Building your first Ethereum DApp

Kevin Bluer

Agenda

Agenda

- What is a DApp?
- User Interaction with a DApp
- Stepping Back / Reviewing the Technologies
- Hello World DApp (running locally)
- More Sophisticated Voting Example (on Ropsten)
- Summary and Q&A

About Me

- CTO, Nest.vc
- Full-stack Developer Focused on Node.js
- Currently exploring applications of Ethereum, Smart Contracts, DApps, etc within Nest + Mettā
- Reach me at: kevin@bluer.com

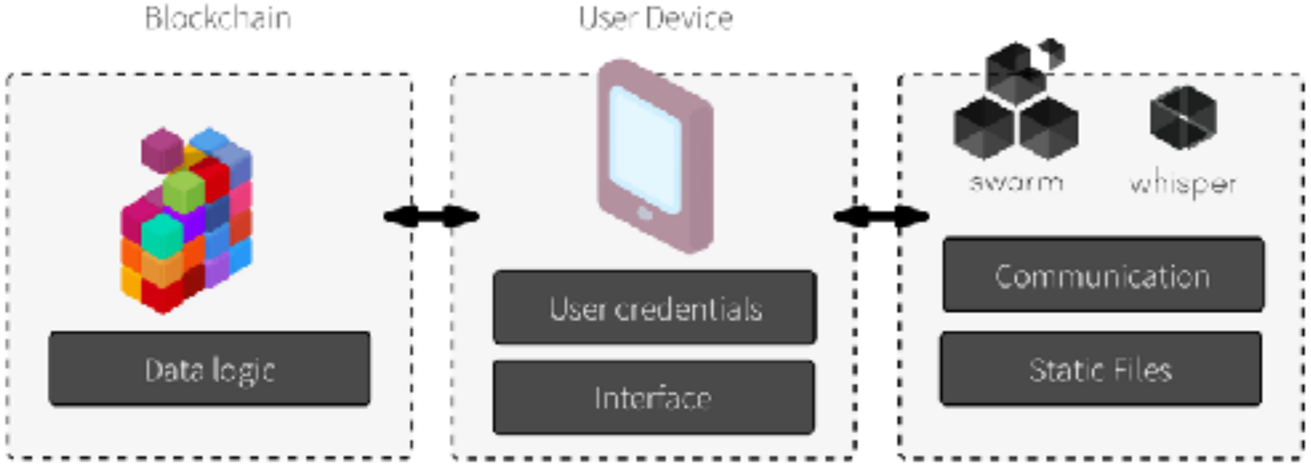
What is a DApp?

Show of Hands

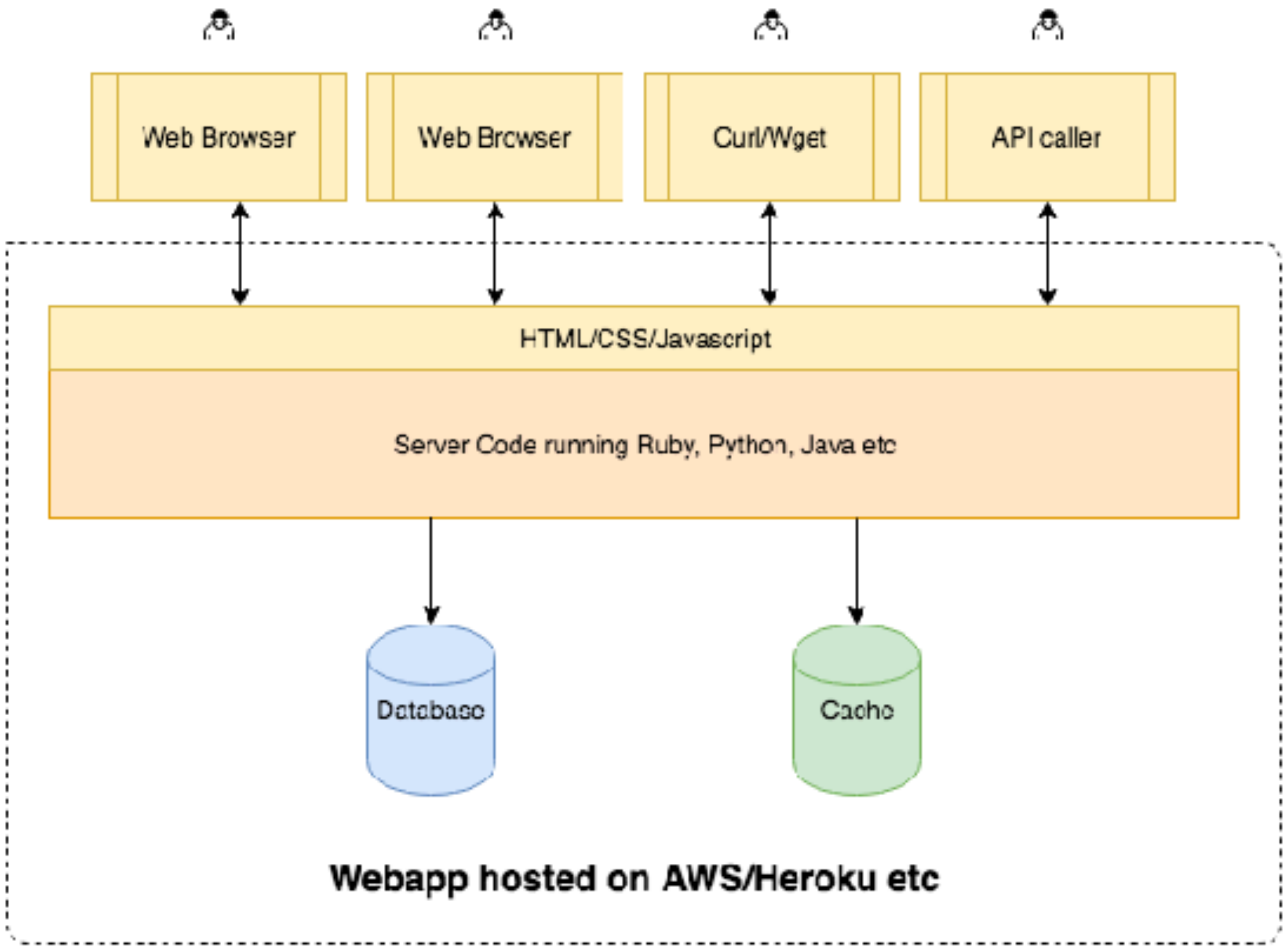
- Who can describe a DApp?
- Who’s used one before in some capacity?
- Who’s participated in an ICO (via a DApp)?

What is a DApp?

- A DApp is a Decentralized Application
- Traditional Web Application on the front-end
- Ethereum (Blockchain) on the back-end

