

Non-Custodial AI Memory

A Local-First Architecture for Consent-Gated Context Hydration in Longitudinal Health AI

Technical Whitepaper

Decoupling persistent personal state from AI-provider custody while preserving cross-session continuity, provenance, and user-controlled authorization.

Kenneth J. Clark, Engineer

Enigma Genetics (Rochester, NY, USA)

21 August 2026

Version 1.0

Publication scope. This paper presents a systems architecture and evaluation framework. It is a conceptual technical contribution and does not report completed clinical validation, regulatory approval, or implementation-specific proof that any implementation prevents all provider-side retention or inference-time disclosure.

Abstract

Longitudinal artificial-intelligence systems benefit from persistent memory, yet persistence changes the privacy and security model of the application: conversational history, health records, preferences, and inferred attributes can become durable state held by an AI service or its memory layer. Recent work has demonstrated that agent memory can be a direct target for extraction attacks, while personal-data-store and local-first research has argued for architectures in which the individual, rather than the application provider, remains the authoritative custodian of personal data. This paper proposes Non-Custodial AI Memory (NCAM), a local-first architecture in which persistent personal state resides in a user-controlled data substrate and is made available to a fresh AI session only through a consent-gated Local Context Broker. The broker authenticates the user, evaluates purpose and scope, resolves only authorized objects, constructs a minimum-necessary session context view, and issues ephemeral read authority bound to a session identifier and expiration. Persistent updates are not written directly by the model: the AI proposes changes that cross a separate write-back gate, where user or policy approval, provenance, versioning, and audit controls are applied. The architecture therefore separates four functions that are often collapsed in contemporary agent systems: storage, authorization, context construction, and inference. We define security invariants, a threat model, a formal grant structure, and measurable evaluation criteria including Context Exposure Ratio, Ambient Authority Ratio, Fresh-Session Reconstruction Fidelity, Provenance Coverage, and Revocation

Enforcement Latency. The approach is particularly relevant to health and genomic AI, where data is longitudinal, highly identifying, and useful across many models and care contexts. NCAM does not eliminate inference-time disclosure to a remote model; rather, it changes persistent custody from a model-centric default into a user-controlled, purpose-bound systems decision.

Keywords: local-first software; AI agent memory; personal data stores; health AI; genomic privacy; dynamic consent; zero trust; provenance; least privilege; user-controlled storage

Core contribution

Persistent personal truth is not the AI model's memory. It is a user-controlled data asset. The model receives an authorized session view of that asset, and persistence requires a separate governed write transaction.

1. Introduction

Large language model (LLM) products increasingly use cross-session memory to improve personalization and continuity. That design is operationally attractive, but it also creates a new high-value repository: a persistent, queryable representation of a person. Wang et al. demonstrated black-box memory extraction attacks against representative LLM agents, establishing that agent memory is itself an attack surface rather than a benign convenience [14]. Emerging 2026 literature similarly frames long-term agent memory as a lifecycle-governance problem spanning write, storage, retrieval, execution, sharing, and forgetting [15].

The privacy problem is amplified in health. A longitudinal health AI may need medical records, laboratory data, medications, family history, wearable observations, genomic files, user goals, and prior interpretations. These data are valuable precisely because they compound over time. A conventional design therefore tends to make the AI application's account database or vector store the authoritative state of the person. This couples continuity to the vendor that stores the memory.

A different architecture is possible. Local-first software reverses the normal cloud hierarchy: the user-controlled copy is authoritative and remote infrastructure, when used, is secondary [2]. Personal Data Store (PDS) research extends this principle to sensitive personal information, explicitly targeting individual control over collection, storage, management, and sharing [3-5]. Dynamic-consent research adds the notion that authorization must be revisable over time and capable of traveling with the data or its permitted uses [7-9].

This paper composes those lines of work around a specific AI systems problem: preserving longitudinal personalization without making model-provider memory the canonical repository of the person. It extends the non-custodial identity and genomic-consent thesis introduced in an earlier Enigma Genetics technical note [1] from biological identity into AI context continuity.

2. Problem Definition and Design Objective

The target system should allow a user to start a new AI conversation, authenticate, authorize access to a user-selected directory or personal data store, reconstruct a high-fidelity current profile, use only the information required for the task, and end the session without requiring the AI provider to maintain the authoritative longitudinal record.

The design objective is therefore not “zero data leaves the device.” That stronger property is achievable only with local inference or suitable confidential-computing arrangements. The objective is narrower and more operationally useful: eliminate provider-controlled persistent personal state as a required dependency, reduce disclosure to task-relevant context, and make every persistent update an explicit governed operation.

