Akamon α: An Interoperability Solution for Cardano and Polygon

MELD Labs

May 2022

Abstract

This paper presents Akamon α, a native bridge between Cardano and Polygon. The objective is to leverage both chains' efficient infrastructure to bridge native assets between them. Akamon aims to be a community-driven bridge open for everyone, emphasizing decentralization and trustless custody. Nevertheless, we have to stay centralized in the first testnet launch and semi-decentralized in the upcoming launches towards the end of 2022. The goal is to make iterative improvements to collect experiences along the way. For instance, an early centralized testnet brings us practical UX, infrastructure, and operations foundations for us to focus on decentralization later on. We will propose our road to decentralization in an Akamon β paper to be released end of May. A β testnet launch will follow suit on June 5th 2022.

1 Introduction

Blockchain interoperability[7][8][1][4] is a relatively new field that is constantly evolving. People build Blockchain bridges to enable communication between different networks. Although most solutions share the core functionality of wrapping assets from one end to another, their architecture may vary significantly. With WBTC, we have a multi-institutional design with trustful custody, KYC, and AML. Projects like RenVM, on the other hand, aim for permissionless and trustless custody.

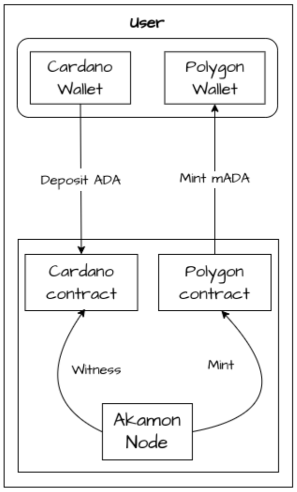
Akamon explores different architectures and techniques by starting with a bridge between an eUTXO[2] ledger in Cardano and an account-based one in Polygon[3]. We start from a fully centralized design and then decentralize it with each new design iteration. We further explore optimizations to prevent network congestion, improve decentralization, governance structure, settlement time, capital efficiency, dApp integration, and more.

This α white paper does not focus on technical depths as more key components will be introduced in β. Instead, it is planned to be a comprehensive introduction that provides insights to the users and all the moving pieces to the development team.

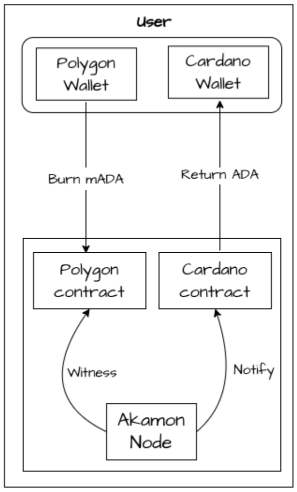
2 Akamon α

Akamon α is the first design iteration among the many we develop. We design the bridge iteratively to focus on a new fundamental concept with each iteration. α focuses on setting up the foundation for the later versions. We also aim to collect UX, infrastructure, and operations experience through it to have more space for decentralization designs in the later iterations. β expects to semi-decentralize the bridge, γ promises to bring more networks to the table, with δ seeking a fuller decentralization design.

The foundation for all iterations is to wrap assets between the chains. For example, an ADA round trip can be seen below.



ADA C->P : ADA on Cardano to mADA on Polygon



ADA P->C : mADA on Polygon to ADA on Cardano

Users first send ADA or mMATIC to an on-chain script to initiate a request. Sending ADA is to mint mADA on Polygon. Sending mMATIC is to burn them on Cardano to get back MATIC on Polygon. Confirmation then happens when validators sign to confirm the request. Once enough signatures are gathered, the users receive tokens on the other end of the bridge. The confirmation process should evolve with each iteration, from centralized (one signature) to fully decentralized. (community nodes).

Another foundation is that the bridge nodes should stake locked tokens to generate additional revenues. This incentivization is key to node operators and users, implying that the more traffic there is, the lower transaction fees.

Due to the iterative nature of the process, it is also essential to build migration frameworks to transition from one iteration to another seamlessly. This process includes both data and assets migration, which is not trivial to do right.

2.1 Wrapped Tokens

In α, we deploy centralized contracts for the wrapped tokens. We start with mADA and mMATIC and will soon extend to mMELD and more native tokens from both ecosystems.

