

Wayhand

A Human-Authority Architecture for Persistent, Replaceable AI-Enabled Computing

Publication candidate 0.3.12 | 22 August 2026

Creator: TEKWITCH | ORCID 0009-0001-9571-8132

Design basis: accepted WAYHAND-MINIMUM-COMPLEXITY-INTEGRATED-SOURCE v0.15.05

Design-source SHA-256: 77808f787ab7750b56f2c696baeb9bcd7b0f7bee47bbb5eb0cdfa60fc9000e58

Constitutional basis: Wayhand constitutional v0.2, SHA-256

5a038e209b9fc37e8f15c8646d32884a8c840f318b374f1e09fa3deaec7beb1b

Status: non-normative architecture and development-status snapshot. The source-to-control/test mapping gate (R-00) and development-host measurement/preservation gate (R-01) have passed their defined scopes; the first runtime trust-boundary implementation/proof gate (R-02) has not passed. No Wayhand has been germinated, commissioned, recovered, or activated. The project does not claim technical closure for Phase 1 or the terminal Seed, external-fitness closure, or overall Wayhand fitness.

Intended publication license: CC BY 4.0

DOI: pending final Zenodo deposit

Abstract

Wayhand proposes an architecture for a local AI-enabled system that is intended to remain the same continuing system even when its models, software components, service providers, or host hardware are replaced. It is governed by an actual human **Authority**. In this paper, Authority means the human whose attributable decisions are required where the design calls for human authorization, and **standing** means recognized validity to establish a particular governing fact. A model can therefore be highly competent without having standing to create the system's purpose, grant **Permission**, accept governing source, or count as human approval. Before Wayhand permits a **consequential action** - an operation that changes protected local state or causes an effect outside the system - the facts that justify that action must be established separately and remain independently checkable.

The design concentrates compromise protection on two especially sensitive assets: protected Authority knowledge stored for later use, and the capability to cause external effects. It separates institutional state, human assurance, protected memory, action validation, denial, and effect capability into independently enforceable roles. Three processes can positively advance state needed for consequential action: wayhand-core, wayhand-authorityd, and wayhand-actiond. wayhand-memoryd protects stored Authority knowledge but cannot authorize action; Hold Guard can block action but cannot authorize it; and a separate Credential/Effect-Capability boundary issues only the bounded capability needed for an already validated effect while keeping reusable credentials outside wayhand-actiond.

Wayhand also treats models and engineering systems as powerful but non-authoritative; preserves offline reasoning, initialization, recovery, and reconstruction when networks or providers are unavailable; tests whether replaceable components remain replaceable after real use; and scopes relationships between Wayhands so privilege in one relationship does not silently create privilege in another. The originating project is designed to terminate in a **Public Seed**: a generic, independently reconstructable release artifact from which a person can **germinate** a new Wayhand with a new identity and human Authority without depending on a continuing upstream service. The proposed contribution is primarily architectural synthesis and proof discipline. Established ideas from protection architecture, capability security, high-assurance isolation, software-supply-chain provenance, reproducible builds, usable security, local-first software, AI risk governance, and agent interoperability provide relevant precedent, candidate mechanisms, and failure lessons. The Wayhand source requires any external mechanism that becomes a production binding to be selected and qualified separately under Wayhand's own requirements; citation or similarity does not itself adopt a dependency or transfer upstream assurance. The accepted source is an implementation-and-proof specification; accepting that source does not itself prove that an implementation exists or that the design is fit for use.

Reader terminology

- **Wayhand:** a persistent local AI-enabled system whose identity and governing state are intended to survive replacement of models, software components, providers, and host hardware.
- **Authority:** the actual human whose purposes the system serves and whose attributable decisions are required wherever the design says human authorization is necessary. A model, developer, vendor, peer, credential holder, or restored machine state does not become Authority merely by being capable or available.
- **Standing:** recognized validity to establish a particular governing fact. Different facts have different owners: the ability to compute, write data, or possess credentials does not by itself create valid source acceptance, human approval, purpose, or Permission.
- **Purpose:** the currently valid human-originating objective or governing intent for which action is being considered. Purpose answers *what the system is presently serving*, not merely what a model proposes to do.
- **Permission:** a bounded, attributable authorization for a consequential action or class of actions. Permission is distinct from purpose and from technical capability; possessing a password, token, credential, or tool does not itself constitute Permission.

- **Human assurance:** evidence that the required human Authority was authenticated at the assurance level claimed and that an exact human decision or delegation is attributable to that Authority. Human assurance is evidence about the human decision path; it does not itself create Purpose or Permission.
- **Consequential action / effect:** an operation that changes protected local state or the outside world rather than merely producing advisory reasoning. Examples include installing software, changing credentials, sending protected data, invoking an external service, or causing another durable external effect.
- **Positive authority:** the ability to create or advance valid state that can help enable consequential action. A deny-only component can block action without having positive authority because blocking cannot create authorization.
- **Source Anchor / source standing:** the independently verified identity of the exact accepted specification that the running system is supposed to obey. Source acceptance is one governing fact; it does not itself create purpose, Permission, or human approval.
- **Public Seed:** the generic terminal release artifact from which a new Wayhand can be created without inheriting a prior instance's identity, Authority, credentials, or private state. The design requires the Seed to preserve an independently verifiable boot and reconstruction path for supported hardware.
- **Phase 1:** the source's initial implementation-and-proof profile for establishing the core runtime trust boundaries on the first supported development/deployment substrate. It is a roadmap/proof scope, not a different or reduced Wayhand product.
- **Germination:** the controlled process that turns a Public Seed into a new Wayhand. Germination establishes a new system identity and human Authority, prepares the target host, and validates the state required before normal operation; it is distinct from restoring an existing Wayhand or reproducing one from evidence.
- **Commissioning:** the transition from the restricted germination environment into ordinary operation under Wayhand's normal controls. Commissioning establishes the declared operating profile; it does not itself grant Permission for a particular consequential action.
- **Recovery / reproduction:** recovery restores continuity of an existing Wayhand from authorized continuity material and then re-establishes current governing state before consequential work can resume. Reproduction creates a new Wayhand with a new identity rather than continuing the old one.
- **Hold Guard / SAFE HOLD:** an independently enforceable deny-only control for consequential activity. Entering hold blocks affected action; leaving hold does not itself authorize anything or automatically resume prior work.
- **Protected Authority knowledge at rest:** durable, non-public human content and sensitive retrieval/provenance state stored for later use. wayhand-memoryd protects that stored material; plaintext already legitimately released into an active consumer context is outside the at-rest confidentiality claim for that particular compromise event.
- **Technical closure:** a claim that a defined set of machine-checkable requirements has been reproducibly satisfied for exact source, build, and environment identities. It does not stand in for human usability or other properties that require external observation.
- **External fitness:** evidence from humans or genuinely independent observation for properties that internal technical tests cannot honestly establish, such as comprehension, burden, usable control, or whether an unfamiliar person can perform a required procedure without project-specific coaching.
- **Constitutional:** fixed source-defined rules about what the system may treat as valid, who may establish particular governing facts, and which powers must remain separated. The term does not assert personhood, sovereignty, or moral agency.

1. Problem and Design Position

AI-enabled computing systems increasingly combine model reasoning with tool use, coordination, software modification, and actions through external services. Research such as ReAct illustrates an influential pattern of interleaving model reasoning with actions in external environments, while protocols such as MCP and A2A standardize access to tools and communication between heterogeneous agents.[3][12][13] These developments increase what an AI-enabled system can do. They do not, by themselves, determine which facts the system may treat as valid authorization, which human purposes remain current after software or provider replacement, what recovered state may safely resume, how stored private knowledge is protected from a compromised reasoning component, or how usable human control is demonstrated rather than merely asserted.

