Decentralized Systems

Optimistic Rollups

Smart Contract Scalability

- Ethereum is “the slowest computer” (and most expensive) in the world
- **Every verifier executes everything**
  - Worst redundancy possible
- We can extend payment channels to **state channels**
  - Limited to two-player interactions (e.g., chess)
- Can we do better?

Layer-2 Blockchains

- **Blockchains on top of blockchains**
- Goal:
  - Layer-2 (L2) transactions are run on **a few machines**
  - The underlying L1 blockchain **guarantees correctness**
- We'll see a technique called **optimistic rollup**
  - Reference implementation: Arbitrum

References

- Kalodner et al. Arbitrum: Scalable, private smart contracts. Usenix Security 2018.
- Finematics. ROLLUPS - The Ultimate Ethereum Scaling Strategy? Arbitrum & Optimism Explained – YouTube video, 2021.
- Arbitrum. Inside Arbitrum Nitro.

Optimistic Rollup

Rollup Transaction

- A transaction that “rolls up” several blocks of the L2 blockchain and bundles it all in one transaction
- Posted by a **Manager** that created the blocks
- Blockchain state as a Merkle tree, input messages as attachments to the L1 block
  - We’ll see more on this later
- Meaning: “The state with hash A and this input leads to state with hash B”

Optimistic Rollup Idea

- There is a **happy path** followed **most of the time**
  1) A **manager** publishes its assertion “from state A and this input we get to state B” (and send these messages/currency from this chain to these L1 wallets)
  2) Some other entities (confusingly also called managers) verify that the computation is correct
  3) If after some time (e.g., blocks) nobody disputes that, the new state is **confirmed**

Optimistic Rollup: Challenge

- Managers need to **stake** some currency
- If an assertion is **challenged**:
  - The L2 chain's progress is paused
  - The conflicting managers play a **game** on the L1 chain
  - The L1 smart contract of the L2 chain will make sure it will be won by whoever is correct
  - The loser pays part of its stake to the winner, the rest goes somewhere else (e.g., the verifier)
- It *should* be irrational to willingly post **wrong states**

Challenge: Bisection Game

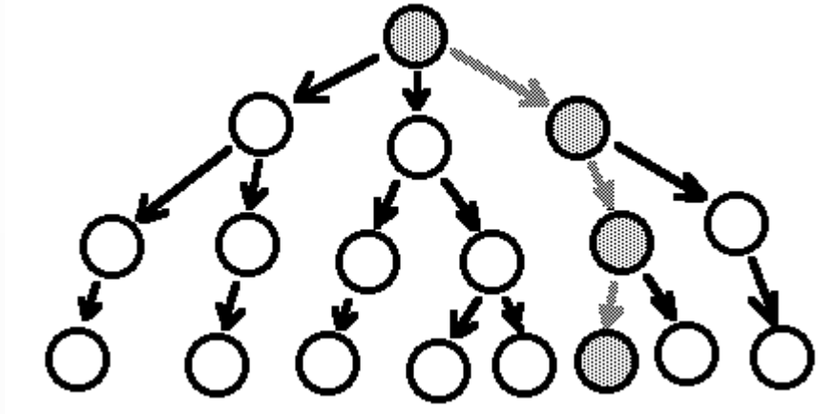
- A: “State A leads to state B in N steps”
- B: “I challenge it!”
- A: “State A leads to state C in N/2 steps, and C to B in N/2 steps”
- B: “Prove you get to from A to C (or C to B) in N/2 steps”
- A: “You get from A to D in N/4 steps, and from D to C in N/4 steps”
- …

Bisection Game: Result

- A and B get in log(n) steps to **a single step**
  - n is the number of execution steps in the whole “rollup block”
  - If either doesn’t answer within a deadline, **they lose**
- That step gets **replayed on the main chain** to decide the winner
- How to do this?