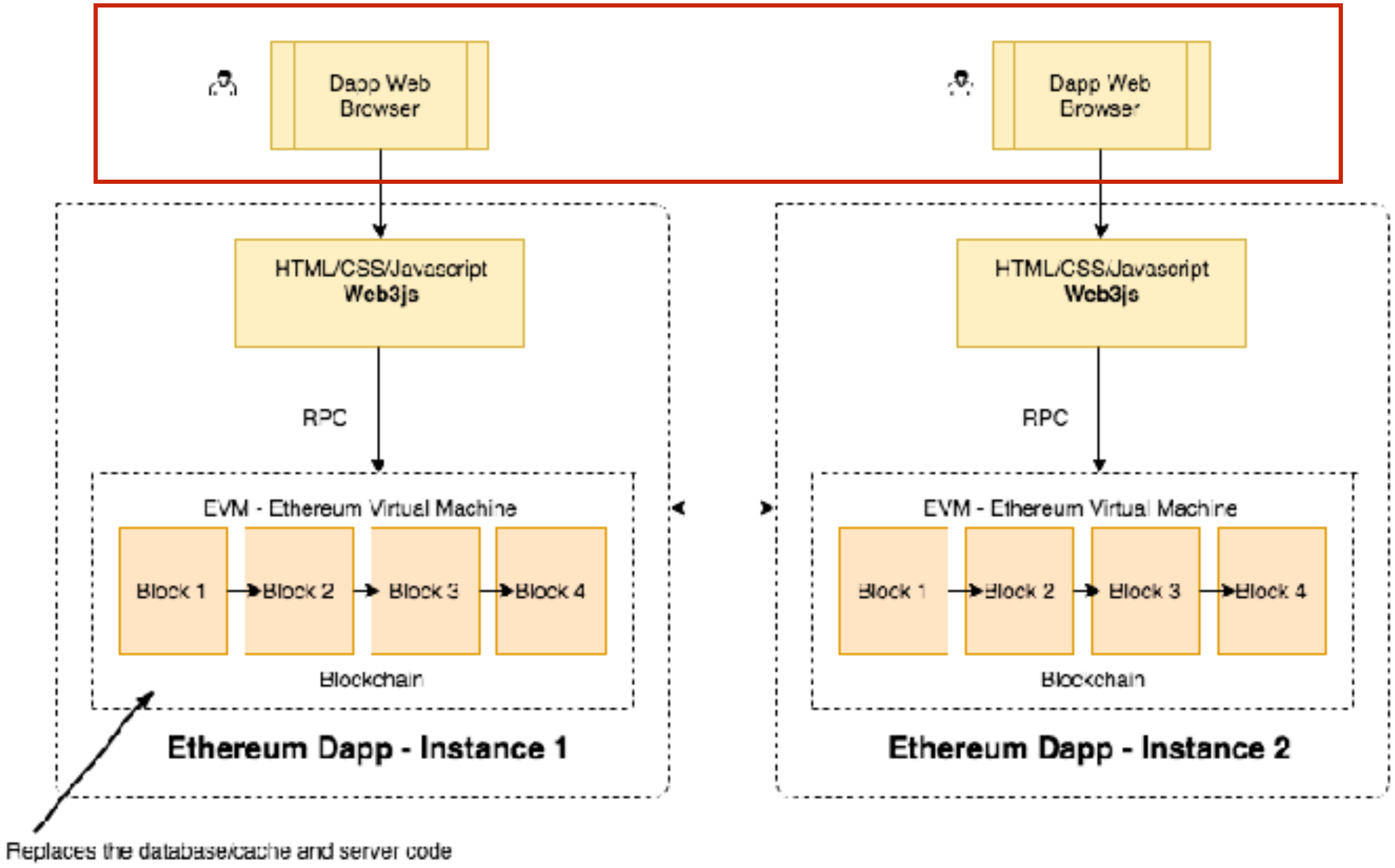


So it's not this…


Web Browser
Web Browser
Curl/Wget
API caller
HTML/CSS/Javascript
Server Code running Ruby, Python, Java etc
Database
Cache
Webapp hosted on AWS/Heroku etc

Mahesh Murthy (https://medium.com/@mvmurthy/ethereum-for-web-developers-890be23d1d0c)

More like this…



Mahesh Murthy (https://medium.com/@mvmurthy/ethereum-for-web-developers-890be23d1d0c)

Characteristics

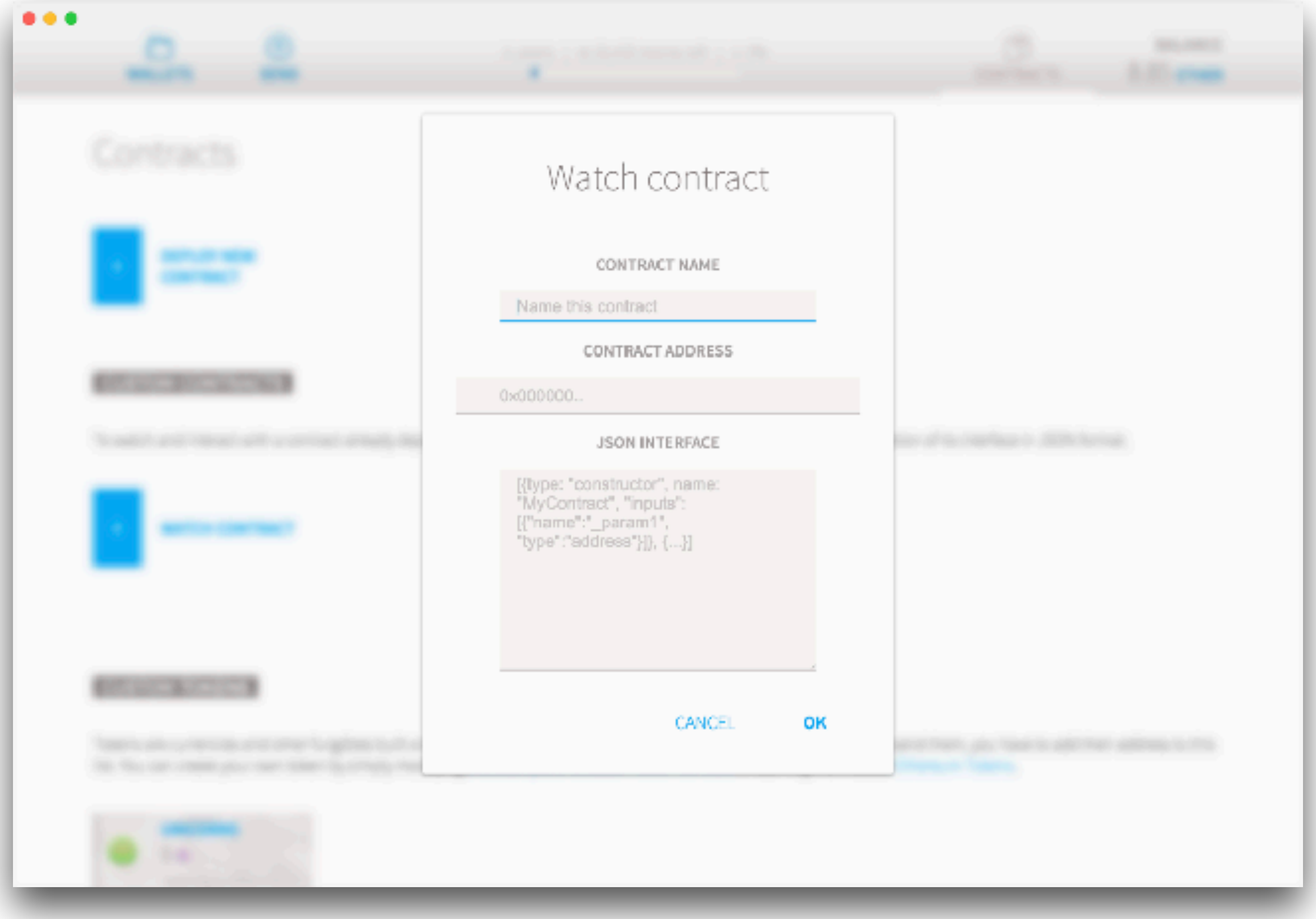
- Every client communicates with a blockchain instance (as opposed to a centralized server)
- You can’t do this through a standalone browser (without a plugin)…why?
  - *Because browsers can’t directly interact with an Ethereum wallet (yet)*
- This interaction is required to facilitate the “trust” (or proof that you own the wallet / are who you say you are)
- So…

User interaction with a DApp

Interacting with a DApp

- Wallet (to interact with a smart contract directly)
- Mist Browser
- Chrome via MetaMask