2.1 Architectural properties

- User authority: the user selects the persistent storage location and can move, back up, disconnect, or replace it without migrating an AI vendor account.
- Fresh-session semantics: each AI session begins without relying on vendor-managed longitudinal memory as authoritative truth.
- Purpose-bound hydration: personal data is retrieved only after authentication and consent for a defined purpose, scope, duration, and action set.
- Least disclosure: the model receives a session view rather than ambient access to the complete personal corpus.
- Separate write authority: model output cannot silently become persistent truth.
- Provenance: material facts and derived profile elements retain source and version lineage.
- Model portability: the same persistent personal state can hydrate different model runtimes without rewriting the underlying profile.
- Fail-closed behavior: ambiguous identity, scope, consent, source integrity, or write authority results in denial rather than permissive fallback.

3. Related Work

3.1 Local-first software and data ownership

Kleppmann et al. define local-first software as software that treats the local copy as primary rather than merely a cache subordinate to the server. Their seven ideals include offline operation, longevity, security and privacy, and ultimate user ownership and control [2]. NCAM borrows the authority inversion but applies it to AI memory: the persistent personal substrate is primary; model context is a derived, temporary view.

3.2 Personal data stores and personal health data spaces

PDS literature describes architectures in which individuals collect, control, store, and manage personal data rather than surrendering that function to each service provider [3]. Health-specific work argues that PDS ecosystems can provide meaningful control, visibility, and selective sharing across care and research contexts [4]. The

openPDS/SafeAnswers work further demonstrates an important architectural idea: services can receive computed answers rather than unrestricted high-dimensional source data, converting part of the privacy problem into a constrained security interface [5]. More recent personal health data-space reference architectures similarly center data-subject control and continuity of care [17].

3.3 Dynamic consent

Kaye et al. describe dynamic consent as a personalized digital interface through which individuals can tailor and modify consent choices over time, with consent preferences associated with data as it is reused [7]. A decade of experience in the CHRIS study provides evidence that granular, revisable consent is operationally feasible in longitudinal research, while also highlighting communication and usability as central implementation issues [8]. ConsentChain demonstrates machine-readable consent elements and limited-access tokens in a genomic-sharing architecture [9]. NCAM applies these principles to AI context retrieval and persistence.

3.4 Agent memory privacy

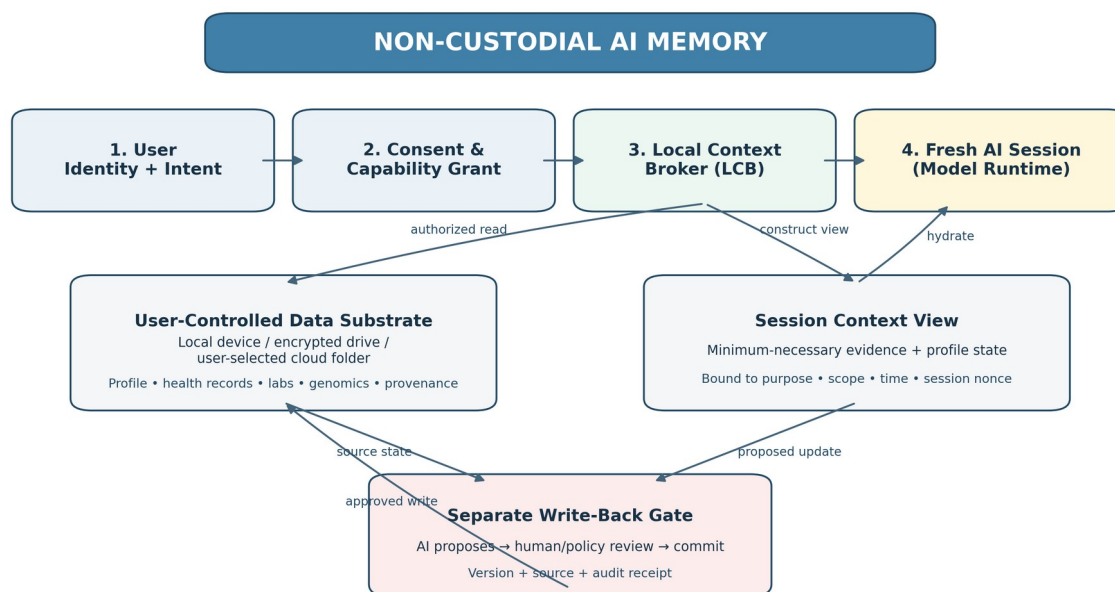
Cross-session memory improves continuity but creates persistence, statefulness, and propagation risks. The ACL 2025 MEXTRA work shows that private information stored in agent memory can be extracted through adversarial prompting [14]. A 2026 security survey argues that durable memory requires lifecycle governance anchored in provenance, versioning, retention, rollback, and policy rather than defenses applied only at retrieval time [15]. Very recent privacy-aware memory research likewise emphasizes selective retrieval of exact private values only when task requirements and consent justify exposure [16]. These results support an architectural separation between persistent user state and model runtime memory.

