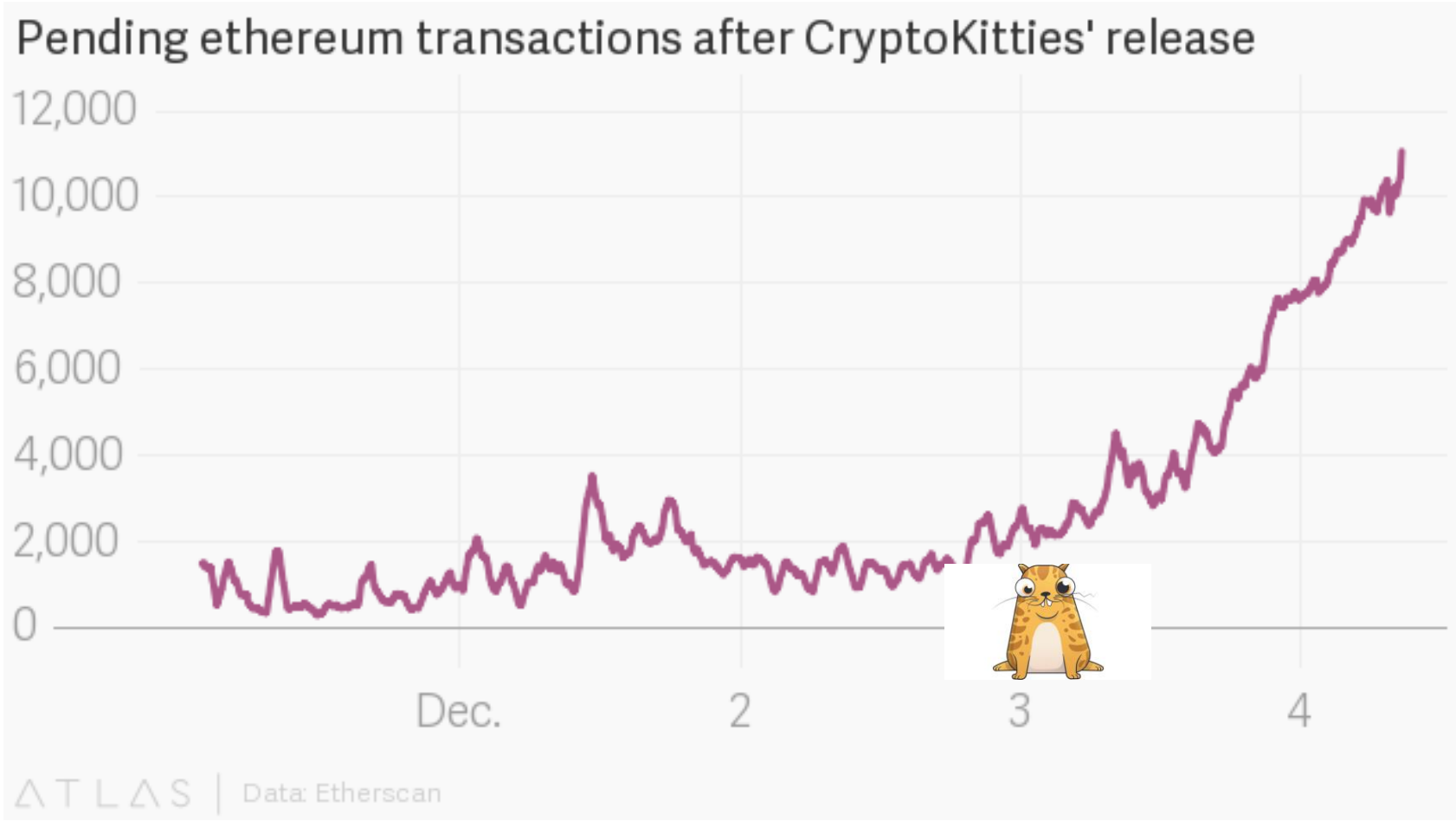
Scale or Perish?

Payment Channel, State Channel & Plasma

For discussion purposes

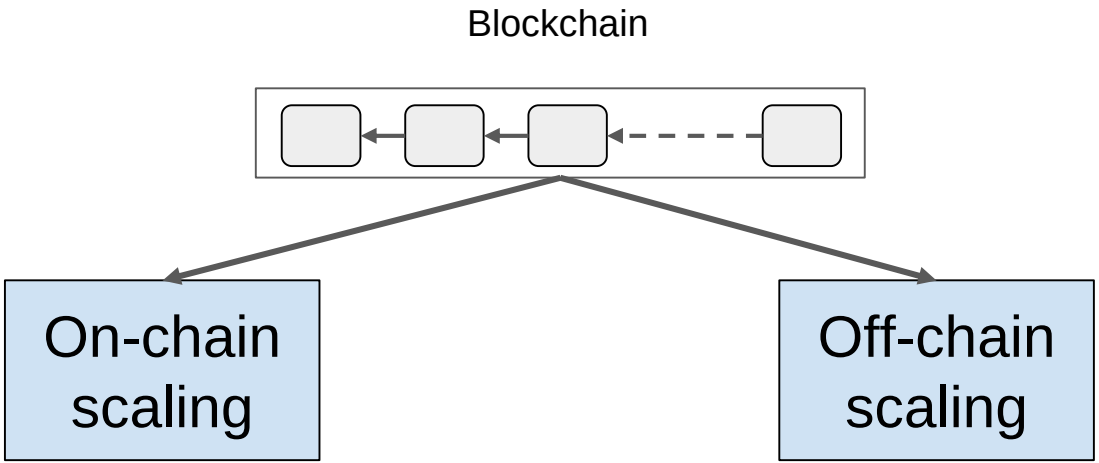
DBSystem, SoC, NUS


Pending ethereum transactions after CryptoKitties' release
12,000
10,000
8,000
6,000
4,000
2,000
0
Dec.
2
3
4
ATLAS
Data: Etherscan

Agenda

1. Overview
2. Payment channels
3. State channel
4. Plasma
5. Discussion

Overview

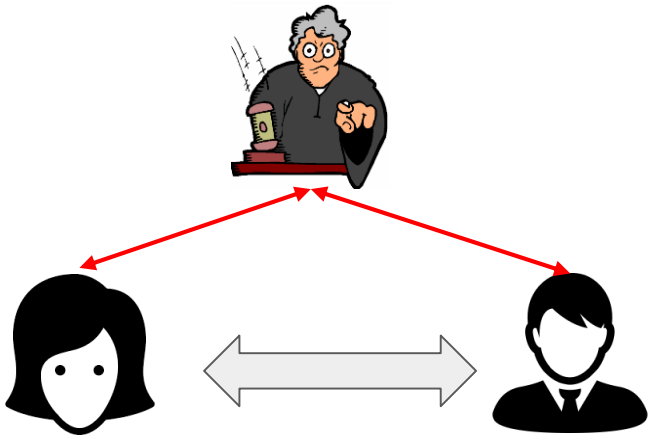

Blockchain
On-chain scaling
Off-chain scaling
Sharding,
trusted hardware,
etc.
Sidechains,
Payment channels,
State channels,
TrueBit,
Plasma

