

Governed Memory for Physical AI

Evidence Roles, Integrity and Revocation from Experience to Action

Dr. Carsten Stöcker
Spherity GmbH

Prepared: 20 October 2026



Except where otherwise noted, this work is licensed under the Creative Commons Attribution 4.0 International License (CC BY 4.0). © 2026 Dr. Carsten Stöcker, Spherity GmbH.

Abstract

A robot can use a past experience to change its next action without retraining its artificial intelligence (AI) model. This makes memory part of the control system. A grasp that worked with one gripper may fail with another; a failed grasp can still provide useful evidence about slipping. We propose a Memory Supply Chain Security framework that governs how experience is captured, assessed, shared and used. It separates recorded events, claims about them, evidence for those claims, permission to reuse them and permission to act. A Memory Bill of Materials traces sensor data, calibration, processing and model dependencies. A Robot Bill of Materials identifies the receiving hardware, software, AI models and skills. Assurance applies to their assessed combination with memory and operating conditions, rather than to a memory or skill in isolation. Every operational reuse requires current checks. Revalidation may reuse existing evidence or require new experiments, depending on whether the proposed use remains covered. Enterprise identities, protected machine wallets and scoped delegation establish who may issue or consume evidence. A verifiable evidence graph records the selected support and carries withdrawals through active context to operation admission. Failed experience can remain useful, while copied records do not become independent confirmation. Conditional proofs and a finite protocol study support the admission and withdrawal logic; robot-level reliability remains to be evaluated. We assume a stable robot, skill and model configuration during each assessment interval, while allowing memory-conditioned behaviour to change. Provenance and scoped authority establish neither factual truth nor physical safety.

Keywords: physical AI; robot memory; memory supply chain; verifiable evidence graphs; delegated authority; runtime assurance; revocation

Contents

1	Introduction	4
2	Related work and contribution boundary	5
2.1	Experience retrieval and memory-based adaptation	5
2.2	Poisoning and lifecycle controls	5
2.3	Provenance, assurance and revocation	6
2.4	Software supply chain mechanisms and the memory extension	6
3	Evidence semantics for robot experience	7
3.1	Episodes, interpretations and permitted roles	7
3.2	Observations, analyses and copies	7
3.3	Typed dependencies and selected support	9
3.4	Memory bills of materials	10
3.5	Memory within the robot software and skill stack	10
3.6	Robot bills of materials and assessed combinations	11
4	Architecture and memory lifecycle	12
4.1	System boundary and adversary	12
4.2	Protected memory verification	14
4.3	From observed episode to receiver admission	16
4.4	Withdrawal, receipts and recovery	17
4.5	Proportionate revalidation and configuration changes	17
5	Formal model and conditional guarantees	19
5.1	Objects, support witnesses and protected state	19
5.2	Assumptions and transitions	21
5.3	Representation and operation admission	22
5.4	Replication and evidence dependence	25
5.5	Scoped withdrawal and alternative support	26
5.6	Timed exposure and counterexamples	27
6	Executable finite protocol study	27
6.1	Model, method and reproducibility	27
6.2	Results and enforcement mutations	28
6.3	Interpretation and limits	29

7	Prospective evaluation of robot behaviour	29
7.1	Study tracks and controlled comparisons	29
7.2	Statistical units and response measurements	31
7.3	Supply chain and authority evaluation	33
8	Implementation and discussion	33
8.1	A feasible implementation path	33
8.2	What the results establish	35
8.3	Availability, privacy and governance limits	35
9	Conclusion	35
9.1	Guideline recommendation	36
A	Physical-assurance interface	37
B	Artifact record and declarations	37
C	Memory Supply Chain, Enterprise Identities, and Verifiable Attestation Profiles	38
C.1	Scope and relation to software supply chain security	38
C.2	Supply chain threat model	38
C.3	Verifiable credential data profiles	40
C.4	Delegation and verification exchange	42
C.5	Status distribution and bounded stale admission	44
C.6	Evaluation and assurance boundary	45
	References	46

1 Introduction

A robot’s behaviour depends on more than its trained artificial intelligence (AI) model. Sensors supply observations; software and control services interpret them; skills provide ways to act; and memories supply experience from earlier attempts. A retrieved episode can alter a plan or action even when the model is unchanged. Here, **policy weights** are the learned numerical parameters of a model that selects actions or action proposals. They are distinct from the operator’s policy rules about permitted use. Keeping these parameters fixed does not keep behaviour fixed: new observations, instructions, memory and computational state can change the output.

Memory therefore belongs in the robot’s assurance boundary. It may contain a video, a force trace, an action sequence, an assessed failure or a derived representation supplied to the acting model. It can inform selection of a skill, its parameters or the next step within it. A skill may be a conventional controller, a learned policy or a composition of smaller skills. The operating system, robotics middleware, drivers, control timing and observation/action interfaces connect those decisions to the physical robot. Section 3 defines these relationships and the configuration records needed to assess them.

Consider a grasping skill that uses an earlier successful episode as guidance. Its evidence concerns a particular gripper, camera calibration, control interface, AI model and range of objects. Replacing the gripper or changing action scaling can undermine that evidence even if the memory file is unchanged. Conversely, a new episode from the same assessed configuration need not require repeating the entire robot validation programme if an accepted memory-update contract already covers its content, representation and use. This is a conditional reuse decision, not a guarantee that every episode will improve performance. A skill obtained from a marketplace has the same limitation: a validation report supports its stated configuration and operating envelope, not every receiving robot.

Robota-Embed provides a concrete research setting. It retrieves physical experience using vision, language and action, and supplies retrieved material through policy-specific context interfaces. Its Embed2Evolve study changes memory and context settings around fixed policy and embedding weights, then freezes the selected configuration for testing [1]. We study the governance obligations on this memory path. The paper does not assume that a memory compatible with one policy interface will have the same effect through another.

The grasp example also shows why evidence requires more than a success label. An unsuccessful attempt may contain a useful force trace, a successful approach segment and an incorrect completion label. Deleting the whole episode would lose evidence; approving it as a complete demonstration would exceed that evidence. Copying it to twenty robots improves availability without creating twenty observations. A later finding that its calibration was defective may require reassessing the force interpretation even though the recorded bytes remain unchanged. Database removal alone cannot withdraw influence already present in a model cache or plan.

The research question is: **how can a robot retain and share useful experience while enforcing current, configuration-specific conditions on its influence over physical execution?** Our answer links memory governance to the supply chains of the robot, its models and its skills. A Memory Bill of Materials (MBOM) describes the origin and processing of an episode. A proposed Robot Bill of Materials (Robot BOM) describes the receiving configuration and its component relationships. A verifiable evidence graph connects both inventories to assessed claims, scoped authority and the exact premises selected for an admission. Cryptographic identity and custody support attribution; applicability and physical assurance require separate evidence.

We make four contributions. First, we separate task outcome, measurement quality, claim validity and reuse role, retaining useful failures while counting recognised copies only once when

assessing evidential support. Second, we define a Memory Supply Chain Security profile with explicit configuration coverage, enterprise identity and attenuating delegation. Third, we bind selected support to consumed representations and operation permits, so changed or withdrawn premises restrict affected reliance without deleting unrelated observations. Fourth, we provide conditional proofs, an executable finite protocol study and a prospective robot evaluation design. The finite study tests a bounded admission and withdrawal fragment; it does not validate robot configurations, credential interoperability or physical performance.

The scope is external memory within an approved configuration. During each assessment interval, the skill implementations, AI artifacts, robot configuration and underlying response within declared tolerances are assumed stable, with no unmodelled behavioural or other drift. Intended changes caused by admitted memory remain in scope. This assumption does not imply repeatable outcomes or remove environmental variation within the assessed envelope. Online weight learning, unbounded hardware drift and automatic transfer of assurance after configuration changes are excluded. A relevant configuration change triggers the assessment process in Section 4.5. Sections 3 and 4 define configuration and revalidation; Section 5 states the formal assumptions and gates. The enterprise-to-edge implementation and behavioural evaluation remain proposed.

2 Related work and contribution boundary

2.1 Experience retrieval and memory-based adaptation

Robota-Embed’s Embed2Evolve workflow retains ordinary failed attempts and limits retrieval to at most one anchor per source episode. Its appendix distinguishes episode outcome from the correctness of every local action [1]. The prospective comparison in Section 7 retains both baseline features. Our extension concerns the permission and evidence status of a particular claim or segment, including its continued influence after retrieval. It does not assume that failure retention or episode-level diversity is absent from the baseline.

Just-In-Time Memory (JIT-Memory) uses a Test-Verify-Write loop to refine mitigation skills from successful and failed execution cases [2]. Its risk-state graph supports both hazard triggering and trace verification. We infer that an error shared by those uses can compromise both checks; this is a limitation of the validation architecture, not a reported attack result. We accordingly treat local validation as an attributable assessment with declared dependencies. Repeating a judgment through the same observation path does not establish independent confirmation.

2.2 Poisoning and lifecycle controls

AgentPoison demonstrates attacks through malicious material inserted into memory or knowledge stores under specified injection and embedding-access assumptions [3]. Memory Injection Attack (MINJA) shows that query/output interaction can induce a memory-writing agent to persist adversarial material without direct attacker access to the memory bank [4]. MemoryGraft studies poisoned procedural experience retained through an ingestion workflow and later retrieved by an agent [5]. These studies motivate separate treatment of unauthorised byte changes and authorised writes containing unjustified content. Neither category is resolved by establishing the identity of the writer alone. Their experimental settings do not establish physical robot compromise or fleet-wide propagation.

Memory-Aware Propagation and Link Enforcement Guard (MAPLE-Guard) regulates memory writes, retrieval, promotion and cross-agent reuse using shared risk and state metadata [6].

Lifecycle gating is therefore prior work. Its memory-manager metadata are assumed intact, and its reported full-feedback configuration uses evaluator-assisted outcome information. Our proposed evaluation matches the observable metadata available to each deployable control and identifies oracle feedback separately. The additional question here is how an admitted representation, retained state and a pending physical operation remain bound to current evidence after memory has been read.

2.3 Provenance, assurance and revocation

Provenance standards distinguish entities, activities, agents and derivation [7]. Verifiable credentials provide a representation for attributable claims [8]. Such structures help express who asserted what and how an artifact was produced. They do not by themselves establish physical truth, evidence independence or permission for a receiving robot to execute an action. We use provenance as an input to claim-specific assessment and retain uncertainty about unobserved common causes.

Spherity’s evidence-graph research connects sources, transformations, claims and current status [9]. Physical AI Compliance Evidence (PACE) applies evidence graphs to configuration-specific physical-AI assurance [10]. Its separation of evidence coordination from local safety functions informs our boundary between memory governance and physical control. The cascading-trust and runtime-revocation studies address propagation, current authority, dependency-aware containment and receiver-side enforcement in agent networks [11, 12]. The present paper specialises these ideas to episodic observations, their typed uses, mutable robot context and the last reversible physical-command boundary. It does not claim that dependency-based revocation is new. Spherity’s enterprise-wallet control-plane proposal supplies a further antecedent for organisational identity, bounded mandates and action-specific authorisation [13]. We specialise that proposed authority separation to the roles and selected dependencies of robot memory, rather than treating it as independent validation of the present protocol.

Runtime assurance provides a complementary control boundary. Neural Simplex switches between an advanced controller and a baseline controller under stated recoverability assumptions [14]. Spherity’s risk-bounded runtime-assurance formulation separates a current mandate from assurance for the proposed action [15]. Our protocol requires both obligations at dispatch. Memory eligibility cannot replace them, and neither memory similarity nor episode success supplies a numerical bound on harm.

The gap addressed by this paper is consequently specific: maintaining a current, explicit evidence basis from a reusable episode through retained computational state to a concrete operation, while preserving useful failures and preventing recognised replication from increasing empirical support. Establishing that this composition is useful in practice requires the behavioural evaluation specified in Section 7.

2.4 Software supply chain mechanisms and the memory extension

Supply-chain Levels for Software Artifacts (SLSA) describes source and build integrity requirements and provenance [16]. In-toto specifies authorised workflow steps, functionaries and artifact rules, including thresholds [17]. Sigstore supports verification of artifact signatures and expected signing identities [18]. CycloneDX and System Package Data Exchange (SPDX) can describe software, AI and machine-learning inventories with relationships and provenance [19, 20]. These are relevant foundations, not complete memory-governance protocols. Their attestations do not establish physical measurement truth, current task applicability or the safety of a command conditioned on the artifact.

The extension proposed here is a receiver-enforced profile connecting these mechanisms to evidence roles, selected authority paths, retained robot state and scoped withdrawal at effect admission. It does not claim novelty for signing, dependency inventories, approval thresholds or delegation attenuation in isolation. Appendix C maps the reused mechanisms to their limited roles and states the additional semantics needed for interoperable verification. In particular, an MBOM is a proposed memory-specific graph profile, and an enterprise decentralised identifier (DID) requires a separately recognised organisational and authority binding [21].

3 Evidence semantics for robot experience

3.1 Episodes, interpretations and permitted roles

An episode records what the capture path observed and issued. A claim states something about that record or the underlying event. An assessment evaluates a claim using a method and evidence. A reuse permission concerns a particular receiver, task and role. An execution permission concerns one concrete operation. These objects have different issuers, lifetimes and failure conditions. Collapsing them into a single trusted-memory flag hides distinctions that become essential during correction and withdrawal.

Episode metadata separates task completion, monitored constraint outcomes, intervention history and measurement quality. Each field admits unknown or disputed values. A successful task can include a constraint violation or human intervention. A failed task can contain an accurate observation or a successful segment. A blocked proposal is evidence that a proposal occurred and was rejected; it is not direct observation of the physical harm the proposal might have caused.

The running grasp example illustrates claim-specific handling (Figure 1). A completion classifier may be wrong while a separately assessed sensor trace remains useful. A successful approach segment may qualify as a demonstration for that segment under its own conditions. The failed grasp can support a counterexample or regression scenario. These uses require different views of the same episode. A counterexample view can expose assessed conditions and outcomes without exposing an unapproved action sequence as a positive instruction. Whether a neural policy respects that distinction remains an empirical question.

Failure is thus a reason to inspect and classify evidence, not a sufficient reason to delete or promote it. A valid counterexample within the stated domain can refute a universal claim. One failure does not establish its frequency, prove its cause or justify a numerical risk estimate. Repeated exposure to the same failed episode does not change those limitations. Audit retention is also distinct from policy exposure: incident material may contain adversarial instructions and should remain subject to separate access controls.

3.2 Observations, analyses and copies

An evidence bundle records observation origins and substantive assessment activities separately. A copied trace adds no physical observation. A new analysis of the trace may add a useful interpretation, uncertainty estimate or applicability assessment. A new physical trial adds an observation but may share a sensor defect, calibration process or evaluator with earlier trials. A new signature may confer an authorised approval while adding neither an observation nor an analysis. The architecture retains these distinctions rather than converting all additions into a larger count of confirmations.

