NHOS TRACK™ LOCAL HEALTH INTELLIGENCE ARCHITECTURE

A Local-First Model for Interconnected Personal Health Applications

Technical Research Paper — v0.2.3 • First Release August 2026
NHOS Labs, Inc.

Abstract

NHOS Track™ is a personal-health application environment comprising a broad collection of specialized health-tracking and health-information functions designed to operate within a user-controlled local environment. This paper proposes a research program to investigate whether a large collection of specialized personal-health applications can maintain persistence, continuity, and meaningful interoperability while remaining capable of operating without routine dependence on centralized cloud storage. The study focuses on local-first execution, cross-application context continuity, persistence, offline operation, usability, reliability, cryptographic state protection, and measurable data minimization.

The paper does not claim that NHOS Track™ establishes a new local-first computing paradigm, nor does it claim completed scientific validation. Instead, it positions NHOS Track as an implemented research environment through which the composition of local-first principles with personal-health application ecosystems can be studied empirically. The research describes the architecture, research questions, and validation framework at a level sufficient for technical evaluation, while implementation-specific mechanisms remain abstracted as proprietary engineering details.

1. Research Motivation

Personal health technologies increasingly span tracking, assessment, education, medication management, wellness monitoring, and other specialized functions. Research on personal health records (PHRs) has identified interoperability, privacy and security, usability, data quality, personalization, and fragmented information as persistent implementation concerns. Systematic reviews also emphasize the importance of patient interaction with and control over personal health information. These findings provide a foundation for investigating alternative application architectures in which continuity is maintained locally rather than requiring every function to depend on a centralized repository.

Local-first software research provides a complementary architectural foundation. Local-first principles emphasize local ownership, offline capability, user control, and reduced dependence on centralized services. NHOS Track investigates how such principles might be composed across a large personal-health application environment rather than treating a single application as the unit of study.

2. Research Problem

The research problem is not whether an individual health tracker can operate offline. The more consequential question is whether a broad ecosystem of specialized personal-health functions can remain useful and coherent when local execution, local persistence, and cross-application continuity are treated as architectural requirements.

A multi-application environment introduces questions about context transfer, persistence, consistency, provenance, recovery, usability, scalability, and privacy. These questions become particularly important when applications are expected to operate independently while still contributing to a continuous user experience.

3. Research Gap

Existing literature addresses local-first software, personal health records, health-information interoperability, privacy/security, and digital health applications from related but generally distinct perspectives. PHR research identifies interoperability and fragmented health information as important concerns; interoperability research largely addresses exchange across health information systems; local-first research focuses on user ownership and local availability of data. NHOS Track investigates the composition of these concerns within a large interconnected personal-health application environment.

4. Central Research Question

Can a large interconnected environment of specialized personal-health applications maintain local persistence, cross-application continuity, and meaningful interoperability while supporting fully offline operation and minimizing unnecessary external transmission of user-generated health context?

5. Secondary Research Questions

- **RQ1 — Interoperability:** Can specialized personal-health functions exchange the context required to continue a defined workflow?
- **RQ2 — Persistence:** Can locally maintained user state survive normal application reuse, navigation, and recovery scenarios?
- **RQ3 — Offline operation:** Which workflows can be completed without network connectivity?
- **RQ4 — Continuity and user effort:** Does cross-application context reduce specified repetitive input and improve workflow continuity?
- **RQ5 — Privacy/data minimization:** What information, if any, is transmitted outside the local environment during defined workflows?
- **RQ6 — Reliability and scalability:** How do handoff success, recovery, storage, and performance behave as the number and diversity of interconnected functions increase?

- **RQ7 — Architectural Security and Protected Implementation Boundaries:** How can local-first personal-health architectures enforce state integrity and confidentiality while maintaining appropriate abstraction boundaries that protect proprietary implementation mechanisms?

6. Working Hypothesis

A personal-health application ecosystem designed around local execution, persistent user-controlled state, and structured cross-application context can support meaningful continuity among specialized functions while reducing routine dependence on centralized storage.

This is a working hypothesis, not a claim of completed scientific validation.

