Hong Kong JavaScript and Node.js

Welcome 🙌

Agenda

Agenda

- Housekeeping
- **Blockchains & JavaScript**, Kevin Bluer
- **Graph DB and Node.js - Building StackOverflow Clone**, Song Cho
- **Introduction to Promises**, Kareem Elshahawe
- end();

Housekeeping

Thank You Mettā

- http://meta.co

METTĀ

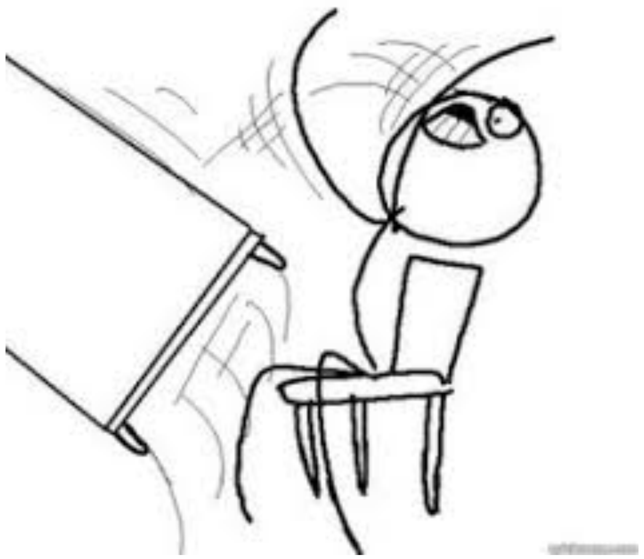
Blockchains & JavaScript

Kevin Bluer

Session Agenda

Caveat…

- Changing / evolving so fast
- There’s so much out there
- I’m still learning it all!



Session Agenda

- Goals
- WTF are blockchains anyway?
- Review four major platforms (Bitcoin, IPFS, Hyperledger, IPFS, and Ethereum)
- Smart Contract (and DApp) in Ethereum
- Summary + Q&A

Session Goals

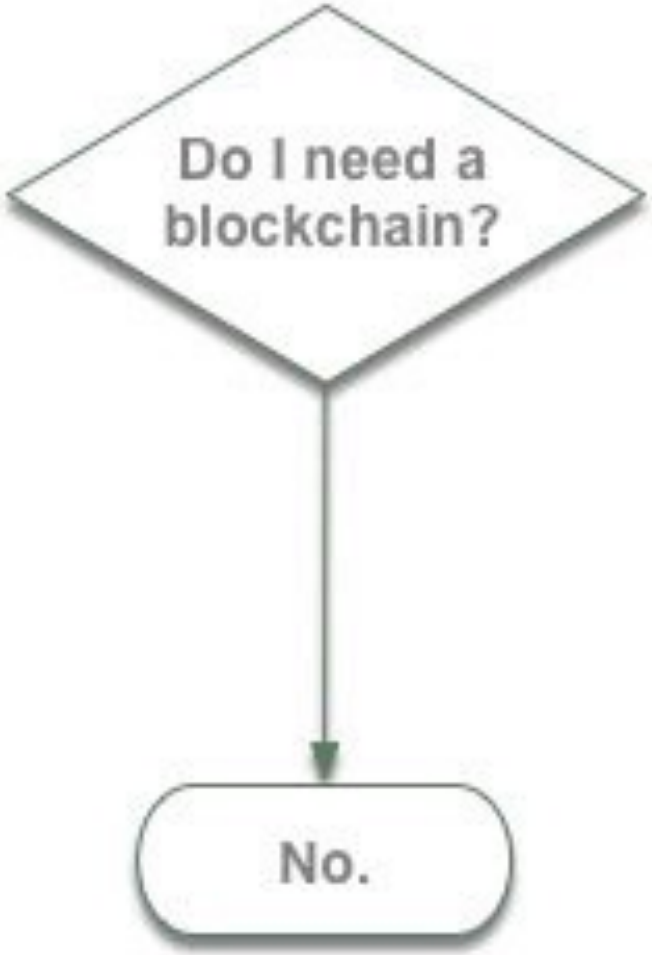
- Come away with an appreciation of blockchain(s)
- Understand types of applications, uses, etc
- **See how to apply to your day-to-day AND via JS**

Show of Hands

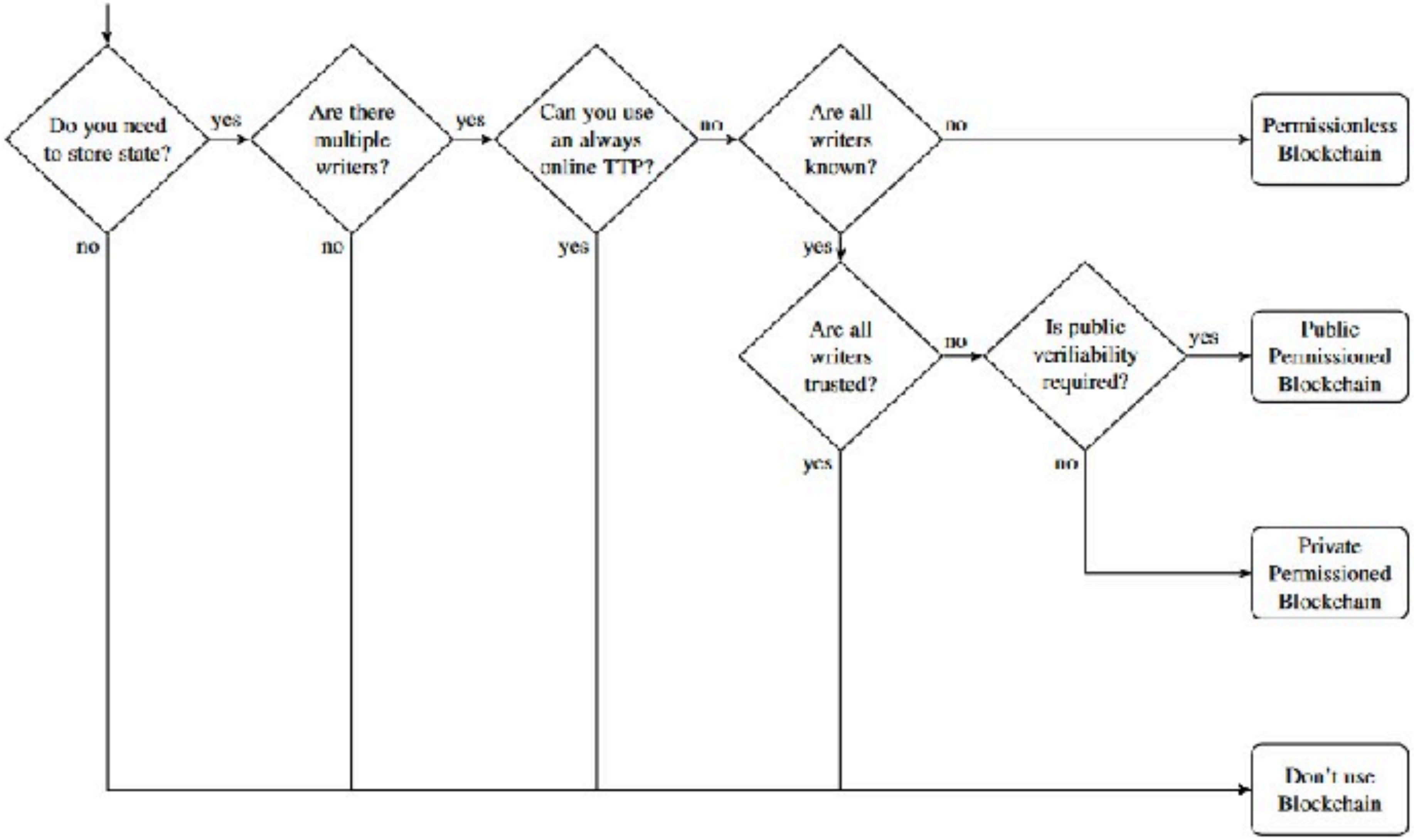
- Heard of blockchain?
- Can describe blockchain?
- Heard of BTC, ETC, LTC? Bought or mined?
- Heard of Smart Contracts, Dapps, Mist, Solidity, Web3, Hyperledger, Fabric, FTA?

Blockchain(s)

Do I / you need a blockchain?



Really, though?


