Research Article

HAMU: Autonomy Without Sovereignty — Cognition, Continuity, and Authority in Persistent AI Systems

Raúl Badillo[1]

1. Parabelo, Mexico

HAMU (Host-Aware Mediation Unit) is a conceptual architecture for hosting persistent artificial intelligence (AI) agents that can carry out assignments, check their work, and correct mistakes. Plans and results can outlast the permissions that governed the original task. HAMU calls it authority collapse when a host treats a saved plan, an external document, or an earlier approval as permission to act beyond what was delegated.

In the workspace their host provides, agents can use tools, consult information, and propose operations. Before an operation takes effect, a mediator checks it against policies, grants, and records held in protected custody. HAMU models the proposal as a transition. Its semiformal rule accounts for what the operation would do, whom it would affect, which permissions apply, and how it would be carried out.

Using conceptual model building within a design science research framework, the study examines four hypothetical scenarios and states nine design propositions. In one scenario, an agent reviewing its work finds a mislabeled ticket that its plan overlooked. Its grant already covers label changes in that project, so it can fix the error on its own. The same correction in a second project would require a grant that covers that project.

Authorization can also change over time. Restoring a snapshot may recover a plan to send a document after the one-time grant for that send has been consumed. Across these cases, host mediation lets the agent continue work within its delegation while holding any step that needs further authority.

Correspondence: papers@team.qeios.com — Qeios will forward to the authors

Research Approach and Scope

Research Question and Objective

How should a host organize the work of a persistent AI agent so it can carry out delegated tasks on its own? The design problem is to give the agent enough discretion to finish and correct its work, including when it draws on material retained from an earlier task. That material can remain available as permissions and circumstances change. The study takes the host architecture as its object of design and the **transition** as its unit of analysis. Each transition makes it possible to examine what the agent proposes and what the host needs to establish before allowing that step.

Research Design

Following Jaakkola[1], the study builds a **conceptual model** from literature selected for its bearing on decisions the host needs to make. Capabilities explain what a grant covers, while information-flow control and contextual integrity address what may be shared and with whom. Provenance helps trace information back to its source. Research on agents, prompt injection, memory, and recovery brings these decisions into the setting of persistent systems. Section 4 develops the connection between each body of work and the model.

Design science research (DSR) frames the contribution as design knowledge[2]. The architecture assigns responsibilities within the host, and a semiformal rule specifies when to admit a transition. Section 10 maps the design to the six core components of a design theory identified by Gregor and Jones[3].

Across four hypothetical scenarios, the analysis traces what the host retains, what the agent receives or proposes, and what the mediator checks. Changing the recipient, resource, or status of a grant shows how each condition can affect the decision. Nine propositions state requirements of the design, and counterexamples identify arrangements that violate those requirements or cases where the host needs more information to decide. The conclusions concern the conceptual design under the following assumptions.

Domain and Assumptions

In HAMU, **persistence** means that information or state survives the interaction or run that produced it and remains available for later use. Lin et al.[4] describe this property for memory entries that can be

retrieved after their original context ends. It also applies to a preference, a pending plan, or a record of a completed action. **Continuity** is the system's ability to use retained state to carry work forward, as when a task stays in storage until there is a reason to resume it.

HAMU applies to systems that retain state and can use tools or communicate with third parties. A host may span several components or services. To govern an effect, the host's designated controls must cover every route capable of producing it. The mediator must also be able to consult policies and records whose integrity is protected independently of the model's output.

Contribution

By representing proposed operations as transitions, HAMU connects decisions about access, retention, disclosure, and action to a common rule. Sending a document, for instance, may use up a grant. When the agent proposes another transmission, the execution record matters even if the plan stays in memory.

1. What HAMU Is

HAMU describes how to organize a working environment for a persistent AI agent. Its **host** runs or coordinates the agent, provides tools and information, and governs access along the routes it controls. Within that environment, a mediator checks requests against policies, grants, and records the host protects. HAMU calls this arrangement **host mediation**.

Under this arrangement, an agent can **see** what it needs for a task, **remember** material admitted for later use, **communicate** with authorized recipients, and **act** through tools. One assignment can cover several of these operations, each subject to its own conditions.

2. Origins and Development Since February 2026

In **February 2026**, I began exploring what an AI agent could do with direct access to an operating system. For the experiment, I gave it access to a laptop running Ubuntu without an intervening agent framework.

I asked it to activate the camera, take a photo, and send it through Telegram. It combined the available tools without a function built for that sequence. In another experiment, I had Clio, the primary agent, share a chat with Gigi, a subagent using a less capable model. When Gigi handed the conversation over to Clio, it continued in the same channel with a different model and context.

doi.org/10.32388/BUMY1R

Watching the agent combine tools showed me how much it could do with what was already available. The handoff raised a question about continuity. What would the host have to retain for another agent to pick up the work?

