Following our framework of Aave's infrastructure evaluation and the positive signaling by the community onSnapshot, we present the analysis of the zkSync Era network, regarding its suitability to deploy an instance of the Aave v3 (v3.1) protocol.

DISCLOSURE.

This is an independent assessment of different technical components that we consider important for the Aave software to run optimally, not a categorical analysis stating the network is "good" or "bad", and no kind of "requirement" for Aave to be deployed on the candidate network. That decision is up to the Aave governance, no matter our opinion.

In addition, currently, we have absolutely no financial/investment/services-engagement/any kind of interest in the zkSync ecosystem.

When doing the evaluation, we contacted the zkSync team (Matter Labs) as an important source of information, which has always been exemplary and supportive, but everything in this report comes finally and exclusively from our independent criteria.

Report

1. Introduction to zkSync

zkSync Era is a Layer 2 ZK rollup, using validity proofs to have low cost transactions, but inheriting the security of the Layer 1, Ethereum.

Other high-level characteristics of zkSync are:

- Implementing its own virtual machine execution environment (zkSync VM), but majorly EVM compatible.
- Native account abstraction and paymasters (allowing to pay "gas" with different assets).
- State updates based on state diffs versus transaction inputs on other rollups.
- The validity proofs system is based on a combination of STARK and SNARK proofs.

zkSync Era was released (mainnet) onMarch 24th, 2023.

2. Our methodology

This report is not trying to be a full analysis of the zkSync Era network, instead focusing on the aspects important to the Aave community should it decide to deploy the Aave v3 (v3.1) liquidity protocol there.

In addition to being extensive, this report tries to be simple enough for all participants in Aave governance to understand. But given its technical nature, it is unavoidable to assume a certain familiarity with some relevant concepts, such as rollups, oracles, RPC nodes, or blockchain explorers, amongst others.

In order to simplify the interpretation of this report, we will evaluate each key component for Aave separately, and assign simplified "grades", defined as follow:

Optimal

. Fulfilling all minimal requirements, and with extra positive aspects.

Good

. Fulfilling the requirements, but improvements can be made.

Acceptable

. Fulfilling the requirements, but with "buts".

Needs improvement

. Mandatory requirements not fulfilled at all. Aave will not work properly

3. Evaluation

3.1. Oracle Infrastructure

The Aave protocol uses different types of oracles in a standalone blockchain. We consider that having Chainlink providing oracles, especially for prices, is a must for any deployment, with alternatives only considered on an ad-hoc, given the

additional complexity added on evaluation/integration.

3.1.1. Price feeds

Used by the Aave protocol to price listed assets

zkSync Era HAS

Chainlink price feeds as of now.

There are other oracle providers, like Pyth, but we would only recommend launching with Chainlink oracles.

3.1.2. L2 Sequencer Uptime feed

A “flag” parameter indicating in real-time to the Aave protocol if the sequencer of a rollup (or any network involving some centralisation on sequencing of transactions) is properly running.

zkSync DOESN’T HAVE

an oracle for fetching L2 Sequencer Uptime, but we have confirmed with the Chainlink team that the feed will be available in the following weeks.

3.1.3. Proof-of-Reserve feeds

Component indicating to the Aave protocol if the reserves backing an asset are healthy, for example in bridged tokens.

zkSync Era DOES NOT HAVE

proof-of-reserve feeds.

3.2. Blockchain explorer

For Aave, same as for any other blockchain project, block explorers like Etherscan are a fundamental component, specifically for the following:

1. Verifiable smart contracts code visualization.
2. Read/write interface with smart contracts.
3. Basic data analysis tool for misc aspects like token holders.

zkSync Era has multiple block explorers, including but not limited to:

- https://era.zksync.network/ powered by Etherscan technology.
- A blockscout instance https://zksync.blockscout.com/.
- An official explorer maintained by Matter Labs https://explorer.zksync.io/.
- Phalcon Explorer.

3.3. Compatibility with Ethereum RPC standard

Basic compatibility with the Ethereum nodes RPC de-facto standard (eth_, web3_

) is quite an important requirement for Aave or any other protocol, given that it helps to have tools built for Ethereum (or other similar networks) working out-of-the-box just by plugging them to a node, of zkSync Era in this case.

zkSync Era HAS

major JSON-RPC compatibility, only with some differences that should not affect the Aave v3 protocol.

3.4 Compatibility with Ethereum account format (addresses)

One of the strengths of non-Ethereum networks (e.g. Polygon, Avalanche C-Chain, etc) is its compatibility with Ethereum private/public keys of accounts. This allows existing account holders on those networks to use the others without creating an ad-hoc wallet for it.

