AZO Wallet

Contact Details:

Krish

: rony.kris@gmail.com

Summary

AZO wallet is a state-of-the-art non-custodial browser-based wallet built for Aztec and designed for a smooth user experience.

It will be built as a browser extension since most of the dApps are built and optimised for the desktop screen and having a browser based wallet will give the user flexibility to use the wallet and the dApp simultaneously without having to switch screens/apps.

AZO will support zkChain's native functionalities, including gas less transactions, and privacy-enabled transfers. By incorporating advanced cryptography and a user-friendly interface, this wallet will serve as the gateway for users to harness the full potential of Aztec's zero-knowledge ecosystem.

It will enable user to interact with the Aztec and harness the seamless scalability of the L2 chain while using advanced features like the option to perform private transactions using QR codes, and the ability to pay the gas fees in ERC20 tokens.

AZO wallet is aiming to be the go-to wallet for the Aztec, focusing more on making the wallet usage learning curve non-existent.

User's can choose to log in to the wallet via the secure Passkey mechanism or social media logins. Users will get the option to log in via Google or Twitter using Web3Auth or Privy which will further enhance the UX of onboarding.

Estimated Start Date

: We will start as early as the next day the proposal gets approved

Estimated End Date

: ~31st Jan 2025

About Us :

- Saurabh Shetty

: has been a web3 and blockchain developer relations professional since 5+ years. Skilled in developer outreach, technical writing, and community management. Alongside he is the Director of TPG_Karnataka, Web3 Developer community, Mina Navigator Hackathon Winner. GitHub | LinkedIn

- Krish

: has roughly 7 years of web3 developer experience across numerous chains. He is experienced in building distributed scalable solutions and writing zk circuits in zokrates, O1js and noir. He has also been an Aztec ecosystem builder since the last few months and has also built a wallet in Alpha Build 1 for Aztec.GitHub

- Nikhil

: is an ace designer and frontend developer with expertise in scaling protocols and managing product. He has 4+ years of web3 experience on chain with marketing and content creation. He is an ambassador at Arbitrum and member of several crypto exchanges. GitHub | Linkedin

- Ayush

: A full stack developer with several wins and two years of experience in the web3 ecosystem. He has won Filecoin track prizes at ethIndia and worked in the Mina ecosystem. As a Core Member of the UniDAO program led by Devfolio, he fosters blockchain innovation in his campus by hosting weekly sessions on DeFi, DAOs, IPFS, NFTs, and hands- on projects. Devfolio | GitHub

Technical Details

Design Decisions:

- The wallet is designed to be a browser extension wallet first as the vast majority of dApps are designed to work smoothly on a full browser. Also, the user won't have the hassle of switching between the wallet and dApp as in the case of mobile wallets. Mobile wallets can be built in the later stages after analysing the traction and the community requirements.
- The onboarding flow will be devoid of seed phrases and password management as these data can be misplaced or lost. Instead, passkeys can be used so that the authentication/authorization can be offloaded to the user's primary device. Social logins will also be a great option for the less tech-savvy users as these are familiar means of logging in to an application.

Features:

Onboarding:

- A complex onboarding process acts as a deterrent for many users to get onboarded to web3.
- A wallet being an entry point to the on-chain world has a huge responsibility to make the user on boarding as simple as possible while maintaining the security.

Passkeys:

- AZO Wallet will remove these on boarding complexities involving seed phrases & passwords by enabling Passkeys.
- Passkeys will allow the user to get onboarded to the wallet by registering their device of choice (Windows, Mac, Android, iPhones or USB Security keys like Yubikey). The wallet of the user will be linked to this registration and can be generated on the fly.
- This will improve security as the wallet keys don't need to be stored on the client side reducing the attack surface. Moreover, the authentication/authorization security will be offloaded to the user's device thus inhering the highest level of security.

Social Logins:

- For the less tech-savvy users, they can log in using their more familiar means of authentication like Log in via Google or Twitter, or any other social logins. This can be achieved by integrating Web3Auth or Privy. Account generation and Key management will be handled by these downstream services.

Functionalities:

Gas Abstraction:

- AZO users will have the option to pay for gas fees via any erc20 token using a predefined paymaster based on the best quotes. More advanced users will be able to choose their preferred paymaster for their gas fees.

