ON THE FEASIBILITY OF
DECENTRALIZED DERIVATIVE MARKETS

SHAYAN ESKANDARI, JEREMY CLARK, VIGNESH SUNDARESAN, MOE ADHAM

1st Workshop on Trusted Smart Contract
In Association with Financial Cryptography 17
Malta - April 07, 2017
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.
ON THE FEASIBILITY OF DECENTRALIZED DERIVATIVES MARKETS

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
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.
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

CONTRACT STATUS
1. Request Contract
2. Wait for contract to confirm
3. Wait for price of starting block
4. Wait for price of ending block
5. Contract Settlement

START NEW CONTRACT
PRICE FEED

- What was out there?
  - Smart Contract oracles ([smartcontracts.com](http://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)
PRICE FEED

- What was out there?
  - Smart Contract oracles ([smartcontracts.com](https://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)
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
ON THE FEASIBILITY OF DECENTRALIZED DERIVATIVES MARKETS

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
ON THE FEASIBILITY OF DECENTRALIZED DERIVATIVES MARKETS

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]
ON THE FEASIBILITY OF DECENTRALIZED DERIVATIVES MARKETS

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]

```solidity
def payAndHandle(uint optionId, address addr, uint amount)
private
    returns (bool success) {
        if (addr.send(amount)) {
            optionPaid(optionId, addr, amount); // event for successful
        } else { throw; }
        return true;
    }
```
ON THE FEASIBILITY OF DECENTRALIZED DERIVATIVES MARKETS

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]

```solidity
function payAndHandle(uint amount) private returns (bool success) {
    if (addr.send(amount)) {
        optionPaid(optionId);
    } else { throw;
}

function exercise(uint optionId) public isOpen(optionId) returns (bool) {
    //LogMe("exercise called");
    PriceDetails memory pricesToCheck;
    (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]
ON THE FEASIBILITY OF DECENTRALIZED DERIVATIVES MARKETS

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)
ON THE FEASIBILITY OF DECENTRALIZED DERIVATIVES MARKETS

THANK YOU

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