OpenST Protocol White Paper

Benjamin Bollen, Nishith Shah, Lionello Lunesu, Sunil Khedar, Antoine Cote,
Jason Banks, Jason Goldberg, Matt Chwierut, Brian Lio

Draft published for peer review* v0.8.3, last updated 5 November 2017

1 Introduction

Ethereum introduced to the blockchain toolset stateful accounts. The storage space associated with these accounts is write-protected by code, and we refer to these accounts as smart contracts[1] This general purpose capability of Ethereum has spurred a vast wave of innovations and a leading use-case of Ethereum has been the ability to create a token on top of Ethereum. For tokens on Ethereum, ERC20[2] has recently been adopted as the standard interface for a smart contract defining such a token. However in order to engineer utility into a token to incent greater uptake, several more challenges become apparent. Some of these challenges include latency, transaction fees, scale and privacy; for these challenges we attempt to contribute towards solutions in this technical whitepaper. Other challenges come from legal requirements and economic modeling to support a token economy; confronting these forms an integral part of the OST project[3].

With Simple Token we set out to be pragmatic about the capabilities of existing decentralization technologies and look to find answers that solve for these problems today with a clear roadmap towards internet-scale performance. All the while we strive that all technical mechanisms are open and independently cryptographically verifiable.

OpenST operates as a non-profit and governs the development of the OpenST Protocol. In addition it performs high-level guardian tasks on the instance of the protocol that is associated with the OST EIP20 token on public Ethereum. These guardian tasks are limited but necessary when a technical answer only is insufficient. A primary example can be the review of a new member company which desires to launch its own branded token within the OpenST platform.

First we establish a lexicon to help present the new patterns OpenST Protocol introduces in the token space. For a young token economy, speculative value of a token can easily drown out the intended utility. We describe how a utility token can

[*] for review contact review@simpletoken.org

[1] We use smart contract interchangeably for the code associated with an account, or the stateful instantiation of that code; the meaning should be clear from the context.

[2] See EIP20 (formerly ERC20)

[3] Find our thinking on governance, economic modeling and incentives in OST sidepapers.

be built on top of value assets that back it.

We introduce Simple Token as a freely tradable EIP20 token on Ethereum mainnet. OST can be staked as a valuable crypto-asset to mint utility tokens. Furthermore Simple Token functions as the base token on the utility chains accounting for gas consumption.

We continue to outline how desired behavior can be enforced on the utility tokens so that the token serves the user transactions within an existing consumer application. We call such utility tokens branded tokens. We describe how user financial sovereignty is preserved on the blockchain as a user can choose to hard-exit the value of the branded tokens on Ethereum mainnet.

In the next section of the protocol we describe how rich interactions in consumer applications can be mapped to fundamental transactions on the blockchain by providing an API for developers to integrate OpenST into their application - making the development of a consumer tokenized economy simple.

While these last mechanisms are off-chain mechanisms, we consider them an integral part of the protocol to enable any third-party developer to build on top of the OpenST Platform. Additionally including them in the protocol enables open innovation and audits to realize best practices for minimizing correlatable data on chain of pseudo-anonymous accounts that could empower inferring personally identifiable data or application metrics. Future work in this part of the protocol layer as such includes technology to increase the noise-to-signal ratio on-chain, or reduce the signal by moving it off-chain with payment channels.

To conclude we describe how no trust needs to be placed in the validator pool of the utility chain, as in the case of chain halting, all ownership state can be carried back to the value chain.

We recapitulate the protocol at a high level using an explicit example and build on this example to illustrate how an end-user can use OSTSimple Token on Ethereum to obtain and interact with a branded token.

We describe how OpenST Platform can be an open network of utility chains, serving different consumer applications and provided by third-parties.

While the OpenST Protocol and Platform concerns only application logic - and OpenST will focus on implementing it as Ethereum smart contracts - and off-chain technology, it is still worthwhile to detail our considerations with respect to the available chain technology upon which the OpenST Platform can execute. We discuss these architectural requirements lastly and how OpenST can contribute to existing projects in this area. This concludes the OpenST Platform as the second part of this paper.

Lastly we put forward the roadmap for OpenST.

2 Related Work