Plasma Chains

CS839 – Kai Mast

Recap: Layer 2

Goal: Higher scalability and lower transaction fees.

- Move computation off the main chain
- Only rely on the global chain during deposit, withdrawal, or failure
  - There usually is some fixed timeout window (assumes synchronous time)
- Payment/State channels are only one kind of Layer 2 solution!

Plasma: Scalable Autonomous Smart Contracts

- Whitepaper published in 2017
- **Authors:** Joseph Poon (Lightning Whitepaper) and Vitalik Buterin (Ethereum)
- **Today:** Plasma is the basis/inspiration for other off-chain solutions

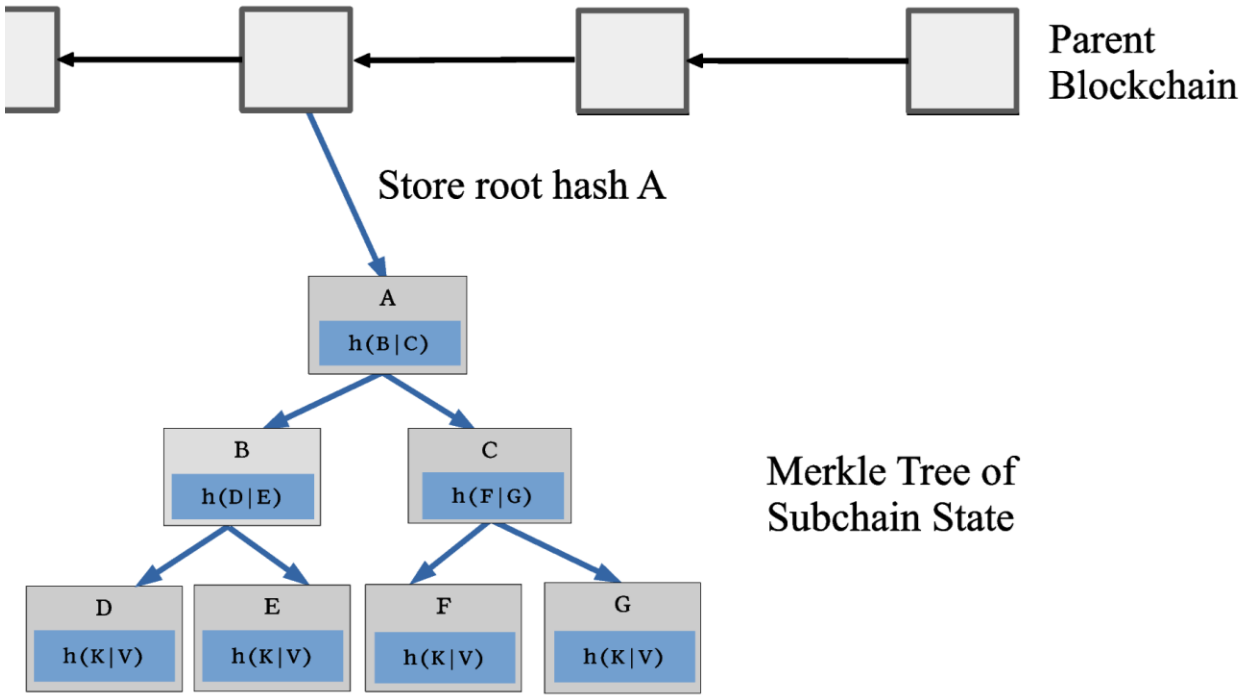
Payment Channels vs. Plasma Chains

- Joint account set up between two parties on the main chain
- State is updated using commitments
- Either party can close the channel at any time

- A single party (or small set of validators) sets up a plasma chain and associated contract on the main chain.
- Many users can move their funds into the plasma chain

Anchoring Plasma Chains

- Parent blockchain holds a smart contract associated with the subchain
- Each plasma chain is its own blockchain:
  - Only store headers/metadata on the parent blockchain
    - Merkle root of transactions
    - Bitmap vector of UTXOs
  - Like a regular blockchain, periodically batch updates and create new blocks