Recognised pure replicas are transport or encoding variants of the same logical evidence object without new observational or analytical content. They are canonicalised before support calcu-

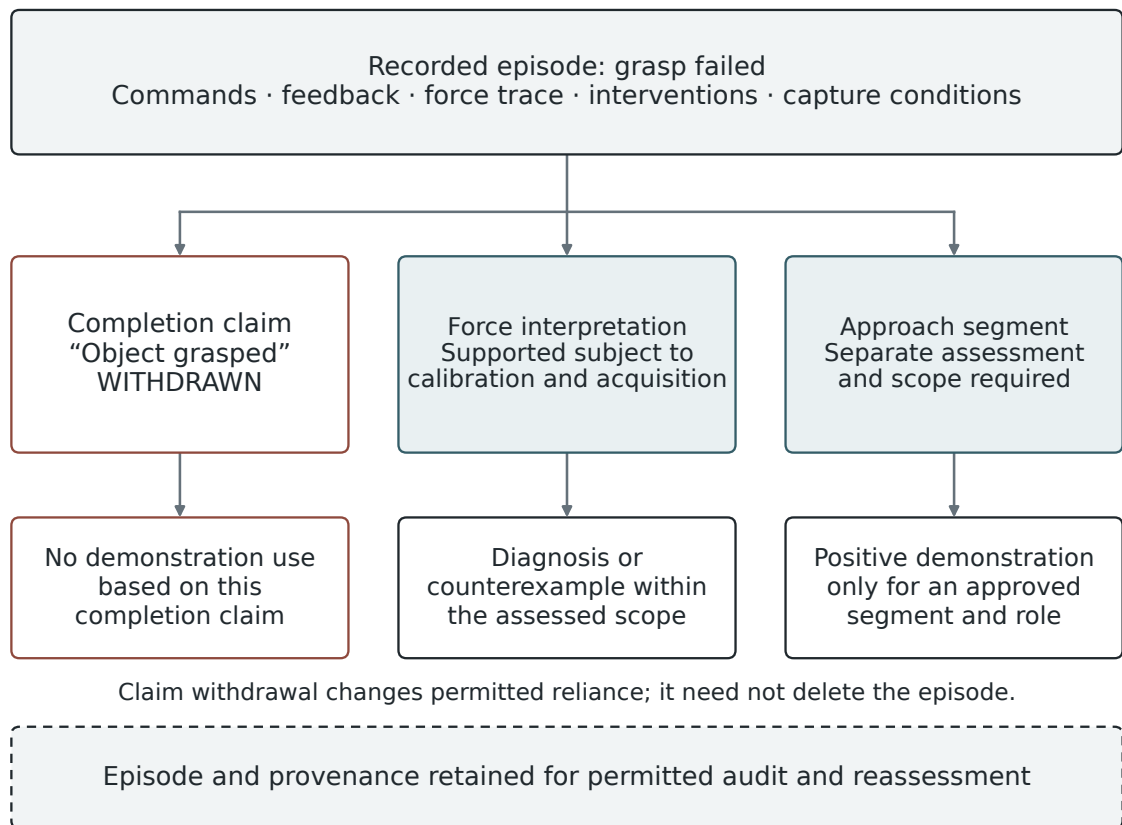


Figure 1: One episode can support different claims and reuse roles. Correcting the completion label does not erase the capture. A calibration withdrawal can separately affect the force interpretation. These are illustrative assessment outcomes, not experimental data.

lation and before allocation of retrieval context. This second requirement matters: a system can report unchanged confidence while allowing copied records to occupy most of the policy’s context. Origin quotas and canonical ordering prevent that specific amplification for faithfully identified replicas.

Crops, summaries and transformed views are not automatically pure replicas. They may remove qualifications, change the segment under discussion or add an analytical result. They retain their source lineage, but their semantic effect requires separate assessment. Conversely, distinct identifiers do not prove distinct origins. Cryptographic hashes detect equality or changes in committed bytes; they do not identify every paraphrase, forged lineage claim or hidden common cause.

Support is claim-specific. A camera calibration may be relevant to geometric clearance but irrelevant to an independently logged command identifier. The dependency graph records known common causes and unresolved dependencies. Apparently disjoint provenance is insufficient to assert statistical independence. A conservative assessment can restrict reuse when the available support cannot meet a requirement; it should not invent independence to obtain an approval.

3.3 Typed dependencies and selected support

We distinguish three edge types. A materialisation edge records an input used to construct a representation. An assurance-reliance edge records a premise selected to justify a particular claim or permission. A causal-history edge records an event that preceded or influenced another event without necessarily supporting its interpretation. This distinction prevents a correction from erasing every later observation merely because the events occurred in sequence. Identity bindings, delegation credentials and status assertions are explicit graph objects. Whenever current authority is required, the selected authority path is an assurance dependency of the permission, rather than an unrecorded property of its signer.

Each attributable assertion binds the issuer identifier, verification method, subject identifiers or content digests, predicate type, applicable interval and supporting evidence references. An enterprise decentralised identifier anchors the issuer’s verification relationships; it does not itself establish the issuer’s legal identity, competence or authority [21]. The receiver additionally requires a policy-recognised enterprise binding and the scoped credentials or trust-registry entries that authorise the assertion. Historical verification records the DID-document version or equivalent key-history evidence used at issuance. Current reliance separately checks applicable key-compromise, issuer-authority and credential-status restrictions. Key rotation must not silently substitute a new key for the key that authenticated an archived statement.

Evidence requirements may contain alternatives. A receiver therefore records a selected support witness: the actual observations, assessments and branches used in its decision, including the applicable authority dependencies. If either of two assessment methods satisfies a requirement, the witness identifies the selected method and any required independence assessment. The existence of an unused alternative cannot silently repair an old permit after its selected premise is withdrawn. A fresh decision may adopt a valid alternative and issue a new permit.

Witnesses resolve to identified observations, substantive assessment activities and policy-recognised authority roots through finite, acyclic derivations. Repeated fleet endorsements cannot form a self-supporting cycle. Signatures authenticate assertions about graph edges; they do not prove that a declared input was actually used, that every dependency was declared, or that different assessors are independent. The adequacy of the leaf evidence, methods and operating domain remains an assessment obligation. A verifiable evidence graph is therefore a graph whose identified assertions and selected verification obligations can be checked, not a graph whose every statement is thereby true.

3.4 Memory bills of materials

A **Memory Bill of Materials (MBOM)** is a versioned, package-rooted instance of the evidence graph G_t . For package p , write $M_p = (V_p, E_p, \mu_p)$, where $V_p \subseteq V(G_t)$, $E_p \subseteq E(G_t)$ and μ_p is an authenticated manifest specifying the package root, graph encoding, schema versions and node commitments. MBOM denotes the memory-specific application profile proposed here, rather than an established interoperability standard. It must be distinguished by its profile identifier from other uses of the acronym, including CycloneDX manufacturing bills of materials [19]. Its purpose is to expose the supply chain of the memory representation and the premises on which a receiver might rely.

The graph ties raw sensor traces and commanded actions to capture configuration, sensor identities and calibration assertions; preprocessing, segmentation and labelling to their inputs, parameters, software and execution environments; and embeddings, summaries and assessments to the exact encoder or assessor weights, adapters and configurations used. The policy weights and runtime configuration are recorded as compatibility and consumption dependencies where relevant. Digests identify exact artifacts rather than mutable model names. Referenced software bills of materials (SBOMs) or AI and machine-learning bills of materials describe software and model components; they do not replace episode-specific lineage, role assessments or authority paths [19, 20].

The MBOM distinguishes the producer’s declared graph from the receiver’s verified projection and selected witness. Membership proofs can establish inclusion in an authenticated manifest, but cannot establish that an omitted dependency never existed. Raw data may remain in an access-controlled store, provided the receiver can obtain the evidence or an explicitly accepted assessment required by its policy. A digest alone cannot discharge a requirement to inspect the trace. Distribution, reassessment and status changes append identified objects without changing previous commitments. A new package or derived view obtains a new root; a current status overlay determines which dependencies remain usable.

For a proposed use, the receiver computes a conservative closure over materialisation and selected assurance edges. This closure includes the authority, calibration and configuration premises actually selected. A package signature neither selects that witness nor authorises its use. Two packages containing the same trace retain one observation origin; two signatures from separate enterprise wallets do not create independent physical confirmation.

3.5 Memory within the robot software and skill stack

The **acting policy** maps observations, task inputs and retained context to action proposals, directly or through skill calls. In a learned policy, weights are numerical parameters set by training. They may reside in a shared AI model, a skill-specific model or both. Architecture, adapters, quantisation, preprocessing, observation normalisation and inference settings also affect interpretation; a weight digest alone does not identify the executable behaviour. External episodic memory is distinct from these weights. Its retrieved video, trajectory, tokens or summary can condition the policy without updating them. Selecting a runtime parameter, including a scheduled gain, can remain within the assessed baseline when the skill contract explicitly covers that parameter and range. Changing the controller implementation, fixed gains, skill program, model adapter or permitted parameter envelope creates a configuration change.

The **robot software platform** comprises the host operating system, firmware, drivers, inference runtime and robotics middleware. Robot Operating System 2 (ROS 2) exposes robotics communication interfaces through a middleware abstraction [22]. A ROS distribution identifier alone does not identify the host operating system, device drivers, firmware or complete con-

trol stack. **Base skills** are reusable capabilities such as joint motion, pose tracking, gripper closure or navigation. **Task skills** compose or parameterise these capabilities, for example to pick, inspect and place an object. This is a functional decomposition, not a required vendor taxonomy. A skill contract identifies inputs, units, coordinate frames, preconditions, supported parameters, completion conditions, timing assumptions and failure/recovery behaviour. Learned and conventional implementations both need such interfaces. A base skill is not automatically an assessed safety controller.

A skill marketplace can distribute an identified implementation with signed assessment reports, dependencies and supported configurations. In this proposed use, “validated” means that a named assessor evaluated a stated claim using an identified method and operating envelope. It does not imply universal portability, independence or certification. The receiver verifies provider and assessor authority, artifact versions, dependency compatibility and coverage of the intended skill composition. Local skill installation is governed separately from admission of memories used by that skill. Evidence for each component alone does not establish evidence for their interaction.

Operational memory use includes selecting a skill, choosing its parameters, supplying a demonstration, informing a recovery proposal, or populating retained context that can affect later decisions. The graph records the role and actual representation at this interface. An isolated audit read does not invoke the acting-policy gate, but still follows data-access rules. Loading an episode into an acting model’s context is operational exposure even if the user describes it as “for information”. A counterexample can influence behaviour and is not exempt from applicability checks.

3.6 Robot bills of materials and assessed combinations

We propose a **Robot Bill of Materials (Robot BOM)** as a versioned, instance-bound configuration graph within G_t . It links an identified robot to its manufactured parts, installed artifacts, interfaces, calibration evidence and approved operating constraints. It is a decomposition of the as-operated system, not just a mechanical teardown or parts list. A design-level robot model specifies structure and intended capabilities; an instance record binds that design to the actual serialised device, component revisions, tooling, maintenance state and applicable calibration. A robot-description file using the Unified Robot Description Format (URDF) can contribute structural information [23], but is not the complete inventory or an assurance case.

Write the baseline configuration as $\beta_j = (R_j, O_j, B_j, S_j, A_j, \rho_j)$. Here R_j identifies the physical robot model and instance, including kinematics, relevant dynamics, joints, actuators, sensors, tools and calibration references; O_j identifies the software platform and timing/interface settings; B_j identifies base controllers and skills, including fixed gains or gain-scheduling rules, permitted runtime parameter ranges and limits; S_j identifies task skills and their composition; A_j identifies the AI architectures, weights, adapters and inference configurations; and ρ_j identifies the memory encoder, retrieval, representation and context-management rules. Each entry resolves to versioned graph objects rather than a mutable product name. Components shared across entries are referenced once, with their relationships preserved. Hardware quantities and health conditions require measurement evidence and tolerances, not a hash of the physical object.

The Robot BOM references software and AI inventories and identifies units, coordinate frames, observation/action layouts, normalisation, command scaling and execution rates. It includes relevant mounting, payload limits, compute devices, safety devices and the independent assurance controller within the hardware and software/control entries. NVIDIA’s policy-deployment guidance illustrates why these details matter: a policy requires compatible joint order, gains, observation/action processing and timing [24]. We generalise that implementation concern into a

receiver-side assessment obligation. The memory’s MBOM records the source configuration and transformations; the Robot BOM records the receiving configuration. Transfer between them requires explicit coverage or an assessed adaptation, not equal task names or a high retrieval score.

The unit of assurance is the combination of β_j , the actual memory view and retained context, the requested reuse role, the task and the operating conditions. Separate approval of a skill, AI model, robot and memory does not establish this combined claim. Nor must every accepted deployment be byte-identical to the source robot: evidence may justify a bounded family of configurations or a specific transfer relation. Such coverage must preserve the relevant correlations between components. A report for skill A with gripper X and skill B with gripper Y does not cover either skill with either gripper.

A reliability claim is therefore scoped to defined outcomes, conditions and uncertainty. Reliability is not an intrinsic property of a memory. Identical artifacts do not guarantee identical outcomes in a stochastic physical system. Conversely, a changed component need not invalidate every claim: a justified impact analysis may show that a specific measurement interpretation or use remains supported. The selected evidence witness records that analysis. Unsupported equivalence, unknown installed state or missing dependencies cannot be treated as proof that the old assurance transfers.

4 Architecture and memory lifecycle

4.1 System boundary and adversary

The proposed architecture separates the acting policy from the services that capture experience, assess claims, admit representations and authorise physical execution. Within an assessment interval it assumes the stable baseline β_j of Section 3.6 and A5. Adaptation occurs through controlled additions to external memory and changes to the experience supplied as context. A change to a skill implementation, runtime, model artifact or retrieval rule creates a new configuration version. Runtime inputs selected within an assessed contract do not by themselves change that baseline. A capture service records episodes; local and backend assessors produce scoped endorsements; each receiving robot decides which representations it may reuse. A protected gateway mediates consequential commands. A separately assessed controller provides physical restriction or recovery. These are logical roles. Their implementation may be colocated, but the isolation boundary must match the stated attacker capability.

Enterprise and edge responsibilities

Let $\mathcal{P}$ be the set of principals, each with policy-recognised identity bindings and explicitly scoped authorities. Six logical roles divide responsibility. A **Robot Original Equipment Manufacturer (OEM)** identifies the manufactured device, boot and attestation roots, supported configurations and reference measurements. An OEM identity credential does not give the OEM unrestricted authority over an operator’s tasks. A **Fleet Operator** holds the locally recognised operational mandate, determines admission policy, enrolls robots and delegates bounded consumption authority. A **Sensor/Calibration Provider** asserts a sensor or tool’s identity, calibration results, uncertainty and valid envelope. Its authority concerns the specified measurement claim, not general robot operation.

A **Third-Party Assessor/Auditor** issues identified assessments of claims, methods or processes within its recognised competence and scope. Independence, where required, is separately justified;

a different DID is insufficient. An **Ingestion/Curator Pipeline** records transformations, preserves dependencies and packages candidate memory. It may not promote an episode merely because it can sign or distribute it. An **Edge Robot Agent** captures local experience, presents its machine identity and requests bounded reuse. Its protected admission and execution services enforce the receiver’s policy. This logical separation can be implemented within one organisation, but permissions and compromise domains must remain explicit. The three evidence credentials in Appendix C do not replace the operator’s separate delegation instrument.

Enterprise wallets manage organisational keys, credentials and issuance policies. In the proposed deployment, cloud hardware security modules (HSMs) protect issuer keys for the operator, curator and participating providers or assessors. Their authenticated outputs cross the enterprise boundary as verifiable assertions. They are not trusted merely because they arrived through the fleet backend. The edge machine wallet uses device-bound keys protected by a Trusted Platform Module (TPM) or supported trusted execution environment (TEE) and exposes narrow signing and presentation operations. A wallet manages credentials and keys; the admission gate, protected status state and execution gateway enforce policy. Protecting a key without protecting those decisions does not satisfy the architecture.

The operator delegates an envelope to a named robot or to a designated fleet service governing a specific fleet. A fleet service may issue robot-specific child grants only within the onward-delegation authority it receives. The envelope specifies resources or package namespaces, claim and task scope, permitted reuse roles, operation class, configuration and operating domain, validity interval, maximum status age and onward-delegation constraints. Fleet-to-robot delegation requires a current membership binding to the receiver’s device key. Child envelopes can only narrow the parent scope. Collection, assessment, distribution, memory consumption and physical execution are distinct privileges; a consumption delegation does not itself grant execution authority. Trust roots and any changes to them are provisioned through a separate authenticated governance process. A package cannot introduce its own trust anchor.

Two forms of trust are distinguished. Enforcement trust concerns verification, admission rules, protected keys and status, dependency recording, state binding, clocks and command mediation. Evidence reliance concerns the fallible assertions supplied by sensors, evaluators, simulation models and other robots. The protocol can enforce a mistaken assessment correctly. Authentication establishes the origin and integrity of an assertion; it does not establish the assertion’s physical truth. The architecture therefore treats source identity, evidence support, permission to reuse and permission to execute as separate decisions.

The adversary may submit misleading material through permitted interfaces, exploit ingestion software, modify unprotected storage, substitute retrieval artifacts, replay valid older state, or compromise a contributing robot or validator. An unknown vulnerability is classified by the access it grants. The design makes no general claim to detect zero-day exploitation. Narrow candidate-write interfaces, isolated parsers, restricted privileges and resource limits reduce the exposed surface and constrain some consequences. Their effectiveness depends on the exploit remaining outside the protected enforcement boundary. A separate process on a compromised kernel does not supply isolation against that kernel.