Public/Private Transactions:

- The users will be able to perform public as well as private transactions enabling them to hide their addresses from any observers for a private transaction.

QR-enabled transactions:

- The users will have a choice to perform transactions (native asset transactions and contract calls) using a QR code. The QR transaction can support both public and private transactions where the public key of the sender will also be present in the QR for the data to get encrypted before executing the call.

Swaps/Bridges:

- Support for swaps and bridges to allow the users to convert their tokens within the Aztec chain and also bridge to another chain. This feature will require the existence of a swap/bridging service/SDK.

Account Recovery:

- The user can download an encrypted version of their keys that is used as an entropy to generate the wallets. The encrypted wallet can be either a JSON file with the encryption data and the encryption scheme details or a QR containing these data. The user can recover their account by either uploading the JSON file or scanning the QR code. This will truly add meaning to the phrase "Not your keys, not your crypto" as the encrypted wallet will always be at the custody of the users.

Batch Transactions:

- Since the wallet of the user is a smart contract wallet, Azo can support transaction batching. This batch transaction

calldata can be sent to the entry point contract which will execute all the individual transactions. Batching can enable atomicity of the transactions and less waiting times.

Other functionalities include:

- Track the synced block
- Public key of the wallet
- State of the wallet
- Transaction activity
- List of holding tokens
- Balances in tokens, native asset and USD equivalent
- Transaction activity
- List of holding tokens
- Balances in tokens, native asset and USD equivalent

Below are the front end screen design of AZO Wallet:

[

Onboarding

720×1200 45.8 KB

](https://europe1.discourse-cdn.com/flex013/uploads/aztec/original/2X/1/13bbc649d04ee3741827c61d41dfca7339b78f88.jpeg)

[

Login

720×1200 56.6 KB

](https://europe1.discourse-cdn.com/flex013/uploads/aztec/original/2X/9/9afd20f26a325ff6e4aa7bdc4d668c53c15bd4d0.jpeg)

[

Swap

720×1200 40.7 KB

](https://europe1.discourse-cdn.com/flex013/uploads/aztec/original/2X/4/4204ddd853ddb4b0e4392be393e7dc05faf5156d.jpeg)

[

mainscreen

720×1200 65.3 KB

](https://europe1.discourse-cdn.com/flex013/uploads/aztec/original/2X/a/a41d026be252bb1ce0a74e70c2b6867c67bd10e6.jpeg)

Note:

We will be working closely with Aztec team to improve the product and add relevant features and suggestions.

Milestones & Roadmap:

Milestone 1: Basic Working Version of the Wallet

Goal:

Implement core functionality to get a minimal, usable version of AZO Wallet.

Deliver By:

~31st Oct 2024

1. Passkeys Implementation

:

- Integrate the device registration for Passkeys (Windows, Mac, Android, iPhone, Yubikey).
- Develop basic wallet generation linked to device registration.
- Integrate the device registration for Passkeys (Windows, Mac, Android, iPhone, Yubikey).
- Develop basic wallet generation linked to device registration.
- Wallet UI & Basic Features

:

- Set up UI for showing transaction history, token balances, and synced block.
- Display wallet public key and provide a view of holdings and balances in native assets and tokens.
- Set up UI for showing transaction history, token balances, and synced block.
- Display wallet public key and provide a view of holdings and balances in native assets and tokens.

Milestone 2: Complete Basic Features & Start Devnet Testing

Goal:

Implement some of the remaining features and test wallet functionality on dev

Deliver By:

~30th Nov 2024

1. Gas Abstraction

(basic version):

- Implement predefined paymaster option for ERC20 tokens to handle gas fees.
- Implement predefined paymaster option for ERC20 tokens to handle gas fees.
- Public/Private Transactions

:

- Enable both public and private transactions.
- Develop the ability to hide user addresses for private transactions.
- Enable both public and private transactions.
- Develop the ability to hide user addresses for private transactions.
- Account Recovery

:

- Provide encrypted wallet recovery via JSON files or QR codes for account recovery.
- Test and refine the encryption and recovery flow.
- Provide encrypted wallet recovery via JSON files or QR codes for account recovery.
- Test and refine the encryption and recovery flow.
- Basic Testing on Devnet

:

- Deploy wallet to devnet for testing with various dapps and transaction types.
- Focus on usability, security, and performance testing.
- Deploy wallet to devnet for testing with various dapps and transaction types.
- Focus on usability, security, and performance testing.