Key Problem: Availability

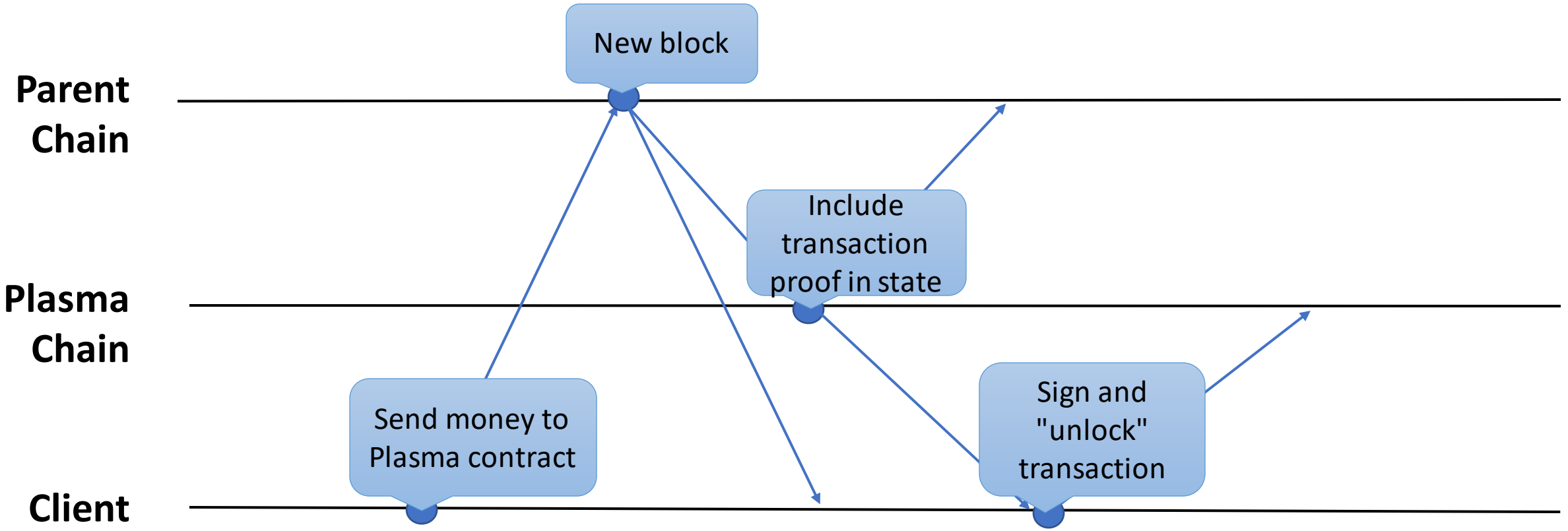
- Plasma chain provider might become unavailable at any point in time
- Chain provider might hide blocks to gain an advantage
- Users must constantly monitor the plasma chain and store state locally