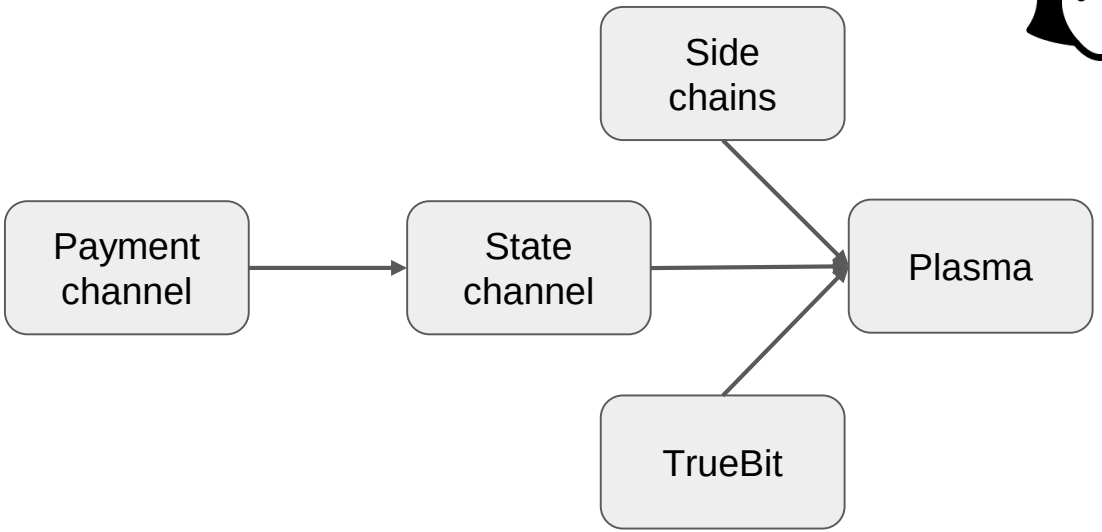
Overview

- On-chain scaling
  - More blocks per second
  - Scale consensus
- Off-chain scaling
  - **Avoid** transactions on blockchain as much as possible
  - Why?
    - Better latency
    - Finality
    - More transaction volumes

Overview

- Off-chain scaling: recurring theme
  - Off-chain communications
  - Blockchain as a fair judge
- History



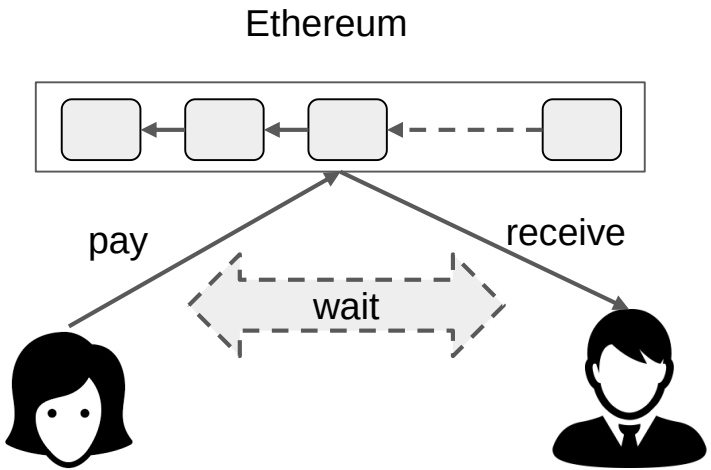


Agenda

1. Overview
2. **Payment channel**
3. State channel
4. Plasma
5. Discussion

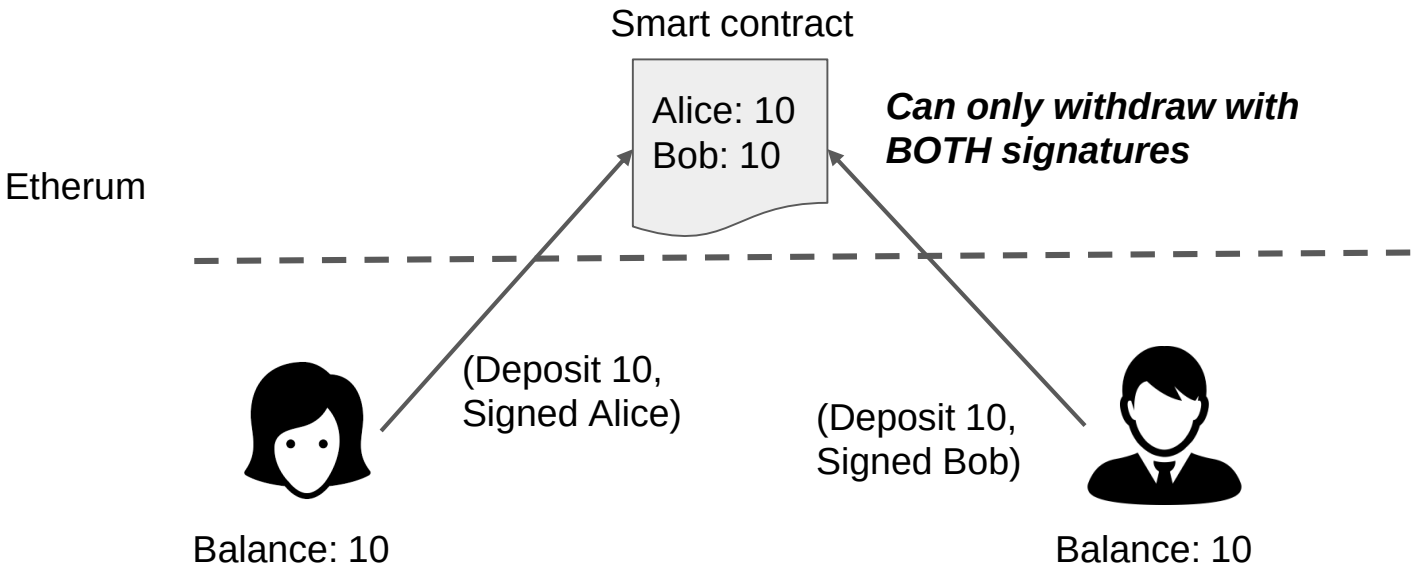
Payment channel - Problem

- Alice wants to pay Bob 0.0001 ETH
- Problems:
  - Fees (0.2-5 USD)
  - Wait for many confirmation blocks (~hours)
  - Cryptokitties