7. Architectural Principles

- Local-first execution where technically and functionally appropriate.
- User-controlled persistence of application state and health context.
- Interoperability through structured context rather than unnecessary duplication of user input.
- Offline capability as an architectural property to be measured by workflow.
- Data minimization as an explicit design and evaluation objective.
- Specialized applications may remain modular while participating in a shared ecosystem.
- Implementation mechanisms should remain distinguishable from architectural concepts.
- Privacy, correctness, safety, usability, cryptographic integrity, and clinical validity must be evaluated as separate properties.
- Architectural abstraction of client storage drivers, execution mechanisms, and underlying database primitives, allowing the research to focus on observable architectural behavior while preserving proprietary implementation details.

8. Conceptual Architecture

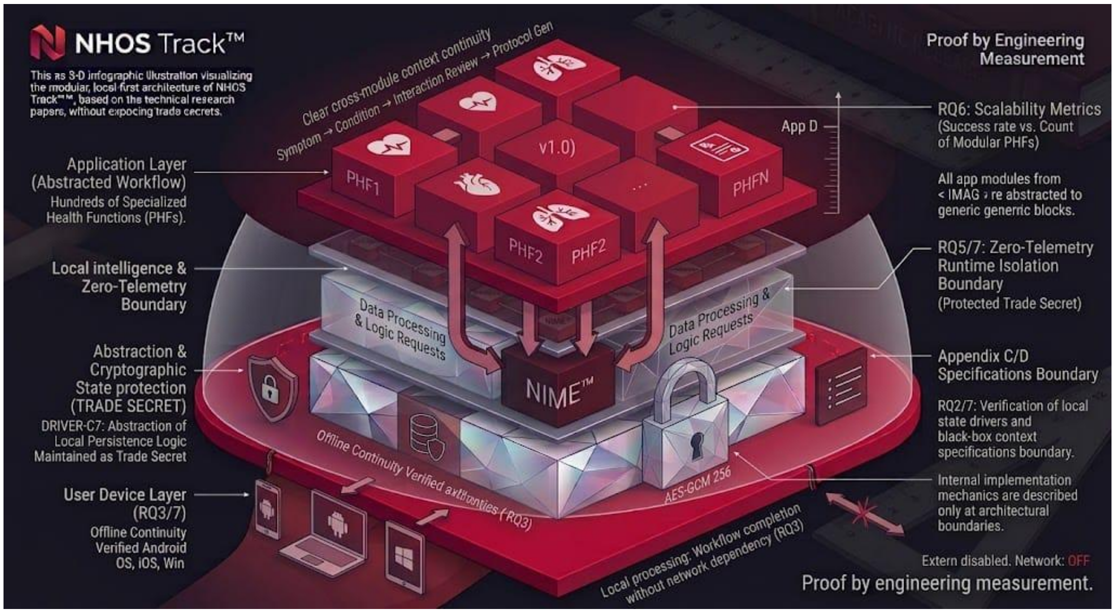

Figure 1. Conceptual Architecture of NHOS Track™. *The model represents five conceptual architectural responsibilities—application, intelligence/processing, context, persistence, and user device—rather than five necessarily separate physical software tiers.*

The proposed model can be represented as five conceptual layers: Application Layer, Intelligence/Processing Layer, Context Layer, Persistence Layer, and User Device Layer. These labels describe architectural responsibilities and should not be interpreted as a claim that the current implementation consists of five physically separated software tiers.

- **Application Layer —** specialized personal-health functions and user workflows.
- **Intelligence/Processing Layer —** local rules, calculations, transformations, mappings, and other application processing.
- **Context Layer —** structured state and workflow information that can be consumed by another application.
- **Persistence Layer —** mechanisms that retain relevant state locally across sessions.
- **User Device Layer —** the client runtime and local computing environment under the user's control.