Wayhand addresses those questions by separating **capability** from **standing**. A model may be able to plan an operation without having standing to establish the purpose for that operation. A credential may make an operation technically possible without constituting Permission to use it. A recovered record may prove that an action was once authorized without proving that the authorization is still current. Wayhand therefore requires the governing facts for consequential action - current purpose, attributable human authorization where required, Permission, accepted source, hold state, and bounded effect capability - to remain distinct and independently checkable.[1][21]

This separation is intended to preserve human control across replacement and recovery. The actual human Authority is the source of the human decisions that the architecture requires; models, developers, vendors, peer systems, restored state, and possession of credentials do not silently acquire equivalent standing. Consequential operations are constrained not only by what software can technically execute, but also by whether the exact request is justified by current governing state and can be carried out through the system's bounded consequence path.[1][21]

The design is therefore different from a model constitution in the Constitutional AI sense. Constitutional AI uses written principles to shape or train model behavior through critique and AI feedback.[4] Wayhand also has source-defined rules, but it does not rely on a model's obedience to those rules as the decisive security boundary. The required facts are represented outside ordinary model cognition in typed state, independently verifiable human decisions, separately checked source identity, deny-only controls, and bounded effect capabilities. A model's confidence, agreement, or self-description cannot substitute for a missing governing fact.[1]

2. Design Principles

Human Authority supplies the human decisions the system requires; it is not merely advisory. Where the design requires a human decision, the decision must be attributable to the actual human Authority; advice or agreement from a model, developer, vendor, peer, credential holder, or restored machine state cannot silently replace it.

Purpose, human assurance, source standing, Permission, and action are different facts. Accepting source does not authorize an action. Permission does not define the purpose it serves. Successful execution does not renew Permission or accept new source. Each required fact is established through its own path and must still be current when the exact action is attempted.[1]

The two highest-consequence protected assets are stored Authority knowledge and external execution capability. In security terms these are the design's two *crown jewels*: compromise of either defeats a central protection claim. A compromised model, UI, engineering workshop, or ordinary institutional-state component may cause serious damage, but that compromise should not by itself grant retrospective access to the protected memory archive or unrestricted ability to cause effects outside the protected boundary.[1]

Local capability is a continuity property. The minimum reasoning, germination, recovery, reconstruction, and governing-state paths must remain usable without assuming that a network service or a particular provider is available. Better networked services may be added, but they cannot be the only route to continuity.[1]

Models and tools are replaceable components. Reasoners, runtimes, providers, interfaces, databases, languages, and hosts may change without creating a different Wayhand when the identity-bearing governing state and human Authority remain valid. Replaceability is therefore a property of the surrounding architecture, not a promise that any particular model or implementation will remain permanent.[1]

Self-engineering is orchestrated, not self-authorizing. Models, coding systems, humans, peer systems, or future tools may propose or build candidate software. Candidate output does not gain standing merely because it was generated successfully; controls the candidate does not own must test and explicitly admit it, introduce it while consequential action is blocked, and preserve a known rollback path.[1]

Privilege does not spread automatically across federation relationships. A Wayhand may cooperate with other Wayhands for reasoning, compute, or bounded actions, but authority or privilege in one relationship does not automatically create identity, source acceptance, credentials, installation rights, recovery rights, or human Authority in another relationship.[1][21]

Replaceability is tested after real use. A component that can be replaced only on a clean installation is not sufficiently replaceable if legitimate accumulated state later creates avoidable lock-in. The design therefore treats exit burden, residue, and migration difficulty as evidence that must remain visible rather than being hidden by a nominal replacement interface.[1]

Proof cannot be stronger than the observation that supports it. A proposed test, historical result, waiver, unknown result, or synthetic fixture cannot be summarized as current evidence that a live property has been established. Machine-checkable claims require reproducible technical evidence; claims about human comprehension, burden, or unfamiliar-user performance require genuinely external human evidence when those are the properties being claimed.[1]

2.1 Why the separations follow from failure cases

Compromised cognition or engineering workshop. A capable reasoner or coding environment can generate convincing plans, requests, explanations, tests, and malicious candidate software. Wayhand therefore treats cognition as advisory and workshop output as untrusted candidate bytes. Neither is given a direct, unrestricted route to credentials or external effects, and neither gains governing standing merely because its output is sophisticated or persuasive.[1]

Protected-memory compromise. Durable personal history is a different risk from plaintext that has already been intentionally supplied to an active reasoning context. Wayhand places stored Authority knowledge and its retrieval state behind wayhand-memoryd. Ordinary recall is limited by consumer, session, time, and volume; deeper or bulk access requires stronger current human assurance. Releasing material for local cognition is also a separate decision from authorizing that material to leave the protected system through an external action.[1]

Stale recovery. A backup or restored machine can contain queues, credentials, relationships, permissions, purpose records, and process state that were legitimate when captured but should no longer be trusted as current. Restoration therefore does not mean automatic resumption. Recovery reconciles the restored material against current human and source state, invalidates stale execution state, updates the currentness markers used to reject replay, and keeps consequential action blocked until a fresh authorization path has been completed.[1]

Federated compromise. Cooperation between capable systems creates useful paths for reasoning, shared compute, and bounded action, but those same paths can become unintended privilege bridges. Wayhand therefore binds each relationship to explicit local state, gives it separate scope and currentness checks, and makes privileges non-transitive: compromise of one peer or relationship does not automatically authorize another.[1][21]

These failure cases are design constraints, not claims that each underlying mechanism is novel. They explain why Wayhand does not reduce governance to a single general agent carrying a policy prompt, a credential store, and unrestricted tools.

3. Architecture at a Glance

The architecture can be read as a sequence of independently checked roles. Three components can **positively advance** state needed for a consequential action: one maintains institutional records and policy state, one verifies the human Authority and exact human decisions, and one independently validates and dispatches the exact action. Wayhand calls these the three **positive-authority chokepoints**. Other mandatory boundaries may store protected information, block action, or issue narrowly scoped effect capability, but they cannot create authorization by themselves.[1]

For an ordinary consequential action to proceed, the required governing facts must agree for the same exact request. wayhand-core supplies the relevant institutional records, including current purpose and Permission records. wayhand-authorityd supplies independently verifiable human-assurance and decision evidence. wayhand-actiond checks those inputs together with current source, hold state, anti-replay/freshness state, and capability state; it is the sole normal boundary that dispatches consequential action and reconciles the outcome afterward.[1]

Protected memory and execution capability remain separate from those three positive-authority roles. wayhand-memoryd protects stored Authority knowledge and limits recall, but cannot create purpose, Permission, human approval, source acceptance, hold release, credentials, or external effects. The independent Hold Guard can deny or contain consequential activity but cannot authorize it. The Credential/Effect-Capability boundary keeps reusable secrets outside wayhand-actiond and exposes only the exact bounded handle required for an already validated operation. The exact operation is then performed by a qualified, guard-contained runner rather than by giving ordinary cognition or wayhand-actiond ambient external authority.[1]

In the current initial implementation-and-proof profile (Phase 1), the design therefore has three mandatory positive-authority processes - wayhand-core, wayhand-authorityd, and wayhand-actiond - plus mandatory wayhand-memoryd, the independent Hold Guard, and the Credential/Effect-Capability boundary. The capability boundary may be implemented by sufficiently strong host/provider primitives; when those primitives cannot prove the required separation, the source requires a separate wayhand-capabilityd process.[1]

wayhand-core owns deterministic institutional state and policy machinery, including typed purpose and Permission records, but being able to write core state does not make core the human Purpose Authority. wayhand-authorityd owns Authority assurance and exact human-decision/delegation ceremonies. wayhand-actiond owns independent consequential-action validation, sole normal dispatch, and outcome reconciliation. wayhand-memoryd owns protected Authority knowledge at rest and the accounting that prevents routine recall from becoming retrospective enumeration of the archive.[1]

One v0.15.05 detail is important to that separation. core owns the branch-local purpose freshness binding (purpose_branch_id plus purpose_epoch), but a new positive epoch is not valid merely because core writes a larger number. Every positive transition must be justified by independently verifiable purpose-standing evidence, and wayhand-actiond checks that branch/epoch separately from the human purpose-governance decision cursor. Stale, decreased, reused, or fabricated purpose epochs are ineligible for consequential dispatch.[1]

