ARBITRUM
FOUNDATION

Validating Bridges and "Rollups" as a Scaling Solution for Cryptocurrencies

Patrick McCorry

Who has coins deposited in an exchange or online service?

Bitstamp

coinbase

Custodial and private database

Custodial Sequencer

Offers super user experience, low fees and no need to learn anything about crypto!

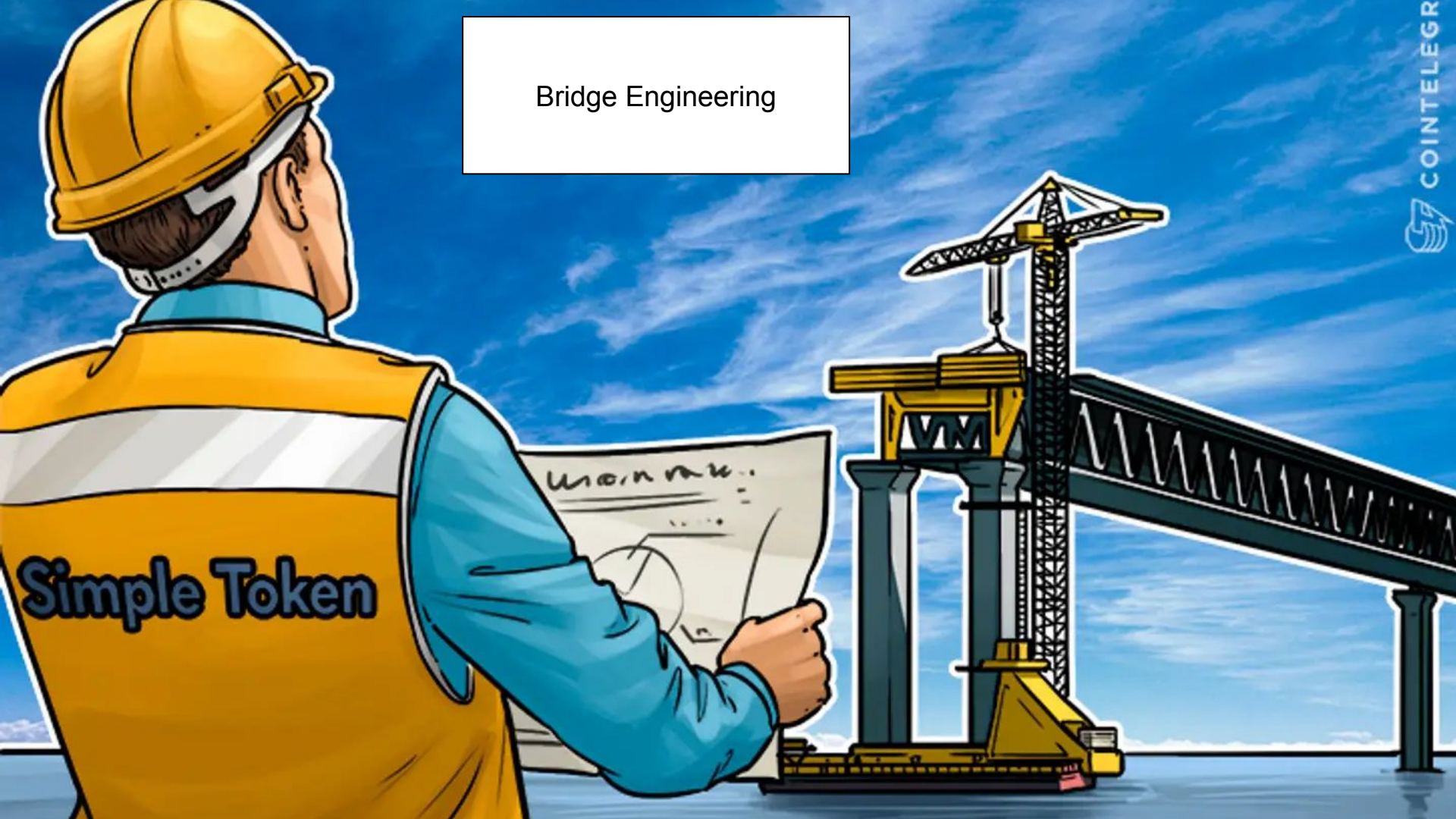
Full custody.

They can freeze/confiscate/lose our funds.

Not auditable.

The layer-2 database is opaque and not publicly auditable. Are they running a fractional reserve? Who knows.

What is at the heart of this
type of interaction with Coinbase et al?


Bridge Engineering
Simple Token
COINTELEGR

A bridge from Ethereum to FTX

Ethereum & Users
Blockchain network

Bridge contract
Holds user funds

Single authority
One ring to rule them all

A bridge from Ethereum to FTX

Ethereum & Users
Blockchain network

Bridge contract
Holds user funds

Single authority
One ring to rule them all

```
Deposit()

Withdraw()

Allowance(user, coins)
```

A bridge from Ethereum to FTX

Ethereum & Users
Blockchain network

Bridge contract
Holds user funds

FTX

Single authority
One ring to rule them all

`Deposit()`

`Withdraw()`

`Allowance(user, coins)`

Alice can withdraw 1,000 ETH

A bridge from Ethereum to FTX

Ethereum & Users
Blockchain network

Bridge contract
Holds user funds

`Deposit()`

`Withdraw()`

`Allowance(user, coins)`

FTX has informed me that Alice can withdraw 1,000 ETH.

I trust FTX - the database must be OK

A bridge from Ethereum to FTX

Ethereum & Users
Blockchain network

Bridge contract
Holds user funds

Single authority
One ring to rule them all

Alice

`Deposit()`

`Withdraw()`

`Allowance(user, coins)`

Alice withdraws 1,000 ETH

A bridge from Ethereum to FTX

Ethereum & Users
Blockchain network

Generically, what is happening?

Single authority
One ring to rule them all

Alice

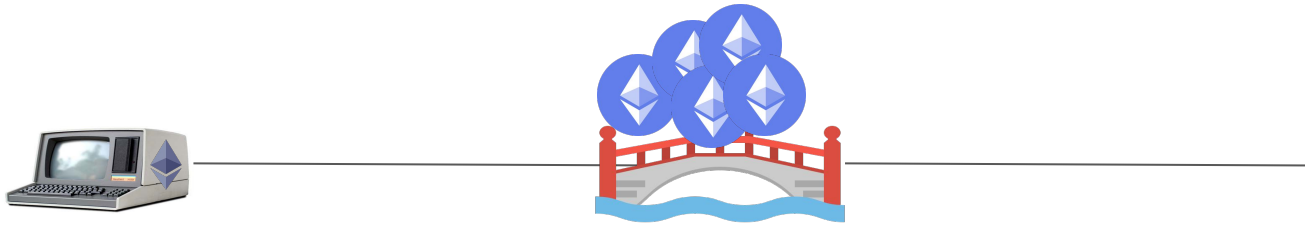
`Deposit()`

`Withdraw()`

`Allowance(user, coins)`

Alice withdraws 1,000 ETH

A bridge from Ethereum to an off-chain system



Ethereum & Users
Blockchain network

Bridge contract
Holds user funds

Off-chain database
(account balances, program state, etc).

Assets ========== Liabilities

Trust assumption

Before processing a withdrawal, I need to check the database is OK

Trust assumption for bridges have evolved over time

Thanks to Hasu for the terminology

Single authority

One ring to rule them all

coinbase Bitstamp

FTX

Thanks to Hasu for the terminology

Single authority

One ring to rule them all

coinbase Bitstamp

FTX

Multi-authority

K of N parties

RSK

Liquid by Blockstream

Thanks to Hasu for the terminology

Single authority

One ring to rule them all

coinbase Bitstamp

FTX

Multi-authority

K of N parties

RSK

Liquid by Blockstream

Crypto-economic bridge

Staked investment in its success

polygon

Previously Matic Network

Thanks to Hasu for the terminology

Polygon's proof of stake bridge:

Binance	506,183,677
Stakin	322,033,445
All nodes	206,676,574
Web3Nodes	123,762,284
Anonymous 94	100,622,650
Decentral Games	70,018,093
Total	**1,329,296,723**

Attack Target: 1,283,657,130 matic

9/12/2021

Single authority

One ring to rule them all

coinbase Bitstamp

FTX

Multi-authority

K of N parties

RSK

Liquid by Blockstream

Crypto-economic bridge

Staked investment in its success

polygon

Previously Matic Network

Thanks to Hasu for the terminology

Single authority
One ring to rule them all

coinbase Bitstamp

FTX

Trusting <10 parties to protect our funds

… sucks a bit right?

Multi-authority
K of N parties

RSK

Liquid by Blockstream

Crypto-economic bridge
Staked investment in its success

polygon
Previously Matic Network

Human processes to protect a set of keys

A very difficult to replicate process

Taylor Monahan maintains a larger list

https://docs.google.com/spreadsheets/d/1ZEEAmXjpN8kL9BvITg9GKu-dbeUra6c14YLpLkCp5Zo/edit?usp=sharing

Name	Tokens	Comment
MtGox (2014)	850k BTC	6% of all bitcoin
Bitcoinica (2011)	61k BTC	Linode hosting provider hacked
Bitfloor (2012)	24k BTC	Wallets stored on server
Bitstamp (2015)	19k BTC	Hot wallet hacked
BTER (2015)	7k BTC	Inside job
Gatecoin (2015)	185k ETH	Hot wallet hacked
Bitfinex (2016)	119k BTC	Compromised server
Bithumb (2018)	2k BTC	Hot wallet hacked
Zaif (2018)	6k BTC	Hot wallet hacked
Coincheck (2018)	$534m NEM tokens	Hot wallet hacked
Coinbin (2019)	$26m in tokens	Inside job
CoinBene (2019)	$45m in tokens	Hot wallet hacked
Binance (2019)	7k BTC	Hot wallet hacked

FTX

QUADRIGACX

Embezzlement and fraud

Human protected bridges are no different to TradFi or legacy banking except that central banks provide deposit guarantees for customers.

They have a insurer of last resort **because they cannot be trusted.**

In crypto, there is no insurer of last resort.

Satoshi Nakamoto Quotes

Trust Engineering

The root problem with conventional currency is all the trust that's required to make it work…

"Barely a fraction in reserve"

Banks must be trusted to hold our money and transfer it electronically, but they lend it out in waves of credit bubbles with barely a fraction in reserve.

Cryptography > blind trust

*What is needed is an electronic payment system based **on cryptographic proof instead** of trust, allowing any two willing parties to transact directly with each other without the need for a trusted third party.*

Old school motto

Can we transact on a off-chain system, while still allowing users to maintain self-custody of their funds?

Enabling Blockchain Innovations with Pegged Sidechains

Adam Back, Matt Corallo, Luke Dashjr,
Mark Friedenbach, Gregory Maxwell,
Andrew Miller, Andrew Poelstra,
Jorge Timón, and Pieter Wuille*†

2014-10-22 (commit 5620e43)

Abstract

Since the introduction of Bitcoin[Nak09] in 2009, and the multiple computer science and electronic cash innovations it brought, there has been great interest in the potential of decentralised cryptocurrencies. At the same time, implementation changes to the consensus-critical parts of Bitcoin must necessarily be handled very conservatively. As a result, Bitcoin has greater difficulty than other Internet protocols in adapting to new demands and accommodating new innovation.

We propose a new technology, *pegged sidechains*, which enables bitcoins and other ledger assets to be transferred between multiple blockchains. This gives users access to new and innovative cryptocurrency systems using the assets they already own. By reusing Bitcoin's currency, these systems can more easily interoperate with each other and with Bitcoin, avoiding the liquidity shortages and market fluctuations associated with new currencies. Since sidechains are separate systems, technical and economic innovation is not hindered. Despite bidirectional transferability between Bitcoin and pegged sidechains, they are isolated: in the case of a cryptographic break (or malicious design) in a sidechain, the damage is entirely confined to the sidechain itself.

This paper lays out pegged sidechains, their implementation requirements, and the work needed to fully benefit from the future of interconnected blockchains.

At the heart of the original sidechain paper was a protocol to build ***a trustless bridge.***

… but if the "other blockchain" went offline, for whatever reason, then the funds got stuck!

Can we really build a bridge that protects us from an all powerful authority – even they go offline?

Lightning strikes create plasma via a very strong jolt of electricity. Most of the Sun, and other stars, is in a plasma state. Certain regions of Earth's atmosphere contain some plasma created primarily by ultraviolet radiation from the Sun. Collectively, these regions are called the ionosphere.

https://scied.ucar.edu › learning-zone › sun-space-weather

Plasma - UCAR Center for Science Education

It all began with Plasma

Plasma: Scalable Autonomous Smart Contracts

Joseph Poon joseph@lightning.network
Vitalik Buterin vitalik@ethereum.org

August 11, 2017
WORKING DRAFT
https://plasma.io/

Abstract

Plasma is a proposed framework for incentivized and enforced execution of smart contracts which is scalable to a significant amount of state updates per second (potentially billions) enabling the blockchain to be able to represent a significant amount of decentralized financial applications worldwide. These smart contracts are incentivized to continue operation autonomously via network transaction fees, which is ultimately reliant upon the underlying blockchain (e.g. Ethereum) to enforce transactional state transitions.

We propose a method for decentralized autonomous applications to scale to process not only financial activity, but also construct economic incentives for globally persistent data services, which may produce an alternative to centralized server farms.

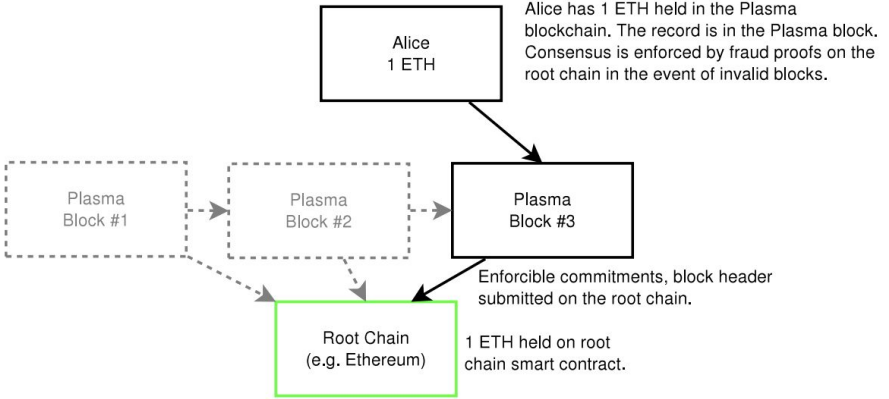
Plasma is composed of two key parts of the design: Reframing all blockchain computation into a set of MapReduce functions, and an optional method to do Proof-of-Stake token bonding on top of existing blockchains with the understanding that the Nakamoto Consensus incentives discourage block withholding.

