Roadmap to "Blockchain"

Your IT Organization:

How to Help your IT Staff Go from Square One to Competence and Dominance in Blockchain Technologies

William Favre Slater, III
President & CEO
Slater Technologies, Inc.
August 24, 2018
http://billslater.com/blockchain
Blockchain Resources

https://goo.gl/baf6Uc
Bitcoin Resources

https://goo.gl/HhtCU7
Agenda

• Introduction and Where Are We Right Now?
• The Problem
• The Solutions
• Required Skills
• DApps and DApp Environment
• Case Studies
• The Challenges
• Solving the Challenges and Winning
• The Roadmap
• Some Valuable Resources
• Conclusion
• References
INTRODUCTION AND WHERE ARE WE RIGHT NOW?
Introduction

- William Favre Slater, III
  - Career Professional in Information Technology. Specialities: Cybersecurity, Data Centers, Blockchain, Infrastructure Management, Application Development, Project Management, Program Management, Service Management and Risk Management. This is me: http://billslater.com/interview

- I am also a former U.S. Air Force Officer and a Patriotic American Citizen who loves America, believes in the U.S. Constitution and the Bill of Rights.

- I am also a writer, and Adjunct Professor at the Illinois Institute of Technology for more than 10 years.

- I am happily married to my Soul Mate, who is my Best Friend and the Love of my Life, Ms. Joanna Roguska, who is a professional web developer and a native of Warsaw, Poland. We have been happily married since December 2000, and she became a U.S. Citizen in November 2006.
First Job Out of College

2LT William F. Slater, III
United States Air Force
Computer Systems Staff Officer

Strategic Air Command Headquarters
Offutt Air Force Base, NE
Circa late 1970s – UNCLASSIFIED Configuration
Microsoft Chicago Data Center

US-294 Northbound.
Two miles south of OHare
International Airport

Microsoft Chicago Data Center in Northlake, IL. Actual street view photo from Google Maps

Microsoft Chicago Data Center in Northlake, IL. Actual architect’s drawing from 2007 - 2008
## Microsoft Chicago Data Center

### CH1

<table>
<thead>
<tr>
<th></th>
<th>Colo Rooms</th>
<th>Cabinets</th>
<th>Servers per Cabinet</th>
<th>Total Production Servers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Second Floor</td>
<td>4</td>
<td>240</td>
<td>42</td>
<td>40,320</td>
</tr>
<tr>
<td>First Floor</td>
<td>1</td>
<td>56</td>
<td>2400</td>
<td>134,400</td>
</tr>
</tbody>
</table>

### CH2

<table>
<thead>
<tr>
<th></th>
<th>Colo Rooms</th>
<th>Cabinets</th>
<th>Servers per Cabinet</th>
<th>Total Production Servers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Second Floor</td>
<td>4</td>
<td>240</td>
<td>42</td>
<td>40,320</td>
</tr>
<tr>
<td>First Floor</td>
<td>1</td>
<td>48</td>
<td>2400</td>
<td>115,200</td>
</tr>
</tbody>
</table>

- **Size:** 705,000 square feet
- **Power:** 120 MW (enough to power 87,000 homes)
- **Critical Load for IT Equipment:** 60 MW
- **No. of Physical Servers:** > 330,000 Servers

---

Microsoft Chicago Data Center
Operations Team
Summer 2008

601 Northwest Hwy, Northlake, IL
WHERE ARE WE RIGHT NOW?
Bitcoin Prehistory

Bitcoin did not come out of the blue, it's not a fad
It's the result of 40 years of research, development and demand
A lot of talk, not a lot of walk.
Shortage.
Huge shortage.
Simple economics.
Still very early.
Ethereum / Initial release date

July 30, 2015

Ethereum was proposed in late 2013 by Vitalik Buterin, a cryptocurrency researcher and programmer. Development was funded by an online crowdsale that took place between July and August 2014. The system went live on **30 July 2015**, with 11.9 million coins "premined" for the crowdsale.
1801 Ethereum DApps currently

Source: https://www.stateofthedapps.com/

Copyright 2018 - DAperNetwork
Apple's App Store Is Growing by 1,000+ Apps a Day

Number of new apps submitted to Apple's App Store per month

Source: pocketgamer.biz
Weekly Bitcoin & Blockchain Statistics

The week of: August 13th - 20th, 2018

- **Miners Revenue in USD**: $84,820,309
  - **Change** 4.62%
- **BLX Average price this week**: $6,318.10 USD
  - **Change** 2.39%
- **Transactions on the Blockchain Ledger**: 1,540,381
- **New Bitcoin Created**: 13,274
  - ($83.86M USD new money supply)
  - **Change** 2.48%
- **Bitcoin Transacted across the Blockchain**: 774,389k
  - **Change** 7.49%
- **Average Blockchain Transaction in USD**: $3,176.27
  - **Change** 1.69%

Bitcoin is traded by 50+ regional markets (fiat pairs) across 200+ trading platforms.
The Problem

• A significant shortage of Blockchain Developer Talent
  – As of February 2018, it was reported that for every experienced Blockchain Engineer, there are 14 jobs available.

