WHY: A Local-First Cognitive Operating System for Personal Intelligence

June 4, 2026

Abstract

Today's AI systems are increasingly competent but still discontinuous. They can reason over text, images, tools, and code, yet they rarely inherit the full temporal context of a person's life. They do not naturally remember what changed yesterday, why a decision mattered last month, which relationships shape a project, or which constraints should silently govern the next action. The result is an intelligence that can answer questions but cannot yet become a durable operating layer for a person.

WHY proposes a local-first cognitive operating system for personal intelligence. Its central claim is that useful autonomy requires a machine to become time-native around its owner: it must index the user's digital world, convert activity into an event-grounded semantic graph, preserve identity and provenance over time, and coordinate agents that can retrieve context, act under constraints, and leave auditable records. The unit of computing becomes the goal rather than the app.

The architecture is not merely a larger model or a memory plugin. It is a personal temporal substrate: local events, semantic memory, action traces, and permissioned inference slices compose into a system that remembers, adapts, and acts while preserving sovereignty. When many such nodes opt in, the web shifts from link retrieval to inference exchange: machines answer from public knowledge and permissioned private context without requiring raw personal data to leave its owner.

1 Introduction

The personal computer gave individuals control over computation. The web gave them access to documents and services. Mobile computing made software ambient but also fragmented personal life across apps, accounts, feeds, and notifications. Artificial intelligence now introduces a different possibility: a computer that remembers the person's world, understands the user's goals, and acts across tools without forcing the user to manually operate every interface.

This paper uses WHY to denote a local-first cognitive operating system for personal intelligence. WHY is not a single model, chatbot, app launcher, or cloud assistant. It is an operating layer that binds models, memory, tools, actions, identity, and permissioned network exchange around a user's own machine. Its purpose is simple: make the computer autonomous without making the person dependent on a remote platform's memory of them.

The motivating observation comes from time-native intelligence. Current AI has become extraordinary at spatial inference: embeddings, correlations, attention, retrieval, and pattern completion. Yet personal intelligence depends on temporal structure: what happened, when it happened, why it mattered, what changed, what the system promised, what the user approved, and what should persist. A model without temporal continuity may be intelligent in a session but not yet intelligent as a companion, assistant, or operating layer.

The missing primitive is not a larger context window. It is a durable personal substrate. A context window is a temporary selection. A substrate is a continuing world model. The former lets a model answer a prompt. The latter lets a machine inherit a life.

WHY implements this thesis by turning the user's machine into a personal intelligence node. The node observes local artifacts, conversations, workflows, relationships, media, applications, and activity. It constructs a semantic graph grounded in event history. Agents retrieve from this graph, coordinate tools, and operate asynchronously. The system records what was requested, what context was used, what changed, and what was approved. When opted in, the node can also answer or route queries across a peer network by licensing computed slices rather than raw files.

1.1 Why Current AI Is Not Yet Personal

The dominant AI interface remains episodic. A user begins a session, supplies context, receives output, and then repeats the process later. Even when memory is added, it is usually implemented as a database of facts or summaries external to the model. This helps continuity, but it does not by itself create a personal operating layer. The system still lacks a unified model of time, provenance, permission, action, and consequence.

Personal intelligence requires at least four properties. First, continuity: every session should inherit the relevant past. Second, locality: the most sensitive context should remain near the owner. Third, agency: the system must be able to coordinate tools and change state under constraints. Fourth, accountability: actions must leave records that can be inspected, corrected, and learned from.

Without these properties, the assistant remains a responsive surface rather than an autonomous layer. It may answer well, but it cannot reliably manage the user's world.

1.2 The WHY Thesis

WHY is based on a single thesis:

personal intelligence requires time-native memory under local sovereignty.

The minimum object is therefore not a prompt but a personal event:

$$\boxed{e = \text{observation} + \text{time} + \text{provenance} + \text{semantics} + \text{policy}.}$$

An observation without time is trivia. Time without provenance is rumor. Provenance without semantics is a log file. Semantics without policy is a privacy hazard. Policy without memory cannot support useful action.

This paper develops WHY as the architecture that turns these events into a living semantic graph and autonomous execution layer.

1.3 Contributions

This paper makes five contributions. First, it adapts the theory of time-native intelligence to personal computing by replacing global-first consensus with local-first temporal memory. Second, it defines personal intelligence as a tuple of event history, semantic graph, agent policy, action record, and user-controlled context. Third, it specifies a four-layer architecture: local OS substrate, semantic graph, autonomous agents, and inference network. Fourth, it gives a mathematical model of memory reconstruction, context compounding, and bounded action. Fifth, it proposes an implementation and evaluation path for a product that is rigorous enough to study and practical enough to ship.