Accepted source is not whatever source a resident component happens to claim. The exact accepted specification is represented by a **Source Anchor** whose identity is independently pinned and verified by the trust-critical processes. This is a prerequisite that those processes check rather than a permission emitted by a single source daemon. Likewise, Permission remains distinct even though no separate permissiond process exists: wayhand-actiond must independently verify the exact Permission record, the associated human-decision or delegation basis, current Purpose, source state, current hold state, anti-replay/freshness state, capability state, and the exact request before dispatch.[1]

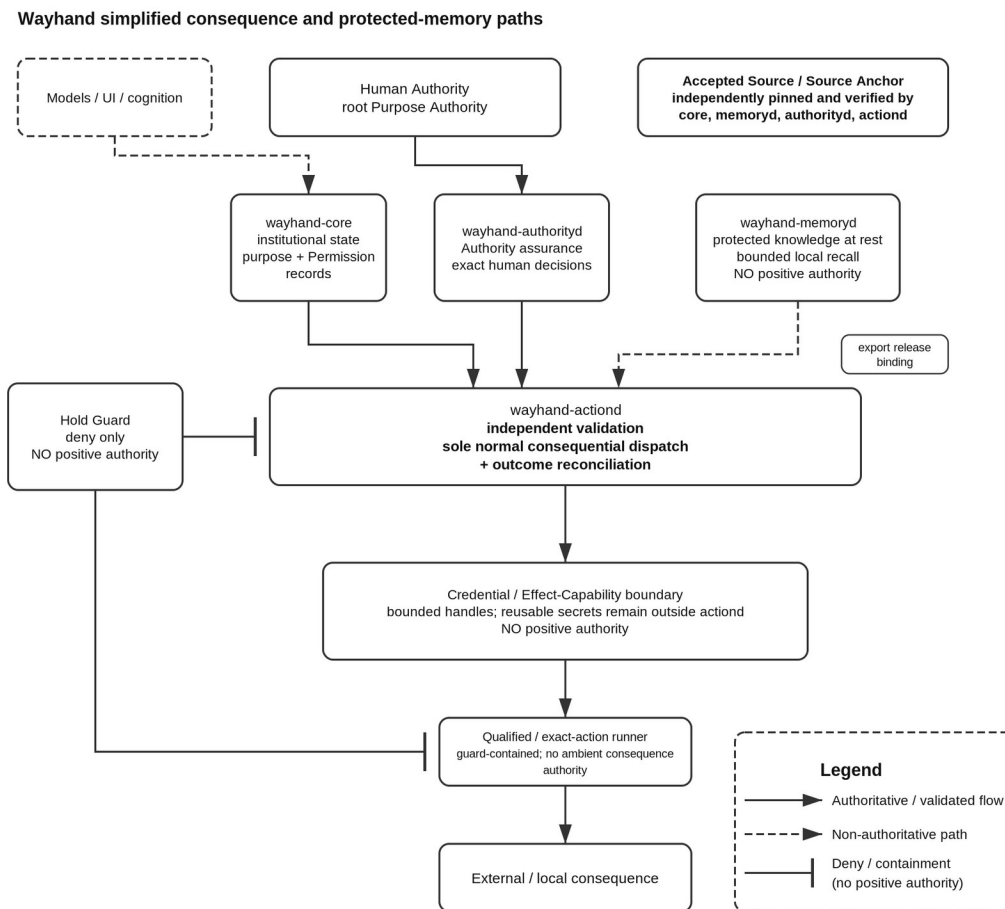


Figure 1. Simplified current Wayhand consequence and protected-memory paths. Solid arrows show validated or enabling flow; dashed arrows show non-authoritative proposal or protected-memory release paths; tee-ended lines show deny/containment rather than authorization. The Source Anchor is shown as a verified prerequisite rather than as a one-way authority source. Local release of protected memory does not authorize external export. For an external effect, *wayhand-actiond* independently validates the request, the *Credential/Effect-Capability* boundary provides only the bounded capability for that request, and a *Hold-Guard-contained* exact-action runner performs the operation.

The separation is deliberately asymmetric. Compromise of *wayhand-core* can corrupt ordinary institutional state and may exploit whatever genuinely current standing and previously delegated authority remain valid, so core is not harmless. But core alone cannot authenticate the human, obtain unrestricted protected-memory archive access or archive keys, release the independent Hold Guard, mint reusable credentials, redefine the independently pinned Source Anchor for the other trust-critical processes, or directly invoke an unrestricted external consequence channel. The design likewise does not claim to eliminate failures that defeat an independent boundary itself: compromise of *wayhand-memoryd*, the human-authentication factor path, the *Credential/Effect-Capability* root, a shared kernel/administrator, or a verifier used by multiple trust-critical components remains a crown-jewel or common-mode limitation.[1]

4. Germination, Seed, and Identity

The lifecycle is easiest to read in four distinct states. A **Public Seed** is a generic release artifact with no inherited Wayhand identity or human Authority. **Germination** uses that Seed to create a new Wayhand identity and establish its local human Authority. **Commissioning** is the transition from the constrained germination environment into ordinary operation under Wayhand's normal controls; commissioning does not itself grant Permission for any particular action. **Recovery** restores the continuity of an already existing Wayhand, while **reproduction** creates a different Wayhand with a different identity.[1]

The Seed therefore contains no preexisting instance identity, live credentials, private instance state, live action queue, or inherited approval that could make a new machine silently become someone else's Wayhand. On the fully compatible minimum path, one verifiable Seed storage/boot device and one functioning x86 host are sufficient. The human establishes a local Authority password and a distinct restore passphrase. Network access, a phone, a second computer, or a hardware authenticator cannot be prerequisites for that minimum path; stronger authentication and durability may be added as stronger profiles and must be described honestly rather than implied to exist when they do not.[1]

The terminal Public Seed is not defined as one permanent operating-system image. The distributed release must, however, include at least one independently verifiable directly bootable removable-media germination environment for the x86 configurations it declares supported, together with the source, builder, specification, and validation material needed to reconstruct that boot path. If firmware does not boot the removable medium automatically, the Seed must also expose a

conspicuous mounted-host entry path. Before a human Authority exists, host inspection is evidence gathering rather than governance: observed machine facts, local documentation, human reports, model inference, contradictions, and unknowns are kept visibly distinct so that an inference is not presented as a measured fact.[1]

Seed size is a measured engineering choice rather than a permanent constitutional number. The complete canonical distribution, required germination/recovery workspace, and reasonable headroom must fit on one ordinary commodity removable-storage device in the smallest declared capacity class that evidence shows to be adequate. Moving to a larger class requires an actual need and a stop condition; spare capacity is not itself a reason to grow the payload.[1]

The human release archetype is the **ordinary stranger**: a previously unexposed person for whom the only assumed operational skill is the ability to access the Seed contents on a functioning x86 device. Wayhand-specific knowledge and Linux, system-administration, partitioning, firmware, cryptographic, or diagnostic expertise may not be assumed. PIAAC is the OECD Programme for the International Assessment of Adult Competencies; Wayhand uses its Level 1 literacy band only as a deliberately low initial communication/interface-design baseline; no participant is tested, scored, certified, or judged qualified to be Authority by PIAAC.[22] The interface and local reasoner may explain concepts and adapt upward as the person demonstrates greater understanding.[1]

Recovery is deliberately not automatic resumption. It uses authorized continuity material to restore the same Wayhand identity, then reconciles the restored state against the current human Authority and current accepted source, invalidates stale credentials, queues, sessions, relationships, and other replayable work as required, and leaves consequential action blocked until the normal current authorization path is re-established. Reproduction instead creates a new independently authorized Wayhand. Merely possessing old state, a Seed, a backup, or a machine that once ran the system does not create present Authority.[1]