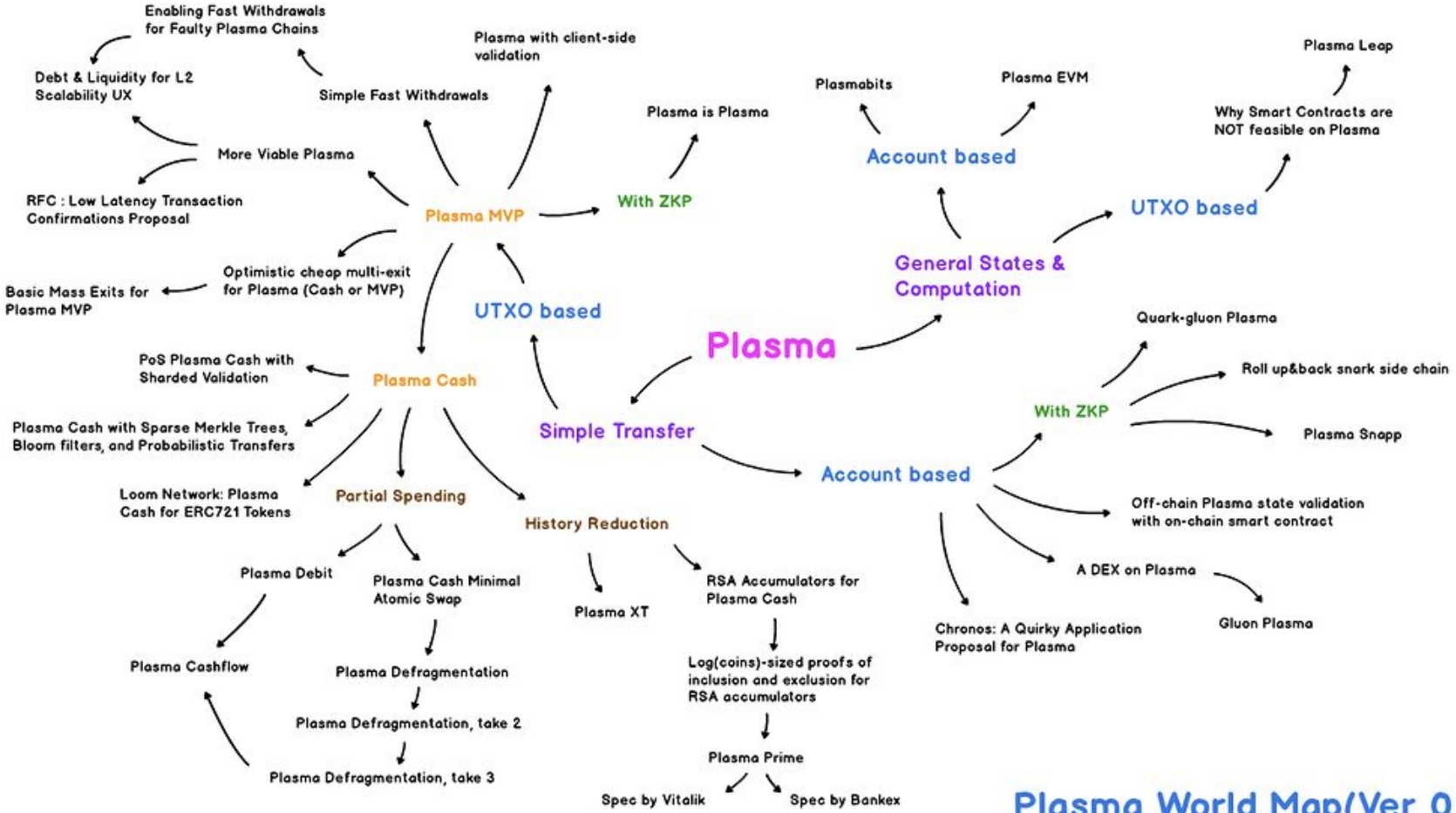
This construction is achieved by composing smart contracts on the main blockchain using fraud proofs whereby state transitions can be enforced on a parent blockchain. We compose blockchains into a tree hierarchy, and treat each as an individual branch blockchain with enforced blockchain history and MapReducible computation committed into merkle proofs. By framing one's ledger entry into a child blockchain which is enforced by the parent chain, one can enable incredible scale with minimized trust (presuming root blockchain availability and correctness).

The greatest complexity around global enforcement of non-global data revolves around data availability and block withholding attacks, Plasma has mitigations for this issue by allowing for exiting faulty chains while also creating mechanisms to incentivize and enforce continued correct execution of data.

As only merkleized commitments are broadcast periodically to the root blockchain (i.e. Ethereum) during non-faulty states, this can allow for incredibly scalable, low cost transactions and computation. Plasma enables persistently operating decentralized applications at high scale.

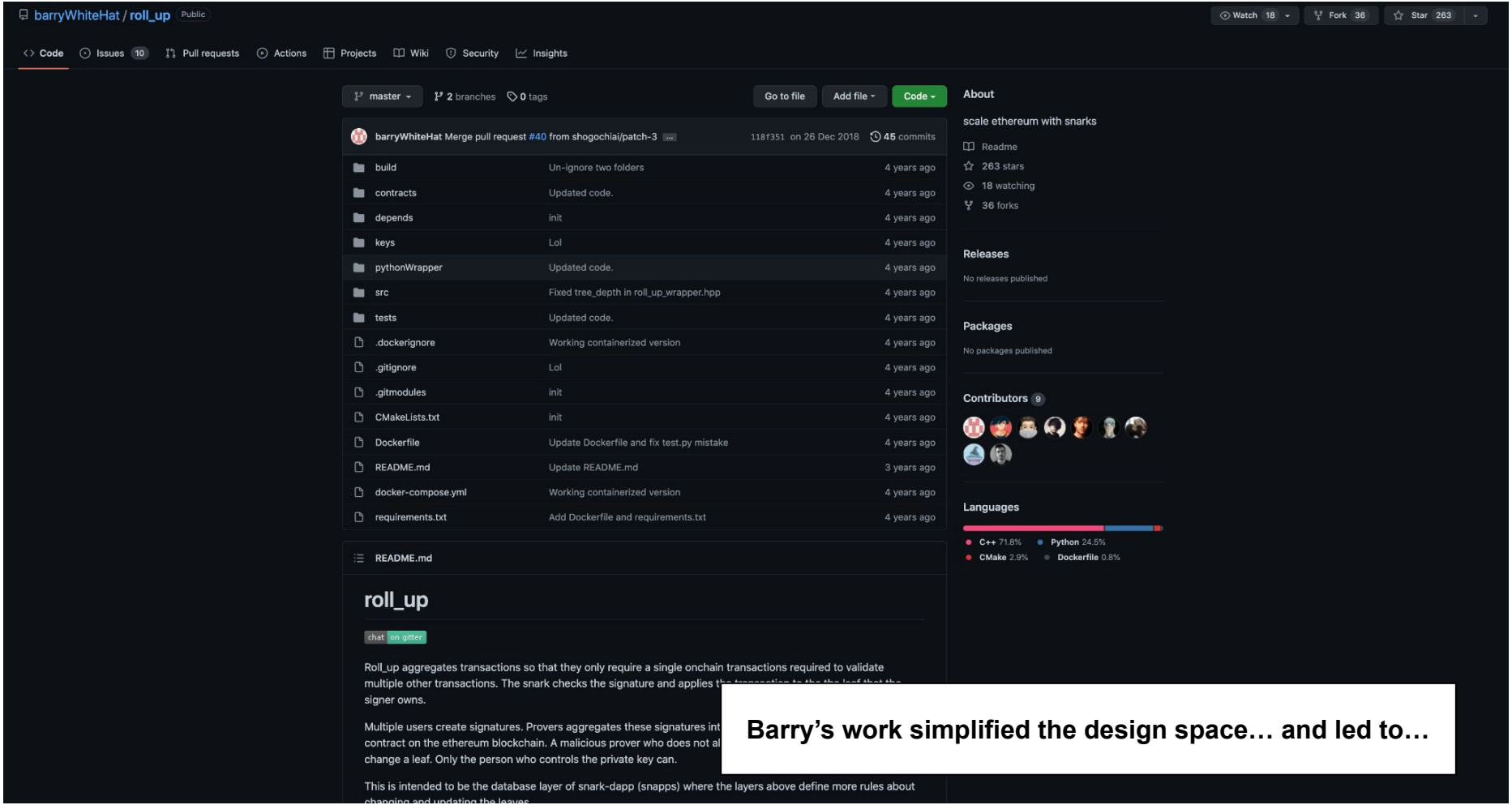
Again, an impossible paper to read





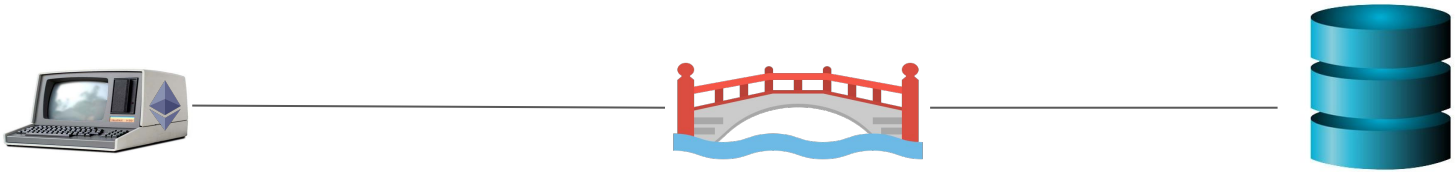
Plasma World Map(Ver 0.1)

Created by Aiden (aiden.p@ontther.io)



Barry's work simplified the design space… and led to…

The Validating Bridge (rollups)



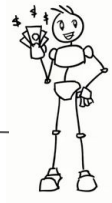
Deposit()

Withdraw()

Update(<.....>, <.....>)

The Validating Bridge (rollups)

update db



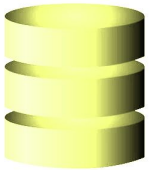
Executor

Deposit()

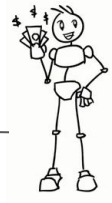
Withdraw()

Update(<.....>, <.....>)

The Validating Bridge (rollups)



update db



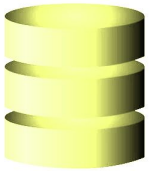
Executor

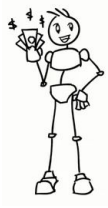
Deposit()

Withdraw()

Update(<.....>, <.....>)

The Validating Bridge (rollups)





Executor

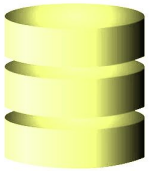
Deposit()

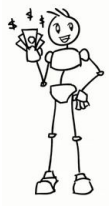
Withdraw()

Update(<.....>, <.....>)

Proposed update to db

The Validating Bridge (rollups)





Executor

Deposit()

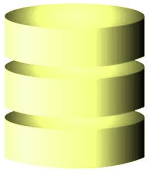
With

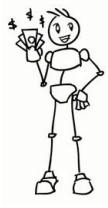
Upd

~~OK i'll accept this update~~

You must CONVINCE me why this update to the database it is correct.

The Validating Bridge (rollups)





Deposit()

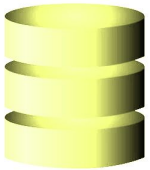
Withdraw()

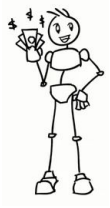
Update(proposed_update, evidence)

Convincing evidence

Executor

The Validating Bridge (rollups)





Executor

Deposit()

With

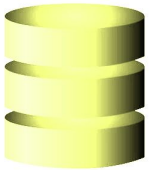
Upd

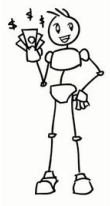
Database is safe and alive?

validates

I'll accept it.

The Validating Bridge (rollups)





Executor

Deposit()

Withdraw()

Update(proposed_update, evidence)

LOOPRING

Optimistic Ethereum

ARBITRUM

zkSync

STARKWARE

Continuously check the integrity of all proposed updates

The Validating Bridge (rollups)

Ultimately, the layer-1 blockchain, Ethereum, is protecting you.