2 From Apps to Goals

Applications were built for a world in which humans performed most of the work. The user searched, compared, clicked, copied, moved, scheduled, purchased, summarized, remembered, and decided. Software provided tools, but the human supplied continuity.

Autonomous intelligence reverses the interface. When a system can reason, retrieve, and act, the unit of computing becomes the goal. A user should not need to open five applications to plan travel, compare prices, update a calendar, save a receipt, and summarize constraints. The user should state the goal; the operating layer should coordinate the required systems under the user's policies.

WHY calls this shift *life on autopilot*, but the phrase should not be misunderstood. Autopilot is not abdication. A competent autopilot remains instrumented, constrained, interruptible, and accountable. The user delegates, the system acts within scope, and the record remains inspectable.

2.1 A Canonical Example

Consider a morning brief. A conventional assistant can summarize messages if the user reconnects the relevant accounts, supplies context, and asks the right question. A WHY node has a different substrate. It knows the user's active projects, travel schedule, recent documents, family commitments, recurring purchases, open decisions, and prior preferences. It can identify what changed overnight, distinguish noise from consequence, propose actions, and ask for approval where risk is high.

The output is not merely a text summary. It is a set of actionable records: flights that changed, messages that require response, meetings that need preparation, inventory that should be restocked, and decisions waiting for the user. Each recommendation is linked to context, provenance, and confidence. Each action can be accepted, rejected, deferred, or automated under policy.

2.2 The Second Half of Human Knowledge

The public web made published information searchable. It did not make private context computable. The second half of human knowledge lives on hard drives, phones, message histories, notes, photos, spreadsheets, browsing trails, calendar records, code repositories, and local application state. It is not easily accessible to search engines because it is personal, permissioned, fragmented, and often too sensitive to centralize.

The next web is therefore not only a web of pages. It is an inference network across sovereign nodes. A node can answer a query from its local context by exposing a computed slice: an embedding, summary, proof, citation, score, or artifact. The raw file need not move. The person remains the owner of the context. The network gains useful intelligence without requiring total extraction.

3 Related Work

WHY sits at the intersection of several long traditions. Time-native intelligence argues that AI systems need ordered history, persistent identity, consequence, and shared reality. Cognitive psychology distinguishes episodic memory, semantic memory, reconstruction, and reconsolidation. Distributed systems define logical time, happens-before relations, replication, and consensus. Human-computer interaction has long imagined computers as augmentation systems rather than mere tools. Modern machine learning supplies representation learning, attention, retrieval, world models, and planning.

Lineage	Primary insight	WHY adaptation
Time-native intelligence	Intelligence needs irreversible sequence, identity, and consequence.	Make time local-first around a person before making it globally shared.
Extended mind	Tools and environments can become part of cognition.	The personal computer becomes an active cognitive substrate, not a passive device.
Episodic memory	Memory is reconstructive and temporally situated.	The node reconstructs personal context from event history and semantic graph state.
World models	Agents need predictive internal models of environment and action.	The personal graph becomes a user-specific world model for action.
Distributed systems	Independent actors need logical clocks, provenance, and agreement.	Personal nodes maintain local time and selectively synchronize computed context.

The boundary is important. WHY is not merely a vector database because vectors alone do not encode time, provenance, action, or policy. It is not merely a chatbot because chat is an interface, not a substrate. It is not merely an agent framework because tool calls without personal memory remain episodic. It is not merely a blockchain because most personal intelligence should remain local, private, and low latency. Optional anchoring or network consensus may help shared inference, but the primary cognitive substrate begins on the user's machine.

3.1 Why Not Just RAG?

Retrieval-augmented generation is a useful mechanism, but it is not a theory of personal intelligence. RAG retrieves text chunks relevant to a query. A personal intelligence layer must reconstruct context from a temporally ordered world model, reason over constraints, act under policy, and update itself from the consequence of action.

System	What it provides	What is missing
RAG	Query-conditioned retrieval over documents.	Temporal continuity, action records, user policy, and causal state.
Long context	More tokens available to a model.	Persistent graph structure and provenance across sessions.
Vector database	Similarity search over embedded items.	Identity, time, relationships, commitments, and consequences.
Agent framework	Tool calls and planning loops.	Owner-specific memory, local sovereignty, and auditable action history.
Personal data warehouse	Central storage of user data.	Local-first control, semantic reconstruction, and autonomous execution.