Do you need to store state?
yes
no
Are there multiple writers?
yes
no
Can you use an always online TTP?
no
yes
Are all writers known?
no
yes
Permissionless Blockchain
Are all writers trusted?
no
yes
Is public verifiability required?
yes
no
Public Permissioned Blockchain
Private Permissioned Blockchain
Don't use Blockchain

“When do you need blockchain? Decision models.”, **Sebastien Meunier**

Overview

Overview

- **“A blockchain is a shared ledger that everyone trusts to be accurate forever”**, David Siegel
- Data Store (think of ledger)
- Decentralized (no single owner, but incentives to participate)
- Replicated (BitTorrent)
- Trustless (depending on the platform)
- Programmable (depending on the platform)

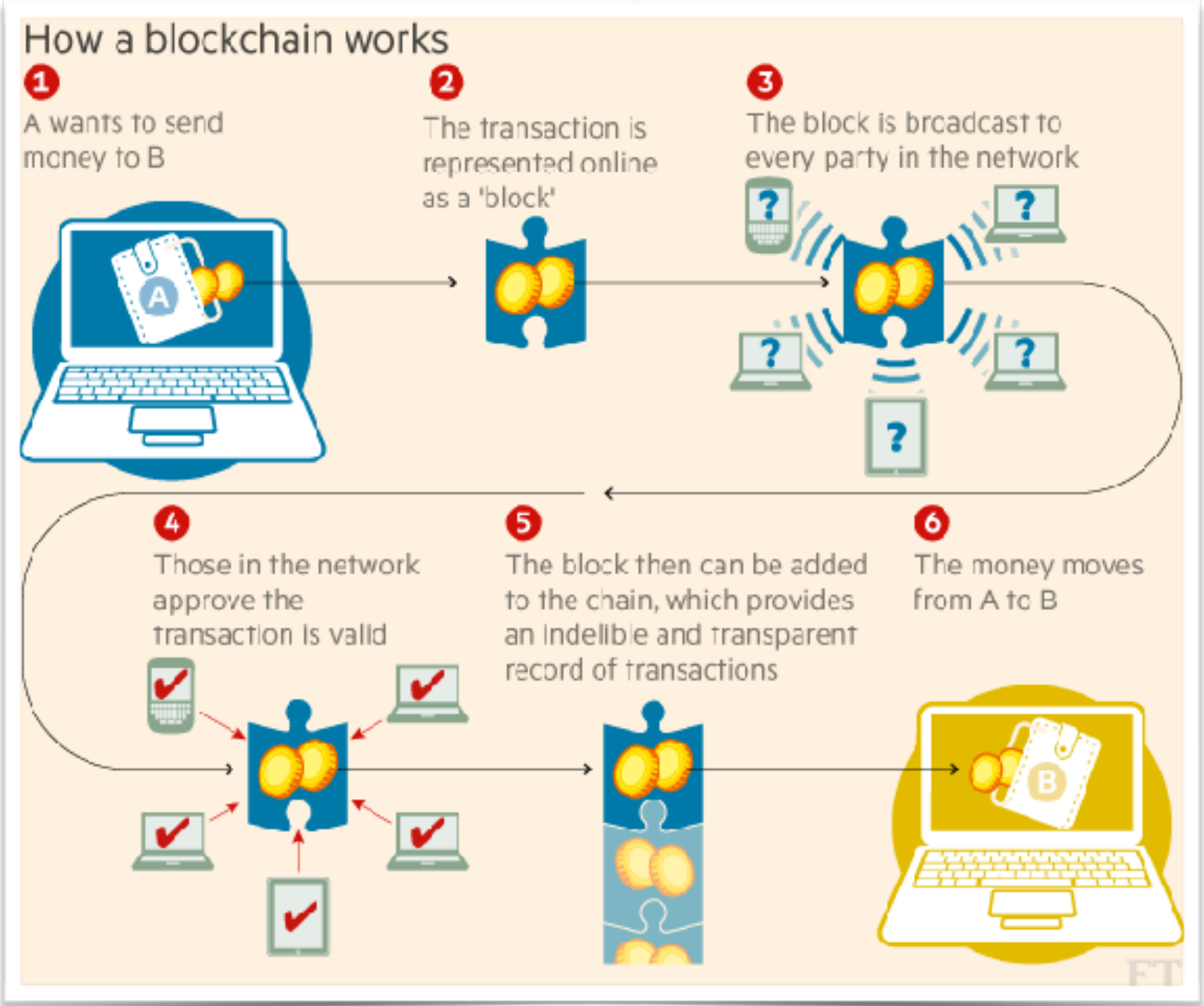
Applications / Use Cases

- Currencies
- Public / Private Ledgers (e.g. Honduran Land Reg.)
- Raising Capital via ICOs
- DAO (Decentralized Autonomous Organization)
- Voting Systems

Platforms

- Bitcoin, Litecoin, etc
- Ethereum
- Hyperledger
- IPFS
- Etc

Cryptocurrency Example



How a blockchain works

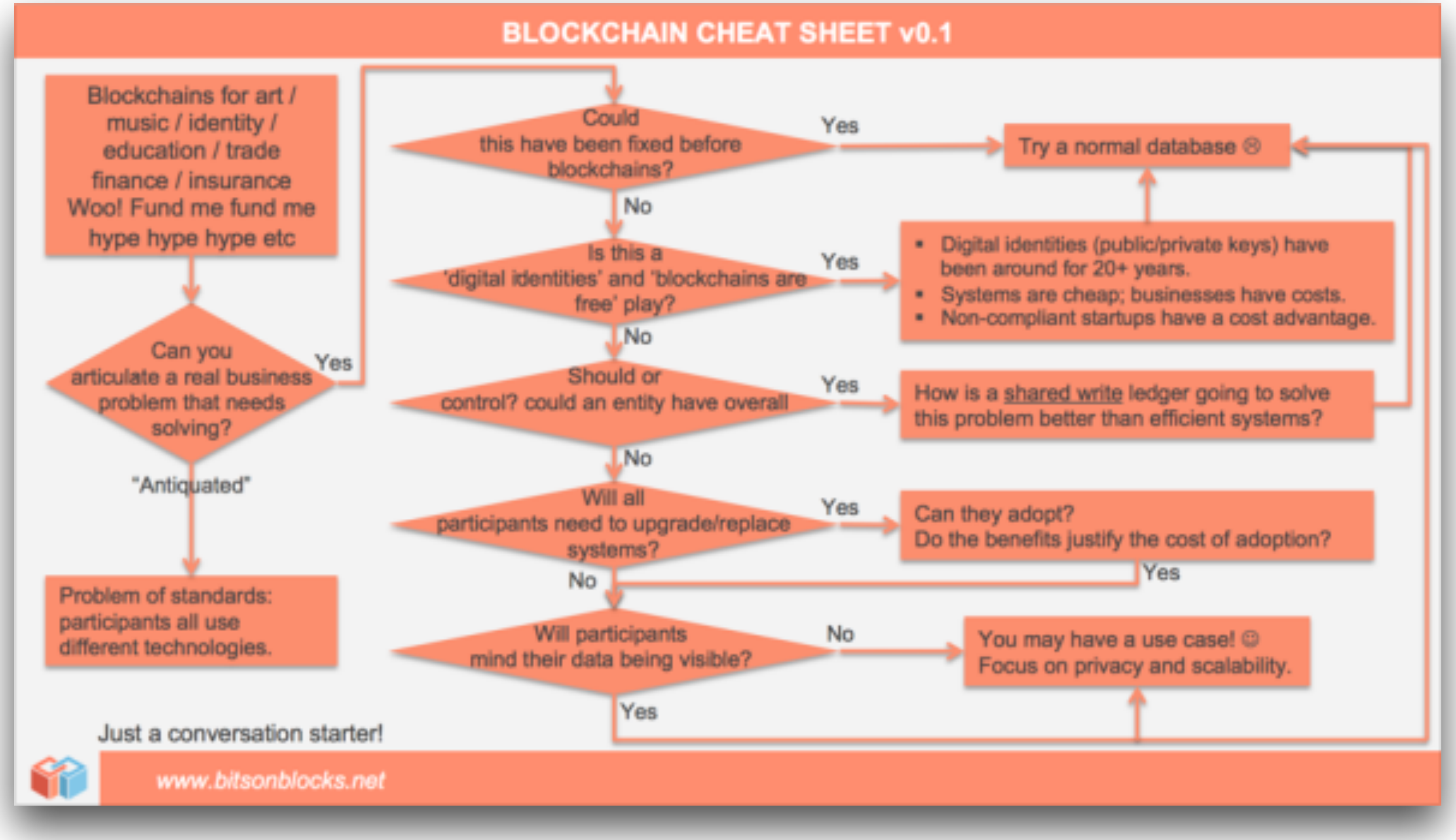
1. A wants to send money to B
2. The transaction is represented online as a 'block'
3. The block is broadcast to every party in the network
4. Those in the network approve the transaction is valid
5. The block then can be added to the chain, which provides an indelible and transparent record of transactions
6. The money moves from A to B

The Good News

- Nascent space + lots of opportunity
- Lots of JS / Node libraries out there
- Wild potential!
  - “Once you see the world from a more adaptive, agile mindset, you’ll see that the blockchain really is going to change everything forever.”