Executor

LOOPR

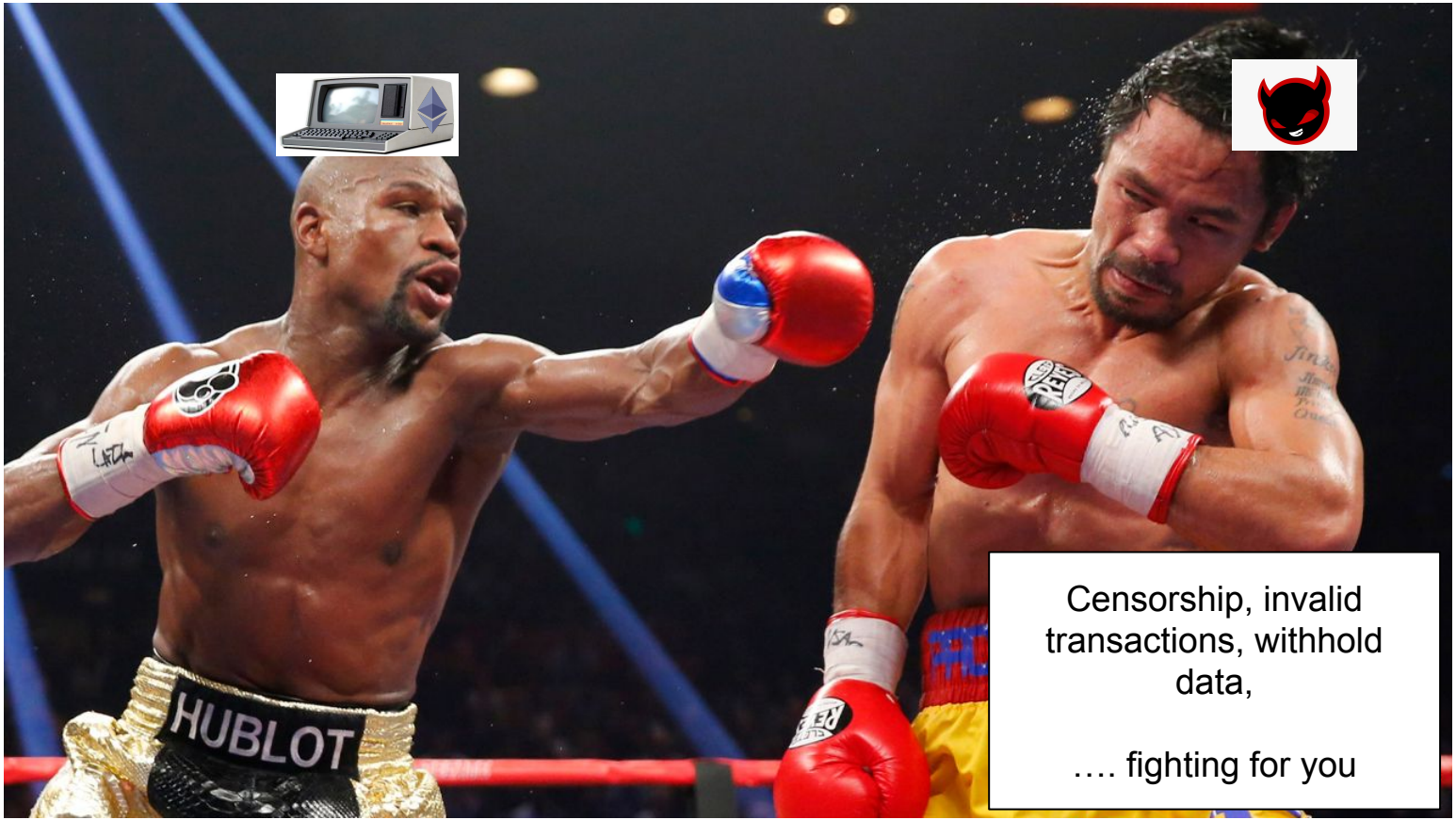
OP Optimistic Ether

ARBITRUM

zkSync

STARKWARE

Continuously check the integrity of all proposed updates


HUBLOT
Censorship, invalid transactions, withhold data,
…. fighting for you

Sounds so cool….

… but how do validating bridges work?

Let’s try to define the environment

- Agents
  - Who are the players?
- Overview of a validating bridge
  - How does it work at a high level?
- Threat Model and Security properties
  - Who is our adversary? And what special powers do they have?
  - What are we trying to secure?

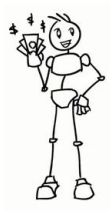
Agents

User

Likes mooncats

Sequencer

Orders transactions



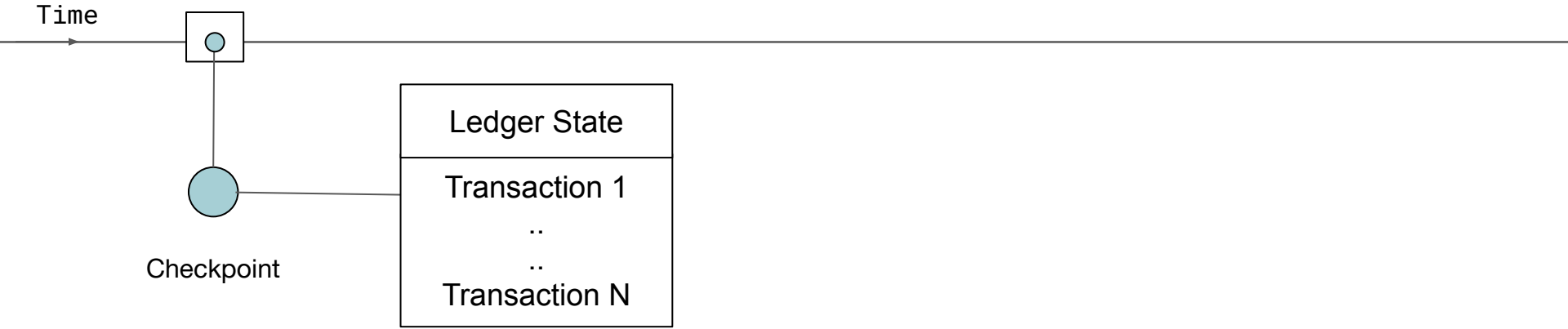
Executor

Executes transactions

Collect transactions for ordering

Sequencer

Optimistic transaction
ordering service



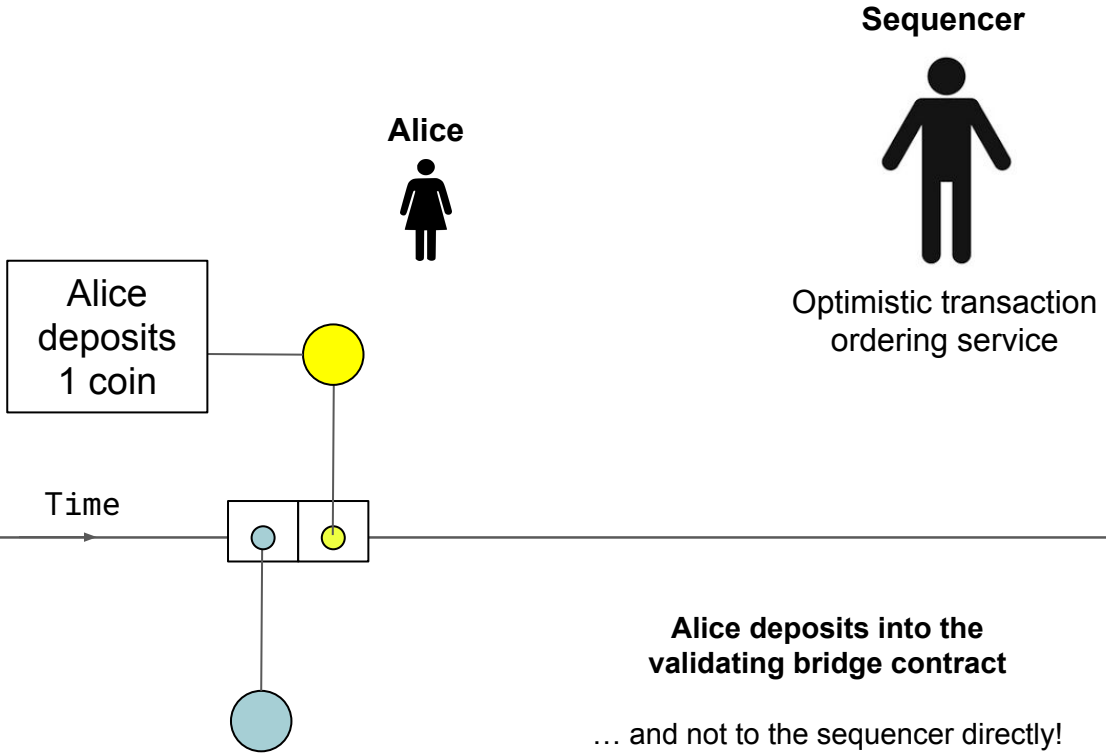
Collect transactions for ordering

Alice

Sequencer

Optimistic transaction ordering service

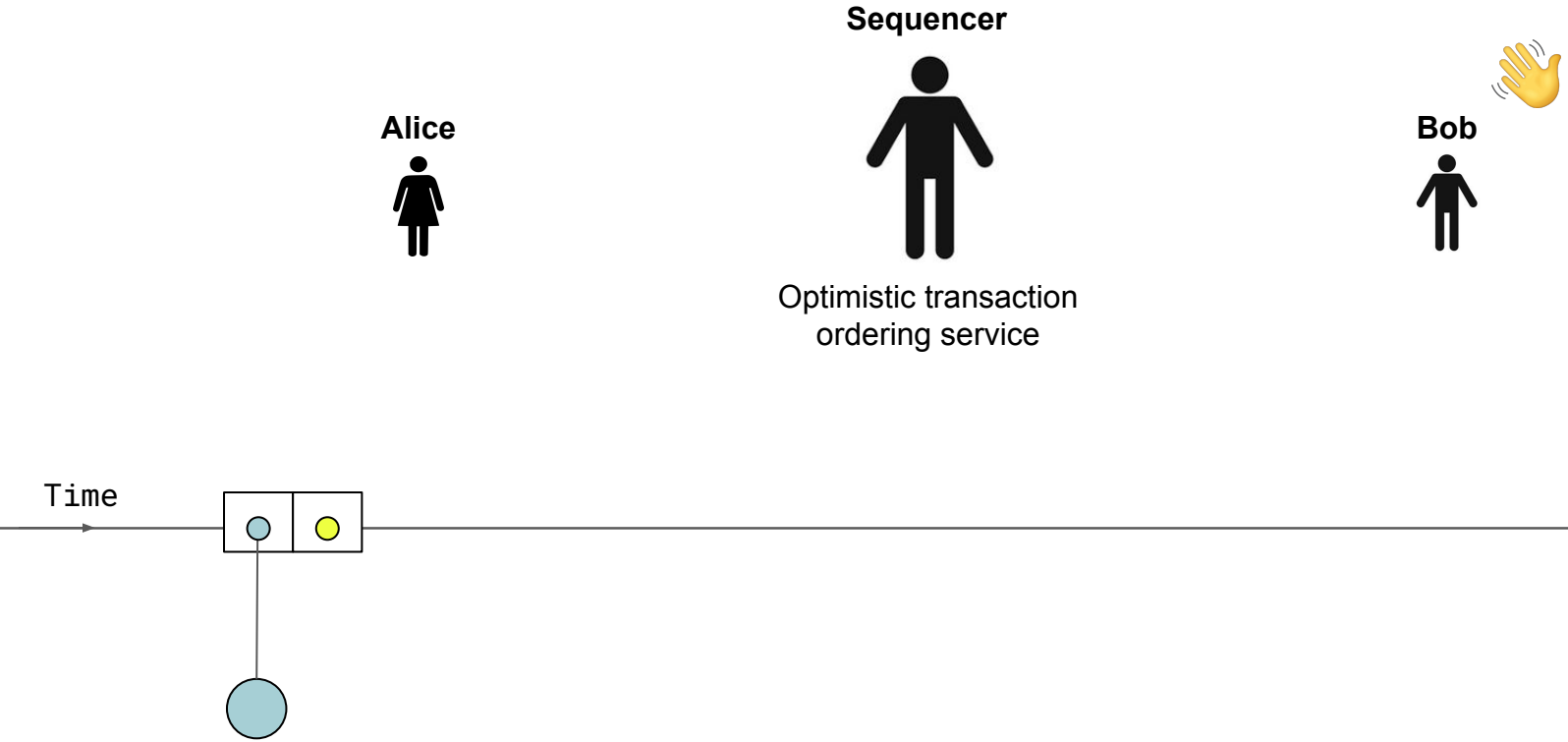
Collect transactions for ordering



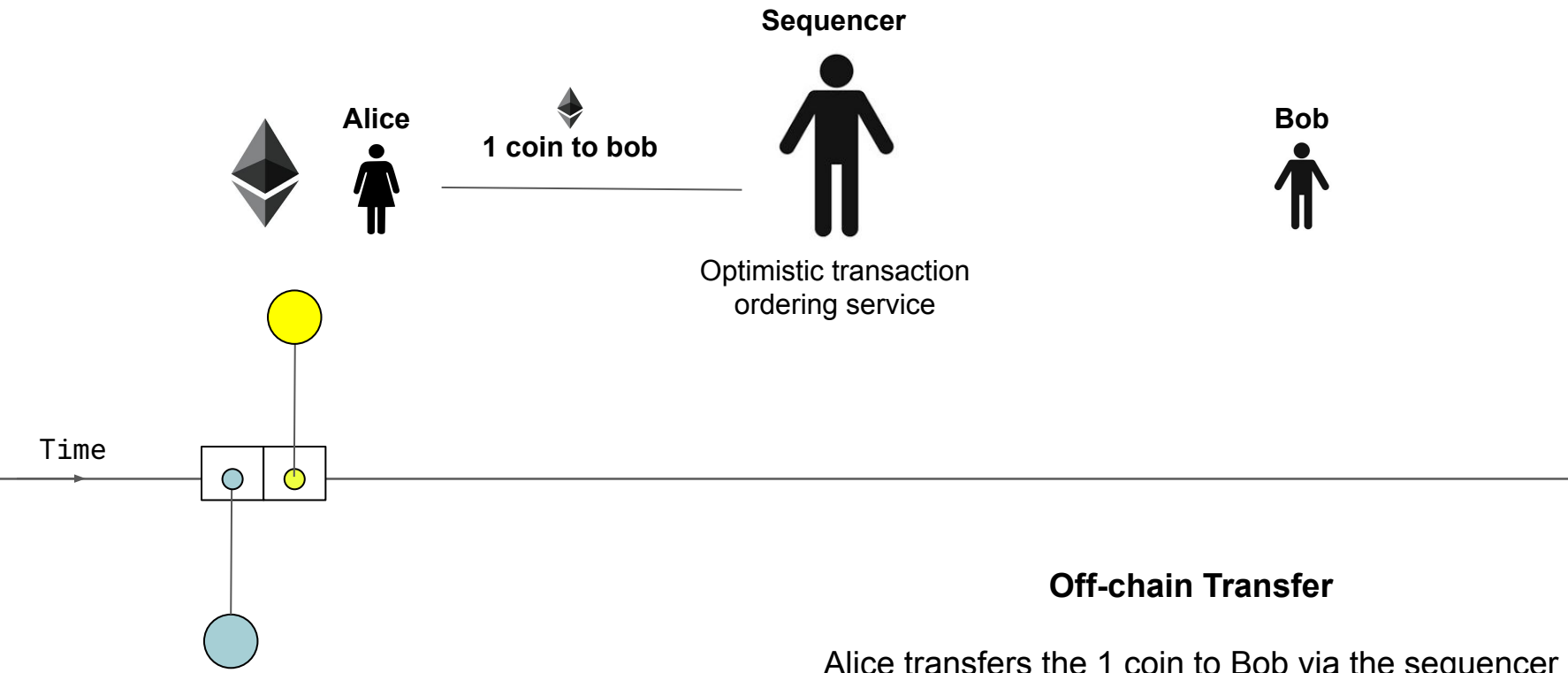
Alice deposits into the validating bridge contract

… and not to the sequencer directly!