The attack surface includes records, outcome labels, provenance, embeddings, index membership, retrieval settings, intermediate summaries, active context and command queues. It also includes mechanisms that interpret ordinary observations as instructions. An authentic recording can contain malicious text or misleading sensor input. Such material requires claim assessment and controlled representation, even when no stored byte has been changed. A compromised protected verifier, dependency recorder or execution gateway violates the corresponding protocol assumptions. These cases delimit the claims rather than being silently attributed to successful monitoring.

Established adversarial-machine-learning taxonomies distinguish poisoning and privacy attacks [25]. The following mapping additionally separates execution compromise and is specific to the proposed protocol.

Table 1: Threat capabilities and mitigation boundaries. These are design obligations, not results of an attack benchmark.

Attacker capability	Relevant control	Residual boundary
Writes to records, indexes or labels	Authenticated manifests, protected reference digests and verification on use	Authentic false assertions still require assessment
Substitutes buffers after verification	Sealed views, protected state versions and complete mediation	Uncovered device or remote state breaks the binding
Replays older accepted state	Durable epochs, absolute expiry and single-use permits	Unknown remote withdrawal remains bounded only by stated delivery or expiry assumptions
Uses legitimate ingestion to persist poison	Candidate isolation, scoped assessment and separate role grants	A fallible assessor can still approve harmful content
Leaks keys or other secrets through a side channel	Key isolation, reviewed implementations and platform-specific leakage controls	Hashing does not detect passive leakage; compromised trusted keys weaken authentication
Exhausts or compromises the monitor	Reserved checking capacity, health leases and restricted operation on loss of coverage	A compromised enforcement component violates the relevant trust assumption

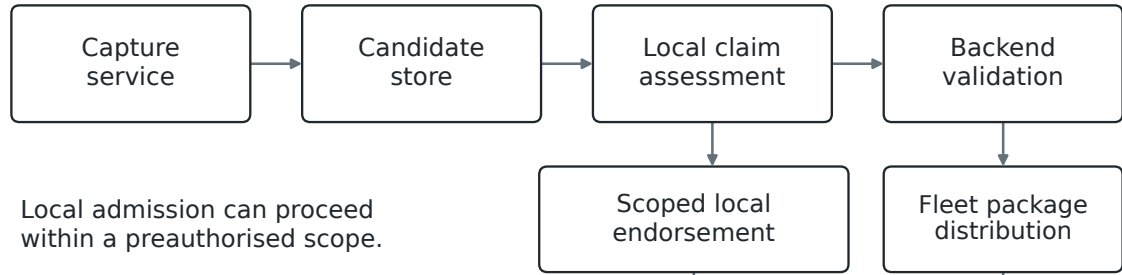
4.2 Protected memory verification

Memory integrity protection combines authenticated commitments with verification of the representation actually consumed. Immutable records and their metadata are bound to authorised manifests. Indexes and derived views identify their source records, encoder or transformation version, and retrieval configuration. Reference commitments, status epochs and verification policy reside within the protected boundary. Verification occurs during ingestion, loading and use; background scans and write-event monitoring provide supplementary coverage. Verified file reads are one implementation mechanism, but file integrity alone does not cover record membership, names, deletion or application semantics [26].

An authenticated snapshot prevents an unauthorised update from becoming an accepted replacement only when the verifier checks the relevant commitment and protects the subsequent use. A check followed by an unprotected read permits substitution between verification and consumption. The proposed loader therefore supplies a pinned representation or a protected state-version identifier with recorded dependencies. Legitimate updates create new versions through admission. They do not overwrite the accepted state or retrospectively alter its audit record. Protected monotone status prevents rollback below a known version; current status or enforced expiry is still required to address newer withdrawals that have not yet arrived.

Active state needs its own coverage argument. Continuously changing accelerator buffers do not have a single stable reference hash. Central processing unit (CPU) isolation does not establish protection for device memory, direct memory access (DMA) or remote model caches. An implementation must instrument and protect the relevant state, use a supported immutable

A Capture and claim assessment



B Protected receiver services

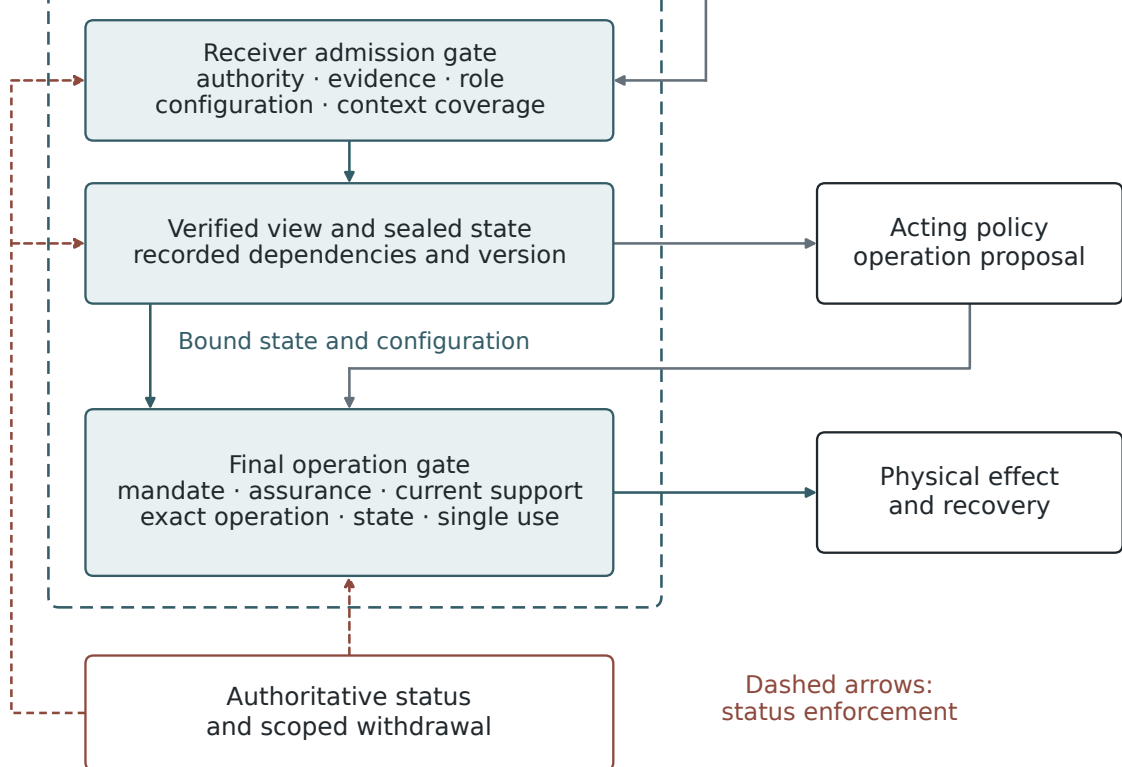


Figure 2: The proposed lifecycle separates capture and fallible assessment from protected receiver and operation admission. The receiver checks authority, evidence, reuse role, configuration and actual-context coverage. Withdrawal reaches both gates and affected retained state; delivery alone grants no permission. The enforcement boundary must cover the actual consumption and retained-state paths.

representation, or rebuild the affected session from currently admissible inputs in a trusted runtime. Inaccessible retained state cannot be declared clean through a session identifier. Where the required state cannot be controlled, the guarantee must exclude that path. Even complete input tracking establishes dataflow provenance, not a proof of semantic non-interference within a neural policy.

A mismatch establishes inconsistency with an authenticated reference. It does not establish malicious intent, identify an attacker or reveal the vulnerability used. Periodic inspection can miss a transient change restored before a scan. Passive side-channel leakage may leave every checksum unchanged; leaked keys can later support apparently authentic poison [27]. Key protection, reviewed cryptographic implementations, physical access controls and platform-specific leakage mitigations therefore address a different problem from content hashing. Fault injection and runtime corruption require their own platform assumptions. Monitor health supervision and reserved checking capacity support availability, but a heartbeat does not prove the correctness of the checker. The response follows a protect, detect and recover structure with explicitly bounded coverage [28].

4.3 From observed episode to receiver admission

New local experience enters a candidate store through the capture service. The record preserves observations, issued commands, measured feedback, timestamps and their uncertainty, configuration, measurement quality and relevant prior context. Interventions, blocked proposals, human assistance and interruptions remain explicit. A commanded grasp is distinct from its measured outcome. The edge machine wallet authenticates an episodic capture statement and binds it to the capture identity and available hardware-attestation evidence. This records responsibility for the statement; it does not establish that the sensor measured the physical world correctly. Capture authority grants neither claim endorsement nor future demonstration reuse.

A local assessor evaluates identified claims or segments within an operator-delegated envelope. It may check completeness, timing, sensor health, command-feedback consistency and additional observations. Its assessment is a new attributable object, including the method, result and dependencies. Shared sensors, calibration, models or execution environments remain possible common causes. The robot can therefore learn from its own experience without authorising its own unrestricted promotion policy: only the protected admission service can grant a supported role within the existing delegation. Communication loss cannot enlarge that scope or renew expired authority. A failed task can yield an accepted counterexample or a separately supported recovery segment; task success is not a prerequisite for preserving evidence.

Backend assessment receives an access-controlled package containing the exact record, claim, MBOM and transformation identifiers. Replay, configuration-aware simulation, held-out cases, hardware tests and human review supply different forms of support. Reproducing a faulty trace in a faulty model supplies no independent physical observation. Assessors sign role-specific results using their enterprise credentials. The curator signs the resulting package manifest and transformation attestations through its enterprise wallet. Cloud HSM protection limits key exposure under its implementation assumptions; it does not make the curated package factually correct. The fleet operator’s distribution approval and scoped consumption delegation are separately identifiable assertions. A distributor may transport these objects without receiving authority to endorse their content.

At the enterprise-to-edge boundary, the robot verifies the package commitment, producer and assessor credentials, issuer authority, selected delegation chain, freshness of relevant status, and compatibility with its own identity, fleet membership, tooling and task. The enterprise-signed package is an input to this decision, not an edge admission token. The edge wallet proves

possession of the recipient key where the protocol requires it, while the protected gate evaluates the applicable predicates and records a receiver-specific witness. The resulting local admission binds the package and role to a sealed representation, protected state version, policy version, status epochs and inherited expiry. The later operation permit is issued only after the separate mandate and physical-assurance checks in Section 5. Appendix C specifies the presentation and verification exchange.

Delivery, acceptance, activation, first use and completion remain distinct events. Limited release cohorts, source quotas and defined withdrawal conditions constrain propagation. Independent recipient decisions do not imply independent evidence. Copies and re-signatures preserve the original empirical origin; new analysis adds analytical coverage, while a new physical observation can still share a calibration defect. Replica handling applies before context allocation, so repeated packages cannot dominate retrieval despite an idempotent support rule. A receiver's new physical episode can enter the same candidate-to-assessment process and be shared with the backend without inheriting demonstration permission from its source package.

4.4 Withdrawal, receipts and recovery

Admission binds the selected representation, accepted claims, reuse role and retained state to their current dependencies. A concrete operation then receives a single-use permit covering the recipient, command, parameters, configuration, continuation contract, status epochs and expiry. The final gateway checks the mandate and physical-assurance requirements for that same operation. Permit validation, consumption and admission share a serialisation order with local withdrawal installation. Downstream queues require equivalent mediation; otherwise queued commands count as committed work outside the new-admission guarantee.

Withdrawal concerns an identified claim, endorsement or permitted scope. Typed dependencies distinguish construction, assurance reliance and causal history. Removing a completion claim can invalidate demonstration reuse without falsifying a separately supported force measurement. If the capture system or calibration is compromised, that measurement may require reassessment too. Corrected claims retain their history. Audit retention uses separate access rules: preserving an incident containing malicious instructions does not make it suitable for operational context.

Receipts identify the operation, receiver and state transition that was acknowledged. Delivery is not acceptance, and acceptance is not completion. Pending effects remain recorded until supported completion or cancellation evidence arrives; a timeout does not establish cancellation. On an integrity alert or unavailable support, the system restricts affected use, invalidates unused dependent permits and reconciles pending work. Recovery may require stabilising a payload rather than removing power. Its authority, observations and physical contract must be justified separately. These transitions specify the proposed protocol; their latency, coverage and physical consequences remain subjects for implementation and evaluation.

4.5 Proportionate revalidation and configuration changes

In this protocol, **artifact and admission verification** checks commitments, identities, status and declared conditions. **Validation** establishes evidence that a particular use meets specified requirements in its intended setting. **Revalidation** reassesses whether that evidence remains sufficient after a new input, changed premise or finding. It may reuse existing evidence; it does not always require a new physical experiment. Every operational memory admission and affected effect admission performs the applicable current checks. Repeating an entire experimental programme for every retrieval is neither required by this protocol nor justified by the mere fact of retrieval.

An accepted **memory-update contract** states which memory representations, roles, content conditions, combinations and update procedures are covered for a baseline and task envelope. It specifies context limits, ordering or composition constraints, required content assessment, monitoring and restriction conditions. Coverage must concern the memory-conditioned system, not just the policy without memory. A format check or a claim that all memories come from the same robot cannot establish that coverage. If the system has no justified class-level contract, admission requires evidence for the particular memory-conditioned use.

Four cases determine the extent of reassessment:

1. **Use already covered by current evidence.** An unchanged admitted view may be reused within the same assessed combination and envelope after current applicability, authority, status, state and monitor checks. No additional experiment is required by this protocol. The decision still refers to the accepted witness; an old allow response is not a reusable operation permit.
2. **A new memory within an assessed update contract.** The receiver assesses its provenance, content, role and compatibility, and checks the resulting context composition against the contract. Existing behavioural evidence may discharge the experimental obligation if it covers that class of update. It does not remove the need to assess the new episode’s claims. Several individually acceptable memories may form an unsupported or conflicting combination; composition coverage is required before activating them together.
3. **An uncovered change or proposed use.** A new gripper, skill version, model adapter, action mapping, retrieval rule, memory role or task envelope requires an impact assessment. If an existing assessed equivalence or transfer relation covers the change, it can support a fresh admission without repeating experiments. Otherwise targeted replay, simulation, hardware tests or qualified review must supply the missing evidence. A broad model or control change, or uncertainty about the affected dependencies, can require system-level revalidation. No generic distance between model versions or memory embeddings establishes a sufficient validation scope.
4. **Contradictory evidence, withdrawal or drift.** An incident, calibration defect, observed response change or withdrawn supporting assessment restricts the affected use while its premises are reassessed. Unchanged digests do not justify continuation. Recovery follows its separately assessed contract. A change outside A5 ends the current assessment interval; the original episode can remain evidence even when its operational reuse is suspended.

For example, an assessed contract might cover grasping specified containers with a particular parallel gripper, fixed skill/model versions and bounded memory views. A newly assessed slip episode could enter as a counterexample without repeating all hardware trials if that use and context composition are covered. Using its action sequence as a positive demonstration is a different claim. Moving to a suction tool requires a new compatibility decision; existing geometric observations may remain useful while grasp-force advice loses its support. These are illustrative decisions conditional on adequate evidence, not reported experimental outcomes.

An authority-only change can require a new delegation or witness without new physical tests when the underlying behavioural evidence remains applicable. A status refresh alone neither renews that evidence nor establishes physical stability. An unrelated installed component need not trigger whole-system revalidation if sound dependency and isolation evidence excludes it from the affected use.

The evidence graph records the baseline commitment, source and receiving Robot BOMs, memory root and view, role, task envelope, update-contract identifier, tests or analyses, limitations,

assessor, authority, current status and the receiver’s decision. A coverage assertion is an attributable assessment node with materialisation and assurance dependencies. The selected witness includes either applicable existing coverage or the new validation result, together with the evidence that the installed configuration matches. The graph thereby records the justification for further testing or for relying on existing evidence; a signed assertion that no testing is needed cannot justify itself.

