NHOS LOCAL INTELLIGENCE ARCHITECTURE

A Privacy-Preserving, Local-First Model for Interconnected Health Intelligence Applications

Technical Research White Paper • Version 1.2 • August 2026

NHOS Labs, Inc.

Research Stage: Architectural Model & Validation Framework

Abstract

NHOS Labs, Inc. investigates a local-first architecture in which specialized health-information applications can operate as interconnected components of a single privacy-preserving environment. Rather than treating each application as an isolated product, the NHOS model is designed to allow selected application results and user-controlled context to become available to subsequent workflows while minimizing unnecessary transmission and centralized retention of user information.

This paper defines the architectural proposition, research gap, research questions, working hypothesis, system model, conceptual data flow, structured context model, interoperability model, local persistence model, privacy objectives, implementation-status framework, validation framework, experimental design, risks, research relationships, and future experiments for the NHOS Local Intelligence Architecture. IndexedDB is treated as an implementation-level browser-local persistence mechanism rather than as the architecture itself. Claims are deliberately separated into implemented capabilities, preliminary evidence, architectural hypotheses, and matters requiring empirical validation.

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

1. Research Context

NHOS is being developed as a privacy-first, evidence-informed health-information environment containing multiple specialized applications. The platform includes interconnected capabilities for natural-health exploration, symptom and condition mapping, medication–herb interaction review, protocol generation, evidence and citation access, wellness tracking, and intelligence/research functions.

The architectural challenge is not merely how to make individual tools useful. It is how to preserve continuity when a user moves between tools without requiring every application to become a separate cloud-dependent service.

This motivates a research direction distinct from the NHOS Intelligence Matrix Engine™ (NIME™) architecture work: the present program focuses on the composition and continuity of intelligence across applications under local-first and privacy-preserving constraints.

2. Research Problem

Conventional multi-application health platforms may rely on centralized services to maintain identity, state, histories, and cross-application context. NHOS investigates whether a meaningful portion of cross-application continuity can instead be achieved through locally retained, user-controlled context and structured application interoperability.

Central research problem: How can privacy-preserving, local-first health intelligence be composed across interconnected applications while minimizing unnecessary transmission and centralized retention of user context?

3. Related Architectural Approaches and Research Gap

Local-first software, edge computing, distributed processing, privacy-preserving architectures, personal health records, and health-information interoperability each address important portions of decentralized health-information handling. NHOS does not claim to invent these underlying concepts.

The research gap investigated here is their composition within a specialized multi-application health-intelligence environment, with emphasis on user-controlled context continuity, structured cross-application handoffs, local persistence, and measurable data minimization.

The proposed contribution is therefore architectural and evaluative: to define a concrete model for composing specialized health-information applications and establish experiments that can determine whether useful continuity can be achieved without routine centralized retention of user-generated health context.

For future journal submission, this section should be expanded into a formal literature review with scholarly citations. The present white paper deliberately avoids implying that the underlying local-first, edge, interoperability, or privacy concepts originate with NHOS.

Existing approach	Primary focus	Relevant limitation/gap	NHOS research emphasis
Local-first computing	Local state and availability	General-purpose orientation	Health-specific application composition and continuity
Edge/local computing	Computation near the user	Often infrastructure-oriented	User-controlled health context and workflow continuity
Personal health records	Personal health information	May be centralized or siloed	Cross-application local continuity

Health-information exchange	Institutional data exchange	Often institution/interoperability focused	Application-level composition under local-first constraints
Privacy-preserving systems	Data protection and minimization	Properties vary by architecture	Measurable minimization of health-context transmission

4. Central Research Question

Can interconnected health-information applications provide useful cross-application context and continuity through a local-first architecture while minimizing unnecessary transmission and centralized retention of user-generated information?

