Ethereum Overview

Agenda

11:30-11:45 Check-In, networking

11:45-12:45 Ethereum Overview

12:45 Announcements, Networking

Active Planning Committee

John Lindsay, Patent Attorney

Tony Schuman, Investment Advisor

Todd Russell, Gov't Contract Opportunities

See me to become more active in this or other chapters
To support meetings like this, www.ieee.org/join

Ethereum Overview

Ethereum

Blockchain-based distributed computing platform and operating system featuring “smart contract” (scripting language) functionality.

“A virtual machine for programs on a blockchain”

John Lindsay

Blockchain Series

- Loose Plan
  - ~~1st Presentation - Blockchain Basics~~
    - ~~Mile wide/inch deep~~
    - ~~At least not wholly conflate Bitcoin and Blockchain~~
  - 2nd Presentation - Ethereum overview
    - Overview of Ethereum, “Smart Contracts,”
  - 3rd Presentation - Code along
    - Bring laptop and follow along

Ethereum Overview

- Cryptocurrencies
  - 2008: Bitcoin whitepaper from “Satoshi Nakamoto”
  - 2009: Bitcoin client released
  - 2011: Litecoin (first “altcoin”)
  - 2014: Ethereum whitepaper released, crowdsale
- Ether
  - Cryptocurrency generated & used in the Ethereum platform
  - ~$525 : 1 ether (4/18/18)
  - Units
    - Ether 1.0
    - 1 Wei = 10^{-18} Ether

Ethereum Overview

- Bitcoin (decentralized currency)
    - Designed to transfer cryptocurrency between users
- Ethereum (decentralized currency with associated complex scripting language)
    - Executes “Smart Contracts”
        - Programs executed in the Ethereum Virtual Machine
    - Classic contract: “I promise to send you $100 if my presentation is rated 5 stars”
    - Smart contract: “I send cryptocurrency into a program executed on the blockchain which sends cryptocurrency to you for the 5 star rating, otherwise return to me ” ***

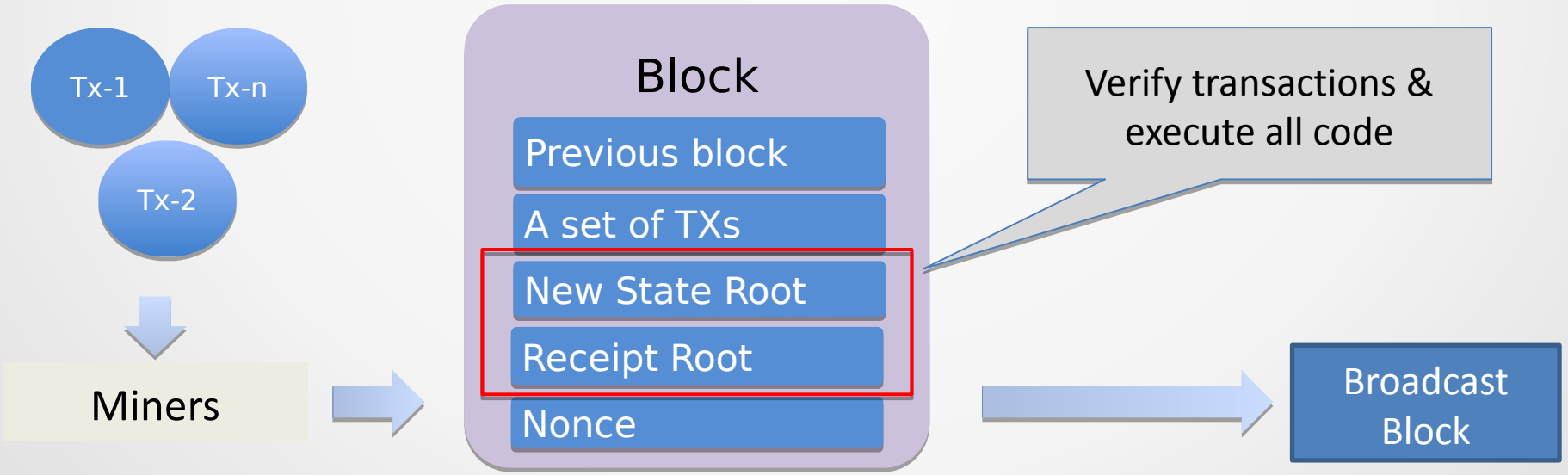
Ethereum Overview

- Ethereum – Use: “Smart Contracts”
  - White papers – “A Next-Generation Smart Contract and Decentralized Application Platform”
    - “Yellow Paper” provides further detail
  - Platform vs application
    - Smart Contracts, Dapps, tokens, ICOs, ... platform
  - Data
    - Ether transactions, smart contracts code
  - Actors
    - Ether holders, parties/actors in contracts
  - Interface
    - Miners, “Wallets”, JSON-RPC/Solidity

Ethereum Overview

Block Mining

Nodes on the blockchain process every transaction and stores the state
Every 15 seconds

