BootNode | Wheez(i) Wallet

Contact Details

pablo@bootnode.dev, maxi@bootnode.dev

Summary

This project is a browser-based cryptocurrency wallet built on Aztec Wallet Wheez(i), developed during the AlphaBuild 1 hackathon. It features a backend for 2FA and data storage, along with sustainable code for various smart contract account options. The wallet offers diverse account creation methods, comprehensive token and note management, DApp integration, and advanced features like paymasters and batch transactions. It prioritizes flexibility and ease of use with options to toggle between basic and advanced modes.

Estimated Start and End Date

October 10, 2024 - February 10th, 2024

About You

BootNode is a long-term, end-to-end engineering partner that helps speed up the delivery and growth of protocols, dApps, and networks. We collaborate with networks and protocols at an engineering level to help design, build, integrate, and test software.

The team will consist of:

- Blockchain/Noir engineer who has experience working with Noir for 1+ years.
- Blockchain/Noir engineer who has participated in AlphaBuild 1 hackathon and written a game on Aztec to test out dev experience.
- Blockchain / Solidity engineer who has 6+ years of experience to assist and review.
- Front-end engineer with 4+ years of experience in cohesive and fluent dApp development for Ethereum using React and Typescript.
- UX/UI designer and UI Developer who worked on the design and UX for Gnosis Bridge and Explorer.
- QA Engineer with proven experience in blockchain QA, providing faster identification of critical bugs and effectively reducing post-launch fixes and downtime (has experience testing 30+ dApps and interacting with 5+ different wallets).
- Project Manager with proven experience in blockchain projects.

The team's experience includes:

- [Gnosis Bridge and Explorer

](https://bridge-explorer.gnosischain.com/): we designed and built:

- Unified Gnosis Bridge:

A cross-chain UI/UX for bridging DAI and other tokens that uses xDAI and OmniBridge protocols under the hood. The Unified Bridges UI automatically detects the contracts that fulfill the end user bridging needs, without requiring users to know about xDAI, Omnibridge, locked tokens, claims, validator signatures, and many more niche wording and intricacies that hinder the real use case: bridging funds into and out of Gnosis Chain.

- Gnosis Bridge Explorer:

Displays transactions from both Gnosis native bridges (xDAI and OmniBridge) in a single explorer, shows validator status and signatures, bridge limits, and transactions' status, and allows users to claim from a single UI which means a significant improvement in what we like to call "User anxiety pill" because users get to see the lifecycle of a bridging transaction before hitting Discord with a ticket.

- Unified Gnosis Bridge:

A cross-chain UI/UX for bridging DAI and other tokens that uses xDAI and OmniBridge protocols under the hood. The Unified Bridges UI automatically detects the contracts that fulfill the end user bridging needs, without requiring users to know about xDAI, Omnibridge, locked tokens, claims, validator signatures, and many more niche wording and intricacies that

hinder the real use case: bridging funds into and out of Gnosis Chain.

- Gnosis Bridge Explorer:

Displays transactions from both Gnosis native bridges (xDAI and OmniBridge) in a single explorer, shows validator status and signatures, bridge limits, and transactions' status, and allows users to claim from a single UI which means a significant improvement in what we like to call "User anxiety pill" because users get to see the lifecycle of a bridging transaction before hitting Discord with a ticket.

- zkSync Azure Wallet
- we researched, designed, spec'd, and built a multi-sig wallet MVP for enterprises, fully leveraging Account Abstraction. It only requires a user's web2 credentials (i.e., Microsoft Azure, Google, etc.).

Part of the team has been working with Safe on building out the first version of Gnosis Safe.

Details

Solution description

- Browser wallet on top of Aztec Wallet Wheez(i) that we built as part of the AlphaBuild 1 hackathon.
- A simple backend for 2FA authentication and data storage will be given (notes discovery, backups, etc.), as a way to export secrets and protect the user from any dangerous change.
- Code sustainability to allow for more options for smart contract accounts.