5. Secondary Research Questions

- **RQ1 — Interoperability:** Can specialized NHOS applications exchange sufficient structured context to support meaningful multi-step workflows?
- **RQ2 — Local persistence:** Can browser-local persistence reliably retain selected context across application interactions and sessions?
- **RQ3 — Privacy:** Can useful cross-application functionality be achieved without requiring centralized storage of the underlying user context?
- **RQ4 — Continuity:** Can cross-application context reduce repetitive user input and improve workflow continuity?
- **RQ5 — Reliability:** How reliably can context be preserved, retrieved, and propagated across supported browsers and operating conditions?
- **RQ6 — Performance:** What latency, storage, memory, and computational costs are introduced by maintaining local application context?

6. Working Hypothesis

A local-first architecture can support meaningful interoperability among specialized health-information applications by maintaining selected user-controlled context locally and exposing structured application outputs to subsequent workflows, thereby reducing the need for unnecessary transmission or centralized retention of user-generated information.

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

7. Architectural Principles

- **Local-first operation:** Where technically feasible, application processing and state management should occur on the user's device.
- **User-controlled persistence:** Selected context should remain under user control rather than requiring a centralized account.
- **Composable applications:** Specialized applications should be capable of participating in larger workflows.

- **Structured context exchange:** Interoperability should rely on defined data structures and application interfaces rather than fragile page-level assumptions.
- **Data minimization:** The system should avoid transmitting information that is not necessary for the requested operation.
- **Privacy by architecture:** Privacy objectives should influence system design rather than being added only at the presentation layer.
- **Evidence-aware operation:** Health-information workflows should preserve distinctions between information, evidence, inference, and user-generated context.
- **Graceful degradation:** Individual applications should remain useful when cross-application context is unavailable.

8. Five-Layer System Model

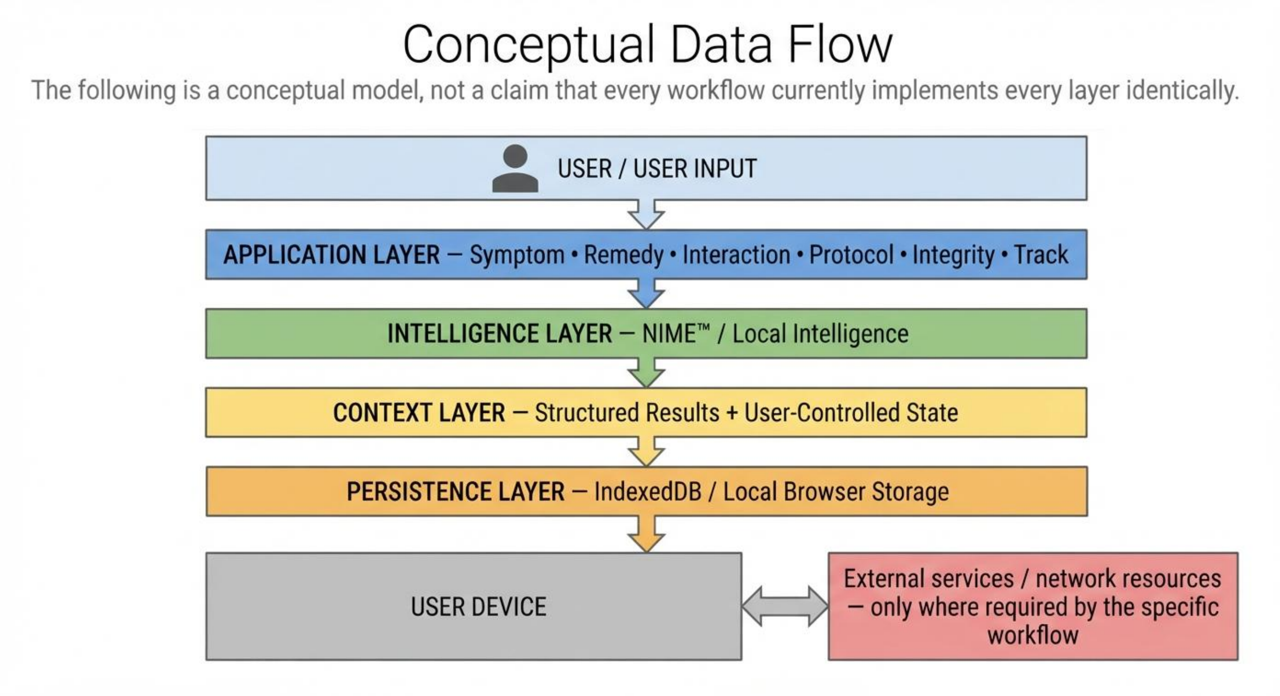
The proposed model comprises five conceptual layers:

