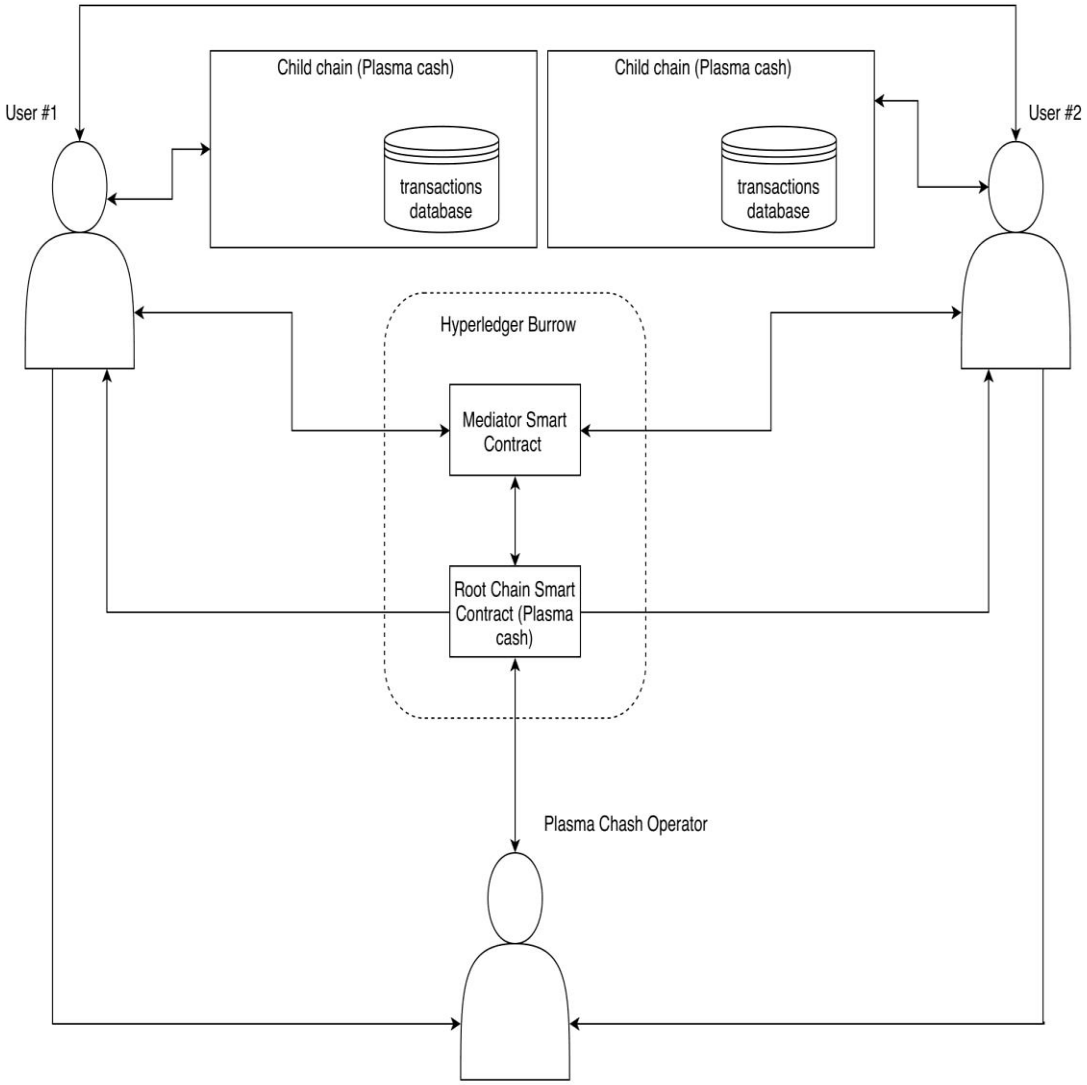

Plasma Cash

- Hyperledger Burrow - main internal blockchain with support EVM
- Mediator Smart Contract – smart contract which serves as a buffer between the user and Plasma Cash. Contracts of type like that can be multiple to avoid overflow storage. Its responsible for the follow task:
    - Token withdrawing status to make sure for adding tokens in Plasma Cash enough funds for entering
    - Lock users tokens before he/she don’t withdraw it from Plasma Cash
    - Organize safety token withdrawing from Plasma Cash
- Root Chain Smart Contract (Plasma cash) - Plasma Cash Layer #1
- Child chain (Plasma cash) - application on user device + local database which store transaction information (and another related data) definite user
- User #1, User #2 – users of Atmosphere

- Plasma Cash Operator – operator of Plasma cash which accept transactions from users, validate and create new blocks in Root Chain Plasma Cash

Flow

Standard token adding in Plasma Cash:

1) User #1 делает makes an input request in Plasma Cash(for example, with ID = ABC) through special function in Mediator Smart Contract.
2) Mediator Smart Contract:
    a) Validate token balances with ID = ABC which belong to User #1
    b) If enough tokens on the account, then it transfers these tokens to a specialized buffer account, from this account tokens can only output
    c) Call Deposit в Root Chain Smart Contract function
3) Function Deposit in Root Chain Smart Contract immediately create new block with new one transaction
4) User #1 save in his/her local database, which has been Child chain (Plasma cash), information about transaction.
5) User #1 ready to token exchange with another participants of Plasma cash

Standard token withdrawing from Plasma Cash:

1) User #1 call startExit function in Mediator Smart Contract, where send validate proof of his/her balance in Plasma Cash.
2) Mediator Smart Contract sent a part of this information in startExit Root Chain Smart Contract function.
3) Root Chain Smart Contract validate withdrawing funds and back result in Mediator Smart Contract.
4) Mediator Smart Contract make token transfer to User #1 balance

Standard transfer between User #1 and User #2

1) User #1 through Child chain (Plasma cash) want create token transfer of a certain type (next, just token) with User #2, User #1 create transaction and sign it his/her private key.
2) User #1 provide proof his/her token balance with transactions to User #2
3) User #2 validate proof with Root Chain Smart Contract and information from his/ her local database (in the event if he’s already has on the balance of Plasma cash tokens of this type, then this information is in the user's database)
4) If proof is right, User #2 call about it to User #1
5) User #1 send transaction to Plasma Chash Operator
6) User #1 send information about it to User #2
7) Plasma Cash Operator once again validates the transaction and writes it to the block Plasma cash
8) Plasma Cash Operator validate block and record information in Hyperledger Burrow
9) Transfer is ended

Estimation

Plasma Cash with elements of Plasma XT:
- Root Chain Smart Contract (Solidity Smart contract) – **340 hours.**
- Mediator Smart Contract (Solidity Smart contract) - **80 hours.**
- Child chain (Golang) include work with checkpoints - **300 hours.**
- Plasma Cash Operator back end (Golang) – it's part for interaction with Child Chain. Include work with checkpoints from operators side and automate validation and block forming systems and database system to storage all transaction- **390 hours**

Total:**1110 hours**