The Less Good News

- “It’s complicated” :) Or at least it can be…



BLOCKCHAIN CHEAT SHEET v0.1

Blockchains for art / music / identity / education / trade finance / insurance Woo! Fund me fund me hype hype hype etc

Can you articulate a real business problem that needs solving?

Yes

"Antiquated"

Problem of standards: participants all use different technologies.

Could this have been fixed before blockchains?

Yes

Try a normal database ☹

No

Is this a 'digital identities' and 'blockchains are free' play?

Yes

- Digital identities (public/private keys) have been around for 20+ years.
- Systems are cheap; businesses have costs.
- Non-compliant startups have a cost advantage.

No

Should or could an entity have overall control?

Yes

How is a shared write ledger going to solve this problem better than efficient systems?

No

Will all participants need to upgrade/replace systems?

Yes

Can they adopt? Do the benefits justify the cost of adoption?

Yes

No

Will participants mind their data being visible?

No

You may have a use case! ☺ Focus on privacy and scalability.

Yes

Just a conversation starter!

www.bitsonblocks.net

Quick Dissection

BTC Transaction Example

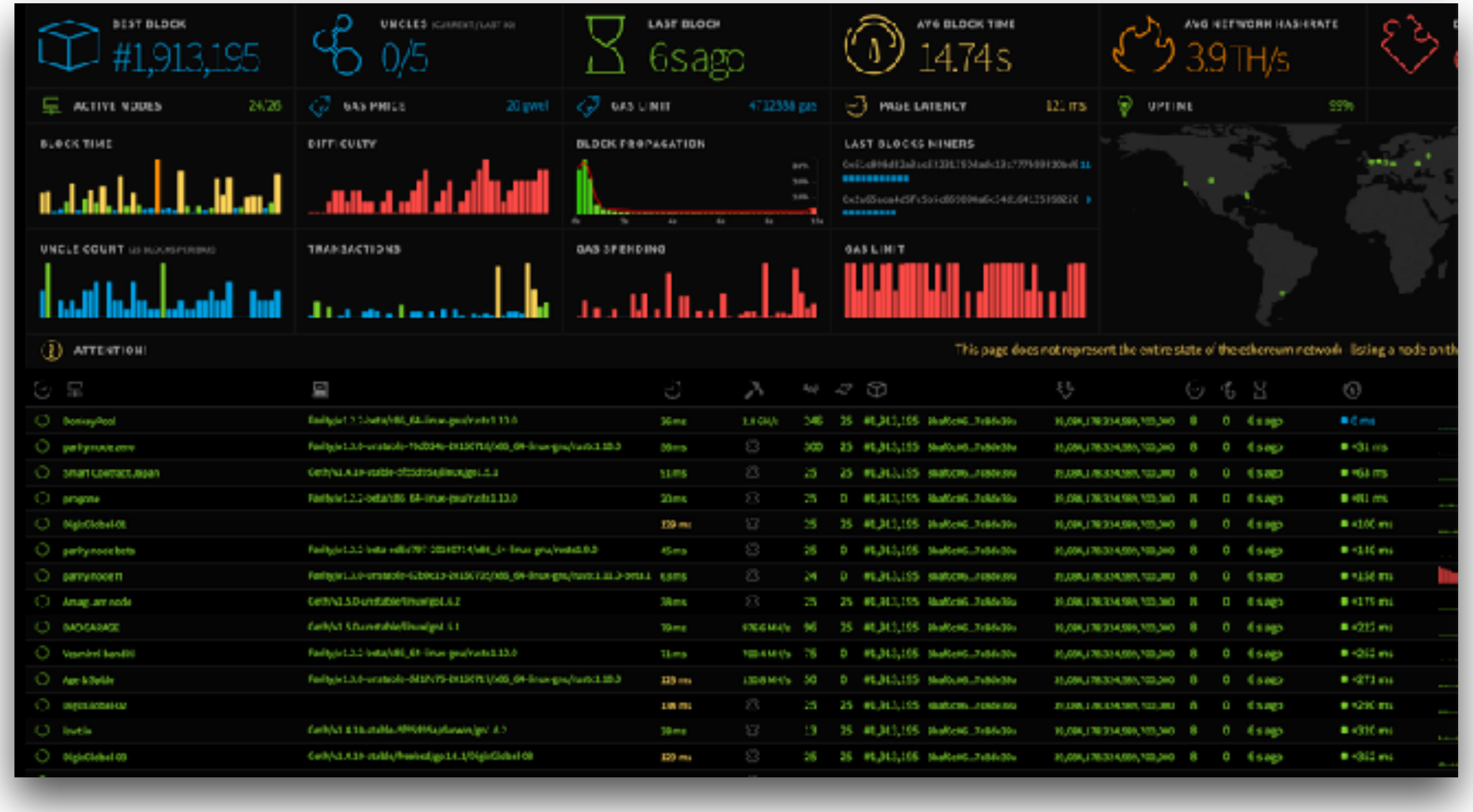
```
{
 "hash":"f4184fc596403b9d638783cf57adfe4c75c605f6356fbc91338530e9831e9e16",
 "ver":1,
 "vin_sz":1,
 "vout_sz":2,
 "lock_time":0,
 "size":275,
 "in":[
   {
     "prev_out":{
       "hash":"0437cd7f8525ceed2324359c2d0ba26006d92d856a9c20fa0241106ee5a597c9",
       "n":0
     },

"scriptSig":"304402204e45e16932b8af514961a1d3a1a25fdf3f4f7732e9d624c6c61548ab5fb8cd410220181522ec8eca07de4860a4acdd12909d831cc56cbbac46
22082221a8768d1d0901"
   }
 ],
 "out":[
   {
     "value":"10.00000000",

"scriptPubKey":"04ae1a62fe09c5f51b13905f07f06b99a2f7159b2225f374cd378d71302fa28414e7aab37397f554a7df5f142c21c1b7303b8a0626f1baded5c72a70
4f7e6cd84c OP_CHECKSIG"
   },
   {
     "value":"40.00000000",

"scriptPubKey":"0411db93e1dcdb8a016b49840f8c53bc1eb68a382e97b1482ecad7b148a6909a5cb2e0eaddfb84ccf9744464f82e160bfa9b8b64f9d4c03f999b8
643f656b412a3 OP_CHECKSIG"
   }
 ]
}
```

Block (Group of Tx) Examples

- Bitcoin Block - https://blockchain.info/block/00000000000000000062e8d7d9b7083ea45346d7f8c091164c313eeda2ce5db11
- Ethereum Block - https://etherscan.io/block/3931138

Mining

- Means of incentivizing participation (via “rewards”)