• The technical concepts and value are sound and the tools exist, but experienced people and project managers are very difficult to find.
The Solutions

- Train your staff with in-house training and/or mentors that experienced “experts”
- Send your staff to one or more Blockchain Developer Bootcamps (think DApperNetwork by Eric Chung)
- Provide self-training courses and hope for the best
Dappernetwork Ethereum DApp Bootcamp Attendees from Our Blockchain Team – at mHub, Chicago, Illinois, August 11, 2018

Kristen Counter  
William Slater  
Sara Shatdarsanam
The Required Skills for a Blockchain Development Staff

<table>
<thead>
<tr>
<th>Rank</th>
<th>Skill</th>
<th>Co-occurrences</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Blockchain</td>
<td>397 (100.00%)</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Finance</td>
<td>200 (50.38%)</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>JavaScript</td>
<td>184 (46.35%)</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Node.js</td>
<td>168 (42.32%)</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Ethereum</td>
<td>151 (38.04%)</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Bitcoin</td>
<td>146 (36.78%)</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>SQL</td>
<td>142 (35.77%)</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Cryptocurrency</td>
<td>139 (35.01%)</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Java</td>
<td>134 (33.75%)</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>NoSQL</td>
<td>125 (31.49%)</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Git (software)</td>
<td>123 (30.98%)</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>React</td>
<td>122 (30.73%)</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>Test Automation</td>
<td>118 (29.72%)</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>Front End Development</td>
<td>115 (28.97%)</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>Smart Contracts</td>
<td>111 (27.96%)</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>Solidity</td>
<td>107 (26.95%)</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>Linux</td>
<td>106 (26.70%)</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>AngularJS</td>
<td>104 (26.20%)</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>Docker</td>
<td>101 (25.44%)</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>Redis</td>
<td>98 (24.69%)</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>MySQL</td>
<td>93 (23.43%)</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>Banking</td>
<td>93 (23.43%)</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>Amazon AWS</td>
<td>92 (23.17%)</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>Telecoms</td>
<td>88 (22.17%)</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>Agile Software Development</td>
<td>84 (21.16%)</td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>ES6</td>
<td>84 (21.16%)</td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>CSS</td>
<td>77 (19.40%)</td>
<td></td>
</tr>
</tbody>
</table>
Additional Required Skills for a Blockchain Development Staff

- Web3.js
- DApp development
- UI and UX Design and Testing Skills
- Deep understanding of compiled code, Gas, and the Ethereum Virtual Machine (EVM)
- Secure coding
- Defensive coding
- Egoless Programming
- Stringent Code Reviews
- Networking
- Understanding of Protocols
- Planning
- Requirements
- Technical Specifications and Writing
- Design
- Architecture – Infrastructure, Data, and Security
- Testing – Testing – Testing
- Simulation
- Troubleshooting

And don’t forget
PROJECT MANAGEMENT!
Web3.js Tech Stack

MIST BROWSER

DECENTRALIZED APPLICATIONS

SWARM (STORAGE)  WHISPER (MESSAGING)  EVM (CONSENSUS)

HARDWARE CLIENTS

INTERNET

Figure 2.4: Web 3.0 tech stack for Ethereum, Source: Ethereum stack exchange
The Required Skills - Quick Preview

- Web3.js
- EVM
- Ethereum Blockchain
- Ethereum Ecosystem
- Gas
Ethereum Architecture

The Ethereum blockchain stack consists of various components. At the core, there is the Ethereum blockchain running on the peer-to-peer Ethereum network. Secondly, there's an Ethereum client (usually Geth) that runs on the nodes and connects to the peer-to-peer Ethereum network from where blockchain is downloaded and stored locally. It provides various functions, such as mining and account management. The local copy of the blockchain is synchronized regularly with the network. Another component is the web3.js library that allows interaction with the geth client via the Remote Procedure Call (RPC) interface.