9. Representative Composable Workflow

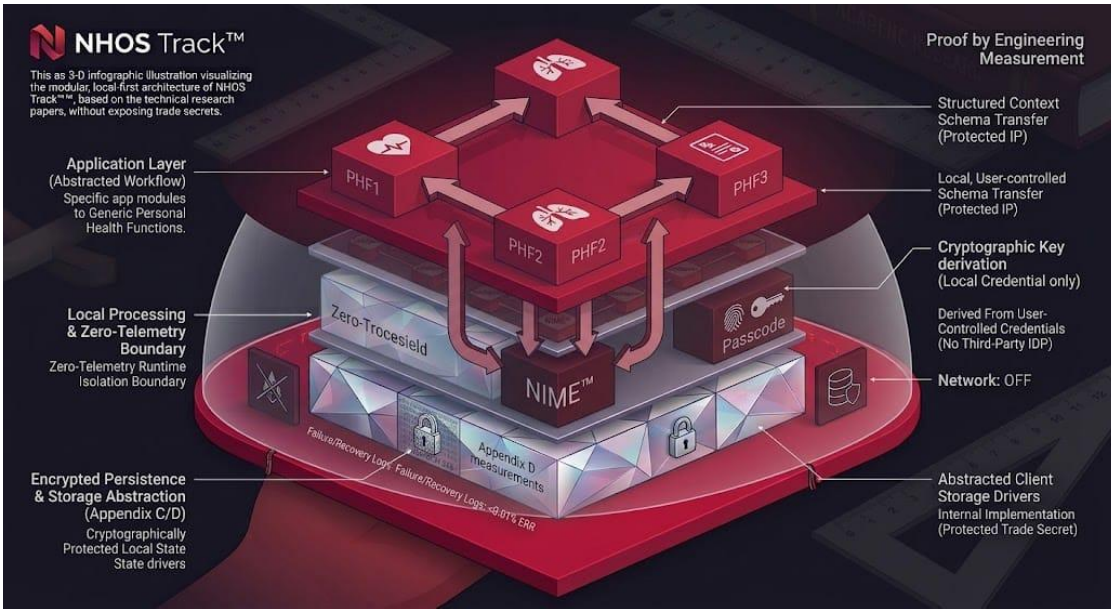

Figure 2. Representative local context continuity and cross-application workflow. *The illustration depicts specialized personal-health functions participating in a locally maintained workflow through structured context transfer and local processing. It represents an architectural research model rather than a clinical pathway or validated clinical workflow.*

A representative workflow may involve a user moving from one specialized function to another while selected context is retained or made available locally.

Health observation → assessment → relevant information → tracking → historical continuity

This sequence is an architectural example, not a clinical pathway and does not imply that every step is clinically appropriate or that NHOS Track provides diagnosis or treatment.

10. Meaningful Interoperability

For this research program, meaningful interoperability is operationalized as:

- successful transfer of the required context;
- correct interpretation by the receiving function;
- avoidance of specified manual re-entry;
- preservation of relevant provenance or source context where required; and
- successful continuation of the defined workflow.