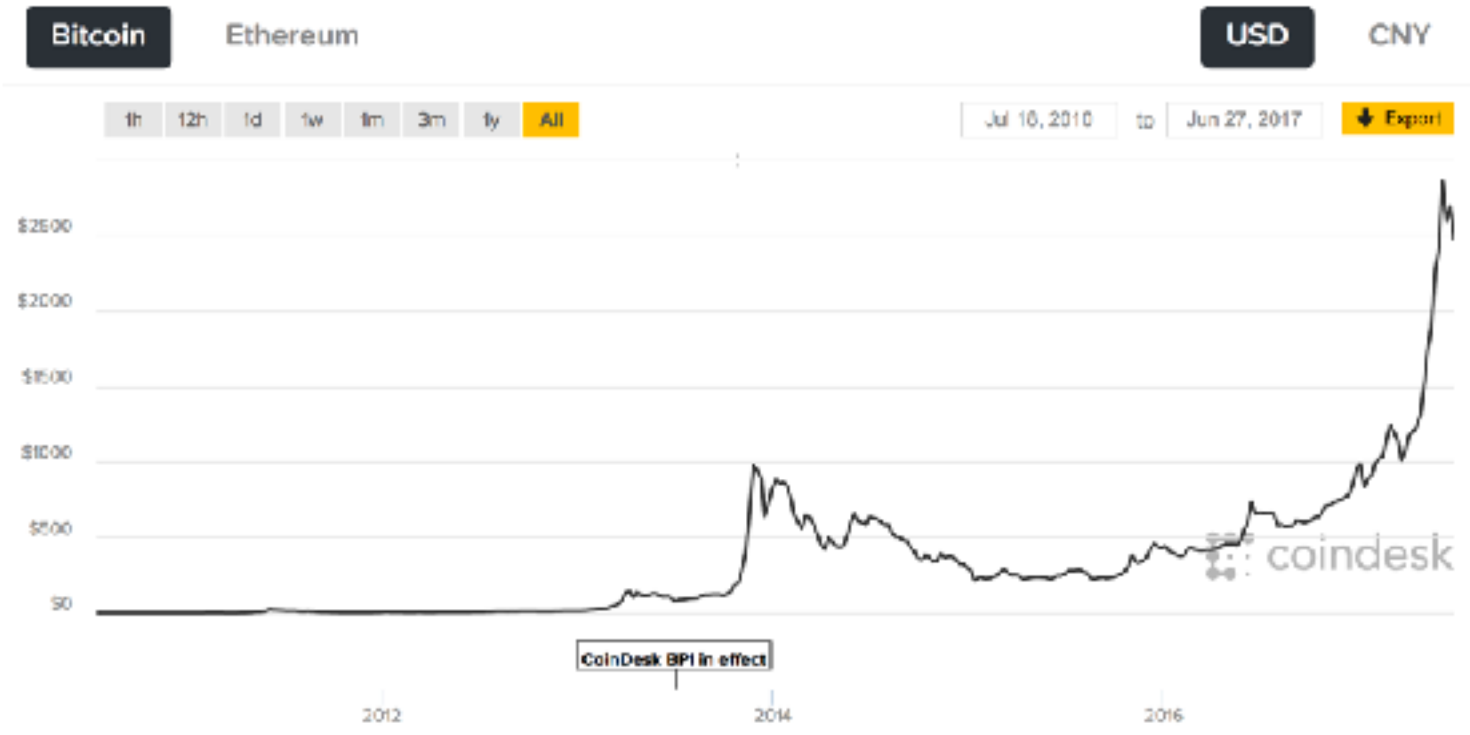


https://ethstats.net

Bitcoin

Bitcoin Overview

- Cryptocurrency (obviously)
- Currently on fire, *I think*!
- Market capitalization: $41,330,152,143.00 📈📉



“Programming” with Bitcoin

“Programming” with Bitcoin

- Not programmable per se (given it’s purely a ledger)
- However, you can programmatically…
  - Generate addresses
  - Create transactions
  - Store data (via OP RETURN and other techniques … examples here and below)

BitcoinJS

Bitcoin + JS = BitcoinJS

- https://bitcoinjs.org
- “A pure JavaScript Bitcoin library for node.js and browsers. **Used in production by over 1.5 million wallet users**, BitcoinJS is the backbone for almost all Bitcoin web wallets in production today.”

blockchain.info API

blockchain.info API

- https://blockchain.info/api
- https://github.com/blockchain/api-v1-client-node
  - **MyWallet** - Interact with or create a Blockchain Wallet
  - **blockexplorer** - View data for addresses, blocks, transactions, and more
  - **exchange** - Get real-time bitcoin exchange rates
  - **pushtx** - Push custom transactions
  - **Receive** - Receive notifications for payments
  - **Socket** - Live notifications for transactions and blocks
  - **statistics** - Fetch historical blockchain data and statistics

IPFS (and equivalents)

IPFS Overview

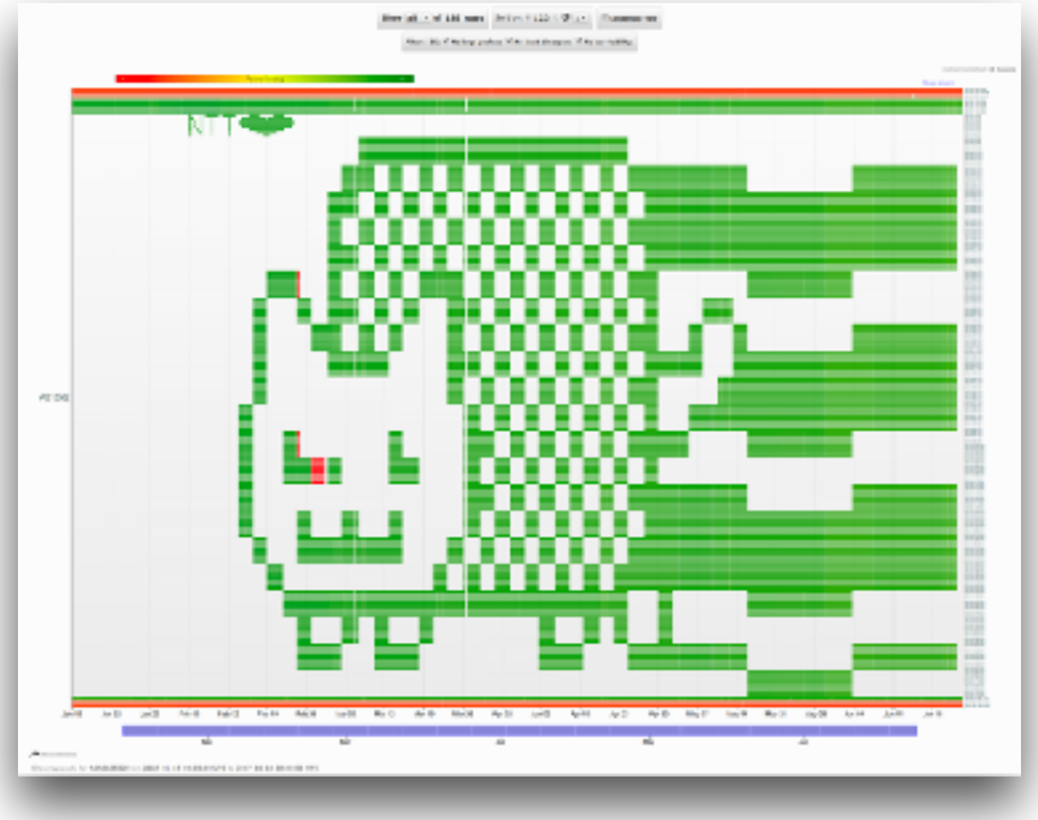
- **InterPlanetary File System**
- “A peer-to-peer hypermedia protocol
  to make the web faster, safer, and more open.”
- “The average lifespan of a web page is 100 days.”
- “IPFS aims to replace HTTP and build a better web for all of us.”

IPFS (and equivalents)

- Instead of a central server, a peer to peer network is used to establish connections. Public key cryptography is built into the node addressing system and content addressing is used to index content. Both node and content addresses are stored in a decentralized naming system called IPNS.

Example

- https://gateway.ipfs.io/ipfs/QmcnTG8JBgvp3gAD7aDFVqZ53bYKfwadXwdr5YvW8HYwi5



Alternatives

- Swarm
- Storj
- Maidsafe

How?

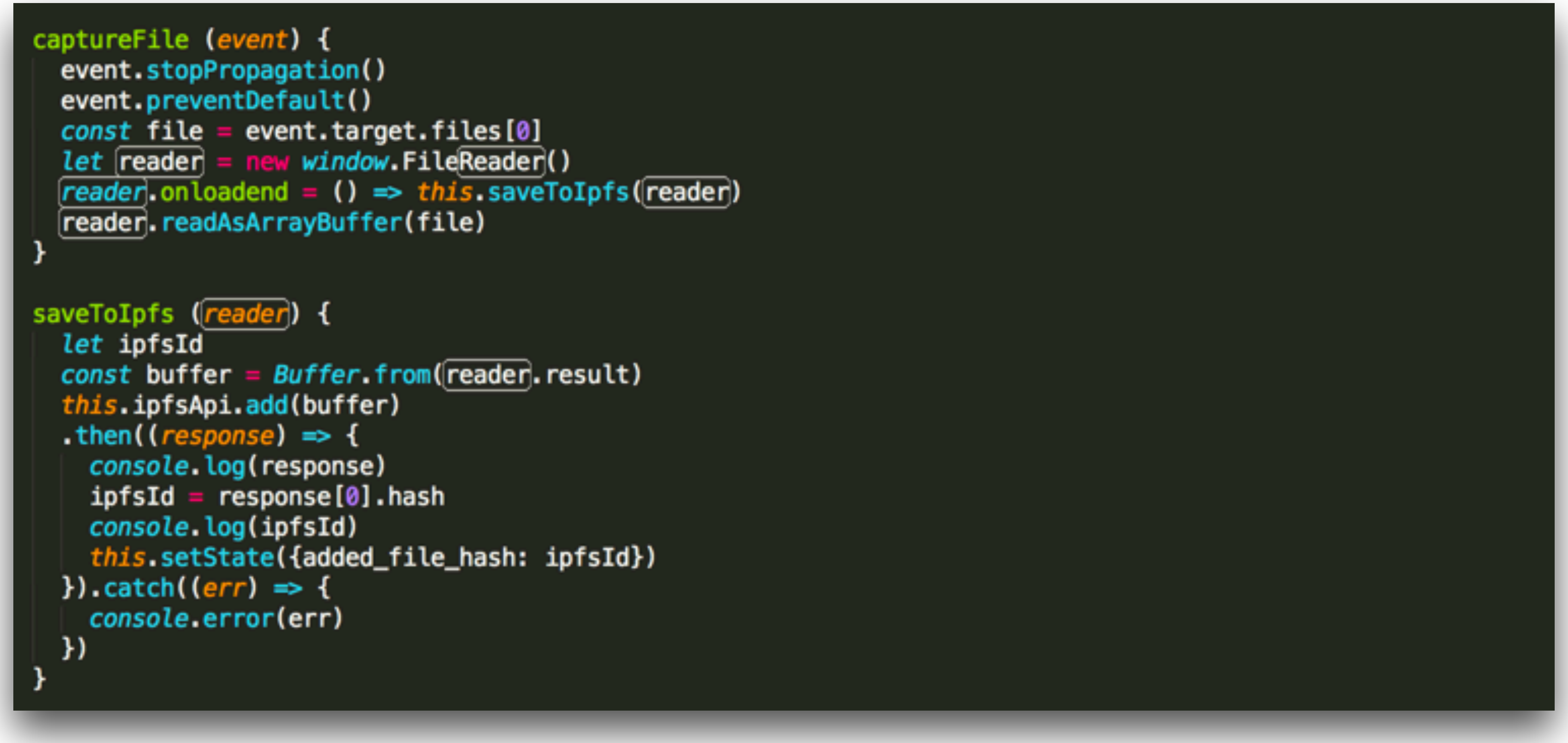
- How does it work? Incentives…FileCoin?
- https://github.com/ipfs/js-ipfs
- https://github.com/ipfs/js-ipfs-api