A significant strength of non-Ethereum networks is supporting the same account derivation system Ethereum, so EVM wallets are natively compatible.

zkSync Era is fully compatible with the Ethereum public-private key account format.

3.5. RPC public endpoints and providers

Basic and reliable public RPC infrastructure is a must for Aave, as it is the way to connect to the network, both for data reading and transaction submission.

Currently, besides the public RPC hosted by Matter Labs (https://mainnet.era.zksync.io/), zkSync Era has support from numerous RPC node providers, including but not limited to:

- Alchemy
- Quicknode
- Blast
- DRPC

3.6. Custom behaviour (lack of) of the execution layer

Whenever a network has custom/extended behavior with respect to Ethereum, it is important to be aware of it and evaluate if it has any impact on the Aave protocol.

Examples of this potential behavior are the presence of new pre-compiles (compared with Ethereum), EVM opcodes, native account abstraction/meta-transactions or chainId definition out of the norm.

zkSync Era has numerous differences to Ethereum and its various sidechains. Some more relevant ones are listed below:

- Contract deployments use a specific system contract, which has for consequence that using the CREATE and CREATE2 opcodes is only possible if the bytecode to be deployed is known at compile-time.

This required some adaptation in the Aave v3 peripheral infrastructure (e.g. deployment orchestration), but doesn't cause any blocker.

- zkSync has their own implementation of account abstraction, which could be used to improve user experience on Aave. Certain functionality on Aave (e.g. permit()

on aTokens or meta-transaction support on the Aave Pool) will only work for EOAs, but this is not a problem, as "smart accounts" can have the same functionality natively to zkSync.

- Memory is handled differently, being allocated byte-per-byte, this could have an impact on inline assembly that expects memory to be allocated on a word (32 bytes)-by-word basis, but this should not have an impact on Aave.
- Historically, when fetching the block number or timestamp, the zkSync EVM was returning the L1 batch block number and timestamp rather than the L2's. However this behaviour has changed since a previous upgrade and now block.number

, block.timestamp

and blockhash

work as expected, returning directly L2-related data.

- zkSync Era contracts handle immutables differently by using an immutable simulator contract. This should not affect Aave.
- zkSync handles gas differently, including a fee for publishing data. This should not affect Aave, apart from having variations on gas cost compared with other networks where the protocol lives.

In addition to the previous, there are other technical differences between zkSync and Ethereum (e.g. lack of some precompiles, but addition of others), but from our analysis we don't see any impact on Aave v3.

3.7. Support of wallet providers

Wallet products like Metamask, Ledger, Coinbase Wallet, and others, are fundamental pieces of the infrastructure for users to access the Aave protocol. So it is a strong requirement for a network to be supported by a subset of them.

Given it is major EVM compatibility, zkSync is supported by the majority of chain-agnostic EVM wallets.

3.8. On-chain multi-signature infrastructure

The permissions on the Aave ecosystem are directly held by on-chain governance smart contracts.

However, different protection/emergency mechanisms, like the capability of canceling cross-chain governance proposals, or pausing an Aave asset/pool, depend on the Aave Guardian, which is capable of acting faster than the governance process.

Consequently, having on-chain multi-signature contracts is a requisite to have Aave on a different network, with a high preference for industry-standard tools like Gnosis Safe.

zkSync Era DOES HAVE

a deployed instance of the Gnosis Safe contracts.

3.9. Transactions simulation infrastructure (fork)

Lately, a really important development experience component is the ability to execute test transactions (simulations) on forked production networks.

A good part of the tooling around Aave depends on simulations by using different libraries/frameworks like Hardhat, Foundry, or Tenderly. This way, it is possible to rapidly prototype new developments, get extra assurances on governance proposals and protocol upgrades, change risk parameters, etc.

In terms of tooling that can be used for simulation supporting zkSync Era:

- Tenderly doesn't support zkSync Era.
- Phalcon simulator doesn't support zkSync Era.
- Hardhat supports zkSync Era.
- Matter Labs have been working on Foundry support of zkSync Era, currently in pretty advanced stage. More specifically, Aave contracts have been extensively using for testing purposes of the tooling.

3.10. Chain data/indexing solutions

For different projects and entities integrating Aave, and even if not a blocker for deployment, it is important that solutions like TheGraph or Dune are operating on the candidate network, to avoid building from scratch data pipelines.

zkSync Era is supported on multiple data-indexing solutions, including both limited to Dune or TheGraph.

3.11. Bridging infrastructure: assets, messages

Given the central role of Ethereum in the DeFi and Aave ecosystems, bridging infrastructure to/from is a must for any candidate network.

