Hi Everyone,

We (Andreas Freund, Dan Shaw, and Tas Dienes) are working to understand how the Ethereum Foundation can better support the Layer 2 ecosystem and ensure its health and success going forward.

Over the course of the last couple of months and in various conversations with people in the L2 ecosystem, enterprises, and our own research the subject of L2 Interoperability has come up again and again as a concern with regard to “Balkanization” of the L2 ecosystem into islands rather than a cohesive continent, especially from enterprises fearing vendor-lock-in.

In this note, we are attempting to define the term “L2 Interoperability” and what it entails in terms of characteristics. While attempting to be as comprehensive as possible, we are neither claiming correctness nor completeness. This is merely a “strawman” to begin a robust discussion about L2 Interoperability, hopefully leading to an L2 Interoperability standard that can drive broader adoption and more value creation.

L2 Interoperability Definition: Layer 2 Solution A (L2S-A) and Layer 2 Solution B (L2S-B) are said to be interoperable if Alice, or one of her delegates, can transfer one or more fungible or non-fungible assets from L2S-A to Bob, or one of his delegates, on L2S-B without having to exit the asset(s) to the underlying blockchain(s) and in an, ideally censorship-resistant, manner such that neither Alice, nor Bob, nor one of their delegates can double-spend the asset(s) at any point before, during and after the asset transfer on either the participating L2Ss or the (permissionless) Blockchains used by either L2Ss.

Note, that the above definition refers to East-West interoperability between L2Ss. In other words the ability of Alice in L2S A to functionally interact with Bob in L2S B, such as doing a trade. We defer North-South interoperability, in other words, the interoperability of components such as transaction processing or storage within an L2S, or the interoperability of an L2S and an L1 to a later post. We also defer the discussion of “Wallet Interoperability” to a later post since East-West interoperability as defined below is a prerequisite to defining the relevant wallet interfaces for standardized L2S objects required for East-West interoperability. Anyone interested in diving deeper into the question of wallet interoperability should look at the principles laid out in the W3C Universal Wallet draft specification.

Also note that we define a Layer 2 solution as a secondary framework or protocol that is built on top of an existing (permissionless) blockchain, that has the same security assurances as the underlying blockchain, that has the ability to arbitrate and resolve L2 state disputes on the underlying blockchain and that significantly increases the transaction throughput of the underlying blockchain while simultaneously, significantly reducing the transaction data footprint.

We suggest that an East-West interoperability solution has but is not limited to the following characteristics in no particular order:

- Resolvable, public key controlled identifiers for all participants following established standards such as W3C DIDs to enable portability across L2Ss. Resolvable refers to the identifier being used to discover its controlling keys and other control, authorization, and service attributes of an identifier. For example, an Ethereum address can function as a resolvable public key controlled identifier. See for example the specification of an Ethereum based DID method (did:ethr).

- Discoverable authentication/authorization capabilities based on common, well-established frameworks such as OAuth2, OpenIDConnect (OIDC), SIOP DID AuthN (OIDC compatible DID auth) to avoid reinventing the wheel and also signaling openness to enterprises.

- Discoverable and negotiable services such as Authentication and Authorization endpoints, a price oracle endpoint etc. In this context, a service has consumers, providers, input and output parameters, and associated business logic that transforms the input into the output parameters, and where discoverable in this context means that a Service Consumer can find a Service provider and what the capabilities of the service are as well as the required input and output parameters, and where negotiable in this context means that the Service Consumer and Provider can negotiate how the service is delivered. This is required to be able to automate interoperability processes between L2Ss.

- Bi-directional and mono-directional services where bi-directional services in this context refer to direct and either synchronous or asynchronous service-consumer-to-service-provider or vice versa asset transfer via APIs and where mono-directional in this context refers to services that either extract assets from or deliver assets to an L2S or asset via APIs. Alice and Bob might want to exchange assets across L2Ss or just extract and keep their assets locked and in their wallet before committing to another L2S.

- Standardized set of APIs such as REST representing common asset functionalities and consisting of payloads with

- defined API endpoint functionalities such as transfer, lock, unlock, exit, deliver, swap

- standardized API envelopes consisting of for example L2S origin and target metadata, security parameters etc.

- standardized asset payloads describing the assets and their current state; a current state can consist of for example (zero-knowledge) proof sets (asset-history, asset-locks, asset-state) and asset description can consist of for example asset type, asset ID, asset owner, anchor contract(s) for proof verification(s)

- See also related work here.

- defined API endpoint functionalities such as transfer, lock, unlock, exit, deliver, swap
- standardized API envelopes consisting of for example L2S origin and target metadata, security parameters etc.
- standardized asset payloads describing the assets and their current state; a current state can consist of for example (zero-knowledge) proof sets (asset-history, asset-locks, asset-state) and asset description can consist of for example asset type, asset ID, asset owner, anchor contract(s) for proof verification(s)
- See also related work here.

are critical to be able to have the same “words and grammar” to be able to talk to one

another.

- Discoverable Standard Transport security such as JOSE with JWS/JWE or DID Comm is critical to ensure security and privacy at all times and beyond HTTPS. See also related work here.

We are very much looking forward to hearing your thoughts on the above definitions and characteristics to spur discussion about how we could set out to formulate, at least some standards, and what would be the priorities to implement such standards.

Subsequently, and based on the discussions about the above we will want to explore the required tooling around L2 Interoperability such as wallets capable of managing interoperable assets through standardized interfaces and able to integrate with L2S using a set of standardized interfaces, L2 explorers etc.

All the best,

Andreas, Dan, and Tas.

cc @tasd