Before activating a planned configuration change, the receiver restricts affected use and resolves pending operations under their continuation or recovery contracts. A configuration transition then installs a new authenticated baseline and advances the relevant protected configuration/status state in the same order as admission. Unused permits bound to the old configuration cannot be transferred. Affected retained context is revalidated or reconstructed from admissible inputs under A2-A4. Unchanged observations and independently supported claims remain available. The new decision and any new operation permit bind the new baseline and selected witness; only justified, unaffected support is reused.

5 Formal model and conditional guarantees

5.1 Objects, support witnesses and protected state

An episode is an immutable capture object

$$e = (\text{id}, h(p), m, P, \sigma), \quad (1)$$

where p contains recorded observations, commands, feedback and interventions; m identifies capture conditions; and P identifies provenance parents. The source signature σ authenticates a canonical, domain-separated encoding of $(\text{id}, h(p), m, P)$ under the issuer and verification-method binding recorded in the capture metadata. The payload digest is computed over the exact identified payload encoding. It does not establish that a sensor reading, self-report or interpretation is correct. A claim c identifies an episode or segment and a statement about it. An endorsement binds a claim to a validator, result, supporting evidence, method, assessed domain, permitted reuse roles, assumptions and current-status reference. Consequently, an episode can support a rejected completion claim and a separately accepted failure measurement.

A receiver assesses a claim set C for task scope s and reuse role r . Roles distinguish positive demonstration, counterexample, recovery support and other uses. Audit and assessment reads follow separate access rules. They do not automatically authorise exposure to the acting policy.

Evidence requirements may contain alternatives. For example, a claim may require either accepted assessment b_1 or accepted assessment b_2 , together with a current calibration claim. An admission decision therefore records a **selected support witness** W : a finite, acyclic derivation showing which requirements and alternatives were accepted. All conjunctive requirements must be represented; an accepted disjunctive requirement records its selected branch. This prevents an unspecified alternative from retrospectively justifying a stale decision.

Let G_t contain episode, claim, endorsement, representation, context, plan and permit nodes, together with enterprise/device identity bindings, calibration assertions, transformation environments, model artifacts, delegations and status objects. Edges distinguish materialisation, selected assurance reliance and causal history. Materialisation records which inputs constructed an artifact. Assurance edges record the premises of a particular witness, with live authority reliance as an explicitly labelled subtype. Causal-history edges record preceding events with-

out automatically establishing evidential dependence. Shared calibration, model or validation dependencies appear wherever the relevant claim relies on them.

The protected state is

$$\Sigma_t = (G_t, \nu_t, Q_t, V_t, \Pi_t, N_t, L_t), \quad (2)$$

where ν_t contains monotone status epochs and authority-policy high-water state, Q_t scoped restrictions, V_t sealed representation/state versions, Π_t permits, N_t consumed nonces and L_t unresolved operation records. The Robot BOM baseline β_j , index and retrieval configuration, monitor coverage and health leases, and approved execution configuration are authenticated dependencies. This state describes governance; it is not a complete physical model of the robot. The authenticated configuration includes P_t^j , the receiver's authority policy.

Scoped delegation and issuer authority

Let P_t^j be the protected policy of receiver j . It identifies accepted authority roots, their action and claim domains, allowed verification methods, status authorities and delegation limits. An authority root grants a defined scope; it is not a universal trusted-issuer flag. An authorised policy change advances the policy epoch. Neither an incoming MBOM nor a new DID can provision its own root.

A request is $q = (p, z)$, where p is the acting principal and $z = (j, f, a, o, r, s, x, t)$ identifies the receiver, fleet, action, resource or claim identifiers, reuse role, task, configuration and operating context, and decision time. Fleet membership and environmental conditions are verified inputs. Actions distinguish capture, assessment, endorsement, distribution and consumption. A physical operation has a separate mandate. A scope Θ denotes the set $\mathcal{S}(\Theta)$ of complete request contexts; attenuation means $\Theta' \preceq \Theta$ exactly when $\mathcal{S}(\Theta') \subseteq \mathcal{S}(\Theta)$. The principal is outside this tuple because delegation changes the principal while narrowing its permissible requests.

A chain $\pi = (g_0, d_1, \dots, d_\ell)$ starts at a local grant g_0 to principal p_0 . Each signed delegation d_i binds its issuer p_{i-1} , subject p_i , parent digest, scope Θ_i , remaining delegation budget $b_i \in \mathbb{N}_0$, validity interval and status reference. The parent must permit further delegation, $b_{i-1} > 0$, $b_i \leq b_{i-1} - 1$, and $\Theta_i \preceq \Theta_{i-1}$. For a consumption request, the terminal subject is the named robot or enrolled policy-workload principal. A fleet-level delegation names a designated fleet service as delegate; that service issues an explicit attenuated child grant to the named robot. Current membership and device-key binding are additional scope conditions, not permission for a robot to assume the fleet service's identity or delegation rights. A direct root grant is the zero-delegation case. Separate paths may discharge separate obligations, but their permissions cannot be spliced into a broader request.

An implementation uses a restricted scope language with sound inclusion checking; unsupported or unproved inclusion is rejected. Inclusion must preserve correlations between dimensions. A parent that permits demonstration for object A and counterexample use for object B does not permit demonstration for both objects. Child expiry cannot exceed parent expiry, and removal of a configuration constraint is a widening. Operational scopes and evidence roles are not arbitrary natural-language strings or an implicit hierarchy of privileges.

Conceptually split the selected witness as $W = (W_{\text{ev}}, W_{\text{auth}})$. The first part records accepted evidence and alternative branches. The second pins the chosen authority paths, root-policy version and live authority dependencies. Historical-authority records establish the authority asserted at capture or issuance under the receiver's timestamp and key-history policy. Live-authority reliance requires current eligibility for the selected operational use. An expired capture

credential does not alone erase an observation captured during its valid interval; a current robot membership or consumption delegation is a live condition. Compromise findings can nevertheless require historical assertions to be reassessed.

5.2 Assumptions and transitions

The following assumptions delimit the guarantees.

A1 - Protected enforcement. The adversary cannot alter the verifier, gate rules, protected reference commitments, relevant keys, locally provisioned authority policy or durable status state. Accepted DID resolution, key history and status-signer bindings satisfy the verification policy; an untrusted package cannot supply replacement trust anchors. Canonical serialisation is unambiguous; the authentication and hash primitives satisfy their stated security assumptions.

A2 - Complete mediation and sound lineage. Every acting-policy reuse and ordinary physical-command admission crosses its respective gate. Recorded dependencies conservatively include relevant input, selected evidence, selected live authority paths and retained state. Scope inclusion checking is sound for the admitted policy language. Extra dependencies are permitted. Unknown lineage causes restriction or complete reconstruction; it is not treated as independence. Untracked remote state or accelerator buffers violate this assumption.

A3 - Bound consumption. A verified representation is sealed before use. The policy consumes that representation, and retained state transitions preserve its dependency history. A state identifier names a protected version, not merely a reusable session name.

A4 - Serialised and durable admission. Status installation, authorised authority-policy and baseline-configuration updates, permit validation and irreversible command admission have a common local serialisation order. A root removal, scope change or relevant configuration transition cannot race a cached successful check; relevant policy-version and epoch changes participate in the final transaction. Once consumption and insertion of the unresolved-operation record are durably committed in the same admission transaction before command release. If release or acknowledgement is uncertain, the operation remains unresolved until supported reconciliation establishes its outcome; timeout alone does not permit retransmission or establish cancellation. Restart preserves epochs, restrictions, consumed nonces and unresolved operations. Commands beyond the boundary are already committed unless the downstream receiver supplies an equivalent guard.

A5 - Stable baseline during an assessment interval. The installed baseline β_j and its relevant software, skill and AI artifacts are fixed. The underlying robot, sensor and controller response remains within the assessed tolerances and uncertainty model, with no unmodelled wear, calibration, timing, behavioural or other drift. Task inputs, physical state and environmental variation remain within the declared assessment envelope. The model's response may be stochastic, and admitted memory may deliberately change action selection; neither effect is excluded. A5 isolates memory-conditioned adaptation from changes in the underlying system. It does not assert that the changed behaviour is safe or useful. A digest match establishes artifact identity, not this physical stability assumption. Evidence of departure ends the interval and triggers Section 4.5. The protocol properties under A1-A4 do not prove A5; it is an additional scope assumption for interpreting configuration-specific behavioural evidence.

The transition system permits capture, endorsement, role admission, representation construction, permit issue, effect admission, restriction, reinstatement, assessed configuration transition and restart. Capture appends a candidate without granting reuse. Endorsement adds an attributable assessment without granting receiver permission. Only the protected receiver gate grants a role-specific admission. Only the execution gate admits a concrete operation. Restriction advances

the relevant epoch; reinstatement requires the designated status authority and advances it again. For an irrevocably revoked credential, reinstatement means a fresh credential and decision; lifting a reversible suspension also advances the epoch. Neither revives an old permit. Recovery commands require their own mandate and assurance. A timeout does not close an unresolved operation.

5.3 Representation and operation admission

For equations (3)-(4), fix a protected receiver decision record containing the baseline β_j , the proposed context/state identifier χ and the relevant operating-condition observations ζ_t . The request-context field x in Section 5.1 binds these values. Every predicate in that decision uses the same record; an untrusted package cannot replace its fields. Suppressed arguments below refer to this fixed record. Define the receiver predicate

$$\begin{aligned} E_t^j(e, C, s, r, W) = & \text{AuthenticMember}_t(e, C, W) \wedge \text{Authorised}_t^j(e, C, s, r, W) \\ & \wedge \text{CurrentApplicable}_t^j(C, W, s, r) \wedge \text{WitnessAccepted}_t^j(W) \\ & \wedge \neg \text{Restricted}_t^j(e, C, s, r) \wedge \text{MonitorReady}_t^j(s, r). \end{aligned} \quad (3)$$

The authority conjunct explicitly depends on the selected witness. For its named path, define

$$\begin{aligned} \text{AuthorityOK}_t^j(q, \pi) \iff & \text{LocalAllows}(P_t^j, q) \wedge \text{RootedAt}(\pi, P_t^j) \\ & \wedge \text{AuthenticLinked}(\pi) \wedge \text{Attenuated}(\pi) \\ & \wedge \text{AuthorityStatusOK}_t^j(\pi) \\ & \wedge \text{SubjectBound}(q, \pi) \wedge z(q) \in \mathcal{S}(\Theta_{\text{leaf}(\pi)}). \end{aligned} \quad (3a)$$

Here authentic linkage checks each signature, issuer-key binding, parent digest and issuer-to-parent-subject relationship. Subject binding ties the request to the terminal principal through a protected local identity, suitable proof of possession or the signed assertion being evaluated. A credential holder need not be its subject. Attenuation includes the delegation budget and scope constraints above. Define

$$\text{Authorised}_t^j(e, C, s, r, W) \iff \bigwedge_{q \in \mathcal{Q}_j(e, C, s, r, W)} \text{AuthorityOK}_t^j(q, \pi_q). \quad (3b)$$

Protected policy generates $\mathcal{Q}_j$, which necessarily includes the robot's consumption request and all live-authority obligations required by policy for selected endorsements and operational premises. Equations (3a)-(3b) evaluate these live requests at decision time t . Historical capture and issuance authority is checked separately at its reference time by authentic membership and witness acceptance, subject to current retrospective compromise or withdrawal restrictions; it is not silently required to remain live forever. The sender cannot omit an obligation to obtain an empty conjunction. Each π_q is fixed in W_{auth} . This is the explicit-witness refinement of $\text{Authorised}(e, C, s, r)$; selecting a different path requires a fresh decision.

AuthorityStatusOK verifies issuer recognition and applicable issuer-authority credentials, delegation credentials, key restrictions, local denials and authorised status-signer relationships. For live dependencies it also checks absolute validity, accepted status freshness and protected rollback state. A Bitstring Status List supplies the status of a particular credential; it is not by itself an issuer-trust registry [29]. Status-authority verification terminates at protected anchors and cannot bootstrap itself through circular credentials. "Current" means acceptable under the

declared freshness policy, not instantaneous knowledge of all remote withdrawals. Unknown status beyond the permitted interval blocks the affected operational use.

Authentic membership includes manifest, index and transformation bindings. The identifier χ names the proposed sealed context and retained-state version after staged incorporation of the memory. Let $\mathcal{V}_W$ be the assessed coverage relation selected in W_{ev} , over complete tuples $(\beta, \chi, C, s, r, \zeta)$. Coverage can describe one assessed instance or a justified class under a memory-update contract. Define

$$\begin{aligned} \text{CurrentApplicable}_t^j(C, W, s, r) &\iff \text{InstalledMatch}_t^j(\beta_j) \\ &\quad \wedge (\beta_j, \chi, C, s, r, \zeta_t) \in \mathcal{V}_W \\ &\quad \wedge \text{CoverageCurrent}_t^j(W, \beta_j, \chi, s, r, \zeta_t). \end{aligned} \tag{3c}$$

The last conjunct checks the selected coverage assessments and their applicable conditions, status and expiry, including required calibration and monitoring evidence. Membership in $\mathcal{V}_W$ is an accepted assessment under the stated methods and uncertainty, not a proof of reliable behaviour. The receiver must resolve the proposed context and baseline before granting admission; until then the predicate is false. Witness acceptance determines whether the claim-specific evidence and coverage method satisfy the assessment policy; it is not implied by AuthorityOK. A failed episode must meet these obligations for its intended role. Failure alone grants no permission.

Candidate representations and any speculative context state are constructed in an isolated staging path without permission to issue ordinary commands. Operational activation occurs only after the protected checks accept the exact staged representation and state. An implementation that cannot stage or reconstruct the relevant state within its declared boundary cannot claim this admission guarantee for that path.

A deployment profile specifies bounded, machine-checkable applicability conditions for the selected coverage assertion and identifies the assessment evidence that justifies them. Unsupported conditions or unresolved membership block the affected use. The gate does not decide whether an arbitrary neural policy is safe.

The reuse gate enforces

$$\begin{aligned} \text{PolicyReuse}_t^j(e, C, \xi, s, r, W) &\Rightarrow E_t^j(e, C, s, r, W) \\ &\quad \wedge \text{VerifiedPinned}_t^j(e, C, \xi, s, r, W) \\ &\quad \wedge \text{UseCovered}_t^j(\beta_j, \xi, v_{\text{state}}, s, r, W). \end{aligned} \tag{4}$$

Here ξ is the actual representation consumed. An authenticated file followed by an unprotected mutable buffer does not satisfy the second conjunct. UseCovered checks that this representation, its role and ordering, and the actual assembled context and retained state identified by v_{state} match the complete combination covered by the selected witness. It rechecks equation (3c) for the actual consumption, including its operating conditions. Individual eligibility of every source episode is insufficient to discharge this composition check. If assembly or retained-state evolution falls outside the accepted coverage, a fresh assessment is required before that use. This is enforcement of a scoped assessment, not proof of the policy's interpretation or physical reliability.

For operation u , let $D(u, W)$ be the conservative closure of its actual materialisation dependencies and selected assurance premises, including the live authority paths in W_{auth} . This closure follows dependencies back to their premises and includes selected delegations, issuer-authority and status-signer bindings, the protected policy version and applicable status checkpoints. Historical provenance is labelled separately from a requirement for continuing authority. Unselected

MBOM alternatives are not automatically premises. The closure also contains the selected Robot BOM baseline, memory-composition coverage and any transfer or revalidation assessment. The operation identifies its principal, recipient, command, parameters, deadline and continuation contract. Its authenticated permit binds

$$\eta = \text{Auth}_K(\text{dom}, j, h(u), v_{\text{state}}, h(W), h(D(u, W)), \nu_t|_{D(u, W)}, h(\gamma_t), T_\eta, n). \quad (5)$$

Here dom identifies the permit protocol and encoding, j identifies its gate audience, and K is a key recognised for permit issuance by that receiver. The profile distinguishes permit authentication from evidence-credential signatures. A permit for another audience or protocol is rejected. Let W_η denote the selected witness authenticated by this permit. Binding validation verifies the permit authenticator, the witness digest and the digest and scoped epoch bindings of the presented dependency closure. Here γ_t includes the commitment to β_j , execution configuration and monitor coverage/health; W binds the selected configuration coverage and its premises. Memory and retained context remain separately bound by their representation and state versions. A relevant configuration change invalidates this binding even if the memory bytes are unchanged. The deadline T_η cannot exceed the applicable absolute expiries of selected live authority, status, assessment and monitor dependencies. Changing the selected authority path or relevant policy version requires revalidation and a new permit; an unused alternative cannot repair an existing one. At the final boundary the gate checks the exact operation and actual protected state version, current dependency status and monitor readiness, matching epochs, an unused nonce, and current mandate and physical-assurance decisions. These checks and admission form the transaction required by A4.