The distinction can be stated as a design equation:

$$\text{personal intelligence} \neq \text{model} + \text{retrieval}.$$

It is closer to

$$\text{personal intelligence} = \text{model} + \text{time} + \text{graph} + \text{policy} + \text{action}.$$

4 System Model

Let $\mathcal{P}$ denote the set of persons, $\mathcal{N}$ the set of personal nodes, $\mathcal{D}$ the set of digital artifacts, $\mathcal{E}$ the set of events, $\mathcal{G}$ the set of semantic graphs, $\mathcal{A}$ the set of agents, $\mathcal{T}$ the set of tasks, $\mathcal{C}$ the set of contexts, $\mathcal{M}$ the set of models, and $\mathcal{R}$ the set of action records:

$$\begin{aligned} \mathcal{P} &= \text{persons}, & \mathcal{N} &= \text{personal nodes}, \\ \mathcal{D} &= \text{digital artifacts}, & \mathcal{E} &= \text{events}, \\ \mathcal{G} &= \text{semantic graphs}, & \mathcal{A} &= \text{agents}, \\ \mathcal{T} &= \text{tasks}, & \mathcal{C} &= \text{contexts}, \\ \mathcal{M} &= \text{models}, & \mathcal{R} &= \text{action records}. \end{aligned}$$

In plain English, a person owns a node. The node indexes artifacts and observes events. Events update a semantic graph. Agents retrieve context from the graph, call tools, and produce action records. Models supply inference, but the node supplies continuity.

Let

$$\mathsf{H} : \{0,1\}^* \to \{0,1\}^{256}$$

be a collision-resistant hash function. Let $\mathsf{Canon}(x)$ be a deterministic canonicalization function. Let t be logical time local to a node, augmented by wall-clock time when available. Let G_t denote the personal semantic graph at time t.

Definition 4.1 (Personal Node)**.** A personal node $n \in \mathcal{N}$ is a local-first runtime controlled by a person $p \in \mathcal{P}$. It stores event history E_n, graph state G_n, policies Π_n, agent permissions, model configuration, and action records. A node may synchronize selected commitments or computed slices with other nodes, but raw personal state remains local by default.

Definition 4.2 (Personal Event)**.** A personal event $e \in \mathcal{E}$ is a typed observation:

$$e = (id, source, type, time, payloadHash, parent, entities, policy, sig).$$

The payload may be local and private. The event commits to its payload through $payloadHash$, links to prior state through $parent$, and carries policy governing retention, retrieval, and disclosure.

Definition 4.3 (Semantic Graph)**.** A semantic graph $G_t = (V_t, E_t, W_t)$ is a typed, weighted, time-indexed graph whose vertices represent people, projects, artifacts, tasks, decisions, places, media, goals, and concepts. Edges represent relations such as authored, mentioned, depends-on, scheduled, purchased, approved, contradicted, or caused.

Definition 4.4 (Personal Intelligence)**.** Personal intelligence at time t is the capacity of a node to transform a user goal g, local event history $E_{\leq t}$, graph state G_t, and policy Π into bounded action:

$$\mathsf{PI}(n, t, g) = \mathsf{Act}(\mathsf{Retrieve}(g, G_t, E_{\leq t}), \Pi, \mathcal{M}).$$

The output is not only an answer; it may be a plan, action, artifact, deferral, request for approval, or refusal.

4.1 Threat Model

WHY assumes adversaries may include malicious documents, prompt injection, compromised applications, untrusted model outputs, stale memory, confused agents, remote peers, or platform intermediaries. An adversary may attempt to exfiltrate private context, poison memory, create false associations, trigger unsafe actions, or induce a model to ignore policy. A correct implementation verifies provenance, records actions, distinguishes raw context from computed context, enforces local policy, and requires approval for high-risk effects.

5 Time-Native Personal Intelligence

Time-native intelligence begins from the claim that intelligence without history is imitation and agency without consequence is theater. WHY accepts this claim but relocates the primary temporal substrate. For a person, the first ledger should not be a public chain. It should be the local sequence of observed and approved life events.

5.1 Spatial and Temporal Systems

Modern AI systems excel at spatial intelligence:
- embeddings and semantic similarity,
- pattern recognition over large corpora,
- attention over tokens and modalities,
- statistical completion and synthesis.

Personal intelligence also requires temporal intelligence:
- ordered experience,
- persistent identity,
- causal and counterfactual reasoning,
- memory reconstruction,
- consequence-aware action.