This architecture can be visualized in the following diagram:

The Ethereum stack showing various components
EVM Operation and Architecture

Source: Mastering Blockchain by Imran Bashir (Published by Packt.)
Byte Code Executed by the EVM

**Runtime bytecode**

Raw hex codes:

```
606060405260e060020a6000350463989e17318114601c575b6000565b34600057602960043
5603b565b6040805198252519081900360200190f35b600281015b91905056
```

Opcodes:

```
PUSH1 0x60 PUSH1 0x40 MSTORE PUSH1 0x2 PUSH1 0x0 SSTORE CALLVALUE PUSH1 0x0
JUMPI JUMPDEST PUSH1 0x45 DUP1 PUSH1 0x1A PUSH1 0x0 CODECOPY PUSH1 0x0
RETURN PUSH1 0x60 PUSH1 0x40 MSTORE PUSH1 0xE0 PUSH1 0x2 EXP PUSH1 0x0
CALLDATACALC DIV PUSH4 0x989E1731 DUP2 EQ PUSH1 0x1C JUMPI JUMPDEST PUSH1
0x0 JUMP JUMPDEST CALLVALUE PUSH1 0x0 JUMPI PUSH1 0x29 PUSH1 0x4
CALLDATACALC PUSH1 0x3B JUMP JUMPDEST PUSH1 0x40 DUP1 MLOAD SWAP2 DUP3
MSTORE MLOAD SWAP1 DUP2 SWAP1 SUB PUSH1 0x20 ADD SWAP1 RETURN JUMPDEST
PUSH1 0x2 DUP2 ADD JUMPDEST SWAP2 SWAP1 POP JUMP
```
While wei is the most used unit, there are also others denominations, here's a brief overview & conversion table of them in Ether:

<table>
<thead>
<tr>
<th>unit</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>wei</td>
<td>1e-18</td>
</tr>
<tr>
<td>kwei - ada</td>
<td>1e-15</td>
</tr>
<tr>
<td>mwei - babbage</td>
<td>1e-12</td>
</tr>
<tr>
<td>gwei - shannon</td>
<td>1e-9</td>
</tr>
<tr>
<td>szabo</td>
<td>1e-6</td>
</tr>
<tr>
<td>finney</td>
<td>1e-3</td>
</tr>
<tr>
<td>ether</td>
<td>1</td>
</tr>
<tr>
<td>kether-grand-einstein</td>
<td>1000</td>
</tr>
<tr>
<td>mether</td>
<td>1000000</td>
</tr>
<tr>
<td>gether</td>
<td>1000000000</td>
</tr>
<tr>
<td>tether</td>
<td>100000000000</td>
</tr>
</tbody>
</table>

And in wei:

<table>
<thead>
<tr>
<th>unit</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>wei</td>
<td>1</td>
</tr>
<tr>
<td>kwei - ada</td>
<td>1000</td>
</tr>
<tr>
<td>mwei - babbage</td>
<td>1000000</td>
</tr>
<tr>
<td>gwei - shannon</td>
<td>1000000000</td>
</tr>
<tr>
<td>szabo</td>
<td>100000000000</td>
</tr>
<tr>
<td>finney</td>
<td>100000000000000</td>
</tr>
<tr>
<td>ether</td>
<td>1000000000000000</td>
</tr>
<tr>
<td>kether-grand-einstein</td>
<td>1000000000000000000</td>
</tr>
<tr>
<td>mether</td>
<td>1000000000000000000</td>
</tr>
<tr>
<td>gether</td>
<td>1000000000000000000</td>
</tr>
<tr>
<td>tether</td>
<td>1000000000000000000</td>
</tr>
</tbody>
</table>

Source: Ethereum, Tokens & Smart Contracts: Notes on getting started by Eugenio Noyola
Gas

Gas is required to be paid for every operation performed on the Ethereum blockchain. This is a mechanism that ensures that infinite loops cannot cause the whole blockchain to stall due to the Turing-complete nature of the EVM. A transaction fee is charged as some amount of Ether and is taken from the account balance of the transaction originator.

A fee is paid for transactions to be included by miners for mining. If this fee is too low, the transaction may never be picked up; the more the fee, the higher are the chances that the transaction will be picked up by the miners for inclusion in the block. Conversely, if the transaction that has an appropriate fee paid is included in the block by miners but has too many complex operations to perform, it can result in an out-of-gas exception if the gas cost is not enough. In this case, the transaction will fail but will still be made part of the block, and the transaction originator will not get any refund.

Transaction cost can be estimated using the following formula:

\[
\text{Total cost} = \text{gasUsed} \times \text{gasPrice}
\]

Here, \text{gasUsed} is the total gas that is supposed to be used by the transaction during the execution and \text{gasPrice} is specified by the transaction originator as an incentive to the miners to include the transaction in the next block. This is specified in Ether. Each EVM opcode has a fee assigned to it. It is an estimate because the gas used can be more or less than the value specified by the transaction originator originally. For example, if computation takes too long or the behavior of the smart contract changes in response to some other factors, then the transaction execution may perform more or fewer operations than intended initially and can result in consuming more or fewer gas. If the execution runs out of gas, everything is immediately rolled back; otherwise, if the execution is successful and there is some remaining gas, then it is returned to the transaction originator.

