SHAYAN ESKANDARI, JEREMY CLARK, VIGNESH SUNDARESAN, MOE ADHAM

ON THE FEASIBILITY OF DECENTRALIZED DERIVATIVE MARKETS

1st Workshop on Trusted Smart Contract
In Association with Financial Cryptography 17
Malta - April 07, 2017

bitaccess

UNIVERSITÉ Concordia UNIVERSITY

DAPPS ARE COOL, EH?

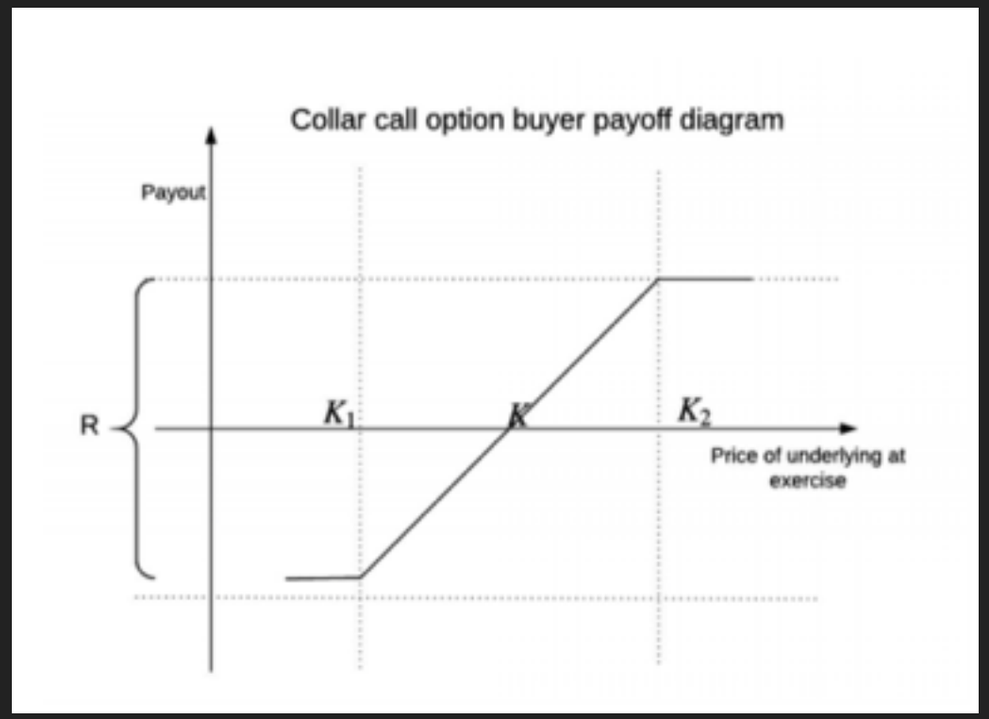
- Blockchain will be a thing, sometime in the future (IPFS, DNSChain, …)
- It seems like Decentral Applications, a.k.a Smart Contracts, will be too
- What is the state now?
- let’s experiment

WHY DECENTRALIZED DERIVATIVE MARKETS?

- Derivative markets are often cited as a potential target
- First time in history that we have infrastructures/testbeds to implement real smart contracts [Szabo], e.g Ethereum, RSK
- Fintech is also a cool thing now, specially when blockchain is involved

DERIVATIVES

- Two parties enter an agreement
  - The first stands to profit if a specified security (e.g., stock) appreciates in value over a specified time-period
  - the second stands to profit if it falls
- State of derivatives:
  - Need a broker
    - Trust the 3rd party (Money transaction and derivative settlements)