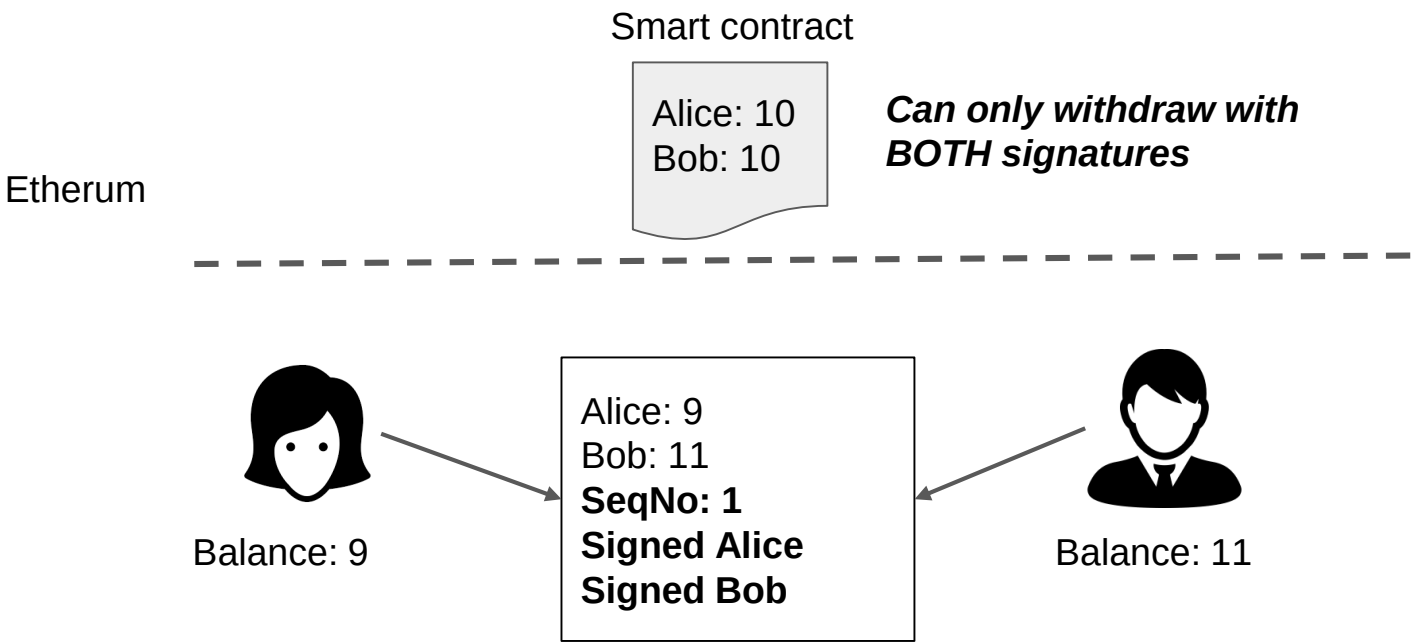
Payment channel - Solution

- Ethereum version of Lightning network
- Setup



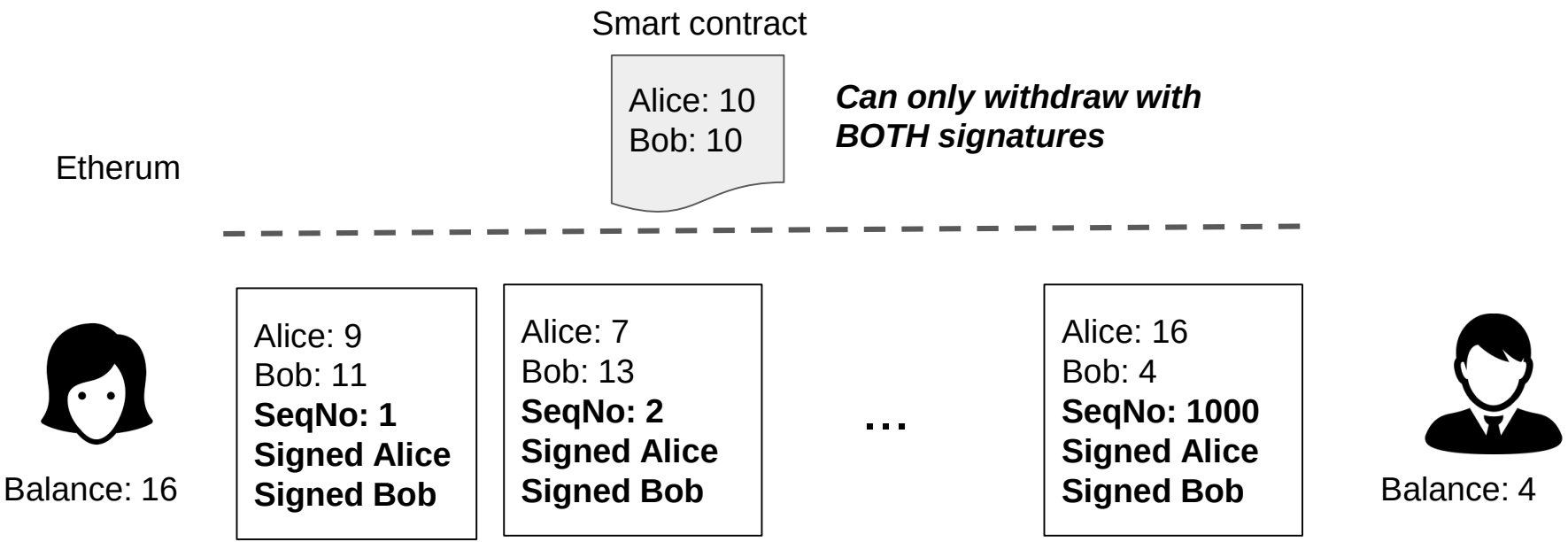
Payment channel - Solution

- Pay

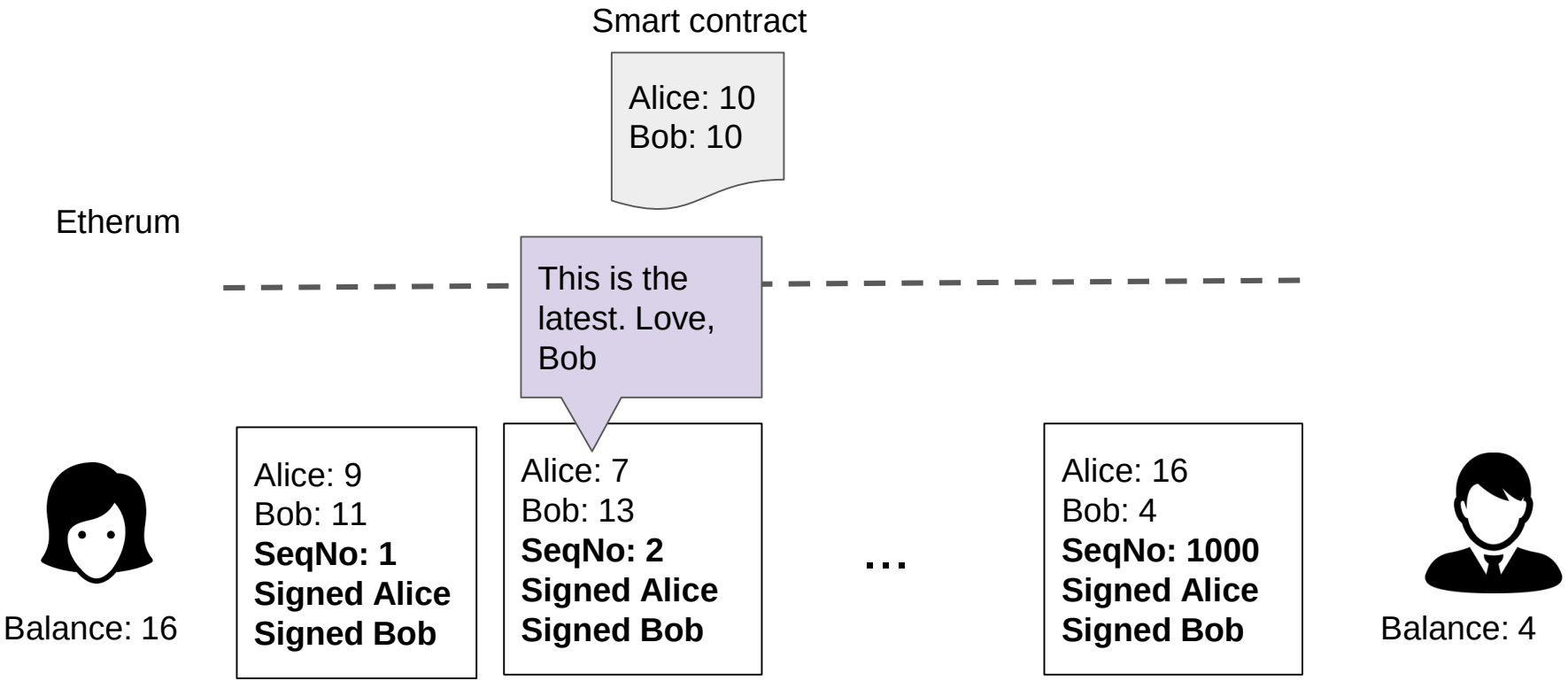


Payment channel - Solution

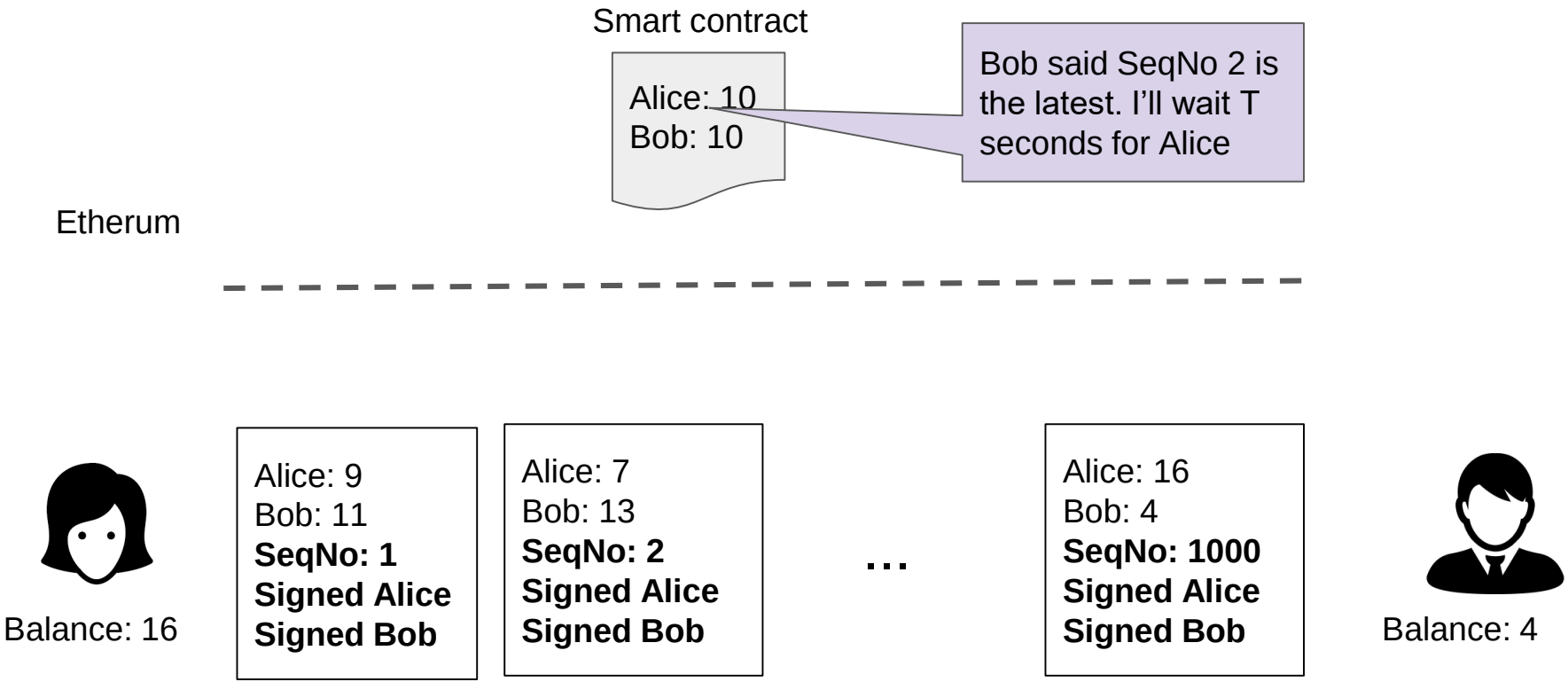
