

Different Tastes of Ethereum Plasma

ARTEM VOROBEV




About Me

- Developing Software for 12+ Years
- Team Lead at BANKEX Foundation
- Blockchain developer

Work On

- Ethereum Plasma
- BUTTON Wallet

BANKEX Foundation

- R&D Team, Plasma 
- Web3Swift 
- Educational courses & Hackathons
- Open Source 

BUTTON Wallet

- Crypto **wallet** right **inside Telegram Chat**, private keys are stored in QR code on the client side, supports **BTC, LTC, BTH, ETH, ETC, Waves**
-  Russians hackers startup from Silicon Valley  that won 10 hackathons  last year including ETH
- 80K Users, 500K USD Investment from VC

20\$ for the best plasma related question



buttonwallet.com

<https://t.me/buttonwalletbot>

[@buttonwalletbot](#)

Let's go back to Ethereum Plasma

- Solves Ethereum scalability problem (~15 TPS)
- **Level 2** scaling solution
- **It's not a payment channel** like Lightning network or Raiden
- **Side chain with centralized block production***
- Whitepaperer Joseph Poon & Vitalik Buterin
<https://plasma.io/plasma.pdf> August 11, 2017

Important to understand

- Plasma in general is a **protocol** i.e. concept that at research state right now
- All the ideas live at the ethresear.ch research. The whitepaper is not enough to get a grasp on the plasma. Some places are obsolete

R&D

- A lot of plasma implementation was released last year, and much more are coming
- A lot of R&D done by [plasma implementers group](#) and ethereum developers community

Key components

- **Plasma Smart Contract**

Bridge between main net and side-chain. Gives security guarantees for the users

- **Plasma Operator**

Centralized block producer that assemble blocks side chain with a speed of lite and publish block headers to the Smart Contract.

- **Client app & Plasma Validator**

Deposits, transfers, exits and withdraw on plasma side-chain. As validator constantly stay online and monitor side chain blocks, ready to start exit if the operator makes a double spend, block withholding by any reason.

What plasma can do

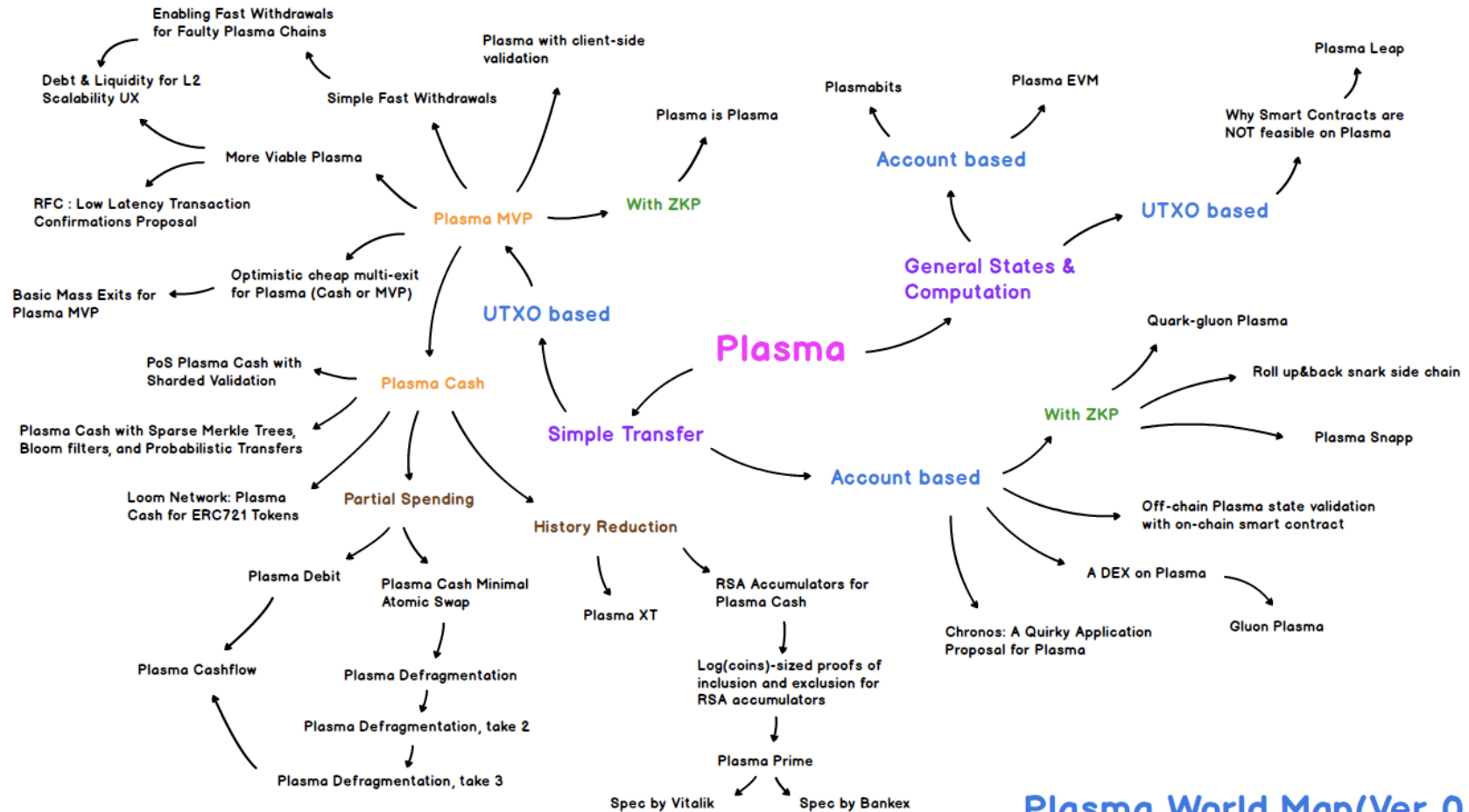
- Speedup Ethereum to the speed of light or at least 50 - 1000 times
- Tremendously reduce a transaction price $\sim 0.0001\$$
- Support fungible ERC20 and non-fungible ERC721 deposits and exchange
- Potentially suites for any Blockchain with turning complete smart contracts, smart contracts inside of side-chain...

