Firewallet by Clarified Labs | Grant Proposal

Contact Details

hello[at]clarified[dot]io

Summary

Clarified Labs, Inc. proposes Project Firewallet

, a secure, user-friendly wallet solution for Aztec. Project Firewallet

will:

- Propose an open standard for communication between dApp websites and Aztec wallets/PXE
- Develop a client library and browser extension wallet that implements the above standard.
- Develop a PXE Firewall

to protect users from potentially malicious requests and provide a mechanism for fine-grained control over PXE access.

- Create a pluggable authentication system for account contracts that enables simple integration of new authentication mechanisms and rules.

Future Firewallet work is also planned that is beyond the scope of this grant proposal.

Estimated Start & End Dates

- Start Date: October 21, 2024
- Ready for Testnet: December 2024
- End Date: February 2025 (Ready for Mainnet)

About Us

Clarified Labs, Inc. is an early-stage startup focused on solving core trust and security issues in the blockchain space. Our founders, Tim & Chris, have decades of experience in the security industry and extensive experience protecting highly sensitive data by building foundational security services such as key management, access control, and detection infrastructure for both Web2 and Web3 companies. Clarified Labs is also currently developing compliance-friendly tooling for bridging and shielded transfers that will be launching alongside Aztec's testnet.

Details

Phase 1: Foundations

Time estimate: ~1.5 months, 2024-10-21 - 2024-12-09

In Phase 1, we start by building these foundational wallet components:

1. Aztec Wallet Provider Standard
2. Aztec Wallet Provider Library
3. Firewallet Browser Extension
4. PXE Firewall

Additionally, once we have the basic design finalized, Phase 1 will include obtaining legal counsel to review the project to identify regulatory concerns along with developing a Privacy Policy and Terms of Service for Firewallet.

The first development task is proposing an an open standard for communications between dApp websites and Aztec wallets – an Aztec Wallet Provider standard[1]

. This protocol will largely just expose the PXE & Aztec Node APIs via the browser extension and mechanisms for discovering and bootstrapping communications with a wallet. Think EIP-1193, but without the need for content script injection. We will also develop a convenience library to go along with the provider standard.

The Firewallet Browser Extension will be built on top of the wallet provider library and provide the user with a basic UI for managing PXE accounts and access. We chose a browser extension model for this initial version because it provides a familiar, simple integration point that is under the end user's control, and its location in the browser enables collecting metadata such as the web origin of a request that can then be used for things like access controls and phishing detection by the PXE Firewall.

The PXE Firewall sits between the Firewallet Extension and the PXE itself and enables access control and rule enforcement. We believe a firewall component is necessary to protect the PXE from potentially adversarial websites that may use it to violate the user's privacy (for example, through unauthorized calls to getRegisteredAccounts()

and getRecipients()

). It will act in conjunction with the browser extension to enable the user to enforce rules such as "allow

from

".

---

title: Phase 1 Overview

graph LR

subgraph b[Browser] web(https://dapp) ext(Firewallet Extension) web -- Wallet Request\n(jsonrpc) ---> ext end

subgraph fw[PXE Firewall] direction TB pxe(PXE) node(Aztec Node) pxe --> node end

b ----> fw

Phase 2: Account Contracts

Time estimate: ~1.5 months, 2024-12-16 - 2024-02-03

Phase 2 will focus on Account Contracts. The first task will be to build a pluggable authentication system for Aztec account contracts. We plan to start with the existing account contract reference implementations, extending them so that a single account contract can support any of the different key types, and also support basic key rotation. We are taking inspiration from Gnosis Safe and plan to follow a similar module design which is familiar to many contract wallet developers and enables users to select modules that fit their individual needs.