On Cardano, the minting policy verifies that the minting transaction is signed by the bridge key with a simple `txSignedBy`. The on-chain script is defined in `Akamon.Cardano.Mint.Policy`. We then expose a `mintingPolicy :: PubKeyHash -> MintingPolicy` function to constructs the minting policy from a bridge key read from the environment variable `CARDANO_SIGNING_KEY`. This is better than reading both values from the bridge configuration, as a mismatch would lead to invalid minting transactions. A sloppy Cardano node setup would further risk the collateral of the minting transaction.

The policy id of the α `mMATIC` on the Cardano testnet (`1097911063`) is `f6f965943c738b48513277c1baf4770aa3873f2f624bfd2567322d4b`.

On Polygon, the token contract defined in `mADAToken.sol` verifies that the minting transaction is signed by the `MINTER_ROLE` and the burning transaction is signed by the `BURNER_ROLE`. Our deployment script in `3_deploy_v0.js` grants the bridge contract (defined in `AkamonBridge.sol`) these roles after its deployment. The Akamon node reads this bridge key from the `POLYGON_PRIVATE_KEY` environment variable.

The token contract of the α `mADA` on Mumbai is `0x222D66A3878772F1A3597cd8265dea569B4410A5`.

These α wrapped tokens will be abandoned from β, where the minting of the wrapped tokens is verified by a group of many more agents.

2.2 Akamon Core

Akamon Core is a Haskell library that defines the bridge's core types and utility functions. While we utilize standard tooling for on-chain code (Haskell and Plutus for Cardano, Solidity for Polygon), we use Haskell as the main programming language for the off-chain components.

For the time being, `akamon-core` includes token, address, transaction representations on both chains, time and status data types, testing utlity, and more. The key dependencies on Cardano are `cardano-api`, `cardano-ledger-core`, and `cardano-wallet-core`. The key one on Polygon is `web3-solidity`.

As we support more chains, we intend to port the actual representations to their packages like `akamon-cardano` and `akamon-polygon`. `akamon-core` then focuses on defining the generalized typeclasses for all the representations and utility.

2.3 Akamon DB

Akamon DB defines the database schema for a bridge node. We currently have an `Akamon.DB.Schema.Event` relation tracking users' minting and redeeming requests on both chains, and a `Akamon.DB.Schema.Fulfillment` relation tracking the bridge's fulfilment of those requests. Through these two, the bridge node listens to new transactions on both chains to insert new Akamon events and spins dedicated worker threads to fulfil them.

While the database schema can be SQL-generic, we currently use PostgreSQL as a strong standard choice. We spin a local instance through Docker for local development and testing and run an AWS RDS one in production. The sample Docker setup can be found in `docker-compose.yml`. An initialization script can be found at `init.pgsql`.

Note that this database setup only works for the centralized α as certain states are not recorded or quickly recovered from on-chain data. From β, we have a dedicated indexer that only syncs on-chain data without updating anything like in α. An Akamon database should then be a local cache for fast queries that can be re-synced from scratch any time.

2.4 Akamon Cardano

Akamon Cardano implements the specific Plutus minting policy, events, actions, errors, and testing utility on the Cardano side of the bridge. For α, a wrap ADA or redeem mMATIC transaction on Cardano sends ADA or mMATIC to the bridge with the Polygon recipient address in the transaction metadata. The bridge then sends mADA or MATIC on Polygon accordingly. On the other end, the bridge builds, signs with `CARDANO_SIGNING_KEY`, and submits a transaction to the recipient when receiving wrap MATIC or redeem mADA events on the Polygon end.

Currently, α heavily depends on Cardano Wallet for most interactions, including bridge wallet management, querying incoming transactions, building and sending transactions, and more. A Cardano Node socket is only required to fetch protocol parameters for building the minting transactions. However, Cardano Wallet's role will reduce with each iteration – to be replaced by PAB and other off-chain toolings that MELD has been building.

For integration testing, we have a Dockerized Cardano private testnet in `docker/cardano-private-testnset` and a local setup in `docker-compose.yml` to test against. A production setup replaces this private testnet with a Cardano Node connected to a public testnet. We always need a Cardano Wallet connected to one of the Cardano nodes' socket. The bridge node then connects everything through the `CARDANO_WALLET_HOST`, `CARDANO_WALLET_PORT`, `CARDANO_WALLET_ID`, `CARDANO_WALLET_PASSPHRASE`, and `CARDANO_SIGNING_KEY` environment variables.