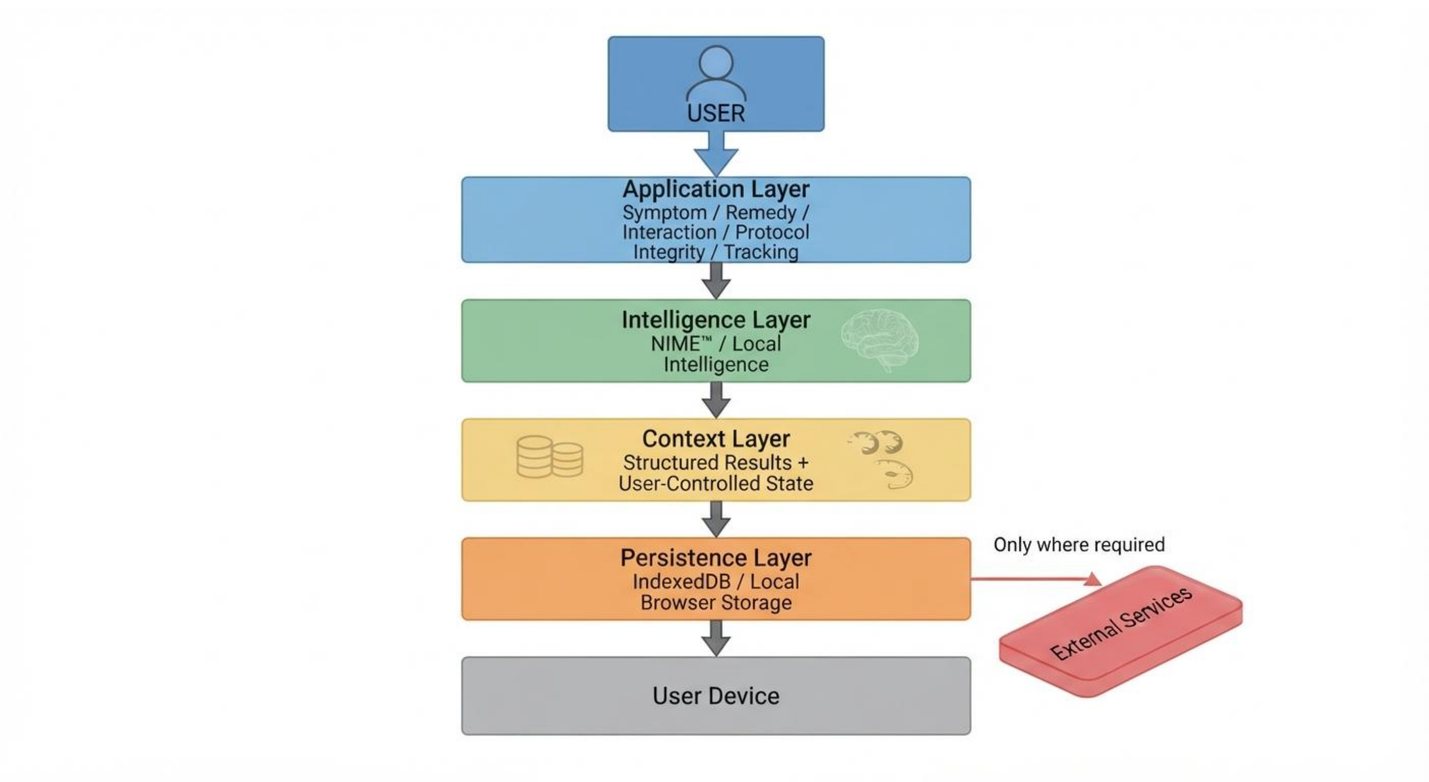
- **Application Layer —** specialized NHOS applications such as Remedy Revealer™, Protocol Generator™, Interaction Checker™, Symptom Checker, Integrity Checker™, and NHOS Track™.
- **Intelligence Layer —** NIME™ and related local intelligence functions that interpret, classify, organize, or transform information.
- **Context Layer —** structured application outputs and selected user-controlled state made available to subsequent workflows.
- **Persistence Layer —** browser-local mechanisms such as IndexedDB used to retain selected state and context.
- **User Device Layer —** the execution environment in which local processing and persistence occur.