Running a Single Step

- Remember that a state is a **Merkle tree**
- A reveals **only the parts of the state needed to run the instruction**
  - **Merkle proof:** you expand the nodes containing the parts of the state touched by the instruction
  - The L1 chain **verifies the hashes**
  - It **runs the instruction**
  - It **verifies the result**



Sequencing Input

- A **sequencer** takes all the messages, compresses them, and posts them to the blockchain as a binary “blob”
  - Such blobs are **erased after some time** (~18 days in Ethereum)
  - Enough time to **challenge** the state, then they’re lost
  - Blocks have been introduced as **Proto-Danksharding** in March ‘24 (input was stored in CALLDATA before)
- Blobs are way cheaper than storing blockchain data
  - The security comes from the fact they’re signed, as usual

Avoiding Censorship

- To make sure the sequencer can’t censor messages, you can also **post** the ignored ones **on the blockchain**
- They will **have to be included** in the next rollup block
- No punishment for the sequencer: they may not have seen it in good faith

Putting It Together

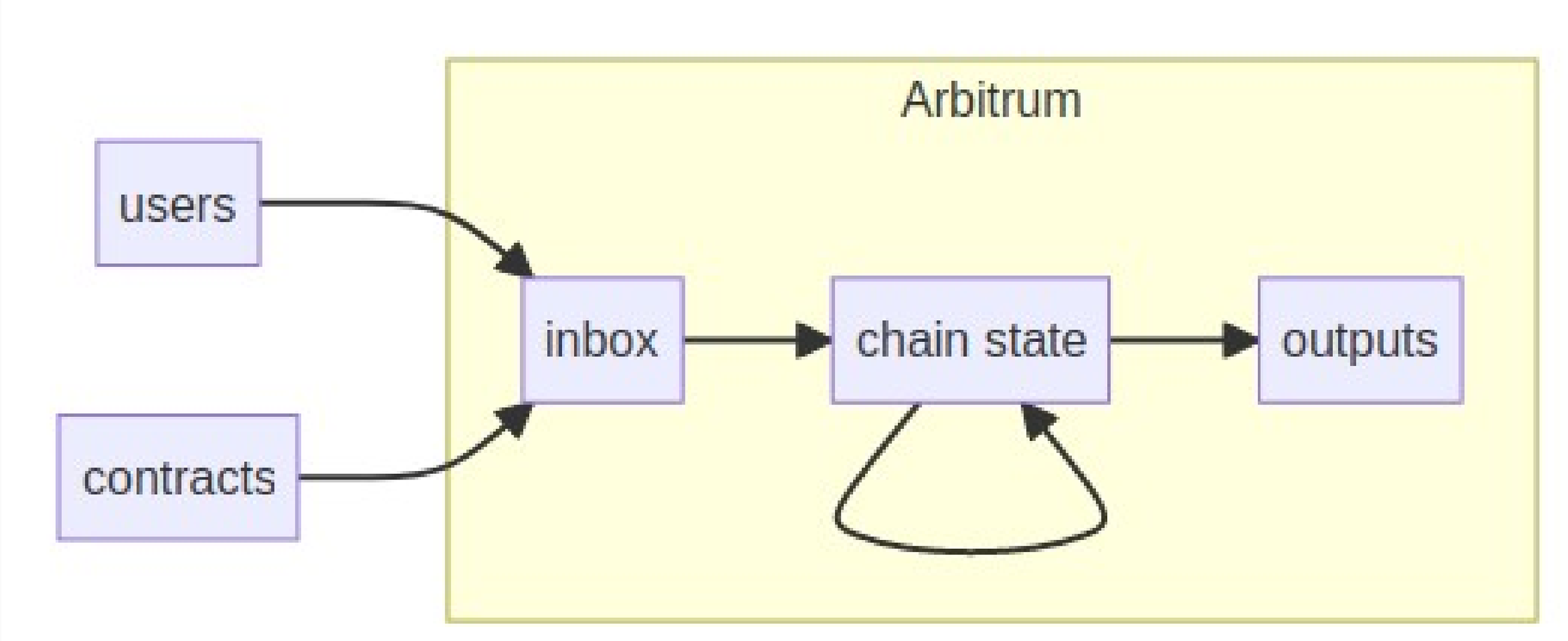
- An L2 chain that is secure if the L1 chain is secure
- …and if there's at least a honest manager verifying the rollup transactions
- The L2 chain can be faster and cheaper, because it's verified by much fewer nodes
  - Also, gas limits can be way larger
- Users have ways to bring currency from the L2 to the L1 chain