- Many payments latter



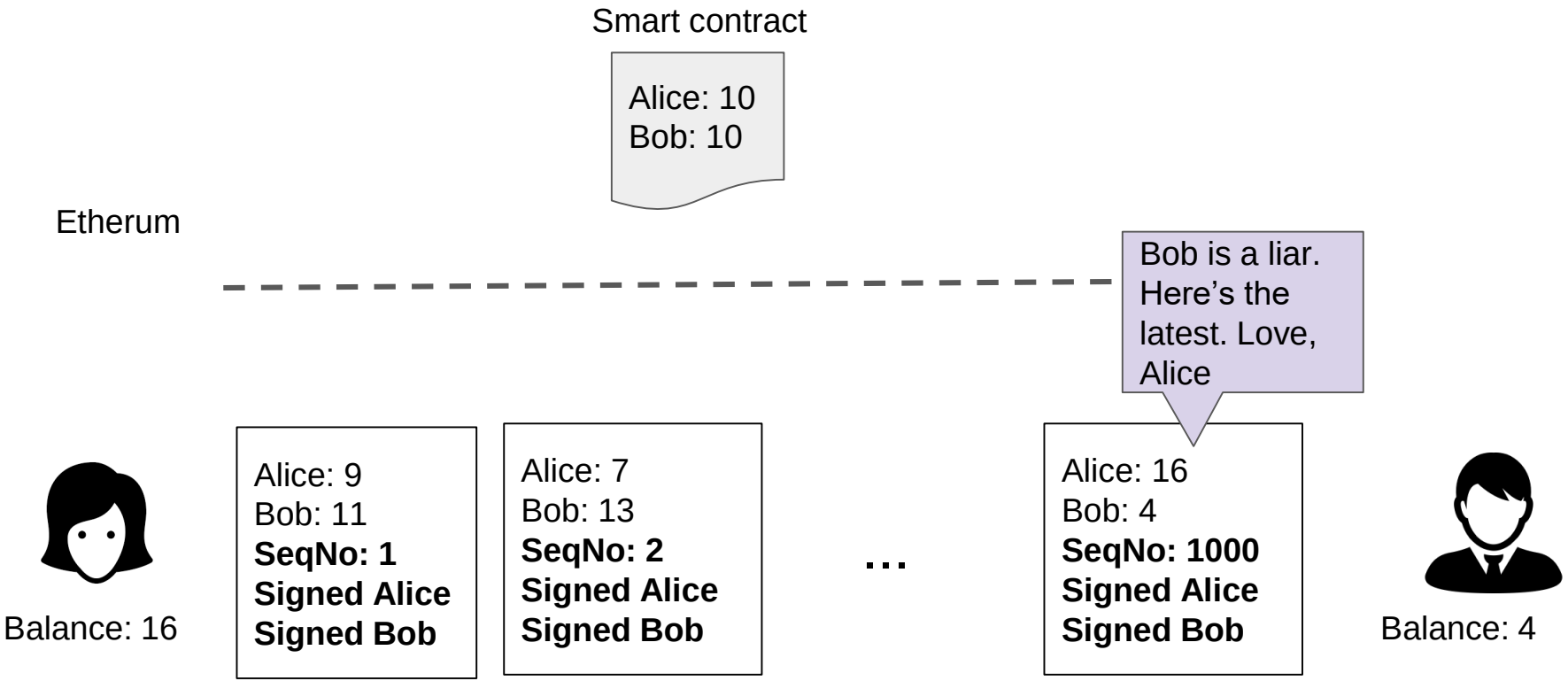
Payment channel - Solution



Payment channel - Solution



Payment channel - Solution



Payment channel - Solution

Smart contract

Alice: 16
Bob: 4

SeqNo 1000 is latest. Pay out 16 to Alice, 4 to Bob.