Plasma Proof-of-Stake and Bonds

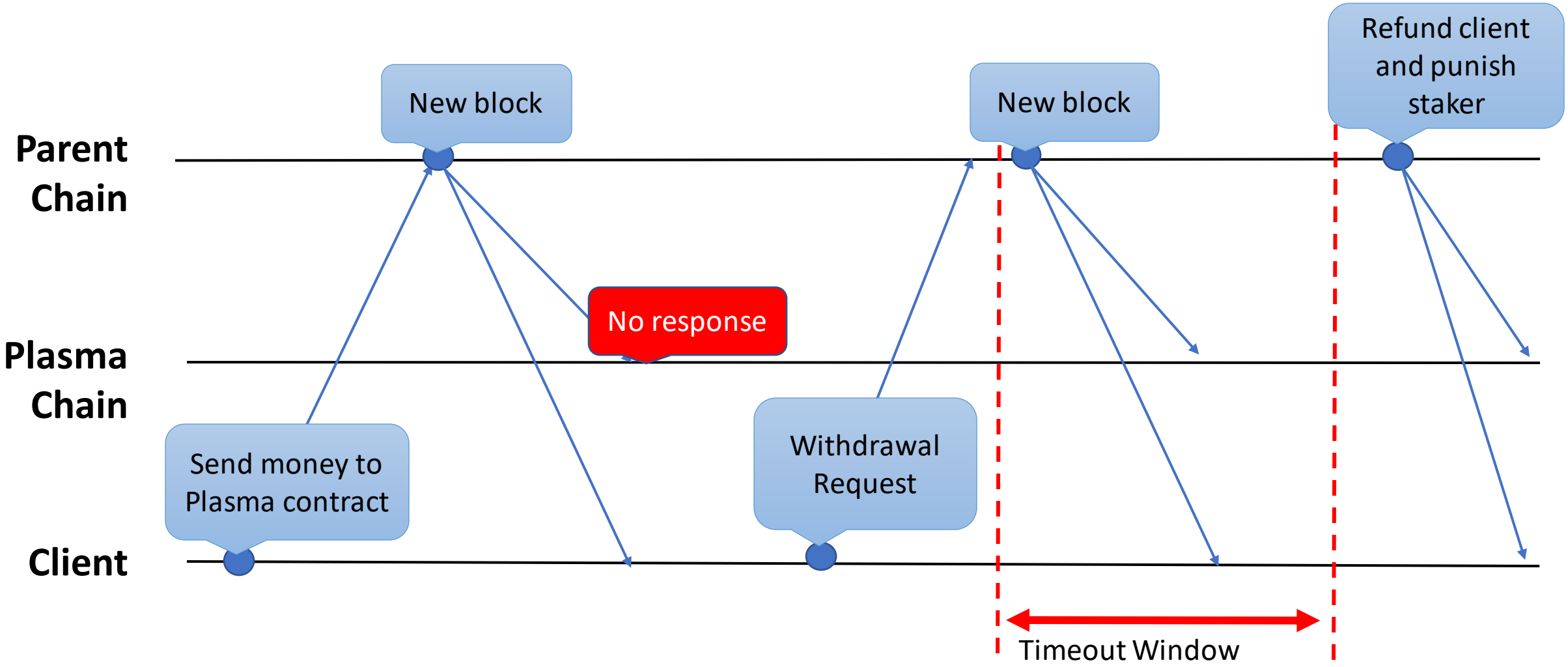
Goal: Use stake to assign voting power and ensure correctness

- Multiple parties can lock up stake in the Plasma chain's contract
  - Each party becomes a **validator** associated with a **bond**
- Size of a validators bond determines their voting power
  - Leader are assigned probabilistically
- If validators misbehave users generate a fraud proof and the smart contract penalizes them by slashing their bond
  - More on fraud proofs later