5. Reasoning Federation and Self-Engineering

Wayhand requires at least one local offline reasoning path that has been qualified for the offline tasks it claims to perform, including the minimum germination path. No particular model is part of Wayhand's identity. Additional local or external reasoners may collaborate as an advisory group - what the source calls a reasoning federation - but their agreement, confidence, reputation, or vote cannot establish Purpose, Permission, source acceptance, or human approval. Diversity of models is therefore useful for competence and cross-checking without turning model consensus into governance.[1]

Self-engineering is treated as a long-lived capability rather than a license for software to authorize itself. A Wayhand may identify a deficiency and obtain engineering from a local model, another reasoning participant, an external coding system, a future equivalent, or a human. Before committing to a material bespoke mechanism, it reviews relevant standards, primary research, inspectable implementations, and known failure lessons to determine whether a proven mechanism can be reused or narrowly adapted. The source calls this bounded step the **Existing-Space/Mechanism Review (ESMR)**. It is engineering evidence, not permission to install a product, create an account, accept terms, or add a network dependency. The change process then returns to the actual property that is missing, chooses the smallest adequate change, obtains human Authority or Permission where the resulting effect is consequential, tests the candidate, verifies the result, and stops when the property is satisfied.[1]

The core safety argument is intentionally stronger than "the engineering tools are trustworthy." Models, agents, compilers, shells, imported code, benchmarks, and adversarial tools are assumed capable of producing arbitrary candidate software, including convincing malicious replacements for trusted components; Thompson's trusting-trust result is a foundational reminder that clean source inspection alone cannot establish the trustworthiness of the toolchain that produced an executable.[28] Candidate software cannot move itself from candidate status into trusted use, install itself, or authorize itself. It crosses into an immutable untrusted-candidate area, is tested using decisive controls and acceptance criteria that the candidate does not own, is checked for any expansion of the privileges or protected data it could reach, and is introduced while consequential action remains blocked. The design also requires runtime enforcement and a known-good rollback path rather than relying only on pre-installation tests.[1]

Some components enforce the boundaries that protect stored Authority knowledge, exact human decisions, external-effect capability, hold, source verification, and the candidate-admission process itself. Changing that enforcement set is treated as a higher-risk operation: the candidate still cannot decide its own tests, and deployment requires a fresh exact human decision over the identified replacement and material trust-boundary change, activation while the system remains held, and post-activation verification before normal protected-memory or external-effect capability is restored. The source calls this set the **Crown-Jewel Enforcement Set (CJES)**, but the important property is the independent replacement rule, not the acronym.[1]

6. Dependencies, Locality, and Replaceability

Wayhand is compatible with local-first goals but is not simply a local-first application. Local-first software emphasizes offline operation, local ownership/control of data, privacy, longevity, and collaboration without making a remote server the sole authoritative copy.[5] Wayhand independently shares several of those locality and continuity goals, but applies them to a different system problem: preserving human Authority, reasoning availability, protected memory, recovery, self-engineering, software provenance, and replaceability of components or hosts without silently replacing the Wayhand itself.[1] The local-first reference is therefore conceptual precedent, not a claim that Wayhand adopts the local-first data model or CRDT architecture.

External dependencies are permitted after germination, but they are admitted deliberately rather than appearing as a side effect of a recommendation. Before a service, model, account, provider, protocol endpoint, device, or other dependency is connected, the human must be able to understand the material consequences: what the dependency is and can do; what becomes trusted or opaque; what data, credentials, and network access it receives; what persists or runs in the background; what it costs and

requires to maintain; what happens if it becomes unavailable; how it is removed or revoked; what state or lock-in may remain; what is still unknown; and what relevant alternatives exist. A favorable benchmark or model recommendation cannot by itself install software, create an account, spend money, grant credentials, or accept terms.[1]

For each admitted dependency, Wayhand keeps a bounded local evidence package containing the version-relevant documentation and observed technical evidence actually needed to operate, diagnose, recover, extract state from, replace, or retire the dependency. The normative source calls this the **Dependency Evidence Package**. It is refreshed when material events make the stored evidence stale, not by continuously crawling a vendor. Legal, licensing, contractual, or access uncertainty is recorded separately as sourced fact, interpretation, uncertainty, alternatives, and consequence; an unresolved material legal-risk choice returns to the human Authority rather than becoming an automatic model prohibition or permission.[1]

Replaceability is tested after representative use, because real accumulated state can create coupling that a clean-install demonstration does not reveal. Configuration, history, migration burden, residual state, exit cost, and dependence on incumbent-specific behavior are therefore evidence. The Authority may still choose a more coupled incumbent when its benefits justify the tradeoff, but the coupling remains visible. Conversely, a temporary component whose required function has been satisfied elsewhere may be retired when retirement is the smallest adequate intervention and continuity obligations remain satisfied; replaceability does not imply perpetual replacement of everything that once existed.[1]

7. Federation Without Contagion

Separate Wayhands do not merge their identity or governing state merely because they cooperate. Each retains its own human Authority, credentials, protected stores, recovery path, accepted-source state, and explicit records for each relationship. A Wayhand can maintain several relationships at once; the state created for one relationship is tied to that relationship and to its current version so that old or cross-relationship records cannot silently be reused elsewhere.[1][21]

The practical rule is **non-contagion**: a capability granted for one relationship does not automatically become a capability of either whole Wayhand. For example, one Wayhand may lend another a bounded amount of compute for a benchmark. That lease may allow the agreed computation for the agreed interval, but it does not give the borrowing peer the right to install software on the lender, inherit its credentials, approve later actions, recover its state, change its accepted source, or become Authority for it. Suspending or revoking one relationship likewise should not change a separate relationship or the local Wayhand's identity.[1]

This differs in emphasis from agent-interoperability protocols. A2A provides a standard for communication among independent agents, while MCP standardizes access to tools and resources and includes protocol-level authentication/authorization mechanisms and security guidance.[12][13] They are relevant candidate transport or integration mechanisms, not preselected Wayhand dependencies. A2A v1.0 requires authorization checks but leaves authorization boundaries to each agent's authorization model; its in-task TASK_STATE_AUTH_REQUIRED mechanism is a workflow for requesting authorization/credentials, not a definition of Wayhand Purpose, Permission, relationship standing, or human Authority.[13] Wayhand therefore layers its own constitutive checks over any such transport: protocol participation, authentication, or a protocol credential cannot itself create Wayhand identity, Purpose, Permission, source acceptance, recovery rights, relationship standing, or human Authority.[1] SAGA is a closer contemporary neighbor for user-governed multi-agent relationships because it adds lifecycle oversight, user-defined inter-agent access policy, and cryptographic access-control tokens; Wayhand differs by keeping each Wayhand's relationship standing and constitutive identity/Authority/source/recovery state local and non-contagious rather than depending on a central Provider to define those facts.[1][32]

8. Threat Model

The Phase 1 threat model gives special attention to two protected asset classes: **durable non-public Authority knowledge stored for later recall** and **the capability to cause consequential effects**. It also protects the integrity and currentness of human decisions, Purpose, Permission, accepted source, hold state, recovery state, and inter-Wayhand relationships. The architecture does not claim that process separation defeats a shared kernel, administrator, cryptographic root, trusted-time source, or common verifier that all relevant boundaries depend on; those are explicit common-mode limits.[1]

Threat / failure	Required architectural response	Explicit residual boundary
Compromised model, UI, core, adapter, renderer, or engineering workshop	Treat reasoning and workshop output as advisory/untrusted. Keep stored private memory and reusable credentials outside ordinary cognition and core; require the independent action path for consequential effects and a separate independently controlled admission path for candidate software.	Plaintext already legitimately released into a compromised active context can be exposed, and ordinary institutional state in core can be corrupted. A defect in a shared kernel, administrator, or common verifier may cross multiple boundaries.
Attack on protected stored memory	wayhand-memoryd owns the protected at-rest content and retrieval state, limits cumulative recall, requires stronger current human assurance for deep/bulk access, and treats local recall as separate from authorization to export the material.	Compromise of memoryd or its cryptographic root defeats the protected-at-rest claim for that boundary; the design does not describe this as ordinary contained damage.