Milestone 3: Optimize for Public Testnet

Goal:

Final optimization of wallet for public testnet deployment.

Deliver By:

31st Dec 2024

1. Swaps/Bridges

:

- Integrate SDKs for swaps and bridges, allowing users to convert tokens and bridge to other chains.
- Integrate SDKs for swaps and bridges, allowing users to convert tokens and bridge to other chains.
- Batch Transactions

:

- Add support for transaction batching via smart contract wallet entry points.
- Ensure that atomic execution of transactions works smoothly.
- Add support for transaction batching via smart contract wallet entry points.
- Ensure that atomic execution of transactions works smoothly.
- Basic Dapp Interactions

:

- Develop injection of window.zk

into browser DOM for zk-chain dapp connections.

- Begin WalletConnect integration for connecting to dapps via QR code or URL.
- Develop injection of window.zk

into browser DOM for zk-chain dapp connections.

1. Begin WalletConnect integration for connecting to dapps via QR code or URL.
2. Advanced Testing on Public Testnet

:

- Complete optimization and testing of wallet functionalities on the public testnet.
- Ensure robust handling of multiple transactions, gas fee abstraction, and security for both private and public transactions.
- Complete optimization and testing of wallet functionalities on the public testnet.
- Ensure robust handling of multiple transactions, gas fee abstraction, and security for both private and public transactions.

Milestone 4: Implement advanced features

Goal:

Implement the remaining features and begin testing with early users.

Deliver By:

31st Jan 2025

1. Social Logins

:

- Integrate Web3Auth or Privy for social login options like Google or Twitter.
- Handle account generation and key management through these services.
- Integrate Web3Auth or Privy for social login options like Google or Twitter.
- Handle account generation and key management through these services.
- Metamask Snaps Exploration

:

- Perform feasibility study and profiling
- Begin implementation of AZO Wallet as a snap inside Metamask if feasibility allows.
- Perform feasibility study and profiling
- Begin implementation of AZO Wallet as a snap inside Metamask if feasibility allows.
- QR-enabled Transactions

:

- Implement QR-based transactions for native asset transfers and contract calls.
- Support both public and private transaction types with QR codes that include public keys.
- Implement QR-based transactions for native asset transfers and contract calls.
- Support both public and private transaction types with QR codes that include public keys.
- Optimize and Scale UI

:

- Refine UI for better performance and handling of transaction histories, balances, and account information.
- Refine UI for better performance and handling of transaction histories, balances, and account information.
- Gas Abstraction (Advanced)

:

- Expand functionality to allow users to select paymasters and compare fee options.
- Expand functionality to allow users to select paymasters and compare fee options.
- User Testing on Public Testnet

:

- Expose to user for early testing
- Incorporate user feedback
- Expose to user for early testing
- Incorporate user feedback

Continuous Improvements

Beyond Month 4

,

- focus on gathering user feedback
- make cross-chain functionality performant
- explore partnerships for swaps/bridges and paymasters
- implement additional features, functionalities to improve user flexibility. To list a few:
- Make Authwits readable
- Add on/off ramp integrations
- Complete Snaps implementation
- Prepare for Mainnet launch!

Grant Amount Requested: $96,000

Grant budget breakdown:

Milestone

Purpose

Amount ($)

1

Engineering and Design

20k

Software Development

Frontend

Design

2

Engineering

Software Development

30k

Frontend Integration

Tools, Softwares

3

Engineering

25k

Software Development

Frontend Integration

Testing

4

Engineering

21k

Software Development

Frontend Integration

Testing

Grant budget rationale :

- The amount will be distributed amongst the developers working on the project for the duration of their deliverable.
- Portions of the fund could also be used to purchase tools/licenses if the need arises during the course of the project.

Questions:

1. Could engineering share features / functionalities that could break between devnet and testnet?
2. What support could we expect from Aztec post completing our milestones?