11. Architecture Versus Implementation

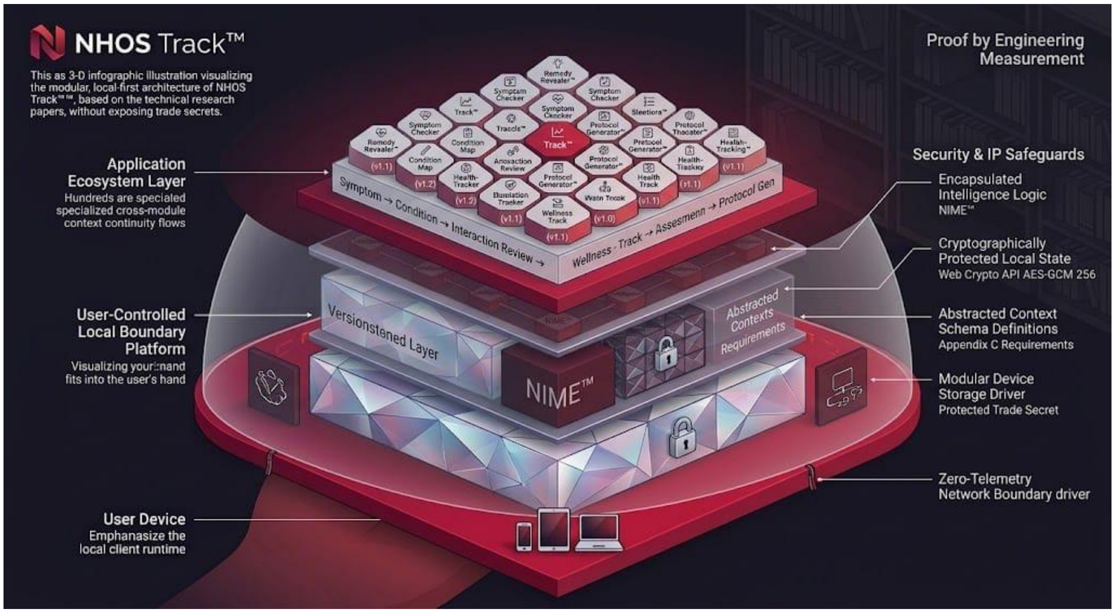

Figure 3. Architectural abstraction and implementation boundary. *The illustration distinguishes architectural responsibilities from implementation-specific mechanisms, enabling evaluation of observable system behavior without requiring disclosure of proprietary engineering details.*

Architectural concept	Representative implementation mechanism	Research status
Local persistence	Abstracted local storage engine & encrypted local state drivers	Implemented; formal measurement required
Application interoperability	Shared/local context handoff protocols	Implemented / preliminary evidence; validation required
Offline capability	Locally executable workflows	Implemented for defined functions; workflow-specific testing required
Specialized application ecosystem	Multiple personal-health functions	Implemented
Privacy/data minimization	Local-first design objective, a design objective of no routine telemetry for defined local workflows., and network behavior	Validation planned through instrumentation
Intellectual Property Protection	Abstracted context schema and encapsulated client logic	Proprietary implementation details

12. Context Model & Architectural Abstraction

The research model requires a structured representation of transferable context. This paper describes the architectural role of context, including schema versioning, required and optional fields, provenance, source function, timestamps, validation, retention semantics, deletion behavior, compatibility, and user visibility. These elements define the properties that a production context model should support without prescribing a specific implementation.

Implementation-specific mechanisms—including production schema structures, database engine configurations, internal key mappings, client execution structures, and local storage primitives—are treated as engineering details rather than prerequisites for evaluating the research questions. The research therefore focuses on the observable behavior, interoperability requirements, persistence characteristics, and validation criteria of the context model while maintaining appropriate boundaries around proprietary implementation details.

This approach allows the architectural model and its research questions to be evaluated independently of any particular database engine, source-code structure, or implementation framework.