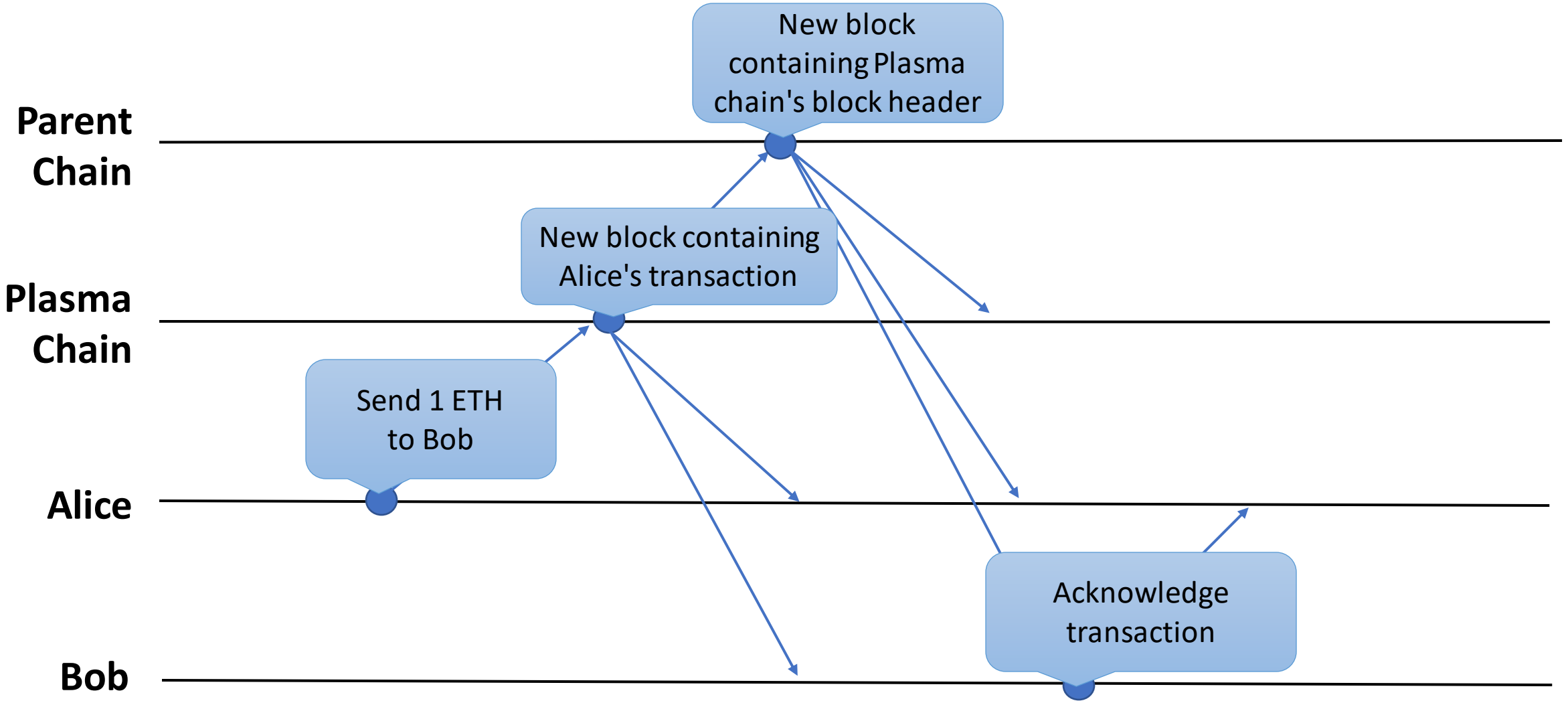
Plasma Deposits



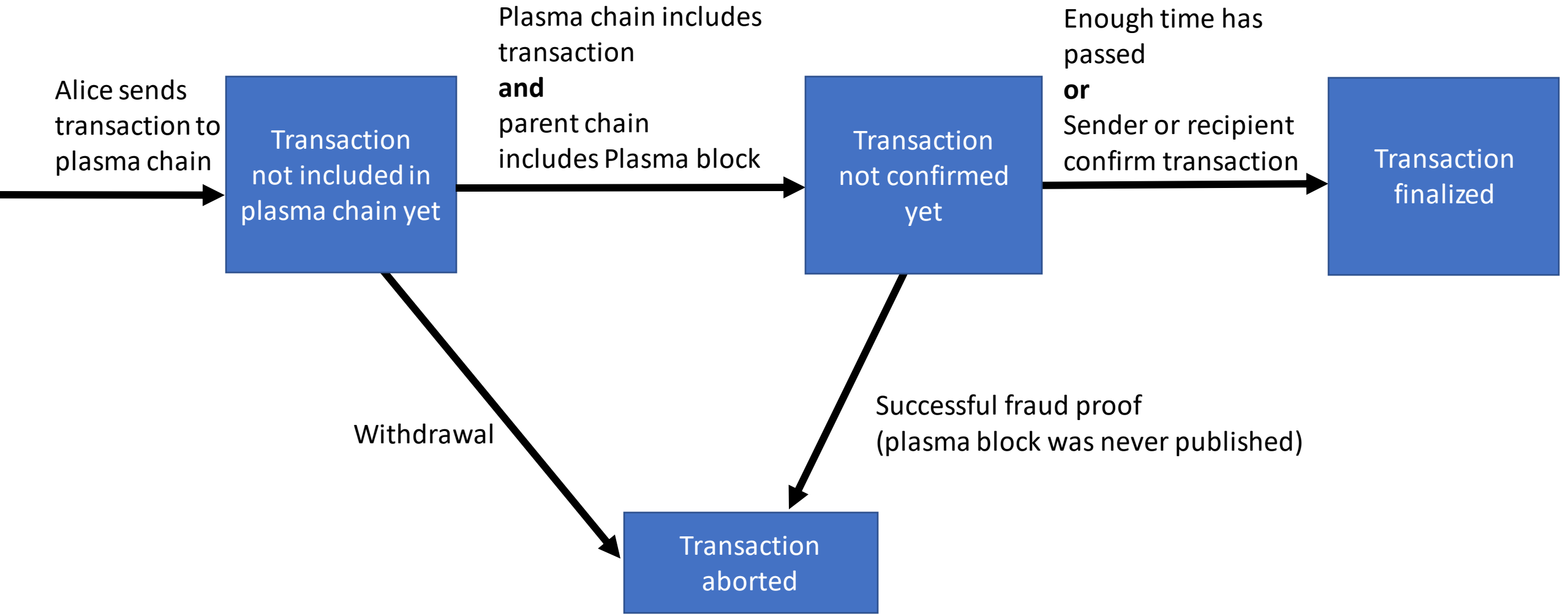
Plasma Deposits with Fraud



Plasma State Transitions



Plasma State Transitions