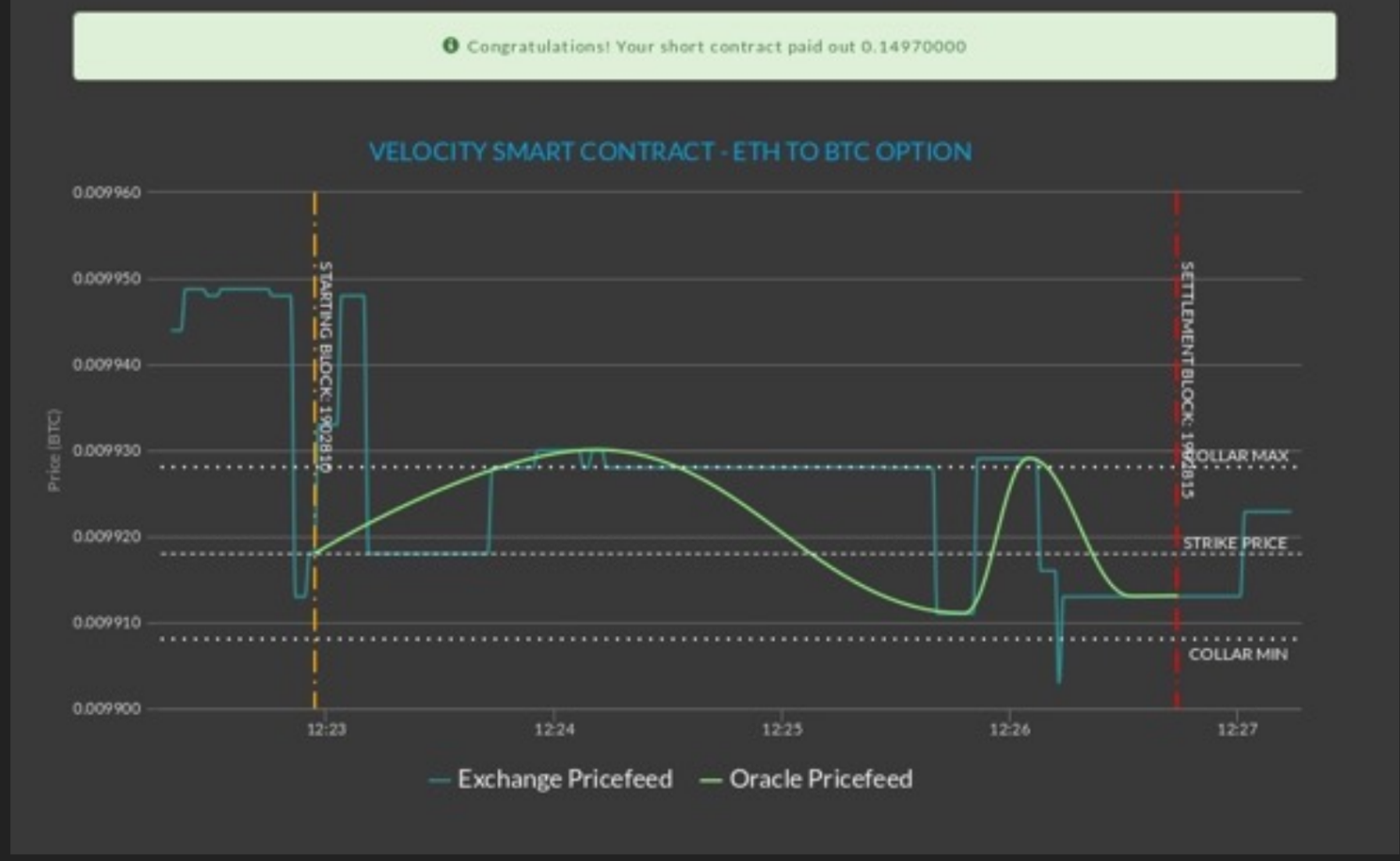
DESIGN CHALLENGES

- Terms of the contract
  - Implemented in Solidity
- Counter party risk
  - Decentralization (Replace the broker with smart contract)
  - Capped Reward (2x)
- Price Feed, Oracles
- Underlying Financial Model
  - inflations/deflation of ETH might change the real outcome