Interacting with a DApp - Wallet



Watch contract

CONTRACT NAME

Name this contract

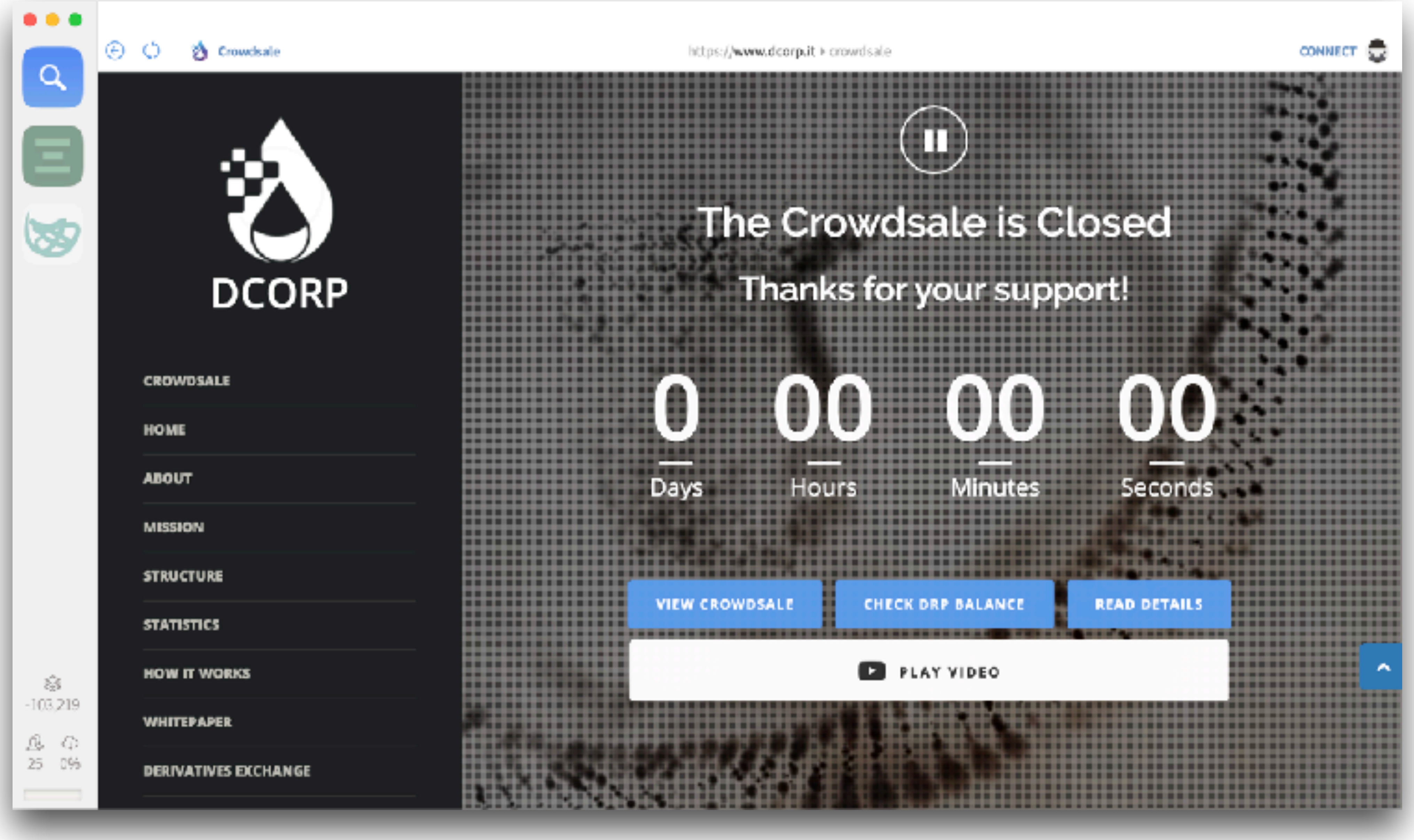
CONTRACT ADDRESS

0x000000..

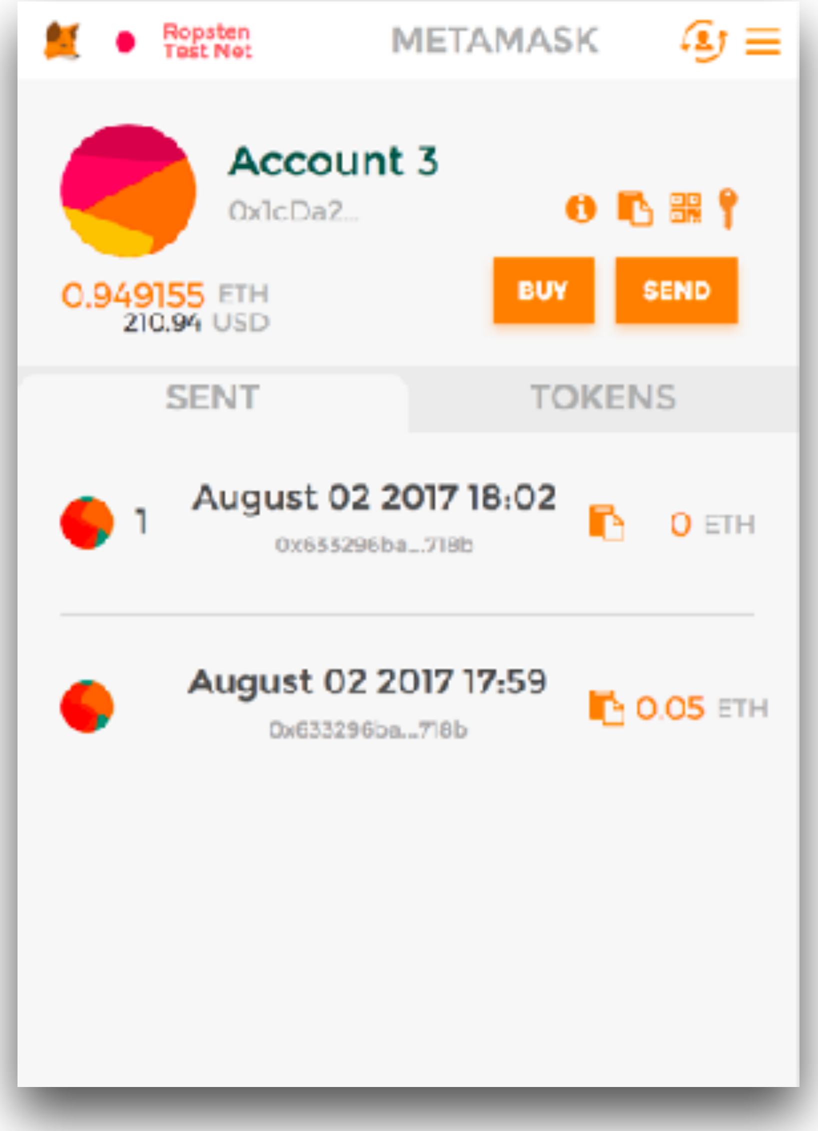
JSON INTERFACE

```
[{type: "constructor", name:
"MyContract", "inputs":
[{"name":"_param1",
"type":"address"}]}, {...}]
```

Interacting with a DApp - Mist



Interacting with a DApp - MetaMask


Ropsten Test Net METAMASK

Account 3
0x1cDa2...

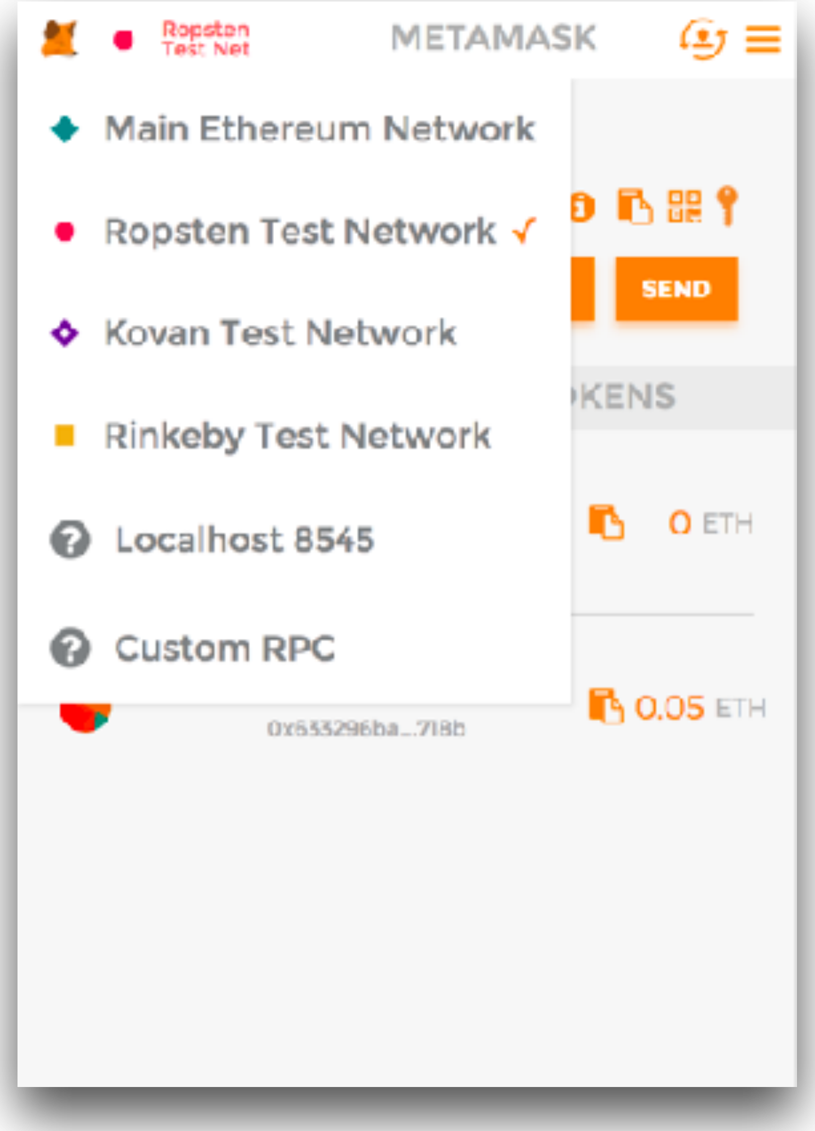
0.949155 ETH
210.94 USD

BUY SEND

SENT TOKENS

1 August 02 2017 18:02
0x633296ba...718b
0 ETH

August 02 2017 17:59
0x633296ba...718b
0.05 ETH


Ropsten Test Net METAMASK

Main Ethereum Network
Ropsten Test Network ✓
Kovan Test Network
Rinkeby Test Network
Localhost 8545
Custom RPC