3.5 Zero trust, least privilege, and delegated authority

NIST Zero Trust Architecture rejects implicit trust based on network location and emphasizes continuous authentication, authorization, and minimum access to resources [10]. The NIST Privacy Framework further identifies selective disclosure, local processing, decentralization, authorization, deletion, and data-minimizing audit as privacy-engineering controls [11]. Macaroons show how delegated credentials can be attenuated by caveats that constrain where, when, by whom, and for what purpose a request is authorized [12]. NCAM uses the same design logic for session-bound file and object access.

3.6 Provenance

W3C PROV defines interoperable concepts for entities, activities, agents, derivation, and responsibility [13]. In NCAM, provenance is not ancillary metadata: it is required to distinguish source records from AI-generated interpretations and to prevent unsupported session output from silently becoming the next session's truth.



Persistent truth stays with the user. AI receives session-bound authority, not ambient custody.

Figure 1. Reference architecture for Non-Custodial AI Memory. Persistent user state remains outside the model runtime; a Local Context Broker creates an authorized, session-specific view.

4. System Model

NCAM separates the system into six logical components. Implementations may combine components physically, but their trust roles should remain distinct.

COMPONENT	RESPONSIBILITY
User / principal	Authenticates, declares intent, grants or denies access, approves material persistent changes.
Personal Data Substrate (PDS)	User-selected persistent storage: local disk, encrypted external storage, NAS, user-controlled cloud drive, Solid-style pod, or equivalent. Holds canonical profile plus source objects.
Local Context Broker (LCB)	Trusted intermediary that authenticates requests, evaluates grants, resolves objects, filters/minimizes data, builds session context, and mediates writes.
Consent / policy engine	Represents purpose, object scope, permitted operations, time limits, revocation state, and optional sensitivity constraints.
AI session runtime	Receives the hydrated context view, reasons over it, and produces answers or proposed updates. It is not the authoritative memory store.

COMPONENT	RESPONSIBILITY
Provenance / audit layer	Records source hashes, profile versions, grant references, material derivations, approved writes, revocations, and session receipts.

4.1 Persistent data model

A practical substrate should not be a single opaque “memory file.” It should distinguish authoritative source objects from derived operating state. A minimal layout can contain:

- profile.json or profile.md - current user-confirmed facts, preferences, goals, active medications, important conditions, care-team context, and unresolved questions;
- manifest.json - object identifiers, paths, MIME types, hashes, sensitivity labels, dates, and provenance links;
- sources/ - raw or normalized records such as FHIR bundles, PDFs, laboratory exports, genomics files, wearable summaries, and user-authored notes;
- derived/ - AI-generated summaries and analyses explicitly marked as derived, with source references and model/version metadata;
- policies/ - consent grants, revocations, purpose restrictions, retention policies, and sharing rules;
- audit/ - append-oriented receipts for reads, writes, profile updates, and policy changes.

Healthcare interoperability is improved when source material is normalized into standard resource structures where practical. HL7 FHIR R4 provides normative REST, JSON/XML, terminology, and key Patient and Observation resources that can act as one interchange layer without dictating the user’s physical storage location [20].

4.2 Grant structure

For a session σ , the broker issues or internally evaluates a grant $G\sigma$ with at least the following fields:

$$G\sigma = \langle \text{subject}, \text{session_id}, \text{object_scope}, \text{operations}, \text{purpose}, \text{issued_at}, \text{expires_at}, \text{nonce}, \text{policy_version} \rangle$$

The grant is intentionally narrower than an account login. Authentication answers “who is this?”; the session grant answers “what may this specific session do, to which objects, for what purpose, and until when?”