TERMS OF THE CONTRACT
SIMPLIFIED OPTIONS CALL, POC

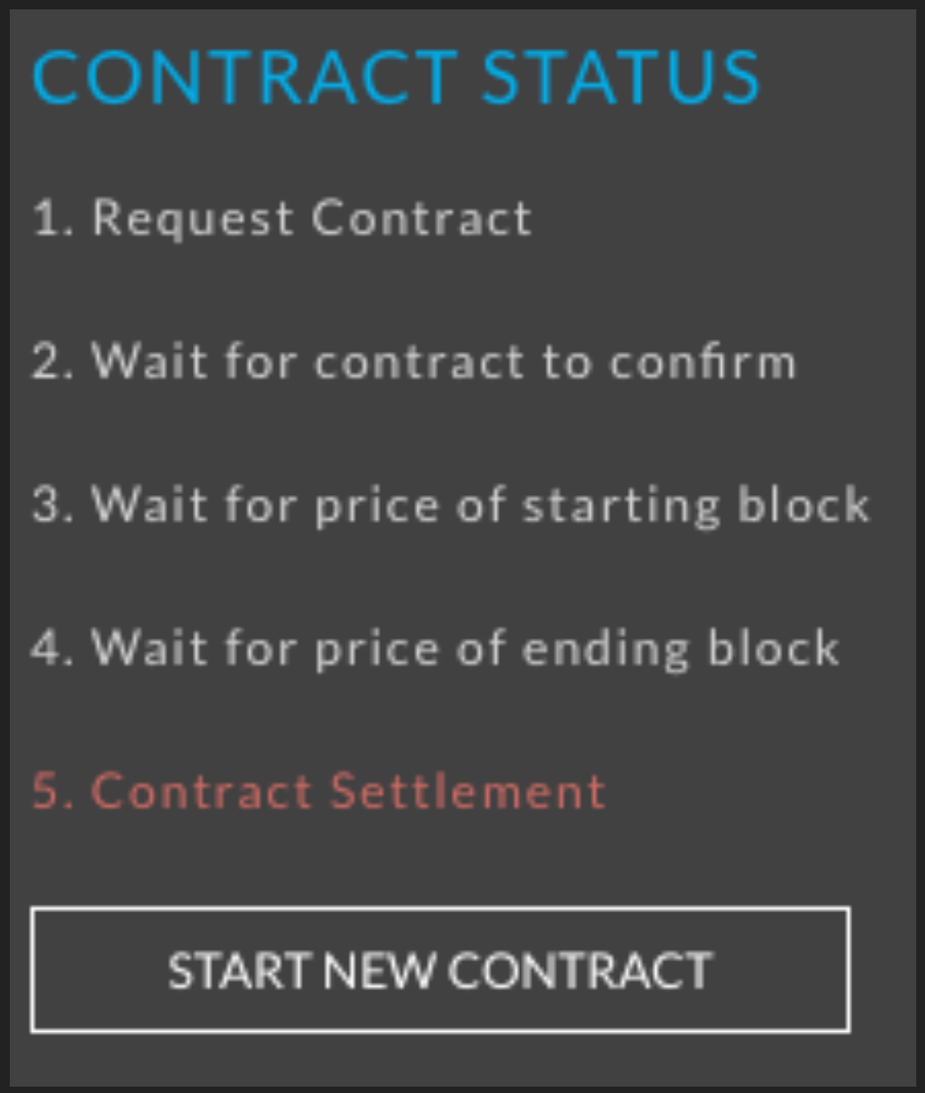
1. Alice enters a contract by **sending** the deposit to either GoLong() or GoShort() for the specified price pair.
2. Bob takes the opposing position and **sends** his deposit
3. Any of the parties **call** settle() anytime after contract's expiry time
4. Smart Contract checks the prices and pays out

```
function goLong() public
                 hasEnoughFunds(msg.value)
                 checkMargin(msg.value)
                 payable
                 returns(uint) {
    lastOptionId = newOption(msg.sender, msg.value, true);
    LongOption(lastOptionId, msg.sender, msg.value, block.number);
    return lastOptionId;
    }
```