2.5 Akamon Polygon

Akamon Polygon implements the Polygon side of the bridge. Both the mADA token and the bridge contracts are written with OpenZeppelin's Upgradeable Contracts. We then build the ABI with Nix for MELDapp integration. We utilize Truffle for development, compiling the contracts, writing tests, and deployment.

For the off-chain components, `akamon-polygon` implements the Polygon events, actions, errors, and testing utility that the bridge node needs. For α, a wrap MATIC or redeem mADA transaction on Polygon sends ADA or mMATIC to the bridge contract with the Cardano recipient address. The bridge contract then emits an event for the node to catch before initiating the Cardano transactions accordingly. For Polygon contract interaction, the node signs with the key read from the `POLYGON_PRIVATE_KEY` environment variable. The contracts must have been deployed with the same key.

Akamon Polygon heavily depends on the `hs-web3` Haskell library. At the infrastructure level, it depends on a Polygon RPC to interact with the Polygon network. For development and integration testing, we have a Dockerized private testnet in `docker/hardhat` and a local setup in `docker-compose.yml` to test against. A production setup replaces the private testnet with a Polygon Node (Heimdall plus Bor).

2.6 Akamon Node

The Akamon Node ties all core components together into a single executable service, including database, Cardano, and Polygon configurations and interactions with their respective runtime services. The node also contains other configurations and utilities for re-trying policy and stress testing. If we design the `akamon-core` abstractions right, the node modules should stay tightly small even when we support more chains going forward. Two current potentials are Avalanche and Nervos.

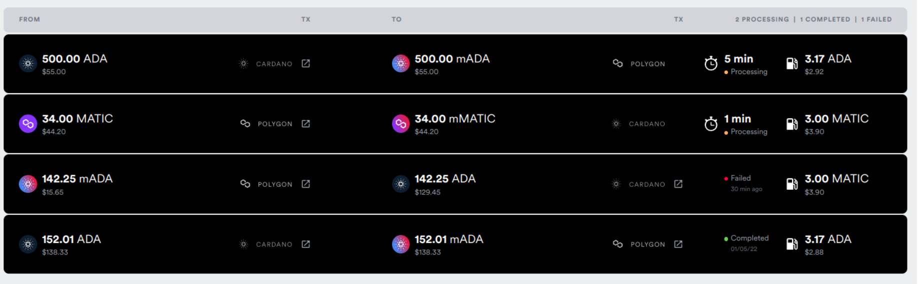
`docker-compose.yml` lists all the services we need for a local development setup and a full integration test suit with simulated concurrent users. Apart from the PostgreSQL database, Cardano and Polygon private testnets, and Cardano Wallet, we also have a `cardano-submit-api` server to connect Nami with the Cardano private testnet.

For production, we use Nix to build a Docker image for the node, which works well with the `haskell.nix` setup we already use for development. A sample production setup with the Dockerized node is in `docker-compose.alpha.yml`. For deployment, we mainly utilize Terraform and AWS (RDS, EC2, Fargate, S3, Cloudfront, and more). A good to have is sharing production resources with other MELD services.

2.7 Akamon API

The Akamon API server is the only other executable that the bridge builds. It serves data from the Akamon DB for UI rendering, with other endpoints to improve UX and MELDapp integration, including bridge addresses, fees, wrap ETA, transaction submit, status, and listing.

As we decentralize the bridge with time, community members should be able to build and operate their own Akamon frontend on top of its node and API server. For now, MELDapp will be the first and only interface to Akamon.



FROM	TX	TO	TX	2 PROCESSING | 1 COMPLETED | 1 FAILED	
500.00 ADA $55.00	CARDANO	**500.00** mADA $55.00	POLYGON	**5 min** ● Processing	**3.17** ADA $2.92
34.00 MATIC $44.20	POLYGON	**34.00** mMATIC $44.20	CARDANO	**1 min** ● Processing	**3.00** MATIC $3.90
142.25 mADA $15.65	POLYGON	**142.25** ADA $129.45	CARDANO	● Failed 30 min ago	**3.00** MATIC $3.90
152.01 ADA $138.33	CARDANO	**152.01** mADA $138.33	POLYGON	● Completed 01/05/22	**3.17** ADA $2.88

3 Akamon β

We do not intend to release the centralized α to mainnet. We have been designing a more distributed and decentralized β in parallel. The main goal of an α testnet launch is to collect UX, infrastructure, operations, and community management experiences to have more space to focus on decentralization later on.