The architecture is dual. The model proposes; the node remembers. The model predicts; the graph grounds. The agent acts; the record persists. Intelligence emerges at the interface between high-dimensional inference and temporally ordered personal state.

5.2 Event-Grounded Memory

Let $E_n(t)$ be the ordered event history of node n up to time t. Let $G_n(t)$ be the semantic graph produced by applying a graph update function U_G to event history:

$$G_n(t) = \mathsf{Reduce}(E_n(t), U_G, G_0).$$

Memory is not a table lookup. It is a reconstruction:

$$\mathsf{Memory}(n, q, t) = \sum_{e_i \in E_n(t)} \alpha_i(q, t) \cdot \mathsf{Embed}(e_i) + \sum_{v_j \in G_n(t)} \beta_j(q, t) \cdot \mathsf{Embed}(v_j).$$

Here α_i and β_j are attention weights conditioned on query, recency, salience, source reliability, user feedback, and policy. This formulation matters because personal memory is both episodic and semantic. An event says what happened. A graph says what it came to mean.

Requirement 5.1 (Temporal Provenance)**.** Every durable memory SHOULD be traceable to one or more events. A node MAY store summaries, embeddings, and derived relations, but those derived objects SHOULD carry provenance links to the events from which they were inferred.

5.3 Continuity Without Total Recall

Human memory is not an exhaustive archive. It is selective reconstruction under attention, salience, and forgetting. WHY should not log every token, pixel, or transient thought. It should preserve the events needed to reconstruct useful context while compressing, summarizing, and retiring low-value data under explicit policy.

For event e_i, define a retention score

$$r_i(t) = \lambda_s s_i + \lambda_u u_i + \lambda_f f_i + \lambda_c c_i - \lambda_p p_i - \lambda_a a_i(t),$$

where s_i is semantic salience, u_i is user-confirmed importance, f_i is future utility estimate, c_i is causal centrality, p_i is privacy risk, and $a_i(t)$ is age. Retention is therefore not a storage accident. It is a policy-weighted cognitive decision.

6 Architecture

WHY is organized into four layers.

6.1 Layer 1: Local OS Substrate

The local OS substrate indexes the user's digital world on-device. Files, conversations, notes, photos, browsing, calendar events, workflows, applications, and activity remain close to the person. The index should support embeddings, metadata, provenance, permissions, and revocation.

6.2 Layer 2: Semantic Graph

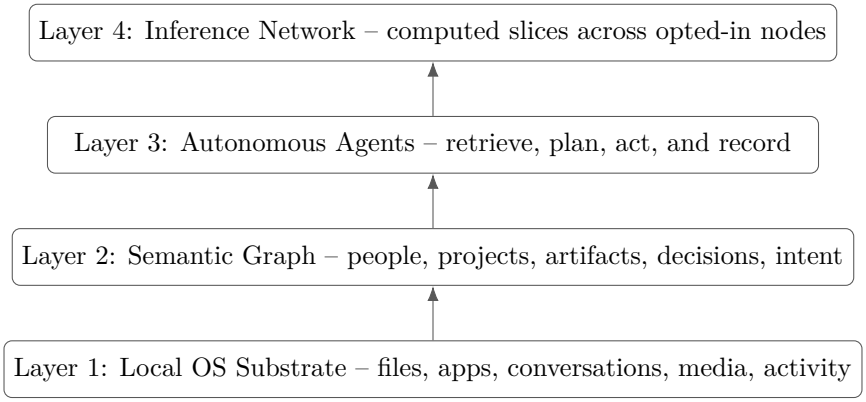
The semantic graph connects people, projects, decisions, assets, deadlines, goals, and constraints. It is the living structure through which context compounds. A document is not merely a file. It may be part of a project, authored by a collaborator, attached to a decision, contradicted by a later note, and relevant to a future meeting.

6.3 Layer 3: Autonomous Agents

Agents retrieve context, coordinate tools, evaluate outcomes, and operate asynchronously. They are not unconstrained scripts. Each action is governed by policy, risk class, tool permission, and user approval. Work moves across time instead of inside a single prompt.

6.4 Layer 4: Inference Network

When opted in, a node may query a swarm of peers. Synthesized answers from public sources and permissioned private sources can replace link lists. Nodes negotiate data access and expose computed context rather than raw files. The network becomes an inference layer over locally owned knowledge.



7 The Personal Semantic Graph

The personal semantic graph is the core data structure of WHY. It is not a generic knowledge graph. It is an owner-specific, time-indexed, policy-bearing graph grounded in local events.

