Memoir.ai — System Design Whitepaper

Architecture Vision, Design Rationale, and System Strategy

Document Version: 1.0

Status: Technical Whitepaper

Owner: Platform Architecture & Strategy

Last Updated: YYYY-MM-DD

---

1. Executive Summary

Memoir.ai is designed as a privacy-first personal archive platform that consolidates fragmented digital histories into a unified, searchable, and narratively explorable timeline. The system combines local-first architecture, encrypted storage, deterministic ingestion pipelines, and AI-assisted narrative generation while preserving user data sovereignty.

This whitepaper outlines the architectural vision, technical decisions, operational strategies, and long-term system goals enabling Memoir.ai to function as a scalable yet privacy-preserving personal knowledge platform.

---

2. Problem Statement

Modern digital life is fragmented across messaging apps, social networks, emails, media libraries, and devices. Users lack tools to unify, search, and contextualize this history while maintaining ownership and privacy.

Cloud-based aggregation tools often compromise privacy or create proprietary lock-in, preventing users from retaining long-term control over their data.

Memoir.ai addresses this by delivering:

• Unified ingestion across platforms

• Searchable chronological reconstruction

• AI-assisted narrative generation

• Local data ownership

• Vendor-independent export capabilities

---

3. Design Principles

System architecture adheres to the following principles:

Local-first operation

User data ownership

Encrypted persistence by default

Deterministic ingestion pipelines

Traceable AI outputs

Minimal cloud dependency

Long-term portability

Cloud services handle metadata and billing only, never personal content.

---

4. Architectural Overview

Memoir.ai operates as a desktop application composed of:

• Electron main process

• React renderer interface

• Background worker processes

• Encrypted vault database

• Job runner queue system

• AI narrative processing pipeline

• Hybrid search subsystem

• Optional cloud metadata synchronization

The architecture isolates compute-heavy tasks from user interaction to preserve responsiveness.

---

5. Execution Model

The system uses a multi-process model:

Main Process

Manages application lifecycle, IPC routing, and filesystem access.

Renderer Process

Hosts the UI and interacts with backend services through secure IPC bridges.

Worker Processes

Handle ingestion, indexing, AI inference, and export packaging without blocking UI execution.

This separation maintains stability under heavy workloads.

---

6. Data Flow Strategy

Key data flows include:

Ingestion Flow

Archives are parsed, validated, normalized, and stored in encrypted databases.

Narrative Flow

Selected evidence is transformed into citation-backed narratives via local AI inference.

Sync Flow

Subscription and workspace metadata synchronize with cloud services without transmitting personal content.

Each flow emphasizes traceability and recoverability.

---

7. Storage & Security Model

Storage architecture divides data into:

Vault Partition

Encrypted SQLCipher database and media assets.

Config Partition

Non-sensitive application settings.

Security measures include:

- AES-256 encryption
- Ephemeral in-memory keys
- Process isolation
- Strict IPC validation
- Sanitized logging

Personal content never leaves the vault without explicit export.

---

8. AI Narrative System Design

The AI subsystem converts event clusters into narratives while preserving factual grounding.

Pipeline stages include:

- Evidence sampling
- Context construction

• Local inference execution

• Citation mapping

• Hallucination verification

• Narrative version storage

All outputs remain editable and versioned.

---

9. Search Architecture

Search combines lexical and semantic retrieval:

• SQLite FTS keyword search

• Vector embedding semantic search

Results blend scores for accurate retrieval while maintaining encrypted local storage.

---

10. Background Job Strategy

Heavy tasks execute through job queues featuring:

• Priority scheduling

• Retry mechanisms

• Checkpointing

• Crash recovery

• Memory throttling

Jobs resume safely after interruption.

---

11. Performance Strategy

Performance objectives include:

• Fast vault unlock

• Smooth timeline scrolling

• Sub-second keyword search

• Efficient large archive imports

• Controlled memory usage

Optimization techniques include chunked ingestion, caching, and worker throttling.

---

12. Data Portability & Longevity

The platform guarantees long-term access through:

• Open export formats

• Structured JSON schemas

• Media hash indexing

• Narrative markdown mirroring

• Integrity checksum validation

Users can reconstruct archives without vendor tools.

---

13. Cloud Metadata Layer

Cloud components support:

• Subscription management

• Device metadata

• Job health monitoring

Strict row-level security ensures tenant isolation while excluding personal vault content.

---

14. Reliability & Recovery Strategy

Reliability features include:

• Import resumption

• Database integrity checks

• Backup restoration paths

• Worker crash recovery

• Retry and idempotency safeguards

System failures must never permanently block data access.

---

15. Future Architectural Directions

Planned evolutions include:

• Multi-device vault synchronization

• Collaborative shared vaults

• Improved semantic reasoning

• Enhanced indexing and summarization

• Incremental inference improvements

Future changes must preserve backward compatibility and data sovereignty.

---

16. Conclusion

Memoir.ai combines local-first architecture, encrypted storage, deterministic ingestion, and AI-assisted summarization to create a private, durable personal history platform. The system design prioritizes user control, scalability, and verifiable processing, ensuring long-term reliability and portability for personal archives.