With one clarification - "it depends"

- Speedup Ethereum to to speed of light or at least 50 - 1000 times ***depends on plasma design***
- Tremendously reduce a transaction price $\sim 0.0001\$$ ***depends on plasma design***
- Support fungible ERC20 and non-fungible ERC721 deposits and exchange ***depends on plasma design***
- Potentially suites for any Blockchaint with turing compleat smart contracts ***it depends on***



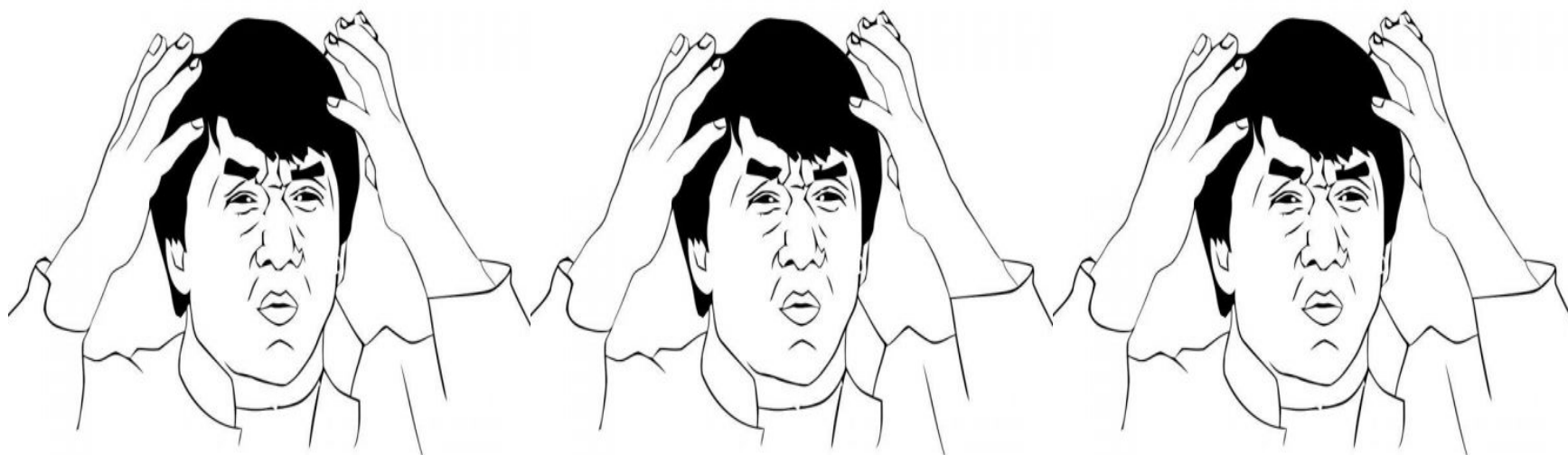
Plasma World Map



Plasma World Map(Ver 0.1)