A website that keeps track of latest gas price and provides other valuable statistics and calculators is available at https://ethgasstation.info/

Note: Your Solidity Code compiles to Byte Code. Each Byte Code Instruction has a predetermined Amount of Gas Value required for execution. Run out of Gas and your program stops. Or worse, Write a bad program that:

1) loops endlessly
2) Needlessly allocates huge array structures
3) Thoughtlessly allocates storage
4) Or, instantiates Smart Contracts too often and things will get EXPENSIVE REALLY FAST
Ethereum, just like any other blockchain, can be visualized as a transaction-based state machine. This definition is mentioned in the Ethereum yellow paper written by Dr. Gavin Wood.

The core idea is that in Ethereum blockchain, a genesis state is transformed into a final state by executing transactions incrementally. The final transformation is then accepted as the absolute undisputed version of the state. In the following diagram, the Ethereum state transition function is shown, where a transaction execution has resulted in a state transition:
Ethereum Blocks and the Ethereum Blockchain

Blocks and blockchain
As discussed earlier in this chapter, blocks are the main building blocks of a blockchain. Ethereum blocks consist of various elements, which are described as follows:

- The block header
- The transactions list
- The list of headers ofommers or uncles

The transaction list is simply a list of all transactions included in the block. Also, the list of headers of uncles is also included in the block.

The most important and complex part of a block in Ethereum is the block header. Block header consists of various elements which are introduced here.
The following figure shows the detailed structure of the block and block header:

![Diagram of block structure with block header]

Source: Mastering Blockchain by Imran Bashir (Published by Packt.)
Ethereum Blocks and the Ethereum Blockchain

### The genesis block
The genesis block varies slightly from normal blocks due to the data it contains and the way it has been created. It contains 15 items that are described here.

From [https://etherscan.io/](https://etherscan.io/), the actual version is shown as follows:

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Timestamp</td>
<td>(Jul-30-2015 03:26:13 PM +UTC)</td>
</tr>
<tr>
<td>Transactions</td>
<td>8893 transactions and 0 contract internal transactions in this block</td>
</tr>
<tr>
<td>Hash</td>
<td>0x4e56740f876ae8c010b86a40d5f5b745a118d906a34e69ac8c0db1cb8fa3</td>
</tr>
<tr>
<td>Parent hash</td>
<td>0x0000000000000000000000000000000000000000000000000000000000000000</td>
</tr>
<tr>
<td>SHA3 uncles</td>
<td>0x1d6c4de8dec75d7aa85567b6cdd41ad312451b948a7413f0a142fd40d49347</td>
</tr>
<tr>
<td>Mined by</td>
<td>0x0000000000000000000000000000000000000000000000000000000000000000 IN 15 secs</td>
</tr>
<tr>
<td>Difficulty</td>
<td>17,179,869,184</td>
</tr>
<tr>
<td>Total difficulty</td>
<td>17,179,869,184</td>
</tr>
<tr>
<td>Size</td>
<td>540 bytes</td>
</tr>
<tr>
<td>Gas used</td>
<td>0</td>
</tr>
<tr>
<td>Nonce</td>
<td>0x0000000000000000000000000000000000000000000000000000000000000042</td>
</tr>
<tr>
<td>Block reward</td>
<td>5 Ether</td>
</tr>
<tr>
<td>Undes reward</td>
<td>0</td>
</tr>
<tr>
<td>Extra data</td>
<td></td>
</tr>
<tr>
<td>Gas limit</td>
<td>5,000</td>
</tr>
</tbody>
</table>

Source: Mastering Blockchain by Imran Bashir (Published by Packt.)
On January 3rd, 2009 Satoshi Nakamoto unleashed the genesis block into the wild. The genesis block contains the first 50 BTC block reward and cannot be spent by anyone. The bitcoin software was released on Sourceforge the web-based service for open source developers. The original Satoshi client was compiled using Microsoft’s visual studio and is written in the programming language C++. Satoshi began building the code in 2007 and was the only developer that made software changes to the bitcoin protocol up until mid-2010.

The first block also includes text from the UK Times newspaper saying “03/Jan/2009 Chancellor on brink of second bailout for banks,” a reference to bitcoin’s birth-date and possibly mocking fractional reserve banking. With Satoshi’s creation, the total number of bitcoins in circulation will never exceed 21 million BTC.

Source: https://news.bitcoin.com/bitcoins-quirky-genesis-block-turns-eight-years-old-today/
The genesis block has received 1,073 transactions since its inception, with random people adding over 16 BTC to this unspendable address. No one knows why the genesis block was created to be unspendable, as there is no explanation in the Satoshi source code version 0.1. Additionally, the genesis block was hard-coded, and nearly all altcoins derived from bitcoin have this initial unspendable block reward.