Arbitrum

About Arbitrum

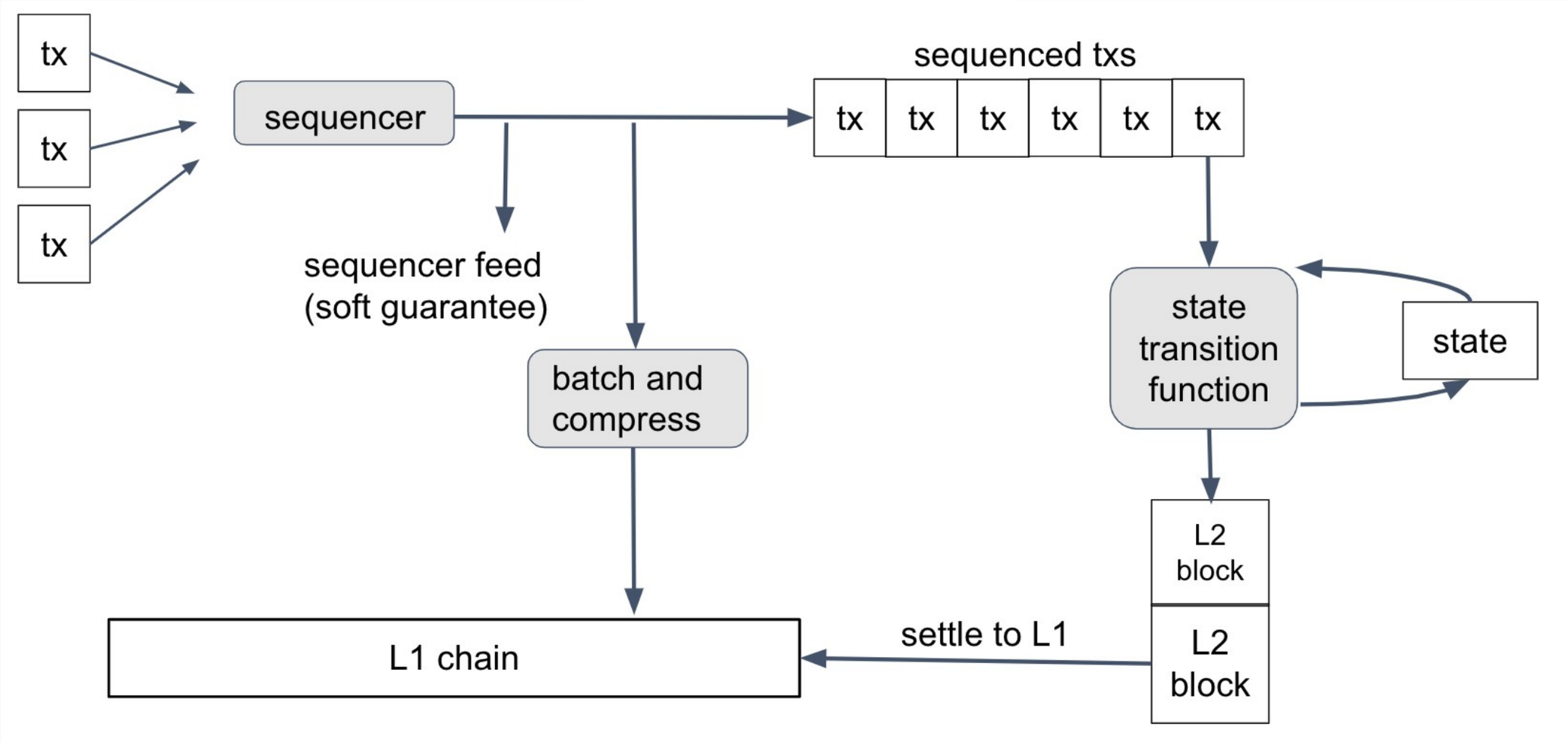
- The most popular L2 network at the moment
- Based on the protocol described in a 2018 USENIX Security paper
- Currently used by many users, while still under development

Architecture (1)