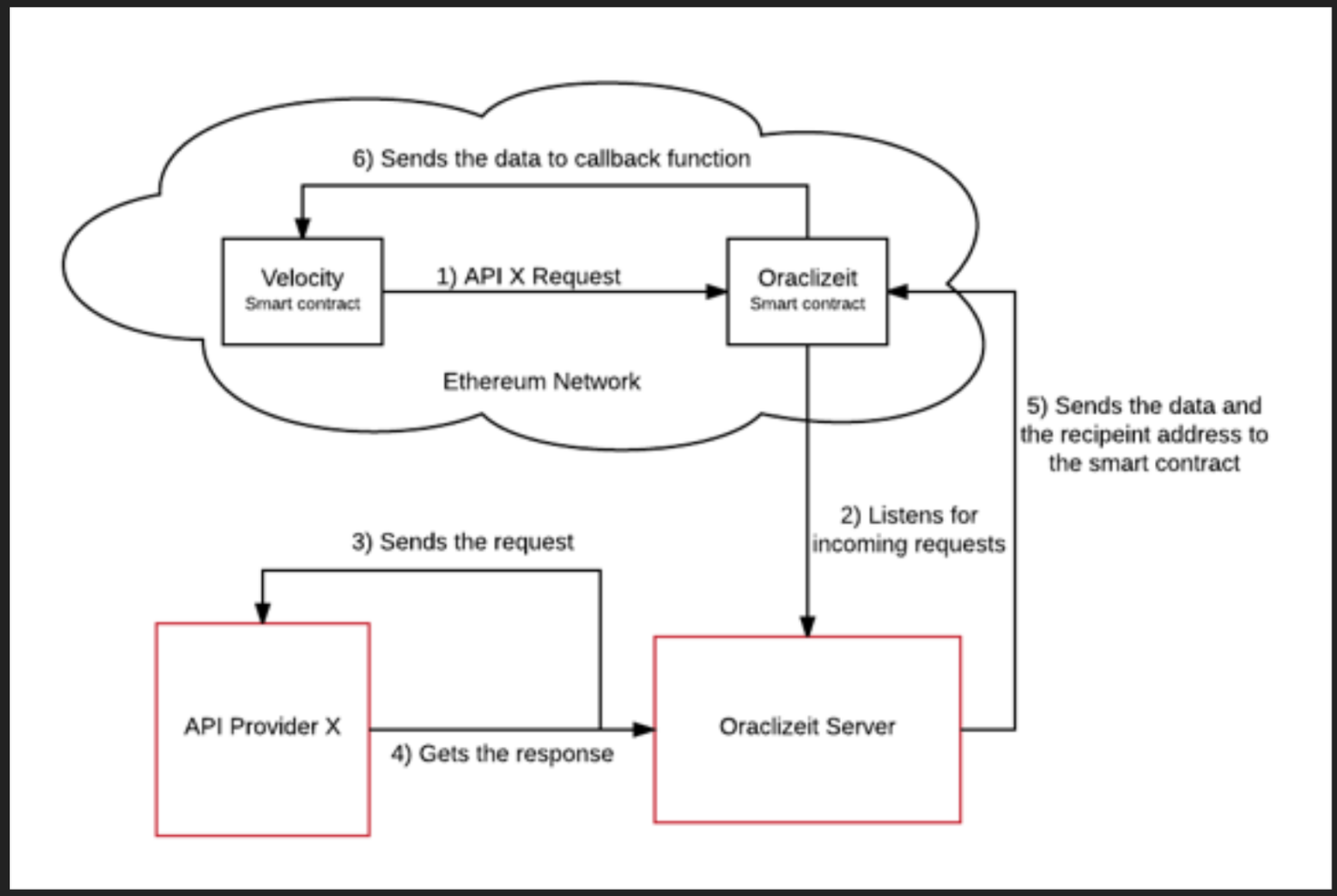
OPTIONS SMART CONTRACT (DEPLOYED ON ETHEREUM TESTNET)

- Deposit: 0.1 ETH
- ETH/BTC pair
- Smart contract acts as Bob and takes the opposing position and escrows the funds
- Expiry time: 5 Ethereum blocks
- Alice "should" **settle()**
  - Rejects if not expired
  - No incentive to settle a loss
    - Smart contract settles the first open contract for *msg.sender*
      - *settle_all() script*



PRICE FEED

- What was out there?
  - Smart Contract oracles (smartcontracts.com)
    - Updates daily
  - Oraclize
    - Call and Callback
    - Central Blackbox Solution
    - Starting price and
      Expiry price (2 calls) -> Expensive -> Exercise() -> (mostly) runs out of gas
    - TLSNotary- proof [optional]
    - Sometimes it needs to email support (decentralized Support request lol)



bitaccess

PRICE FEED

- What was out there?
  - Smart Contract oracles (smartcontracts.com)
    - Updates daily
  - Oraclize
    - Call and Callback
    - Central Blackbox Solution
    - Starting price and
      Expiry price (2 calls) -> Expensive -> Exercise() -> (mostly) runs out of gas
    - TLSNotary- proof [optional]
    - Sometimes it needs to email support (decentralized Support request lol)

```
3  //oraclize_setNetwork(2); //
4  priceUrl = "json(https://www.bitstamp.net/api/v2/ticker/btcusd).last";
5  function updateBTCUSDFromFeed(uint delay){
6    oraclize_query(delay, "URL",
7      priceUrl, 400000);
8    }
9  function __callback(bytes32 myid, string result, bytes proof) {
10   if (msg.sender != oraclize_cbAddress()) throw;
11     uint BTCUSDFeed;
12     BTCUSDFeed = parseInt(result, 2);
13   exercise() // this function exercises the contract to calculate the
     ↪  payouts
14   }
```