Collect transactions for ordering



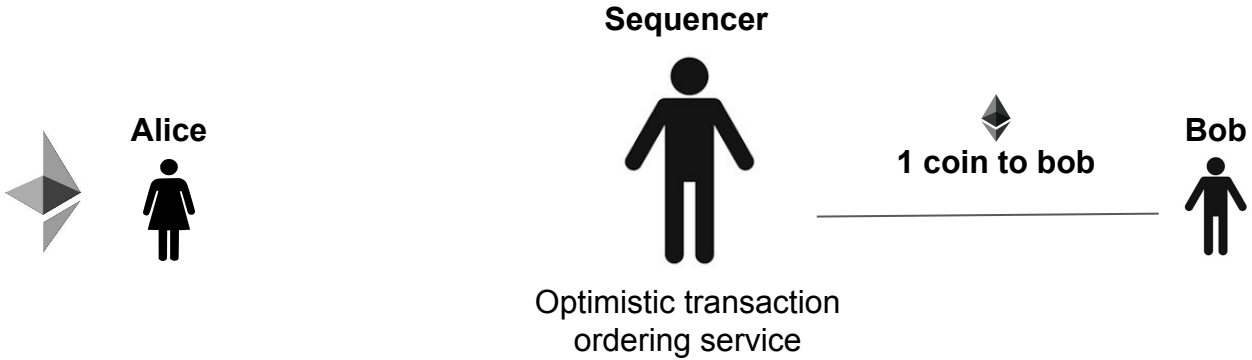
Collect transactions for ordering



Off-chain Transfer

Alice transfers the 1 coin to Bob via the sequencer

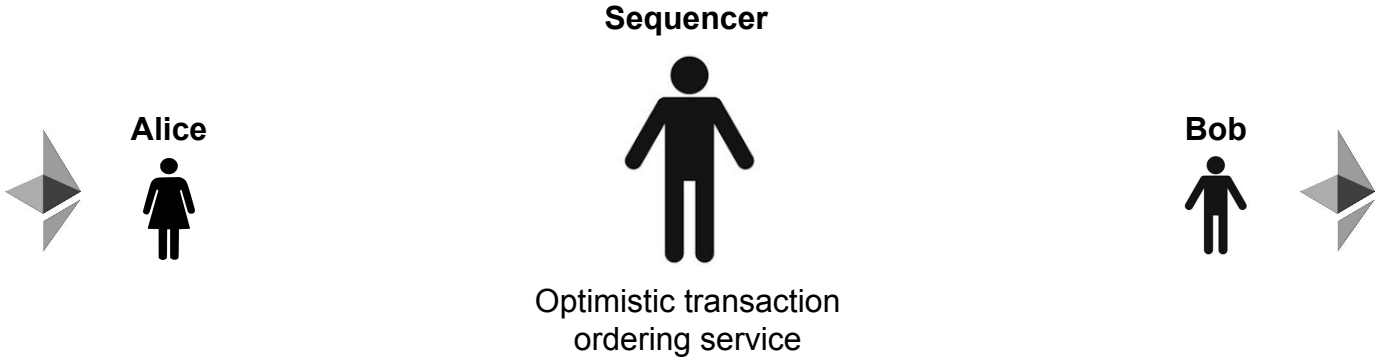
Collect transactions for ordering



Off-chain Transfer

Alice transfers the 1 coin to Bob via the sequencer

Collect transactions for ordering



Sequencer waits around….
For more off-chain transfers…

Alice to Bob transfer is “pending” and not yet confirmed.

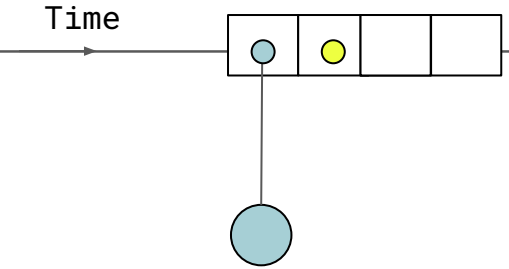
Collect transactions for ordering

Sequencer

Alice

Bob

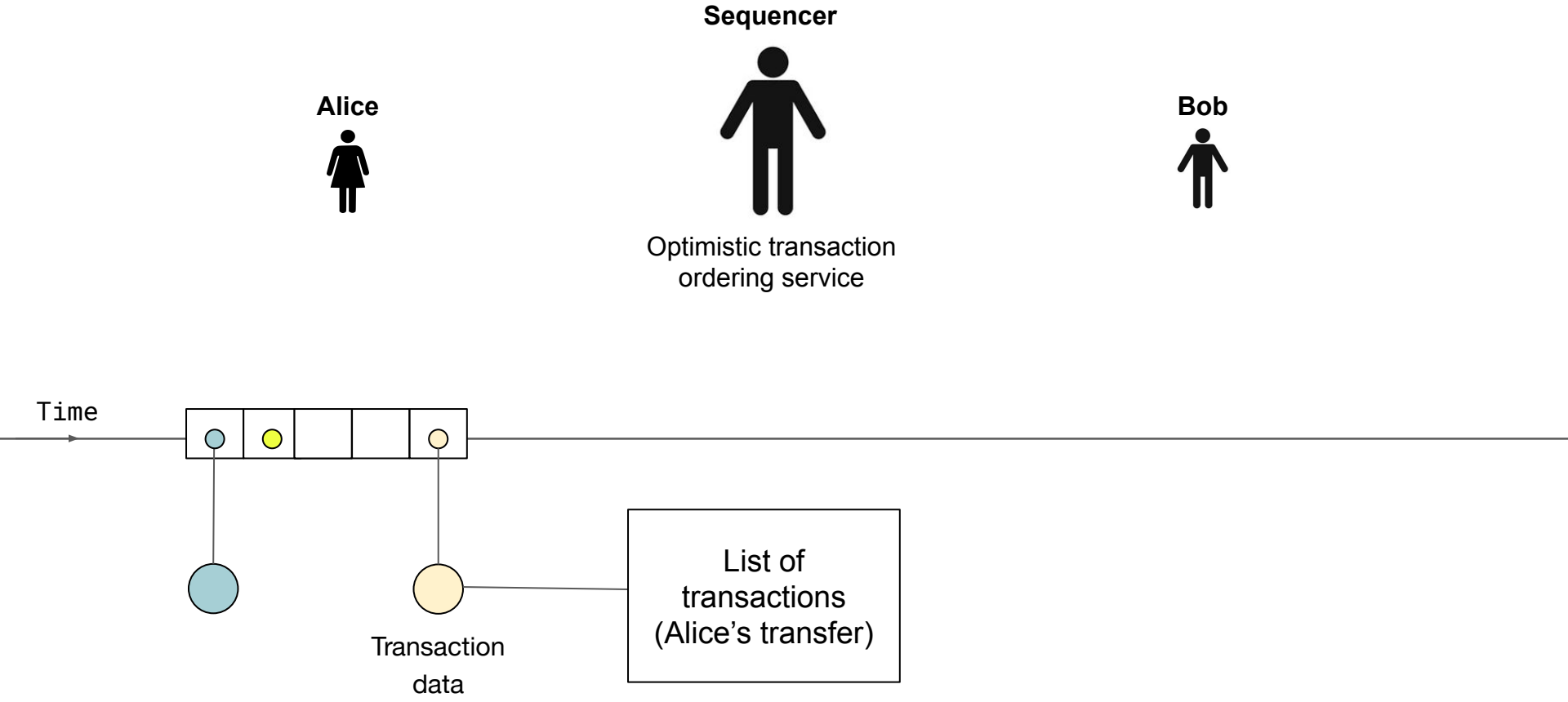
Optimistic transaction ordering service



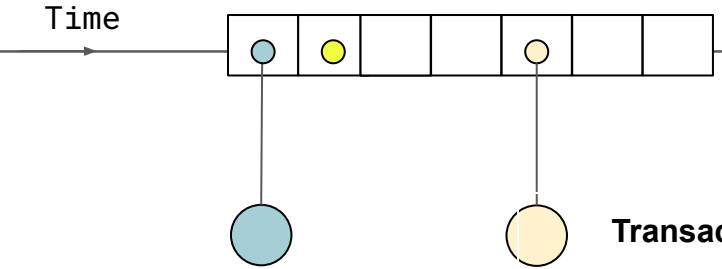
Sequencer waits around….
For more off-chain transfers…

Alice to Bob transfer is pending and not yet confirmed.

Bridge contract orders the pending transactions

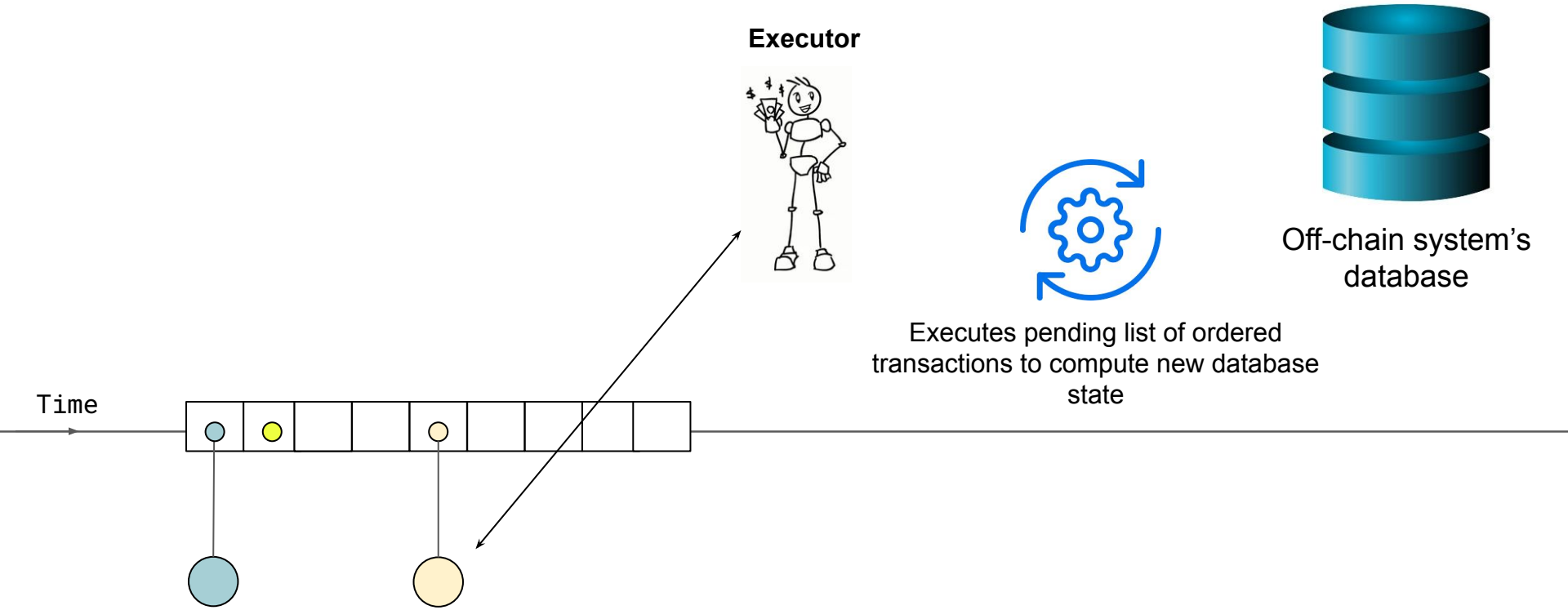


Bridge contract orders the pending transactions



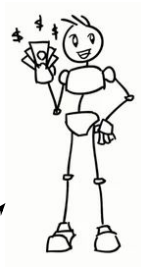
Transactions are “ordered”, but not executed.

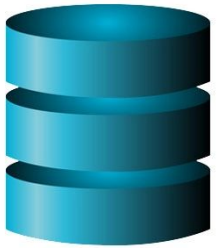
Convince a validating bridge of final execution