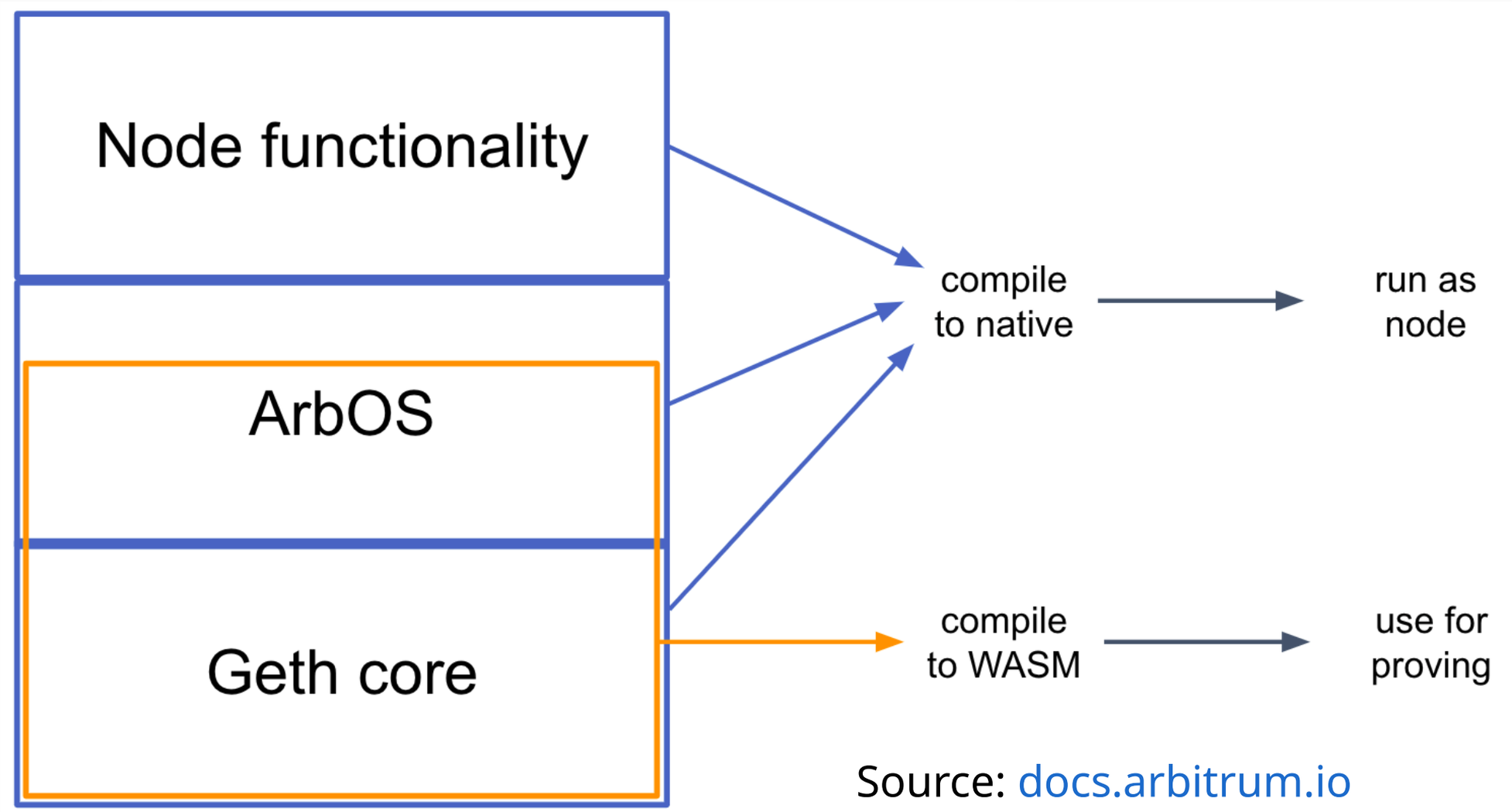
Source: docs.arbitrum.io

Architecture (2)