SEND
KENS
0 ETH
0x633296ba...718b
0.05 ETH

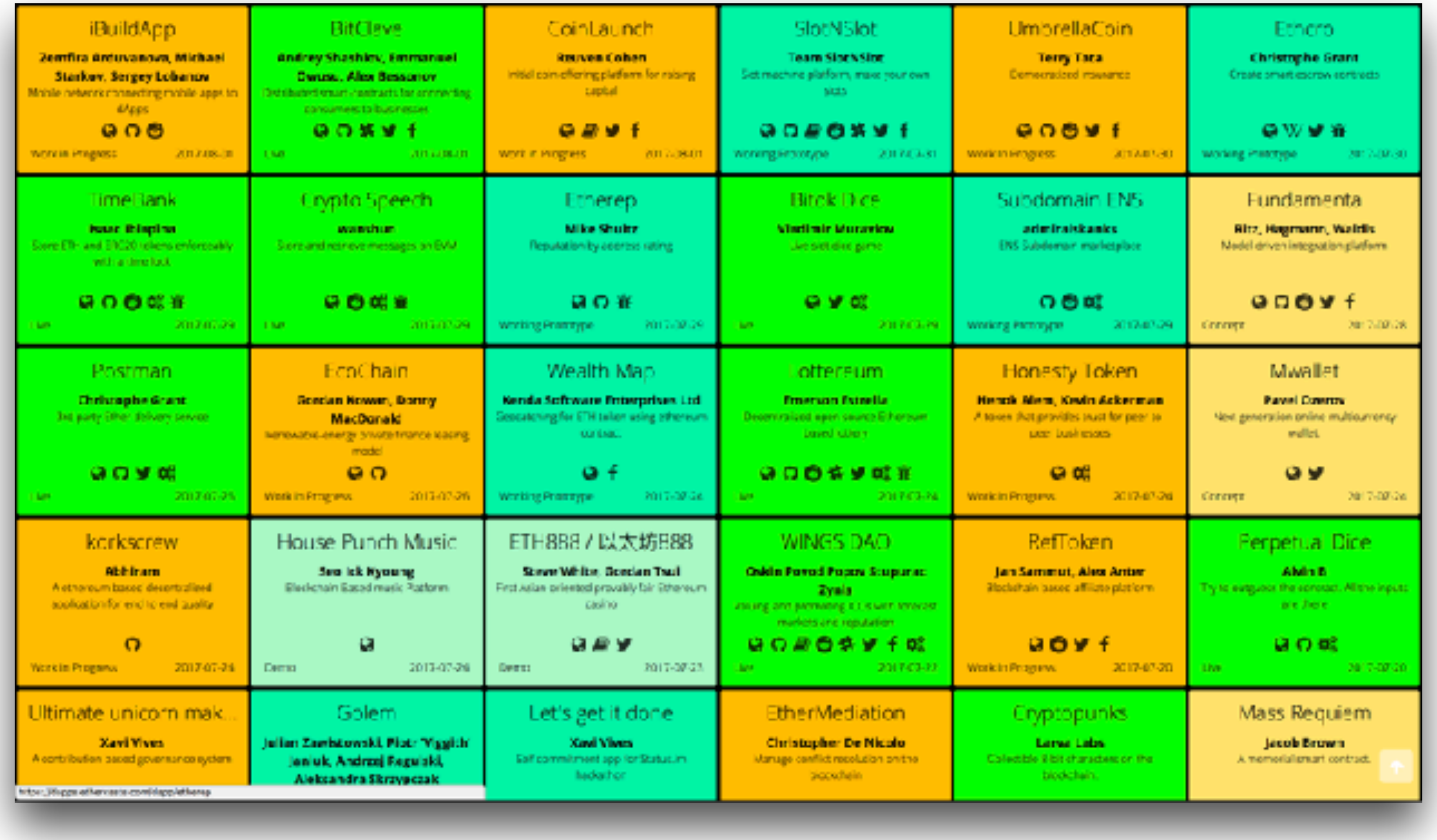
Interacting with a DApp - MetaMask

- “MetaMask is a bridge that allows you to visit the distributed web of tomorrow in your browser today. It allows you to run Ethereum dApps right in your browser without running a full Ethereum node.”
- Supports various networks (e.g. Main Ethereum, Testnets such as Ropsten, or local)
- https://metamask.io/
- https://medium.com/metamask
- https://github.com/MetaMask/metamask-plugin

Examples of DApps

Example DApps

- https://dapps.ethercasts.com



iBuildApp
Zemfira Arutyunova, Michael Starkov, Sergey Lobanov
Mobile network connecting mobile apps to dApps
Work in Progress [illegible]

BitCleve
Andrey Shashkov, Emmanuel Dwusu, Alex Bessonov
Distributed smart contracts for connecting consumers to businesses
Live [illegible]

CoinLaunch
Reuven Cohen
Initial coin offering platform for raising capital
Work in Progress [illegible]

SlotNSlot
Team SlotNSlot
Slot machine platform, make your own slots
Working Prototype [illegible]

UmbrellaCoin
Terry Tara
Decentralized insurance
Work in Progress [illegible]

Ethcro
Christophe Grant
Create smart escrow contracts
Working Prototype 2017-07-30

TimeBank
Isaac [illegible]
Store ETH and ERC20 tokens enforceably with a time lock
Live 2017-07-29

CryptoSpeech
wanchun
Store and retrieve messages on EVM
Live 2017-07-29

Etherep
Mike Shultz
Reputation by address rating
Working Prototype 2017-07-29

Block Dice
Vladimir Muravlev
Live slot dice game
Live 2017-07-29

Subdomain ENS
admiralsbanks
ENS Subdomain marketplace
Working Prototype 2017-07-29

Fundamenta
Ritz, Hagmann, Waldis
Model driven integration platform
Concept 2017-07-28

Postman
Christophe Grant
3rd party Ether delivery service
Live 2017-07-26

EcoChain
Gordon Nowen, Danny MacDonald
Renewable-energy private finance leasing model
Work in Progress 2017-07-26

Wealth Map
Kenda Software Enterprises Ltd
Geocaching for ETH token using ethereum contract
Working Prototype 2017-07-26

Lottereum
Emerson Estrella
Decentralized open source Ethereum based lottery
Live 2017-07-26

Honesty Token
Henrik Alem, Kevin Ackerman
A token that provides trust for peer to peer businesses
Work in Progress 2017-07-26

Mwallet
Pavel Dzerns
Next generation online multicurrency wallet.
Concept 2017-07-26

korkscrew
Abhiram
A ethereum based decentralized application for end to end audits
Work in Progress 2017-07-26

House Punch Music
Seo Ick Hyoung
Blockchain Based music Platform
Demo 2017-07-26

ETH888 / 以太坊888
Steve White, Gordon Tsai
First Asian oriented provably fair Ethereum casino
Demo 2017-07-23

WINGS DAO
[illegible]
Raising and permeating [illegible] forecast markets and reputation
Live 2017-07-22

RefToken
Jan Sammut, Alex Antar
Blockchain based affiliate platform
Work in Progress 2017-07-20

Perpetual Dice
Alvin B
Try to outguess the contract. All the inputs are there
Live 2017-07-20

Ultimate unicorn mak...
Xavi Vives
A contribution based governance system

Golem
Julian Zawistowski, Piotr 'Viggith' Janiuk, Andrzej Regulski, Aleksandra Skrzypczak

Let's get it done
Xavi Vives
Self commitment app for Status.im hackathon

EtherMediation
Christopher De Nicola
Manage conflict resolution on the blockchain

