Software Requirements Specification (SRS)

Introduction

1.1 Vision

The **Hybrid Backup Service with Decentralized Sovereignty** project aims to merge the performance of centralized storage with the resilience and independence of decentralized systems. It ensures that users retain **full control** of their **encrypted data** while maintaining the simplicity and speed expected from modern backup services - something the current cloud-based internet has lost, evident in the Google Cloud outage back in early June and the AWS outage that was experienced mid-October 2025.

1.2 Purpose and Scope

The system provides a **hybrid backup architecture** integrating a **centralized caching layer** for fast access with a **decentralized backend (Walrus)** for secure, long-term storage.

Key functions include:

- Uploading and downloading encrypted files.
- Performing **client-side encryption** and optional caching.
- Supporting **lazy uploads** for offline or deferred transfers.
- Allowing **direct recovery** from Walrus in case of service failure.

The system demonstrates how decentralized storage can be made practical and user-friendly without sacrificing sovereignty or reliability.

Overall Description

2.1 System Overview

The service consists of two core layers:

Centralized Cache Layer: Handles fast data access, transaction abstraction, and synchronization with Walrus.
Decentralized Walrus Layer: Stores encrypted blobs for durability and independent recovery.

On top of this, we should have a cohesive and fluid UI that coordinates uploads, downloads, encryption, caching, and wallet-based transactions.

2.2 Key Personas

- **End User:** Uploads, downloads, and recovers files through the client interface.
- **Service Provider:** Operates the cache server and manages synchronization with Walrus.
- **Developer:** Extends or integrates the system into external applications or platforms.

2.3 Assumptions

- Users connect a Sui-compatible wallet for WAL/SUI transactions.
- Encryption keys are stored securely on the client.
- Network connectivity to Walrus and the caching service is available.
- The Walrus SDK and Sui Wallet SDK are reliable for file and transaction operations.

User Stories

1. As a user, I want to be able to download files securely, regardless of the failure of a centralized system.
2. As a user, I want to be able to check my Sui and WAL balance so that I know if I have enough coins to store a file.
3. As a user, I want to view a list of all my uploaded files so that I can easily manage, access, or delete them when needed.

Specific Requirements

Functional Requirements

1. The system should encrypt files and decrypt them after download using a symmetric encryption scheme (e.g. AES-GCM).
2. The system should handle lazy upload so the user perceives the upload as happening much faster than it actually is.
3. The system must ensure that user files and wallet identifiers remain private and inaccessible to developers or third parties.

Non-Functional Requirements

1. The user should experience a smooth and responsive file upload experience, ensuring that users perceive upload times as comparable to standard cloud storage services (ie.y Google Cloud).

Appendices / Glossary

- **AES-GCM:** Advanced Encryption Standard with Galois/Counter Mode. A symmetric encryption algorithm provides both confidentiality and data integrity.
- **Backend:** The server-side components responsible for logic, data management, and communication with decentralized networks.
- **Blob ID:** A unique identifier or hash assigned to each uploaded file to verify its authenticity and integrity.
- **Caching Layer:** A temporary storage component that stores frequently accessed data or recently uploaded/downloaded files to reduce latency and improve system responsiveness. Examples include in-memory caches such as Redis or local browser caches.
- **CLI:** Abbreviation for Command Line Interface
- **Decentralized Storage:** A storage model in which files are distributed across multiple nodes, eliminating single points of failure and enhancing data sovereignty.
- **Encryption:** The process of converting data into a coded format that can only be read or decrypted by authorized parties.
- **Frontend:** What our end-users interact with, and houses all the client-side operations.
- **Lazy Upload:** A technique where files are uploaded asynchronously, allowing users to continue interacting with the system before the upload fully completes.
- **Sui:** A Layer-1 blockchain designed for fast, secure, and scalable transactions, used in this system for managing wallet balances and transaction validation.
- **Symmetric Encryption:** A method of encryption that uses the same key for both encryption and decryption operations.
- **Wallet:** A digital account or cryptographic key pair used to manage tokens or digital assets on the Sui blockchain.
- **Walrus:** A decentralized data storage framework built by Mysten Labs that provides secure and distributed file persistence.