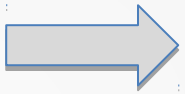


Ethereum Overview

Smart Contract Code

```
contract Greetings {
    string greeting;
    function Greetings (string _greeting) public {
        greeting = _greeting;
    }

    /* main function */
    function greet() constant returns (string) {
        return greeting;
    }
}
```



Bytecode seen on the blockchain

```
6060604052604051610250380380610250833981016040528.....
```

```
PUSH 60
    PUSH 40
    MSTORE
    PUSH 0
    CALLDATALOAD
.....
```

What people get from a disassembler

Ethereum Overview

- Programs executed in the Ethereum Virtual Machine
  - Transactions are more than just values, also programs that run when blocks are processed by nodes.
- if HAS_EVENT_X_HAPPENED()
  send(party_A, 1000)
  else:
  send(party_B, 1000)

Ethereum Overview - Applications

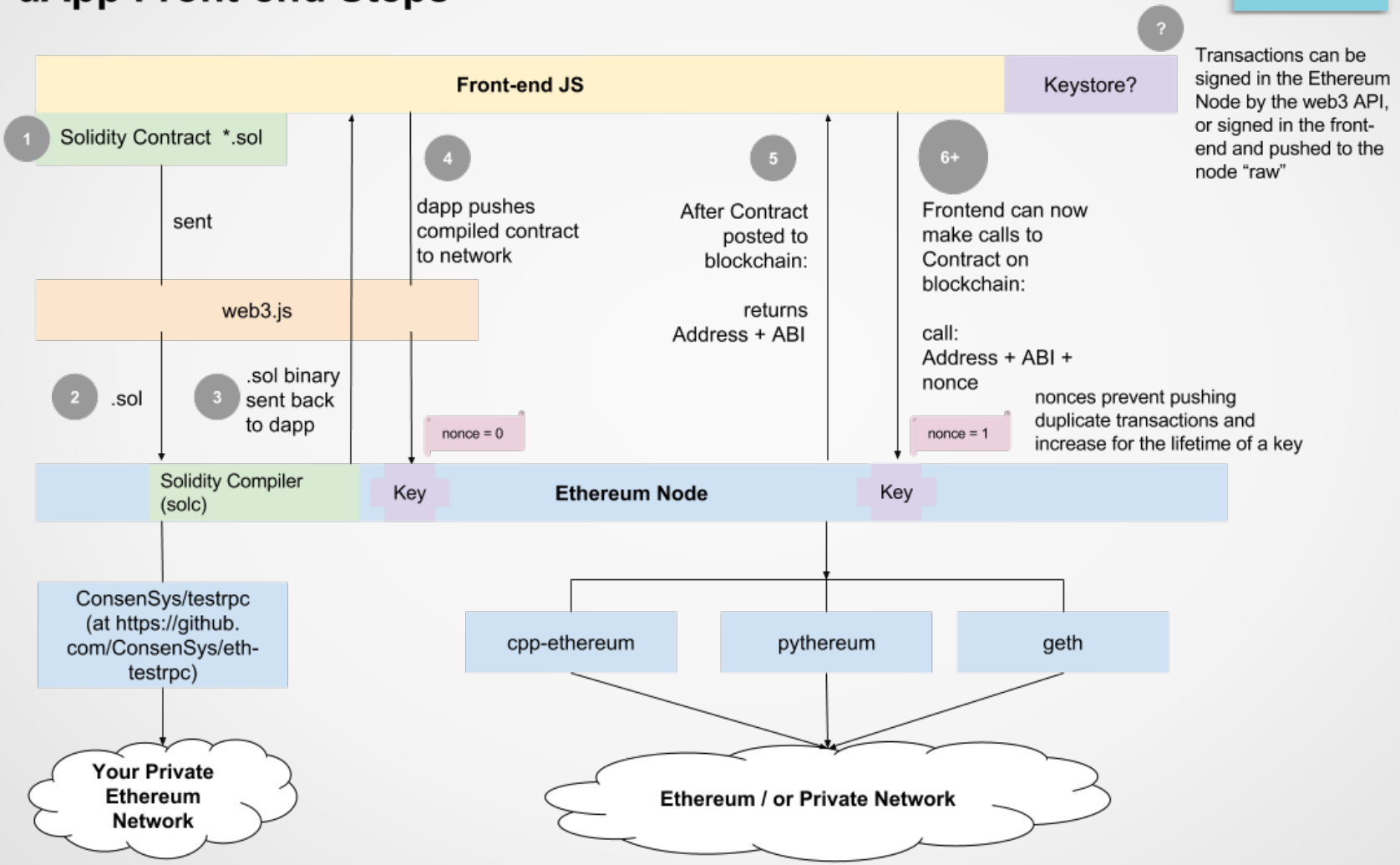
- Sample Applications
  - Multi-signature (Require M of N “owners” to agree in order to transfer)
  - Prediction markets (contributors predict events)
  - Crowdfunding
  - Etherian World (Minecraft-ish)
  - EtherTweet
  - Tokens/Initial Coin Offerings
  - Smart Locker
- Today’s applications
  - Cryptokitties
  - Selling/buying IoT sensor data
  - Toll tag logging and payment

Ethereum Overview – Language/API

- General Application Architecture
- “Smart Contract” (Ethereum Clients)
  - Solidity (Javascript-ish)
  - LLL (Lisp Like Language)
  - Serpent (Python-ish)
  - Mutan (C-ish) (deprecated)
- User/device facing (Connecting to Ethereum Clients)
  - Web3 (Javascript)
  - Web3j (Java/Android)
  - Nethereum (.NET)
  - Ethereum.rb (Ruby Gem)

Ethereum Overview

dApp Front-end Steps



A **Contract Creation Transaction** is shown in steps 1-5 at above.