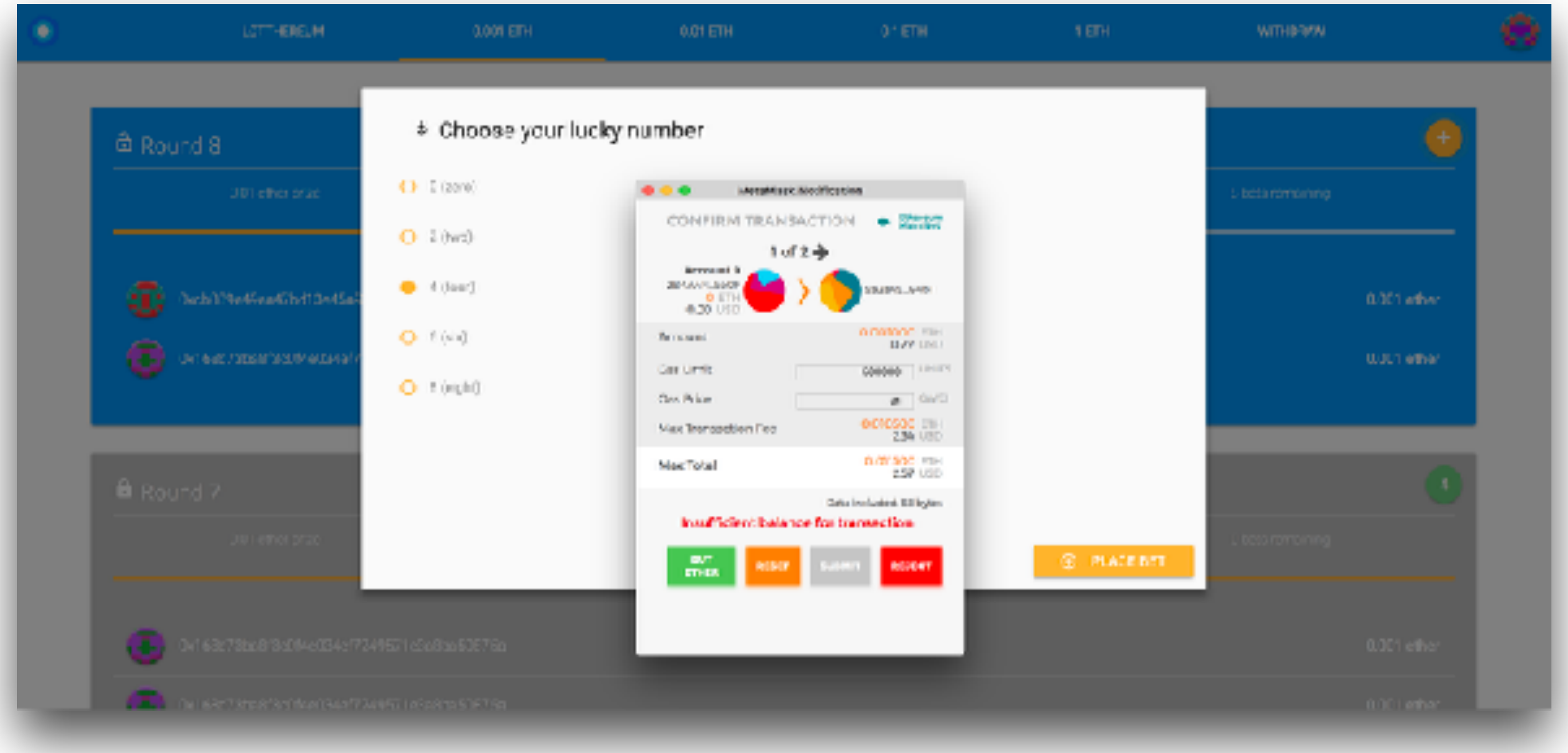
Cryptopunks
Larva Labs
Collectible 8-bit characters on the blockchain.

Mass Requiem
Jacob Brown
A memorial smart contract.

https://dapps.ethercasts.com/dapp/ethmap

Lotthereum

- https://lotthereum.github.io
- “Lotthereum is a decentralized open source Ethereum based lottery”