4.3 Context hydration

The Local Context Broker creates a context view $H\sigma(q)$ for a query or session objective q . The broker should minimize the material released across the inference boundary while preserving sufficient evidence for the requested task:

$$H\sigma(q) = \text{Filter}(\text{Resolve}(\text{object_scope}), \text{purpose}=q, \text{policy}=G\sigma, \text{minimum_necessary}=\text{True})$$

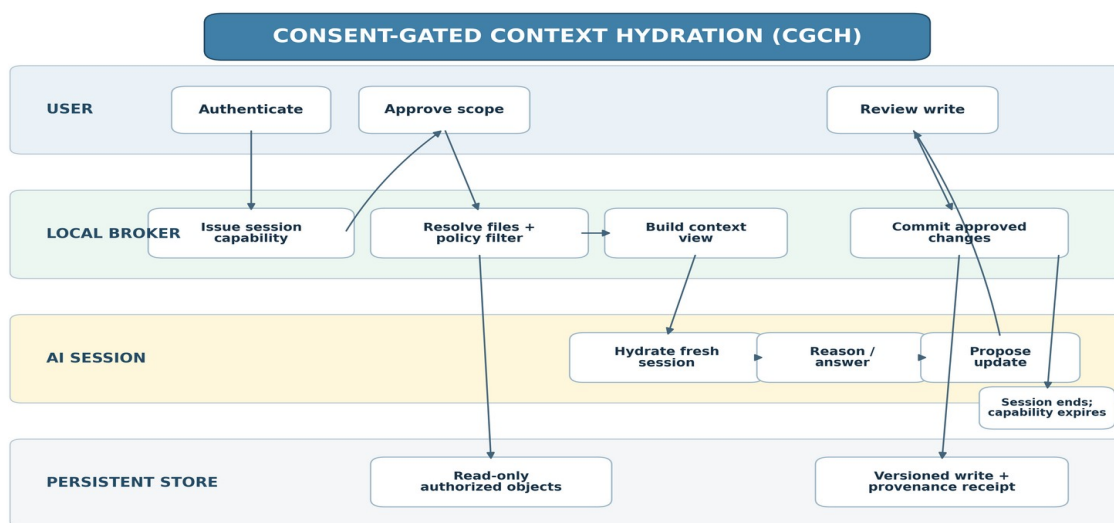
In a low-risk workflow, $H\sigma$ may contain only the compact profile plus a few relevant recent records. In a complex genomic or longitudinal analysis, the authorized view may be much larger. The architectural requirement is not an arbitrary token cap; it is that scope is explicit and the model cannot silently widen it.

4.4 Write-back transaction

Read continuity and persistent memory formation must be separated. The model may propose an update ΔS , but the broker should not treat the proposal as authoritative. A persistent write becomes eligible only after validation and approval:

Commit(ΔS) only if Approve(subject/policy, ΔS) $\wedge$ Provenance(ΔS) $\wedge$ ValidSchema(ΔS)

This prevents prompt injection, model error, transient user statements, or mistaken inferences from automatically contaminating future sessions. High-risk fields can require stronger rules than low-risk preferences; for example, a medication change inferred from conversational text should not overwrite a verified medication list without explicit confirmation or a trusted source.



Read authority is ephemeral. Write authority is separate. The persistent profile remains outside the model runtime.

Figure 2. Consent-Gated Context Hydration (CGCH). Each session receives ephemeral read authority; persistent modification crosses a distinct approval and provenance boundary.

5. Consent-Gated Context Hydration Protocol

1. Fresh session creation. The application creates a new session identifier and does not preload vendor-managed longitudinal memory as authoritative profile state.
2. Authentication. The user authenticates to the Local Context Broker using the implementation's chosen identity mechanism.

3. Intent declaration. The application states the requested purpose, e.g., appointment preparation, medication review, record explanation, genomic interpretation, or general profile Q&A.
4. Scope presentation and consent. The broker presents the data categories or objects required and obtains a grant that is purpose-, action-, and time-bounded.
5. Object resolution. The broker resolves only objects within the grant and checks path boundaries, hashes/version state, sensitivity labels, and revocation rules.
6. Local preprocessing. Where feasible, parsing, de-identification, summarization, feature extraction, or query planning occurs locally so that raw data need not cross the inference boundary unnecessarily.
7. Context construction. The broker assembles a session context view with profile facts, source excerpts or structured observations, provenance references, and explicit uncertainty or missing-data indicators.
8. Inference. The AI runtime receives the view and performs the requested reasoning. It has no independent authority to enumerate the user's storage substrate.
9. Proposed persistence. If the session produces a material fact, preference, correction, or derived summary worth retaining, it is emitted as a structured proposed change rather than written directly.
10. Write authorization and commit. The user or policy engine approves, rejects, edits, or defers the proposed change. Approved changes are versioned, linked to sources, and recorded in the audit layer.
11. Termination. Session capabilities expire; temporary context is cleared according to runtime policy; the next session reconstructs state from the user-controlled substrate.

6. Security and Governance Invariants

I1 - Custody independence: Longitudinal continuity must remain possible if the model provider, application vendor, or inference model changes.

I2 - No ambient file authority: The model runtime cannot freely enumerate or traverse the user's data substrate.