Writing these obligations explicitly, the final predicate is

$$\begin{aligned} \text{Admit}_t(u, \eta) \iff & \text{Mandate}_t(u) \wedge \text{Assurance}_t(u, X_t) \\ & \wedge \text{DependenciesValid}_t(D(u, W_\eta)) \\ & \wedge \text{BindingEpochValid}_t(u, \eta) \\ & \wedge \text{Unexpired}_t(\eta) \wedge \text{Unused}_t(\eta). \end{aligned} \quad (6)$$

Dependency validity includes the continuing applicability of the selected evidence and coverage relation to the operation's current task, configuration and operating conditions, using the observations required by that assessment. A fresh status response alone does not establish this applicability. The separate physical-assurance predicate evaluates the proposed operation and relevant physical state. X_t denotes the physical-state information required by that mechanism. Binding validity includes the actual sealed state version, operation parameters, execution configuration and monitor readiness. A successful test of this predicate is not a reusable allow decision: the test, nonce consumption and commitment share one serialised transaction.

Proposition 1 - Bound representation integrity. Under A1-A3, a consumed admitted representation matches its authenticated byte commitment and protected version, except with the failure probability of the cryptographic assumptions.

Proof. Verification checks source and output commitments and their declared transformation bindings. Accepting altered committed bytes requires either an accepted unauthorised commitment or distinct bytes matching the verified digest, contradicting authentication or collision resistance. After verification, A3 prevents substitution before or during consumption. A2 binds the resulting state version and its dependencies. Therefore a different consumed representation requires a cryptographic failure or violation of A1-A3. Authentication alone does not prove that an untrusted transformation was computed correctly; that requires separate assessment or

verified execution. The argument neither validates the physical event nor proves that the policy interprets the view as intended. $\square$

Proposition 2 - Scoped authority and role non-escalation. Fix a well-formed protected local policy and assume A1-A4, sound scope inclusion checking and the delegation/status rules above. Every newly admitted reuse request lies within its accepted root grant and every selected delegation on its recorded authority path. Capture, copying, delivery, authentication, an endorsement outside the required domain, or admission in another role cannot alone create that permission. Admission of a governed physical command additionally requires the current mandate and assurance of equation (6).

Proof. A zero-length path permits only requests in its root scope. Appending a delegation requires a permitted issuer, authentic parent binding, remaining delegation budget and a child scope contained in its parent. Induction on path length and transitivity of set inclusion place every permitted request within all ancestor scopes. A1 excludes unauthorised key or path substitution under the cryptographic assumptions. Authority/status checks reject a selected path after relevant local withdrawal installation and enforce its inherited expiry otherwise. Only the protected admission transition adds reuse permission, and it checks equation (3) with the recorded paths. Capture, copying, distribution and assessment add objects or assertions without grants. Restriction narrows permission; reinstatement requires a fresh authorised decision and epoch; restart preserves restrictions under A4. Induction over transitions establishes the invariant. Complete mediation and equation (6) separately impose execution obligations. The claim concerns granted authority, not factual truth or a policy’s semantic interpretation of a counterexample. $\square$

An authorised change of root policy may deliberately widen permissions; it is a separately governed transition, not attenuation along an existing chain. A fresh assessment and separately authorised demonstration admission may also promote a suitable segment of a failed episode. Its previous counterexample permission is insufficient for that promotion.

5.4 Replication and evidence dependence

Let κ map verified transport representations to canonical logical evidence objects. A recognised pure replica copies or re-encodes existing material without adding an observation or substantive assessment. It retains the original evidence and assessment references. The canonicaliser is assumed sound for the admitted replica transformations and does not merge substantively different evidence. Define

$$\begin{aligned} K(A) &= \{\kappa(a) : a \in A\}, \\ \Phi_j^{\text{sup}}(A; c, s, r) &= f_j(K(A); c, s, r). \end{aligned} \tag{7}$$

Proposition 3 - Pure-replica invariance. With authority, status, applicability and the assessment policy fixed, adding recognised pure replicas leaves evidential support unchanged.

Proof. Each added replica has the same canonical object as an existing member. Thus $K(A \cup \text{Rep}(A)) = K(A)$. Substitution into equation (7) yields equal support decisions. A ranker receiving the same canonically ordered objects and fixed randomness also receives unchanged input; replicas cannot obtain additional context slots. This conclusion requires canonicalisation before allocation, rather than merely deduplicating a reported confidence value. $\square$

A new authorised approval may change permission while adding no physical observation. A new analysis or derived view can add useful analytical coverage and is not automatically a pure replica. Distinct observation origins do not establish statistical independence: shared

sensors, calibration, models or datasets can produce common errors. The proposition establishes idempotence for recognised copying, not independence or detection of dishonest origin aliases.

5.5 Scoped withdrawal and alternative support

For withdrawn dependency q and scope ω , define

$$\mathcal{A}_t(q, \omega) = \{v : q \rightsquigarrow_{\omega}^{\text{mat,assur}} v\}. \quad (8)$$

The relation $\rightsquigarrow_{\omega}^{\text{mat,assur}}$ includes a zero-length path and follows only materialisation and selected assurance edges relevant to the withdrawn scope. Scope propagation is conservative: where the implementation cannot establish that a dependent use is outside the affected scope, it restricts that use pending reassessment. Installation marks the affected uses unavailable and advances protected epochs. The closure includes the actual selected witness, with authority reliance included in *assur*. Withdrawal of a selected parent delegation, issuer authority or robot membership therefore reaches dependent admissions and unused permits without asserting that their underlying observations are false. If b_1 is withdrawn from a witness using $b_1 \vee b_2$, the old permit does not become valid merely because b_2 exists. The receiver must assess a fresh witness and issue a new permit. Witness replacement cannot erase a materialisation dependency: retained context that actually used withdrawn material requires justified revalidation or reconstruction.

Withdrawal does not assert that every descendant observation is false. Causal-history edges support review separately. A withdrawn completion claim need not invalidate a separately supported force trace; withdrawal of its calibration support may invalidate the force interpretation. Records may remain available for investigation in both cases. This preservation is an access and retention property, not a truth guarantee.

Lemma 4 - Scoped preservation. Let I_{Σ} map archived identifiers to immutable bytes. If a withdrawal changes only scoped status, reliance metadata and affected active artifacts, then $I_{T(\Sigma)}(e) = I_{\Sigma}(e)$ for every previously archived object e . A decision with unchanged dependency and status bindings retains its eligibility value when all other gate inputs are fixed.

Proof. The withdrawal transition’s write set excludes archived content, proving the first statement. The second follows by substitution of identical inputs into equation (3). Status records can extend the archive without rewriting earlier objects. Shared dependencies, conservative over-approximation or complete session reconstruction can still interrupt useful work. The lemma therefore concerns retention and scoped eligibility, not uninterrupted execution or physical truth. $\square$

Theorem 5 - No stale-permit commitment after local withdrawal. Under A1-A4, once a scoped withdrawal is installed, no later ordinary admission can commit an operation relying on that withdrawn scope with its earlier permit.

Proof. Consider an admission ordered after installation. By A2, the relevant withdrawn dependency is included in $D(u, W)$, possibly through conservative closure. Installation either invalidates that dependency’s permitted use or changes an epoch bound by equation (5). The final gate consequently rejects dependency validity or epoch equality. A4 excludes an intervening withdrawal between a successful final check and its commitment. A permit created after installation cannot validly select the restricted premise. Reinstatement advances the epoch and therefore cannot revive the earlier permit. The argument also covers newly constructed descendants because final admission checks dependency status, rather than relying solely on previously marked graph nodes. A command ordered before installation is already committed and lies outside the theorem. $\square$

5.6 Timed exposure and counterexamples

Let authoritative withdrawal occur at t_r , and consider the withdrawal episode before any authorised reinstatement. On the reference time scale, define Δ_{admit} as the supremum of the elapsed times from authoritative withdrawal to any later stale-dependent admission before reinstatement, taking zero when no such admission occurs. Suppose delivery is bounded by L , affirmative status expires at most τ after its authoritative issuance time, relevant clock disagreement is at most ϵ , and local installation or expiry enforcement takes at most δ . Each relevant source must satisfy these assumptions, or conservative constants must cover all such sources. No affirmative status for the withdrawn scope may be issued after t_r ; cached responses and permits cannot extend its expiry. Then

$$\Delta_{\text{admit}} \leq \min(L, \tau + \epsilon) + \delta. \quad (9)$$

Proof. Delivery enables restriction by $t_r + L + \delta$. Independently, every previously issued affirmative statement expires no later than $t_r + \tau$ in authoritative time. Bounded clock disagreement and local enforcement make it unusable by $t_r + \tau + \epsilon + \delta$. The first effective mechanism supplies equation (9). During an unbounded partition, set $L = \infty$. Without either bounded delivery or enforced expiry, no finite cutoff follows. $\square$

If admissions satisfy a token-bucket envelope with burst b and rate λ , the number of additional admissions in this interval satisfies

$$N_{\text{additional}} \leq b + \lambda \Delta_{\text{admit}}. \quad (10)$$

This counts admissions, not damage. Detection and adjudication precede t_r ; earlier commitments and physical recovery require separate analysis.

The assumptions have direct counterexamples. A cached successful check followed by withdrawal and later dispatch violates Theorem 5 when A4 is removed. Database deletion without context invalidation or final dependency checks leaves an already prepared plan usable. Resetting an epoch after restart can make an old permit match again. Counting signatures instead of canonical evidence lets copied records inflate apparent support. Finally, an entirely authentic but false episode can pass every integrity check: the protocol preserves the distinction between enforcing an assessment and establishing its factual correctness.

6 Executable finite protocol study

6.1 Model, method and reproducibility

We implemented a finite transition model in Python and explored its reachable graph using breadth-first search. The purpose is to test the admission and withdrawal logic against adverse event orderings and explicit enforcement mutations. The model is an executable abstraction of the protocol, not a robot implementation. It does not execute a learned policy, cryptographic algorithm, simulator or physical controller.

The abstraction contains one receiving robot, one captured episode, two reuse roles, one active context, one retained permit and one operation identifier. The roles are demonstration and counterexample. Demonstration status has three epochs: initially live, withdrawn and reinstated following an abstract authorised reassessment. Counterexample status remains unchanged. Delivery and installation are distinct transitions; late status messages cannot lower the protected

epoch. One irreversible deadline event represents expiry, and at most one crash/restart is permitted. This finite domain is an explicit modelling restriction. The search has no depth cutoff and continues until every reachable state has been expanded.

An abstract context is identified only by its role and epoch. This is sufficient to expose the selected status and reuse races but does not represent a complete sealed byte-state identity. The permit represents one operation identifier and its role/epoch binding. Specification-only history records whether the operation ever committed, the highest locally installed epoch and whether completion remains unresolved. These history fields expose rollback and replay; weakened implementations cannot use them to repair their state.

Nine properties are checked: role-grant separation, consistency with installed status, single use, abstract context binding, unexpired commitment, pending-ledger preservation, status high-water preservation, scoped preservation and fresh-epoch use after reinstatement. The last property is a focused subcase of status consistency. Commit properties are checked on every enabled commit edge. State properties are checked on every successor, and preservation is checked on the relevant installation transition. Breadth-first predecessor chains produce shortest counterexamples under the model’s action vocabulary.

The accompanying supplement contains the complete source, model bounds, deterministic output, state-by-state counterexamples and a source digest. The command `python3 check_protocol.py --out results` reproduces the study with Python 3.10 or later and no external packages. The outputs report labelled transitions, including enabled self-loops, separately from distinct states. Counts describe the explored graph; they are not measures of security strength or runtime overhead.

6.2 Results and enforcement mutations

Table 2 reports the complete reachable graphs. No checked property is violated in the protected abstraction. Each weakened variant changes a stated enforcement condition while retaining the remaining model structure. These mutations provide positive controls: the checker reports concrete failures when conditions used in the formal argument are removed.

Table 2: Finite-model exploration results. “Transitions” denotes enabled labelled edges, not execution trials. Reinstatement also exposes stale-epoch use in the first three weakened enforcement variants.

Variant	States / transitions	Observed property failure
Protected	3,192 / 12,494	None among the nine checked properties
Non-atomic admission	3,668 / 14,608	Stale status, expired commitment and context mismatch
Database-only restriction	3,192 / 12,536	Old context and permit commit after installed withdrawal
Reset on restart	5,036 / 19,116	Epoch rollback, lost pending work and repeated commitment
Untyped admission	4,776 / 20,904	An endorsement for one role grants another role

In the non-atomic variant, a ten-transition counterexample captures an episode, endorses and admits demonstration use, constructs context, issues a permit and caches a successful check. Authoritative withdrawal, delivery and local installation then occur before the cached decision commits. The final three events do not invalidate an already cached allow decision. This is

the check-to-use race excluded by A4. The same variant admits an expired permit in eight transitions and mismatched abstract context in ten.

The database-only variant blocks future admission but does not recheck the current status of an existing context and permit. Its shortest stale commitment requires nine transitions: capture, endorse, admit, construct, issue, withdraw, deliver, install and commit. The failure occurs without modifying the episode. It demonstrates why changing future retrieval cannot alone withdraw influence already carried into a plan.

The restart mutation resets local status, nonce consumption and the pending ledger. An epoch high-water failure occurs in four transitions; unresolved work is lost in seven; a repeated commitment occurs in ten. The untyped mutation produces an unauthorised cross-role grant in three transitions. Its otherwise protected final gate still checks the role’s endorsement, so this result establishes a grant-separation failure, not a physical effect through an unendorsed role. Both directions of cross-role promotion are reachable.

The protected graph contains 120 commit edges, so its result is not obtained by disabling execution altogether. Fresh admissions after reinstatement are possible at the new epoch. The study also includes deterministic representation assertions: adding one hundred faithful wrappers does not increase substantive evidence keys; adding a new analysis changes analytical coverage without adding an observation; and two observation origins can retain a common calibration dependency. These examples exercise the representation rules. They do not evaluate semantic duplicate detection or statistical independence.

6.3 Interpretation and limits

The finite result supports the internal consistency of the stated protocol fragment. It does not verify arbitrary fleet size, repeated crashes, unlimited epochs, alternative support witnesses, a complete dependency graph or concrete cryptographic and isolation mechanisms. Complete mediation, valid endorsements and physical-assurance approval are assumed. Delegation attenuation, issuer-policy changes, MBOM witness resolution, Robot BOM matching, configuration and actual-context composition coverage, revalidation decisions, credential parsing and hardware-backed status persistence are additional obligations not exercised by this state space. These state-space counts do not verify those mechanisms. No memory-corruption or side-channel attack is executed. The model does not prove liveness, recovery performance or a bound on harm.

In particular, the timing result in Section 5 is a conditional analytical result; the checker has no physical clock or network-delay model. Its withdrawal invariant starts at local installation, and a still-valid receiver may commit before installation. A production implementation would need refinement evidence connecting its buffers, durable transactions and command queues to the abstract transitions. The absence of a counterexample in this bounded model cannot substitute for that evidence.

7 Prospective evaluation of robot behaviour

7.1 Study tracks and controlled comparisons

The behavioural claims remain hypotheses. We specify three study tracks so that memory controls can be evaluated without conflating them with policy training or controller changes. Track A uses frozen memory, weights and retrieval settings to isolate integrity, eligibility, active-state withdrawal and final admission. Track B permits controlled additions to memory with fixed weights, testing local assessment and useful failed experience. Track C introduces a limited

heterogeneous fleet with changed tooling or payload, testing applicability, copied evidence and cross-recipient withdrawal. Each Track C configuration forms a separate stable assessment interval under A5; transfer between configurations must be assessed explicitly. Development and held-out evaluation remain separate in every track.

Principal comparisons use the same physical safety controller, task distribution, observation budget and compute budget. Otherwise a change in controller protection could be mistaken for a benefit of memory governance. A controller ablation belongs in simulation. First compare mechanisms under identical memory and retrieval settings; then conduct a separate end-to-end study in which each method may adapt its settings under the same development budget.