Source: https://news.bitcoin.com/bitcoins-quirky-genesis-block-turns-eight-years-old-today
DAPPS AND DAPP DEVELOPMENT
Web Apps and DApps

Web 2.0 (Server based architecture)
Google Play
Google Play Store
iTunes
Apple iTunes Store

Web 3.0 (Serverless architecture)
Mist Browser
Ethereum (Unleashed)

Backward Compatibility
Architectural Divide
Web Application

Presentation
HTML, CSS, Javascript

Interaction / Communications
HTTP, HTTPS

Logic
Web app code (e.g. MVC implementation)

Data
Database or local storage

Static Storage
Server or Cloud Storage

Decentralized Application (Dapp)

Presentation
HTML, CSS, Javascript

Interaction / Communications
Whisper

Logic
Contract and Dapp Javascript

Data
Blockchain

Static Storage
Swarm or IPFS

Source: Blockchain Applications: A Hands-on Approach by Arsheep Bahga and Vijay Madisetti
DApps

Decentralized architecture, Source: Ethereum stack exchange
``` solidity
contract Sample {
    // state variables
    uint256 data;
    address owner;

    // event definition
    event logData(uint256 dataToLog);

    // function modifier
    modifier onlyOwner()
    {
        if (msg.sender != owner) throw;
    }

    // constructor
    function Sample(uint256 initData, address initOwner) {
        data = initData;
        owner = initOwner;
    }

    // functions
    function getData() returns (uint256 returnedData) {
        return data;
    }

    function setData(uint256 newData) onlyOwner {
        logData(newData);
        data = newData;
    }
}
```
contract Sample
{
  //state variables
  uint256 data;
  address owner;

  //event definition
  event logData(uint256 dataToLog);

  //function modifier
  modifier onlyOwner() {
    if (msg.sender != owner) throw;
    
  }

  //constructor
  function Sample(uint256 initData, address initOwner){
    data = initData;
    owner = initOwner;
  }

  //functions
  function getData() returns (uint256 returnedData){
    return data;
  }

  function setData(uint256 newData) onlyOwner{
    logData(newData);
    data = newData;
  }
}
Steps to DApp Development

**Analysis**
Identify the entities involved, their roles and types of interactions between them (e.g. contract owner, users, devices)

**Design**
Model the entity attributes as state variables and interactions between them as functions. Also capture the dependencies and constraints.

**Implementation**
Implement the contracts (including state variables, functions, modifier and events) in a higher-level languages such as Solidity For Dapp, also implement the front-end (HTML and CSS) and backend (Javascript).

Source: Blockchain Applications: A Hands-on Approach by Arsheep Bahga and Vijay Madisetti
Analysis Stage

(Example Business Case: Crowdfunding Application)

Campaign Owner

- Creates campaign
- Checks Campaign status

Crowdfunding Campaign

- Fund campaign

Campaign backers

Source: Blockchain Applications: A Hands-on Approach by Arsheep Bahga and Vijay Madisetti
Design Stage
(Example Business Case: Crowdfunding Application)

State Variables

- address public owner;
- uint public backers;
- uint public deadline;
- string public campaignStatus;
- bool ended;
- uint public goal;
- uint public amountRaised;
- struct Backer { address addr; uint amount; }
- mapping (uint => Backer) backers;

Campaign Owner

- Creates campaign
- Checks Campaign status

Crowdfunding Campaign

- Fund campaign

Campaign backers

- Crowdfunding(uint deadline, uint goal)
- checkGoalReached ()
- fund()
Implementation Stage
(Example Business Case: Crowdfunding Application)

User (owner)
- Creates and owns
  - Externally Owned Account
  - Transaction: Create Contract
    - uint deadline
    - uint goal
  - Contract Crowdfunding
    - State Variables
      - struct Backer {address addr; uint amount;}
      - address public owner;
      - uint public backers;
      - uint public deadline;
      - string public campaignStatus;
      - bool ended;
      - uint public goal;
      - uint public amountRaised;
      - mapping (uint => Backer) backers;
    - Functions
      - Crowdfunding(uint_deadline, uint_goal)
      - checkGoalReached ()
      - fund()  
      - destroy()
    - Modifiers
      - onlyOwner()
    - Events
      - Deposit (address_from,uint_amount)
      - Refund (address_to,uint_amount)
  - Transaction: checkGoalReached ()
  - Check if crowdfunding goal is reached

Users (backers)
- EOAs send transactions
  - EOAs
  - Transaction: fund ()
    - value
  - Fund campaign

BLOCKCHAIN APPLICATION TEMPLATES
Blockchain Application Templates

Many-to-One

Contract owner

Creates and owns

Externally Owned Account (EOA)