7.1 Graph Update

Let e_t be a new event. The graph update is

$$G_{t+1} = U_G(G_t, e_t, \phi(e_t), \Pi),$$

where $\phi(e_t)$ extracts entities, relations, embeddings, source confidence, and candidate actions, and Π is policy. The update may add vertices, update edge weights, create temporal links, mark contradictions, or request human clarification.

Edge weights should incorporate both semantic and temporal evidence:

$$w_{ij}(t) = \sigma(\theta_1 sim(v_i, v_j) + \theta_2 cooccur(v_i, v_j) + \theta_3 recency_{ij} + \theta_4 feedback_{ij} + \theta_5 causal_{ij}).$$

This is a deliberately general form. The key requirement is not a specific weight function but that the graph preserve the difference between semantic similarity, temporal proximity, causal dependence, and user-confirmed importance.

7.2 Context Compounding

Useful context compounds. Each event can increase the utility of prior events by creating new relations. A flight confirmation becomes more useful after a calendar invite. A note becomes more useful after a meeting. A document becomes more useful after a decision references it.

Proposition 7.1 (Context Compounding)**.** *Let $U(G_t, q)$ be the expected utility of graph state G_t for a distribution of queries $q \sim Q$. If graph updates preserve provenance and add at least one policy-permitted relation that improves retrieval for a nonzero-measure subset of Q, then expected retrieval utility is non-decreasing under the update, excluding storage and privacy costs.*

Proof. For any query unaffected by the new relation, the previous retrieval path remains available. For a query improved by the new relation, the retrieval set gains a path with higher expected relevance. Since the subset of improved queries has nonzero measure, expected utility weakly increases. Privacy and storage costs are excluded by assumption and must be accounted for separately in retention policy.

8 Personal World Models

A personal semantic graph is not sufficient by itself. To act well, the system must predict how the user's world changes under possible actions. This is the personal version of a world model. It is not a simulation of the entire physical world. It is a task-relevant model of the user's digital, social, and operational environment.

Let z_t be a latent state summarizing the node's graph, recent events, user preferences, and active goals. Let a_t be a candidate action. A personal world model estimates

$$\hat{z}_{t+1} = F_\theta(z_t, a_t),$$

and predicts observations, costs, user approvals, and downstream consequences:

$$(\hat{o}_{t+1}, \hat{c}_{t+1}, \hat{u}_{t+1}, \hat{r}_{t+1}) = H_\theta(\hat{z}_{t+1}).$$

The point is not omniscience. The point is counterfactual discipline. Before acting, the node should ask: if I do this, what state will I create, what risk will I impose, what approval is needed, and what evidence will remain?

8.1 Training Signal

The training signal for personal intelligence is not only next-token likelihood. It includes user acceptance, correction, reversal, latency, provenance quality, privacy cost, and long-horizon goal satisfaction. A generic objective is

$$\mathcal{L}_{WHY} = \lambda_p \mathcal{L}_{pred} + \lambda_g \mathcal{L}_{graph} + \lambda_a \mathcal{L}_{approval} + \lambda_r \mathcal{L}_{risk} + \lambda_s \mathcal{L}_{sovereignty}.$$

Here $\mathcal{L}_{pred}$ penalizes failed prediction, $\mathcal{L}_{graph}$ penalizes incorrect or unsupported graph updates, $\mathcal{L}_{approval}$ penalizes actions the user rejects, $\mathcal{L}_{risk}$ penalizes unsafe behavior, and $\mathcal{L}_{sovereignty}$ penalizes unnecessary exposure of private context.

This objective makes the product scientifically interesting. A WHY node can become better because it observes not just language but the consequences of assistance in a real personal environment.

8.2 Counterfactual Planning

Let A_g be the set of possible action plans for goal g. The node should choose

$$a^* = \arg\max_{a \in A_g} \mathbf{E}[U(z_{t+k} \mid z_t, a)] - \mathsf{Cost}(a) - \mathsf{Risk}(a) - \mathsf{Leakage}(a).$$

This expression turns autonomy into constrained optimization. It is not enough for a plan to be plausible. It must be useful, low-friction, safe, and privacy-preserving.

9 Autonomy and Action

The next interface is not a chat window. It is an orchestration layer where agents can retrieve memory, understand constraints, coordinate tools, evaluate outcomes, and leave auditable records. This requires a separation between proposing and acting.