Convince a validating bridge of final execution

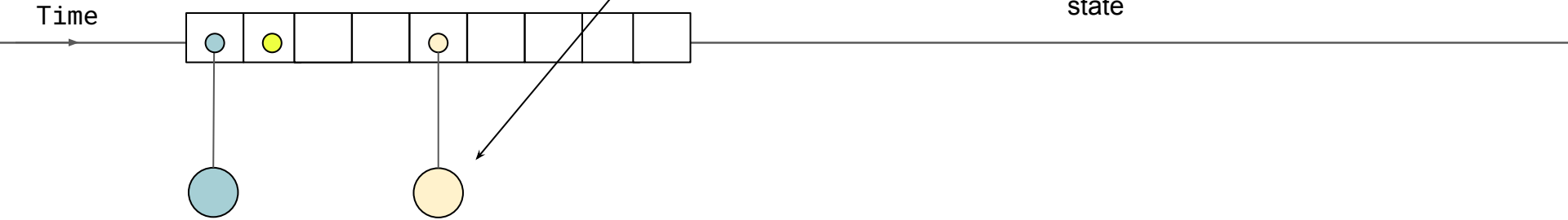
Executor



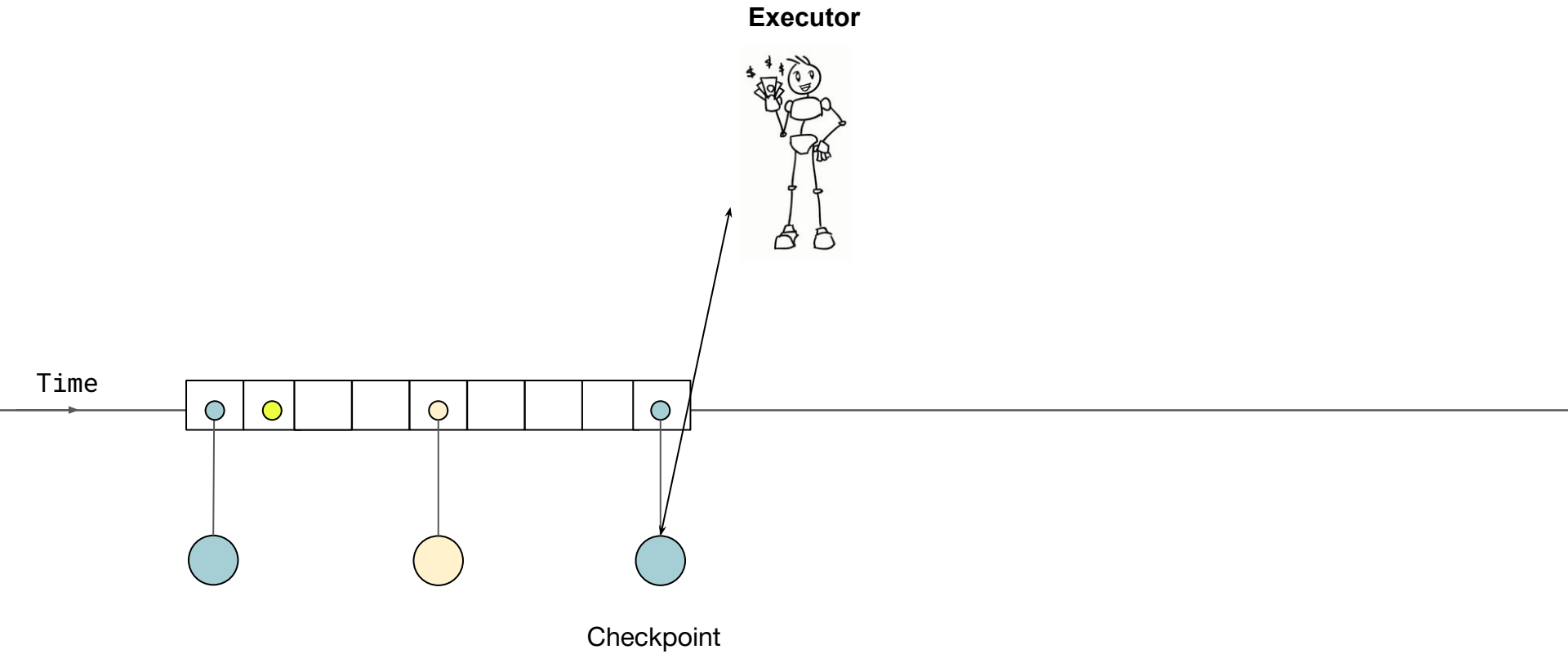


Off-chain system's database

Executes pending list of ordered transactions to compute new database state



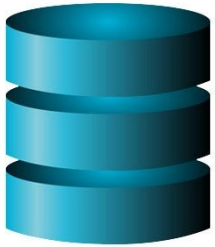
Convince a validating bridge of final execution



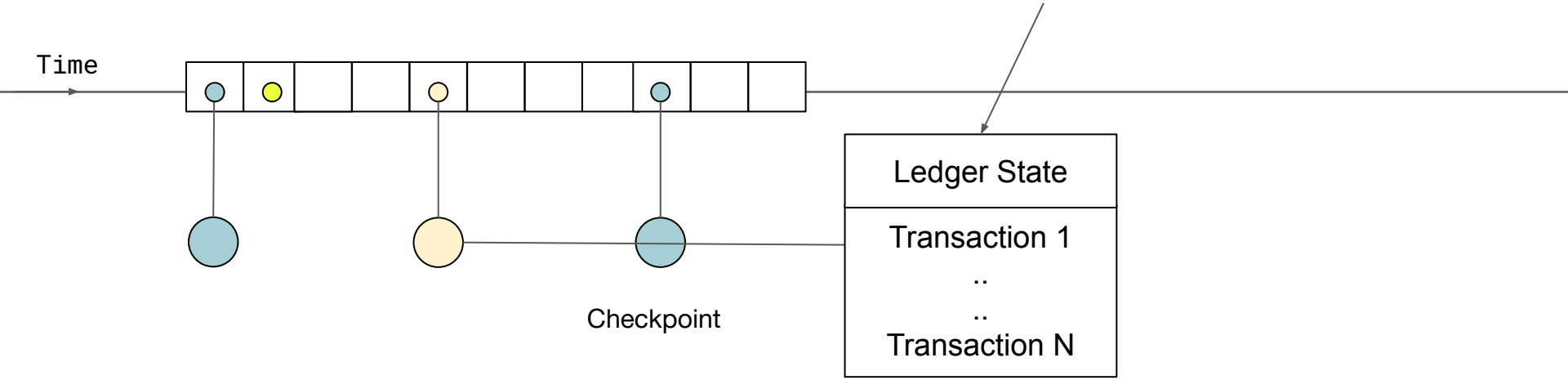
Continuously convince a validating bridge

It is a continuous process that never really ends

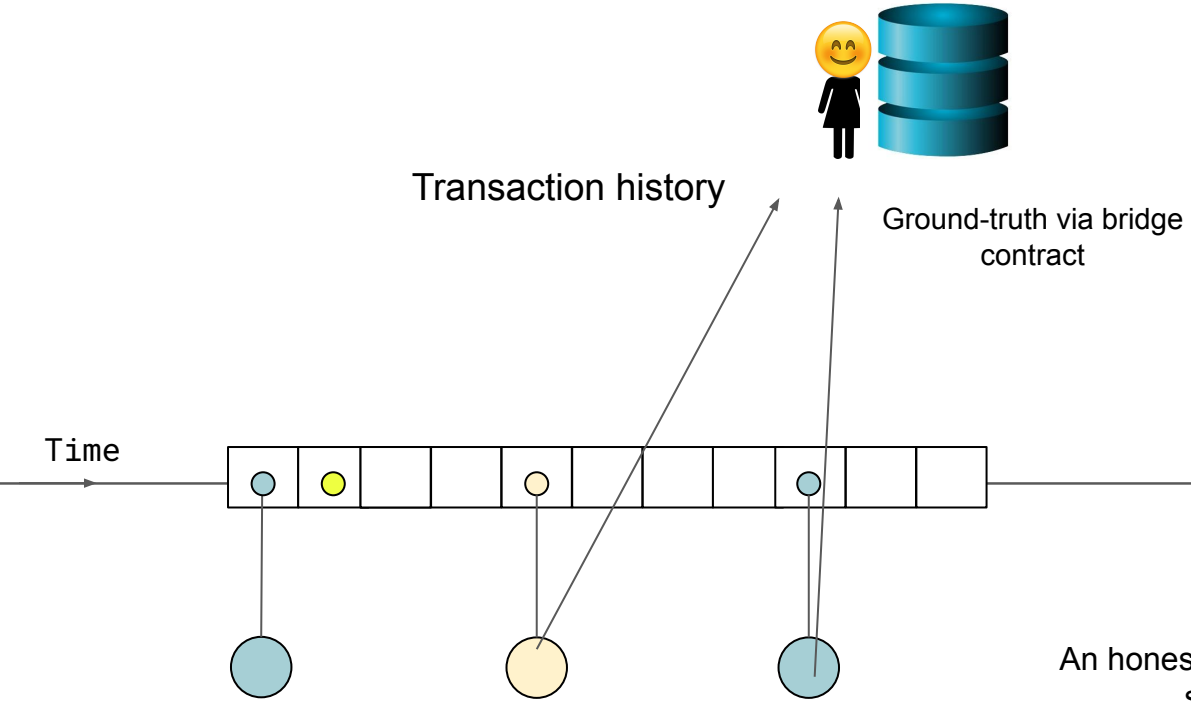
- **Checkpoint** asserts a new update to the database.
- **Execution** dictates the correctness of the update.



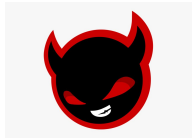
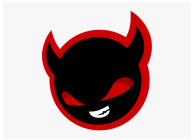
Off-chain system's database

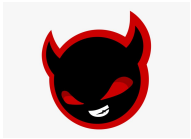
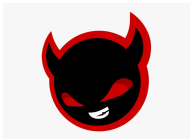
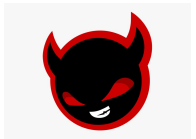
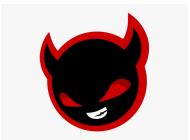


Proof of reserves and fully auditable by default



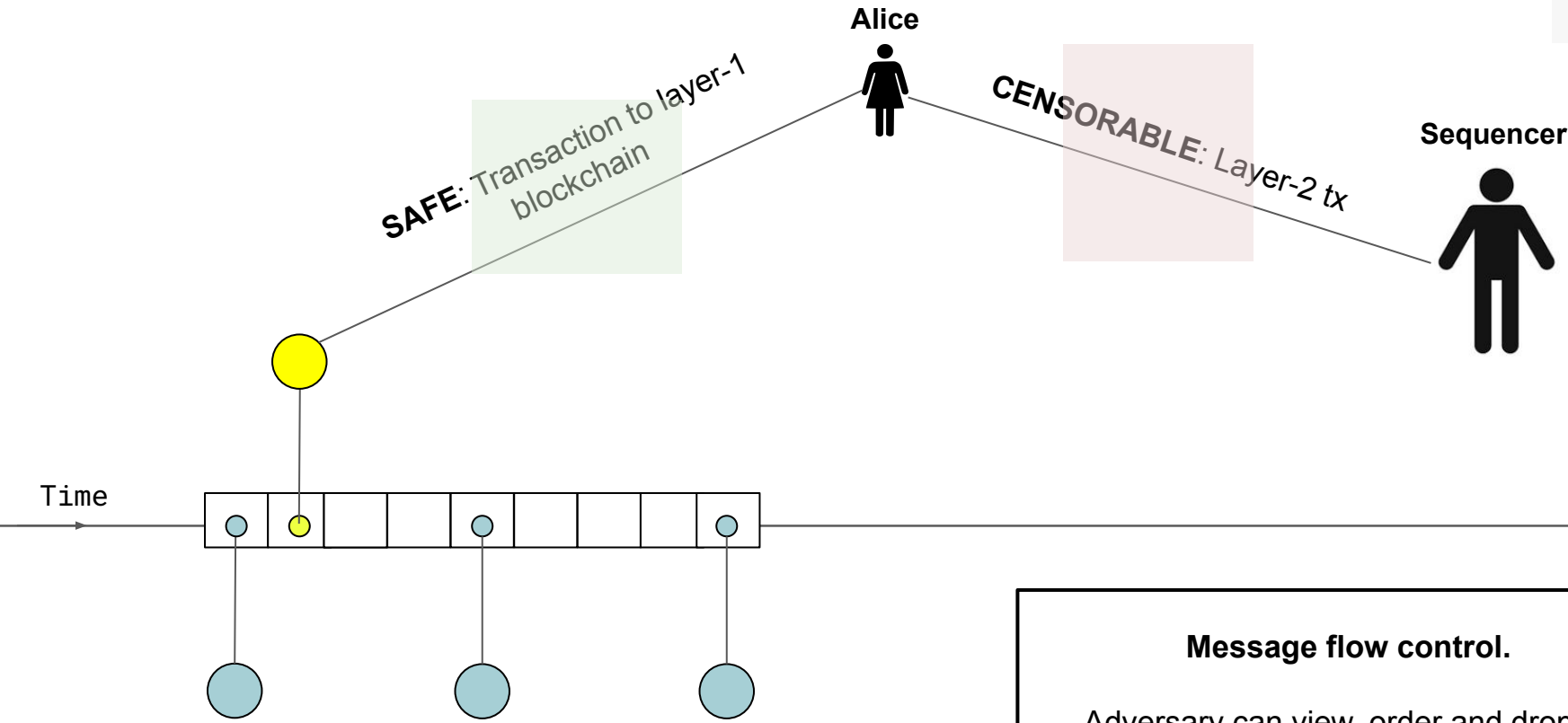
An honest party (Alice) can recompute the off-chain system's database independently.

Adversarial threat model

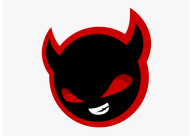
Adversarial Model

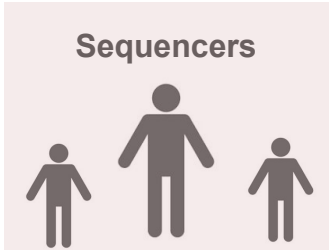


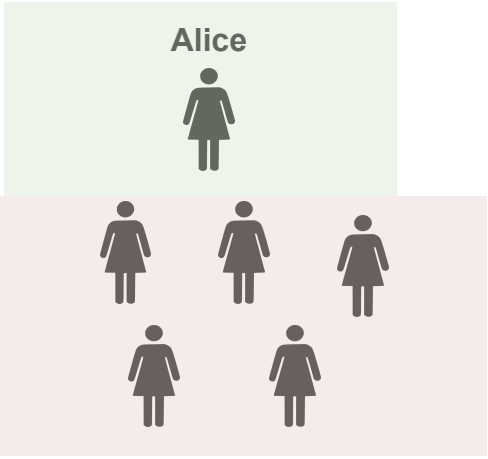
Message flow control.

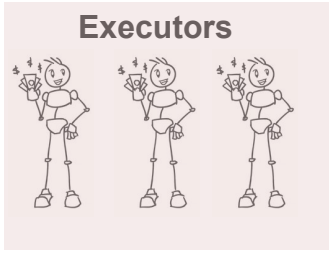
Adversary can view, order and drop all messages except for transactions sent to the layer-1 blockchain.

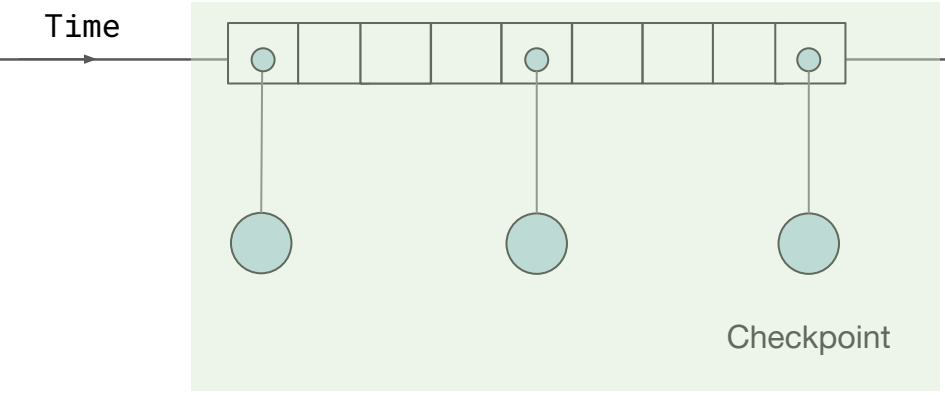
Adversarial Model











Corrupt nearly all parties

An honest user, optionally a challenger, and the blockchain (smart contract) vs everyone else.