User needs and how they interact with the bridge should stay the same regardless of its underlying architecture. One should not sacrifice UX to be a little bit more decentralized if that is quantifiable in the first place.

We have already been writing the β paper with actual technical discussions on the economics model, governance model, and contract designs leading towards decentralization. The paper will be released at the end of May. A β testnet launch will follow suit on June 5th 2022.

4 Related Work

4.1 WBTC

WBTC[5] is one of the first widely used interoperability solutions in the history of blockchain. It wraps Bitcoin to Ethereum trustfully but utilizes a multi-institutional design to distribute trust and power. WBTC's design has become outdated nowadays, with certain drawbacks like KYC and AML. That said, it has brought many right ideas to the interoperability space, including but not limited to the emphasis on security, trust model design, DAO-driven governance, sidechain for scalability, atomic swaps for wrapped tokens, and regulation.

4.2 RenVM

RenVM is a Byzantine fault-tolerant network that enables universal interoperability between blockchains. By combining consensus with their secure multiparty computation (MPC[6]) algorithms, RenVM can instantiate a decentralized, permissionless, and trustless custodian capable of locking assets on one chain and minting one-to-one pegged representations of them on other chains. In this way, users can interact with multiple applications, assets, and chains with only one transaction.

At a glance, RenVM does everything right and should be a role model for trustless bridges in general. The only disadvantages are a complex design compared to trustful solutions, a non-trivial requirement on the number of validators and their collateral, and a closed-source node at writing.

Building native Cardano support on RenVM is also something we are planning to explore.

4.3 Force Bridge

Force is a bridge that connects Nervos with other blockchain systems. Nervos has recently launched the first mainnet version that supports Ethereum. Cardano, Bitcoin, and other networks should follow eventually.

The bridge will maintain the light client in a multi-signature notary scheme at the first stage. A committee consisting of the Nervos Foundation and the community members will submit headers to the light client and their signature.

The bridge will replace the multi-sig-light-client in the second stage with a consensus-based-light-client. It will be fully decentralized, where everyone can submit headers. The contract will verify the header with the consensus algorithm of the chain.

We are studying their solutions for potential inspiration and collaboration.

5 Conclusions

This α paper introduces all the components one may need to build a bridge between Cardano and Polygon. The design is intentionally centralized first to build a foundation for UX, infrastructure, and operations. Decentralization works have been done in parallel and will be introduced in a β paper at the end of May 2022. Blockchain interoperability is a relatively new field so much more innovation is still awaiting. We expect to improve many contract designs with the upcoming Cardano hard fork. Off-chain infrastructure and tooling on all ecosystems can only get better with time as well.

References

[1] Rafael Belchior, André Vasconcelos, Sérgio Guerreiro, and Miguel Correia. A survey on blockchain interoperability: Past, present, and future trends, 05 2020.

[2] M. Chakravarty, James Chapman, K. Mackenzie, Orestis Melkonian, M. P. Jones, and P. Wadler. The extended utxo model. In *Financial Cryptography Workshops*, 2020.

[3] Jaynti Kanani, Sandeep Nailwal, and Anurag Arjun. Matic white paper, 2020.

[4] Pascal Lafourcade and Marius Lombard-Platet. About blockchain interoperability, 05 2020.

[5] Kyber Network, BitGo Inc, and Repulic Protocol. Wrapped tokens, 2019.

[6] Ross Pure and Zian-Loong Wang. Renvm secure multiparty computation, 2020.

[7] Alexei Zamyatin, Mustafa Al-Bassam, Dionysis Zindros, Eleftherios Kokoris-Kogias, Pedro Moreno-Sanchez, Aggelos Kiayias, and William J. Knottenbelt. Sok: Communication across distributed ledgers. In Nikita Borisov and Claudia Diaz, editors, *Financial Cryptography and Data Security*, pages 3–36, Berlin, Heidelberg, 2021. Springer Berlin Heidelberg.

[8] Alexei Zamyatin, Dominik Harz, Joshua Lind, Panayiotis Panayiotou, Arthur Gervais, and William Knottenbelt. Xclaim: Trustless, interoperable, cryptocurrency-backed assets. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 193–210, 2019.