9.1 Goal as the Unit of Computing

Let a user goal be

$$g = (intent, constraints, deadline, risk, approvalPolicy).$$

The node transforms g into a plan

$$\pi = (steps, tools, requiredContext, approvals, rollback).$$

Each action x is executed only if

$$\mathsf{Allowed}(x, g, \Pi, t) = 1.$$

The purpose of the operating layer is not to remove the user from the loop. It is to move the user to the correct point in the loop: high-level intent, policy, approval, and exception handling.

9.2 Action Records

Every meaningful action SHOULD leave an action record:

$$r = (goal, contextRefs, toolCalls, outputs, approvals, effects, time, hash).$$

Action records make autonomy auditable. They also teach the system. A booking that the user accepts reinforces preferences. A suggested action the user rejects becomes negative evidence. A mistake that requires correction becomes part of future caution.

Requirement 9.1 (Reversible First)**.** For local actions over user-owned state, a WHY agent SHOULD prefer reversible staging before irreversible mutation. A staged file movement is preferred to immediate deletion. A draft response is preferred to sending an email. A proposed purchase is preferred to an automatic purchase unless policy already permits the transaction.

9.3 Bounded Authority

Personal intelligence is dangerous if authority is ambient. An agent that can read everything, spend freely, message anyone, and remember without policy is not an assistant; it is an unbounded system process with a personality. WHY agents must operate under explicit scopes, tool permissions, risk classes, and approval gates.

10 Local Sovereignty and Inference Exchange

The most valuable dataset is the one that describes a person's digital world. For that reason, ownership must be architectural rather than rhetorical. A local-first system keeps raw personal context near the user, exposes computed slices only under policy, and supports revocation.

10.1 Computed Slices

A computed slice is a bounded answer derived from private context:

$$s = (query, answer, citations, confidence, policy, proof, expiry).$$

The raw data need not leave the node. The slice may include a summary, embedding, score, retrieval result, or artifact. It may also include cryptographic commitments to source events so that another party can verify consistency without seeing the underlying files.

10.2 The Inference Network

When nodes exchange computed slices, the web changes. Search asks for pages. Inference asks for answers grounded in permissioned context. A query can be answered by public sources, a user's own graph, and opted-in peer nodes that possess relevant private knowledge. This is not a social feed. It is a market and protocol surface for high-fidelity inference.

The network SHOULD obey three constraints. First, raw personal files are not the default unit of exchange. Second, every exchange has a policy and expiry. Third, value should accrue to the owner of the context that improved the answer.

11 Security and Alignment

Personal intelligence has a different safety profile from general chat. It is closer to an operating system process with memory and agency. The system must therefore align not only outputs but also context access, tool use, retention, disclosure, and action.

11.1 Memory Integrity

The semantic graph is vulnerable to memory poisoning. A malicious email may assert a false deadline. A document may include prompt injection. A website may instruct the agent to leak secrets. A compromised tool may return fabricated metadata. WHY must assign source reliability and policy to events before updating durable memory.

Requirement 11.1 (Quarantined Inference)**.** Untrusted artifacts SHOULD enter the graph as quarantined evidence until corroborated by trusted sources, user approval, or low-risk use. A node MUST distinguish "observed in an untrusted source" from "believed as durable user context."

11.2 Authority Separation

Retrieval authority and action authority must be separate. A model may be allowed to read a document for summarization without being allowed to send emails, purchase goods, or mutate files. If action authority is inferred merely from retrieval authority, prompt injection becomes escalation.

Theorem 11.1 (No Ambient Authority)**.** *If every external action requires an explicit policy grant and policy grants are not derivable from retrieved content, then untrusted content alone cannot authorize a new external action in a conforming* WHY *node.*

Proof. An external action executes only if a policy grant exists. Retrieved content may influence the model's proposal but cannot create a policy grant by assumption. Therefore untrusted content alone cannot satisfy the authorization condition for execution.

11.3 Auditability

An autonomous system becomes trustworthy when the user can answer four questions after the fact: what did it know, why did it act, what did it change, and who approved it? WHY action records provide the minimum substrate for this audit. They do not guarantee correctness, but they preserve the evidence required for correction.

12 Mathematical Properties

12.1 State Derivation

Theorem 12.1 (Provenance-Bounded State)**.** *If every durable graph update u_t is derived from an event set E_t and update function U_G, then every durable graph fact $f \in G_t$ has at least one provenance path to an originating event or human assertion.*

Proof. The initial graph G_0 is empty or composed of explicit human assertions. Each update adds or modifies facts only through $U_G(G_t, e_t, \phi(e_t), \Pi)$. By induction over event time, any new fact either derives from e_t, from prior facts that already have provenance paths, or from a human assertion. Thus each durable fact has a provenance path.