13. Privacy, Data Minimization, and Local Security Safeguards

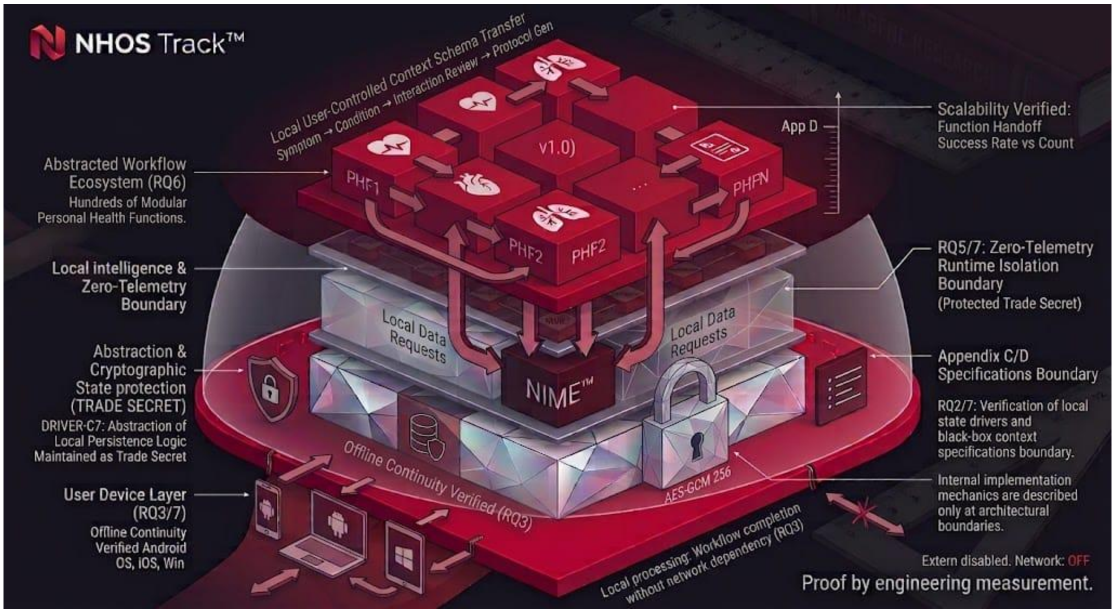

Figure 4. Conceptual local-first privacy, persistence, and security boundary. *The illustration depicts local processing, protected local persistence, user-controlled context, and an architectural boundary around external network activity. Specific implementation mechanisms are abstracted and require empirical verification..*

The architecture is designed around privacy-oriented objectives: local processing where feasible, encrypted local persistence, offline capability, zero-telemetry defaults, and minimization of unnecessary external transmission. These design objectives do not constitute proof of privacy or security. Privacy properties must be evaluated against deployed behavior, including actual network requests, payload characteristics, local storage security, and failure modes.

To safeguard local health context without relying on cloud-based identity providers, the architecture establishes a zero-trust local boundary. The intended security property is that protected local health state remains unreadable to an observer who does not possess the required key material; this property requires empirical verification. This aligns the architectural objectives with rigorous research discipline without turning design targets into unverified security claims.

Local processing does not automatically establish correctness, safety, confidentiality, or clinical validity. The system may also contain workflows or future components that require external resources; therefore, no absolute claim that all information can never leave the device is made.

14. Validation Framework

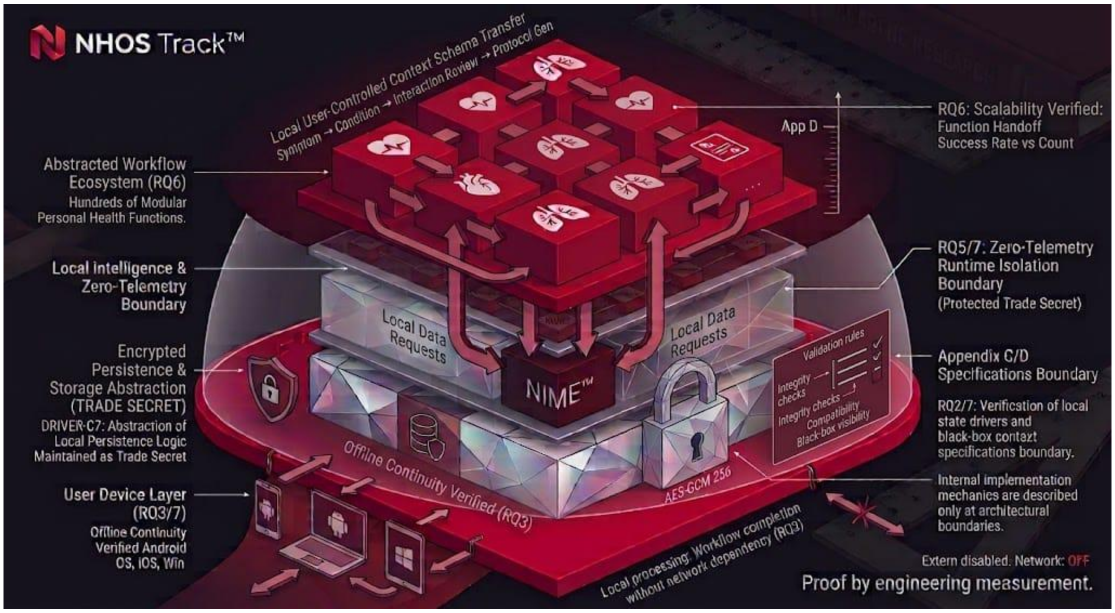

Figure 5. Proposed NHOS Track™ validation and measurement framework. *The illustration maps the proposed evaluation environment to application composition, local processing, persistence, context continuity, network behavior, and scalability measurements. It represents a validation framework rather than completed empirical results.*