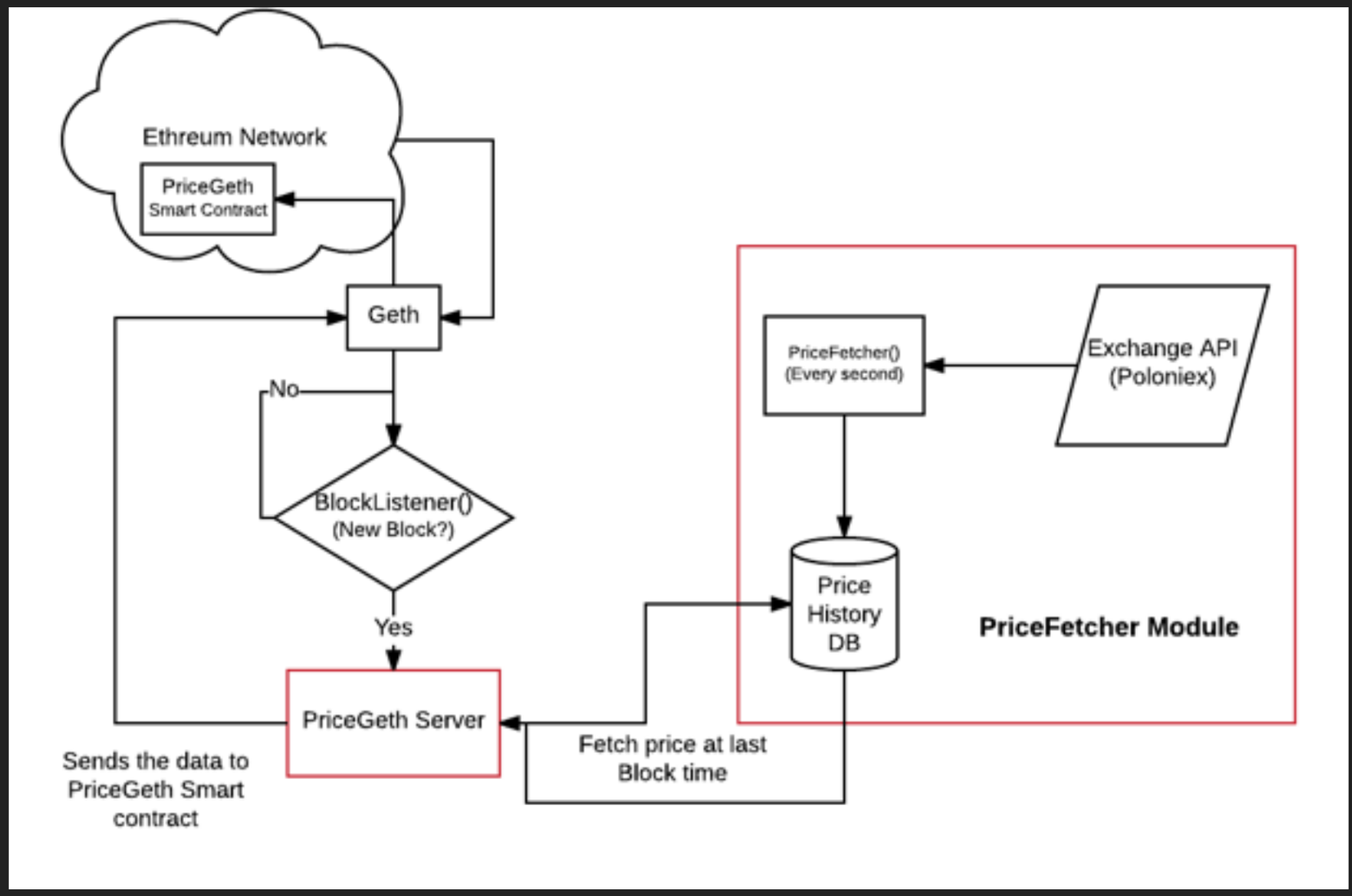
PRICE FEED (CONT) – PRICEGETH

- We designed a new Price oracle (https://github.com/VelocityMarket/pricegeth/)
- Publish price pairs on every blocktime on Ethereum blockchain
- Free for all smart contracts to use, even historical data
- Publisher pays the gas - incentive?
- Publisher can implement a token (ERC20) to get paid for API calls

```
{ address: '0x685c662cE0779ea3b6bBA84948CA08F04Fc877ff',
  blockHash: '0xee272a6048d9a583d610890de408981eacfe0cbd7bab107f00be9da51288ea60',
  blockNumber: 1667534,
  logIndex: 1,
  transactionHash: '0x596adc436fce0432fada47819203ab6835da3e73199e9db71083bf417a01d497',
  transactionIndex: 1,
  event: 'PriceUpdated',
  args:
   { timestamp: { [String: '1474324837'] s: 1, e: 9, c: [Object] },
     blocknumber: { [String: '1667532'] s: 1, e: 6, c: [Object] },
     USDBTC: { [String: '6100000076699'] s: 1, e: 12, c: [Object] },
     BTCETH: { [String: '211199900'] s: 1, e: 8, c: [Object] },
     BTCETC: { [String: '20989899'] s: 1, e: 7, c: [Object] },
     BTCDOGE: { [String: '3900'] s: 1, e: 3, c: [Object] } } }
```

PRICE FEED (CONT) – PRICEGETH

- **SPOILER**: Still not Decentralized enough!
- Can use Intel SGX to be more secure, but still we have single source of information
- Used Python (web3.py), NodeJS (web3.js) and Solidity



ANYWAYS... DEMO

ANYWAYS... DEMO

DEPLOYMENT AND SECURITY OF DAPPS

- Not fun to test, fix, deploy, test again
- Payout best practice and logical bugs
- “Known” security issues (Reentrancy Vulnerability a.k.a DAO bug, etc)
- Security Analysis tools, Oyente [Luu]

DEPLOYMENT AND SECURITY OF DAPPS

- Not fun to test, fix, deploy, test again
- Payout best practice and logical bugs
- “Known” security issues (Reentrancy Vulnerability a.k.a DAO bug, etc)
- Security Analysis tools, Oyente [Luu]

```
function payAndHandle(uint optionId, address addr, uint amount)
         private
         returns (bool success) {
  if (addr.send(amount)) {
        optionPaid(optionId, addr, amount); //event for successfu
  } else { throw;}
  return true;
}
```

DEPLOYMENT AND SECURITY OF DAPPS

- Not fun to test, fix, deploy, test again
- Payout best practice and logical bugs
- “Known” security issues (Reentrancy Vulnerability a.k.a DAO bug, etc)