These layers are conceptual. Exact boundaries and interfaces should be documented against the production implementation before being presented as a finalized reference architecture.

9. Conceptual Data Flow

The following is a conceptual model, not a claim that every workflow currently implements every layer identically.





Layer / Step	Description & Components
USER / USER INPUT	Initial entry point for user requests, parameters, or interactions.
APPLICATION LAYER	Symptom • Remedy • Interaction • Protocol • Integrity

	• Track
INTELLIGENCE LAYER	NIME™ / Local Intelligence processing & classification
CONTEXT LAYER	Structured Results + User-Controlled State handoff
PERSISTENCE LAYER	IndexedDB / Local Browser Storage for client-side persistence
USER DEVICE	Execution environment on device; external services consulted only where required.

10. Architecture Versus Implementation

The table below contrasts the high-level architectural concepts with their current representative implementations within the NHOS environment.

Element	Architectural concept	Current / representative implementation
Intelligence	Intelligence layer	NIME™ and related local intelligence functions
Context	Structured cross-application context	NHOS context/state mechanisms
Persistence	Local user-controlled persistence	IndexedDB / browser-local storage
Applications	Specialized composable modules	NHOS health-information applications
Device	User-controlled execution environment	Supported browser/device
Network	Optional external dependency	Workflow-dependent external resources

The implementation column is representative rather than exhaustive and should be updated as the production architecture is formally documented and tested.

11. Structured Context Exchange

A central architectural concept is that an application's output can become a structured starting point for another application. To make this concept technically testable, NHOS should represent transferable context through explicit schemas rather than relying solely on page-level navigation.

The following is an illustrative schema pattern, not a claim that it is the production NHOS schema:

```
{
  "contextType": "health-exploration",
```

```
  "sourceApplication": "remedy-revealer",
  "timestamp": "2026-08-21T12:00:00Z",
  "condition": {
    "label": "Example condition",
    "source": "NHOS"
  },
  "selectedItems": [],
  "evidenceReferences": [],
  "userGenerated": true,
  "retention": "local"
}
```

The production schema should define required and optional fields, provenance, timestamps, retention semantics, versioning, validation rules, deletion behavior, and compatibility rules. A future technical specification should document the actual schema used by NHOS.

A future production version of Appendix C — Proposed NHOS Context Interoperability Specification — should formalize the production schema once it has been documented and tested. It should address schema versioning, required and optional fields, provenance, retention and deletion semantics, validation, integrity checking, user visibility, and compatibility across application versions.

12. Application Interoperability

The distinguishing proposition is that an NHOS application is not necessarily an isolated endpoint. A result can become a structured starting point for another application.

A representative composable health-information workflow may proceed from symptom information to condition mapping, from condition mapping to remedy information, from remedy information to interaction review, from interaction review to protocol generation, and from protocol generation to personal tracking.