I thought of a character in *The Sims*. Even while nobody is interacting with the character, the environment keeps track of what has happened. An agent's environment could likewise retain a plan for days, even after the permission to carry it out expired. I began asking how a host could govern what an agent does with its access over time.

3. The Problem

3.1. A Calendar Question

Someone asks whether the operator can meet at two. After checking the calendar, the agent replies, "They can't make it then. They have to stop by the elementary school." In answering the scheduling question, it also reveals a detail about the operator's family life.

"Busy" would have answered the question. The host can give the agent that result while keeping the full entry in custody. Contextual integrity explains why information appropriate to the operator's calendar can be out of place in a reply to someone else[5].

3.2. Authority Collapse

Alongside instructions from the operator, an agent reads documents, tool results, memories, and messages from third parties. Each can inform its work. Before letting a proposal proceed, the host needs to know which records supply evidence and where authority for the operation comes from.

Greshake et al.[6] show how instructions embedded in external sources can reach a language-model application. Liu et al.[7] show how such material diverted responses in the applications they studied. In a persistent system, it may also enter memory or influence a later proposal.

Authority collapse occurs when the host lets material interpreted by the agent stand in for the authorization required for an operation. The agent might encounter a request in a document, a plan whose grant has expired, or a record explaining an earlier approval. Acting on any of that material requires a grant that covers the operation now.

3.3. From an Instruction to an Operation

System and user messages guide the model. Before a proposal can take effect, the mediator consults a source of authority whose validity can be established independently of the text the model receives. Bhattarai and Vu[8] argue that a learned distinction between instructions and data provides insufficient guarantees for deterministic authorization.

3.4. What a Tool Can Do With Its Permissions

An operator might assign ticket updates in one project even though the support tool has credentials for several. To decide whether a proposed update may proceed, the mediator checks both the project and the change against that assignment.

3.5. Errors Without Adversarial Intervention

Even during routine work, an agent can act on an outdated memory or choose a sensible correction that touches a resource outside the assignment. Authorization depends on the effect of the proposed operation in either case.

3.6. Governing the Agent Between Interactions

When a new event makes a stored task relevant again, the agent can consult its earlier plan and results. As work resumes, the host applies the policy then in force.

Taken together, a file, a notification, and a memory can reveal a fact that none reveals on its own. Where policy protects that fact, the host must consider how the items were obtained, why they were retained, and what the agent proposes to do with them together.

3.7. Work the Agent Can Carry Out

By delegating a set of operations, an operator can leave their order to the agent and authorize it to review and correct its work. When a step falls outside the assignment, the host can hold it and let the authorized work continue.

3.8. Actors, Threats, and Assumptions

Operators delegate permissions over resources they are entitled to govern. Acting within that environment, the agent plans the work and proposes operations for the mediator to check. Once a

proposal is admitted, the responsible component or service attempts the operation. Records in custody let the host make and reconstruct these decisions.

Documents, messages, and tool results can come from sources an adversary controls. A decision can also turn on an ordinary change, such as a grant expiring while a plan is waiting to run.

Complete mediation requires the designated controls to cover every route capable of producing the effect under review, including routes with separate credentials.

4. Foundations of the Model

4.1. What a Delegation Covers

Capability-based security explains how authority can be tied to a resource and a set of operations, and how a holder can pass on a narrower set of permissions[9][10]. For HAMU, this distinction matters when a tool has broader access than the grant covering a proposed operation.

4.2. What Information Can Move

Reading a source, presenting a limited view to the agent, and sharing the result with a third party move information across different boundaries. Information-flow control provides ways to govern how data moves or influences decisions across domains[11][12][13]. Contextual integrity adds norms about what may be shared, between which roles, and under what conditions[5][14].

4.3. What the Host Knows About Retained Material

The PROV family offers a vocabulary for describing where data came from and how it was derived[15]. CamFlow shows how to capture those relationships across a system[16]. Romanchuk and Bondar[17] describe *semantic laundering*, where passage through a trusted interface gives a weakly supported claim undue weight. In HAMU, provenance traces the material's history, evidence supports the conclusions drawn from it, and policy governs how it may be used.

4.4. Where to Check an Operation

Greshake et al.[6] document cases in which instructions from external sources reached a language-model application. Liu et al.[7] formalize the problem and evaluate five attacks and ten defenses with ten models

and seven tasks. Their evaluation found limits in prevention or detection, and some defenses reduced performance on the intended task.

Several approaches illustrate where a host can check a proposal before it takes effect. CaMeL separates control flow from data flow[18], while Odersky et al.[19] track capabilities in programs generated by agents. IsolateGPT isolates execution in language-model applications[20], and RLBox illustrates fine-grained isolation within a browser[21]. LlamaFirewall monitors agent steps during execution[22]. Kim, Guo, and Song[23] organize the broader literature on agent attacks and defenses.