Threat / failure	Required architectural response	Explicit residual boundary
Compromised human-decision path	Isolate wayhand-authority; bind each significant human decision to the exact decision content/request and record it separately from model presentation or core state.	In the minimum password-only profile, compromise of the applicable password/authority path can forge the local approval evidence that profile relies on. A hardware factor without an independently trusted transaction display proves a factor event, not necessarily that the human saw the same decision content.
Compromised action/runner path or credential theft	Keep reusable secrets outside wayhand-action and cognition. Give action and the guard-contained runner only independently validated capabilities bounded to the actual request, target/destination, data/resource scope, use count or lease, expiry, and current hold/source/recovery state.	A still-valid bounded handle may permit the exact effect it was created for. Compromise of the credential/capability provider exposes whatever authority that provider's own isolation and scope permit.
Malicious candidate for a trusted component	Assume the workshop can construct arbitrary malicious software. Candidate software cannot choose the decisive tests that admit it, install itself, or silently expand its trust footprint. Changes to the enforcement components protecting memory, human decisions, consequence, hold, source verification, or promotion require fresh human acceptance and activation under hold with rollback/post-verification.	Finite testing cannot prove the absence of every delayed or rare-trigger malicious behavior. Runtime containment and capability limits remain part of the security claim.
Replay, stale restore, rollback of governing state, or clock rollback	Tie consequential requests and receipts to current versions of source, Purpose, Authority state, memory state, relationships, recovery state, process identity, and hold state; after restore, reconcile against what is current before allowing new consequential action.	Compromise of trusted time or a verifier shared by the relevant boundaries remains a common-mode risk that must be measured rather than assumed away.
Compromised peer Wayhand	Keep each Wayhand's identity, Authority root, credentials, stores, source state, recovery, and relationship records independent; do not accept transitive privilege from a peer.	Deployments that actually share a kernel or administrator cannot claim the same containment as genuinely separated hosts.
Loss of network, provider, or preferred model	Preserve local governing state, offline germination/recovery, and at least one qualified local reasoning path.	Capability and performance may narrow; Wayhand does not claim continued access to an unavailable external service.
Unsafe, unknown, stale, or ambiguous state	Fail closed at the narrowest defensible affected consequential scope through the independent Hold Guard when a required fact cannot be established. Releasing hold authorizes nothing and resumes nothing by itself.	Hold can prevent new dispatch; it cannot undo an effect that has already occurred in the outside world.

Wayhand also rejects **authorization by composition**. A broader consequential effect cannot become authorized merely by splitting it into individually permitted steps, transforming or encoding intermediate data, changing labels, relaying through another component, or presenting the same material effect through a different interface. If the aggregate effect would require a different Purpose, Permission, destination, data scope, relationship, resource bound, or other governing fact, the aggregate effect must satisfy that requirement directly. Classification likewise follows the source-defined decisive facts, not names, wrappers, routes, caller-claimed roles, or presentation wording.[1][21]

9. Evidence and Proof Architecture

Wayhand separates three questions that are easy to blur together. First, **can a technical property be reproduced with an exact procedure and pass/fail criterion?** Second, **does a property depend on an actual person's comprehension, burden, or unaided ability, so that internal testing cannot honestly establish it?** Third, **are both kinds of evidence current for the same exact source, build, environment, and scope?** The proof system keeps those answers separate rather than treating a large technical test suite as proof of human usability.[1]

Machine-checkable properties use technical-conformance evidence: exact source/build identities, fixtures, procedures, environments, outputs, observer/tool identities, limitations, and reproducible results. A clean reconstruction by an independent machine or team can still be technical evidence when the success criterion is completely specified. Human comprehension, sustained operator burden, and an ordinary stranger's ability to carry out a procedure without project-specific coaching are external-fitness properties; internal simulations can prepare those tests but cannot close the human claim.[1] Security-usability work by Whitten and Tygar and Cranor supports the methodological premise that security properties depending on human performance must be evaluated as human-in-the-loop properties rather than inferred solely from technical mechanism quality. [19][29]

Technical closure is always limited to a named scope and exact identities. It means that every applicable technical claim in that scope has passed at the required evidence strength, no unresolved technical blocker or known regression remains, and any source-required preliminary external probe has not exposed an unresolved material finding that plausibly points to a technical defect. External-fitness closure means that every applicable human or genuinely external fitness claim has also passed while the technical evidence it depends on remains current. Overall Wayhand fitness for a scope requires both, together with the original mixed/stage gates and any separately required human/source/release decision. These proof states do not activate the system, create Authority, or grant Permission. The normative source names them `TECHNICAL_CLOSED(scope)`, `EXTERNAL_FITNESS_CLOSED(scope)`, and `WAYHAND_FITNESS_ESTABLISHED(scope)`. [1]

Evidence is durable history, but closure is not permanent. If later human or independent testing reveals a plausible design or implementation defect, the relevant technical claim is reopened and returned to the earliest affected source or roadmap stage; the earlier result remains recorded rather than being rewritten away. If the later finding is genuinely external-only and changes no technical claim, it still blocks external-fitness closure. This allows usability or independent evidence to falsify an apparently successful technical design when the newly observed failure is actually caused by the design.[1]

Established software-supply-chain mechanisms are relevant to parts of this proof discipline, but their assurance scopes are narrower than Wayhand fitness. Reproducible Builds defines reproducibility as any party recreating bit-for-bit identical specified artifacts given the same source code, build environment, and build instructions; in-toto describes and verifies signed supply-chain step metadata; SLSA structures source/build provenance and supply-chain hardening; and TUF addresses software-update repository authenticity, freshness, rollback, key-compromise, and related distribution attacks.[7][8][9][10][11] These are candidate mechanism families or evidence patterns for particular Seed, source, build, update, and dependency-provenance properties. The Wayhand source requires separate mechanism selection, qualification, and dependency admission before a production binding is created, and none of these mechanisms by itself establishes Wayhand source standing, human Authority, Permission, stage PASS, protected-memory/consequence isolation, recovery fitness, ordinary-stranger fitness, or overall Wayhand fitness.[1]

10. Relationship to Existing Work

Wayhand is positioned against established mechanism families rather than claiming that those mechanisms are new. Some cited works provide broad governance or human-factors context; some are foundational security precedents; some are concrete candidate mechanisms that a future implementation could select after separate qualification; and some are modern neighboring architectures that solve narrower portions of the same agent-security problem. **A citation in this section does not mean that Wayhand has adopted that mechanism, inherited its assurance, or admitted an external dependency.** The accepted source instead requires a bounded Existing-Space/Mechanism Review before material bespoke mechanism commitment and requires any production binding to pass Wayhand's own selection, qualification, provenance, license, dependency, and proof gates.[1]