This is a representative composable health-information workflow, not a clinical pathway, and does not imply that every workflow or recommendation is clinically appropriate.

For research purposes, meaningful interoperability should be operationalized as successful transfer of the required context, correct interpretation by the receiving application, avoidance of specified manual re-entry, preservation of relevant provenance, and successful continuation of the defined workflow.

The research objective is to determine which forms of context exchange are useful, reliable, privacy-preserving, and technically sustainable.

13. Local Persistence and IndexedDB

IndexedDB is an implementation-level browser storage technology that can provide persistent client-side storage for selected NHOS state and user-controlled information. It should not be treated as synonymous with the NHOS architecture.

The research program distinguishes the architectural objective—local, user-controlled continuity—from the particular persistence mechanism used to implement it. Future implementations may use additional or alternative local mechanisms where appropriate.

Key questions include storage reliability, schema evolution, migration, quota behavior, corruption recovery, deletion semantics, browser compatibility, and the boundaries between persistent context and transient application state.

14. Privacy Model

The privacy objective is data minimization: information should remain local when local processing is sufficient for the requested operation, and unnecessary transmission or centralized retention should be avoided.

- Minimize network dependency for workflows that can execute locally.
- Minimize transmission of user-generated health context.
- Minimize centralized retention of user-generated context.
- Provide clear local deletion and reset mechanisms.
- Separate application functionality from unnecessary account requirements.
- Make privacy properties of each workflow measurable rather than relying solely on descriptive claims.

The architecture does not make an absolute claim that all possible information processed by every future NHOS component can never leave a device. Such claims must be established through implementation-specific inspection and testing.

Accordingly, privacy-preserving is used here to describe the architectural objective and design orientation; workflow-specific privacy properties remain empirical questions.

15. Representative Interconnected Workflows

15.1 Natural-Health Exploration

Symptom → Condition Mapping → Remedy Revealer → Interaction Checker → Evidence/Citations → Protocol Generator

15.2 Evidence and Communication

Health communication → Integrity Checker™ → structured findings → recommendations → locally retained review

15.3 Personal Health Continuity

User observation → relevant NHOS tool → structured result → local context → subsequent tracking or review

15.4 Research Demonstration

Controlled input → local processing → measured outputs → persistence test → cross-application continuation → validation metrics

16. Current Implementation Status

These labels are a research-management framework and should be updated only when supported by implementation inspection or documented testing.

Capability	Status / Evidence State
Specialized NHOS applications	Implemented
Local persistence	Implemented
Cross-application context	Implemented / Preliminary Evidence — formal measurement required
Offline/local-first workflows	Implemented / Validation Planned — workflow-specific testing required
Network and payload auditing	Validation Planned
Cross-browser continuity testing	Validation Planned
Formal context schema specification	Active R&D
Human-factors evaluation	Future Research / Validation Planned
End-to-end privacy measurement	Validation Planned

17. Proposed Validation Framework

The architecture should be evaluated through reproducible experiments rather than descriptive claims alone.

Dimension	Representative measure	Validation objective
Local processing	Workflow completion without network dependency	Determine which workflows can execute locally
Network activity	Requests generated during representative workflows	Characterize network dependence
Data transmission	Bytes and payload categories transmitted	Evaluate unnecessary transmission
Persistence	Successful write/read/recovery tests	Measure local state reliability
Context continuity	Successful handoffs between applications	Measure interoperability
Offline capability	Completion rate under disconnected conditions	Evaluate offline resilience
Latency	End-to-end processing and handoff time	Measure performance cost
Storage	Local storage footprint	Characterize resource

		requirements
Reliability	Failure/recovery rate	Identify operational failure modes
Browser compatibility	Cross-browser test matrix	Determine supported environments
Privacy	Information leaving the device by workflow	Validate data-minimization objectives
Usability	Task completion time, errors, and user effort	Assess practical workflow value

18. Experimental Design

- Define a fixed set of representative cross-application workflows.
- Instrument each workflow for network activity, storage events, timing, and context handoffs.
- Run each workflow online and offline where technically applicable.
- Repeat tests across supported browsers and representative device classes.
- Record exact software versions, datasets, schemas, and test conditions.
- Compare isolated-tool workflows with interconnected workflows where appropriate.
- Document failures and recovery behavior rather than excluding unfavorable observations.
- Publish only validated measurements; label untested capabilities as proposed or under validation.

