To start a Plasma chain, one need to take care of many issues. Using another blockchain can speed up this process with some additional benefits such as that chain's consensus and smart contracts. I want to discuss about potential issues arise in doing so. Please assume that all technical points here are discussed with Plasma Cash in mind.

What will change

In original design, the operator will responsible for creating blocks. Now, this job is done by miner/validator of Plasma chain, the job of the operator will be reduced. Another Plasma contract will be deployed to the Plasma chain:

- The operator will listen to events from both chains (deposit, exit, transfer) and interact with Plasma contracts.
- The client can even validate the honesty of the operator as they can access information in both chains.

However, the submitted hash will have to changed. Before, clients submit signed transactions to operator. Now, clients interact directly with contracts. Those transactions is much more complicated than the simple transaction format in original Plasma Cash.

A working solution

Clients have to submit the simple transaction as data to the contract in Plasma chain. This approach is functional, as the operator can simply take the data of transactions and calculate hashes. Actually, anyone can calculate hashes to verify the honesty of the operator. The limitation of this approach is the additional data has to be submitted. Even if clients use some other services, if those services work with Plasma tokens, they have to ask clients to sign those additional data.

- What will happen if a client sign a transaction but that doesn't get included in any block, but still submitted to contract by the operator? In this case, the receiver can exit, but not the sender. This is similar to the Limbo exit in Plasma Cash.
- What will happen if a transaction got included in a block but not in the submitted hash? In this case, the sender must exit, as there is no way for the receiver to exit the token. It can still be spent as a token in the Plasma chain, but clients should actively reject transaction with those tokens.

I'm thinking about how to punish the operator in these case. As every client can validate the result, I think it is possible to hold a vote for the honesty of the operator?

Security of Plasma chain

1. If the Plasma chain has no fork, i.e. 1-block irreversibility

Assume the previous discussed method works, this case is simple and not much different from Plasma Cash. The job of the operator are:

- Watch for Deposit events in Ethereum and mint tokens in Plasma chain.
- Watch for Transfer events in Plasma chain, calculate hashes and submit to Ethereum.
- Watch for Exit events in Ethereum and burn the corresponding tokens in Plasma chain.
- If the Plasma chain can be forked

If the Plasma chain uses POW or POS, a block might be reversed. It is similar to the issue discussed in previous section. Consider an example:

- Block x

: Alice swap token a

for some amount of the chain's currency C

of Bob.

- Hash of block x

is calculated and submitted to Ethereum.

- Block x

got replaced by another block y

which does not have the transaction of token a

.

- Bob exits token a

, it is a valid exit.

- The operator burns token a

, which owner is Alice. Alice also lost those C

.

If Alice and Bob are not cooperating, Alice will lose token a

. In original design, all tokens are generated from Ethereum before being deposited to Plasma chain. So in case of such attack, the transaction got reserved and the original owner can exit these tokens. But this is another blockchain, so there is no guarantee in that case. As a result, users should be warned to only trade with tokens backed by Plasma

. On a side note, atomic crosschain swap with Ethereum still works well in the example above.

In conclusion, I think this might be a worthwhile approach as it allows easier adoption of Plasma. However, there are still some issues that need to be take care of.