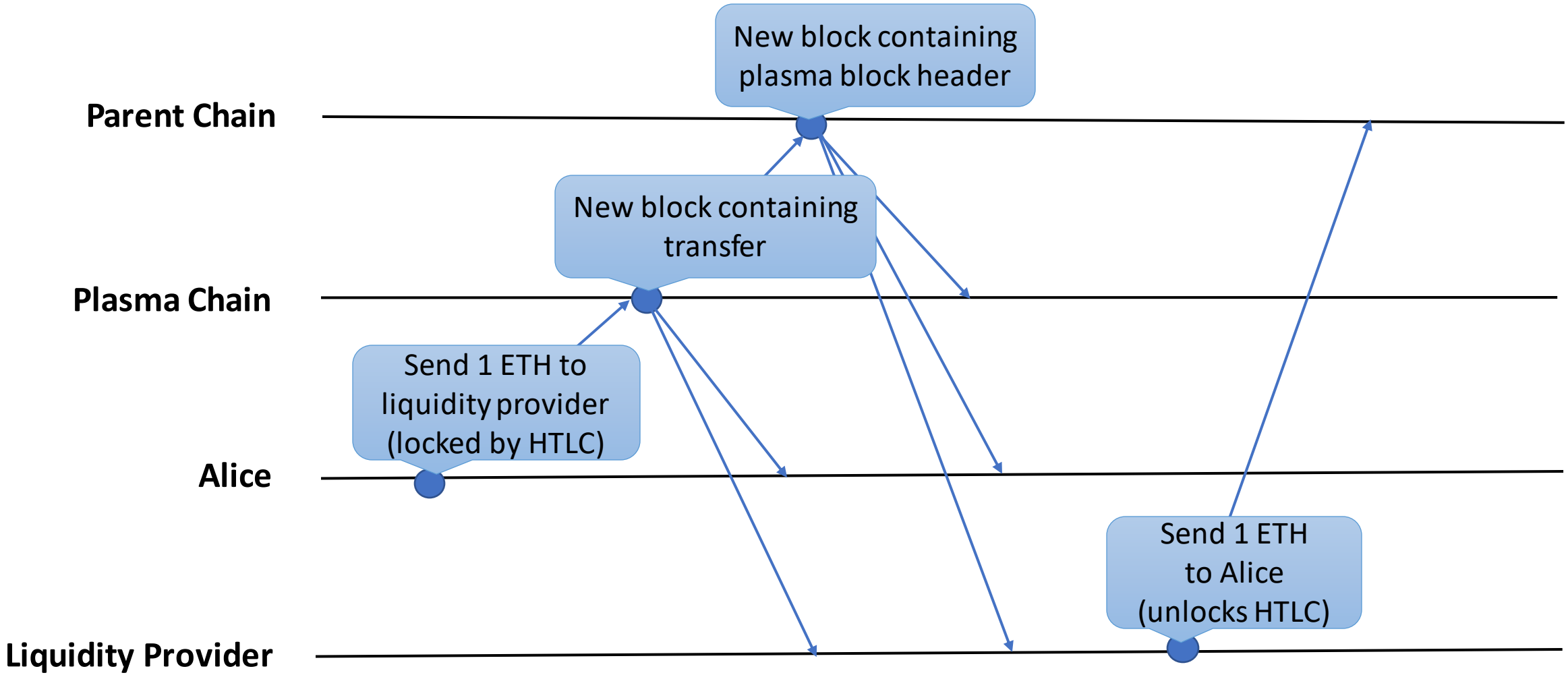


Plasma Withdrawals

- Client sends withdrawal transaction to parent chain
  - Contains the location of the UTXO(s) in the bitmap