A reproducible test record should identify a test ID, browser and device, application versions, workflow, input dataset, expected behavior, observed behavior, network activity, persistence observations, context handoff result, and pass/fail outcome.

19. Research Status Framework

- **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.

This taxonomy is intended as a consistent NHOS research convention for future white papers, technical reports, and validation documents.

Guiding principle: Build transparently. Measure rigorously. Report proportionally.

20. Threats, Risks, and Limitations

- Browser storage limits and implementation differences may affect persistence.
- Local state can be lost through device failure, browser data clearing, or user action.
- Cross-application context can introduce unintended information propagation if schemas and boundaries are poorly defined.

- Local processing does not automatically establish correctness, safety, or clinical validity.
- Health-information workflows can carry safety implications even when computation is local.
- Interoperability may become more complex as the number of applications and schemas grows.
- Offline capability may reduce access to newly updated external information.
- Privacy claims must be tested against actual network behavior and deployed code.
- Context schemas may require migration and compatibility mechanisms as applications evolve.
- Local-first operation does not eliminate all security threats; device, browser, application, and supply-chain risks remain relevant.

21. Relationship to NIME™

The NHOS Local Intelligence Architecture and NIME™ are complementary research programs. NIME™ addresses the intelligence and reasoning architecture used to organize and process health information. This program addresses how multiple applications can participate in a privacy-preserving local environment and maintain useful continuity across workflows.

Together, the programs provide two complementary questions: how health-information intelligence is performed, and how that intelligence can be composed across applications while minimizing unnecessary data movement.

22. Relationship to the Integrity Checker™

The Health-Tech Copywriting Integrity Checker™ provides a practical research demonstrator for the broader NHOS architecture. It can demonstrate local assessment, structured scoring, recommendations, report generation, citation handling, and locally retained review state.

External evaluation can provide empirical observations that inform refinement of both the application and the underlying NIME™ and local-intelligence research programs. External evaluation results should remain subject to applicable permissions and confidentiality requirements.

23. Commercialization and Platform Implications

If validated, the architecture could support a family of specialized health-information applications that share a common local-first environment without requiring each application to reproduce the same data-management and privacy mechanisms independently.

Potential advantages include reduced duplication of user input, reusable local context, consistent privacy controls, modular application development, and clearer pathways for adding applications to the ecosystem. These are prospective benefits and should be evaluated empirically before being represented as demonstrated outcomes.

24. Future Research

- Formal interoperability specification and versioned context schemas.
- Cross-browser persistence and migration benchmarks.
- Automated network and payload auditing.
- Privacy threat modeling for cross-application context exchange.
- Human-factors studies comparing isolated and interconnected workflows.

- Performance benchmarking across representative device classes.
- Reliability testing under storage pressure, browser restarts, and interrupted workflows.
- Formal evaluation of context provenance and user visibility.
- Research into secure local synchronization where explicitly required by the user.
- Longitudinal evaluation of whether continuity materially improves workflow efficiency.
- Scholarly literature synthesis and comparative evaluation against related architectural approaches.

25. Conclusion

NHOS Labs, Inc. is investigating a model in which health-information applications remain specialized at the interface level while participating in a common local-first environment at the architectural level. The objective is not to eliminate application boundaries, but to make those boundaries interoperable where useful while preserving user control and minimizing unnecessary data movement.

The central proposition remains to be tested: meaningful cross-application health-information continuity may be achievable through local processing, structured context exchange, and local persistence without requiring routine centralized retention of user-generated health context.