- Account Address
- Account Keys
- Ether Balance

Calls and Transactions

Contract

- State Variables
- Functions
- Modifiers
- Events

Users

EOAs

Calls and Transactions

Some Current Examples

- Crowdfunding
- Event Registration
- Voting
- Name Registration

Source: Blockchain Applications: A Hands-on Approach by Arsheep Bahga and Vijay Madisetti
Blockchain Application Templates

Many-to-One for IoT Applications

Some Current Examples
- Solar charging stations
- Smart switch

Source: Blockchain Applications: A Hands-on Approach by Arsheep Bahga and Vijay Madisetti
Blockchain Application Templates

Many-to-One for Financial Applications

Contract owner

EOA

Calls and Transactions

EOAs

Calls and Transactions

Contract

State Variables

Functions

Modifiers

Events

Some Current Examples

- Product sales
- Stock photos
- Document verification

Roles

- Seller
- Producer
- Creator

- Buyer
- Consumer
- Verifier

Source: Blockchain Applications: A Hands-on Approach by Arsheep Bahga and Vijay Madisetti
Blockchain Application Templates

Many-to-Many or Peer-to-Peer

Some Current Examples
- Call option
- Interest rate swap

Source: Blockchain Applications: A Hands-on Approach by Arsheep Bahga and Vijay Madisetti
DEVELOPMENT TOOLS
Development Tools

- **Blockchain**
  - Geth
  - Parity
  - Ganache

- **Frameworks**
  - Truffle
  - RiskBlock

- **Editors**
  - Atom
  - Sublime
  - Notepad
  - Eclipse
  - Your favorite text editor

- **Languages**
  - Solidity
  - Javascript
  - LLL
  - Viper
  - Mulu

- **Testing**
  - [https://test.eth.guide](https://test.eth.guide)

- **Publicly listed Contracts**
  - [http://etherchain.org/contracts](http://etherchain.org/contracts)
Preparation

• Installation

You'll need Git, Node Package Manager, MetaMask and Truffle set up.

Click these links and follow directions:
- Git: https://git-scm.com/downloads
- MetaMask: https://chrome.google.com/webstore/detail/metamask/nkbihfbeogaeaoehlefnkodbefgpgknn?hl=en
- Truffle: http://truffleframework.com/docs/getting_started/installation
- If you have Windows: http://truffleframework.com/tutorials/how-to-install-truffle-and-testrpc-on-windows-for-blockchain-development

August 24, 2018
Roadmap to "Blockchain" Your IT Organization: How to Help your IT Staff Ramp-Up - William Favre Slater, III
Preparation

• Reading Material

Reading Material
Ethereum Whitepaper: https://github.com/ethereum/wiki/wiki/White-Paper
Prehistory of Ethereum protocol by Vitalik: https://vitalik.ca/general/2017/09/14/prehistory.html
Just Enough Bitcoin for Ethereum: https://media.consensys.net/time-sure-does-fly-ed4518792679
Byzantine Fault Tolerance primer: https://www.nasdaq.com/article/byzantine-fault-tolerance-the-key-for-blockchains-cm810058
Mechanism Design & Cryptoeconomics primer: https://medium.com/blockchannel/a-crash-course-in-mechanism-design-for-cryptoeconomic-applications-a9f06ab6a976
Solidity Preparation

- Create an Ethereum UserID and Wallet
- Install Blockchain software (geth,parity, or ganache)
  - Geth  https://geth.ethereum.org/downloads/
  - Parity http://parity.io
  - Ganache https://github.com/trufflesuite/ganache
- Install Node.js and NPM (NPM is installed with node.js)  https://nodejs.org/en
- Install Web3.js (gets automatically installed with parity) otherwise:  https://github.com/ethereum/web3.js/
- Install Mist blockchain explorer  https://github.com/ethereum/mist
- Install Metamask https://metamask.io/
- Remix Solidity IDE Editor & Compiler
  - http://remix.ethereum.org/#optimize=true&version=soljson-v0.4.24
- Kovan (Rinkeby) Blockchain
  - https://www.rinkeby.io/#stats
- Install the Solidity Compiler  https://github.com/ethereum/solidity
- Install an editor (preferably Atom, but Eclipse or sublime or another is OK)
CASE STUDIES
Case Study 1

• Timeframe: Summer of 2018
• Location: Chicago
• Topic: Teaching Interns who are Technical People with Graduate degrees free Blockchain classes
• 33 started, 8 remain
• First Project: We are converting and existing Time Tracking GUI Applictaion to an Ethereum DApp
• Second Project: Designing and Implementing a DApp Solution from Scratch
• We are working together from June 1 – December 31
Case Study 2