js-ipfs + js-ipfs-api

IPFS Examples

- > npm install ipfs --global (for the daemon)
- > *jsipfs init*
- > jsipfs daemon (ps aux l grep jsipfs)
- Example (git clone)
  - https://github.com/ipfs/js-ipfs-api/tree/master/examples/upload-file-via-browser
- > npm install
- > npm start

IPFS Examples



```
captureFile (event) {
  event.stopPropagation()
  event.preventDefault()
  const file = event.target.files[0]
  let reader = new window.FileReader()
  reader.onloadend = () => this.saveToIpfs(reader)
  reader.readAsArrayBuffer(file)
}

saveToIpfs (reader) {
  let ipfsId
  const buffer = Buffer.from(reader.result)
  this.ipfsApi.add(buffer)
  .then((response) => {
    console.log(response)
    ipfsId = response[0].hash
    console.log(ipfsId)
    this.setState({added_file_hash: ipfsId})
  }).catch((err) => {
    console.error(err)
  })
}
```

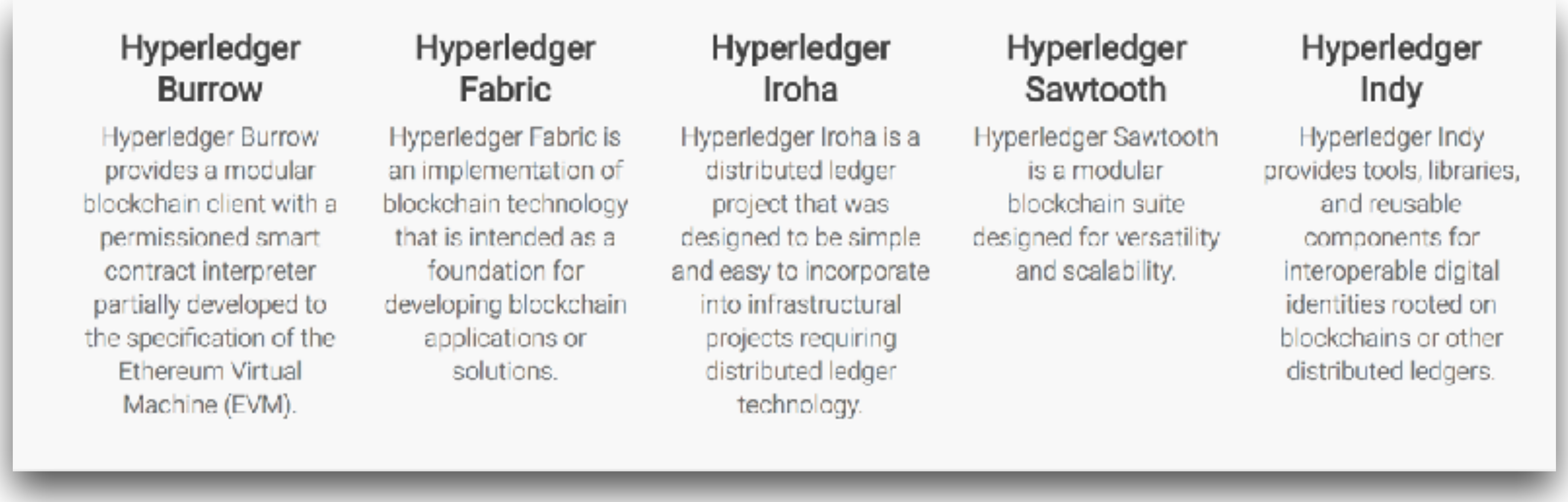
Hyperledger

Hyperledger Overview

- https://www.hyperledger.org
- "Hyperledger is an open source collaborative effort created to advance cross-industry blockchain technologies. It is a global collaboration, hosted by The Linux Foundation, including leaders in finance, banking, IoT, supply chain, manufacturing and technology."
- Creator(s)
- Who's backing?
- Coding against via Hyperledger Fabric

Hyperledger Fabric

◉ https://fabric-sdk-node.github.io



Hyperledger Burrow

Hyperledger Burrow provides a modular blockchain client with a permissioned smart contract interpreter partially developed to the specification of the Ethereum Virtual Machine (EVM).

Hyperledger Fabric

Hyperledger Fabric is an implementation of blockchain technology that is intended as a foundation for developing blockchain applications or solutions.

Hyperledger Iroha

Hyperledger Iroha is a distributed ledger project that was designed to be simple and easy to incorporate into infrastructural projects requiring distributed ledger technology.

Hyperledger Sawtooth

Hyperledger Sawtooth is a modular blockchain suite designed for versatility and scalability.

Hyperledger Indy

Hyperledger Indy provides tools, libraries, and reusable components for interoperable digital identities rooted on blockchains or other distributed ledgers.

Ethereum

Ethereum Overview

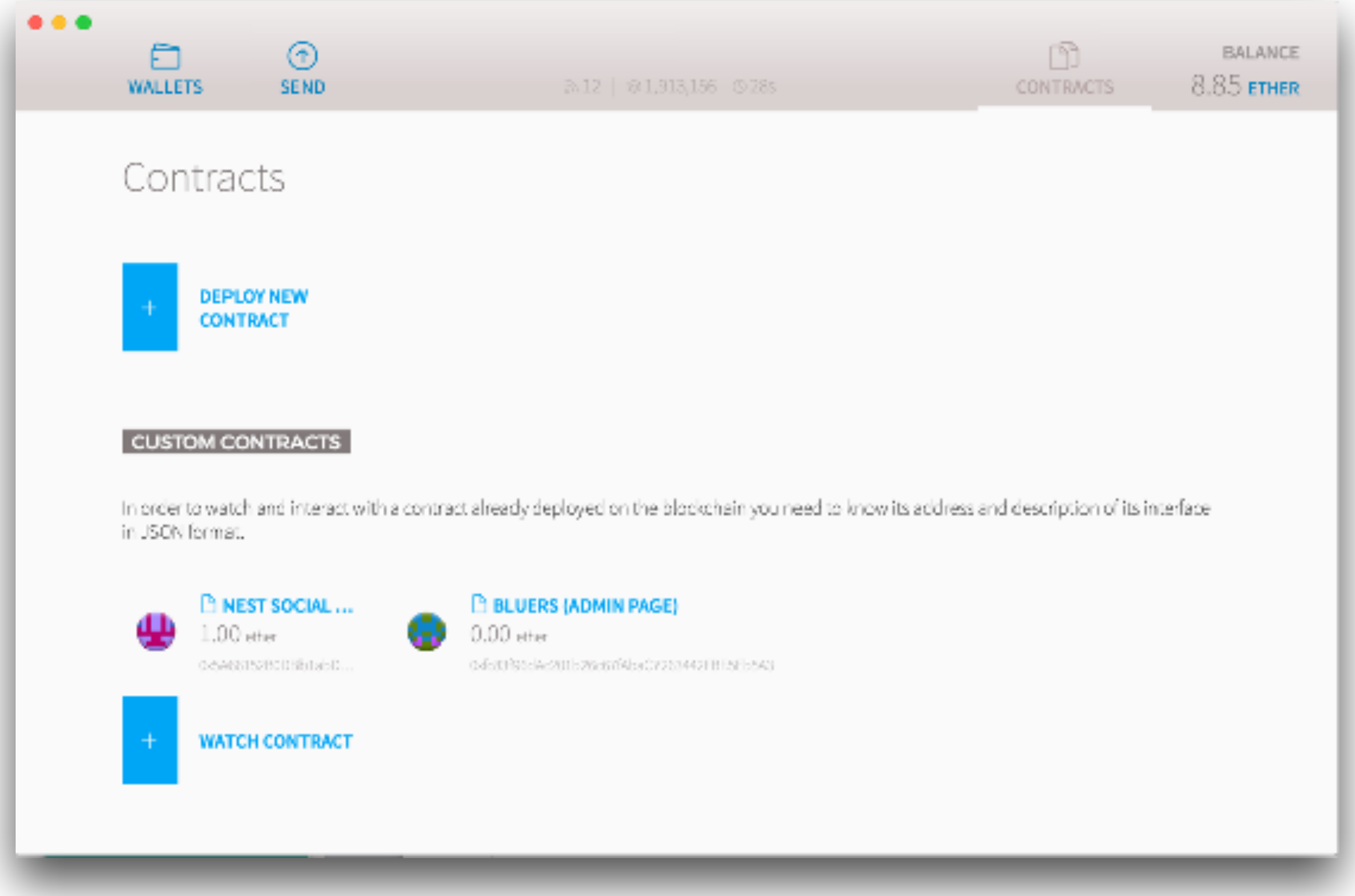
- https://ethereum.org/
- “Ethereum is a **decentralized platform** that runs **smart contracts**: applications that run exactly as programmed without any possibility of downtime, censorship, fraud or third party interference.”
- Market capitalization: $25.396bn

