Abstract

Below we describe a mechanism to enable non-custodial payments in Proof-of-Stake sidechains. We call this mechanism Plasma Bank, as it inherits key concepts from Plasma Cash. Particularly, Plasma Bank uses a Sparse Merkle Tree within the validating client to keep track of balances in bank accounts off-chain.

Data Unavailability

In this scheme, verifiers sign-off on the validity and availability of a particular block.

Consensus: Token Incentivized Sidechain

We derive most of our security assumptions from a Token Incentivized Sidechain

which are backed by Proof-of-Stake consensus. Similarly to Token Curated Registries

in that they have application-specific tokens in which the sidechain verifiers participate in non-subjective voting to arrive at consensus. If Ether were to be used in the token's place, there would be to incentive, or penalty for verifiers to arrive at incorrect consensus. That is, that verifiers are incentivized to arrive at correct consensus to ensure that the value of their token assets increases due to trust in the sidechain. One token, one vote.

Consensus can be typical PBFT, Casper, Tendermint, etc. We currently use our simple on-chain Proof-of-Stake consensus called Andromeda.

Verifiers are also incentivized to run nodes by receiving transaction fees which can be any token, we suggest to use a stablecoin.

This concept is also somewhat similar to federated sidechains such as RSK and Liquid.

State and Balances

A set of verifiers operate verifiers node which communicate with each other through a peer-to-peer network. On the rootchain only a single 32-byte value is stored, which represents state and balances.

[

1124×744

](https://static.slab.com/prod/uploads/posts/images/oqJSrcPSF-fWrcCIzFMaRFVf.png)

We define the machine state root and a set of balance roots all within the same Merkle tree. The balances themselves use Sparse Merkle Trees to keep track of balances within a sidechain. A sidechain, at minimum, has balances for verifiers and its participants. In Token Incentivized Sidechains, participants pay transaction fees. Verifiers receive a portion of transactions fees collected per consensus round based on how many tokens they have at stake.

Event-driven State Machine

The sidechain operators run nodes which are simple event-driven state machines. There is specialized code that ingests input event signals and outputs event signals which continuously updates the state of a machine.

The state machine conditionally updates balances, which is part of its state. Balances are updated based on custom logic, so that balances are only updated when certain conditions is met.

[

1004×624

](https://static.slab.com/prod/uploads/posts/images/5pVDSkPzgsqzt1WqQS9c9oMJ.png)

Plasma Bank: Non-Custodial Payments

We use a 2-way peg to deposit and withdraw tokens from a rootchain smart contract. The underlying assumption is that a federated sidechain will not submit incorrect blocks because of Proof-of-Stake consensus driven by trusted, or at least

known, verifiers.

This is similar to RSK, Liquid and EOS, in that a lot of trust is given to the federation of verifiers.

Plasma Bank is a way of implementing 2-way pegs with a federated sidechain. First an ERC-20 token is deposited into the banking smart contract. Sidechain verifiers listen to events on the smart contract and updates their internal balances after a certain confirmation period.

In order to withdraw funds from the bank, a user must supply the proof of balance and ownership to the smart contract. The smart contract checks the latest valid consensus root of the sidechain and immediately releases funds to the user if the their proof is verified.

The only way that the federated sidechain can take control of funds, is if the majority of the sidechain verifiers becomes corrupt.

Conclusion

What we've described is the most basic mechanism of implementing non-custodial deposits and withdrawals on federated proof of stake sidechains. The idea of Plasma Bank is to create the most simplest mechanism for dealing with payments when there's some level trust in the verifiers of a sidechain.

Project 8 Framework

The reason we've chosen such simple semantics is to build a Sidechain Development Kit (SDK) to build plasma sidechains, starting with on-chain consensus and 2-way pegs for generalized smart contracts.

We're putting together a list of resources about our ideas and set of tools to develop plasma sidechains @ https://project8.build.

You can see an implementation of Plasma Bank here:https://github.com/luciditytech/token-incentivized-sidechains/blob/119541d7b1fc1c4d098791683228913a3e1740c8/contracts/PlasmaBank.sol