- Timeframe: November 2017
- Location: Chicago
- Topic: Placement in Production of flawed Smart Contract
- Results: Loss of over $150 million
$150,000,000 bug
The Challenges

• Huge Learning Curve
• DApps with Web3 and the EVM are not your Father’s Web Developer Workbench
• You can really screw this up – easily
• Learning Egoless Programming
• Turnover – Once people get training and experience the may leave
Solving the Challenges and Winning

• Find and utilize quality resources to accelerate your learning curve and immersion into the Blockchain World

• Establish a Blockchain Expert or Champion imbued with the responsibility to be the Blockchain Evangelist

• Build strong Learning Teams – Use Peter Senge’s Learning Team Disciplines
  – Shared Vision
  – Personal Mastery
  – Mental Modeling
  – Team Learning
  – Systems Thinking

• Stay abreast of Blockchain Technologies and Blockchain Politics and Blockchain Evolution

• Join and participate in Local Blockchain Meetups

• Go International - Get involved with the Internet Society and the Blockchain Special Interest Group - Both are free and the Blockchain SIG has great projects and leadership
  – www.internetsociety.com
  – https://www.isoc-bsig.org/
  – https://www.linkedin.com/company/isoc-blockchain-sig/
Roadmap to "Blockchain" Your IT Organization: How to Help Your IT Staff Go from Square One to Competence & Dominance in Blockchain Technologies

**Orientation**
- Start
  - Learn the Terminology and Concepts.
  - Perform a Baseline Skills Inventory and Assessment
  - Blockchain Introduction and Orientation
  - Review Real-World Use Cases and Applications
  - Read Papers & Join Blockchain Meetup Groups, and other Blockchain-related Organizations like www.isoc-bsig.org

**Preparation**
- Analyze your initial Blockchain Needs and Requirements
  - Baseline your Capabilities
  - Perform Gap Analysis of Needs versus Capabilities
  - Remediate Skills Gap with Consultants, Training, and/or Mentoring
  - Perform Detailed Requirements Analysis
  - Create a Blockchain Solution Design Based on the Detailed Analysis

**Crawl**
- Select the Type of Blockchain
  - Prepare and Validate the Blockchain DApp Development Environment
  - Implement a Prototype a Proof of Concept DApp solution
  - Validate the DApp Prototype
  - Add Additional Features to the DApp Prototype
  - Test, Validate, and Publish Results

**Walk**
- Perform Detailed Requirements Analysis
  - Identify the Appropriate Blockchain Solutions Template
  - Create a Blockchain Solution Design Based on the Detailed Analysis and the Appropriate DApp Template
  - Create an Implementation Diagram for the Blockchain DApp Solution based on the Design
  - Implement the Blockchain DApp Solution based on the Implementation Diagram
  - Test and Optimize the DApp for Optimal Performance, and Validate against Requirements

**Run**
- Review Lessons Learned from Previous Tracks or DApps
  - Focus on Implementing Techniques to Optimize the Analysis, Design, Testing and Implementation
  - Incorporate the use of Agile/Scrum and DevOps in the Blockchain Solution Development Lifecycle
  - Perform Analysis, Design Testing and Implementation based on Previous Experience and Lessons Learned
  - Test and Optimize the DApp for Optimal Performance, and Validate against Requirements & Publish Results
  - Continue Continuously
The High-Level Project Plan

Create and Deliver a 40-Hour Hands-On Course on Blockchain and Blockchain Technologies

<table>
<thead>
<tr>
<th>Initiating Planning</th>
<th>Executing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Phase 0</td>
<td></td>
</tr>
</tbody>
</table>

Legend

- Phase 0: Project Initiation, Planning, and Management
- Phase 1: Analysis – Determine all topical areas to be covered
- Phase 2: Design – Modules, Lectures, Exercises
- Phase 3: Implementation – Create the Modules, Lectures, Exercises for the One Third Content Delivery on January 31, 2019
- Phase 4: Implementation – Create the Modules, Lectures, Exercises for the Remaining Two Thirds Content Delivery on September 1, 2019
- Phase 5: Pilot Delivery

2018

SOME VALUABLE RESOURCES
Sample Code Resources

Mastering Blockchain, Second Edition
by Imran Bashir

Ethereum Smart Contract Development
by Mayukh Mukhopadhyay

Solidity Programming Essentials: A beginner's guide to build smart contracts for Ethereum and blockchain
by Ritesh Modi
Sample Code Resources

Building Blockchain Projects: Building Decentralized Blockchain Applications with Ethereum and Solidity
By Narayan Prusty

Blockchain Applications: A Hands-On Approach
by Arshdeep Bahga and Vijay Madisetti

Ethereum, Tokens & Smart Contracts: Notes on getting started
by Eugenio Noyola
Conclusion