I3 - Session binding: Every data grant is bound to a session or explicitly defined reusable delegation, with expiration and revocation semantics.

I4 - Purpose limitation: Authorization for one workflow does not implicitly authorize unrelated uses.

I5 - Minimum necessary disclosure: The broker should expose only the context required for the approved task, with local preprocessing preferred when it reduces disclosure without destroying utility.

I6 - Read/write separation: Read authority does not imply write authority. Persistent changes require a separate decision path.

I7 - Provenance before persistence: Material derived state must retain source lineage, derivation type, and version metadata before becoming future context.

I8 - Human or policy confirmation for high-impact state: Clinical, identity, financial, legal, or other high-risk persistent facts require stronger admission rules than low-risk conversational preferences.

I9 - Revocation enforcement: Revoked or expired grants must fail closed at the broker, regardless of prior session success.

I10 - Honest inference boundary: If a remote model receives plaintext context, the architecture must not claim that the provider never saw that context. Custody separation is not equivalent to confidential inference.

7. Threat Model

The architecture assumes that the user intentionally authorizes some data processing. It aims to reduce persistent exposure and unauthorized authority, not to make authorized computation magically data-free.

THREAT	FAILURE MODE	PRIMARY MITIGATION
Memory extraction from the AI layer	A malicious or manipulated prompt elicits stored private information.	Do not use provider memory as the canonical store; hydrate session-specific views; minimize scope; expire authority.
Over-broad connector access	An intermediary or plugin can read substantially more than the current task requires.	Object-scoped grants; path confinement; explicit purpose; least privilege; local query planning.
Prompt-driven write poisoning	A malicious record or model output persists false facts into future sessions.	Separate write gate; provenance; schema validation; source ranking; confirmation for material changes.
Confused deputy	An authorized app uses its authority on behalf of an unauthorized request.	Bind capability to subject, session, purpose, action, object scope, nonce, and expiration.
Stale profile state	Old facts are repeatedly rehydrated after newer evidence exists.	Versioning; supersession links; current-state profile separated from source history; contradiction checks.
Provider retention outside the memory feature	A remote inference provider retains prompts or logs under its own operational policy.	Use zero-retention/enterprise controls where available, confidential inference, or local models; do not overclaim architecture-level prevention.
Local device compromise	Malware or a stolen unlocked device reads local health files.	OS/device encryption, secure enclave/key store, access control, backups, endpoint security; residual risk remains.
Availability / loss	User-controlled local data is deleted, corrupted, or inaccessible.	Encrypted backups, replicated user-controlled copies, integrity hashes, recovery plans; local-first ownership carries operational responsibility.

8. Architectural Comparison

MODEL	CANONICAL CUSTODY	PERSISTENT PROVIDER EXPOSURE	USER CONTROL	MODEL PORTABILITY	WRITE GOVERNANCE	NOTES
Provider-native persistent agent memory	Provider/service	High	Low-Med	Provider dependent	Often implicit	High continuity; creates concentrated persistent memory attack surface.
Cloud RAG + vendor vector database	Application vendor	High	Medium	Medium	Application controlled	Good retrieval; canonical state still vendor-custodial.
Local-only LLM + local files	User	None	High	High	Local application controlled	Strong custody; may limit model capability, device support, and interoperability.
Personal Data Store / Solid-style application	User / chosen pod	Low-Med	High	High	Explicit authorization	Strong data agency; not inherently an AI-memory governance protocol.
NCAM / CGCH (this work)	User / chosen substrate	Task-bounded	High	High	Separate read and write grants	Composes local-first custody, dynamic authorization, provenance, and fresh-session AI hydration.

9. Evaluation Framework

Validation of NCAM requires evaluation as a systems architecture, not only by chatbot answer quality. A suitable experiment starts each trial with a fresh model session, reconstructs state from a controlled personal corpus, and measures both utility and authority boundaries.

METRIC	DEFINITION / INTERPRETATION
Context Exposure Ratio (CER)	Amount of user data crossing the inference boundary divided by the eligible corpus size for that session. Lower is generally better at equal task quality.
Ambient Authority Ratio (AAR)	Objects/actions technically reachable by the runtime but not required by the declared task divided by total reachable authority. Target: near zero.
Fresh-Session Reconstruction Fidelity (FSRF)	Validated required profile facts correctly recovered after a new session divided by the expected facts required for the benchmark task.