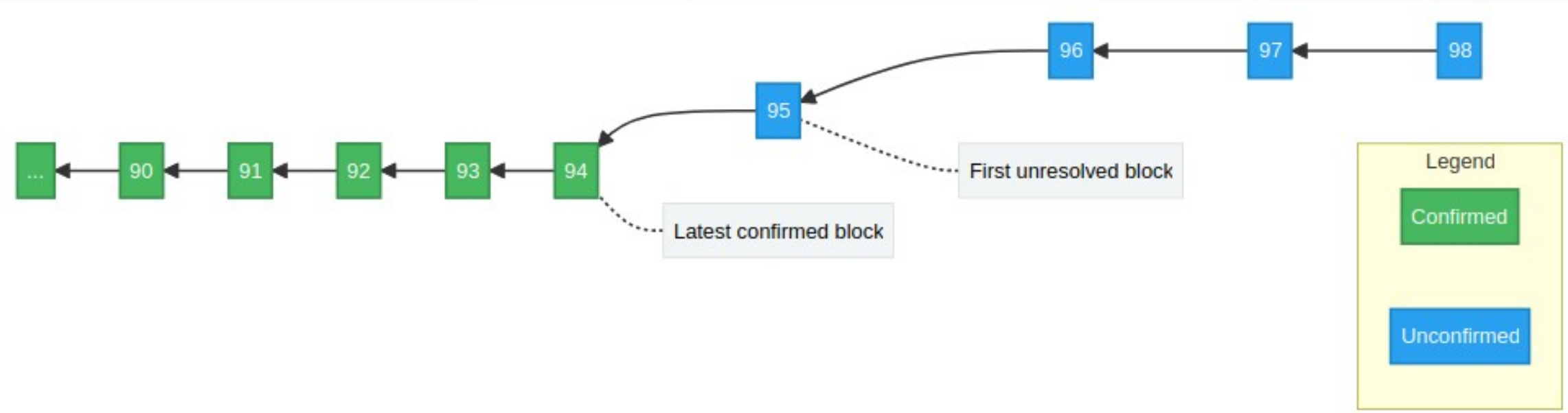
Source: docs.arbitrum.io

Implementation



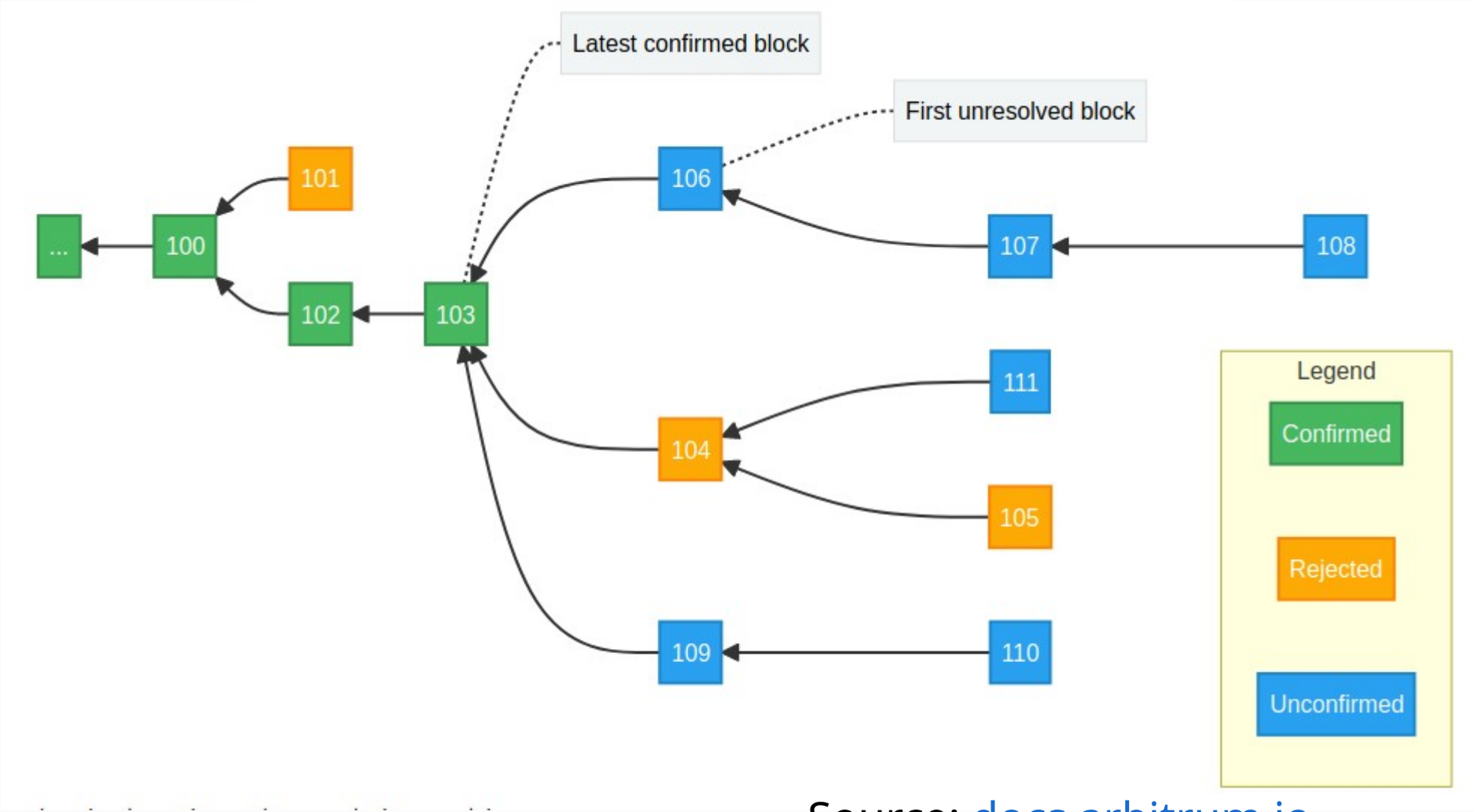
Source: docs.arbitrum.io

Happy Path



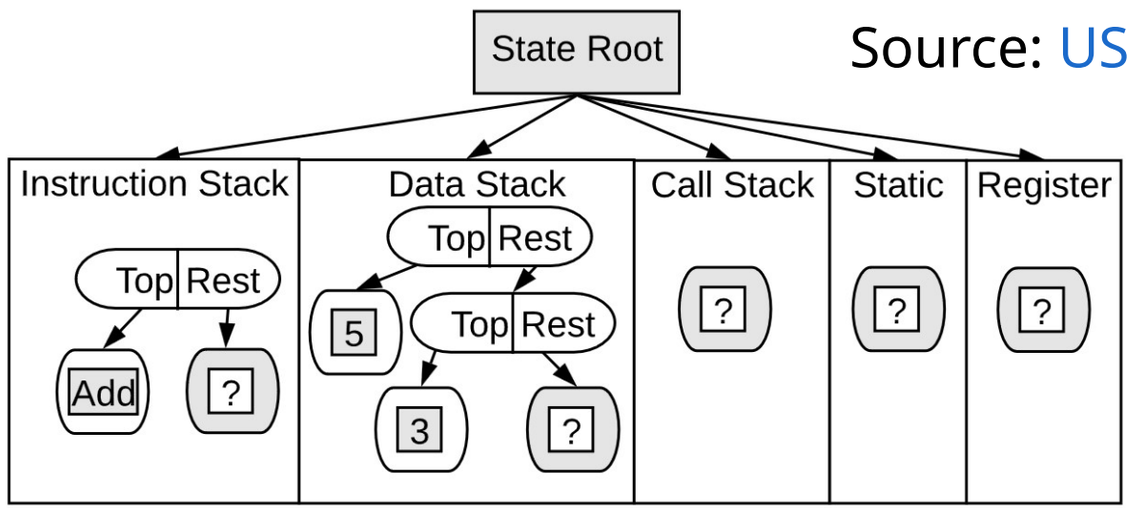
Source: docs.arbitrum.io

(Very) Unhappy Path



Source: docs.arbitrum.io

Virtual Machine for Proofs



Source: USENIX Security ‘18 paper

- The state as a Merkle tree is visible to the VM
- Micro-instructions that only touch elements close to the top of the stack
- EVM instructions are library calls
- O(1) (rather than O(log n)) to run on the L1 blockchain