• We covered:
  – Introduction and Where Are We Right Now?
  – The Problem
  – The DApp Environment
  – The Solutions
  – Case Studies
  – The Challenges
  – Solving the Challenges and Winning
  – The Roadmap
  – Some Valuable Resources
Special Thanks To
Chicago’s Best Blockchain Buddies:

Joe Hernandez
Co-Founder of the
Chicago Blockchain Project

Hannah Rosenberg
Co-Founder of the
Chicago Bitcoin and Open
Blockchain Meetup
Special Thanks To:

Vitalik Buterin
Inventor of Ethereum
PARTING THOUGHTS
"Our deepest fear is not that we think we are inadequate. Our deepest fear is that we are powerful beyond measure. It is our light, not our darkness, that most frightens us. We ask ourselves, who am I to be brilliant, gorgeous, talented and fabulous? Actually who are you not to be? You are a child of God. Your playing small doesn't serve the world. There is nothing enlightened about shrinking so that other people won't feel insecure around you."

Nelson Mandela
South African Leader
Author – Long Walk to Freedom
Nobel Peace Prize Laureate
nmandela@anc.org.za

1998 - Nelson Mandela shows Bill Clinton his jail cell where he was imprisoned for 18 years because of his stance on Apartheid issues
ATTENTION

There are three types of people in the World:

People who
Make Things
Happen

People who
Watch Things
Happen

People who say,
“What Happened?”
Think Differently.

Andreas Vollenweider
Milwaukee, Wisconsin
October 1998
Photo by John Owrey
Failure is not fatal, but failure to change might be.

John Wooden
If you think you understand quantum mechanics, you don't understand quantum mechanics.

--- Richard P. Feynman ---

If you think you understand Blockchain, you don't understand Blockchain.

--- William Favre Slater, III ---
If you think education is expensive, consider the cost of ignorance.
REFERENCES
References:

Best Blockchain Texts

- **Blockchain Basics: A Non-technical Introduction in 25 Steps**
  - by Daniel Drescher

- **Building Blockchain Projects: Building Decentralized Blockchain Applications with Ethereum and Solidity**
  - By Narayan Prusty

- **Ethereum, Tokens & Smart Contracts: Notes on getting started**
  - by Eugenio Noyola

- **Blockchain Applications: A Hands-On Approach**
  - by Arshdeep Bahga and Vijay Madisetti

- **Ethereum Smart Contract Development**
  - by Mayukh Mukhopadhyay

- **Solidity Programming Essentials: A beginner's guide to build smart contracts for Ethereum and blockchain**
  - by Ritesh Modi

- **Introducing Ethereum and Solidity: Foundations of Cryptocurrency and Blockchain Programming for Beginners**
  - By Chris Dannen

- **Mastering Blockchain - Second Edition**
  - by Imran Bashir
References: Best Blockchain Texts

- Blockchain Enabled Applications: Understand the Blockchain Ecosystem and How to Make it Work for You
  - by Vikram Dhillon, David Metcalf, Max Hooper

- Distributed Ledger Technology: The Science of the Blockchain
  - by Roger Wattenhofer

- The Book of Satoshi: The Collected Writings od Bitcoin Creator Satoshi Nakamoto
  - By Phil Champagne
References


References

References

References


Dedication

• Dedicated with never-ending love, respect, and gratitude to my dear Father-in-law and Mother-in-Law, Wiesiek Roguski (http://billslater.com/wiesiek) and Wiesia Roguska (http://billslater.com/wiesia).

August 24, 2018

Roadmap to "Blockchain" Your IT Organization: How to Help your IT Staff Ramp-Up - William Favre Slater, III
Presenter Bio:
William Favre Slater, III

- Lives in Chicago; Cybersecurity professional by day, Professor at night
- Married to my Best Friend and Soul Mate, Ms. Joanna Roguska
- Current Position – Project Manager / Sr. IT Consultant at Slater Technologies, Inc. Working on projects related to
  - Security reviews and auditing
  - Blockchain consulting
  - ISO 27001 Project Implementations
  - Subject Matter Expert for preparing Risk Management and Security Exams at Western Governor’s State University in UT
  - Providing subject matter expert services to Data Center product vendors and other local businesses.
  - Designing and creating a database application that streamlines program management, security management, risk management and reporting activities, for management of teams of IT workers and developers in teleworking environments. It will first be a Windows application and then be ported to the web.
  - Developing and presenting technical training materials for undergraduate and graduate students at the Illinois Institute of Technology in the areas of Blockchain and Blockchain development, Data Center Operations, Data Center Architecture, Cybersecurity Management, and Information Technology hardware and software.
  - Created an eBook with articles about Security, Risk Management, Cyberwarfare, Project Management and Data Center Operations
  - Professor at Illinois Tech for 10 years