- There exist some delay to allow for fraud proofs
  - Other parties may prove that the funds have already been spent

Fast Withdrawals



Adversarial Mass Exit

Goal: Users want to leave a faulty Plasma chain, e.g., one that is withholding blocks, and move to a different one

- Users organize around an exit coordinator that coordinates with the destination chain
  - There can be multiple mass exits, and thus exit coordinators, at the same time
- The exit coordinator validates the source chain as much as possible
  - Because it is faulty, not all state may be available
- The exit coordinator then creates a mass exit initiation transaction (MEIT) on the parent chain
  - There is a bond attached to the transaction in case the coordinator fails
  - The coordinator may charge a fee
  - All participants sign off the transaction
- After some challenge period, the mass exit is finalized

Mass Exit Disputes

Case 1: A user sees their UTXO in a MEIT without their consent

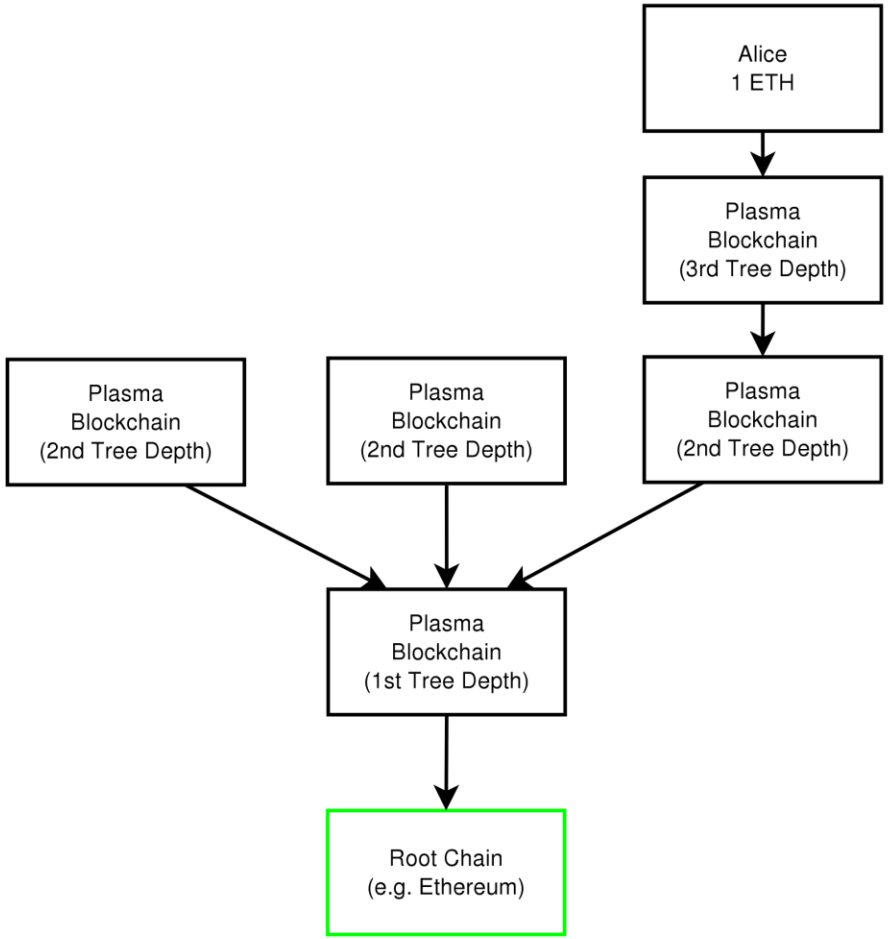
- The user broadcasts a challenge attached with a bond
- If challenge is not disputed, the mass exit is cancelled
- If the challenge is disputed, Alice's bond is slashed