4.5. What Changes Between Uses

Memory governance extends from admission and retrieval to sharing and forgetting[4]. In agents that share memory among parties with different permissions, GateMem evaluates utility, access control, and forgetting[24]. Sharma[25] studies defenses against memory poisoning across sessions. A proof of concept from Zheng et al.[26] shows how restoration can lead to repeated operations or reuse of consumed credentials.

Maloyan and Namiot[27] systematize prompt injection in coding assistants. In later work, they describe how external input can persist in an always-on agent and lead to a delayed action. They propose provenance gates and give a soundness argument under specified deployment conditions[28]. HAMU traces that passage from external material to memory and then to a proposed action, where admission and execution can have different conditions.

Foundation	**Question it helps HAMU answer**
Capabilities	What grant covers the operation?
Information flow and contextual integrity	What may move, to whom, and for what purpose?
Provenance	How was the material obtained or transformed?
Prompt injection and mediation	Where can the host check the request before it takes effect?
Memory and restoration	What may be retained, and under what conditions may it be used again?

5. How the Host Organizes Authority

5.1. The Thesis

An agent works out what to do, then sends its proposal through a route where the responsible components decide whether the operation may proceed. Their decision depends on the current policy, grant, and state of the resource.

5.2. Five Design Principles

1. **Check permission along the route.** For each proposed effect, the mediator consults the policy and the grant. One delegation can cover several steps, including review and correction.
2. **Give the agent a projection suited to the task.** A projection limits a view to the information its recipient needs and may receive. The agent may work with one view and send another to an external recipient.
3. **Check the grant at the time of use.** Grants specify which uses, resources, and effects they cover. Even after a single-use grant has been consumed, the plan may remain available.
4. **Identify the resource the rule applies to.** Different tools can identify the same resource in different ways. *Canonicalization* brings equivalent representations into a common form so the same rule applies. A change in meaning calls for a check on the new operation. When the mediator cannot establish what the request would affect, it can consider a narrower operation or refer the request for review.
5. **Keep track of what the evidence supports.** Observations, summaries, and hypotheses support different claims. By recording where each came from and how it was produced, the host gives the mediator evidence to assess for the proposed use.

5.3. Three Regions of Responsibility

In **Frontstage**, the agent works with the information and tools available to it. Requests pass through **Middle Stage**, which checks them and routes authorized operations to the components or services that carry them out. **Backstage** holds the protected materials those decisions rely on. The host can distribute these logical regions across components or services.

Region	Responsibility
Workspace (*Frontstage*)	Uses projections and eligible memories to plan work and propose operations.
Mediation (*Middle Stage*)	Checks the operation and route, releases authorized projections, and lets covered operations proceed.
Custody (*Backstage*)	Protects source data, credentials, policies, grants, and records of decisions and outcomes.

6. Cognition, Continuity, and Authority in a Transition

Editing a draft changes its contents. Sending it changes who can see it. HAMU calls a step that changes an object's use, recipient, or possible effect a **transition**, and the mediator decides whether the proposed transition may proceed.

Cognition lets the agent interpret information, plan, and propose a step. To carry work forward, it relies on **continuity**, the ability to use what the host retained. What it may do under current conditions is a question of **authority**.

doi.org/10.32388/BUMY1R

6.1. Constructs, Actors, and Conditions

Construct	Role in a transition
Object	What the operation acts on, such as information, a request, a plan, an artifact, or a reference.
Provenance	Where the object came from, how it changed, and what evidence supports that account.
Requesting actor	Whoever requests the operation, whether a person, entity, or component.
Recipient and receiving domain	Who receives the object or effect, and the domain in which the decision applies.
Purpose	Why the operation is requested.
Route	How a request passes through the components that apply the designated controls.
Policy and grant	Policy sets the rules under which a grant gives an actor permission. One delegation can supply grants for several steps.
Object and authorization state	Current conditions of the object and grant, including admission, restriction, revocation, consumption, and expiry.
Projection	A view limited to what the task and recipient require.
Record	An account of a decision, transformation, or observed outcome, including what it covers and what supports it.

Accurate records can contain private information, and material approved for storage can contain errors. Evidence tells the mediator what it can rely on. Policy governs how it may use that material.

6.2. Minimal Specification of a Transition

With permission to correct a draft, an agent can propose an edit or ask to send the draft to someone else. To distinguish those requests, the mediator identifies what each operation would do, what it would affect, and how it would be carried out. HAMU records the proposal in a tuple, τ.

$$\tau = (o, d, x, k, a, b, p, r)$$

doi.org/10.32388/BUMY1R

Here o and d are the source and destination domains, x is the object, and k is the kind of operation. The requesting actor is a. Depending on the policy, b identifies the recipient, the person or organization whose permissions govern the operation, or both. The remaining fields are purpose p and route r through the host's controls. The mediator classifies the request by what it would do. Summarizing a document and sending that summary to a third party includes a disclosure.