Threat model (power of adversary)

- **Message flow control.** Control the order (and drop) all messages at will except for messages destined for the parent blockchain (eventual delivery protocol assumption).

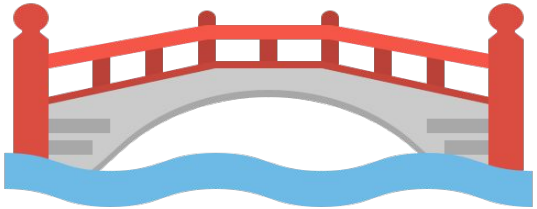
- **Corrupt nearly all parties.** Adversary can corrupt all sequencers and N-1 users. They cannot corrupt one honest user and the parent blockchain.

- **Financially motivated (optional):** Adversary may require to place a security bond in the parent blockchain that is slashed if fraudulent behaviour is detected.

- **Cannot break cryptography:** Weak against hashes, signatures, SNARKS

We have described the most POWERFUL adversary and **some rollups lack the tools to fully constrain or out-right defeat it.**

Goal: Protecting the safety & liveness of the off-chain database.



What the validating bridge checks

Data availability

- Are all state updates to the database publicly available?

State transition integrity

- Are all state updates to the database valid and well-formed?

Censorship-resistance

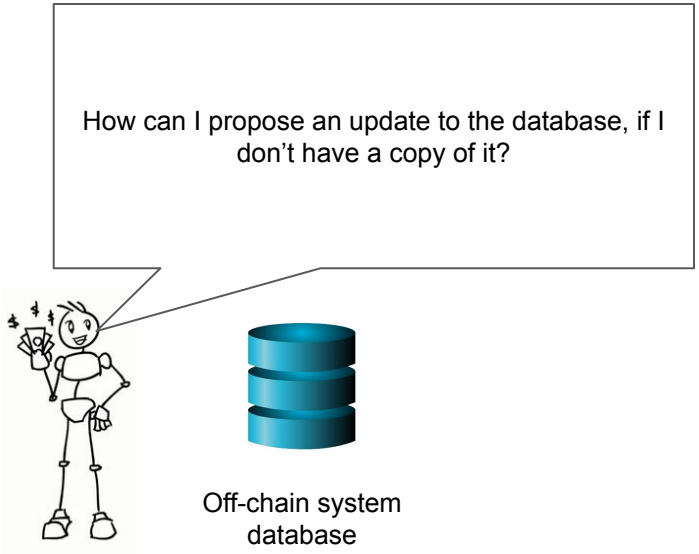
- Can the user self-enforce that a transaction will eventually execute?

The Data Availability Problem

Data Availability Problem

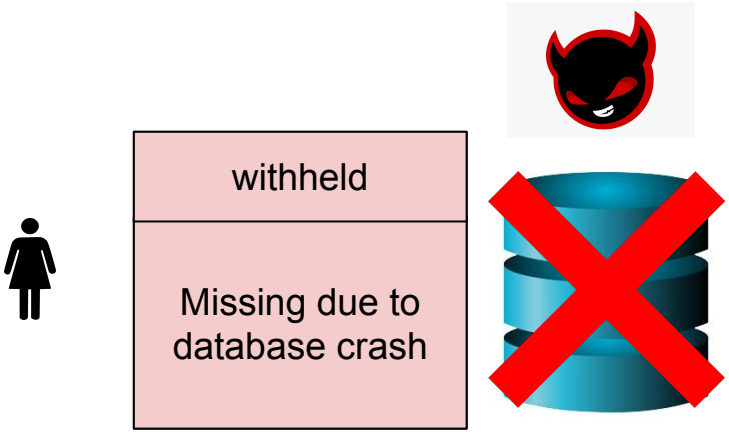
- Why does the data need to be publicly available?
- What data needs to be publicly available?
- How do we guarantee it is publicly available?

Why does the data need to be publicly available?



1 honest party (assistant) assumption

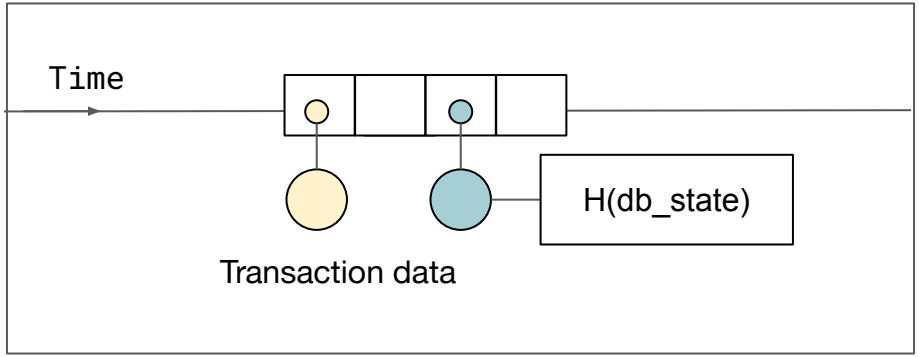
We need to assume there is one party, somewhere on the web, who will have a copy of the database and propose an update.

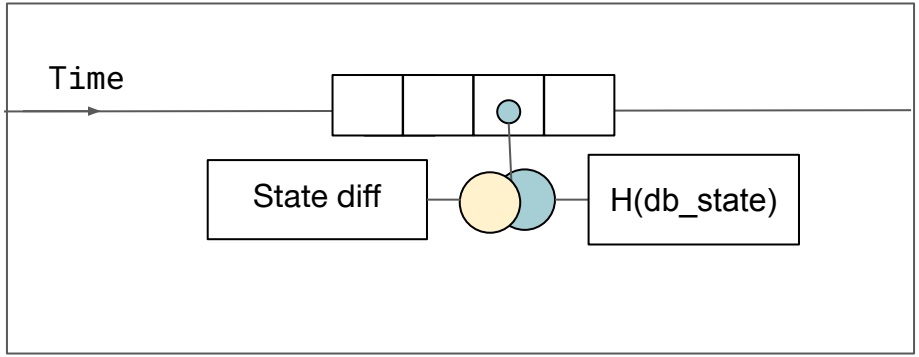


Adversary winning: Safety & Liveness issues

Adversary can freeze the system, potentially steal funds and lie about entries in the database.

What data needs to be publicly available?





Transaction history

Enforces the ordering of all transactions and its execution

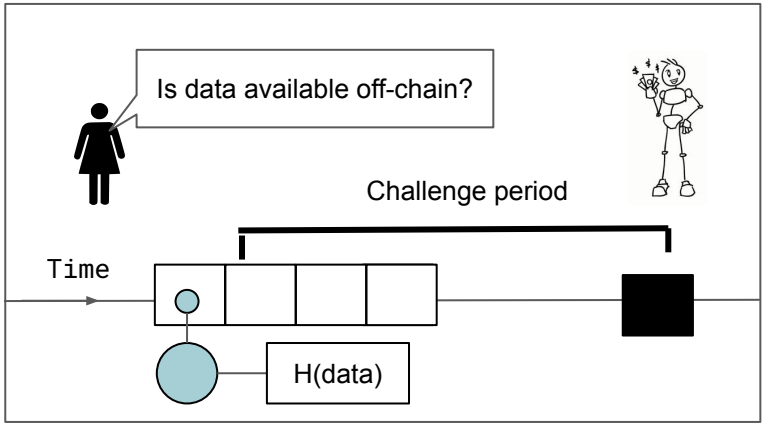
Honest party: Computes all transactions to get a copy of the database

State diffs

Bridge is not aware of individual transactions, just their aggregation

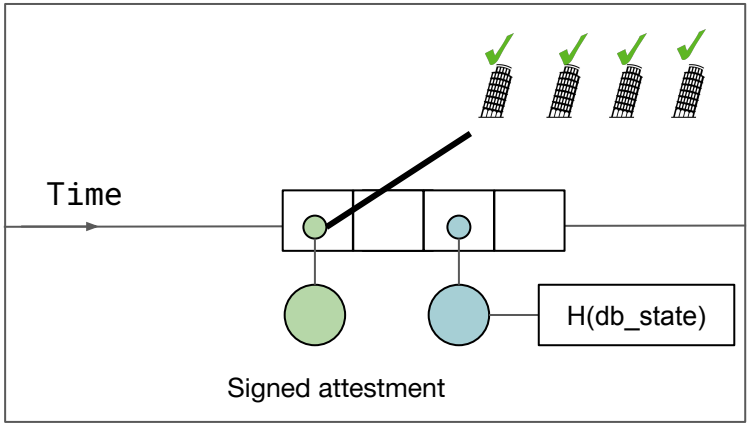
Honest party: Computes all state diffs to get a copy of the database (updates storage slots)

How do we guarantee the data is publicly available?



On-chain data availability challenge

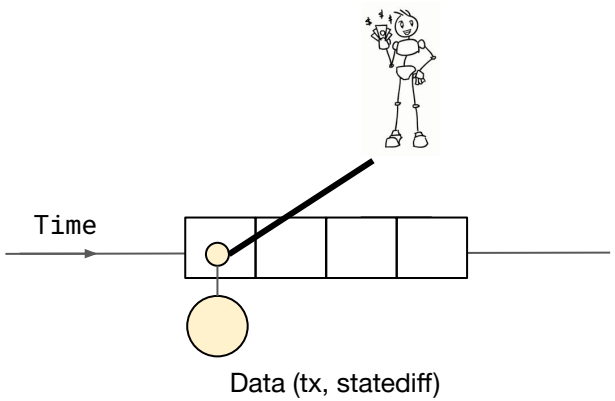
Force the operators to reveal the data via the bridge in a timely manner



Data availability committee

K of N data availability providers will sign off and attest to the fact the data is publicly available

How do we guarantee the data is publicly available?



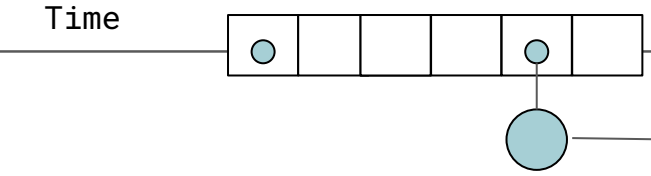
Rollup

Post all the data to the blockchain.

Bingo!

The State Transition Integrity Problem

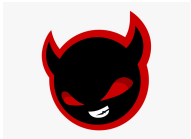
State transition integrity (protecting the layer-2 database)



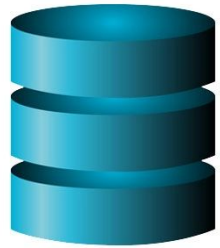
Checkpoint

Checkpoint = H(<send sequencer all coins>)

If the adversary can get **1 invalid transaction processed**, then they can **steal all funds** in the bridge



State
Transaction 1 .. Invalid Transaction .. Transaction N



Layer-2 database

State transition integrity (protecting the layer-2 database)

Time

Bingo!

Optimistic vs ZK
Fault Proofs vs Validity proofs

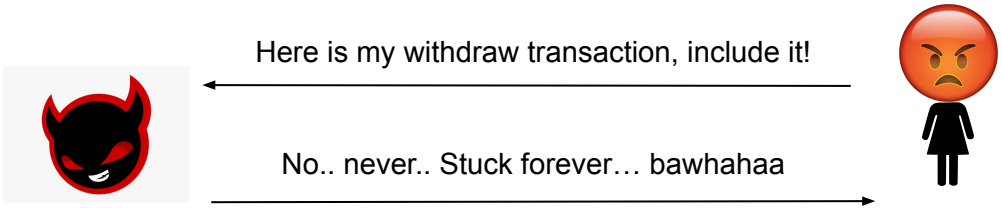
… but we can go deep into this another day :)

Layer-2 database

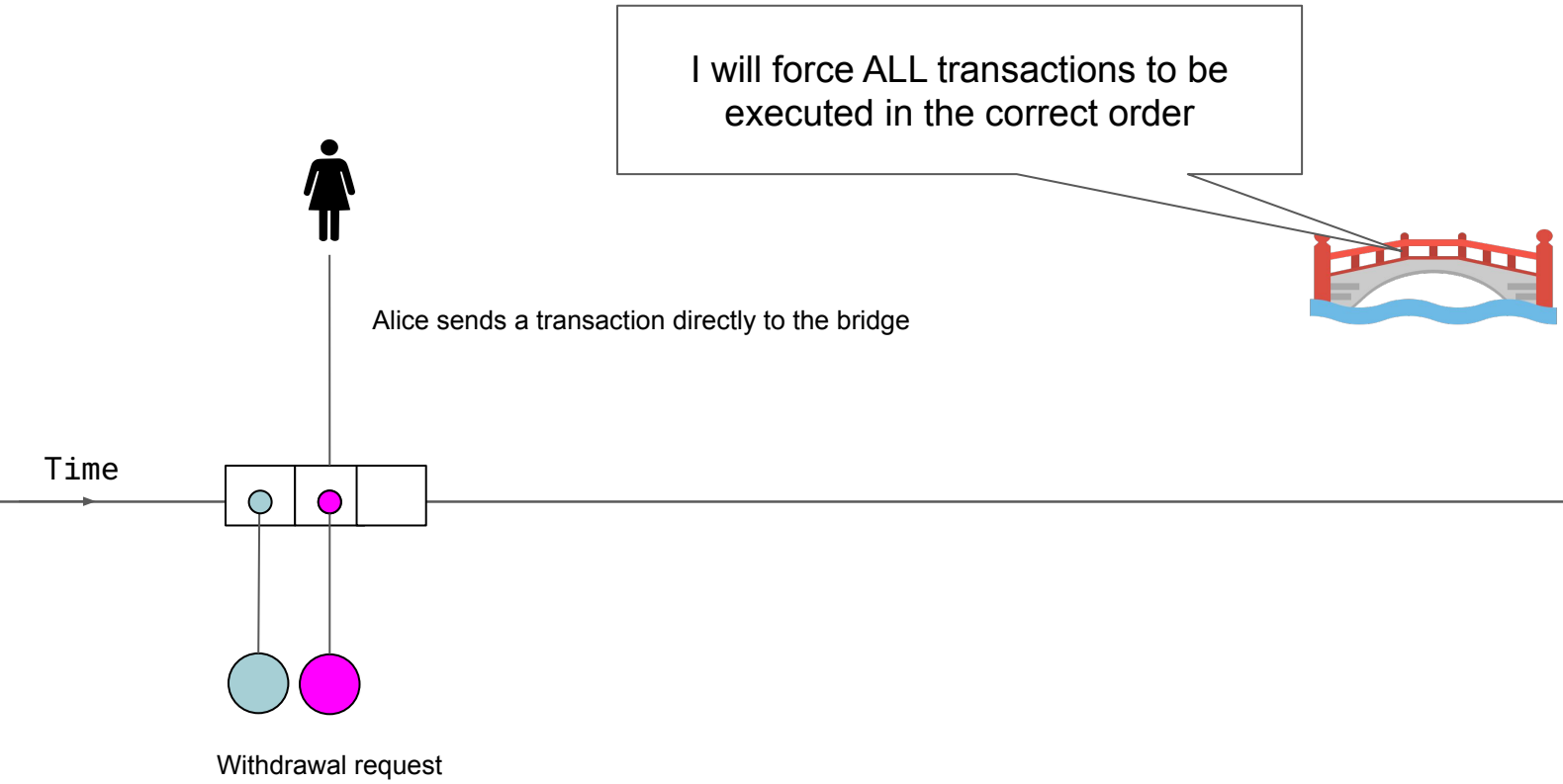
Enforcing censorship resistance

Censorship resistance

How can I withdraw my funds if the sequencer does not cooperate?