Adjacent work	What it actually contributes	Correct Wayhand relationship
AI risk governance (NIST AI RMF / GAI Profile)	Organizational frameworks for identifying, governing, measuring, and managing AI risks, including human oversight and third-party/system context.[2][14]	Governance context only. Wayhand specifies a narrower local runtime and evidence architecture; it neither implements nor replaces an organizational AI risk-management program.
Constitutional AI	A model-training/alignment method that uses a written set of principles plus AI-generated critique/preferences to shape model behavior.[4]	Terminological and conceptual contrast. Wayhand's constitution constrains accepted source, runtime powers, and role separation; compliance by a model with written principles is not human Authority or Wayhand standing. Constitutional AI is not a Wayhand control mechanism.
ReAct	An illustrative language-agent method that interleaves model reasoning traces with task-specific actions in an external environment.[3]	Background for the reasoner/tool-action paradigm that Wayhand must safely surround. Wayhand does not require ReAct and makes no contribution claim about the reasoning algorithm itself.
AgentDojo	A dynamic benchmark/environment for evaluating prompt-injection attacks and defenses when agents execute tools over untrusted data.[24]	Threat and evaluation precedent, not an architecture adopted by Wayhand. It supports treating model/tool interaction with untrusted data as an adversarial surface that must not be the sole security boundary.
CaMeL, IsolateGPT, and ACE	CaMeL extracts trusted-query control/data flow and uses capabilities to constrain unauthorized data flow; IsolateGPT isolates third-party LLM apps and mediates cross-boundary interaction; ACE separates trusted abstract planning from concrete app binding, statically checks information-flow constraints, and enforces data/capability barriers during execution. [25][30][31]	Close system-security neighbors to Wayhand's premise that an LLM or app is not itself a trusted consequence boundary. Wayhand's claimed scope is broader and different: accepted-source standing, human Purpose/assurance/Permission, protected-memory release, deny-only hold, exact consequential execution, recovery, self-engineering admission, non-contagious federation, terminal Seed reconstruction, and staged proof remain separate Wayhand concerns. Similarity does not imply mechanism adoption or equivalent guarantees.
Local-first software	A software design approach emphasizing local ownership/control of data, offline operation, privacy, longevity, and collaboration without making a remote server the sole authoritative copy.[5]	Conceptual precedent for locality, user control, and service independence. Wayhand does not claim to adopt the local-first CRDT/data architecture; it independently carries related continuity goals into identity, Authority, reasoning, protected memory, recovery, and replaceability.
Decentralized Identifiers (DIDs)	A standardized representation and resolution model for decentralized identifiers, DID documents, verification methods, and controller proof of control. [6]	Candidate representation/interoperability mechanism only. A DID, a DID-controller relationship, or proof of DID control would not by itself create a Wayhand instance identity, human Authority, Purpose, Permission, relationship acceptance, or source standing.

Adjacent work	What it actually contributes	Correct Wayhand relationship
zCaps and Macaroons	Capability-style authorization/delegation mechanisms; Macaroons support attenuation through contextual caveats, while zCaps specify delegated authorization-capability chains and restrictions.[15][26]	Candidate mechanism ideas for bounded delegation or capability representation. They are not Wayhand Authority and cannot substitute for Purpose, human assurance, Permission, source standing, aggregate-consequence checks, or the independent Credential/Effect-Capability boundary. The cited zCap document is an incubating Community Group draft, not a W3C Recommendation.
EROS and Capsicum	EROS is a capability-oriented operating system; Capsicum adds practical capability mode and capability-based compartmentalization to UNIX applications.[17][27]	Foundational/substrate precedents for least-authority execution and compartmentalization. A future Wayhand substrate may reuse a qualified mechanism of this class, but neither system is specified as a required Wayhand dependency.
seL4	A capability-based microkernel whose cited 2009 work establishes machine-checked functional correctness of the implementation relative to its abstract specification.[18]	High-assurance substrate precedent. The citation does not prove Wayhand isolation, does not mean Wayhand uses seL4, and should not be read as transferring seL4's verification result to a Wayhand deployment or shared host.
TUF	A software-update framework designed to resist repository/distribution attacks including rollback/freeze, mix-and-match, mirror, and key-compromise scenarios.[7]	Candidate mechanism for bounded update/distribution authenticity and freshness properties. TUF metadata cannot create Wayhand source acceptance or Source Anchor standing by itself.
in-toto, SLSA, and Reproducible Builds	in-toto describes/verifies supply-chain steps and signed link metadata; SLSA structures source/build provenance and supply-chain hardening; Reproducible Builds defines independent artifact reproducibility.[8][9][10][11]	Candidate evidence/provenance mechanisms. They can strengthen build and supply-chain evidence, but do not establish human/external fitness, source standing, dependency admission, stage PASS, release authority, or overall Wayhand fitness. The in-toto specification supplies current mechanism semantics; the 2019 paper supplies research lineage.
Toolchain trust / trusting trust	Thompson's compiler attack demonstrates that clean source inspection alone cannot establish the trustworthiness of the toolchain that produced executable bytes.[28]	Direct precedent for Wayhand's hostile-workshop assumption: candidate-producing machinery may be malicious and therefore cannot be the sole authority for its own admission, decisive qualification, standing, or activation.
MCP	A protocol for model/agent access to tools, resources, and other interaction surfaces, with protocol-level authorization and security guidance. [12]	Candidate tool/resource transport only. MCP authentication or OAuth authorization is not Wayhand human Authority or Permission and does not bypass Wayhand's independent consequence, protected-memory, source, hold, or capability checks.
A2A	A protocol for discovery, communication, and collaborative tasks among independent agents. A2A v1.0 requires authorization checks while leaving authorization boundaries to each participating agent's authorization model, and provides an in-task mechanism for requesting authorization/credentials. [13]	Candidate federation transport only. This separation is compatible with Wayhand layering local relationship acceptance and consequence checks above the protocol; A2A participation, authentication, task state, or credentials cannot itself create Wayhand identity, Authority, Purpose, Permission, relationship standing, source standing, or recovery rights.
SAGA	A user-governed multi-agent architecture with agent lifecycle oversight, user-defined inter-agent access-control policy, a central Provider, and cryptographic access-control tokens.[32]	Direct federation/governance neighbor. Wayhand shares the goal of user-governed agent relationships but requires each Wayhand's constitutive identity, human Authority, source, recovery, credentials, and relationship standing to remain locally defined and non-contagious; a central provider or peer protocol is not constitutive local-runtime authority.
Classical protection principles	Economy of mechanism, fail-safe defaults, complete mediation, separation of privilege, least privilege, and psychological acceptability.[16]	Foundational design precedent. Wayhand independently instantiates analogous principles through narrow consequential mediation, deny-only controls, least-capability execution, role separation, and explicit human-fitness evidence; the citation is conceptual lineage, not a proof that the implementation satisfies those principles.
Whitten/Tygar and Cranor	Empirical evidence and a human-in-the-loop framework showing that security can depend on whether non-malicious humans understand and successfully perform security-critical tasks.[19][29]	Methodological precedent for Wayhand's separation of technical conformance from human/external fitness. The cited works do not prove Wayhand is usable; they support testing human-dependent security claims with actual human evidence rather than inferring them from technical tests or model simulation.
Notary	A separate trusted approval environment that isolates exact security-critical transaction approval from a potentially compromised ordinary host.[20]	Trusted-approval precedent for separating an exact human decision surface from ordinary computation. Wayhand does not adopt Notary's hardware architecture and uses a broader persistent human-assurance/lifecycle design.

Bounded contribution claim. The bounded adjacent-work review summarized above did not identify a single specification that combines all of the following in one lifecycle: an actual human whose attributable decisions are required for the powers reserved to Authority; separate standing for accepted source, current Purpose, human assurance, Permission, protected-memory release, and consequential execution; offline germination from an Authority-free Public Seed; persistent identity across model/provider/host replacement; a self-engineering workshop whose output cannot authorize its own installation; non-contagious relationships between multiple Wayhands; replaceability tested after representative use; a terminal independently reconstructable Public Seed; and a proof model that keeps reproducible technical conformance separate from human/external fitness. Close contemporary work such as CaMeL, IsolateGPT, ACE, and SAGA materially narrows parts of the comparison space and is therefore included explicitly rather than treated as generic agent-security background.[25][30][31][32] This remains a bounded literature-position statement, not proof that no comparable architecture exists. A formal novelty, patent, or priority claim would require a broader systematic prior-art and academic review than this whitepaper attempts.[2]-[20][24]-[32]

11. Public Seed as the Terminal Project Boundary

The originating project is designed to end rather than become a service that every future Wayhand must depend on. Its public milestone is one independently reconstructable, Authority-free, directly bootable Public Seed with no required continuing upstream service, release train, or support institution. The release must include the accepted source, reconstruction material, validation/provenance evidence, at least one offline reasoning route for the claimed minimum path, and the ordinary-stranger start/recovery material needed to verify and germinate the Seed without undocumented project knowledge.[1]