To decide whether the proposal may proceed, the mediator uses the current state σ_t. This includes the policy, the grants and their validity, the object's condition and provenance, and the route configuration. The underlying policies, grants, and records are kept in Backstage. When a policy depends on earlier operations, the mediator also consults the relevant history.

$$\mathit{Admit}(\tau, \sigma_t) = P(\tau, \sigma_t) \land A(\tau, \sigma_t) \land V(\tau, \sigma_t) \land R(\tau, \sigma_t) \land F(\tau, \sigma_t)$$

- P checks whether policy permits that kind of operation on the object in the specified domains.
- A checks the delegation's scope for the actor, operation, purpose, affected resources, and recipient where applicable.
- V checks that the grant remains in force and the object is eligible for the proposed use.
- R checks whether the operation follows the designated mediation route.
- F checks for the evidence policy requires, such as a memory's provenance or a record of prior approval.

Every applicable condition must hold. Permission to correct the draft satisfies A for the edit, but that same editing grant is insufficient for the send. When a required condition cannot be verified, the mediator can consider a narrower request or refer the decision to someone with authority to make it.

Once the request is admitted, the responsible component attempts the operation. In the equation below, σ_{t+1} is the state after the host updates it with what it can establish about the outcome. The result y captures what the operation returns, if anything.

$$(\sigma_{t+1}, y) = \mathit{Apply}(\tau, \sigma_t)$$

An operation can return a projection, a change to a resource, or a response from a service. Approval records show what the host allowed. Execution records show what it knows about the outcome. For example, a timeout can leave the status of a send uncertain. The agent may check what happened under a grant that covers a status inquiry.

Policy determines when a single-use grant is consumed. For the next request, the mediator uses the state in force at the time of the decision. If that state is still σ_{t+1}, the next decision is $Admit(\tau', \sigma_{t+1})$.

6.3. Operations and Permissions

Step	Operation	What the mediator checks
Protected source → view for the task	**Projection**	Which parts of the source the recipient needs for the task and may receive.
Tool result → durable custody	**Admission**	Whether provenance, integrity, and policy permit retention.
Custody → model context	**Retrieval and presentation**	Whether the memory may be used for the task and what part may be presented.
Context → inference provider	**Transfer to the provider**	What may be sent, to which service, and by which route.
Draft → external recipient	**Disclosure**	Whether the grant covers sharing the content with that recipient.
Proposed tool call → attempted operation	**Execution**	Whether a current grant covers the effect. The host records what it can establish about the outcome.
Snapshot → recovered state	**Restoration**	Which data or plans may be restored. Later actions are checked against current grants.

Private editing and disclosure can have different conditions even when they involve the same document. Revoking a grant can change which uses are allowed while leaving the document intact.

6.4. How a Decision Travels Through the System

When the agent requests a memory from Frontstage, the mediator in Middle Stage consults the record and restrictions in Backstage before releasing the permitted view. For a later proposal, the host can still consult the full provenance, version, or revocation record.

7. Memory: Retaining Evidence Without Inheriting Permission

7.1. What the Agent May Retrieve and Reuse

When an agent resumes a task, a memory search can help it find the earlier work. Even after access has been revoked, the index can still rank a memory first. The mediator checks whether the agent may retrieve it for the current task and which parts it may see.

The host makes separate decisions about admission, retrieval, and later use. Lin et al.[4] examine risks across writing, storage, retrieval, execution, propagation, forgetting, and rollback. GateMem studies the usefulness of shared memory alongside access control and user-requested deletion[24]. In HAMU, publishing a retrieved memory or sending it to an inference provider requires a separate check, even when an existing grant covers that use. Keeping the provider's response requires a new admission decision.

7.2. What the Host Keeps for Later Work

During a task, the agent uses temporary context (*hot*). The host can keep admitted memories available for retrieval (*warm*) and preserve source material in custody (*cold*). These roles can be served by different storage mechanisms.

A working summary can help an agent pick up a task later. Where policy permits retention, the host keeps it as a summary. When a decision depends on a verifiable observation, the mediator can consult the original source or its supporting record and release only the view the agent needs.

7.3. What a Record Can Establish

When a proposed operation relies on retained material, provenance records in Backstage help the mediator establish where it came from and under what conditions it was stored, consulted, shared, or restored. If access to a source is revoked, those records can also help locate results derived from it.

For a proposed retry, the host may need to know both what it allowed and what happened. Approval records can establish who received a grant, when, and what it covered. Execution records can establish whether the operation took effect or show that the outcome is still uncertain. The mediator uses those records to assess the proposal under the authority now in force.

8. When the Route or Operating Context Changes

8.1. Checking the Path a Request Takes

An agent may propose sending the same document through two different services. Under the rule in section 6.2, the mediator checks whether the route in the proposal passes through the designated controls. When another service can produce the same effect using its own credentials, complete mediation requires those controls to cover that route too.

8.2. What Changes When the Engine or Provider Changes

A different inference engine can prepare the same request. With the request, route, and relevant state held constant, the same authorization criteria apply.

