OpenST Mosaic

A FRAMEWORK TO SCALE ETHEREUM

Benjamin Bollen, Martin Schenck
for OpenST Foundation

Abstract

We present improvements to the OpenST protocol. OpenST is a framework powered by Ethereum to build token economies. We lay out in detail two contributions, OpenST Mosaic and OpenST Gateway, which work together to scale Ethereum. OpenST Mosaic is a layer-2

1. Introduction

Ethereum [1] with its current proof of work Nakamoto consensus protocol is inherently limited in the number of transactions it can perform per second [2]. In order for Ethereum to gain mass adoption, an increase in throughput is required. Current efforts include state channels [3] like the Raiden Network [4] and side chains [5] like Plasma [6]. A drawback of the proposed solutions is the fact that the user has to be always online in order to ensure integrity of her transactions. Furthermore, in case of a mass exit, a congested Ethereum network could lead to lost funds.

OpenST Mosaic is a holistic solution powered by Ethereum that we expect to securely scale Ethereum to thousands of transactions per second. We achieve that by verifiably finalising batches of sidechain transactions on Ethereum.

2. Related Work

Verifiers' Dilemma [2]

Interblockchain Communication [7]

Casper FFG [8]

3. Our Contribution

4. OpenST-Mosaic

OpenST Mosaic is not conceptually bound to Ethereum. As layer one we assume a blockchain that has byzantine fault tolerance [9] and plausible liveliness [8]. We call that blockchain *Origin* $\mathcal{O}$. We chose Ethereum as Origin for our first implementation. Transactions will take place on a sidechain that we call *Auxiliary* $\mathcal{A}$. Origin and Auxiliary must both

We assume Origin to track all ownership. Origin only lends ownership to Auxiliary. Thus, finality on Origin is always authoritative. If Auxiliary halts, users can recover the assets they own and there is no restriction in time for them to do so, enabling a coordinated mass exit over time.

4.1. OpenST-Mosaic

Lemma 1. *Given a mapping $\mathcal{V}^{\mathcal{O}} \cong \mathcal{V}^{\mathcal{A}}$ on $\mathcal{O}$ there is a proof that any message signature by a validator $\nu^{\mathcal{A}}$ of $\mathcal{V}^{\mathcal{A}}$ on $\mathcal{A}$ belongs to the respective validator $\nu^{\mathcal{O}}$ of $\mathcal{V}^{\mathcal{O}}$ on $\mathcal{O}$.*

Lemma 2 (Remote Accountability)**.** *A vote on $\mathcal{A}$ that violates a Casper Commandment can be punished on $\mathcal{O}$.*

Proof. Given lemma 1, a violation of either Casper Commandment can be proven on $\mathcal{O}$ without the respective transaction residing on $\mathcal{O}$. If a validator $\nu^{\mathcal{A}}$ published two distinct votes for the same target height, the relevant votes can be presented on $\mathcal{O}$ and the signature can be validated. An offending validator $\nu^{\mathcal{O}}$ can be identified according to lemma 1 and its deposit slashed. The same is true if a validator $\nu^{\mathcal{A}}$ votes within the span of its other votes. All relevant votes including their signatures can be presented on $\mathcal{O}$. □

Theorem 3 (Partial View)**.** *Given Origin $\mathcal{O}$ with blocks $b_i^{\mathcal{O}}$, checkpoints $c_i^{\mathcal{O}}$ can be reported on Auxiliary $\mathcal{A}$, such that a set of validators $\mathcal{V}^{\mathcal{O}}$ with stake on $\mathcal{O}$ can reach finality about the reported checkpoints $c_i^{\mathcal{O}}$ on $\mathcal{A}$ as $\mathcal{V}^{\mathcal{A}}$ with accountable safety enforced on $\mathcal{O}$.*

Proof. Accountable safety is enforced according to theorem 2.

□

Theorem 4 (Leveraged Security)**.** *Proof.* Lemma 1. □

Theorem 5. *Proof.* □

4.2. OpenST-Gateway

OpenST Gateway enables chain-to-chain transfer of state objects.

4.3. Reward Structure

On Origin in OST.

rewarding for reported block headers that get finalised

4.4. Dynamic Set of Validators

4.5. Set-Up of An Auxiliary Chain

Start running with gas cost of 0. Set up contract with base token. Set up Gateways, but closed. Give one week of blaming on Origin. How can there be proof on Origin without Aux's state root? Or is that also transferred in the set-up phase? If so: why is it trusted? Open Gateways.

5. Outlook

5.1. Token Economies

5.2. Neo and Cardano

6. Conclusion

References

[1] Wood, G. Ethereum: A secure decentralised generalised transaction ledger (2015). URL `http://gavwood.com/paper.pdf`.

[2] Luu, L., Teutsch, J., Kulkarni, R. & Saxena, P. Demystifying incentives in the consensus computer. *IACR Cryptology ePrint Archive* **2015**, 702 (2015).

[3] Poon, J. & Dryja, T. Lightning network (2015). URL `https://lightning.network/lightning-network-paper.pdf`.

[4] Raiden network. URL `https://raiden.network/`.

[5] Back, A. *et al.* Enabling blockchain innovations with pegged sidechains (2014). URL `https://www.blockstream.com/sidechains.pdf`.

[6] Poon, J. & Buterin, V. Plasma: Scalable autonomous smart contracts (2017). URL `https://plasma.io/plasma.pdf`.

[7] Kwon, J. & Buchman, E. Cosmos: A network of distributed ledgers (2016). URL `https://github.com/cosmos/cosmos/blob/master/WHITEPAPER.md`.

[8] Buterin, V. & Griffith, V. Casper the friendly finality gadget. *CoRR* **abs/1710.09437** (2017). URL `http://arxiv.org/abs/1710.09437`. `1710.09437`.

[9] Castro, M., Liskov, B. & et. al. Practical byzantine fault tolerance. In Leach, P. J. & Seltzer, M. (eds.) *Proceedings of the Third Symposium on Operating Systems Design and Implementation*, vol. 99, 173–186 (1999).