))

(.set contract-object 666)
```

Demo

Transactions in blockchain are extremely slow for a rich application data flow;

if all the applications would keep their data in blockchain, the blockchain size will grow rapidly.

IPFS *(peer-to-peer)*

Swarm *(native to Ethereum)*

Sia

Storj

...

```
struct Message {
    address authorAddress;
    string text;
    string ipfsHash;
    uint date;
}
```

https://github.com/ipfs/js-ipfs

Demo

BigchainDB *(build upon RethinkDB cluster)*

Ties.network *(modification of Cassandra)*

https://github.com/bigchaindb/kyber

I think framing Ethereum as logic, IPFS as disk and BigchainDB as database - a direct parallel to the web stack - is nothing short of genius and really points the way to the future of this industry and, shortly, the web itself.

Vinay Gupta
Ethereum Foundation & Consensys

Blockchain transactions are slow

Blockchain transactions are slow

Existing storage solutions are pretty basic

Blockchain transactions are slow
Existing storage solutions are pretty basic
No good databases

Blockchain transactions are slow
Existing storage solutions are pretty basic
No good databases
Lack of stability

Is it worth the effort?

Thanks!

Slides & code: https://github.com/alwx/clojure-meetup-amsterdam-2017-11