Features

1. Create a new account based on different options (Multisig, Spending Limits, Cloud-Email Recovery/2FA, Guardian Recovery, Paymaster, Authwit)
2. Display Accounts and account details (Keys, Notes, Bloc Sync, PXE)
3. Token Integration (send, receive, show balances, shield, unshield)
4. Send/Receive Notes
5. Dapp Connection Integration
6. Authwit Support
7. Transaction History
8. Fee management (Paymaster Gas Option, Define %)
9. Export/Import Account Data. (Token List, Contacts List, Notes, etc.)
10. Integrate Token Bridge
11. Integrate Explorer
12. Receive Tokens QR/Share option
13. Multiple Paymasters
14. Authwit Readable
15. Batch Tx
16. Separation (toggle) between basic and advanced modes

Wireframes

[

1600×1467 140 KB

](https://europe1.discourse-cdn.com/flex013/uploads/aztec/original/2X/9/9ef8ac2ea28efc43d9f5c64fb73da8f7b7d93620.jpeg)

[

1600×1182 160 KB

](https://europe1.discourse-cdn.com/flex013/uploads/aztec/original/2X/e/e9246280adf8c95c78b8e6791b881e0bcd62cab8.jpeg)

[

1600×1364 151 KB

](https://europe1.discourse-cdn.com/flex013/uploads/aztec/original/2X/9/9b614f2f33ef1f8dcdcdbf659d0e2fab66b9d400.jpeg)

[

1600×1472 162 KB

](https://europe1.discourse-cdn.com/flex013/uploads/aztec/original/2X/a/ad8dfd7f4324588dd14f44c91c3861ba4d9f2f52.jpeg)

[

1600×1012 142 KB

](https://europe1.discourse-cdn.com/flex013/uploads/aztec/original/2X/a/adff9747436422517a75169e1d8c70f8fd0bb176.jpeg)

[

1600×851 87 KB

](https://europe1.discourse-cdn.com/flex013/uploads/aztec/original/2X/a/a2ddeb587fb8d768f9643d4a15424305dd9108bd.jpeg)

[

1600×1004 91.7 KB

](https://europe1.discourse-cdn.com/flex013/uploads/aztec/original/2X/0/0f13873cba2c7d032598eb5b8524a308c57378b7.jpeg)

[

1600×1089 92.3 KB

](https://europe1.discourse-cdn.com/flex013/uploads/aztec/original/2X/5/5ea20461c2f2fcb5f74486137890b3ed3b6a48b0.jpeg)

Implementation Roadmap

Milestone #1

1. Scope definition, project planning, UX/UI design, and solution architecture design.
2. Feedback iteration (share Figma and retrieve feedback from some potential users and Aztec Foundation team members)
3. Refinement of scope, UI, etc.

Timeline

: 2 weeks

Milestone #2:

Iterative development

Frontend, Backend, and Noir development

1. Multisig Contract development
2. Common smart contract interface and feature discoverability

3. Fees Integration (with and without paymaster)
4. Token Integration (send, receive, show balances, shield, unshield)
5. Dapp connect
6. Separation (toggle) between basic and advanced modes
7. Design interactions
8. Functional testing

Timeline

: 4 weeks

Milestone #3:

Iterative development

Frontend, Backend, and Noir development

1. Display Accounts and account details (Keys, Notes, Bloc Sync, PXE)
2. Integrate Token Bridge
3. Send/Receive Notes
4. Notes discovery
5. Account recovery and security (2FA/Cloud/Guardians/Signers)
6. Export/Backup of data
7. Design interactions
8. Functional testing

Timeline

: 4 weeks

Milestone #4:

Functional and UAT

1. Integration testing
2. Regression testing
3. Fix issues and refine the code
4. Ensure compatibility with testnet (if available)

Timeline

: 2 weeks

Milestone #5:

MVP Launch

1. Documentation (final documentation + Explainers, tutorials, how-tos)
2. Deploy MVP and perform final checks.
3. Open Source wallet: Prepare wallet for open-source distribution. Choose an appropriate open-source license and set up a public repository for the project. Ensure all code and documentation meet the standards for open-source projects.

Timeline

: 2 weeks

Post-testnet potential improvements

Frontend, Backend, and Noir development

1. Transaction explorer
2. Transaction history
3. Authwit readable
4. Multiple Paymasters
5. WASM module for PXE
6. TXs batching

Note:

We will include some of the features above if we have the time.

Grant amount requested: $98,000

Grant budget rationale

The budget will cover engineering expenses as per the below milestones.

BootNode will cover infrastructure costs.

Milestones

Timeline

Budget

M1

2 weeks

15,000

M2

4 weeks

30,000

M3

4 weeks

30,000

M4

2 weeks

15,000

M5

1 week

8,000

Total

13 weeks

98,000