The baseline sequence comprises no external memory, ordinary retrieval, authenticated storage, current eligibility filtering, database-only restriction, memory-lifecycle gating and the complete protocol. Existing episode-diversity controls remain enabled where provided. A MAPLE-Guard adaptation must disclose its changed interfaces and receive the same observable metadata as competing deployable methods [6]. Diagnostic ground truth may be used only in a clearly labelled oracle comparison. Exact integration with Robota-Embed depends on the availability of its released artifacts; any substitute must be identified rather than presented as a reproduction [1].

Table 3: Prospective hypotheses and measurements. These experiments have not been performed for this article. Physical constraint outcomes are measured separately from task completion in all applicable comparisons.

Hypothesis	Controlled intervention	Primary outcomes
H1: useful failure evidence	Success-only memory; untyped episodes; typed views; typed views with claim withdrawal	Repeat failure, task completion, unsupported promotion and harmful imitation
H2: replica invariance	One, five, twenty and one hundred pure copies with unchanged evidence and authority	Support changes, duplicate context share, false admission and overhead
H3: observable common causes	Distinct records sharing a calibration or validator fault, with matched new-observation controls	False corroboration, affected recipients and valid evidence retained
H4: current reliance	Withdrawal after retrieval, planning and motion onset; compare future-read blocking and active-state controls	New stale-dependent admissions, useful work lost and reconstruction cost
H5: verified consumption	Periodic scanning; verification on use; verification with protected views	Tampered views consumed, prevented admission, delay and throughput

H1 includes successful segments within failed episodes and successful episodes containing violations or intervention. It tests whether typed views preserve useful information without increasing imitation of unsuitable behaviour. A role label alone is not counted as success. H2 adds genuinely new observations as a positive control, so a system that rejects all additions cannot appear to solve corroboration. Crops, summaries and new analyses form separate treatments because they can change analytical coverage. H3 exposes only the provenance available to the deployed defence and includes unknown or dishonest origin claims as a separate condition.

H4 distinguishes withdrawal of a completion label, a demonstration-applicability assessment and a shared calibration dependency. It compares valid observations retained with legitimate work unnecessarily invalidated. H5 includes transient substitution restored before the next periodic

scan, changes to index or retrieval configuration, replay of old accepted state and monitor overload. Each injected fault is specified by the capability granted to the attacker. Results for one capability do not establish detection of arbitrary zero-day exploitation.

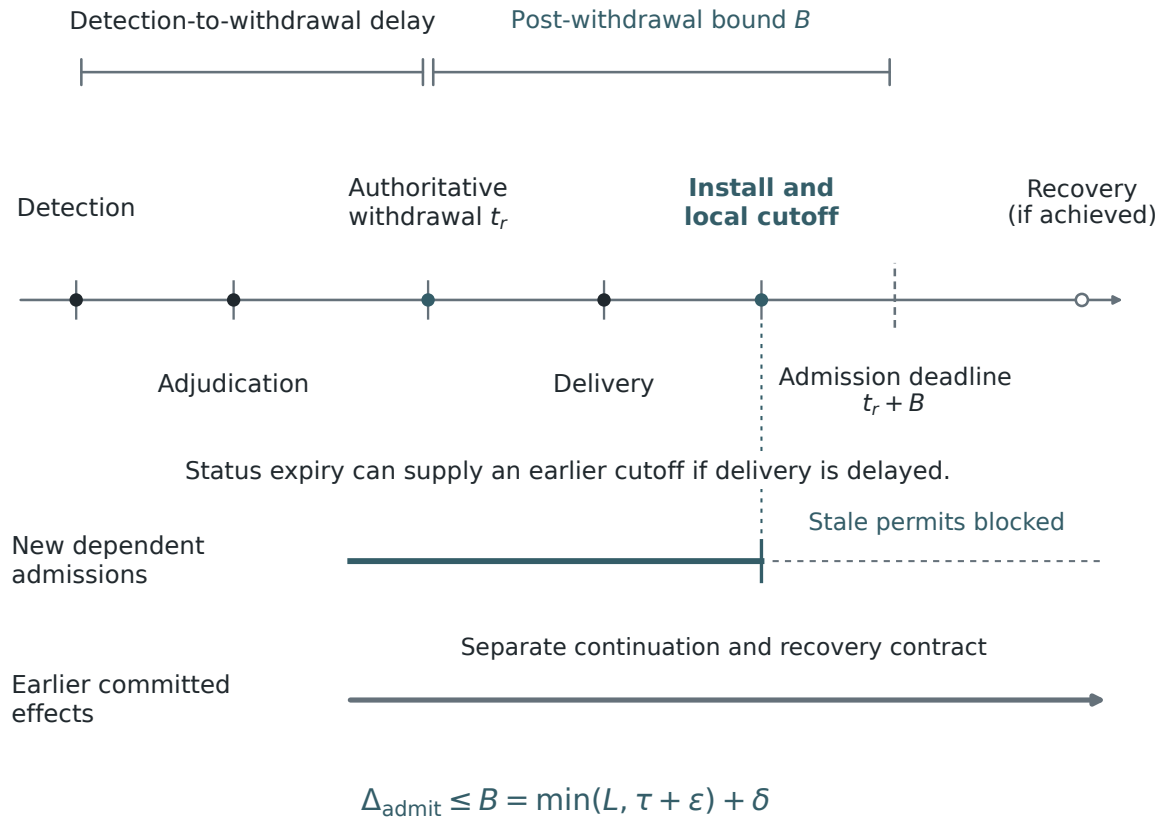
7.2 Statistical units and response measurements

Originating episodes or evidence groups form the sampling units. Copies, repeated retrievals and policy steps do not increase the independent sample size. Where trials share calibration, task instances or validation processes, uncertainty estimates must account for that dependence. Paired seeds and initial states are used where their correspondence is valid. Cluster definitions, exclusions and any hierarchical model are declared before analysis.

A confirmatory study must pre-register its primary outcomes, minimum useful effect, clean-task non-inferiority margin and sample-size rationale. Effect sizes and uncertainty intervals accompany aggregate success rates. Secondary comparisons are adjusted or explicitly exploratory. Constraint violations, rejected proposals and task completion are separate outcomes. Zero observed violations do not establish zero risk. Report all development, validation, reconstruction and failed-trial costs, not only the cost of successful evaluation runs.

Figure 3 separates the response stages. Record detection, adjudication, authoritative withdrawal, receiver delivery, local installation, the last stale admission and physical recovery independently. Report latency distributions and worst observed cases, together with the scheduling assumptions required for a hard bound. A measured percentile cannot establish that bound. Pending operations at withdrawal remain in the denominator until authenticated completion or cancellation closes them. Timeouts do not count as cancellations.

Physical studies should use guarded hardware and benign surrogate hazards. Simulation results, hardware decision-path tests and actual stopping or recovery measurements must be reported as different evidence levels. A deployable configuration needs an explicit account of which CPU, accelerator, remote-state and command paths are covered by the protected boundary.



Schematic sequence; spacing does not encode measured time.

Figure 3: The schematic separates detection-to-withdrawal delay from the stale-admission interval after authoritative withdrawal. Time before detection is not shown. Local installation blocks new affected admissions even when the bound permits a later worst-case deadline. Earlier commitments require a separate continuation and recovery contract; the open recovery marker is conditional. Positions are illustrative and do not encode measured durations.

7.3 Supply chain and authority evaluation

The enterprise-to-edge extension requires a separate protocol testbed. Hold memory content and its assessment constant while varying issuer recognition, parent-chain status, robot membership, delegation attenuation and request binding. Include valid but unauthorised signatures, a changed recipient, a role/resource correlation violation, a revoked parent with a live child and replay across sessions. A positive control admits a correctly scoped new delegation; a system that rejects every package does not satisfy the intended functionality.

Measure end-to-end status propagation under declared publication, cache and load policies. Inject partitions, reordered snapshots, conflicting publication sequences, power loss between checkpoint operations and restoration of an old valid storage image. Record whether post-restart admission remains disabled until the required reconciliation. Report protected-storage writes, cryptographic verification cost, admission latency, availability loss and the longest observed stale-admission interval. A measured maximum supports only the tested workload; a hard bound still requires the scheduling and clock assumptions in Section 5.

A further configuration study should compare memories and skill compositions inside and outside a declared update contract, including conflicting individually admissible memories and a changed observation/action mapping. Record justified reuse of prior validation, targeted new assessments, rejected transfers and required reconstruction. Hold each receiving baseline fixed during its trials and report any deliberate drift injection as an outside-A5 stress test. The present study reports no results for these cases.

Finally, compare poisoning by an authorised curator and collusion by recognised assessors with ordinary byte tampering. Keep observation budgets and available provenance matched. Distinct organisations, duplicated assessments and independently acquired measurements are separate treatments. Examine whether the independence policy detects declared common causes and whether undisclosed collusion remains outside its coverage. These tests evaluate authority enforcement and empirical reliability separately.

8 Implementation and discussion

8.1 A feasible implementation path

The prototype should resolve installed software, model and skill manifests against the Robot BOM, and bind controller interfaces, observation/action mappings and the memory harness to the same baseline. Configuration checks need both artifact evidence and the relevant physical calibration and health observations. A middleware node or ROS package version alone is insufficient. A skill marketplace is an optional distribution source, not a trusted admission service; its assessment reports enter the same receiver-side coverage and authority checks as other evidence.

A minimal prototype should instrument one manipulation harness end to end: candidate creation, claim assessment, MBOM construction, retrieval, context versions, delegation checks and dispatch. A small protected service performs deterministic schema, authority, scope, status and nonce checks. Learned assessors remain fallible and produce separately identifiable endorsements. Enterprise issuer wallets use cloud HSM-backed keys with issuance controls and separate privileges for capture, assessment, packaging and operational delegation. At the receiver, a machine wallet supplies device-bound key operations to the admission service; possession of a signing key is never treated as a general admission capability.

TPM 2.0 can provide protected keys, measured-state quotes and suitably provisioned non-volatile counter facilities [30]. The Device Identifier Composition Engine (DICE) provides a different foundation for deriving device identities and attestation evidence from layered measurements [31]. These mechanisms are not interchangeable, and neither continuously hashes all runtime memory. Attestation acceptance requires reference values, a freshness challenge, a binding to the relevant device or workload and an explicit verifier appraisal. Counter-backed rollback protection additionally requires a crash-consistent storage protocol. The implementation must state which keys, measurements, counters and storage paths its chosen platform actually protects.

For supported Jetson configurations, NVIDIA documents Open Portable Trusted Execution Environment (OP-TEE) trusted applications based on Arm TrustZone and, on specified products and releases, a firmware Trusted Platform Module (fTPM) [32,33]. A small admission verifier and wallet service could be implemented in an appropriately reviewed trusted application. That is a proposed integration, not a measured implementation. The selected hardware model, firmware, provisioning, trusted application and storage backend determine the boundary. Host TEE execution does not establish protection of graphics processing unit (GPU) buffers, DMA paths or remote model caches. Secure boot and dynamic random-access memory (DRAM) encryption address narrower threats and do not by themselves establish continuous application-memory integrity [34,35].

OpenID for Verifiable Presentations (OID4VP) 1.0 can carry requested credentials between a distribution-side holder and the robot-side verifier [36]. A robot-to-service presentation can be a separate exchange. Each verifier checks the negotiated format, request and response binding, nonce, intended audience, holder-key binding where required, issuer proofs and policy-recognised authority. OID4VP does not define the MBOM, delegation semantics or permission to execute. A headless robot profile must specify the supported transport and credential formats and be tested for interoperability. Package bytes retain their own authenticated commitments. Bitstring Status Lists can carry credential suspension or revocation status [29]; the admission service also verifies the authorised status issuer, relevant authority credentials, enforced absolute expiry and rollback state. Cache refresh guidance is not a hard freshness guarantee.

NVIDIA components can support the surrounding evaluation. NVIDIA documents hardware-error monitoring in Halos Core and virtual-machine isolation in its Linux-plus-QNX configuration on IGX Thor; Holoscan supports sensor-stream processing [37,38]. These capabilities do not establish the protected boundary of the proposed memory gate. Isaac Sim and Isaac Lab can host replay and scenario studies [39,40]. Cosmos-generated variations remain labelled as synthetic [41]. OSMO can run backend assessment workflows; recipient activation requires separate controls [42]. Halos Outside-In is documented for prototyping and evaluation, and its hardware-in-the-loop example tests decisions without acting on the simulated vehicle [43,44]. The NVIDIA Halos Safety Assurance Framework for Evaluation (SAFE) describes log-to-simulation consistency checks for autonomous driving; adaptation to robot-memory assessment remains proposed [45]. None of these tool names supplies an endorsement of an episode or a physical-safety claim.

The practical challenge remains complete state binding. Selective invalidation is acceptable only when dependency coverage is sound. Otherwise the implementation must reconstruct the affected session, including relevant device or remote state, from currently admissible inputs in a trusted runtime. If that is unavailable, the corresponding path falls outside the guarantee. Appendix C specifies the authority and status obligations; Section 7 specifies measurements needed to evaluate their latency, availability and behavioural costs.

8.2 What the results establish

The formal results establish conditional properties of representation commitments, role grants, recognised copying and operation admission. The finite study supplies reproducible interleaving evidence for a bounded protocol fragment and counterexamples to weakened enforcement. Neither establishes that an observation is true, that a validator is independent, or that a neural policy interprets a role as intended. A correct gateway can enforce a mistaken assessment. The separation of these claims is necessary to avoid treating governance as a substitute for physical validation.

The proposed behavioural benefits remain empirical hypotheses. Typed failure views may add no useful behaviour or may increase harmful imitation. Origin handling may add little beyond existing retrieval diversity. Conservative reconstruction may remove the latency advantage of memory. Most observed protection may come from the common physical controller. These outcomes would narrow the contribution and should be reported without changing denominators or excluding inconvenient failures.

Evidence freshness has three meanings that must remain distinct: the age of an observation, the validity period of an assessment and the freshness of its authoritative status. A new status response can confirm that an assessment remains in force; it does not establish that the physical environment has remained unchanged. Configuration changes and distribution shift may require new observations or reassessment even when all signatures and status checks pass.

8.3 Availability, privacy and governance limits

Offline operation requires a declared trade-off. Short status lifetimes limit stale admission but can prevent useful operation during a partition. Longer lifetimes improve availability while extending the interval in which remote withdrawal is unknown. Re-signing cached evidence cannot resolve this trade-off. A robot that needs continued operation must have a separately justified local envelope and recovery contract rather than silently renewing expired authority.

Preservation does not require indefinite retention or unrestricted sharing. Episodes can contain personal, commercially restricted or security-sensitive material. Purpose-limited packages, access control and retention rules apply to both raw observations and derived views. If evidence must be removed, dependent assessment and reproducibility claims must be reconsidered. A retained digest can identify a formerly referenced object but cannot replace the missing evidence or satisfy every audit need.

Finally, the trusted enforcement boundary remains a substantive assumption. Compromised keys, incomplete dependency recording, hidden state, dishonest lineage and unobserved common causes can defeat the relevant guarantee. Defining these limits is part of the method. The next research step is an implementation with measured coverage and the controlled behavioural study, followed by refinement evidence connecting concrete transactions to the transition model.

9 Conclusion

Robot memory requires explicit rules for how experience becomes evidence, how evidence permits reuse and how that reuse remains connected to a physical operation. The proposed protocol preserves failed episodes for appropriately assessed uses, treats recognised replication as evidentially idempotent and carries scoped support through consumed representations to final admission. Typed withdrawal restricts current reliance while preserving unrelated records and claims. The Memory Supply Chain Security profile makes enterprise identity, delegated scope

and current issuer authority explicit dependencies. The receiving Robot BOM and selected coverage assertion bind the decision to a particular skill, model, robot and memory combination.

Conditional proofs and a reproducible finite study support the specified admission and withdrawal logic. They do not establish physical reliability, concrete implementation security or the validity of an assessor’s judgment. Revalidation is therefore an evidence-based change decision: existing coverage may suffice, while uncovered combinations or changed premises require additional assessment. The behavioural value, implementation cost and deployment limits remain questions for the proposed evaluation.