---

title: Phase 2 Overview

classDiagram class Account { +addModule(AccountModule) +removeModule(AccountModule) +executeModule(AccountModule, args) }

```
class AccountModule {
    +verify(data) bool
}

Account o-- AccountModule : supports

class Secp256r1Verifier {
    +verify(data) bool
}

class Secp256k1Verifier {
    +verify(data) bool
}

class SchnorrVerifier {
    +verify(data) bool
}

AccountModule <|-- Secp256r1Verifier
AccountModule <|-- Secp256k1Verifier
AccountModule <|-- SchnorrVerifier
```

Phase 2.5: Additional Authentication Mechanisms

If time permits, we plan to implement additional authentication modules for M-of-N and Passkey verification. Additionally, we will again take inspiration from Gnosis Safe and implement Guard functionality to enable an additional layer to implement account policies.

---

title: Phase 2.5 Overview

classDiagram class Account { +addModule(AccountModule) +removeModule(AccountModule) +executeModule(AccountModule, args) +setGuard(GuardModule) +removeGuard(GuardModule) }

```
class AccountModule {
    +verify(data) bool
}

class GuardModule {
    +pre_check(data) bool
    +post_check(data) bool
}

Account o-- AccountModule : supports
Account o-- GuardModule : protected by

class MofNVerifier {
    +verify(data) bool
}

class PasskeyVerifier {
    +verify(data) bool
}

class ValueLimiterGuard {
    +pre_check(data) bool
    +post_check(data) bool
}

class BalanceShouldOnlyIncreaseGuard {
    +pre_check(data) bool
    +post_check(data) bool
}

AccountModule <|-- MofNVerifier
AccountModule <|-- PasskeyVerifier
GuardModule <|-- ValueLimiterGuard
GuardModule <|-- BalanceShouldOnlyIncreaseGuard
```

Phase 3: Security Audit

Time estimate: ~3 weeks, 2025-01-20 - 2025-02-10

Phase 3 is reserved for a third-party security audit. We plan to engage a reputable audit firm during Phase 1 in order to get an audit scheduled for late January or February 2025. Our time will be spent working with the auditors and fixing any findings they have. If there’s any downtime, we’ll spend it improving documentation and polish for the mainnet launch and not adding new features.

Future Work

Phases 1-3 are our focus for the scope of this grant proposal. This section briefly mentions our plans for future work.

- Key & Private Data Management Tools

- Key Management Standards

: The key management problem for traditional

blockchains is already difficult for end users. With Aztec, there are many

more keys to keep track of due to the complexities of private transactions.

We believe there will be a need for purpose-built key management tools and

standards for the Aztec ecosystem to enable portability and interoperability.

- Note Discovery

: Help users keep track of their outbound & inbound transactions

- Data Persistence

: Secret data might be required to create a proof, and a loss of that data could lead to permanently frozen funds. Standards and services for wallets to securely and reliably store secret data need to be developed.

- Key Management Standards

: The key management problem for traditional blockchains is already difficult for end users. With Aztec, there are many more keys to keep track of due to the complexities of private transactions. We believe there will be a need for purpose-built key management tools and standards for the Aztec ecosystem to enable portability and interoperability.

- Note Discovery

: Help users keep track of their outbound & inbound transactions

- Data Persistence

: Secret data might be required to create a proof, and a loss of that data could lead to permanently frozen funds. Standards and services for wallets to securely and reliably store secret data need to be developed.

- Privacy-Preserving Hosted PXE & Node Infrastructure

: Not all users will want (or be able) to run their own nodes. Our design will prioritize user privacy while itigating risks associated with central services. * Hosted PXE: E2EE access to a PXE running in a trusted execution environment with remote attestation. This could also be an easy to deploy self-hostable product.

- Hosted Nodes: A naïve first version could just be a centralized node RPC service for users who are not as privacy-sensitive run in a no/low-log manner. More sophisticated privacy protection techniques will also be explored.

- Hosted PXE: E2EE access to a PXE running in a trusted execution environment with remote attestation. This could also be an easy to deploy self-hostable product.

- Hosted Nodes: A naïve first version could just be a centralized node RPC service for users who are not as privacy-sensitive run in a no/low-log manner. More sophisticated privacy protection techniques will also be explored.