Dimension	Proposed measurement	Evidence status
Local processing	Completion of defined workflows without network dependency	Validation planned
Network activity	Requests generated during controlled workflows	Validation planned
Data transmission	Bytes and payload categories transmitted externally	Validation planned
Persistence Integrity	Encrypted write/read/recovery tests across sessions and restarts	Validation planned
Context continuity	Successful cross-function handoffs without manual re-entry	Preliminary evidence / validation required
Offline capability	Workflow completion while disconnected	Preliminary evidence / validation required
Latency	End-to-end processing and handoff time	Validation planned

Storage footprint	Local encrypted storage footprint and growth rates	Validation planned
Reliability	Failure, recovery, cryptographic validation, and consistency rates	Validation planned
Compatibility	Cross-runtime/device test matrix	Future validation
Privacy & Isolation	Information leaving local environment and cross-origin boundaries	Validation planned
Usability	Task time, errors, repetitions, and user effort	Future validation

15. Experimental Program

The initial validation program should use predefined workflows and fixed test environments. Each test should record the application/function sequence, expected context, observed context, runtime environment, connectivity state, network activity, persistence behavior, cryptographic verification status, completion status, failures, and recovery behavior.

Negative results and failures should be retained rather than excluded. Where sample sizes and experimental design warrant it, quantitative analysis should report appropriate descriptive or inferential statistics. Claims should be reported according to the strength of the resulting evidence.

16. Evidence Status Framework

The following taxonomy distinguishes software existence from scientific validation and should be applied consistently across the NHOS research program:

- **Implemented —** functionality exists in the current platform.
- **Measured —** a defined test has produced reproducible observations.
- **Preliminary Evidence —** technical observations exist, but formal validation remains incomplete.
- **Active R&D —** the capability or question is currently under investigation.
- **Validation Planned —** a formal experiment or benchmark has been defined but not completed.
- **Future Research —** a proposed direction that is not yet implemented.

17. Threats, Limitations, and Safety Boundaries

- Local storage may be lost through runtime clearing, user reset, deletion, corruption, or hardware damage.
- Offline information may become stale when external updates are unavailable.
- Context propagation can create unintended information sharing between functions if boundaries are not correctly designed.
- Local-first operation does not guarantee security against full device compromise, host malware, or shoulder surfing.

- Interoperability does not establish clinical correctness or clinical appropriateness.
- Usability and continuity benefits must be demonstrated empirically rather than assumed from architectural design.
- Host environment variations may affect storage quotas, persistence retention, latency, and offline capabilities.
- Future components may introduce external dependencies that change the privacy profile of specific workflows.

18. Relationship to the NHOS Research Program

This project extends the NHOS research portfolio by examining personal-health application composition. NIME™ addresses how health information may be organized and processed as an intelligence architecture; the NHOS Local Intelligence Architecture addresses composition and context continuity across interconnected applications; NHOS Track extends the investigation into a large personal-health application environment.

The resulting research program therefore progresses from intelligence and reasoning, to application composition, to large-scale personal-health interaction and continuity.

19. Research Roadmap

- **Phase 1 —** Architecture documentation, threat modeling, and terminology stabilization.
- **Phase 2 —** Workflow inventory and interoperability test specification.
- **Phase 3 —** Local encrypted persistence, offline execution, and zero-telemetry network instrumentation.
- **Phase 4 —** Cross-platform reliability, local security boundary, and usability evaluation.
- **Phase 5 —** Analysis of context continuity, storage footprint growth, and scalability.
- **Phase 6 —** Publication of validated results, including failures, trade-offs, and security boundaries.

20. Research Contributions

This paper proposes three research contributions for investigation:

1. A conceptual model for composing specialized personal-health applications around locally maintained context and persistence.
2. An operational framework for evaluating interoperability, continuity, offline execution, persistence, privacy/data minimization, reliability, usability, and security in such an environment.
3. A research boundary that separates publicly describable architectural principles from proprietary implementation mechanisms protected as NHOS Labs, Inc. intellectual property.

These are proposed contributions of the research program and should not be interpreted as independently validated findings.