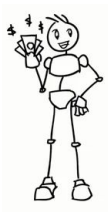
Forced inclusion: Bridge forces ordering of execution



Execution liveness (and the fast path)

Sequencer

Offers the “fast-path” and should have nothing to do with censorship-resistance

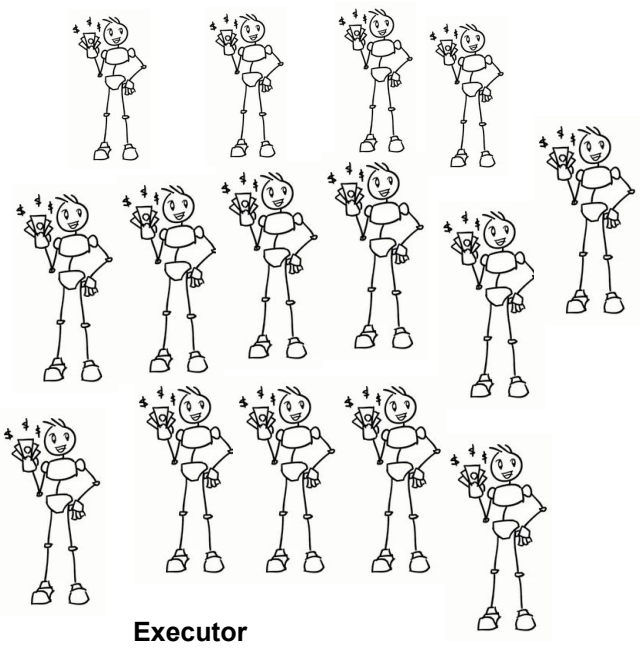


Executor

Trusted with liveness of execution (i.e., a transaction is eventually executed

Execution liveness (and the fast path)

Sequencer can be fully centralized and the off-chain system remains censorship resistant



Executor

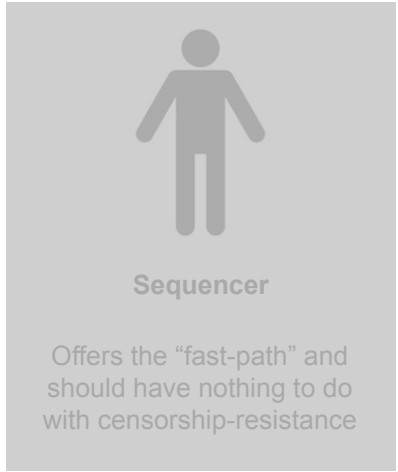
Trusted with liveness of execution

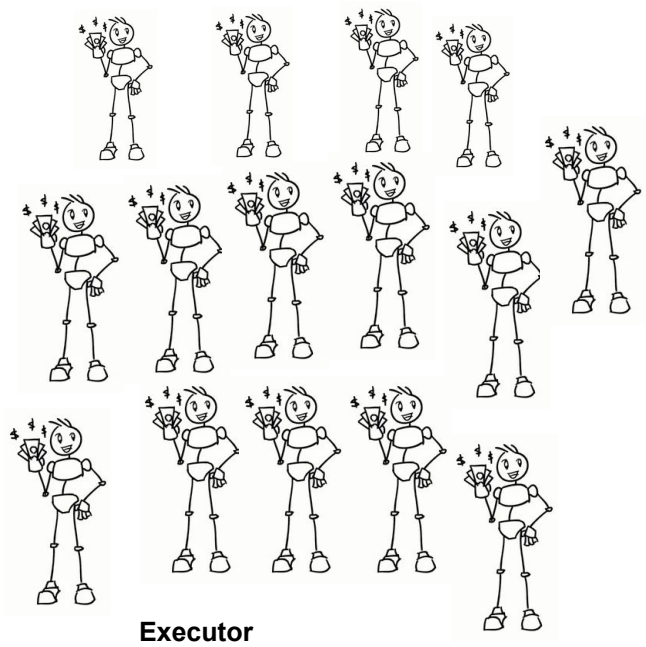
Sequencer

Offers the “fast-path” and should have nothing to do with censorship-resistance

Execution liveness (and the fast path)

Sequencer can be fully centralized and the off-chain system remains censorship resistant





Trusted with liveness of execution

One honest party

Security properties (summarised)

- **Data availability.** How does an honest user access the transaction history and recompute the same layer-2 ledger as everyone else?

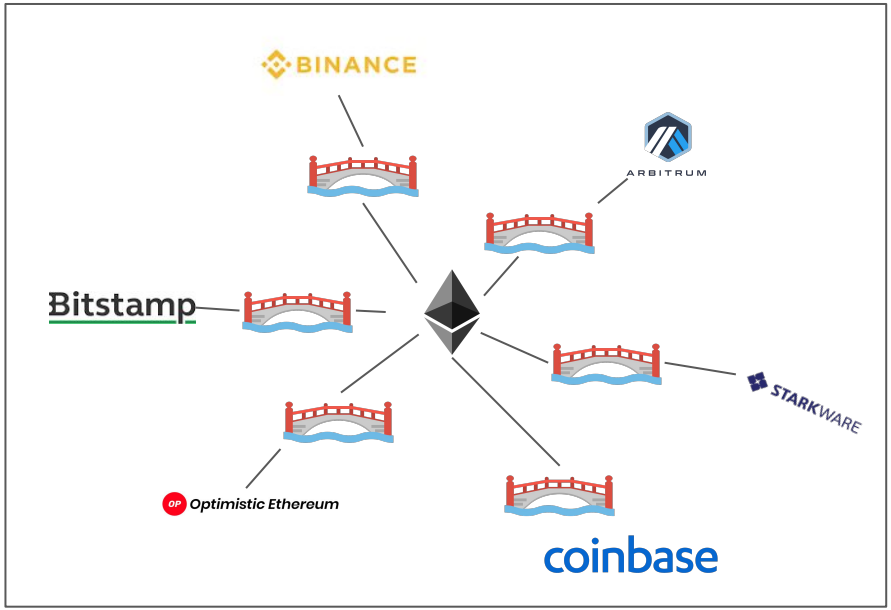
- **State transition integrity.** How can we convince the layer-1 blockchain that all transactions in the layer-2 blockchain are valid?

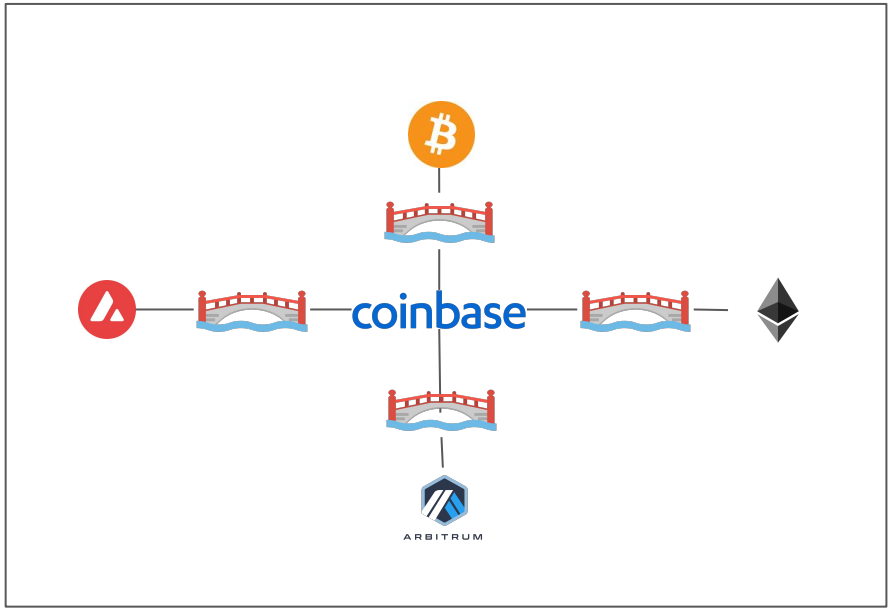
- **Censorship resistance.** How can an honest user withdraw their funds from the layer-2 blockchain without the sequencer cooperation?

If we can satisfy the above properties….

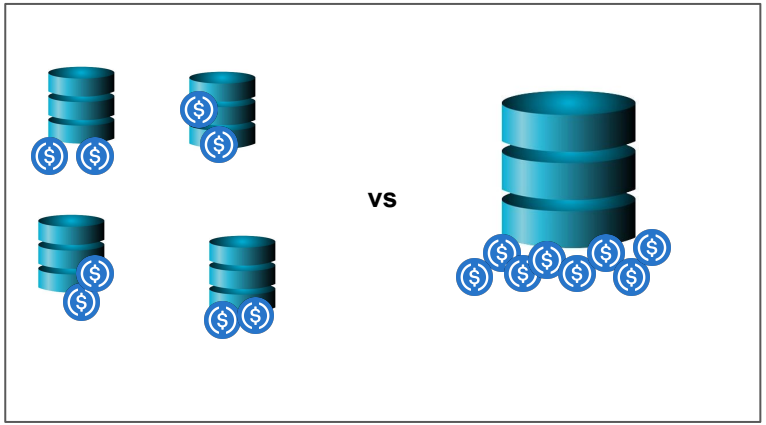

Then, hopefully we slay the beast and deploy a secure layer-2 system…

Other interesting problems emerge


BINANCE
ARBITRUM
Bitstamp
STARKWARE
Optimistic Ethereum
coinbase

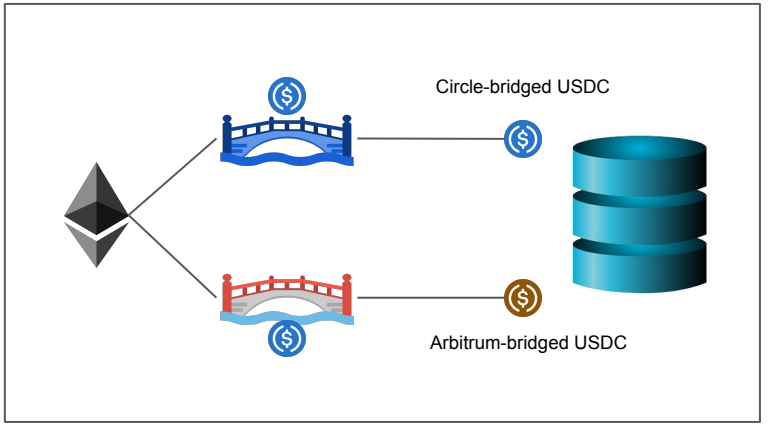

coinbase
ARBITRUM

Fragmentation of assets



Liquidity issues across databases

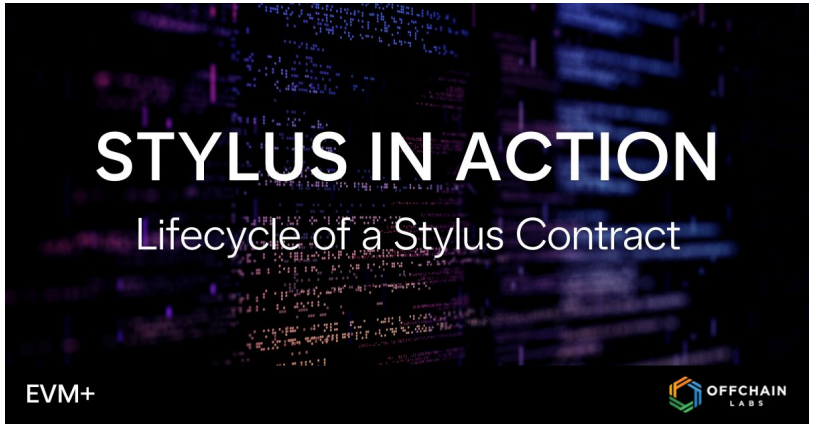
An asset is locked into several databases vs just one big liquidity pool

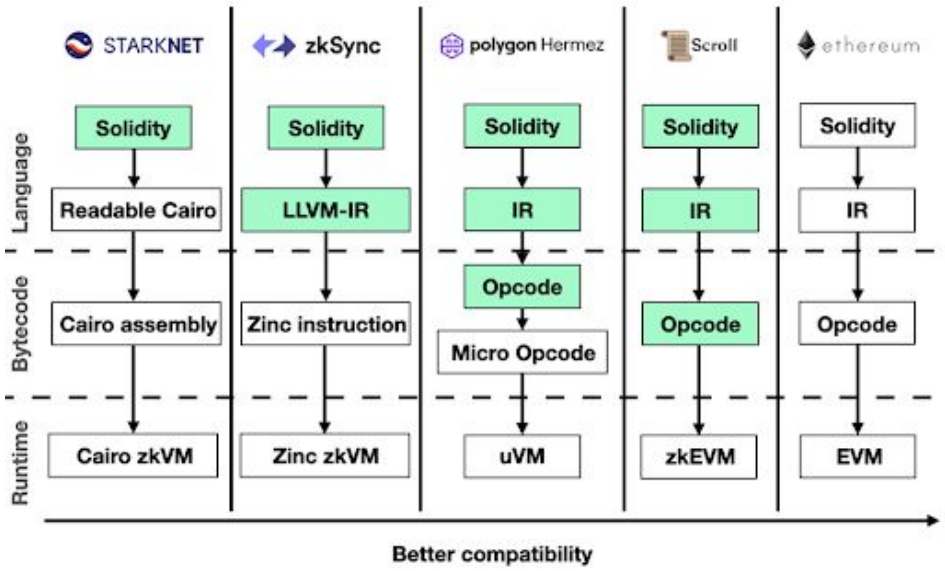


Competing bridges for the same asset

A single database may have several representations of the same asset, but they are not natively fungible with each other.