Ethereum Characteristics

- Wallets (address) AND Contracts (address)
- Programmable (via Solidity or equivalents)
- EVM + Supports Multiple States
- Applications manifest themselves as DApps
- Various environments when building…
  - Ethereum Main Networks
  - Various TestNets (Ropsten, Kovan, etc)
  - Local (via testrpc)

Ethereum Wallet

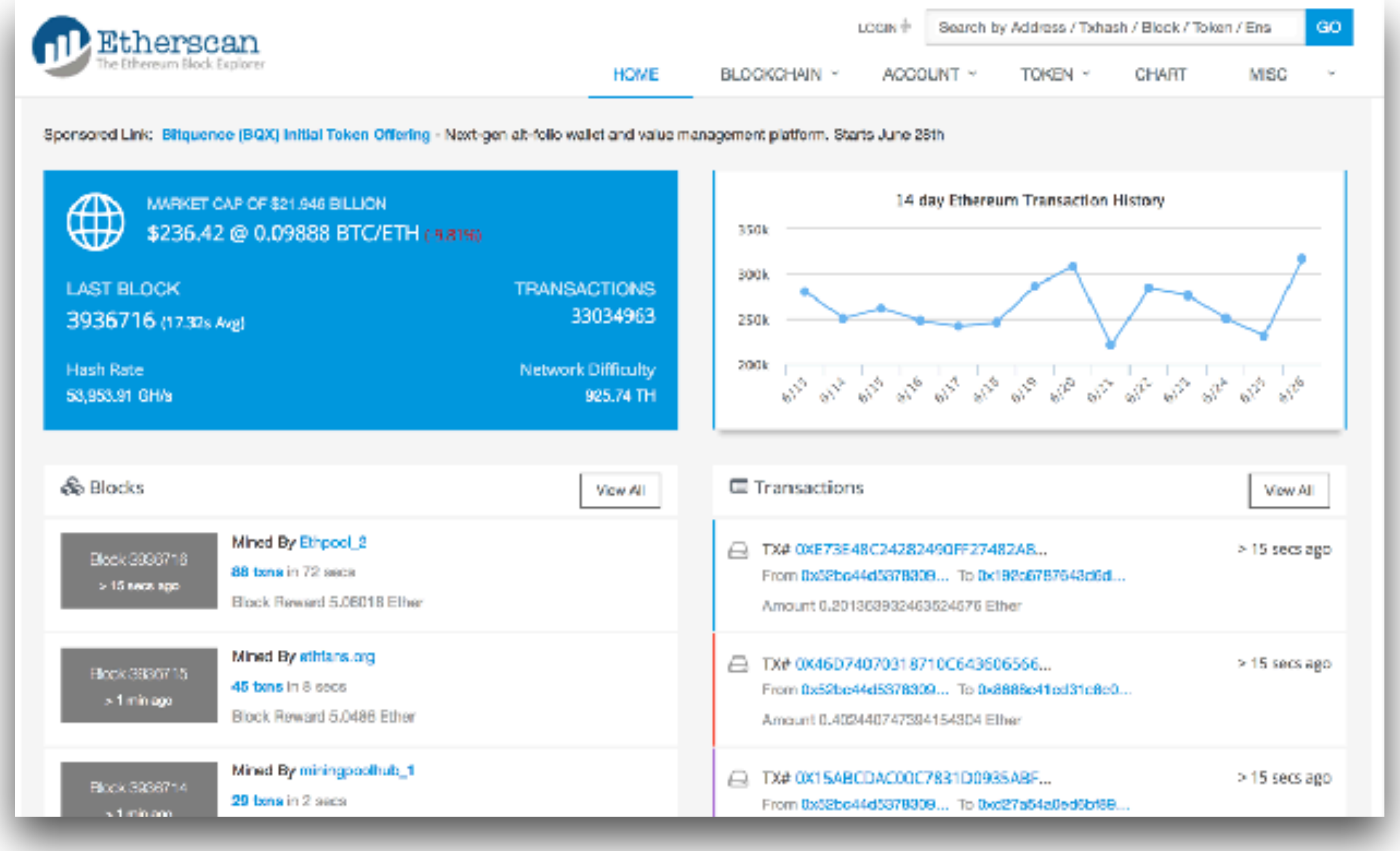
- ◉ Manage Accounts / Send ETH / Manage Contracts



In order to watch and interact with a contract already deployed on the blockchain you need to know its address and description of its interface in JSON format.

etherscan.io

- 0x830E3A6766C753e041aa5B78e94213972a99D400



Etherscan
The Ethereum Block Explorer
LOGIN
Search by Address / Txhash / Block / Token / Ens
GO
HOME
BLOCKCHAIN
ACCOUNT
TOKEN
CHART
MISC

Sponsored Link: Bitquence (BQX) Initial Token Offering - Next-gen alt-folio wallet and value management platform. Starts June 26th

MARKET CAP OF $21.946 BILLION
$236.42 @ 0.09888 BTC/ETH (-5.81%)

LAST BLOCK
3936716 (17.32s Avg)

TRANSACTIONS
33034963

Hash Rate
53,953.91 GH/s

Network Difficulty
925.74 TH

14 day Ethereum Transaction History

Blocks
View All

Block 3936716
> 15 secs ago
Mined By Ethpool_2
88 txns in 72 secs
Block Reward 5.08018 Ether

Block 3936715
> 1 min ago
Mined By ethfans.org
45 txns in 8 secs
Block Reward 5.0486 Ether

Block 3936714
> 1 min ago
Mined By miningpoolhub_1
29 txns in 2 secs

Transactions
View All

TX# 0XE73E48C24282490FF27482AB... > 15 secs ago
From 0x52bc44d5378309... To 0x192c6787543d6d...
Amount 0.201363932463524676 Ether

TX# 0X46D74070318710C643606566... > 15 secs ago
From 0x52bc44d5378309... To 0x8888e41cd31c8c0...
Amount 0.402440747384154304 Ether

TX# 0X15ABCDAC00C7831D0935ABF... > 15 secs ago
From 0x52bc44d5378309... To 0xd27a54a0ed6b18B...