Etherum

Balance: 16

Alice: 9
Bob: 11
SeqNo: 1
Signed Alice
Signed Bob

Alice: 7
Bob: 13
SeqNo: 2
Signed Alice
Signed Bob

...

Alice: 16
Bob: 4
SeqNo: 1000
Signed Alice
Signed Bob

Balance: 4

Payment channel

- Instant confirmation: as soon as both parties sign
- 2 blockchain transactions per channel: open & close
  - Virtually unlimited # off-chain transactions
- Security
  - Rebuttal period T is important
  - Parties are rational
- **Only does payment**

Agenda

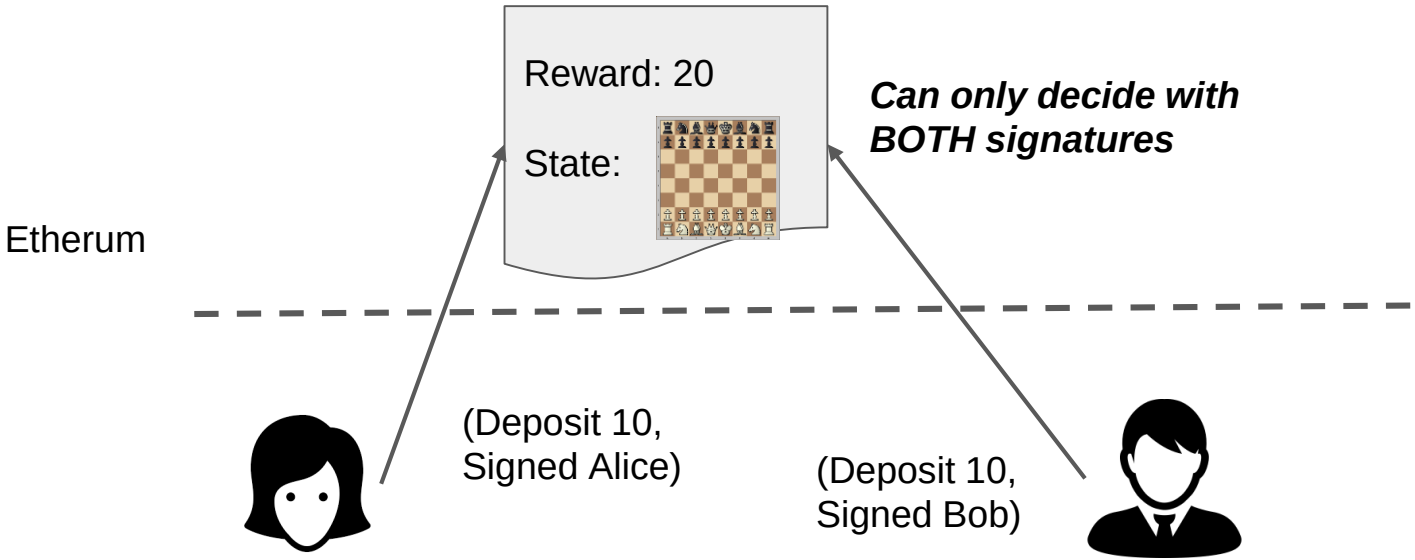
1. Overview
2. Payment channel
3. **State channel**
4. Plasma
5. Discussion

State channel - Problem

- Generalize payment channel for any joint computation:
  - Example: Alice and Bob want to play chess on the blockchain!
- Strawman:
  - A contract to encode the rule and current state of the game
  - Player takes turn to send transactions
- Problems:
  - Take too long
    - Longest official chess game takes 286 moves (20h 15m)
  - Expensive

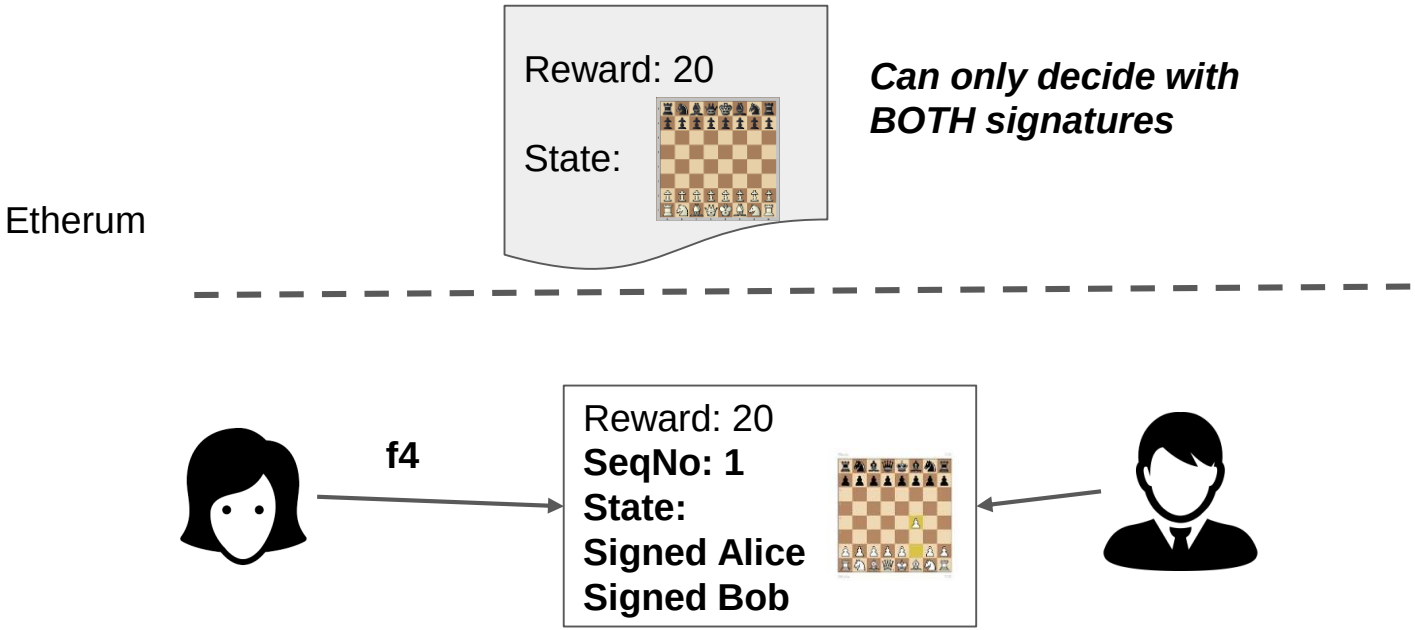
State channel - Solution

- Setup



State channel - Solution

- First move



State channel - Solution

- Many moves later

Reward: 20

State: 

Can only decide with BOTH signatures

Etherum

Reward: 20
SeqNo: 1
State: 
Signed Alice
Signed Bob

Reward: 20
SeqNo: 2
State: 
Signed Alice
Signed Bob

Reward: 20
SeqNo: 100
State: 
Signed Alice
Signed Bob

State channel - Solution

Reward: 20

State: 

Etherum

This is the latest. Love, Bob

Reward: 20
SeqNo: 1
State: 
Signed Alice
Signed Bob

Reward: 20
SeqNo: 2
State: 
Signed Alice
Signed Bob

Reward: 20
SeqNo: 100
State: 
Signed Alice
Signed Bob

State channel - Solution

Reward: 20

State: 

Bob said latest SeqNo=2. I'll wait T seconds for Alice.