12.2 Memory Lower Bound

Let M_t be the memory representation used by the node at time t, and let E_t be the event history. If the system must reconstruct all facts that are not semantically redundant under equivalence relation $\sim$, then

$$H(M_t) \geq H(E_t/\sim) - \epsilon.$$

The result is a version of the TNI memory lower bound adapted to personal intelligence. Useful compression is possible, but lossless personal continuity cannot be obtained from a representation that has discarded the relevant distinctions.

12.3 Causal Consistency

Theorem 12.2 (Action Causality)**.** *If every action record references the goal, context, approval, and prior event root from which it was produced, then a later audit can determine whether the action was causally grounded in the recorded state available at execution time.*

Proof. The action record commits to the prior event root and context references. If the graph and event log are tamper-evident, the auditor can reconstruct the state available at that root. The auditor can then evaluate whether the goal, policy, and context support the action. The theorem does not prove the action was correct; it proves the action can be judged against its recorded causal state.

12.4 Sovereignty Constraint

Let $L(x)$ be the leakage of private information caused by output x, and $V(x)$ be the value of the output. A rational node should release a computed slice x only when

$$V(x) - C(x) - \lambda L(x) \geq 0,$$

where $C(x)$ is computation, coordination, or opportunity cost and λ is the user's privacy-risk parameter. This does not define a universal price for privacy. It defines the design constraint: the system must account for leakage as a first-class cost.

13 Implementation Pathway

13.1 Reference Product

The first WHY product is a local application that indexes the user's digital world, builds a semantic graph, and runs agents over that graph. It should begin with bounded, observable tasks: morning brief, travel planning, reservations, restocking, personal research, folder organization, document preparation, and inbox triage.

The minimum useful node is

$$n = (E, G, I, A, \Pi, R),$$

where E is event history, G is graph state, I is local index, A is agent runtime, Π is policy, and R is action record store.

Listing 1: Minimal WHY node manifest.

```
{
  "why": "0.1",
  "nodeId": "why_node_local_001",
  "owner": "local-user",
  "layers": ["local-index", "semantic-graph", "agents"],
  "models": ["local", "remote-with-policy"],
  "stores": ["sqlite", "sqlite-vec", "encrypted-files"],
  "capabilities": ["brief", "research", "organize", "reserve", "restock"],
  "policy": {
    "rawDataLeavesDevice": false,
    "requiresApproval": ["purchase", "send-message", "delete-file"],
    "computedSlices": "opt-in"
  }
}
```

13.2 Build Sequence

The implementation should proceed in five stages. First, build a local index over files, metadata, notes, calendar, browser history, and selected conversations. Second, build event capture and semantic graph updates. Third, add agents that can retrieve, plan, and stage actions. Fourth, add action records, approvals, and rollback. Fifth, add opt-in inference exchange through computed slices.

13.3 Why Not Start With the Network

The network is more powerful after the node is useful alone. A personal intelligence system must first earn trust by helping the owner without exposing them. Only then should it become a peer in a larger inference network. Local usefulness is the root of network value.

14 Evaluation

WHY should be evaluated as an operating layer, not only as a question-answering system. We propose seven evaluation families.

Metric	Question
Continuity	Does the system carry relevant context across days, projects, and sessions?
Goal completion	Does it reduce steps needed to complete real user tasks?
Provenance	Can outputs be traced to events, artifacts, and decisions?
Action safety	Are irreversible or high-risk actions gated, staged, and logged?
Graph utility	Does the semantic graph improve retrieval and planning over time?
Privacy leakage	How much raw or sensitive context leaves the local node?
User trust	Does the user delegate more over time without losing control?

The critical metric is not raw automation rate. It is trusted autonomy: the amount of useful work completed per unit of user attention while preserving reversibility, provenance, and sovereignty.

15 Risks and Open Problems

First, memory can become wrong. A semantic graph may infer false relations, overweight stale context, or preserve a preference the user has outgrown. The system must support correction, contradiction, decay, and forgetting.

Second, action creates risk. A system that books, buys, deletes, messages, or moves files must treat approval policy as core infrastructure. Unsafe autonomy will destroy trust faster than weak intelligence.

Third, privacy and usefulness are in tension. The most useful context is often the most personal. Local-first architecture reduces exposure but does not eliminate leakage through summaries, embeddings, screenshots, or tool calls.