Case 2: A MEIT includes a UTXO that has already been spent

- Someone creates a challenge (bitmap of a more recent block header) with a proof that the UTXO has already been spent
- Challenge can be disputed
  - Challenger might not have the contents of the block

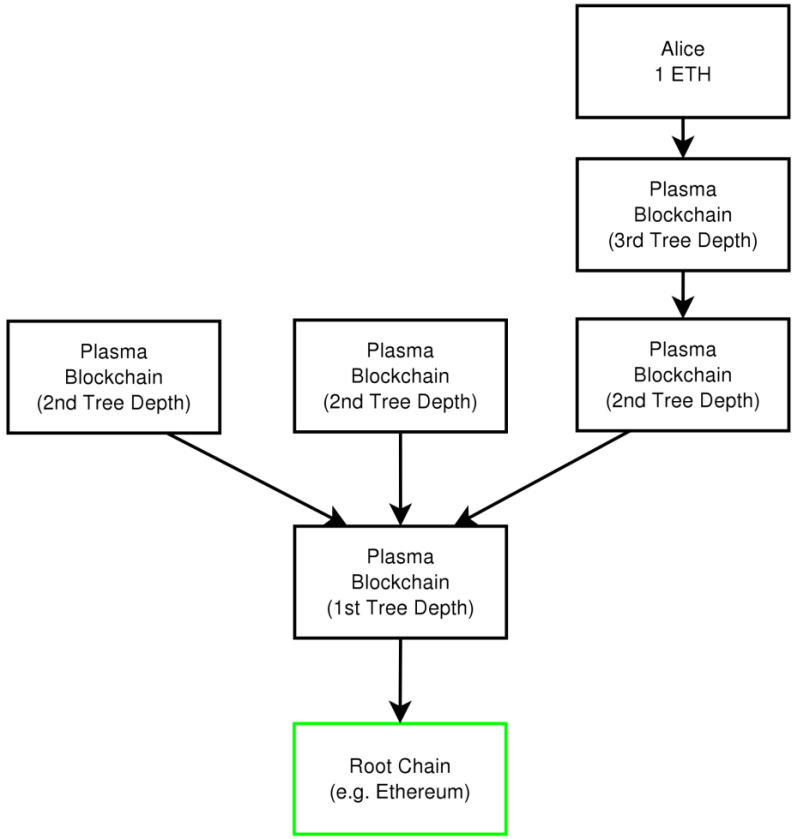
Recursive Plasma Chains

- The parent chain of a plasma chain can be another plasma chain
- For a transaction to be finalized, we must verify all parent chains
- There can be multiple parent chains for additional reliability and fault tolerance



