

Infinity Storage

Neil Roy, Kevin Lee, Edwin Medrano Villela, Awin Zhang, Suhrit Padakanti

Team Mysten

January 9th, 2025 ~ Development Plan

Features

Production Deployment

Deploy the Infinity Storage platform to a secure and scalable production environment (**Vercel**), ensuring reliability and performance. This includes setting up deployment workflows, configuring the production environment, and supporting ongoing updates for real-world use.

S3 Caching (w/Advanced Securities)

As discussed in prior documentation, caching will serve to speed up download operations requested by the user. This will utilize AWS S3 (a centralized service) due to its significantly faster performance compared to downloading directly from the Walrus network. Files will be uploaded asynchronously to S3 buckets when the files are deemed popular – files that have been downloaded at least twice! This reduces redundant cached files, in case files are just uploaded and never retrieved again. Throughout this, better practices when interacting with/using AWS will be held to ensure the privacy and security of our systems.

Point of Sale System (POS)

Seamlessly integrate a payment system for users to add balance to their account via **credit card**, **Apple Pay**, **etc.**, etc and save payment information for ease via **link**. Securely associate payments with user accounts by **webhook-based** verification and balance crediting. Additionally, we need to support development and testing without exposing payment credentials.

User Sovereignty

This feature ensures that even if centralized services, like AWS S3 or other services in our custom server, go down... users can still access their files from the network (and all necessary info tied to them). This is achieved through smart contracts that enforce cryptographic ownership via private key signatures, enabling self-custody wallets and programmable permissions, without intermediaries! And all of this ensures only users can modify their permissions and be completely independent of any single service provider.

File Sharing

Allow users to securely share files without giving up ownership. File sharing via link or one-time password per file. Track access events without exposing the file contents. Outcome should be secure, controlled collaboration on files stored on-chain.

UI/ UX Updates

We additionally want to ensure that our solution is consistent across our site, such as the name of our project, theming, and additional context that may be needed for new users to get comfortable with our application.

Directory System & Additional Customization

Introduce folder-based file storage for uploaded user files. Goals are to support nested directories and allow for simplistic drag and drop organization like users are used to. Store directory data separately from file blobs (either as its own uploaded blob or in cookies). Preserve directory structure regardless of blob uploads or downloads.

Timeline

Sprint 1B

- Production deployment
- S3 Caching (w/Advanced Securities)
- Point of Sale System (POS)
- User Sovereignty (data ownership & access controls)

Sprint 2B

- UI Updates and UX Improvements
- File Sharing Functionality
- Directory System & Additional Customization
- Initial Bug Fixes and Performance Improvements

Sprint 3B

- Comprehensive System Testing
- Bug Fixes and Stability Improvements
- Performance Optimization
- Security Testing and Validation

Sprint 4B

- Final Bug Fixes and Quality Assurance
- Demo Environment Setup
- Presentation and Demo Preparation
- Documentation Finalization
- Final System Validation