Before terminal release, the human-facing claims must be tested with people who have not previously participated in the project; the accepted source requires at least three fresh ordinary strangers on the intact fully compatible terminal path. On a deliberately working supported hardware setup, an ordinary stranger must be able to reach the required entry, germination, clean-boot restore-simulation, first-use, reconstruction, and representative replacement/rollback outcomes without Internet access, repository access, original-author assistance, or project-specific technical coaching. Questions, false starts, and presentation friction are recorded rather than hidden. A project-specific explanatory intervention makes that affected human result non-passing.[1]

Deliberately incompatible or partially compatible setups are tested separately. Those tests are supposed to stop or degrade for the right reason and explain, in ordinary language, what is known, what is unavailable or incompatible, what consequence follows, and what safe next step is actually established. Failure on the intentionally working supported setup cannot be relabeled as a successful "blocker" test. PIAAC Level 1 remains only the low initial communication-design baseline; the participant does not need a PIAAC credential or a measured score, and the interface may teach or adapt upward when needed.[1][22]

A germinated Wayhand may later evolve, admit dependencies, replace models, or acquire new capabilities under its own human Authority. The originating project is not intended to retain governing power over those later systems. This terminality is part of the independence claim: a system is less independent if ordinary operation or reconstruction requires the continued existence or permission of the institution that originally produced its Seed.[1]

12. Current Status and Falsifiability

The current accepted implementation-and-proof source is v0.15.05, exact SHA-256 77808f787ab7750b56f2c696baeb9bcd7b0f7bee47bbb5eb0cdfa60fc9000e58, accepted by external record WAYHAND-SOURCE-ACCEPTANCE-2026-08-20-01. The controlling constitutional source is v0.2, exact SHA-256 5a038e209b9fc37e8f15c8646d32884a8c840f318b374f1e09fa3deaec7beb1b, accepted by external record WAYHAND-CONSTITUTION-ACCEPTANCE-2026-08-17-01.[1][21]

As of this snapshot, two preparatory engineering gates have passed: the source-derived control/test mapping (**R-00**) and development-host measurement/preservation baseline (**R-01**). The first runtime trust-boundary implementation/proof stage (**R-02**) has **not** passed. R-02 is the stage that must establish and prove the initial separated core/authorityd/memoryd/actiond, source/hold anchors, Hold Guard, capability boundary, canonical records, and related trust-boundary behavior on the intended development substrate.[1] A frozen 84-member implementation/build candidate and preserved unprivileged validation result (23 passed, 0 failed) exist for continued engineering, but those are preliminary evidence only; they do not prove the required live isolation, privileged behavior, effect gating, rollback, or other trusted-substrate properties. Required live R-02 proof remains pending. The non-normative engineering-status note [23] binds these status statements to the exact R-00, R-01, R-02 planning, build-identity, and validation records from which they are derived.

No Wayhand has been germinated, commissioned, recovered, or activated. No technical-closure, external-fitness-closure, or overall Wayhand-fitness claim is made. In the formal source terminology, there is no current TECHNICAL_CLOSED, EXTERNAL_FITNESS_CLOSED, or WAYHAND_FITNESS_ESTABLISHED claim for the project.[1]

The architecture is therefore falsifiable in practical engineering terms. It fails its claims if the required isolation/capability boundaries cannot be implemented within acceptable burden; if a compromised ordinary component can enumerate protected stored Authority knowledge or obtain unrestricted external-effect capability; if hold or replay protections fail; if human Authority is not actually intelligible in sustained use; if the one-host minimum cannot preserve the stated protection profile; if replaceability collapses after legitimate accumulated state; if recovery requires undocumented project knowledge; if a compromised peer crosses supposedly independent relationship boundaries; if self-engineering candidates can influence the controls that decide their own admission; or if independent ordinary strangers cannot reconstruct and germinate the terminal Seed on the supported path. The proof architecture is designed to record those outcomes as failures or reopened claims rather than convert them into narrative success.[1]

Conclusion

Wayhand asks whether an AI-enabled local computing system can become substantially more capable without requiring its human Authority to become less authoritative. Its answer is architectural rather than motivational. Competence is kept separate from standing; current Purpose, human assurance, Permission, accepted source, hold state, and effect capability remain independently checkable; durable non-public Authority knowledge and the capability to change protected local state or the outside world are protected as distinct high-consequence asset classes; models and engineering environments remain non-authoritative; recovery does not silently revive stale authority; and relationships with peer systems do not automatically transfer privilege.[1][21]

The design deliberately looks to established mechanisms from protection architecture, capability security, provenance, reproducible builds, local-first systems, usable security, and agent interoperability rather than assuming bespoke invention is necessary. A cited mechanism becomes part of Wayhand only if it is separately selected, qualified, and admitted for an exact source-defined property; upstream claims do not transfer standing or proof. The proposed contribution is the lifecycle and proof discipline that connect those property classes: Authority-free Seed distribution, new-instance germination, persistent identity, human Authority, bounded consequential action, protected memory, self-engineering, recovery, replaceability, non-contagious federation, and evidence that distinguishes reproducible technical conformance from properties that require actual human or independent observation.

This publication is therefore a specification-position document, not evidence that Wayhand already works. The appropriate test is whether implementation can produce the exact technical and external evidence required by the accepted source, and whether later evidence is allowed to reopen claims when reality contradicts the design assumptions.