21. Conclusion

NHOS Track provides a practical environment for investigating whether a large ecosystem of specialized personal-health applications can remain interconnected while operating locally and offline. The research proposition is not that local-first software, personal health records, interoperability, or privacy-aware computing are individually novel. Rather, the proposed contribution is to investigate their composition within a specialized personal-health application environment and to establish measurable criteria for continuity, interoperability, persistence, cryptographic security, privacy, reliability, and usability.

The central proposition remains to be tested. The next stage of the project should therefore be driven by empirical evidence rather than additional architectural claims.

Appendix A — Terminology

- **Local-first:** An architectural orientation in which local operation and user-controlled data are primary design concerns rather than cloud availability being the sole operating assumption.
- **Context continuity:** The ability of a defined workflow to carry required information from one specialized function to another without unnecessary manual re-entry.
- **Meaningful interoperability:** The operational definition provided in Section 10.
- **Data minimization:** Limiting external transmission and retention of information to what is necessary for a defined function.
- **Personal-health application ecosystem:** A collection of specialized functions that can participate in a shared user workflow while remaining modular.
- **Architectural Abstraction:** The separation of architectural concepts from implementation-specific mechanisms, enabling research and evaluation without requiring disclosure of proprietary implementation details.

Appendix B — Evidence Discipline

NHOS Research documents should distinguish implementation claims, measured observations, preliminary evidence, active research, planned validation, and future research. Architectural proposals should not be presented as validated outcomes.

Appendix C — Future Production Context Specification

A future production specification may define schema versioning, required/optional fields, provenance, source function, timestamps, validation, retention, deletion semantics, compatibility, cryptographic integrity controls, and user visibility. Detailed proprietary implementation structures and driver mechanisms are intentionally outside the scope of this draft.

Appendix D — Proposed Validation Record

To convert implementation status into reproducible research evidence, each future validation record should include:

- Test ID and date

- Application/function sequence
- Workflow objective
- Expected context
- Observed context
- Runtime/device/environment
- Connectivity state
- Network requests and payload categories
- Persistence/read/write/cryptographic recovery observations
- Completion status
- Errors and recovery actions
- Evidence artifacts
- Researcher notes

Initial Literature Base

- **Kleppmann, M., Wiggins, A., van Hardenberg, P., & McGranaghan, M. (2019).** *Local-first software: You own your data, in spite of the cloud.* In *Proceedings of the 2019 ACM SIGPLAN International Symposium on New Ideas, New Paradigms, and Reflections on Programming and Software (Onward! '19)*, 154–178. **DOI: 10.1145/3359591.3359737.**
- **Roehrs, A., da Costa, C. A., da Rosa Righi, R., & de Oliveira, K. S. F. (2017).** *Personal Health Records: A Systematic Literature Review. Journal of Medical Internet Research, 19*(1), e13. **DOI: 10.2196/jmir.5876.**
- **Harahap, N. C., Handayani, P. W., & Hidayanto, A. N. (2021).** *Functionalities and Issues in the Implementation of Personal Health Records: Systematic Review. Journal of Medical Internet Research, 23*(7), e26236. **DOI: 10.2196/26236.**
- **Dobrow, M. J., Bytautas, J. P., Tharmalingam, S., & Hagens, S. (2019).** *Interoperable Electronic Health Records and Health Information Exchanges: Systematic Review. JMIR Medical Informatics, 7*(2), e12607. **DOI: 10.2196/12607.**
- **Fernández-Alemán, J. L., Carrión Señor, I., Oliver Lozoya, P. Á., & Toval, A. (2013).** *Security and privacy in electronic health records: A systematic literature review. Journal of Biomedical Informatics, 46*(3), 541–562. **DOI: 10.1016/j.jbi.2012.12.003.**

Publication Status

Technical Research Paper — v0.2.3 • First Release August 2026
NHOS Labs, Inc.
Research Stage: Architectural Model, Architectural Abstraction & Validation Framework

> **NHOS Track™ Local Health Intelligence Architecture**
> A Local-First Model for Interconnected Personal Health Applications
> NHOS Labs, Inc.
>
> **NHOS**: https://nhos.health
> **NHOS Research Registry:** https://nhos.health/research

> **NHOS Track**: https://nhos.health/track