K-Way Dissection

- Alice: “State X leads to Y in N steps”
- Bob: “No. State X leads to:
  - $X_{1,B}$ in N/K steps
  - $X_{2,B}$ in N/K steps from $X_{1,B}$
  - …
  - $X_{K,B}$ in N/K steps from $X_{K-1,B}$”
- A: “No. State $X_{2,B}$ leads to
  - $X_{(2,1),A}$ in N/K^2 steps…”
- Less rounds and messages involved in the dissections

Status

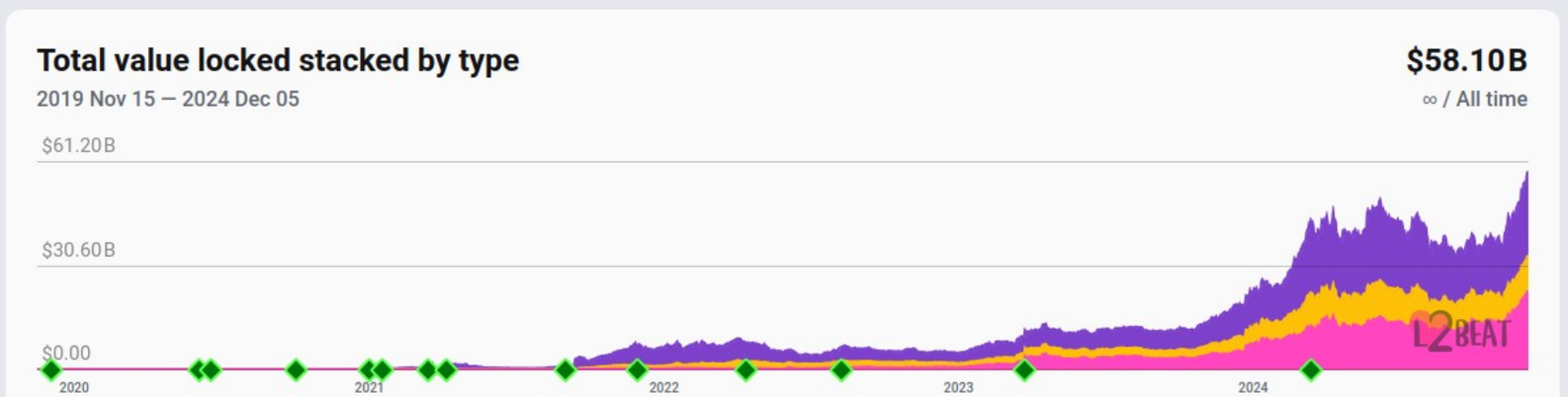
- Currently, **partly centralized**
- Ownership through a DAO governed by tokens
- Validators through an allow list
- The sequencer is centralized
  - It can only delay transactions, not prevent them since they can be posted on the L1 blockchain

L2 Panorama

Zero-Knowledge Rollups

- A promising alternative is based on **crypto proofs** of correctness
- The crypto machinery belongs to the category of **non-interactive zero-knowledge proofs**
  - Proof of correctness that do not reveal any specific information beyond the validity of the statement
  - **zk-SNARK**: zero-knowledge succinct non-interactive argument of knowledge
- Computation-intensive, difficult to prove generic computation

Existing L2 Chains



Type | Stack | Stage | Purpose | RaaS | Host Chain | Exclude associated tokens

Rollups 48 | Validiums & Optimiums 67

#	NAME	TOTAL	CANONICAL	EXTERNAL	NATIVE
1	Arbitrum One	$21.04B ▲15.9%	$7.03B ▲9.74%	$5.39B ▲9.19%	$8.60B ▲26.5%
2	Base	$13.85B ▲17.3%	$4.85B ▲19.3%	$462.52M ▲15.0%	$8.53B ▲16.3%
3	OP Mainnet	$8.78B ▲11.0%	$4.50B ▲10.0%	$493.73M ▲14.9%	$3.78B ▲11.7%
4	Blast	$1.63B ▲1.74%	$7.16M ▲1.41%	$1.33B ▲2.64%	$295.50M ▼2.14%
5	ZKsync Era	$1.37B ▲22.0%	$468.16M ▲8.78%	$13.09M ▲ 242%	$895.56M ▲29.0%

Source: l2beat.com