Changing providers can affect where the context goes, even while the assignment stays the same. Before sending context to the new service, the host checks whether policy and the grant cover the recipient, the route, and the terms of the transfer.

8.3. Restoring State, Publishing, and Leaving a Test Environment

After a snapshot is restored, an agent may see the same plan it saw before a send. The send itself may already have taken effect outside the restored state. For a proposed retry, the host checks the current grant and what the execution record establishes about the earlier attempt (section 9.3).

When the agent proposes publishing a draft, the change is who can receive it. The host checks what can be disclosed to that audience, even when private preparation was already covered. When the agent proposes carrying a test observation into an operational environment, the host checks what can cross that boundary as data or evidence and which grant covers the transfer.

9. Four Application Scenarios

In each scenario, the agent has a way to carry the task forward. What the host needs to establish before that step can proceed depends on what the agent proposes and what has changed.

9.1. Sharing the Same Calendar Reply with Two Recipients

The agent already has the answer “Busy” and a grant to share it with the colleague who asked. It proposes sending the same reply through the same service to another colleague involved in scheduling. The second colleague falls outside the grant, and no other grant covers that disclosure.

Proposed disclosure	Mediator’s decision
“Busy” → colleague who asked	Admit the reply under the current grant, assuming the other conditions are met.
“Busy” → second colleague	Withhold the reply because the grant does not cover that recipient.

With the content, purpose, and route held constant, the change of recipient is enough to change the decision (P3).

9.2. An External Document Reappears in Another Session

While preparing a summary, the agent reads an external document that contains a request to send information outside the assignment. For any portion admitted to memory, custody preserves the source and the conditions of admission. When work resumes, the agent may receive the portion relevant to its task.

Step	Decision
Document → reading context	Present the document as external material and preserve its provenance.
Material read → durable memory	Check what can be retained and record the conditions in custody.
Memory → later context	Check eligibility for the current task before releasing a view.
Retrieved text → proposed message	Identify what the agent proposes to send and to whom.
Proposal → external transmission	Check whether a current grant covers disclosure to that recipient.

Admission records explain why the host kept the material. Transmission requires a grant from the

operator or another competent authority (P1, P2, and P5).

9.3. Restoring a Plan After Its Authorization Has Been Consumed

An operator authorizes one completed transmission of a document to one recipient. After the agent sends it, the host restores a snapshot taken before the send. The earlier plan and approval reappear in Frontstage, while the record in custody still shows that the transmission occurred.

From the execution record, the mediator can establish that the single-use grant was consumed. The approval explains why the first send was allowed, and another transmission needs a grant that covers it (P4 and P5). If a timeout leaves the outcome uncertain, the agent can request a status check before proposing a retry, provided the delegation covers the inquiry (section 10.4).

9.4. Correcting Work Within the Assignment

An operator assigns the agent to correct ticket labels in a support project and review the result. During review, it finds a mislabeled ticket the original plan omitted. Since the grant covers label changes in that project, the mediator lets the correction proceed through the authorized route.

In a second project, the agent finds the same error. Its tool can reach that project, but the grant covers only the first. With no applicable grant for the second, the host holds that correction while the agent finishes the authorized work.

Changing projects changes the resource affected, as P3 describes. For the first project, P9 requires a route that lets the agent make the correction under the existing grant.

10. Mutability and Design Propositions

Gregor and Jones[3] identify six core components of a design theory, alongside two additional components they call principles of implementation and an expository instantiation. The table shows how HAMU develops the six core components in its conceptual architecture.

Component	Where HAMU develops it
Purpose and scope	Work the host enables and effects it governs (Research Approach and Scope, section 3.8).
Constructs	Actors, objects, and conditions of a transition (section 6.1).
Form and function	Three regions of the host, five design principles, and the decision rule (sections 5–6).
Mutability	Changes in state that can alter a decision (section 10.1).
Propositions open to examination	Nine design propositions and cases that contradict them or show their limits (section 10.2).
Justificatory knowledge	Literature that supports the relationships in the design (sections 4 and 10.3).

10.1. When a Decision Changes

Even after a grant expires, a plan can remain available. An agent can make the same proposal, with the same τ as in section 6.2, and get a different decision because its authorization state has changed. Changing the recipient or route can also alter the decision, this time by changing the proposal itself.

Change	What may remain available	What the host checks for the proposed use
Expiry or consumption of a grant	Task, plan, and approval record.	Whether a current grant covers the use.
Revocation of access or consent	Historical records and derived material that policy permits the host to retain.	What may be retrieved, disclosed, or used to take an action.
New purpose or recipient	Object and provenance.	Whether policy permits the use in the new context.
Policy update	Record of earlier authorization.	Which conditions apply now.
Snapshot restoration	Content, references, and plans.	Which grants were consumed or revoked and which effects already occurred.
Change of provider or route	Task objective and proposed content.	Who receives the material, on what terms, and through which controls.
Move from testing to operation	Observations and drafts.	What may be transferred and which grant covers it.
Change in evidence for a source	Claims and summaries derived from the source.	Whether they meet the evidence requirements for the proposed use.