Plasma Map-Reduce

Map send computation/transaction to multiple chains



Reduce collects results using the tree/web structure

Example Applications

Reddit Clone

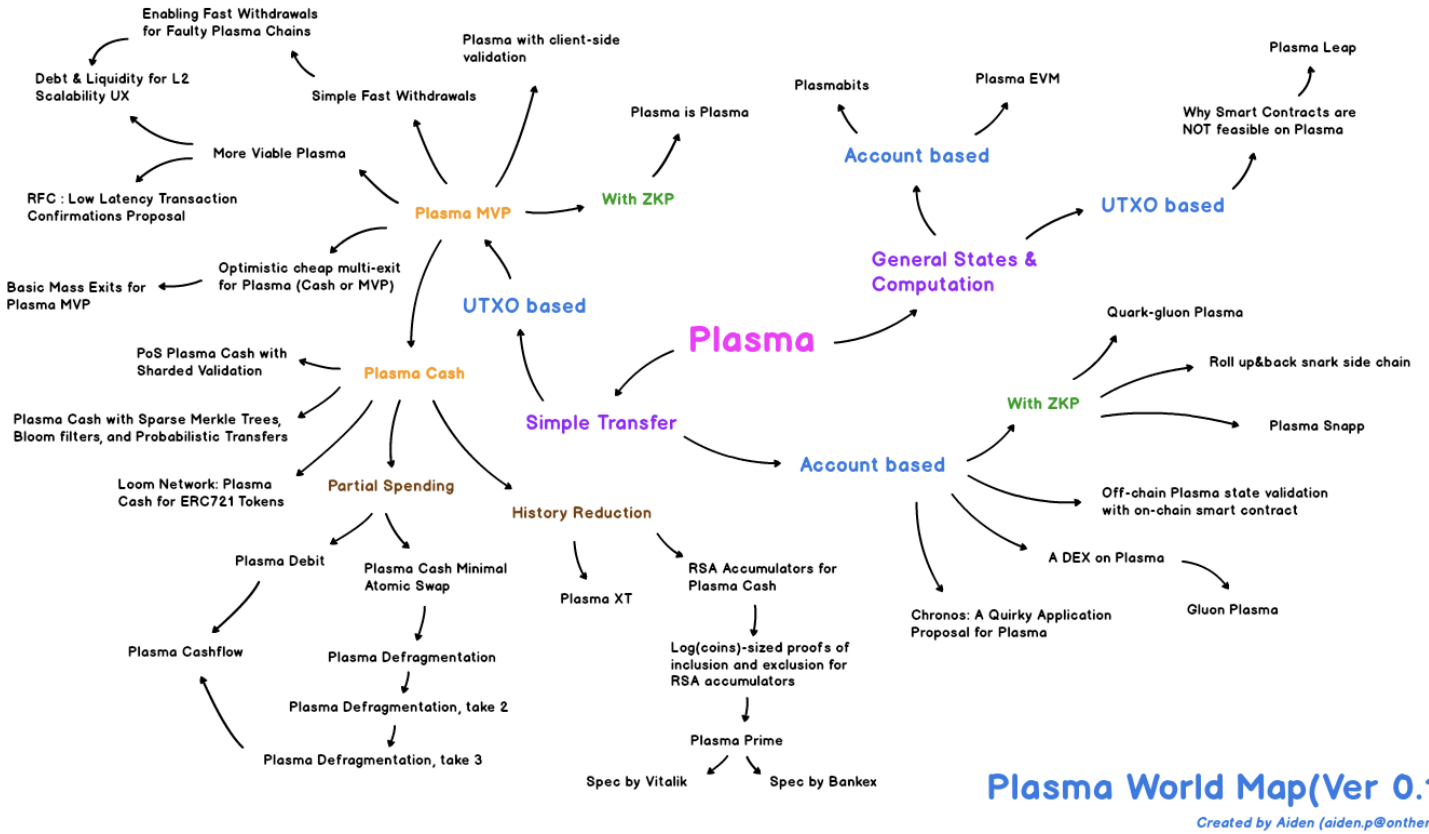
- Each subreddit is its own plasma chain
  - Posts and comments are issued using chain transactions
  - Chain validators enforce access control (who can post what)
- Map-Reduce is used to manage the front page
  - e.g., collect the most popular topics

Private Chains

- Chain validator and users can keep the block contents private, if desired

Plasma Today

Many different variants and implementations exist:



Source: https://medium.com/onther-tech/plasma-world-map-ba8810276bf2

Optimistic Rollups

- Smart contract is stored on the parent chain
- Each transaction/call to the contract is stored on the parent chain
- An off-chain aggregator executes transaction and updates the state on-chain

Advantages:

- Can run any kind of smart contract
- No availability problem; state can always be recomputed on chain

Disadvantage:

- Does not (easily) allow for batching

Layer Two Watchtowers

Goal: Do not require users to audit the off-chain solution constantly

- Users pay a third party to audit the layer two system
- Third party puts up a bond to make them accountable
  - It is penalized if it does not actually audit the L2 system
- See the Pisa [McCorry; 2019] paper for more details

Discussion

- Plasma vs. Lightning?
  - Lightning is less centralized (in theory)
  - Plasma may allow for more complex applications
- Problems with Plasma?
  - Bonds need to be high enough to cover all potential fraud
  - Nested Plasma chains might increase timeouts even more
- Thoughts about Layer 2 in general?
  - Off-chain solutions allow testing new features and protocols easier
  - Trade-off between scalability and availability

That's all for today

- Conclusion:
  - Plasma chains allow off-chain payments and computation, similar to state channels
  - Ensuring availability in layer two solution is still not fully solved
- No lecture on 10/14
- Next time: Selfish-Mining