9.1 Guideline recommendation

We recommend developing an engineering guideline for controlled memory use in physical AI. Its unit of approval should be an assessed use of memory within an identified robot configuration and operating envelope. The guideline should require:

1. **Traceable configuration and evidence.** Maintain linked Robot BOM and MBOM records covering the installed hardware, software, models, skills, memory representations and relevant calibration. Record uncertainty and missing dependencies.
2. **Claim-specific permission.** Distinguish observations, assessments, reuse roles and execution authority. Preserve useful failed experience without promoting unsupported demonstrations. Prevent recognised copies from increasing empirical support; justify any independence requirement separately.
3. **Proportionate revalidation.** Record whether existing evidence covers the intended memory, context composition and use. Require targeted assessment or broader validation when it does not. Stage candidates before activation and restrict affected use before a planned configuration change.
4. **Protected admission.** Verify issuer authority, attenuating delegation, current status and actual configuration. Bind the consumed representation, retained state, selected witness and operation to the same decision. Keep physical assurance as a separate requirement.
5. **Withdrawal and recovery.** Commit nonce consumption and pending-operation records durably before command release. Specify inherited expiry, offline limits, restart reconciliation and a justified continuation or recovery contract. Measure detection, authoritative withdrawal, local enforcement and physical response separately.
6. **Auditable evaluation.** Retain acceptance and rejection reasons, coverage decisions and change history in the evidence graph. Test negative cases alongside valid-use controls and report availability costs. Distinguish finite-model checks, implementation tests and physical experiments.

These recommendations define a research and implementation agenda, not a validated standard or certification scheme. Adoption should depend on evidence that the concrete enforcement boundary satisfies its assumptions and that the assessed robot configuration meets its physical requirements.

A Physical-assurance interface

The memory protocol does not require a numerical harm predictor. A gateway may use a separately justified reachable-set contract,

$$\text{Reach}(X_t, u, \mathcal{W}) \subseteq \mathcal{S}_{\text{safe}}, \quad (11)$$

where X_t is a state-uncertainty set, $\mathcal{W}$ the specified disturbances and $\mathcal{S}_{\text{safe}}$ an assessed safe region. The contract must cover the command interval and relevant continuation or recovery. Its validity depends on the observation and dynamics assumptions. Memory eligibility supplies none of those assumptions automatically.

Alternatively, fix a finite horizon T , let H_k be the first specified harm in decision interval k , and let $\mathcal{F}_{k-}$ contain the available history and selected control. If justified $\mathcal{F}_{k-}$ -measurable bounds satisfy almost surely

$$\Pr(H_k \mid \mathcal{F}_{k-}) \leq \bar{p}_k, \quad \sum_{k=1}^T \bar{p}_k \leq \varepsilon, \quad (12)$$

then conditional expectation and the union bound give

$$\Pr\left(\bigcup_{k=1}^T H_k\right) \leq \mathbb{E}\left[\sum_{k=1}^T \bar{p}_k\right] \leq \varepsilon. \quad (13)$$

Independence between intervals is unnecessary. The horizon, harm event, delayed-effect coverage and adaptive-selection assumptions must be specified. A calibrated estimator under deployment shift is a separate research obligation. A risk ledger persists across session replacement, and successful execution does not refund cumulative expenditure. No such estimator is implemented or validated in this article; the equations state an optional interface obligation, not an empirical risk claim.

B Artifact record and declarations

The supplement contains the finite-model source, its complete generated outputs, the manuscript and figure sources, and the proposed memory-credential schemas, illustrative unsigned fixtures and profile checks. It also records the exact abstraction bounds and reproduction commands. The protocol results remain the finite-state exploration and deterministic representation assertions in Section 6. The supplementary credential-profile checks exercise message structure and specified cross-field conditions; they do not verify cryptographic proofs, hardware attestation, interoperability or delegation-chain enforcement. No physical-robot experiment, user study or deployment assessment is reported. The prospective evaluation in Section 7 has not been executed.

The named author is affiliated with Spherity, whose research supplies part of the architectural background. The argument and comparison distinguish that prior research from the new protocol composition. Artificial intelligence tools were used for manuscript development, source review, code drafting and consistency checks. The supplied code and explicit assumptions permit independent examination; that assistance is not an independent experimental replication or external peer review.

C Memory Supply Chain, Enterprise Identities, and Verifiable Attestation Profiles

C.1 Scope and relation to software supply chain security

Memory Supply Chain Security governs the path from capture and transformation to assessment, distribution, receiver admission and withdrawal. The protected object is a particular representation and its permitted reliance, not simply a downloadable file. The proposed profile composes existing identity and attestation mechanisms with the evidence roles and operation-bound enforcement defined in this paper. It does not claim a new cryptographic primitive, a standardised Memory Bill of Materials (MBOM) format or a software-supply-chain certification level for robot memory.

Supply-chain Levels for Software Artifacts (SLSA) 1.2 supplies a useful separation between source, build and verification obligations [16]. Applied here, the source boundary is the accepted capture and assessment input; the build boundary is the curator’s transformation environment; and the consumer boundary is the robot’s protected gate. This is an architectural analogy. Software provenance requirements cannot establish whether a force sensor was correctly calibrated or a grasp was physically successful. In-toto supplies a model of authorised supply-chain steps and attributable input/output records [17]. The memory profile applies that model to segmentation, labelling, embedding, assessment and packaging, with explicit step types and authorised functionaries. A producer’s statement that a step occurred remains subject to the assurance of its execution environment.

Sigstore can supply artifact signatures and verification material relating an expected signing identity to a recorded signing event [18]. It is an optional mechanism for software, model or transformation artifacts, not a substitute for the operator’s current delegation policy. A historically valid signing event does not establish present permission to consume a memory. An enterprise decentralised identifier (DID) and a Sigstore identity are not automatically equivalent: a policy-recognised binding is required when a claim crosses those identity systems. Transparency can make inconsistent publication detectable to appropriate monitors; an inclusion proof alone does not guarantee globally consistent views or correct content. Sensitive traces and enterprise identifiers need not be published in a public log.

System Package Data Exchange (SPDX) and CycloneDX describe software, artificial intelligence (AI) and machine-learning components and associated metadata [19,20]. An MBOM references such inventories by immutable identifier while adding episode origins, measurement conditions, selected assessment premises, authority paths and status dependencies. Inventory completeness is an explicit implementation obligation. Recording a model-weight digest does not prove that those weights ran, and a measured environment does not prove that its output is scientifically valid. Identity authentication and chain-of-custody attestations establish attributable provenance; valid policy-rooted delegation additionally establishes scoped authority. Neither establishes factual truth or physical safety.

C.2 Supply chain threat model

The adversary may control a producer, storage location, distribution channel or a subset of credential holders; compromise an unprotected ingestion or inference process; replay previously valid objects; or exploit a vulnerability outside the protected boundary. The adversary may also submit authentic but misleading measurements and collude with an assessor. The integrity and admission guarantees remain conditional on A1-A4. Compromise of a protected gate or its effective trust roots violates the relevant enforcement assumption. Collusion across the required

assessment paths can defeat the evidential basis of an admission even while those protocol invariants hold; evidence validity and protocol enforcement are separate claims. Detection delay before authoritative withdrawal is not included in Δ_{admit} .

Poisoned ingestion. Malicious observations, instructions embedded in recordings, fabricated outcome labels or an altered transformation can enter through an authorised writer. Authentication alone leaves this vector open. The profile isolates candidate material, restricts parsers and transformation privileges, binds outputs to exact input and environment commitments, and requires claim-specific assessment before role promotion. Limits on origin contribution and release cohorts constrain amplification. Where available, independently instrumented capture or replay can challenge an assertion. A reproducible malicious transformation remains malicious; reproducibility establishes repeatability rather than truth. Missing sensor samples and unexplained gaps are recorded rather than silently filled with inferred observations.

Stale or defective calibration. A certificate can refer to the wrong sensor, calibration revision, tool geometry, environmental range or time interval. The receiver checks the sensor-to-device binding, method, uncertainty, unit conventions, calibration interval and envelope against the claim. For a historical observation, applicability is evaluated at capture time, including clock uncertainty. Mere later expiry need not invalidate that observation. A later defect finding may withdraw support for measurements made during an identified interval. Current actuation or measurement requires calibration applicable now. This distinction prevents both unjustified reuse and unnecessary destruction of historical evidence. Genuine drift outside a declared envelope can remain undetected and requires health monitoring or recalibration.

Identity spoofing and delegated-authority abuse. An attacker may impersonate an enterprise, replace DID resolution results, steal a workload credential, substitute a robot key or reuse a delegation issued for a different fleet. Verification requires authenticated method-specific DID resolution, a recognised enterprise binding, key-purpose checks, appropriate key history and current compromise information. Every delegation binds its issuer, subject, parent, resource scope, role, receiver rule and time limits. The receiver checks attenuation, fleet membership, proof of possession and replay binding. Locally provisioned authority roots cannot be replaced by a credential inside the incoming package. Device enrollment must bind the operational key to the assessed hardware identity; a self-asserted DID does not supply that binding.

Colluding assessor. A recognised assessor can sign an unjustified endorsement, or several nominally separate assessors can share control, data or a faulty model. Signatures expose accountability but cannot prevent a truthful-looking false judgment. Where the claim demands it, the acceptance policy requires separately justified organisational and technical diversity, independently acquired observations, challenge trials or a qualified external review. The graph records shared ownership, calibration and model dependencies to the extent known. Threshold approval is a governance rule, not a statistical independence result. One compromised assessor may be tolerated only under a stated threshold and threat assumption; collusion across the required independent paths defeats that assumption.

Replay, substitution and withdrawal suppression. Old packages, status lists, fleet memberships or permits can remain cryptographically valid while no longer being eligible. Absolute expiry, protected per-authority high-water state, request nonces and final dependency checks address different parts of this threat. Pinned representations and state-version binding address substitution after verification. Expiry must be enforced during partitions, and denied status cannot be overwritten by a lower or conflicting epoch. Compromised issuing or status authorities may continue producing fresh false affirmations; the timing theorem then requires withdrawal by a still-trusted superior authority or another independent restriction mechanism. It cannot be rescued by faster polling of the compromised issuer.

Resource exhaustion and disclosure. Large graphs, credential chains, compressed status lists and remote references can exhaust a verifier or disclose sensitive relationships. The profile limits graph size, chain depth, decompressed length, verification work and fetch destinations. It pins schema and context versions, applies access control to traces, and avoids treating arbitrary credential URLs as instructions to access internal services. Unknown mandatory terms or unsupported proof algorithms cause rejection. Status caching and aggregate retrieval can reduce tracking, but package identifiers, selective disclosures and transparency records still require a privacy analysis. Frequent status changes can also expose operational events to parties that know the relevant status-list index.

C.3 Verifiable credential data profiles

C.3.1 Common envelope and verification obligations

The supplement defines three proposed application schemas using JavaScript Object Notation (JSON) Schema Draft 2020-12: `EpisodicMemoryAttestation`, `CalibrationAssertion` and `RoleEndorsementCredential`. They use the Verifiable Credentials (VC) Data Model 2.0 envelope [8] and the `JsonSchema` mechanism described by the VC JSON Schema Candidate Recommendation Draft [46]. The memory-specific types and fields are not World Wide Web Consortium (W3C)-standardised vocabulary. Each schema restricts one compacted JSON representation and includes an explicit illustrative application context. The profile uses a versioned namespace; the supplied `example.org` and `did:example` identifiers are non-production examples.

Each envelope requires `@context`, `id`, `type`, `issuer`, `validFrom`, `validUntil`, `credentialSchema`, `credentialStatus`, `authorityReferences` and `credentialSubject`. The first context identifies VC 2.0; the type array contains `VerifiableCredential` and the specific profile type. The schema reference identifies the exact application schema. The status entry names its purpose, decimal-string index and status-list credential. Requiring both validity endpoints, one subject and one status entry is a restriction of this profile. `authorityReferences` identifies separately evaluated authority evidence but does not itself prove the referenced authority. The securing mechanism must bind this field together with the rest of the credential payload.

Resource references carry an identifier, media type, digest algorithm, digest value and digest scope. The example profile uses SHA-256, a Secure Hash Algorithm with a 256-bit digest, over exact raw octets, including encoding and line endings; it does not hash an ambiguously reserialised JSON object. Deployments pin the permitted algorithms, schema/context bytes and decoding rules. Intervals have explicit endpoints; quantities and units use identifiers; numerical envelopes include bounds. Structural validation is followed by chronological, unit, range, identity, dependency and status checks. The latter cannot be inferred from a well-formed identifier or digest. Expired credentials may remain historical evidence under the rules of Section 5; they are not accepted as live authority.

The supplied fixtures are unsigned payloads intended for an external securing envelope. They contain no purportedly valid signature or hardware quote. A deployment must select and verify a supported VC securing mechanism and bind its payload to the issuer's recognised key and authority. JSON validity, JavaScript Object Notation for Linked Data (JSON-LD) interpretation, cryptographic proof verification, attestation appraisal and receiver admission are separate stages. The included structural examples do not establish interoperability of the whole stack.

C.3.2 EpisodicMemoryAttestation

The subject has type `EpisodicMemory` and identifies the episode, robot and capture service. Mandatory fields include `captureInterval`, `clock`, `configuration`, `environmentContext`, `traceManifest`, `sensorRecords`, `taskOutcome`, `constraintOutcome`, `measurementQuality`, `interventions`, `hardwareAttestation` and `provenance`. Each sensor record identifies the sensor, channel, measured quantity, unit, measurement interval, sample count, raw-trace resource and calibration-credential reference. The trace manifest also distinguishes issued commands from measured feedback. Task outcome admits success, failure, unknown and disputed; constraint outcome and measurement quality have separate vocabularies. No rule equates successful completion with valid measurement or suitable demonstration.

The hardware object references raw attestation evidence, its claimed measurement time and challenge binding, together with a separately identified, authenticated attestation result. The result identifies its verifier, evidence digest, appraisal policy, time bounds, claimed coverage and limitations. The Remote Attestation Procedures (RATS) architecture distinguishes evidence produced by an attester, appraisal by a verifier, and attestation results used by a relying party [47]. The receiver authenticates the referenced result, checks its agreement with the copied metadata, and applies its own acceptance policy. Merely placing a verifier's DID in a capture credential proves nothing about that verifier's involvement. A Trusted Platform Module (TPM) quote binding a caller-supplied trace digest does not prove that a protected sensor path produced the trace. The capture process and its sensor-to-record binding need their own coverage argument. Inconclusive attestation remains representable as evidence, but cannot satisfy a policy that requires an accepted appraisal.

C.3.3 CalibrationAssertion

The calibration subject has type `Calibration` and identifies `calibratedItemId` and `itemKind` (sensor or tool), `itemConfiguration`, physical quantity and unit. Timing is recorded in `calibratedAt` and `calibrationValidity`. The method and report are identified by `calibrationMethod` and `calibrationReport`; `calibrationResult` states the assessment outcome. The subject also requires `measurementRange` and `operatingEnvelope`, together with `expandedUncertainty` and `traceabilityReferences`. The uncertainty object identifies its value, unit and coverage factor; the report provides the method and interpretation needed to assess that assertion. The item and configuration references bind tool geometry or sensor configuration rather than relying on a generic model number.

The receiver checks the credential's authenticity and issuer competence separately from whether the calibration applies to the observation or present operation. The relevant time interval, actual item, configuration, quantity, units and measured operating conditions must match. If clock uncertainty overlaps an invalid interval, the receiver requires an explicit assessment rather than silently treating the timestamp as exact. A calibration result of rejected or inconclusive is valid assessment data, but cannot satisfy a positive calibration requirement. Later routine expiry does not retrospectively change a trace; a defect withdrawal covering its capture interval can invalidate the dependent interpretation. Traceability references are assessable claims, not a guarantee of laboratory competence.

C.3.4 RoleEndorsementCredential

The subject has type `RoleEndorsement` and requires `claim`, `role`, `assessmentResult`, `assessment`, `applicability` and `dependencies`. The claim binds its identifier, episode,

episodic-attestation reference and exact segment with start and end offsets. A free-text statement describes the claim; the gate uses pinned, machine-interpretable identifiers and scope rules for authority decisions. The role vocabulary is **demonstration**, **counterexample** or **recovery**. The result is independently **accepted**, **rejected** or **inconclusive**. A failed episode may therefore receive an accepted counterexample endorsement, a rejected whole-episode demonstration endorsement, and an accepted demonstration endorsement for a separately assessed successful segment.