References

- [1] TEKWITCH. *Wayhand Complete Integrated Architecture and Design - Minimum-Complexity Production-Proof Source*, v0.15.05, accepted 20 Aug. 2026; SHA-256 77808f787ab7750b56f2c696baeb9bcd7b0f7bee47bbb5eb0cdfa60fc9000e58. Normative implementation-and-proof source. Acceptance record: WAYHAND-SOURCE-ACCEPTANCE-2026-08-20-01.
- [2] Tabassi, E. *Artificial Intelligence Risk Management Framework (AI RMF 1.0)*. NIST AI 100-1, National Institute of Standards and Technology, 26 Jan. 2023. DOI: 10.6028/NIST.AI.100-1.
- [3] Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K., and Cao, Y. *ReAct: Synergizing Reasoning and Acting in Language Models*. ICLR 2023; arXiv:2210.03629. DOI: 10.48550/arXiv.2210.03629.
- [4] Bai, Y. et al. *Constitutional AI: Harmlessness from AI Feedback*. arXiv:2212.08073, 2022. DOI: 10.48550/arXiv.2212.08073.
- [5] Kleppmann, M., Wiggins, A., van Hardenberg, P., and McGranaghan, M. *Local-First Software: You Own Your Data, in spite of the Cloud*. Proceedings of Onward! 2019, pp. 154-178. DOI: 10.1145/3359591.3359737.
- [6] W3C. *Decentralized Identifiers (DIDs) v1.0: Core Architecture, Data Model, and Representations*. W3C Recommendation, 19 July 2022.
- [7] The Update Framework (TUF). *The Update Framework Specification*, v1.0.36 (specification dated 5 Aug. 2026; release published 10 Aug. 2026). <https://github.com/theupdateframework/specification/blob/v1.0.36/tuf-spec.md> (accessed 22 Aug. 2026).
- [8] in-toto Project. *in-toto Specification*, v1.0.0, 2 June 2023. <https://github.com/in-toto/specification/blob/v1.0.0/in-toto-spec.md> (accessed 22 Aug. 2026).
- [9] SLSA Community. *SLSA Specification*, v1.2, Approved, released 24 Nov. 2025. <https://slsa.dev/spec/v1.2/> (accessed 22 Aug. 2026).
- [10] Reproducible Builds. *Definitions - When is a build reproducible?* <https://reproducible-builds.org/docs/definition/> (accessed 22 Aug. 2026).
- [11] Torres-Arias, S., Afzali, H., Kuppusamy, T. K., Curtmola, R., and Cappos, J. *in-toto: Providing farm-to-table guarantees for bits and bytes*. 28th USENIX Security Symposium (USENIX Security 19), 2019, pp. 1393-1410.
- [12] Model Context Protocol. *Specification*, protocol revision 2026-07-28. <https://modelcontextprotocol.io/specification/2026-07-28> (accessed 22 Aug. 2026).
- [13] A2A Protocol Working Group. *Agent2Agent (A2A) Protocol Specification*, v1.0.0, released 12 Mar. 2026. <https://a2a-protocol.org/v1.0.0/specification> (accessed 22 Aug. 2026).
- [14] Autio, C., Schwartz, R., Dunietz, J., Jain, S., Stanley, M., Tabassi, E., Hall, P., and Roberts, K. *Artificial Intelligence Risk Management Framework: Generative Artificial Intelligence Profile*. NIST AI 600-1, National Institute of Standards and Technology, 26 July 2024. DOI: 10.6028/NIST.AI.600-1.
- [15] W3C Credentials Community Group. *Authorization Capabilities*, v0.4.0-draft. Active Community Group draft; <https://w3c-ccg.github.io/zcap-spec/> (accessed 22 Aug. 2026). This is an incubating draft, not a W3C Recommendation.
- [16] Saltzer, J. H., and Schroeder, M. D. *The Protection of Information in Computer Systems*. Proceedings of the IEEE 63(9), 1278-1308, 1975. DOI: 10.1109/PROC.1975.9939.
- [17] Shapiro, J. S., Smith, J. M., and Farber, D. J. *EROS: A Fast Capability System*. 17th ACM Symposium on Operating Systems Principles (SOSP), 1999, pp. 170-185. DOI: 10.1145/319151.319163.
- [18] Klein, G., Elphinstone, K., Heiser, G., Andronick, J., Cock, D., Derrin, P., Elkaduwe, D., Engelhardt, K., Kolanski, R., Norrish, M., Sewell, T., Tuch, H., and Winwood, S. *seL4: Formal Verification of an OS Kernel*. 22nd ACM Symposium on Operating Systems Principles (SOSP), 2009, pp. 207-220. DOI: 10.1145/1629575.1629596.
- [19] Whitten, A., and Tygar, J. D. *Why Johnny Can't Encrypt: A Usability Evaluation of PGP 5.0*. 8th USENIX Security Symposium (USENIX Security 99), 1999, pp. 169-184.
- [20] Athalye, A., Belay, A., Kaashoek, M. F., Morris, R., and Zeldovich, N. *Notary: A Device for Secure Transaction Approval*. 27th ACM Symposium on Operating Systems Principles (SOSP), 2019, pp. 97-113. DOI: 10.1145/3341301.3359661.
- [21] TEKWITCH. *Wayhand Constitutional Prohibitions and Governing Principles*, v0.2, accepted 17 Aug. 2026; SHA-256 5a038e209b9fc37e8f15c8646d32884a8c840f318b374f1e09fa3deaec7beb1b. Acceptance record: WAYHAND-CONSTITUTION-ACCEPTANCE-2026-08-17-01.
- [22] OECD. *Survey of Adult Skills - Reader's Companion: 2023*. OECD Skills Studies, OECD Publishing, Paris, 2024; revised version Jan. 2026. DOI: 10.1787/3639d1e2-en. Used here only to identify PIAAC and contextualize the Level 1 literacy band; Wayhand's use of that band as an interface-design baseline is defined by the Wayhand source, not by OECD.
- [23] TEKWITCH. *Wayhand Engineering Status Evidence Note*, v0.1, 22 Aug. 2026; SHA-256 1cadddee9b3f1dd31c510c12fe2110ec57ca0d5aa1d7b163188a0cdd82d2a9ca. Non-normative publication-support index binding the R-00/R-01/R-02 development-status statements in this paper to exact project engineering records.
- [24] Debenedetti, E., Zhang, J., Balunovic, M., Beurer-Kellner, L., Fischer, M., and Tramèr, F. *AgentDojo: A Dynamic Environment to Evaluate Prompt Injection Attacks and Defenses for LLM Agents*. Advances in Neural Information Processing Systems 37 (NeurIPS 2024), Datasets and Benchmarks Track, pp. 82895-82920. DOI: 10.52202/079017-2636.
- [25] Debenedetti, E., Shumailov, I., Fan, T., Hayes, J., Carlini, N., Fabian, D., Kern, C., Shi, C., Terzis, A., and Tramèr, F. *Defeating Prompt Injections by Design*. Presented at the IEEE Conference on Secure and Trustworthy Machine Learning (SaTML) 2026; arXiv:2503.18813. DOI: 10.48550/arXiv.2503.18813.
- [26] Birgisson, A., Politz, J. G., Erlingsson, U., Taly, A., Vrable, M., and Lentczner, M. *Macaroons: Cookies with Contextual Caveats for Decentralized Authorization in the Cloud*. Network and Distributed System Security Symposium (NDSS), 2014. DOI: 10.14722/ndss.2014.23212.

- [27] Watson, R. N. M., Anderson, J., Laurie, B., and Kennaway, K. *Capsicum: Practical Capabilities for UNIX*. 19th USENIX Security Symposium (USENIX Security 10), 2010, pp. 29-45.
- [28] Thompson, K. *Reflections on Trusting Trust*. Communications of the ACM 27(8), 761-763, 1984. DOI: 10.1145/358198.358210.
- [29] Cranor, L. F. *A Framework for Reasoning About the Human in the Loop*. Usability, Psychology, and Security 2008 (UPSEC 08), USENIX Association, San Francisco, Apr. 2008.
- [30] Wu, Y., Roesner, F., Kohno, T., Zhang, N., and Iqbal, U. *IsolateGPT: An Execution Isolation Architecture for LLM-Based Agentic Systems*. Network and Distributed System Security Symposium (NDSS), 2025. DOI: 10.14722/ndss.2025.241131.
- [31] Li, E., Mallick, T., Rose, E., Robertson, W., Oprea, A., and Nita-Rotaru, C. *ACE: A Security Architecture for LLM-Integrated App Systems*. Network and Distributed System Security Symposium (NDSS), 2026. DOI: 10.14722/ndss.2026.230352.
- [32] Syros, G., Suri, A., Ginesin, J., Nita-Rotaru, C., and Oprea, A. *SAGA: A Security Architecture for Governing AI Agentic Systems*. Network and Distributed System Security Symposium (NDSS), 2026. DOI: 10.14722/ndss.2026.230869.

Source and Publication Note

Publication verification

Normative design-source identity: WAYHAND-MINIMUM-COMPLEXITY-INTEGRATED-SOURCE v0.15.05; SHA-256 77808f787ab7750b56f2c696baeb9bcd7b0f7bee47bbb5eb0cdfa60fc9000e58. Controlling constitutional source: Wayhand constitutional v0.2; SHA-256 5a038e209b9fc37e8f15c8646d32884a8c840f318b374f1e09fa3deaec7beb1b. A public release of this paper is intended to carry the exact source bytes and external acceptance records so readers can independently verify the cited standing. Development-status statements are separately indexed by the non-normative engineering-status note [23]; they do not alter source standing or substitute for the underlying stage evidence.

This whitepaper is explanatory and non-normative. It intentionally compresses a substantially larger implementation-and-proof specification. If wording here conflicts with the exact accepted Wayhand source or the higher-precedence constitution, those accepted exact-byte sources control. The v0.15.05 source's own internal pre-acceptance status text is immutable historical self-description; current standing is established by its separate human-owned exact-byte acceptance record. This publication candidate creates no implementation, activation, commissioning, Permission, dependency admission, relationship, recovery, or release authority.

The literature comparison is bounded to the cited adjacent work and should not be read as a comprehensive patent, academic-priority, or prior-art search. This candidate is intended for release under Creative Commons Attribution 4.0 International (CC BY 4.0). The DOI and final publication-verification manifest will be inserted during final Zenodo package preparation.