10.2. Design Propositions and Conditions for Examination

P1. Source of authorization. Before allowing an operation, the mediator identifies a grant that covers it. *Contradiction.* Putting an action in a plan is enough for the host to treat it as authorized. An external request can reach that point through the sequence in section 9.2.

P2. Custody and retrieval. Admitting material to durable memory and retrieving it for a task are separate decisions. Presenting it to the model and disclosing it each follow the conditions for that use. *Contradiction.* After access has been revoked, a memory still appears in search results, and the host treats that match as permission to retrieve it (section 7.1).

P3. Sensitivity to context. What may be done with the same content can change with the actor, recipient, purpose, scope, or route. *Contradiction.* The host treats permission to send an availability reply to one person as sufficient to send the same reply to another person outside the grant (section 9.1).

P4. Authority after restoration. When an operation comes from a restored plan, the host checks which grants are current and which effects already occurred. *Contradiction.* Restoring a snapshot taken before a send makes the single-use grant available again after the document was sent (section 9.3).

P5. What records establish. Approval records establish what the host allowed, and execution records establish what it knows about the outcome. For any later operation, the host uses the current grant and state. *Contradiction.* Finding the earlier approval is enough for the host to allow another send (section 9.3).

P6. Authorization along a route. Authorization covers the operation along the route where the required controls are applied. *Contradiction.* After approving a send on one route, the host treats that approval as sufficient for a separate channel with its own credentials (section 8.1).

P7. Independence from the inference engine. Equivalent proposals under the same conditions are judged by the same criteria, regardless of which engine generates them. Here equivalence covers the operation, object, actor, recipient, purpose, route, and authorization state. *Contradiction.* Two equivalent proposals receive different decisions solely because different models generated them. Changing providers can change the recipient or route (section 8.2).

P8. Cumulative disclosure. Individually admissible replies can reveal protected information when combined. To account for the combined disclosure, the host must consider the relevant history or another condition that covers the effect. *Counterexample.* Replies to individual availability requests suggest a recurring commitment that policy protects (section 10.4).

P9. Routes for legitimate work. For each class of task the design recognizes as legitimate, the host must provide routes that let the agent complete the necessary steps when their conditions are met. Grants already in force can cover review and correction. *Contradiction.* Even after the operator delegates label changes in a project and policy permits them, every available route requires new approval for each covered adjustment (section 9.4).

10.3. How the Model's Relationships Are Justified

Foundation	HAMU's inference	Proposition
Capabilities bind grants to resources and operations[9][10].	The model makes the request, and a separate source supplies the grant that authorizes it.	P1
Information-flow policies and contextual norms distinguish domains, actors, and conditions of transmission[11][5].	Permission to observe information and permission to disclose it require separate decisions. For the same information, a change of recipient can also change the disclosure decision.	P3
Risks vary across the memory lifecycle. Shared memory requires access control and forgetting[4][24].	The host checks admission and retrieval at the point each occurs.	P2
Restoration can lead to a repeated effect or reuse of a consumed grant[26].	The host checks a restored plan against the current state of the grant.	P4
Provenance documents derivation. A trusted interface can give a claim undue weight[15][17].	The host uses each record for what it establishes and checks the current grant for a proposed effect.	P5
CaMeL and tracked capabilities show ways to separate external data or generated programs from controls over their effects[18][19].	The decision depends on the route and the state its controls consult, regardless of which model prepared the proposal.	P6–P7

Combined disclosures motivate P8 (section 10.4). P9 follows from the design objective. To finish delegated work, the agent needs a route for the operations it is authorized to perform, as the ticket correction illustrates (section 9.4).

10.4. Tensions and Counterexamples

These cases examine how the model's relationships hold when conditions change. The analysis follows Jaakkola's[1] approach to conceptual model building and draws on Weber's[29] treatment of constructs,

relationships, states, and parsimony.

Minimization and utility. Counts by label can be enough to summarize a support queue. Correcting labels requires knowing which tickets to update. Under the second design principle, the view depends on the task and what its recipient may receive.

Correction and repeated effects. After a timeout, the host may be unsure whether a send succeeded. Under a grant for one transmission, retrying can duplicate the effect. The host records what is known, and an authorized status inquiry can help the mediator decide whether a retry may proceed. Policy also needs to cover cases where the outcome remains uncertain (section 9.3).

Revocation and custody. Revocation can end permission to use data while an audit rule requires retaining a historical record. When deletion is also required, policy has to specify what may remain, who may see it, and for how long. Each use has its own authorization conditions.

Canonicalization and changes in meaning. Two formats can describe the same request. Adding a field that changes the recipient changes the effect, so the mediator checks authorization for the new recipient.