Experimental virtual machines

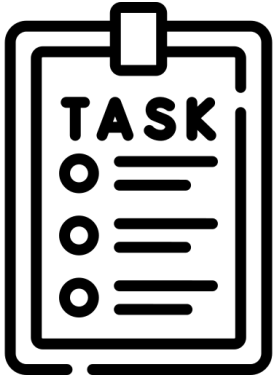




Open-access and “decentralization” of provers

Public bulletin board

Guarantees a consistent view to all parties and implemented as a blockchain (Ethereum)



Execution trace / data blobs

All parties can see the work that needs to be proven correct

Open-membership provers

As long as a user will make a financial commitment then they can participate as a prover

A well-evaluated protocol is missing!

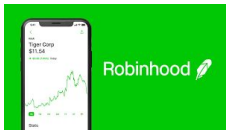
Fair reward for all parties?
Stop one larger prover outcompeting everyone else?
Allocating tasks to provers? Size of tasks?

Order-fairness and Extractable Value (MEV)

I can decide the order of all transactions in the batch!

How much $$ can I extract?

Ledger State
Transaction 1
..
..
Sequencer Tx
Victim Tx
Sequencer Tx
..
..
Transaction N



A potential business model?
Or an outright attack on the system?

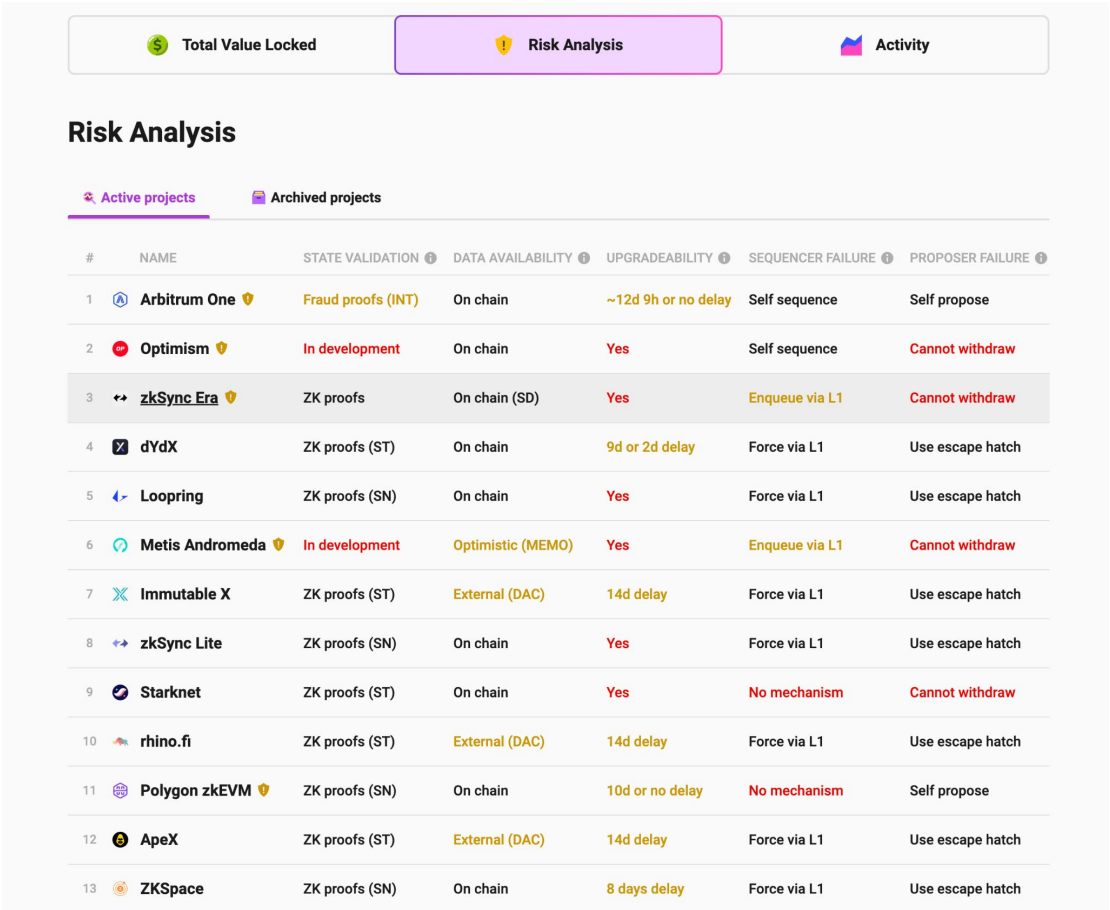
Buying Time: Latency Racing vs. Bidding in Fair Transaction Ordering*

Akaki Mamageishvili[1], Mahimna Kelkar[2], Jan Christoph Schlegel[3], and Edward W. Felten[1]

[1]Offchain Labs
[2]Cornell University
[3]City, University of London

May, 2023

Comprehensive risk analysis



Total Value Locked | Risk Analysis | Activity

Risk Analysis

Active projects | Archived projects

#	NAME	STATE VALIDATION	DATA AVAILABILITY	UPGRADEABILITY	SEQUENCER FAILURE	PROPOSER FAILURE
1	Arbitrum One	Fraud proofs (INT)	On chain	~12d 9h or no delay	Self sequence	Self propose
2	Optimism	In development	On chain	Yes	Self sequence	Cannot withdraw
3	zkSync Era	ZK proofs	On chain (SD)	Yes	Enqueue via L1	Cannot withdraw
4	dYdX	ZK proofs (ST)	On chain	9d or 2d delay	Force via L1	Use escape hatch
5	Loopring	ZK proofs (SN)	On chain	Yes	Force via L1	Use escape hatch
6	Metis Andromeda	In development	Optimistic (MEMO)	Yes	Enqueue via L1	Cannot withdraw
7	Immutable X	ZK proofs (ST)	External (DAC)	14d delay	Force via L1	Use escape hatch
8	zkSync Lite	ZK proofs (SN)	On chain	Yes	Force via L1	Use escape hatch
9	Starknet	ZK proofs (ST)	On chain	Yes	No mechanism	Cannot withdraw
10	rhino.fi	ZK proofs (ST)	External (DAC)	14d delay	Force via L1	Use escape hatch
11	Polygon zkEVM	ZK proofs (SN)	On chain	10d or no delay	No mechanism	Self propose
12	ApeX	ZK proofs (ST)	External (DAC)	14d delay	Force via L1	Use escape hatch
13	ZKSpace	ZK proofs (SN)	On chain	8 days delay	Force via L1	Use escape hatch

Measurement & comparison

Do the systems satisfy the goals of a rollup?

What are people using them for?

How do they compare with each other?

we don’t even have a good formal model of a validating bridge and its variants

Is it still worth it?

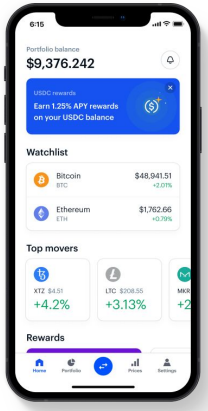
Legacy and Web2

- **No proof of solvency.** Liabilities recorded in a private and brittle system.
- **Funds are freezable.** Operator can ignore withdrawal request and prevent all transfers.
- **Humans protect funds.** Meatspace, not a technology stack, is protecting funds.
- **Closed-source.** No public audit or community approach to attest to quality of technology stack to protect billions of dollars.
- **Lack of oversight.** Very few canaries who can report when TradFi lends your funds out in waves of credit bubbles.

Building towards Web3

- **Proof of solvency.** Anyone can check the system is fully collateralized.
- **Real-time audits.** Anyone can check the validity of all transactions and the database.
- **Forced transactions.** Anyone can forcefully get their transaction processed.
- **Open-participation.** Anyone can contribute towards the system’s operation.

Future Architecture for TradFi in Web3?



Centralised front-end to take transactions

A single, or distributed operators, who take user-generated transactions and passes them on.

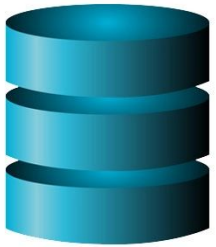
Decentralized backend protecting database

Anyone, including a swarm of cyberhornets, can protect the database integrity

Welcome to Web3: Protecting a global database

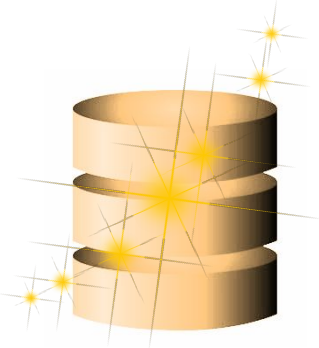
Custodial database

Opaque and not publicly accessible.



Public database

Publicly accessible, but the operators can halt/freeze it



Open Database

Anyone can read, write and protect the database.

Fully trusted

Minimized trust

Welcome to Web3: Protecting a global database

Custodial database

Opaque and not publicly accessible.

Public database

Publicly accessible, but the operators can halt/freeze it

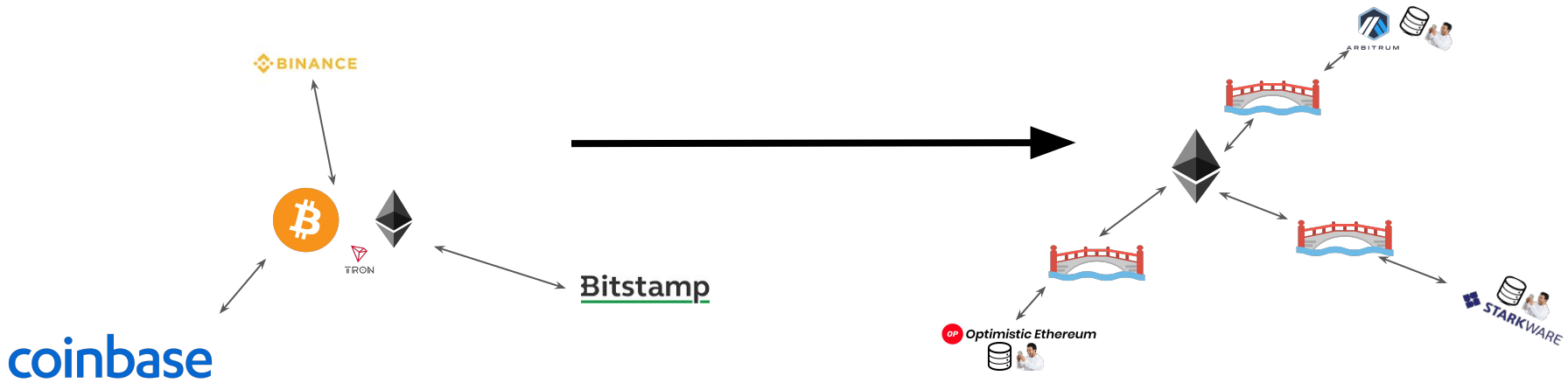
Open Database

Anyone can read, write and protect the database.

Web2 Exchange **Sidechain** **Rollup**

Welcome to Web3

Rise of open databases to
replace custodial services (and exchanges)

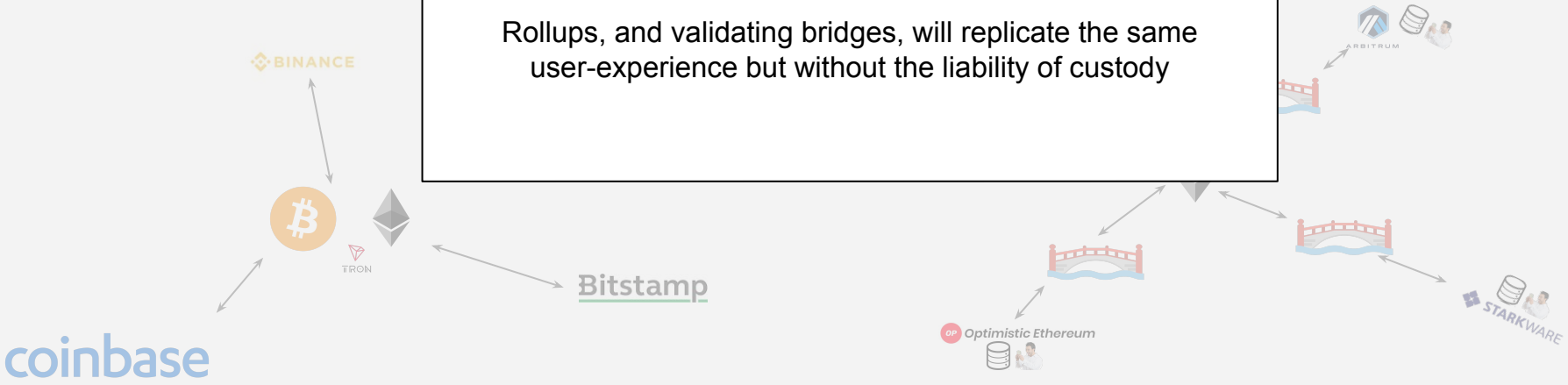


Welcome to Web3

Rise of open databases to

Custody is a liability for most off-chain systems

Rollups, and validating bridges, will replicate the same user-experience but without the liability of custody



It is VERY difficult to replicate human processes to secure billions of dollars

It is VERY difficult to replicate human processes to secure billions of dollars

We just need ONE rollup team to get it right and it can be re-instantiated for all service providers

It is VERY difficult to replicate human processes to secure billions of dollars

---

We just need ONE rollup team to get it right and it can be re-instantiated for all service providers

---

Users do not care about the custody issues.

Operators do and they'll drive its adoption.

Custody is an unnecessary liability.