Etherum

Reward: 20
SeqNo: 1
State: 
Signed Alice
Signed Bob

Reward: 20
SeqNo: 2
State: 
Signed Alice
Signed Bob

Reward: 20
SeqNo: 100
State: 
Signed Alice
Signed Bob

State channel - Solution

Reward: 20

State: 

Etherum

Bob lied. Here's the latest state. Love, Alice

Reward: 20
SeqNo: 1
State: 
Signed Alice
Signed Bob

Reward: 20
SeqNo: 2
State: 
Signed Alice
Signed Bob

Reward: 20
SeqNo: 100
State: 
Signed Alice
Signed Bob

State channel - Solution

Alice won at SeqNo=100. Pay out 20ETH to Alice

Reward: 20

State: 

Etherum

Reward: 20
SeqNo: 1
State: 
Signed Alice
Signed Bob

Reward: 20
SeqNo: 2
State: 
Signed Alice
Signed Bob

Reward: 20
SeqNo: 100
State: 
Signed Alice
Signed Bob

State channel

- Ad-hoc state channel: one channel per game
  - Smart contract encoding rule + how to decide on final outcomes
  - Players sign the latest state off-chain

- Generalized state channel:
  - General channel (smart contract) that allow creating ad-hoc channel
    - Using another smart contract as contract registry
  - Counterfactual.com

State channel

- Huge implication to security:
  - Solve the **fair secure multi-party computation (MPC)** problem, which is impossible without blockchain
- Active research (security):
  - Virtual channels
  - Outsourcing: blockchain watchers

- Assumed:
  - **Fixed set of participants**
  - **Rational players**

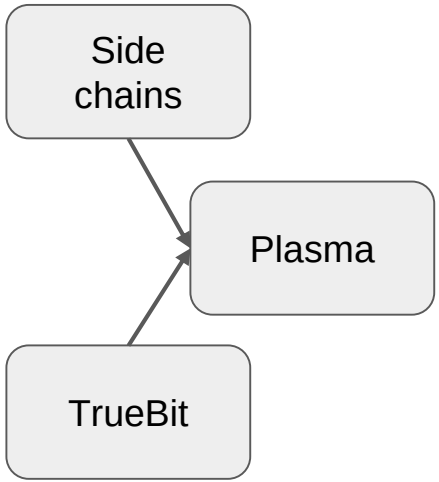
Agenda

1. Overview
2. Payment channel
3. State channel
4. **Plasma**
5. Discussion

Plasma

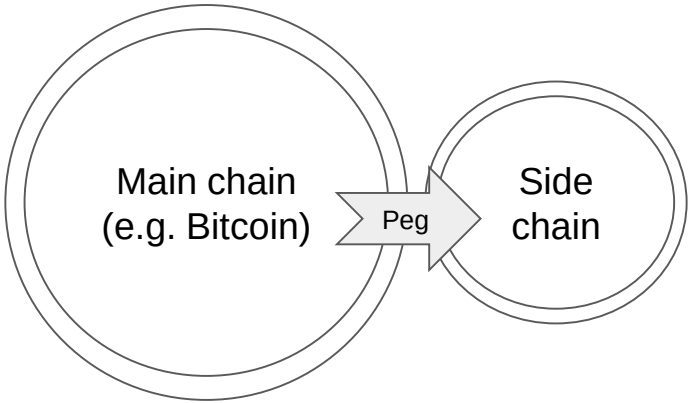
- Problem with State Channels
  - All players must participate
    - Abort = expensive

- Building block 1: sidechain
  - Application-specific blockchain can be a separate blockchain
  - e.g. Cryptokitties and ICOs have their own chains
  - Decision by consensus -> don’t need everyone to participate



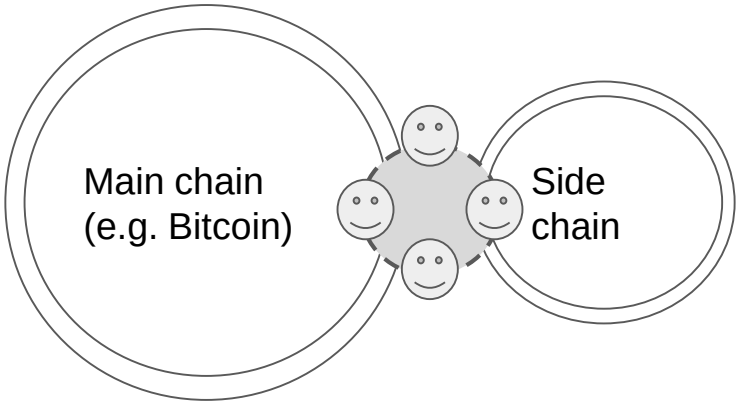
Plasma - Sidechain 101

- Early idea, by *Blockstream*: pegged side chain
  - For experimental design
  - Burn coin from one chain to generate coin on another
  - **Sidechain is independent of main chain**
    - **e.g. can run PBFT consensus**



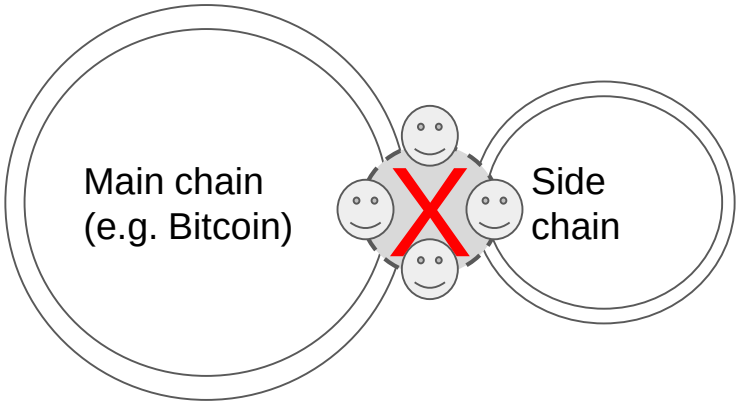
Plasma - Sidechain 101

- Federated sidechain:
  - Coins moving both ways
  - A set of validators are members of both chains
  - Still, two chains are independent



Plasma

- Plasma = sidechain without federation of validators
- Then how to ensure security in the sidechain?
- Building block 2: TrueBit