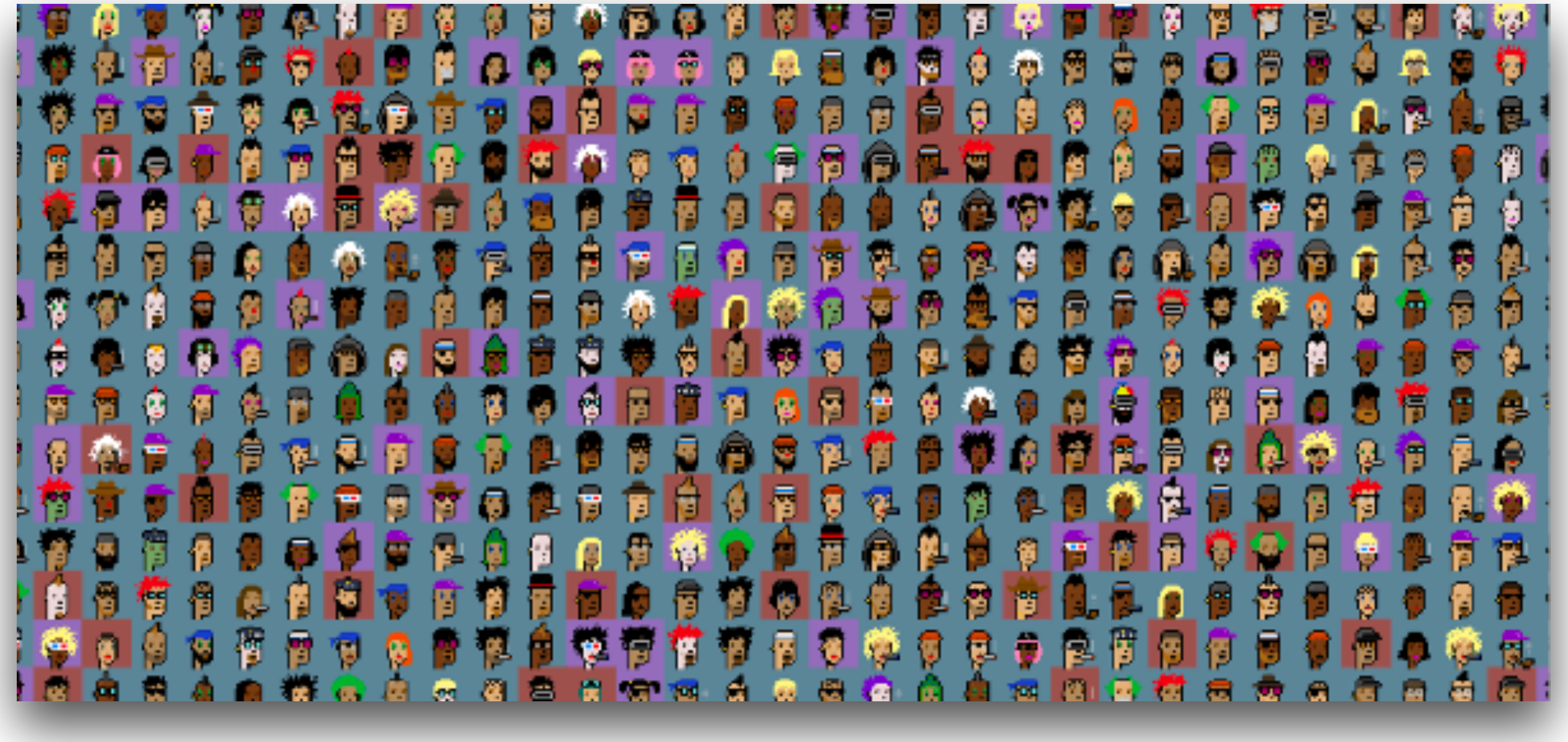


Choose your lucky number

Insufficient balance for transaction

CryptoPunks

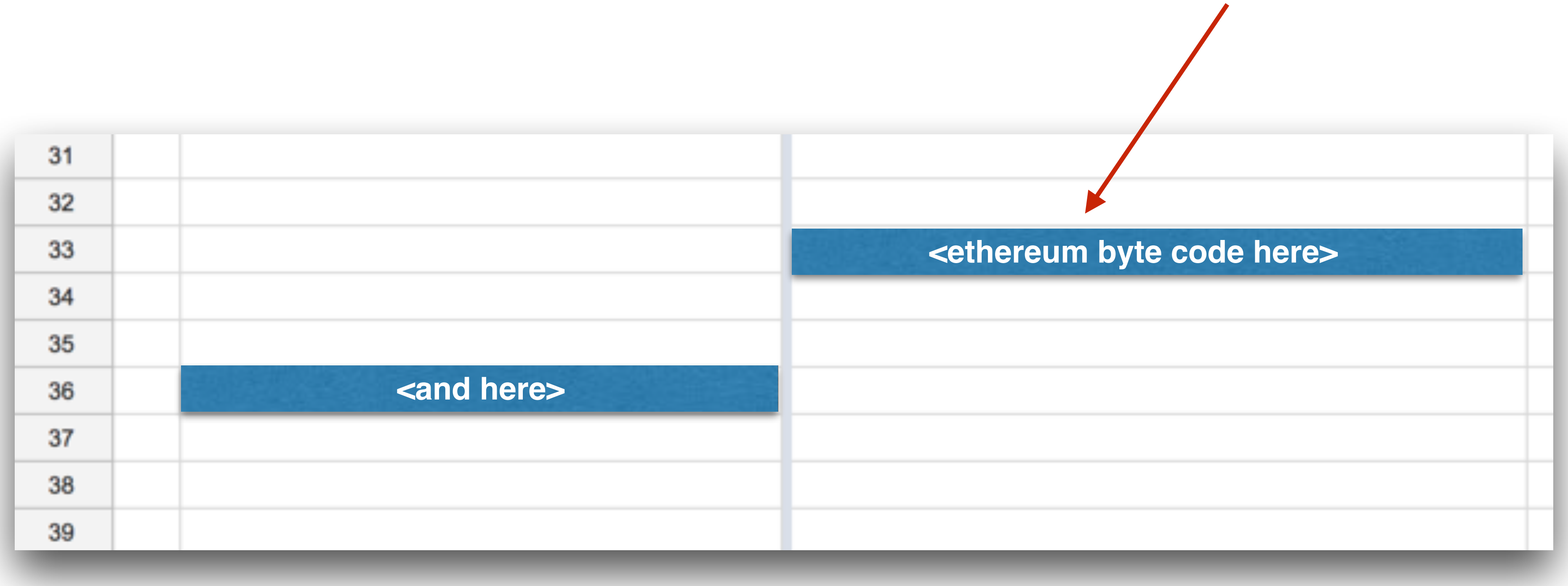
- http://www.larvalabs.com/cryptopunks
- “10,000 unique collectible characters with proof of ownership stored on the Ethereum blockchain.”



Stepping Back

Ethereum Blockchain

- Database - groups of transactions are packaged into blocks … each block is linked to the next one
- Code - all the business logic of your application, which is commonly called a “**smart contract**”



31			
32			
33			<ethereum byte code here>
34			
35			
36		<and here>	
37			
38			
39			

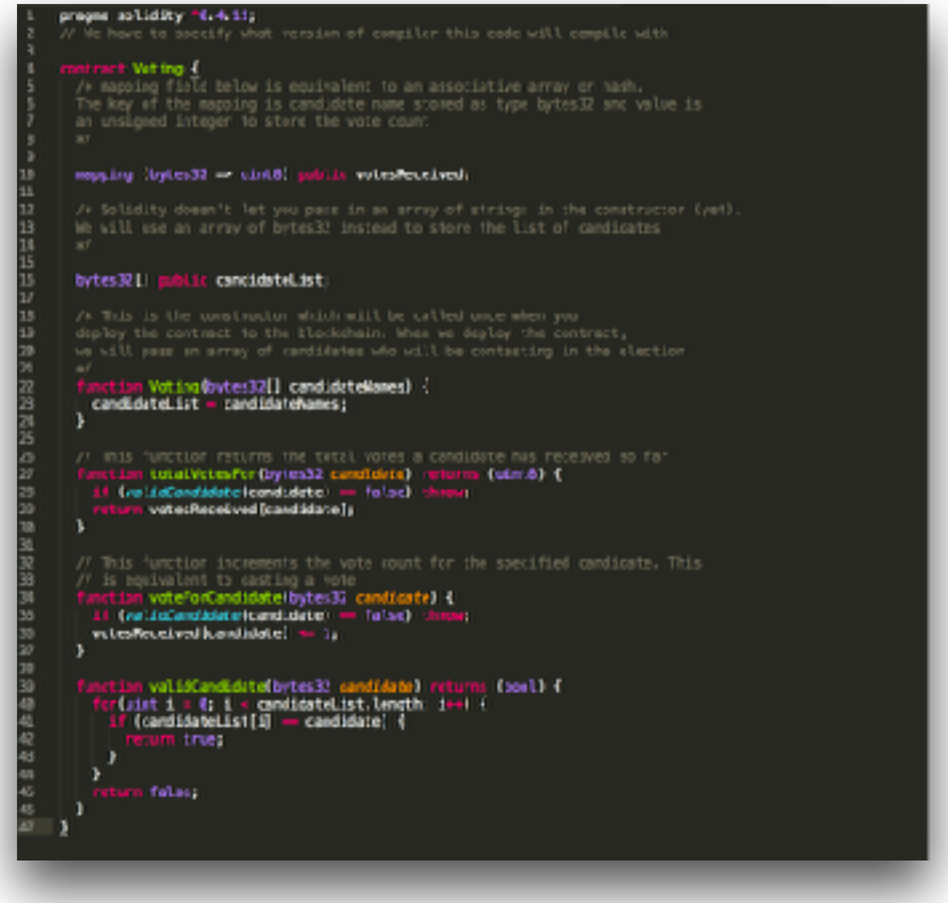
Writing a Smart Contract…3 things

- **Solidity** (most popular although there are others)
- You interface with the Smart Contract through a library called **Web3.js**
  - Referenced in your applications just as you would any other library (e.g. jQuery)
- **Truffle Framework** makes life a bit easier
- *Did you know? 100,000's of smart contracts already deployed :)*

Solidity

Solidity

- "Solidity is a contract-oriented, high-level language whose syntax is similar to that of JavaScript and it is designed to target the Ethereum Virtual Machine"



```
pragma solidity ^0.4.11;
// We have to specify what version of compiler this code will compile with

contract Voting {
  /* mapping field below is equivalent to an associative array or hash.
  The key of the mapping is candidate name stored as type bytes32 and value is
  an unsigned integer to store the vote count
  */

  mapping (bytes32 => uint8) public votesReceived;

  /* Solidity doesn't let you pass in an array of strings in the constructor (yet).
  We will use an array of bytes32 instead to store the list of candidates
  */

  bytes32[] public candidateList;

  /* This is the constructor which will be called once when you
  deploy the contract to the blockchain. When we deploy the contract,
  we will pass an array of candidates who will be contesting in the election
  */
  function Voting(bytes32[] candidateNames) {
    candidateList = candidateNames;
  }

  // This function returns the total votes a candidate has received so far
  function totalVotesFor(bytes32 candidate) returns (uint8) {
    if (validCandidate(candidate) == false) throw;
    return votesReceived[candidate];
  }

  // This function increments the vote count for the specified candidate. This
  // is equivalent to casting a vote
  function voteForCandidate(bytes32 candidate) {
    if (validCandidate(candidate) == false) throw;
    votesReceived[candidate] += 1;
  }

  function validCandidate(bytes32 candidate) returns (bool) {
    for(uint i = 0; i < candidateList.length; i++) {
      if (candidateList[i] == candidate) {
        return true;
      }
    }
    return false;
  }
}
```

Solidity

- JavaScript like syntax
- Object Oriented
- Contract
  - Think of it as a *class* in a OO language
- Contain a constructor which initializes with an array of candidates
- Can contain methods
  - Return total votes
  - Increment vote count

Example Contract
Voting.sol

Solidity

- Constructor invoked once and only once when the contract is deployed and initialized
- Updates to the code don’t change the original contract (or it’s data) / The new deployment will create a new instance of the contract

Web3

Web3.js

- “Ethereum JavaScript API”
- The bridge between your web application and the smart contract in Ethereum’s Blockchain
- “This is the Ethereum compatible JavaScript API which implements the Generic JSON RPC spec. It's available on npm as a node module, for bower and component as an embeddable js and as a meteor.js package.”