METRIC	DEFINITION / INTERPRETATION
Provenance Coverage (PC)	Material persistent facts or derived assertions with resolvable source lineage divided by total material persistent facts/assertions.
Revocation Enforcement Latency (REL)	Elapsed time between revocation and denial of subsequent unauthorized access attempts.
Unauthorized Persistence Rate (UPR)	Unapproved model-generated changes that appear in later fresh sessions divided by all proposed persistent changes. Target: zero.
Hydration Latency	Time from successful consent to availability of the task-ready context view; reported by corpus size and storage medium.
Cross-Model Portability	Difference in reconstruction fidelity when the same user substrate is hydrated into multiple model families under equivalent context budgets.

9.1 Validation protocol

1. Construct a synthetic or fully consented longitudinal health corpus containing structured records, PDFs, medication changes, conflicting observations, and genomic artifacts.
2. Define benchmark tasks with a known minimum evidence set and known profile truth.
3. Run each task under (a) provider-native persistent memory, (b) cloud RAG, and (c) NCAM fresh-session hydration where technically comparable.
4. Measure task accuracy, CER, AAR, FSRF, PC, REL, UPR, hydration latency, and cost.
5. Inject adversarial documents and prompt attempts intended to widen read scope or force persistent false writes.
6. Revoke selected grants mid-experiment and verify fail-closed behavior.
7. Repeat across at least three model families to test whether persistent personal state is genuinely model-portable rather than implicitly encoded for one provider.

This paper does not report numerical security or performance claims for NCAM. Such claims require execution of this protocol or an equivalent independent evaluation. The metrics are included to make the architecture falsifiable and to define the evidence required for implementation-specific claims.

10. Application to Longitudinal Health and Genomic AI

Health is a high-value test domain because longitudinal usefulness and privacy pressure grow together. Personal health records have long been conceived as more than static repositories: they combine data, knowledge, and tools around the individual [6]. Modern personal health data-space research similarly emphasizes continuity, accessibility, and individual control [17].

Genomics raises the stakes further. Genetic data is durable, identifying, and capable of revealing information about relatives; re-identification and attribute-inference risks have been extensively documented [18]. WHO guidance published in 2024 emphasizes ethical governance, privacy, transparency, and responsible access across the genomic data lifecycle [19]. A non-custodial memory architecture is therefore especially relevant when an AI system uses both genomic evidence and longitudinal clinical context.

A practical health implementation can use the user-controlled folder as a portable health-state directory. On session start, the broker reads a compact current profile, identifies the specific source records necessary for the question, normalizes structured observations where feasible, and exposes evidence to the AI only after the user authorizes the purpose. A new clinician-preparation session tomorrow can reconstruct the same current state without relying on yesterday's model conversation as the record of truth.

For genomic interpretation, the broker may authorize specific variant files or precomputed genomic features while keeping unrelated raw sequence inaccessible. Where exact variants are necessary, the architecture should disclose them deliberately and log their use. Where a lower-dimensional answer is sufficient, openPDS-style “safe answer” patterns provide a useful precedent for computing locally and sharing only the needed result [5].

11. Implementation Profile

The architecture is storage- and model-agnostic, but an initial implementation can be deliberately simple. A user chooses a folder on a local device or user-controlled drive. The application stores only enough account metadata to authenticate and locate the user-approved substrate; the health profile itself remains in that substrate. The broker is responsible for file resolution and policy enforcement, not the LLM.

11.1 Minimum viable implementation

- User-selected root directory with a deterministic manifest and canonical current-profile file.
- OS or drive authorization that scopes the application to that root rather than the user's entire filesystem.
- Session IDs and short-lived access grants.
- Local parsers for profile files and common health formats; FHIR R4 JSON support for interoperable structured records [20].
- Retrieval planner that selects candidate evidence before remote inference.

- Structured context envelope containing provenance IDs and sensitivity tags.
- Structured proposed-write envelope; no direct free-text persistence.
- User-visible review for material profile changes.
- Append-oriented local audit records and cryptographic hashes for source/version integrity.
- Provider configuration that minimizes request retention where available; optional local or confidential inference for higher-sensitivity workflows.

11.2 Stronger implementation

A hardened implementation can add hardware-backed device keys, signed manifests, attested broker execution, encrypted object stores, capability credentials with caveats, provenance receipts, remote-drive replication, offline operation, confidential inference, and eventually local semantic indexes. These mechanisms strengthen the design but are not prerequisites for the architectural separation itself.