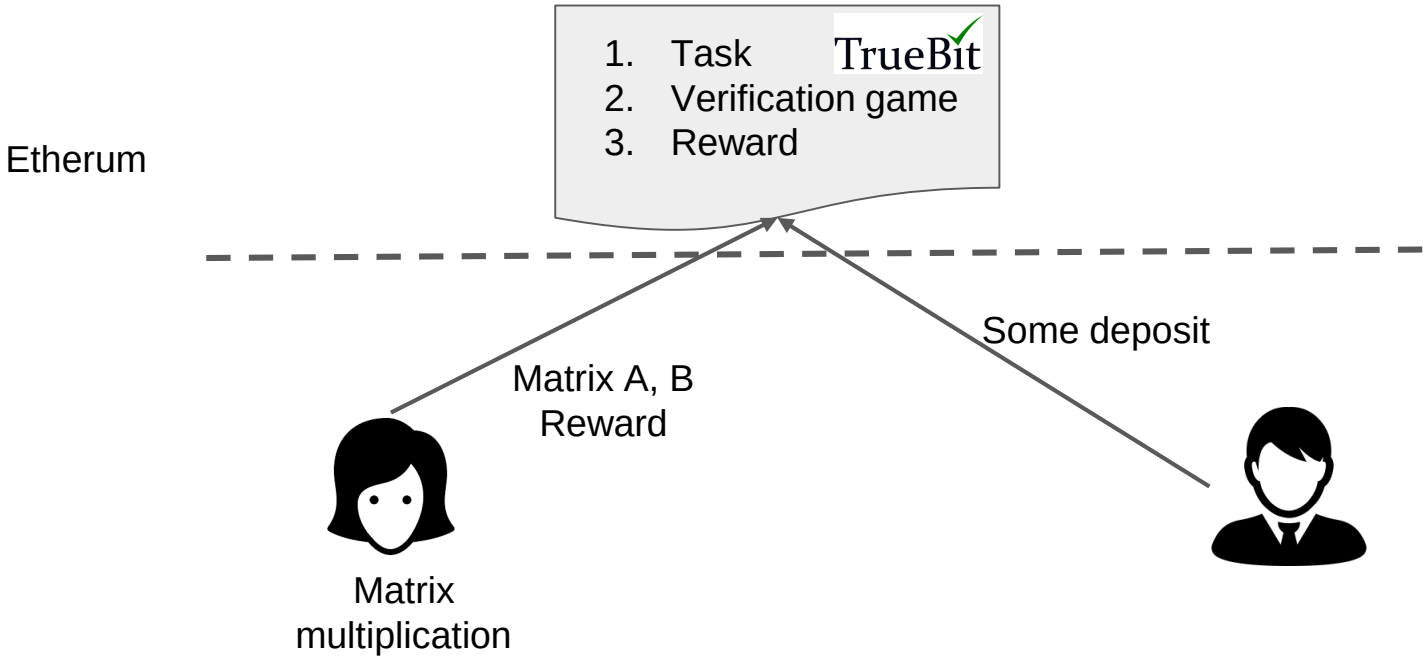


Plasma - TrueBit 101

- Setting: outsourced computation on blockchain (!?)
  - e.g. sorting, matrix multiplication
- Problem: minimizing verification cost when given a solution
  - Benefit: same throughput, but for expensive computations