Truffle

Truffle Framework

- http://truffleframework.com/
- “Truffle is the most popular development framework for Ethereum with a mission to make your life a whole lot easier.”
- Features
  - Smart Contract Compilation, Deployment, etc
  - Scriptable Deployment to Test / Private / Public Networks

Development Envs

Building DApps

- Localhost (TestRPC)
- TestNet (Ropsten)
- Main Network

TestRPC

- “testrpc is a Node.js based Ethereum client for testing and development. It uses ethereumjs to simulate full client behavior and make developing Ethereum applications much faster. It also includes all popular RPC functions and features (like events) and can be run deterministically to make development a breeze.”
- > testrpc
  - Loaded with 10x accounts (each w/ 100 ETH)
- > geth attach http://localhost:8545
- Or via truffle (rather than Geth)

Testnets

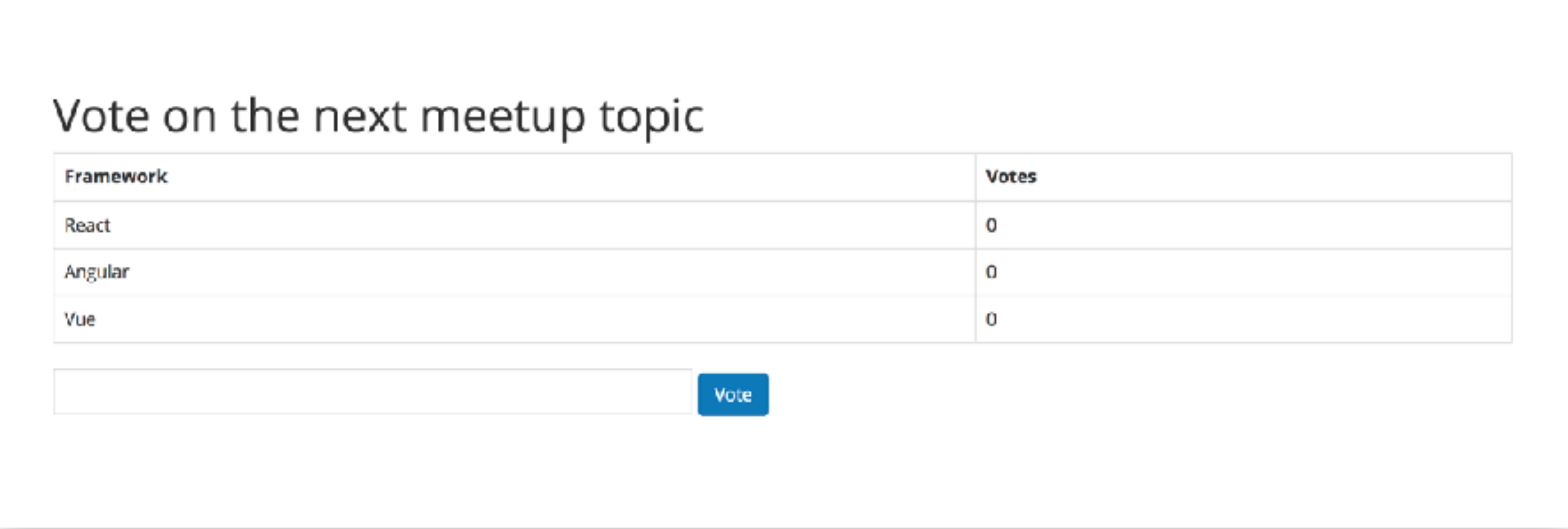
- https://ropsten.etherscan.io/
- https://ropsten.etherscan.io/address/0x1cda2ea9673146dc4bf55662fe14bef11c22ea78

Main Network

- https://etherscan.io/
- https://etherscan.io/address/0x830e3a6766c753e041aa5b78e94213972a99d400

Hello World

Hello World



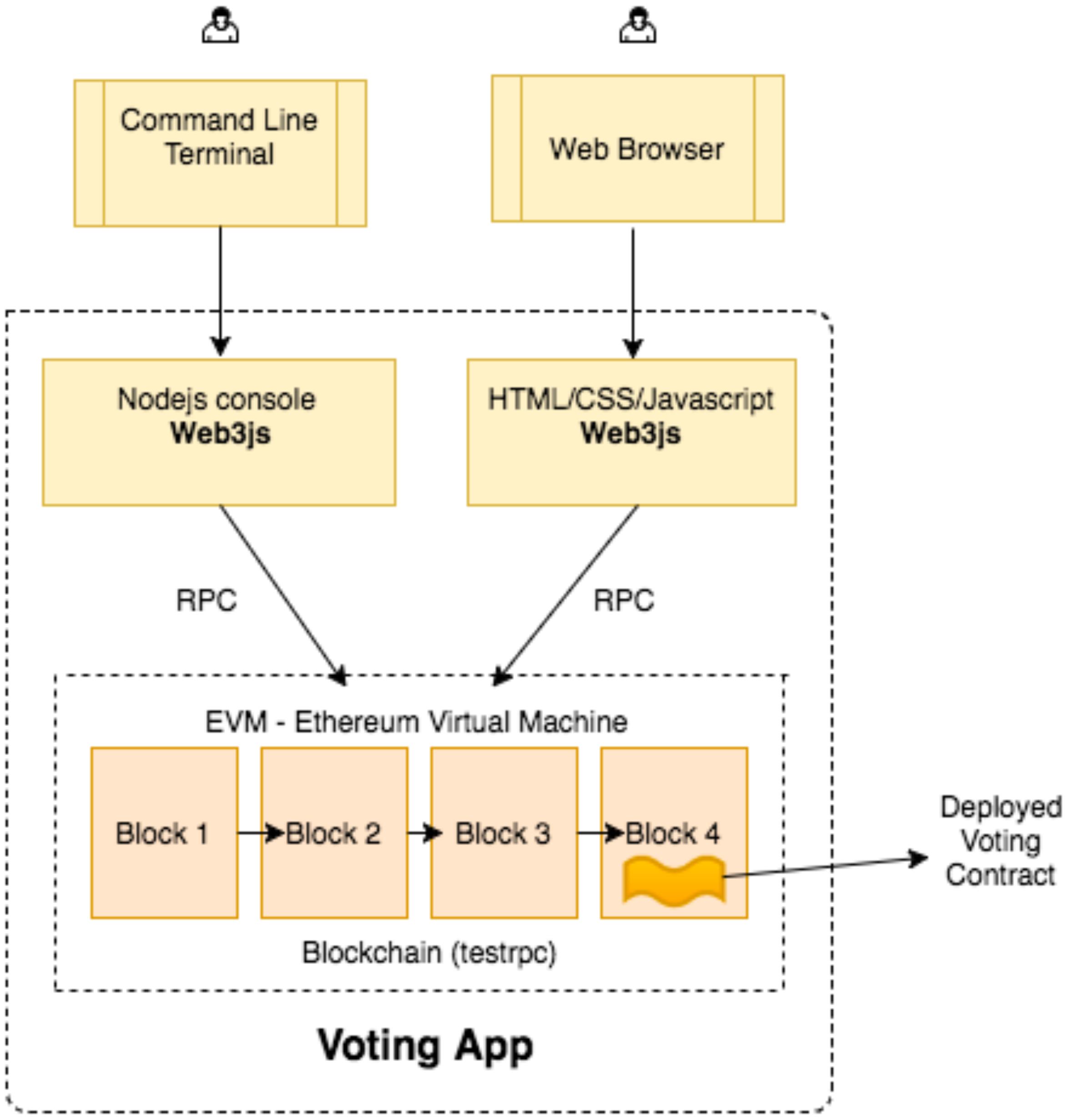
Vote on the next meetup topic

Framework	Votes
React	0
Angular	0
Vue	0

Things to note...

- The backend is Ethereum’s blockchain
- In this case a local test network (e.g. dev env)
- Web3.js is used as the bridge from the client to the blockchain
- MetaMask is used to connect a wallet

What’s Going On?