The research program therefore moves the NHOS concept beyond a collection of tools toward a measurable architectural question: whether an interconnected health intelligence environment can deliver useful continuity while treating privacy and data minimization as architectural constraints.

Appendix A — Terminology

- **NHOS —** Natural Health Operating System.
- **NIME™ —** NHOS Intelligence Matrix Engine™.
- **Local-first —** An architecture that prioritizes local execution and local data handling while permitting network functionality when explicitly required.
- **Context —** Selected structured information generated or retained during a user's interaction with the system.
- **Cross-application continuity —** The ability for relevant context from one application workflow to support a subsequent workflow.
- **Local persistence —** Retention of selected information on the user's device.
- **IndexedDB —** A browser API for persistent client-side structured storage; an implementation mechanism rather than the complete NHOS architecture.
- **Meaningful interoperability —** For this research program, successful transfer of required context, correct interpretation by the receiving application, preservation of relevant provenance, and successful continuation of a defined workflow with no unnecessary manual re-entry.

Appendix B — Evidence Discipline

This white paper intentionally distinguishes implementation from hypothesis. Existing platform functionality may be described as implemented where it is observable in the production system. Performance, privacy, reliability, interoperability, and user-benefit claims should be treated as hypotheses or validation targets until supported by documented tests.

Guiding principle: Build transparently. Measure rigorously. Report proportionally.

Appendix C — Proposed NHOS Context Interoperability Specification

This appendix defines the requirements for a future production context specification. It is not itself the production schema. The specification should be published only after implementation inspection and testing.

Specification area	Requirement to define
Schema version	Explicit version identifier and compatibility policy
Context type	Controlled vocabulary or extensible classification
Source application	Originating application identifier and version
Timestamp	Creation/update time semantics and format
Payload	Required and optional context fields
Provenance	Origin and transformation history where relevant
User-generated state	Clear distinction between user input and system-derived content
Retention	Local retention semantics and lifecycle
Deletion	User-visible deletion/reset behavior
Validation	Schema and field validation rules
Integrity	Mechanisms for detecting unintended alteration where appropriate
Visibility	How users can inspect relevant retained context
Compatibility	Behavior when source and receiving applications use different schema versions

Appendix D — Current Implementation Evidence Record Template

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

- Test ID and date.
- Browser, operating system, and representative device.
- NHOS application names and exact versions/commit identifiers where available.
- Workflow and test objective.
- Input dataset or controlled test case.
- Expected result.
- Observed result.
- Network requests and payload categories.
- Local persistence/read/write observations.
- Context handoff observations.
- Offline/online condition.
- Latency and resource measurements where applicable.

- Failure and recovery behavior.
- Pass/fail determination and reviewer notes.

References and Related NHOS Technical Work

The following references establish the NHOS technical context for this white paper. A future peer-reviewed manuscript should expand Section 3 and this list with independent scholarly literature on local-first software, edge computing, privacy engineering, personal health records, health-information exchange/interoperability, and related distributed architectures.

- NHOS Labs, Inc. — NHOS Intelligence Matrix Engine™ Architecture White Paper, Version 1.4.0, 2026.
- NHOS Labs, Inc. — NHOS platform, research, and technical materials. https://nhos.health/

Publication Status

Technical Research White Paper • Version 1.2 • August 2026
Research Stage: Architectural Model & Validation Framework
Final architectural white-paper version; subsequent major revision should be driven by empirical evidence.

NHOS Local Intelligence Architecture

Technical Research White Paper · Version 1.2 · August 2026

NHOS Labs, Inc.

NHOS: https://nhos.health

NHOS Research Registry: https://nhos.health/research

Local Intelligence Architecture: https://nhos.health

Permanent DOI (Version): https://doi.org/10.5281/zenodo.22070736

Cite All Versions: https://doi.org/10.5281/zenodo.22070736