- Compliance Standards

: Wallet owners should have the ability to specify their own compliance requirements. Wallets, dApps, and Aztec service providers will need to speak common protocols to communicate and verify if the party they are interacting with meets their requirements.

- Aztec-Backed L1 Wallet

: Extend a user’s Firewallet Account with a

companion L1 portal contract to enable first class support for L1 interactions

initiated on Aztec. This enables transaction initiation requirements

(multisig, value, etc) to be handled privately on Aztec

Grant Milestones & Funding Requests

Funding request: $100,000

- Phase 1: $45,000
- Phase 2: $35,000
- Phase 3: $20,000

A breakdown of estimated expenses are below. This is a low estimate and we expect our costs to exceed the amount requested, but we are prepared to self-fund additional expenses.

Grant Milestones | Phase 1: Foundations

Estimated expenses: $45,000

- $15,000: Frontend Development
- $15,000: Backend Development
- $15,000: Legal Consultation

Phase 1 Milestones

- Milestone 1: Wallet Communication Standards Proposal
- Time Estimate: 15 days, ending 2024-11-04
- Time Estimate: 15 days, ending 2024-11-04
- Milestone 2: Initial Wallet Client Library
- Time Estimate: 10 days, ending 2024-11-09
- Time Estimate: 10 days, ending 2024-11-09
- Milestone 3: Initial Firewallet Browser Extension
- Time Estimate: 21 days, ending 2024-12-01
- Time Estimate: 21 days, ending 2024-12-01
- Milestone 4: Initial PXE Firewall Implementation
- Time Estimate: 21 days, ending 2024-12-09
- Time Estimate: 21 days, ending 2024-12-09
- Milestone 5: Legal Consult
- Time Estimate: 21 days, ending 2024-11-25
- Time Estimate: 21 days, ending 2024-11-25

gantt title Firewallet Phase 1: Foundations dateFormat YYYY-MM-DD tickInterval 1week weekday monday Define/Propose Standards :standards, 2024-10-21, 15d M1 : milestone, after standards Client Library :clientlib, 2024-10-31, 10d M2 : milestone, after clientlib Browser Extension :mbe, after clientlib, 21d M3 : milestone, after mbe PXE Firewall :fw, 2024-11-18, 21d M4 : milestone, after fw Legal Consultation :legal, 2024-11-04, 21d M5 : milestone, after legal

Grant Milestones | Phase 2: Account Contracts

Estimated expenses: $35,000

- $15,000: Smart Contract Development
- $10,000: Frontend Development

Phase 2 Milestones

- Milestone 6: Implement module support in Account Contracts & initial auth modules
- Time Estimate: 28 days, ending 2025-01-13
- Time Estimate: 28 days, ending 2025-01-13
- Milestone 7: Add support for Guard modules (if time permits before mainnet launch)
- Time Estimate: 21 days, ending 2025-02-03
- Time Estimate: 21 days, ending 2025-02-03

gantt title Firewallet Phase 2: Account Contracts dateFormat YYYY-MM-DD tickInterval 1week weekday monday section Phase 2 Account Contract Modules :amods, 2024-12-16, 28d M6 : milestone, after amods section Phase 2.5 Guard Modules :gmods, 2025-01-13, 21d M7: milestone, after gmods

Grant Milestones | Phase 3: Security Audit

Estimated expenses: $20,000

- $15,000: Third-Party Auditor (this is a very low estimate)
- $5,000: Fixes

Phase 3 Milestones

- Milestone 8: Third-Party Security audit from a reputable firm (assuming we can get it scheduled in this time)
- Time Estimate: 21 days, ending 2024-02-10
- Time Estimate: 21 days, ending 2024-02-10

gantt title Firewallet Phase 3: Security Audit dateFormat YYYY-MM-DD tickInterval 1week weekday monday section Audit

```
Third-Party Security Audit & fixes :audit, 2025-01-20, 21d
M8: milestone, after audit
```

[1] We recognize that coming to a consensus on a standard for this in our desired time frame is not realistic, but we at least want to share a draft spec and hopefully start collaborating with other wallet development teams. ↩