Setup Steps

1.> npm run dev

2.> testrpc

3.Take one of the test accounts, add to the truffle.js and…

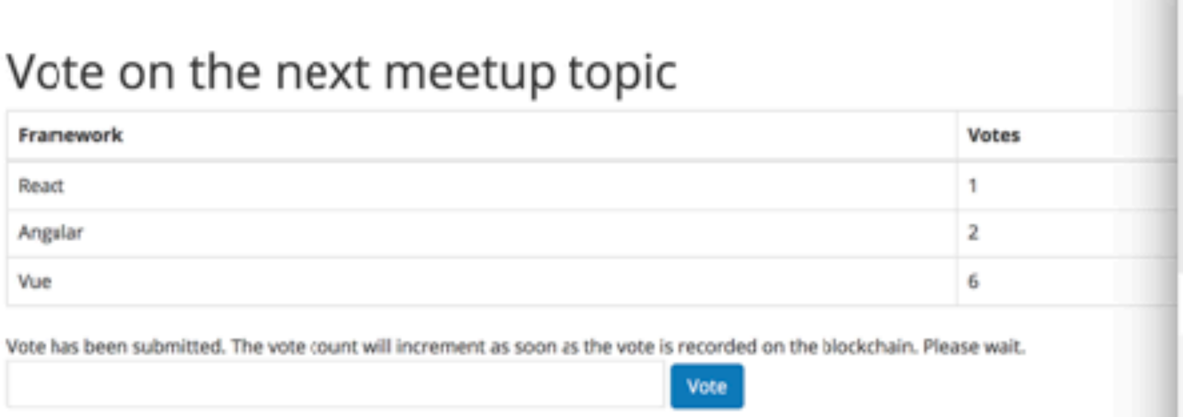
4.> truffle migrate

5.Ensure MetaMask is point a local “Private Network”. Note that the browser refreshes and the votes are now shown

6.Conduct a vote and confirm transaction

ETH + Gas

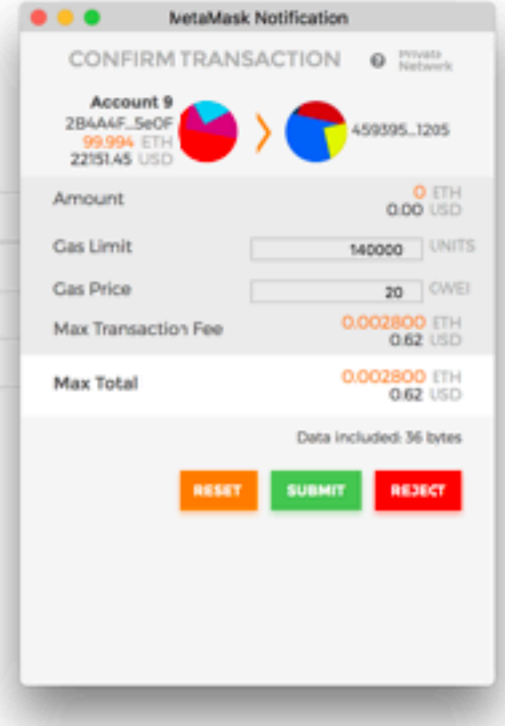
- "It costs money to interact with the blockchain. This money goes to miners who do all the work to include your code in the blockchain."



Vote on the next meetup topic

Framework	Votes
React	1
Angular	2
Vue	6

Vote has been submitted. The vote count will increment as soon as the vote is recorded on the blockchain. Please wait.



Account 9
2B4A4F...5e0F
99.994 ETH
22151.45 USD

459395...1205

Amount: 0 ETH, 0.00 USD
Gas Limit: 140000 UNITS
Gas Price: 20 GWEI
Max Transaction Fee: 0.002800 ETH, 0.62 USD
Max Total: 0.002800 ETH, 0.62 USD

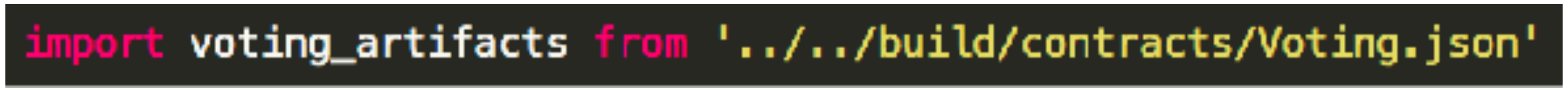
Data included: 36 bytes

Units

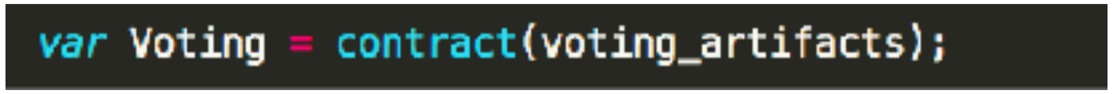
```
var unitMap = {
    'noether':      '0',
    'wei':          '1',
    'kwei':         '1000',
    'Kwei':         '1000',
    'babbage':      '1000',
    'femtoether':   '1000',
    'mwei':         '1000000',
    'Mwei':         '1000000',
    'lovelace':     '1000000',
    'picoether':    '1000000',
    'gwei':         '1000000000',
    'Gwei':         '1000000000',
    'shannon':      '1000000000',
    'nanoether':    '1000000000',
    'nano':         '1000000000',
    'szabo':        '1000000000000',
    'microether':   '1000000000000',
    'micro':        '1000000000000',
    'finney':       '1000000000000000',
    'milliether':    '1000000000000000',
    'milli':         '1000000000000000',
    'ether':        '1000000000000000000',
    'kether':       '100000000000a0000000000',
    'grand':        '1000000000000000000000',
    'mether':       '1000000000000000000000000',
    'gether':       '1000000000000000000000000000',
    'tether':       '1000000000000000000000000000000'
};
```

Reviewing the Web3.js

- The ABI (Application Binary Interface)

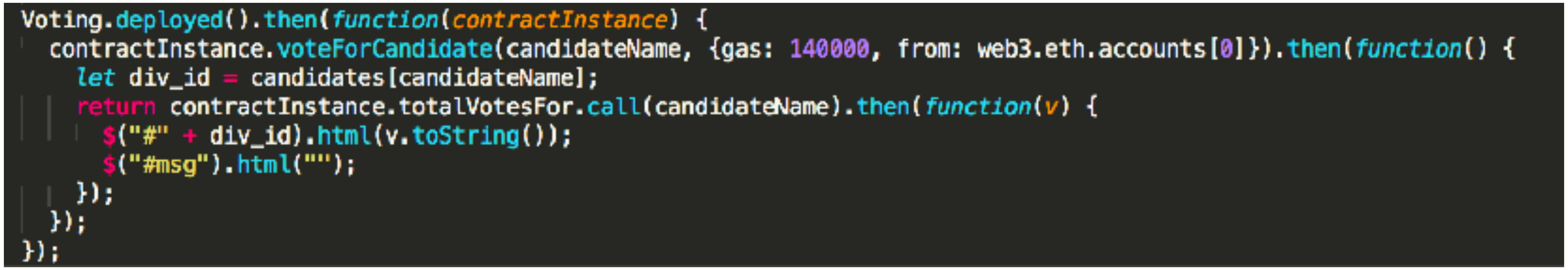


```
import voting_artifacts from '../../build/contracts/Voting.json'
```



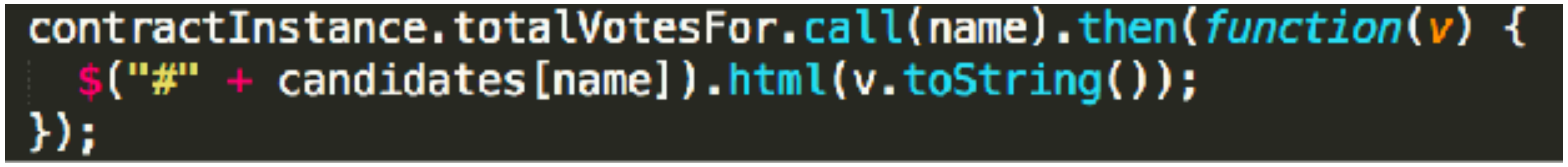
```
var Voting = contract(voting_artifacts);
```

- Voting (note the name, gas, and account params)



```
Voting.deployed().then(function(contractInstance) {
  contractInstance.voteForCandidate(candidateName, {gas: 140000, from: web3.eth.accounts[0]}).then(function() {
    let div_id = candidates[candidateName];
    return contractInstance.totalVotesFor.call(candidateName).then(function(v) {
      $("#" + div_id).html(v.toString());
      $("#msg").html("");
    });
  });
});
```

- Retrieving the current votes (reading doesn't need gas)



```
contractInstance.totalVotesFor.call(name).then(function(v) {
  $("#" + candidates[name]).html(v.toString());
});
```

More Sophisticated Example

Decentralized Voting Application

- https://www.zastrin.com/simple-ethereum-voting-dapp.html
- Purchase some tokens (already done)
- Vote on a candidate
- *0x1cda2ea9673146dc4bf55662fe14bef11c22ea78*

Thank You / Q&A