Cumulative disclosures. Over several weeks, a recipient asks about a range of times, including two o'clock on several Tuesdays. Each reply gives only "free" or "busy." Read together, the replies reveal a pattern of Tuesday absences that suggests a recurring commitment. To protect that pattern, the host needs to consider what the recipient could infer by combining the next reply with earlier ones (P8).

Channels outside the boundary. Even when a host mediates its own tool calls, a connected application may produce the same effect through another channel. Complete mediation of the effect requires covering that channel too (section 3.8 and P6).

10.5. Questions the Model Leaves Open

How should a change of purpose affect work in progress? Which inferences should the host account for when it combines results? What evidence may it retain when a deletion obligation applies? The answers depend on the setting. HAMU identifies the actors, state, and operations a policy needs to address in each case.

11. Conclusions

HAMU describes how a host can give a persistent agent room to decide how to finish delegated work. It provides the tools and information the task requires and uses a mediator to check proposed operations

against protected policies, grants, and records.

Taking the transition as the unit of analysis makes it possible to examine one decision at a time. Keeping a document, presenting it to the agent, and sending it to someone else are three such decisions. The semiformal rule specifies the conditions the mediator checks for each. Past approvals and known outcomes can inform what happens next, alongside the policy and grants currently in force.

A single proposal may be only part of what the host needs to examine. Earlier disclosures can change what a new reply reveals, and an effect can be reachable through several channels. When policy governs cumulative disclosure, the decision must account for the relevant history. Complete mediation requires controls along every route to the effect.

Autonomy without sovereignty means giving the agent a route to finish the work it was assigned. That includes checking the result and making corrections within the delegation. When a proposed step needs further authority, the host can hold it while the rest of the authorized work continues.

Statements and Declarations

Data Availability

Materials for examining the model are available in the article. Sections 5–6 contain the design principles, constructs, and semiformal rule, and sections 9–10 present the hypothetical scenarios, propositions, and counterexamples. Literature informing the analysis appears in the References. This study did not generate or analyze an empirical dataset.

Use of Generative AI

To prepare the manuscript, the author used OpenAI's GPT-6 model through ChatGPT for help with drafting and organization, citation checks against sources, translation, and editing for clarity. The author directed this work and takes responsibility for the accuracy, integrity, originality, and final wording of the manuscript.

References

1. [a], [b] *Jaakkola E (2020). "Designing Conceptual Articles: Four Approaches." AMS Rev. **10**(1-2):18–26. doi:10.1007/s13162-020-00161-0.*

2. ^Hevner AR, March ST, Park J, Ram S (2004). "Design Science in Information Systems Research." MIS Q. **28** (1):75–105. https://aisel.aisnet.org/misq/vol28/iss1/6/.
3. a, bGregor S, Jones D (2007). "The Anatomy of a Design Theory." J Assoc Inf Syst. **8**(5):312–335. doi:10.17705/1jais.00129.
4. a, b, c, dLin Z, Hao X, Fu R, Cui S, Chen K, Li C, Li Z, Xiong F (2026). "A Survey on Long-Term Memory Security in LLM Agents: Attacks, Defenses, and Governance Across the Memory Lifecycle." Proceedings of EMNLP 2026. 52221–52235. https://aclanthology.org/2026.emnlp-main.2296/.
5. a, b, cNissenbaum H (2004). "Privacy as Contextual Integrity." Wash Law Rev. **79**:119–158. https://digitalcommons.law.uw.edu/wlr/vol79/iss1/10/.
6. a, bGreshake K, Abdelnabi S, Mishra S, Endres C, Holz T, Fritz M (2023). "Not What You've Signed Up For: Compromising Real-World LLM-Integrated Applications with Indirect Prompt Injection." Proceedings of the 16th ACM Workshop on Artificial Intelligence and Security. 79–90. doi:10.1145/3605764.3623985.
7. a, bLiu Y, Jia Y, Geng R, Jia J, Gong NZ (2024). "Formalizing and Benchmarking Prompt Injection Attacks and Defenses." 33rd USENIX Security Symposium. 1831–1847. https://www.usenix.org/conference/usenixsecurity24/presentation/liu-yupei.
8. ^Bhattarai M, Vu M (2026). "Trustworthy Agentic AI Requires Deterministic Architectural Boundaries." arXiv. https://arxiv.org/abs/2602.09947.
9. a, bMiller MS (2006). Robust Composition: Towards a Unified Approach to Access Control and Concurrency Control. Johns Hopkins University. https://www.erights.org/talks/thesis/index.html.
10. a, bWatson RNM, Anderson J, Laurie B, Kennaway K (2010). "Capsicum: Practical Capabilities for UNIX." 19th USENIX Security Symposium. https://www.usenix.org/conference/usenixsecurity10/capsicum-practical-capabilities-unix.
11. a, bGoguen JA, Meseguer J (1982). "Security Policies and Security Models." IEEE Symposium on Security and Privacy. 11–20. doi:10.1109/SP.1982.10014.
12. ^Myers AC, Liskov B (2000). "Protecting Privacy Using the Decentralized Label Model." ACM Trans Softw Eng Methodol. **9**(4):410–442. doi:10.1145/363516.363526.
13. ^Sabelfeld A, Myers AC (2003). "Language-Based Information-Flow Security." IEEE J Select Areas Commun. **21**(1):5–19. doi:10.1109/JSAC.2002.806121.
14. ^Nissenbaum H (2010). Privacy in Context: Technology, Policy, and the Integrity of Social Life. Stanford: Stanford Law Books / Stanford University Press.