Real World Applications

- Auctions
- DAOs
- ICOs
  - “Initial Coin Offering”
  - $0.5bn raised in the last X months
  - Examples
    - Bancor ($153m raised in 3 hours)
    - Status.im
    - Dcorp
    - Civic (crashed the Ethereum network)

Building with Ethereum

There’s a few new concepts…

- Geth (or equivalent)
- Gas (execution fee priced in Ether)
- Mist
- Solidity
- Truffle
- DApps
- **Web3.js**
- ABI (Application Binary Interface)

Geth

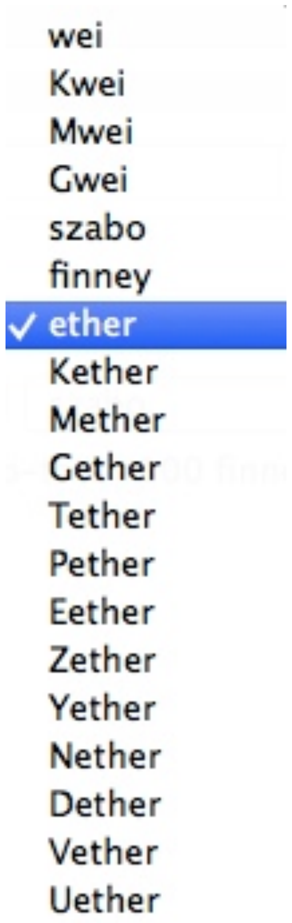
Geth

- CLI for running a full Ethereum node (written in Go)
- Enables the following…
  - Mine ETH
  - Transfer funds between addresses
  - Explore block history
  - Compile + deploy contracts
  - Execute contracts + transactions

Geth Demo (Console)

Geth Demo

- > geth attach http://localhost:8545
- Opens JavaScript console
- > eth.accounts
- > eth.getBalance(eth.accounts[1])
- > eth.sendTransaction({from: eth.accounts[0], to: eth.accounts[1], value: web3.toWei(1, "ether")})
- > eth.getBalance(eth.accounts[1])



Smart Contracts

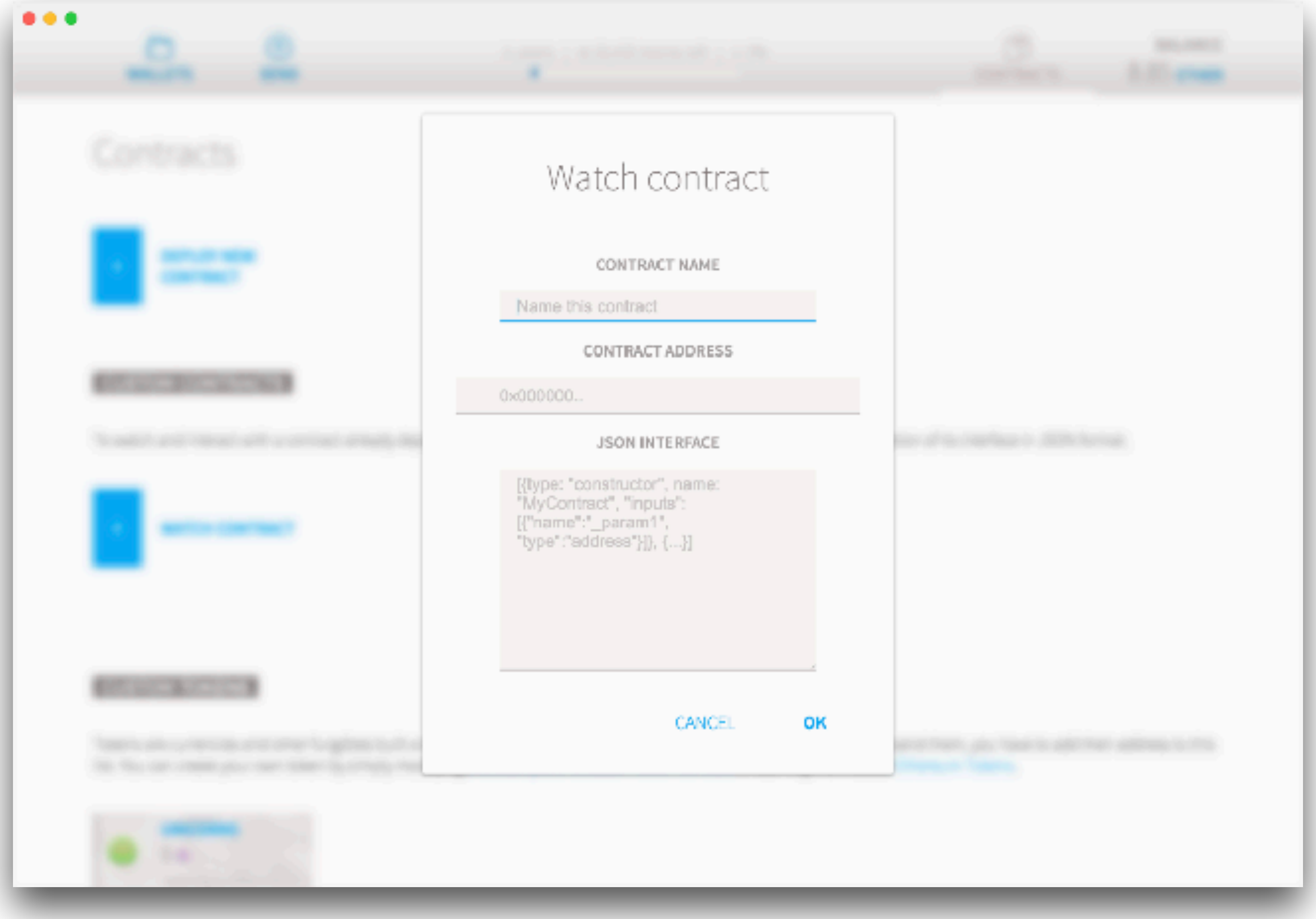
Smart Contracts

- **Where things get more interesting**
- Written a language solidity (or equivalent)
- Compiled down to Ethereum Bytecode (runs on EVM)
- Contracts have their own address when deployed
- Also have an ABI (Application Binary Interface)
  - Parameters, etc

Interacting with Smart Contracts

- Ethereum Wallet
- Via DApps (we’ll discuss shortly)
  - *MIST Browser*
  - *MetaMask*
  - *Status.im*

Interacting with Smart Contracts



Watch contract

CONTRACT NAME

Name this contract

CONTRACT ADDRESS

0x000000..

JSON INTERFACE

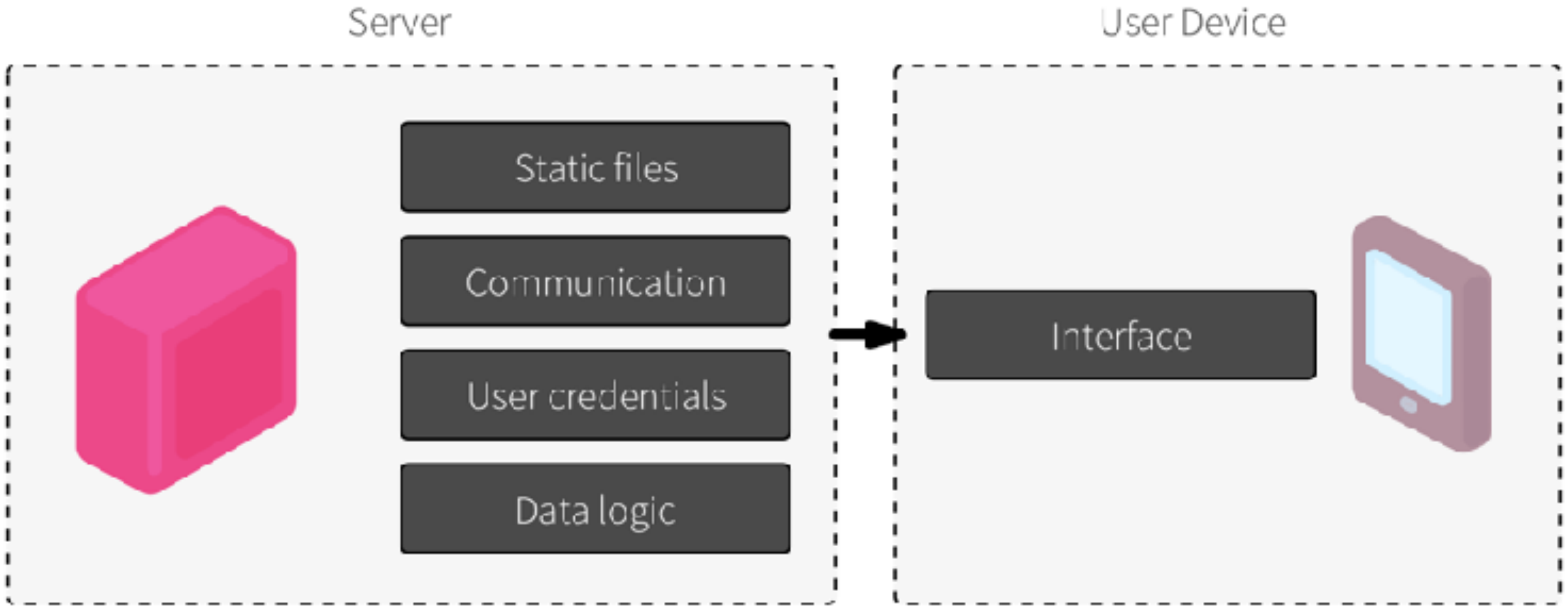
```
[{type: "constructor", name:
"MyContract", "inputs":
[{"name":"_param1",
"type":"address"}]}, {...}]
```