12. Limitations and Open Questions

- Inference-time exposure remains. If plaintext personal context is sent to a remote model, that provider can technically observe the transmitted context subject to its security and retention controls. NCAM limits persistent custody and scope; it does not by itself provide confidential inference.
- Endpoint compromise remains material. User-controlled storage shifts authority toward the user but also makes device security, backups, ransomware protection, and recovery first-class requirements.
- Consent UX can become burdensome. Dynamic consent literature shows that granular control must be paired with usable communication and sensible defaults; asking for approval on every trivial read can produce fatigue [7,8].
- Minimum-necessary context selection is itself an intelligent systems problem. Over-minimization harms answer quality; under-minimization increases disclosure. Retrieval policies should therefore be measurable rather than assumed.
- Persistent derived summaries can encode sensitive inferences even if raw source data is not stored in them. Sensitivity labels and provenance must apply to derived artifacts as well as raw files.
- Cross-device synchronization introduces conflict, availability, and key-management problems familiar from local-first systems [2].
- Provider APIs differ in logging, retention, tool-call behavior, and enterprise privacy guarantees. Runtime configuration is part of the threat model and must be disclosed.
- Clinical safety requires a separate validation layer. A secure memory architecture does not establish diagnostic accuracy, clinical utility, or regulatory status.

13. Contribution and Novelty

The contribution of NCAM is compositional rather than the invention of any single primitive. Local-first data authority, personal data stores, dynamic consent, least-privilege capabilities, provenance, and AI retrieval systems all predate this paper. The

contribution of this work is their composition around a specific separation of concerns for longitudinal AI:

- Persistent personal state is defined as a user-controlled data substrate rather than as an internal feature of the AI agent.
- Every fresh session obtains a temporary context view through explicit authorization rather than inheriting an opaque account memory.
- The intermediary is a security principal and policy enforcement point, not just a retrieval library.
- Read hydration and memory write-back are modeled as distinct authorization transactions.
- AI-generated persistent state is treated as a governed derived artifact with provenance, not as automatic truth.
- System evaluation includes authority and privacy metrics in addition to response quality.

This composition creates a testable architecture for “AI that knows the person” without requiring the AI provider to be the durable custodian of the person.

14. Conclusion

Longitudinal AI does not require longitudinal provider custody. The key systems move is to separate the persistence layer from the reasoning layer. In Non-Custodial AI Memory, the person’s data substrate persists; the AI session is disposable. A broker authenticates the user, evaluates consent, constructs a scoped context view, and expires that authority at session end. Any new “memory” must cross a distinct write gate with provenance and approval before it can influence future sessions.

For health and genomic AI, this inversion is consequential. It preserves continuity while reducing dependence on centralized agent memory, creates an enforceable point for consent and minimum-necessary disclosure, improves model portability, and makes the evolution of personal state auditable. It also draws a hard line around claims: local-first custody is not the same as local inference, and no security architecture substitutes for endpoint security, clinical validation, or transparent provider retention controls.

Empirical validation remains the appropriate next stage. The architecture can be tested against a controlled longitudinal corpus and compared with conventional persistent-memory and cloud-RAG designs using the metrics defined in this paper. If it cannot reconstruct the user’s state accurately, constrain ambient authority, enforce revocation, and prevent unauthorized persistence, then the design has failed. If it can, it establishes a practical path toward personalized AI in which continuity belongs to the individual rather than the model vendor.

15. Publication Declarations

Ethics statement. This paper reports a conceptual systems architecture and literature synthesis; it does not report research involving human participants, identifiable patient

data, or animals. Institutional review board or animal-research approval was therefore not applicable to the work reported here.

Data availability. No new empirical dataset is reported in this paper. External evidence is available through the cited publications and standards. The example context and write envelopes in Appendices A and B are illustrative.

Code availability. No executable reference implementation is included in this publication. Implementation-specific security, privacy, clinical, or performance claims require validation of the corresponding implementation.

Competing interests. The author is affiliated with Enigma Genetics, the organization associated with the architecture described in this paper. This affiliation may constitute a potential commercial interest.

Author contributions. Kenneth J. Clark: conceptualization, systems architecture, methodology, literature synthesis, writing, and review.