15. ^a ^bGroth P, Moreau L (2013). "PROV-Overview: An Overview of the PROV Family of Documents." W3C Working Group Note. https://www.w3.org/TR/prov-overview/.

16. ^Pasquier T, Han X, Goldstein M, Moyer T, Eyers D, Seltzer M, Bacon J (2017). "Practical Whole-System Provenance Capture." Proceedings of the ACM Symposium on Cloud Computing. 405–418. doi:10.1145/3127479.3129249.

17. ^a ^bRomanchuk O, Bondar R (2026). "Semantic Laundering in AI Agent Architectures: Why Tool Boundaries Do Not Confer Epistemic Warrant." arXiv. https://arxiv.org/abs/2601.08333.

18. ^a ^bDebenedetti E, Shumailov I, Fan T, Hayes J, Carlini N, Fabian D, Kern C, Shi C, Terzis A, Tramèr F (2026). "Defeating Prompt Injections by Design." IEEE Conference on Secure and Trustworthy Machine Learning (SaTML 2026). 587–618. doi:10.1109/SaTML68715.2026.00040.

19. ^a ^bOdersky M, Zhao Y, Xu Y, Bračevac O, Pham CN (2026). "Securing Agents With Tracked Capabilities." Proceedings of the ACM Conference on AI and Agentic Systems (CAIS 2026). 812–838. doi:10.1145/3786335.3813127.

20. ^Wu Y, Roesner F, Kohno T, Zhang N, Iqbal U (2025). "IsolateGPT: An Execution Isolation Architecture for LLM-Based Agentic Systems." Network and Distributed System Security (NDSS) Symposium. https://www.ndss-symposium.org/ndss-paper/isolategpt-an-execution-isolation-architecture-for-llm-based-agentic-systems/.

21. ^Narayan S, Disselkoen C, Garfinkel T, Froyd N, Rahm E, Lerner S, Shacham H, Stefan D (2020). "Retrofitting Fine Grain Isolation in the Firefox Renderer." 29th USENIX Security Symposium. 699–716. https://www.usenix.org/conference/usenixsecurity20/presentation/narayan.

22. ^Chennabasappa S, Nikolaidis C, Song D, Molnar D, Ding S, Wan S, Whitman S, Deason L, Doucette N, Montilla A, Gampa A, de Paola B, Gabi D, Crnkovich J, Testud JC, He K, Chaturvedi R, Zhou W, Saxe J (2025). "LlamaFirewall: An Open Source Guardrail System for Building Secure AI Agents." arXiv. https://arxiv.org/abs/2505.03574.

23. ^Kim J, Guo W, Song D (2026). "SoK: Attack and Defense Landscape of Agentic AI Systems." 35th USENIX Security Symposium. 6047–6066. https://www.usenix.org/conference/usenixsecurity26/presentation/kim-juhee-agentic.

24. ^a ^b ^cRen Z, Yang Y, Chen Y, Zhao Z, Fu B, Shu Z, Zhang B, Xu Y, Guo D, Yan S (2026). "GateMem: Benchmarking Memory Governance in Multi-Principal Shared-Memory Agents." arXiv. https://arxiv.org/abs/2606.18829.

25. ^–Sharma T (2026). "SMSR: Certified Defence Against Runtime Memory Poisoning in Persistent LLM Agent Systems." arXiv. https://arxiv.org/abs/2606.12703.

26. a, b–Zheng Y, Yang Y, Zhang W, Quinn A (2026). "ACRFence: Preventing Semantic Rollback Attacks in Agent Checkpoint-Restore." arXiv. https://arxiv.org/abs/2603.20625.

27. ^–Maloyan N, Namiot D (2026). "Prompt Injection Attacks on Agentic Coding Assistants: A Systematic Analysis of Vulnerabilities in Skills, Tools, and Protocol Ecosystems." arXiv. https://arxiv.org/abs/2601.17548.

28. ^–Maloyan N, Namiot D (2026). "Sleeper Channels and Provenance Gates: Persistent Prompt Injection in Always-on Autonomous AI Agents." arXiv. https://arxiv.org/abs/2605.13471.

29. ^–Weber R (2012). "Evaluating and Developing Theories in the Information Systems Discipline." J Assoc Inf Syst. **13**(1):1–30. doi:10.17705/1jais.00284.

Declarations

Funding: No specific funding was received for this work.

Potential competing interests: No potential competing interests to declare.