The zkSync network has proper bridge infrastructure for both transferring assets and generic messaging With respect to the functionality of Governance V3 and a.DI, their native bridge supports cross-chain messaging, and other bridges such as LayerZero, Celer, Orbiter, and RhinoFi.

For bridging assets, as with other chain/rollup bridges, this bridge will deploy a standard upgradeable ERC20 on the first bridging of an asset, inheriting its logic from the OZ ERC20 Upgradeable.

3.12. Commitment in security-incidents

Having proper mechanisms and procedures to prevent and react to security incidents is something quite fundamental for any platform and application, and rollups like zkSync Era are no exception.

From our research and communications with the Matter Labs team:

- There is an ongoing program on Immunefi for zkSync Era with a maximum payout of 1.1 M$.
- A private channel of communication will be kept open between the zkSync team and the assigned technical team of the Aave community (e.g. BGD), for any necessary update concerning the network and consequently, the deployment of Aave on zkSync.

3.13. Network security/technical model

At the core of any candidate network analysis are its morphology (which type of network it is) and security model (which parties are involved in the control over the network; decentralization degree).

The zkSync Era network is a rollup based on validity proofs, inheriting the security of the Ethereum base layer. However, at the moment, there are the following considerations:

3.13.1. Sequencer and Proposer Downtime

Like all rollups, zkSync uses a sequencer, which is responsible for processing transactions on the L2 blockchain itself.

The sequencer, can not skip transactions queued from L1, but canstop processing them. Furthermore, only the whitelisted proposer can publish state roots on L1, so withdrawals would be impossible should it go down.

3.13.2. Custom virtual machine (zkSync VM)

The zkSync VM is quite different from the standard EVM at a low level, introducing another surface for problems. However, the zkSync VM has been worked on for multiple years, and the implementation high-level looks solid, surrounded by multiple different security procedures.

3.13.3. Upgradeability and control model (decentralisation)

Currently, the zkSync Era network still has meaningful centralisation, with a4-of-7 multi-sig having super-admin control on a sophisticated system allowing different types of upgrades (e.g. transparent ones for standard software updates, or “shadow” ones for being able to perform bug fixes in a private manner, without compromising user funds).

Given the early stage of the technology, this control is understandable, and relatively similar to other L2 networks.

zkSync has been working on a decentralized governance system, removing some of the centralisation point, with the initial phase planned to be live in the following month. All the details can be found HERE.

3.13.4. Security audits

zkSync Era has been submitted to multiple audits and security procedures, on all layers (L1, L2, cryptography).

All of them can be found HERE.

3.13.5. Transaction lifecycle

The zkSync documentation contains an extensive explanation of the transaction lifecycle on EraHERE, but to summarise:

1. Users submit their transactions to the L2 Sequencer. Additionally, they also have the possibility to submit them directly to L1 to force their inclusion.
2. Sequencer executes the transactions on L2, and afterwards forwards them to the prover components, creating proofs to submit to the L1 verifier.
3. The cryptographic proof is submitted to the verifier contract on L1, together with the state diff, updating the state of the rollup on Ethereum.

3.13.6. Data availability

As previously commented, zkSync Era currently uses Ethereum as data availability layer. That means that all required data to recreate the state of the rollup is published to Ethereum, including:

- State diffs, representing state changes on the L2 blockchain.
- L2 → L1 logs/messages, for communications between the two layers.
- L2 smart contracts bytecodes, for full visibility on which code actually causes the state diffs and cross-chain communication.

zkSync has made visible effort to facilitate the transparency of the network by providing tooling and documentation on how to reconstruct the L2 state HERE.

In summary, the security model of zkSync looks pretty solid, with very important resources allocated on its design, execution and continuous improvement.

All the technical documentation about zkSync Era can be foundHERE. Additionally, all system’s smart contracts addresses can be found HERE.

4. Summary

From our analysis, we conclude the zkSync Era is suitable for a deployment of the Aave ecosystem, and more specifically the Aave v3 (v3.1) protocol

.

Aspects the community should take into account on the activation vote should be:

- zkSync Era is at the moment in a pretty centralised side of the spectrum, but there are visible/meaningful steps being

taken to improve decentralisation.

- Months ago, zkSync was lacking compatibility with some of the Ethereum tooling, mainly on off-chain components. From our interactions with the Matter Labs team, we have observed a very significant technical effort from their side to improve tooling, and more specifically the support of Aave's, including pretty complex components like Foundry support on a de-facto pretty custom VM.
- We expect to have some technical maintenance overhead for Aave v3 zkSync during the initial months, given that the tooling surrounding it is continuously improving. However we believe this overhead should be acceptable in terms of resources consumption.