References

- [1] Clark KJ. A Non-Custodial Architecture for Persistent Biological Identity and Enforceable Genomic Consent. Zenodo. 2026. doi:10.5281/zenodo.18526597.
- [2] Kleppmann M, Wiggins A, van Hardenberg P, McGranaghan M. Local-first software: You own your data, in spite of the cloud. *Proc ACM SIGPLAN Onward!* 2019:154-178. doi:10.1145/3359591.3359737.
- [3] Fallatah KU, Barhamgi M, Perera C. Personal Data Stores (PDS): A Review. *Sensors*. 2023;23(3):1477. doi:10.3390/s23031477.
- [4] Carmichael L, Hall W, Boniface M. Personal data store ecosystems in health and social care. *Front Public Health*. 2024;12:1348044. doi:10.3389/fpubh.2024.1348044.
- [5] de Montjoye YA, Shmueli E, Wang SS, Pentland AS. openPDS: protecting the privacy of metadata through SafeAnswers. *PLoS One*. 2014;9(7):e98790. doi:10.1371/journal.pone.0098790.
- [6] Tang PC, Ash JS, Bates DW, Overhage JM, Sands DZ. Personal Health Records: Definitions, Benefits, and Strategies for Overcoming Barriers to Adoption. *J Am Med Inform Assoc*. 2006;13(2):121-126. doi:10.1197/jamia.M2025.
- [7] Kaye J, Whitley EA, Lund D, Morrison M, Teare H, Melham K. Dynamic consent: a patient interface for twenty-first century research networks. *Eur J Hum Genet*. 2015;23:141-146. doi:10.1038/ejhg.2014.71.
- [8] Mascalzoni D, Melotti R, Pattaro C, et al. Ten years of dynamic consent in the CHRIS study: informed consent as a dynamic process. *Eur J Hum Genet*. 2022;30:1391-1397. doi:10.1038/s41431-022-01160-4.
- [9] Albalwy F, Brass A, Davies A. A Blockchain-Based Dynamic Consent Architecture to Support Clinical Genomic Data Sharing (ConsentChain): Proof-of-Concept Study. *JMIR Med Inform*. 2021;9(11):e27816. doi:10.2196/27816.

- [10] Rose S, Borchert O, Mitchell S, Connelly S. Zero Trust Architecture. NIST Special Publication 800-207. National Institute of Standards and Technology; 2020. doi:10.6028/NIST.SP.800-207.
- [11] National Institute of Standards and Technology. NIST Privacy Framework: A Tool for Improving Privacy through Enterprise Risk Management, Version 1.0. 2020.
- [12] Birgisson A, Politz JG, Erlingsson U, Taly A, Vrabie M, Lentczner M. Macaroons: Cookies with Contextual Caveats for Decentralized Authorization in the Cloud. NDSS. 2014.
- [13] Moreau L, Groth P, eds. PROV-Overview: An Overview of the PROV Family of Documents. W3C Working Group Note. 2013.
- [14] Wang B, He W, Zeng S, Xiang Z, Xing Y, Tang J, He P. Unveiling Privacy Risks in LLM Agent Memory. Proc 63rd Annual Meeting of the Association for Computational Linguistics. 2025:25241-25260. doi:10.18653/v1/2025.acl-long.1227.
- [15] Lin Z, Hao X, Fu R, et al. A Survey on Long-Term Memory Security in LLM Agents: Attacks, Defenses, and Governance Across the Memory Lifecycle. arXiv:2604.16548. 2026. Preprint.
- [16] Wang W, Si W, Xu X, Xu Y. What to Remember, What to Reveal: Privacy-Aware Memory for Conversational Agents. arXiv:2608.16551. 2026. Preprint.
- [17] Klementi T, Piho G, Ross P. A reference architecture for personal health data spaces using decentralized content-addressable storage networks. Front Med (Lausanne). 2024;11:1411013. doi:10.3389/fmed.2024.1411013.
- [18] Erlich Y, Narayanan A. Routes for breaching and protecting genetic privacy. Nat Rev Genet. 2014;15:409-421. doi:10.1038/nrg3723.
- [19] World Health Organization. Guidance for human genome data collection, access, use and sharing. Geneva: WHO; 2024. ISBN 978-92-4-010214-9.
- [20] Health Level Seven International. HL7 FHIR Release 4, Version 4.0.1. 2019. <https://hl7.org/fhir/R4/>.

Appendix A. Example Session Context Envelope

The following conceptual structure illustrates the boundary between persistent state and a model-visible session view. It is intentionally implementation-neutral.

```
{
  "session_id": "uuid",
  "purpose": "appointment_preparation",
  "grant": {
    "objects": ["profile.current", "labs.last_12_months", "medications.active"],
    "operations": ["read"],
    "expires_at": "2026-08-20T18:30:00Z"
  },
  "profile_view": { "...": "minimum task-relevant facts" },
  "evidence": [
    {"object_id": "lab-2026-07-14", "source_hash": "...", "excerpt": "..."}
  ],
  "constraints": {
    "no_persistent_write": true,
    "uncertainty_required": true,
    "source_citation_required": true
  }
}
```

Appendix B. Example Proposed Write Envelope

```
{
  "proposal_id": "uuid",
  "target": "profile.current.medications",
  "operation": "propose_update",
  "candidate_value": "...",
  "basis": ["source-object-id", "conversation-turn-id"],
  "confidence": "candidate",
  "requires_user_confirmation": true,
  "model": "provider/model/version",
  "status": "pending"
}
```