An **Ether Transfer** or **Function Call Transaction** is assumed in step 6.

Ethereum Overview - Solidity

- Solidity Language
  - Variable Types
    - Common
      - bool, int, uint, uint[]
    - Unique
      - Address sender = 0x6414cc08d148dce9e...;
      - Struct Trade
        { uint quantity;
        uint price;
        string trader;}
      - mapping (address => uint) offers;

Ethereum Overview - Solidity

- Solidity Language (Inside → Outside)
  - Events
    - Events in solidity can be used to log certain events in EVM logs. These are basis for external interfaces (“Flag for external programs/devices”)
    - contract valueChecker {
      uint8 price=10;
      event valueEvent(bool returnValue); *********
      function Matcher(uint8 x) returns (bool) {
      if (x>=price) {
      valueEvent(true);
      return true;
      }
      }
      }

Program Execution Cost

- Solidity Language (Outside → Inside)
  - No simple, native internet retrieval
    - All nodes need to be able to independently validate output state of contract execution
    - oraclize.it

```
import "dev.oraclize.it/api.sol";

contract KrakenPriceTicker is usingOraclize {
    string public ETHXBT;

    function PriceTicker() {
        oraclize_setNetwork(networkID_testnet);
        oraclize_setProof(proofType_TLSNotary | proofStorage_IPFS);
        oraclize_query("URL", "json(https://api.kraken.com/0/public/Ticker?pair=ETHXBT).result
    }

    function __callback(bytes32 myid, string result, bytes proof) {
        if (msg.sender != oraclize_cbAddress()) throw;
        ETHXBT = result;
        // do something with ETHXBT
    }
}
```

Ethereum Overview - Solidity

- Solidity Language

browser/ballot.sol

```
1  pragma solidity ^0.4.0;
2  contract Ballot {
3
4      struct Voter {
5          uint weight;
6          bool voted;
7          uint8 vote;
8          address delegate;
9      }
10     struct Proposal {
11         uint voteCount;
12     }
13
14     address chairperson;
15     mapping(address => Voter) voters;
16     Proposal[] proposals;
17
18     /// Create a new ballot with $(_numProposals) different proposals.
19     function Ballot(uint8 _numProposals) public {
20         chairperson = msg.sender;
21         voters[chairperson].weight = 1;
22         proposals.length = _numProposals;
23     }
24
25     /// Give $(toVoter) the right to vote on this ballot.
26     /// May only be called by $(chairperson).
27     function giveRightToVote(address toVoter) public {
```

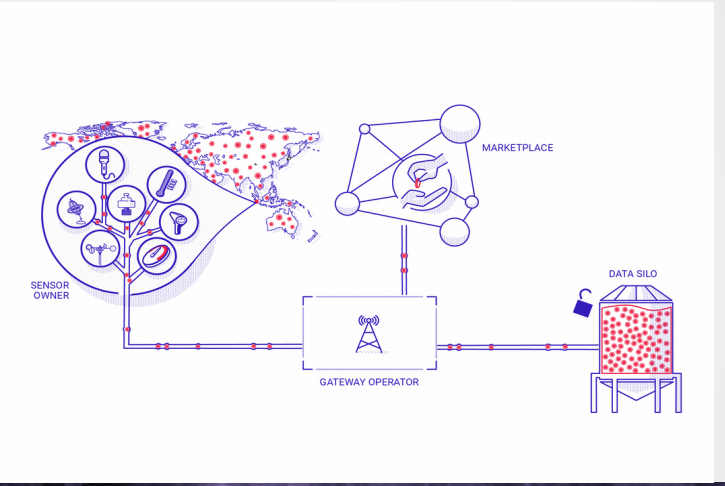
Program Execution Cost

- Gas
  - Transactions on the Ethereum blockchain are required to cover cost of computation they are performing
    - 3 gas - <, >, =
      5 gas - *, /
      50 gas – SHA3
      53000 – contract creation
  - Gas : Gwei
  - Transaction fee = gasprice * consumedgas
  - If startgas is less than needed
    - Out of gas exception, revert the state as if the transaction has never happened
    - Sender still pays all the gas

Ethereum Overview

- Today's applications
  - Cryptokitties

  - Selling/buying IoT sensor data



  - Toll tag logging and payment

Ethereum Overview

Questions?

John Lindsay
Patent Attorney
Coaster, Smooth Driver Application