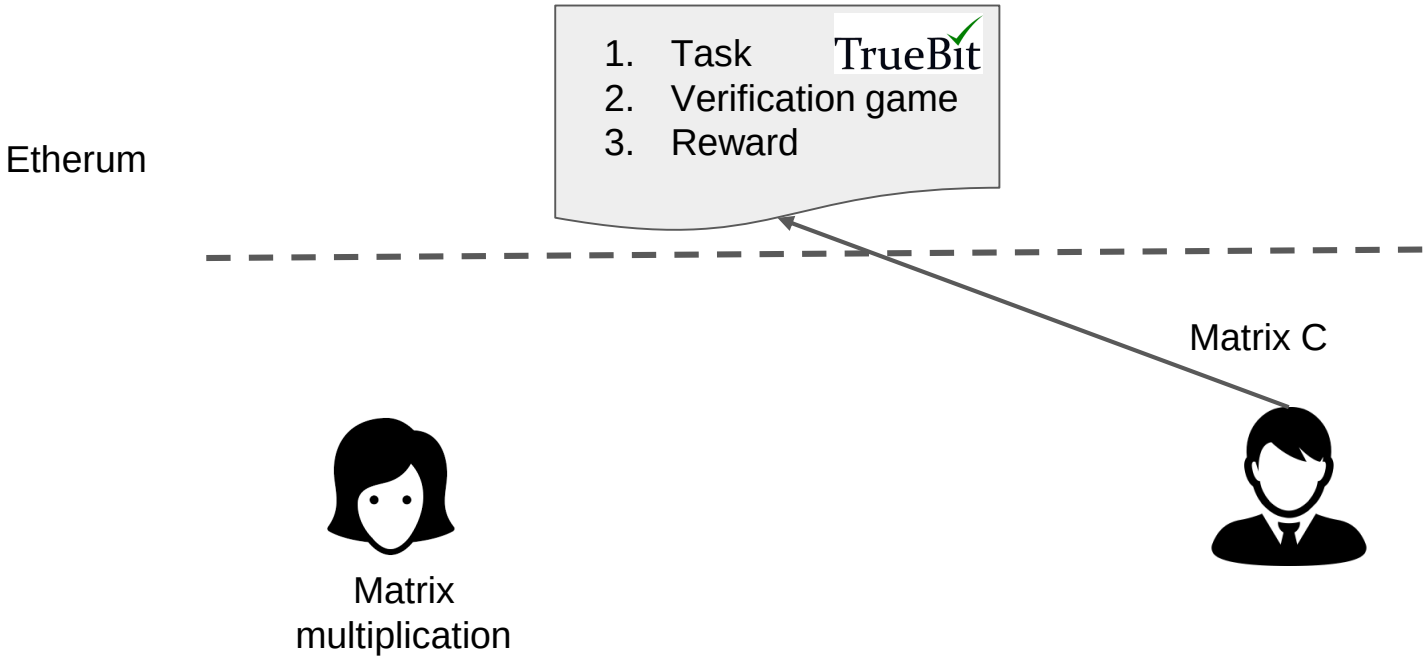
Plasma - TrueBit 101

- TrueBit is a smart contract implementing a verification game



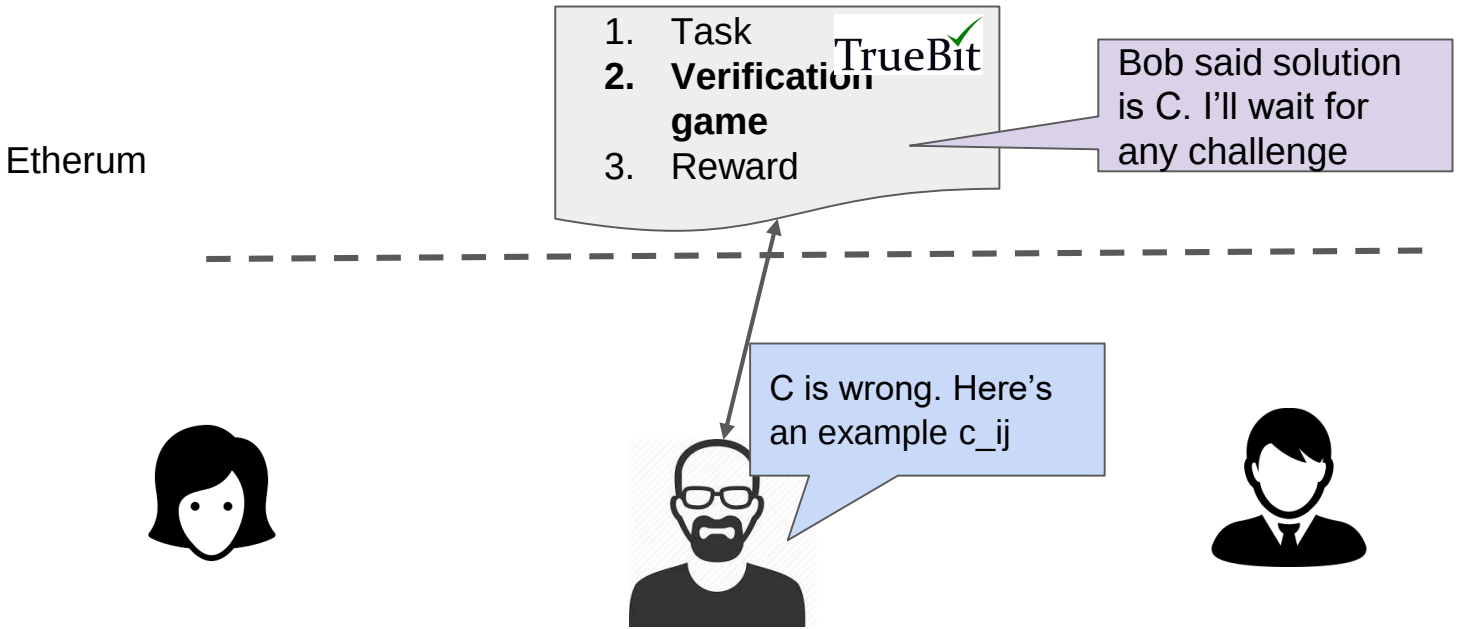
Plasma - TrueBit 101

- TrueBit is a smart contract implementing a verification game



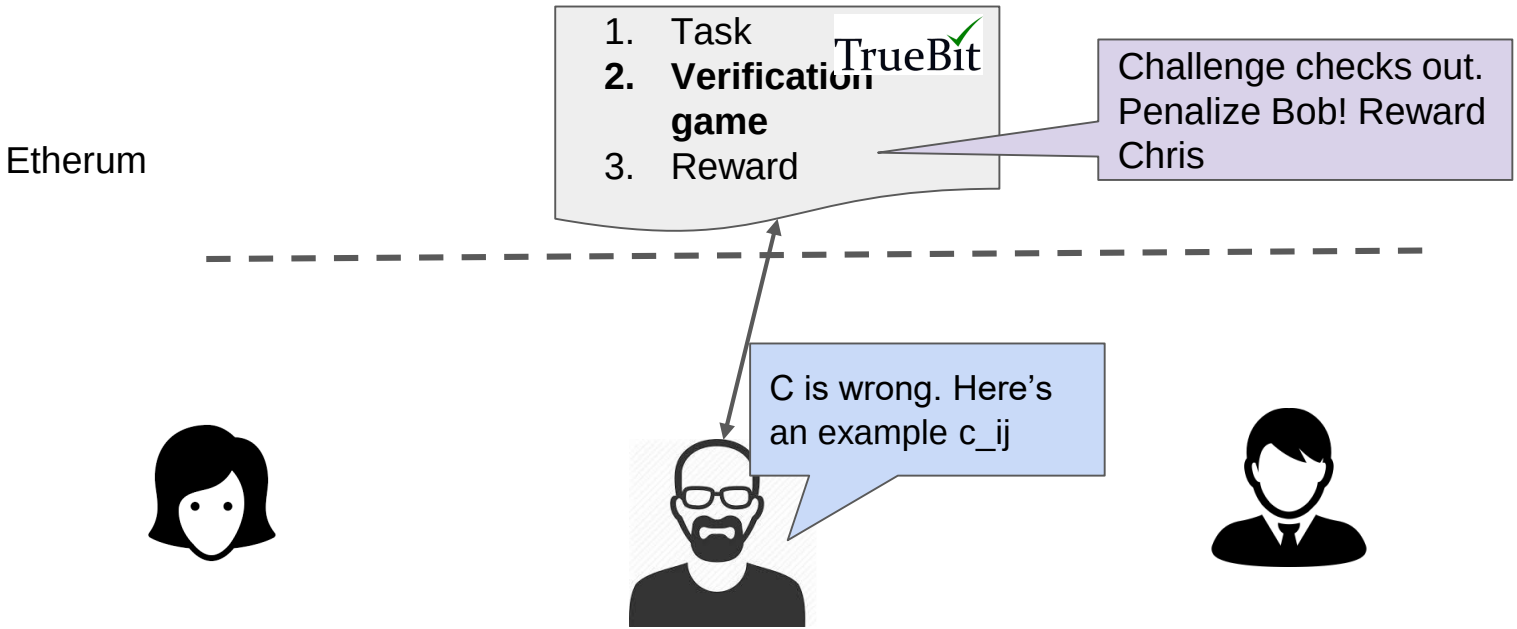
Plasma - TrueBit 101

- TrueBit is a smart contract implementing a verification game



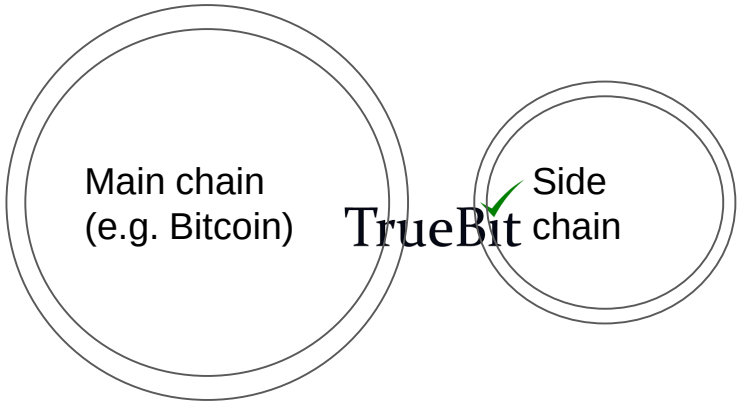
Plasma - TrueBit 101

- TrueBit is a smart contract implementing a verification game



Plasma - TrueBit

- TrueBit-compatible tasks:
  - Verification is cheaper than execution
  - What tasks satisfy it is not clear!
- Key challenge
  - Incentivize Challenger -> intentionally post *wrong solutions*.
- How Plasma uses TrueBit?
  - Task = sidechain application logic
  - Verification game if any user detects wrong-doing in the sidechain



Agenda

1. Overview
2. Payment channel
3. State channel
4. Plasma
5. **Discussion**

Discussion

- None of these solutions are deployed on the main-net
  - Security of state channels are well understood
  - Not so for Plasma
- Reasonable assumptions?
  - Rational parties, incentivized by money
  - Synchronous network (bounded delay on any blockchain requests)
- Plasma:
  - Not clear if any sidechain can be supported (what types of computation CAN Truebit support?)
  - Lack too much details to judge

Discussion

- So far, no experimental results
  - Small scale simulations show Lightnight networks few times better
- Cryptocurrencies-based applications only

- Good for us?
  - Yes, for Forkbase
- Possible in permissioned settings?
  - Payment channel? Yes
  - State channel? Yes (auctions)
  - Plasma? Not sure, can just run multiple instances of Hyperledger