- Security Analysis tools, Oyente [Luu]

```
function payAndHandle(uin
         private
         returns (bool su
  if (addr.send(amount))
        optionPaid(option
  } else { throw;}
  return true;
}
```

```
69  +  modifier isOpen(uint optionId) {if (AllOptions[optionId].closed) throw; _ ;}
70
71     //events
72     event Error(string message);

155       exercise(findOptionId(msg.sender));
156    }
157
158 +  function exercise(uint optionId) public isOpen(optionId) returns(bool) {
159       //LogMe("exercise called");
160       PriceDetails memory pricesToCheck;
161       (pricesToCheck.priceAtBlockStarted, ,pricesToCheck.blockStarted) =
getBTCETH(AllOptions[optionId].StartedAtBlock);
```

DEPLOYMENT AND SECURITY OF DAPPS

- Not fun to test, fix, deploy, test again
- Payout best practice and logical bugs
- “Known” security issues (Reentrancy Vulnerability a.k.a DAO bug, etc)
- Security Analysis tools, Oyente [Luu]

```
function payAndHandle(u
         private
         returns (bool
  if (addr.send(amount)
       optionPaid(opti
  } else { throw;}
  return true;
}
```

```
python oyente.py --error ./finaloptions.sol
Contract ./finaloptions.sol:finalOptions:
Running, please wait...
        ========== Results ==========
          CallStack Attack:      False
          Concurrency Bug:       False
          Time Dependency:       False
          Reentrancy bug exists: False
        ====== Analysis Completed ======
Contract pricegeth.sol:Pricegeth:
Running, please wait...
        ========== Results ==========
          CallStack Attack:      False
          Concurrency Bug:       False
          Time Dependency:       False
          Reentrancy bug exists: False
        ====== Analysis Completed ======
```

```
lOptions[optionId].closed) throw; _ ;}

c isOpen(optionId) returns(bool) {

,pricesToCheck.blockStarted) =
lock);
```

MORE DISCUSSIONS

- Solidity
  - Updates are “hard forks”!
  - Gas Sustainability
    - Storage vs Memory vs …
  - Local Variable limits (16)
- Ethereum Testnet (Morden, moved to Ropsten): http://demo.velocity.technology
- Collar Option library (GPL) (https://github.com/VelocityMarket/Options-Contract)

THANK YOU

Shayan Eskandari
shayan@bitaccess.co
https://twitter.com/sbetamc
github.com/VelocityMarket