Fourth, personalization can narrow perception. A system that knows a user too well may reinforce defaults rather than expanding options. It must preserve space for surprise, exploration, and counterfactuals.

Fifth, networked inference creates incentive problems. If nodes are compensated for useful computed context, they may overstate confidence, leak more than intended, or collude. Reputation, audits, and policy-bound receipts are required.

16 Conclusion

The next model product is not only a better model. It is a better substrate for models to live inside. Personal intelligence requires a machine that remembers what matters, grounds inference

in the user's world, acts under constraint, and improves through time. Without such a substrate, even powerful models remain visitors in a person's life: brilliant for a moment, gone by morning.

WHY proposes the missing operating layer. It makes memory event-grounded rather than conversationally accidental. It makes context a graph rather than a pile of files. It makes autonomy policy-bound rather than ambient. It makes networked intelligence opt-in and computed rather than extractive. It makes the user's machine a node in an inference network without requiring the user's raw life to become someone else's dataset.

Search organized the public web around links. Feeds organized attention around platforms. WHY organizes personal intelligence around time, context, and action. The computer becomes autonomous because it finally has a memory of the world it is acting within.

A Object Summary

Object	Required fields	Purpose
Node	nodeId, owner, stores, models, policies, capabilities	Local runtime controlled by the person.
Event	id, source, type, time, payloadHash, parent, entities, policy	Atomic temporal observation.
Graph fact	subject, relation, object, weight, provenance, time	Derived semantic state with source links.
Goal	intent, constraints, deadline, risk, approvalPolicy	Unit of computing for the autonomous layer.
Action record	goal, contextRefs, toolCalls, outputs, approvals, effects, hash	Auditable record of what the system did.
Computed slice	query, answer, citations, confidence, policy, proof, expiry	Permissioned answer derived from local context.

B Example Action Record

Listing 2: Morning brief action record.

```
{
  "recordType": "WHYActionRecord",
  "why": "0.1",
  "goal": "Produce morning brief and prepare approved actions",
  "contextRefs": [
    "calendar:2026-06-04",
    "project:tokyo-trip",
    "inbox:overnight-priority",
    "shopping:household-restock"
  ],
  "toolCalls": [
    {"tool": "calendar.read", "risk": "low"},
    {"tool": "inbox.summarize", "risk": "medium"},
    {"tool": "reservation.monitor", "risk": "medium"}
```

```
  ],
  "outputs": ["brief.md", "draft-actions.json"],
  "approvals": [
    {"action": "book-reservation", "status": "requires-human"}
  ],
  "effects": "no external state changed",
  "eventRoot": "sha256:0f4c...",
  "recordHash": "sha256:8a91..."
}
```

C References

1. Ashish Vaswani et al. Attention Is All You Need. Advances in Neural Information Processing Systems, 2017.
2. Yann LeCun. A Path Towards Autonomous Machine Intelligence. Open review manuscript, 2022.
3. Leslie Lamport. Time, Clocks, and the Ordering of Events in a Distributed System. Communications of the ACM, 1978.
4. Endel Tulving. Episodic and Semantic Memory. Organization of Memory, 1972.
5. John Locke. An Essay Concerning Human Understanding. 1689.
6. Karim Nader, Glenn E. Schafe, and Joseph E. LeDoux. Fear memories require protein synthesis in the amygdala for reconsolidation after retrieval. Nature, 2000.
7. Daniel L. Schacter. Searching for Memory: The Brain, the Mind, and the Past. Basic Books, 1996.
8. Andy Clark and David Chalmers. The Extended Mind. Analysis, 1998.
9. Judea Pearl. Causality: Models, Reasoning, and Inference. Cambridge University Press, 2000.
10. Stuart Russell and Peter Norvig. Artificial Intelligence: A Modern Approach, Fourth Edition. Pearson, 2020.
11. Vannevar Bush. As We May Think. The Atlantic, 1945.
12. Douglas Engelbart. Augmenting Human Intellect: A Conceptual Framework. SRI, 1962.
13. Mark Weiser. The Computer for the 21st Century. Scientific American, 1991.
14. Tim Berners-Lee, Roy Fielding, and Larry Masinter. Uniform Resource Identifier: Generic Syntax. RFC 3986, 2005.
15. Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The PageRank Citation Ranking: Bringing Order to the Web. Stanford InfoLab Technical Report, 1999.
16. Satoshi Nakamoto. Bitcoin: A Peer-to-Peer Electronic Cash System. 2008.
17. Ockams Inc. WHY About Page. `https://why.com/about`. Accessed June 4, 2026.