Created by Aiden (aiden.p@onther.io)

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



UTXO vs Account Based Plasma

- UTXO based
 - Like bitcoin
 -
- Account based
 - Like Ethereum
 - Uses verifiable computations for state change mutation verification (zero-knowledge proofs)
 -

UTXO based

- **Merkle Trees** Operator publish root hash on the main net
- **Growing history problem** Validator should keep full history, to win exit game. Since speed is high history growing tremendously. Can be solved by history compression based on RSA accumulators or zkSNARK/zkSTARK.
- **Some one need to stay online** and checks what operator publish, to trigger own exits and challenge exits of other participants







Account based, e.g. rollup

- **Requires zero-knowledge proofs to change state** Smart contract accepts only valid 'state transition' by checking proof that operator publish. The operator can't publish wrong blocks. Checking is cheap and can be done on smart contract
- **Requires extensive computation on operator** It's hard to generate zero-knowledge proof
- **Requires trusted setup** (for zkSNARKS). Ethereum can provide required features.
- **Consume more space in the block on the mainnet** Operator also publishes transaction fingerprint.

Plasma state of the art:

- More Viable Plasma
- Plasma Cash
- Plasma Prime, e.g. Plasma Cashflow with history reduction on RSA accumulators
- Plasma Cashflow with history reduction on zkSNARKS
- Account based zkSNARKs Plasma

Plasma state of the art:

- More Viable Plasma 
- Plasma Cash 
- Plasma Prime, e.g. Plasma Cashflow with history reduction on RSA accumulators 
- Plasma Cashflow with history reduction on zkSNARKS 
- Account based zkSNARKs Plasma  

More Viable Plasma

- Fix minimal viable plasma vulnerabilities 👍
- 2 weeks time for an exit, requires liquidity market to fix that 👎

🚫 Showstopper:

History is growing too fast, in one year it's not possible to store it on the server

- It requires ~1 Kb per transaction.
- 100k transaction/second ~ 2 Petabyte per year

Plasma Cash

- Easy to implement 👍
- User track only his own coin 👍
- Doesn't have minimal viable plasma vulnerability 👍
- Suits well only non-fungible assets e.g ERC721 tokens 👎

🚫 Showstopper:

History is growing too fast; it's not possible to store the history on the light client, e.g. mobile client

- 2.5 Gb for a single coin - over the year [history reduction on ethresearch](#)
- 20 Gb for the 8 coins

Plasma Prime - Cashflow with history reduction using RSA Accumulators

- Slices instead of unique coins(plasma cash), similar to plasma debit
- End-user needs to observe only slices that he owns
- The introduced idea of history compression by generation proof of exclusion of slice to a range of blocks
 1) Assign a prime number to slice
 2) When ever slice modification assign a new prime number to slice and put it into the accumulator.
- Proof of exclusion requires less space than full history
- Requires to expensive computations
 After implementing a proof of concept, BANKEX Foundation research team find out that existing RSA Accumulator is a bottleneck that we can't pass at the moment. ❄️

Plasma Cashflow with history reduction using zkSNARKS

- The same idea as Plasma Prime, we fix a problem, by replacing RSA Accumulator with zkSNARK proof of exclusion
- Working solution that fixes the problem of extensively growing history. 👍
- The user should only store the compressed history of his own coins. About 1 - 10 Mb for the small amount of Ether. 👍
- Doesn't have known critical issues that previous designs had 👍
- All the components are known, key research has been completed [1, 2, 3] 👍

Proof of concept designed on the ETH Singapore by BANKEX Foundation ([Github](#))

Account based zkSNARKs Plasma e.g rollup

- Verifiable computation and account-based model can potentially bring features that other plasma can't, e.g. order book for the exchange 👍
- No need to stay online to validate blocks 👍
- Requires a lot of Gas to publish blocks 👎
- About 20 TPS 🔥 on a laptop ~300 - 1500 TPS on cluster 👎

Proof of concept is implemented by barryWhiteHat ([Github](#))

6th of January 2019 alpha version of Ignis Wallet published on the testnet by [Matter.Inc](#) ([Wallet on testnet](#))

Thanks



Chuck likes Plasma!

Questions

BUTTON Wallet with 20\$ in ETH for the best question.



Take a picture of QR, than import it to @buttonwalletbot in telegram

That presentaion on the Github



Done with [Marp](#), amazing markdown to presentation writer