Distributed Apps (aka DApps)

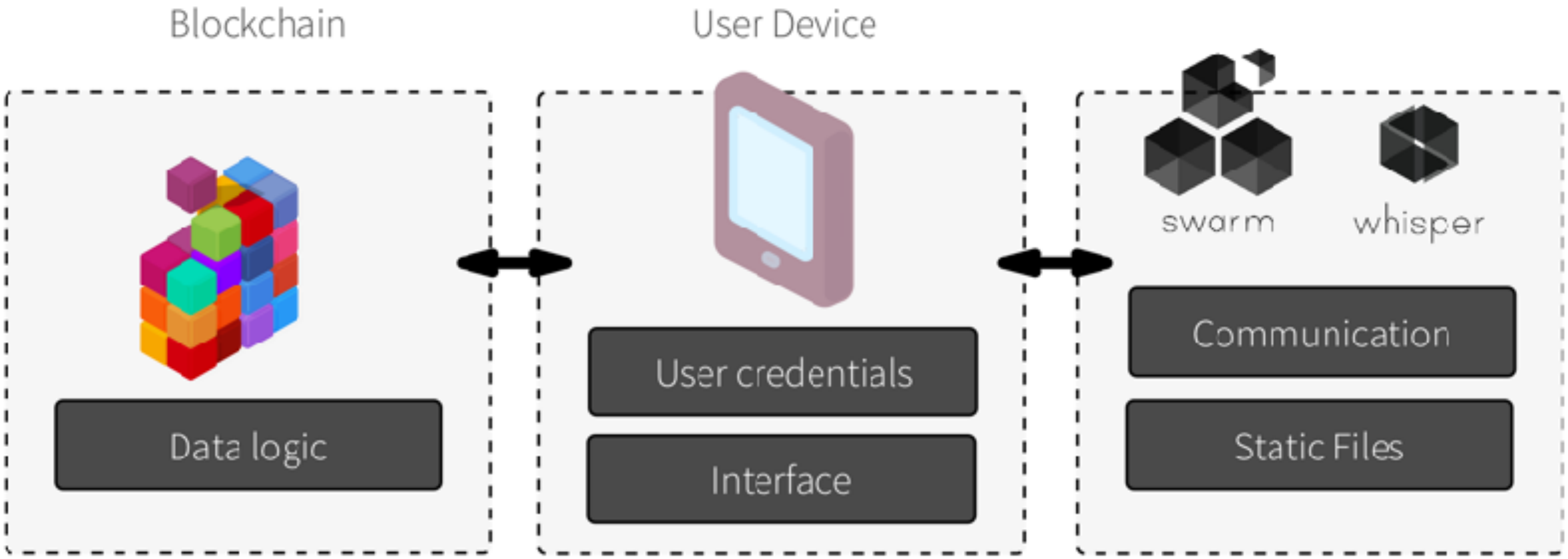
Distributed Apps (aka DApps)

- *Where things get* ***even*** *more interesting?*
- “**DApp** is an abbreviated form for decentralized application. A **DApp** has its backend code running on a decentralized peer-to-peer network. Contrast this with an app where the backend code is running on centralized servers.”

Client-Server vs…



Client-Various…



Building a Real-ish World DApp

Stuff…

- TestRPC
- Truffle Framework
- Solidity
- Web3
- *Angular2*

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.”

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

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 (EVM).”

Web3.js

- Ethereum JavaScript API
- “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.”

Demo:
Real-ish World DApp

Steps

- Steps
  - Run testrpc
  - > truffle compile
  - > truffle migrate
  - > ng serve
- Get the first wallet address from testrpc > eth.accounts
- Transfer some MetaCoins

- https://github.com/Nikhil22/angular2-truffle-starter-dapp

What just happened?

- Compiled the smart contract
- Deployed it to the local test network
- Call the contract via Web3.js

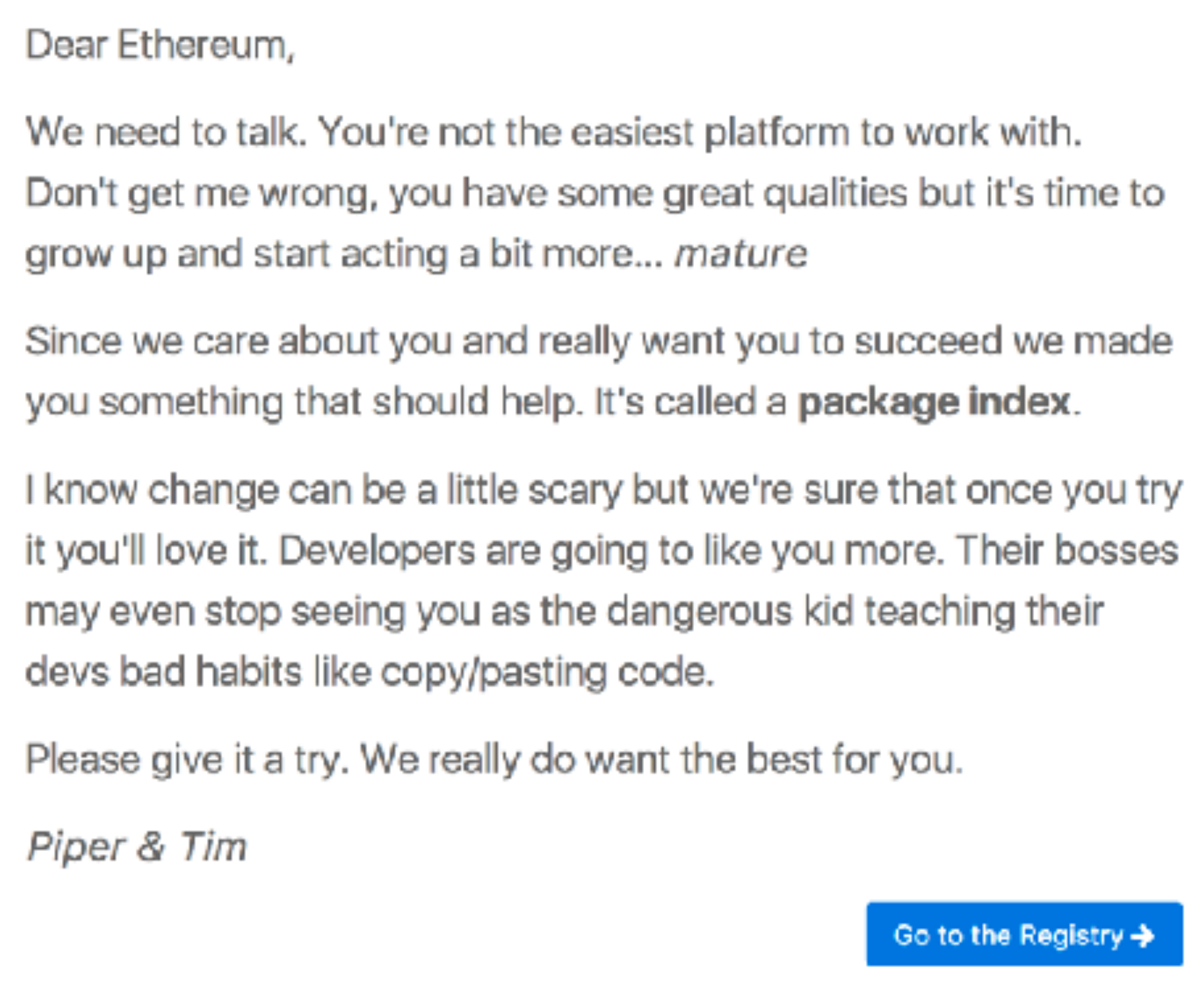
Check balance via Truffle CLI

```
> var account =
"0x0d4ade30817d0817f178695612c2a36dfd46244b";

> var meta;
MetaCoin.deployed().then(function(instance) { meta =
instance; return meta.getBalance.call(account, {from:
account}); }).then(function(balance)
{ console.log(balance.toNumber()); })
```

And last but not least…

Ethereum Package Registry

- https://www.ethpm.com



Dear Ethereum,

We need to talk. You're not the easiest platform to work with. Don't get me wrong, you have some great qualities but it's time to grow up and start acting a bit more... *mature*

Since we care about you and really want you to succeed we made you something that should help. It's called a **package index**.

I know change can be a little scary but we're sure that once you try it you'll love it. Developers are going to like you more. Their bosses may even stop seeing you as the dangerous kid teaching their devs bad habits like copy/pasting code.

Please give it a try. We really do want the best for you.

Piper & Tim

Q&A

Thank You 👍