Assessment fields identify time, method, supporting resources, known common causes, assumptions and limitations. Applicability identifies the receiver class, tasks, configuration references and operating envelope. In the Robot Bill of Materials (Robot BOM) profile, these references resolve to source and receiving baselines and the applicable coverage assertion or memory-update contract. They must cover the intended skill/model/robot combination and memory representation; a generic product-family label is insufficient. Dependencies preserve observation and analysis origins and materialisation or assurance references. An empty list of known common causes means none are declared, not that independence has been established. A new issuer signature or copy does not add an observation origin. The recipient must assess any independence requirement using the declared evidence and unresolved dependencies.

The credential records a role assessment, not a receiver grant. Even an accepted result requires the operator’s valid consumption delegation, current selected support and local admission. A rejected or inconclusive result may be retained for audit without authorising the proposed role. Separate credentials and status references permit claim-specific withdrawal; the MBOM carries the dependencies needed to propagate that restriction. The three evidence schemas deliberately do not encode a universal physical-safety certificate or replace the delegation semantics of Section 5.

C.4 Delegation and verification exchange

The following exchange defines the proposed application semantics. Fleet Package Distribution is a transport and credential-holder service; the Robot Admission Gate is the receiver-side verifier. The operator, curator, calibration provider and assessors remain distinct credential issuers even when their credentials are presented together. OpenID for Verifiable Presentations (OID4VP) can carry presentations, but the package transfer, device attestation, delegation language and local permit protocol are separately profiled.

1. **Provision recognised roots and enroll the device.** The operator installs authenticated policy and domain-limited issuer roots through the device-management process. Enrollment binds the robot’s operational DID/key to its hardware identity, assessed software configuration and fleet membership. The hardware verifier appraises evidence against reference values and records the covered boundary [47]. The membership credential alone cannot prove that a key is hardware protected.
2. **Offer an immutable package.** Distribution provides the package identifier, root digest, MBOM profile and version, applicable role candidates, required credential formats and disclosure policy. The gate checks resource limits and whether its delegated envelope permits considering that resource. Merely receiving the offer has no effect on active context.
3. **Bind the request to both parties.** The gate creates a fresh nonce and an authenticated request for the required credentials. Protected session state binds the nonce to the intended verifier, package root, requested use, recipient key and profile. If distribution must authenticate the robot, a separate robot-to-service presentation or mutually authenticated

channel proves its enrolled identity. The profile binds that key and any required device-attestation exchange to the same session. A challenge copied into untrusted metadata is insufficient.

4. **Present credentials and artifacts.** The distribution-side holder returns a holder-bound presentation of service credentials whose binding it can satisfy under the negotiated format. It separately transfers issuer-signed capture, calibration and role artifacts, the operator delegation chain, membership evidence where required, the MBOM and package commitments, with accepted status snapshots or references. Its presentation proof does not satisfy a credential bound to another entity's key. Robot-bound credentials requiring proof of possession are presented in the separate robot exchange. Supporting evidence retains its original issuer proof and is not promoted into holder-bound identity evidence merely by inclusion in the package.
5. **Verify provenance and authority separately.** The gate checks request binding, selected proof suites, issuer signatures, DID/key history, status-signer authority and all live credential statuses. It requires the authenticated package root to equal the root recorded with the nonce and requested use, then consumes or idempotently closes that presentation session to prevent reuse for a different decision. It checks the delegation from a local root to the exact robot or verified fleet member, including scope inclusion, parent links, expiry and delegation depth. It verifies artifact and graph commitments, transformation identities and required hardware-appraisal results. Status of a non-revoked child cannot repair a withdrawn parent. Unknown mandatory semantics or an unsupported chain cause rejection.
6. **Assess use and select support.** The gate evaluates claim support, historical calibration applicability, current operating conditions and role requirements. It checks the installed Robot BOM baseline, the proposed context composition and coverage under Section 4.5; uncovered use remains inactive until the required reassessment supplies accepted support. It records W_{ev} and W_{auth} , including shared or unresolved causes and any accepted alternatives. A valid package signature cannot discharge this step. A counterexample role may be accepted for a failed task without accepting its action sequence as a demonstration.
7. **Install a receiver-specific admission.** In protected state the gate records the package root, representation commitment, role, scope, witness, policy version, relevant epochs and earliest inherited expiry. It produces an authenticated acceptance or rejection receipt with reason codes and decision identifiers. Activation then binds a sealed representation to the actual consumption path. Receipts may use access-controlled commitments rather than disclose raw traces. Acceptance is not a physical-operation permit.
8. **Mediate effects and subsequent changes.** Planning carries the selected dependencies into equation (5). The execution gateway applies equation (6) at the last reversible boundary. Withdrawal or expiry invalidates affected new admissions and unused permits; earlier commitments remain in the pending-effect ledger. A fresh witness, renewed authority or corrected package requires a new decision and permit. Subsequent local experience starts again as candidate evidence, preserving its relation to the original episode without multiplying its observation count.

This exchange prevents a valid presentation from being treated as blanket fleet trust. Its security still depends on the concrete authenticated channel, proof formats, key bindings, local isolation and complete mediation. Interoperability requires published test vectors and negative tests; naming OID4VP does not establish conformance of the proposed robot profile.

C.5 Status distribution and bounded stale admission

Bitstring Status Lists provide the credential-status representation. The following ordering and enforcement rules are additions of this memory profile. A status publisher issues an authenticated envelope binding its authority, namespace, generation, publication sequence, authoritative issue time, absolute expiry and digest of the complete list snapshot. Per-claim or per-permission operational restrictions have scoped epochs. A separately authenticated delta identifies its base snapshot and resulting commitment; it is not represented as a standard Bitstring Status List credential unless it satisfies that standard’s complete representation requirements.

The receiver verifies the status signer’s authority, all relevant purposes, the index and list binding, accepted time bounds and decompression limits. It rejects a lower publication sequence and rejects conflicting contents at an already accepted sequence. A delta with a missing base requires a valid complete snapshot before affirmative admission resumes. A higher publication sequence alone need not invalidate unrelated permits: permits bind their selected dependency epochs. Authority rotation and failover require an authenticated continuity transition; changing a DID or namespace cannot reset an established high-water mark or clear a restriction.

Credential revocation and suspension have different semantics [29]. This profile never clears a revocation of the same credential. Renewed operational permission after revocation requires a new credential and an advancing local epoch. A suspension may be lifted by its recognised authority, also with a new epoch. Neither transition revives an old operation permit. Status of an issuer’s authority credential, status of its issued assessment and status of the selected memory use are distinct dependencies. Local policy withdrawal can disable reliance even when a remote credential remains unrevoked.

For high-frequency use, the receiver verifies status publications outside the operation path and installs them into protected state. Every admission checks local scoped status, inherited expiry and epoch bindings. A remote exchange or hardware-counter write on every control cycle is unnecessary. Scheduling and resource reservations must nevertheless bound installation and expiry processing under the claimed workload. Push notification can reduce delay; periodic fetches and bounded cache leases provide recovery from lost notifications. Advisory list time-to-live (`ttl`) or Hypertext Transfer Protocol (HTTP) caching rules do not supply enforced expiry.

A hardware monotonic counter can anchor an authenticated checkpoint containing publication high-water marks, scoped epochs, restrictions, policy version and replay state. The counter establishes order, not the age of remote information. Counter protection against reset or redefinition, crash-consistent association with the checkpoint, storage durability and write capacity are platform obligations. Batching checkpoints is permissible only if a restart cannot enable admission from a state that omits an installed restriction. Where that cannot be guaranteed, ordinary admission remains disabled after restart until fresh authoritative synchronisation and pending-operation reconciliation complete. Once consumption and unresolved effects must still meet A4; restoring a signed old file is insufficient.

This distinction matters on Jetson. The cited r38.4 documentation uses normal-world filesystem secure storage by default and lists Replay Protected Memory Block (RPMB) support for AGX Orin, with explicit exclusions for AGX Thor, Orin NX and Orin Nano [48]. Open Portable Trusted Execution Environment (OP-TEE) documents the absence of rollback protection when its RPMB filesystem support is disabled [49]. Consequently, neither “secure storage” nor firmware Trusted Platform Module (fTPM) presence establishes the required persistence guarantee for every device. A deployment must demonstrate its chosen implementation or adopt the restrictive restart rule above.

For a dependency withdrawn at t_r , let L bound delivery from the authoritative decision to the receiver’s enforcement service, including publication, caching and transport delay. Let each affirmative statement have an absolute lifetime of at most τ from authoritative issuance. No authority in the selected path may issue a new affirmative statement for that withdrawn scope after t_r until authorised reinstatement. Permits, proxies and cache refreshes inherit rather than extend expiry. Protected time has disagreement at most ϵ , and installation or expiry rejection takes at most δ . These assumptions give equation (9). For different dependencies, use their individual bounds or conservative constants covering every relevant source.

During a partition, L can be unbounded while expiry still provides a finite cutoff. Without trustworthy elapsed time across suspension or restart, expiry cannot provide that route until time is re-established. A hardware counter does not fix this absence. Without either bounded delivery or enforced expiry, no finite Δ_{admit} follows. The bound starts at authoritative withdrawal, excludes prior detection and adjudication, and limits new admissions rather than already committed motion. After local installation, Theorem 5 applies immediately in the local serialisation order; the remote bound is not permission to continue using the withdrawn dependency.

C.6 Evaluation and assurance boundary

The proposed integration requires separate validation of credential parsing, signature and presentation verification, attenuation, status persistence and runtime enforcement. Negative cases include a valid foreign issuer; a capture-only issuer endorsing demonstration; a child broadening role, fleet, resource or expiry; an unrevoked child with a revoked parent; a swapped package root; a replayed presentation; stale status after restart; and a conflicting status sequence. Matched behavioural trials distinguish authorised poison, common calibration faults and colluding assessors from unauthorised byte substitution. Measure admission and rejection correctness, verification cost, state-reconstruction cost, status-installation delay and useful operation lost during partitions.

The finite results in Section 6 do not exercise these concrete mechanisms. The accompanying credential schemas and fixtures specify proposed message structures and cross-field checks; they are not cryptographic test vectors or evidence of robot safety. A credible implementation study must connect the protected code, key provisioning, storage and command mediation to A1-A4, then evaluate the additional hypotheses in Section 7. Even perfect enforcement establishes only that the selected authority and assessment policy was followed. It cannot turn a collusive endorsement, false sensor reading or unsupported physical model into reliable evidence.

References

Primary sources and mutable documentation reviewed 18-20 October 2026. Preprint and specification versions are identified where available. The memory-specific schemas, delegation semantics and status-enforcement profile are proposals of this manuscript, not guarantees supplied by the cited standards.

- [1] Y. Liu, J. Wang, S. Wu, et al., “Robota-Embed: Unified VLA Embeddings for Robot In-Context Learning and Evolution,” arXiv:2610.19495v1, 2026. Full text. Project and artifact status.
- [2] B. Sima, L. Wang, X. Lu, K. He and X. Yang, “Self-Evolving Just-In-Time Memory for Proactive Embodied Safety,” arXiv:2607.16247v1, 2026. Full text.
- [3] Z. Chen, Z. Xiang, C. Xiao, D. Song and B. Li, “AgentPoison: Red-teaming LLM Agents via Poisoning Memory or Knowledge Bases,” arXiv:2407.12784v1, 2024. Full text.
- [4] S. Dong, S. Xu, P. He, Y. Li, J. Tang, T. Liu, H. Liu and Z. Xiang, “Memory Injection Attacks on LLM Agents via Query-Only Interaction,” arXiv:2503.03704v5, 2026; first submitted 2025. Full text.
- [5] S. S. Srivastava and H. He, “MemoryGraft: Persistent Compromise of LLM Agents via Poisoned Experience Retrieval,” arXiv:2512.16962v1, 2025. Full text.
- [6] W. Xiong, Y. Zhou, J. Wang, S. Gu, B. Tang, Z. Li, F. Xiong, Y. Wen and M. Wen, “MAPLE-Guard: Memory-Aware Link Enforcement Against Memory-Link Poisoning in Multi-Agent Systems,” arXiv:2608.00426v2, 2026. Full text.
- [7] T. Lebo, S. Sahoo and D. McGuinness, eds., “PROV-O: The PROV Ontology,” W3C Recommendation, 30 April 2013. Specification.
- [8] W3C, “Verifiable Credentials Data Model v2.0,” W3C Recommendation, 2025. Specification.
- [9] C. Stöcker, “Evidence Graphs for Industrial AI: Data-plane architectures for global supply chains, Industry 4.0, critical infrastructure, and B2G data exchange from a German perspective,” Spherity Research, 2026. Paper.
- [10] C. Stöcker, “Verifiable Safety Assurance Evidence Graphs for Modular Humanoid Robot Arms: A vendor-neutral architecture for functional safety, cyber resilience, lifecycle assurance and use-case-specific compliance,” Spherity Research, 2026. PACE paper.
- [11] C. Stöcker, “Cascading Trust Failures in AI Agent Networks: A Trust Architecture for Scalable Revocation and Bounded Containment,” Spherity Research, 2026. Paper.
- [12] C. Stöcker, “Runtime Revocation in Multi-Agent Systems: Protocol Verification and Evaluation of Privacy and Containment,” Spherity Research, 2026. Paper.
- [13] C. Stöcker, “European Business Wallets as the Legal Control Plane for Zero Trust AI Agents,” Spherity Research, 1 June 2026. Paper.
- [14] D. T. Phan, R. Grosu, N. Jansen, N. Paoletti, S. A. Smolka and S. D. Stoller, “Neural Simplex Architecture,” NASA Formal Methods Symposium, 2020; arXiv:1908.00528v2. Full text.
- [15] C. Stöcker, “Risk-Bounded Runtime Assurance for Dynamic Multi-Agent Systems under Partial Observation: AI Safety - Verifiable Identity and Evidence for Bounded Cross-Organisation Autonomy,” Spherity Research, 2026. Paper.

- [16] OpenSSF, “SLSA Specification, version 1.2,” Approved specification. Specification. Build provenance. Threats and mitigations.
- [17] in-toto Project, “in-toto Specification, version 1.0.0,” 2 June 2023. Versioned specification. Specification index.
- [18] Sigstore Project, “Security Model” and “Verifying Signatures,” project documentation. Security model. Verification.
- [19] OWASP CycloneDX, “Machine Learning Bill of Materials (AI/ML-BOM),” capability documentation. Documentation.
- [20] SPDX Project, “SPDX Specification 3.0.1: AIPackage,” versioned specification. Specification. Scope and terminology.
- [21] W3C, “Decentralized Identifiers (DIDs) v1.0,” Recommendation, 19 July 2022. Dated specification.
- [22] ROS 2 Project, “ROS 2 middleware interface,” design documentation. Design article.
- [23] ROS Project, “urdfdom: URDF parser,” project documentation and model representation. Repository. URDF package overview.
- [24] NVIDIA, “Deploying Policies in Isaac Sim,” Isaac Sim documentation, version 6.0.0. Deployment guide.
- [25] NIST, “Adversarial Machine Learning: A Taxonomy and Terminology of Attacks and Mitigations,” NIST AI 100-2e2025, 2025. Report.
- [26] Linux Kernel Project, “fs-verity: Read-only file-based authenticity protection,” kernel documentation. Documentation.
- [27] NIST, “Side-Channel Attack,” Computer Security Resource Center glossary. Definition.
- [28] A. Regenscheid, “Platform Firmware Resiliency Guidelines,” NIST Special Publication 800-193, 2018. Publication.
- [29] W3C, “Bitstring Status List v1.0,” Recommendation, 15 May 2025. Dated specification.
- [30] Trusted Computing Group, “Trusted Platform Module Library, Part 3: Commands,” Family 2.0, Level 00, revision 01.84, 20 March 2025. Specification.
- [31] Trusted Computing Group, “DICE Layering Architecture,” version 1.0, revision 0.19, 23 July 2020. Specification.
- [32] NVIDIA, “OP-TEE: Open Portable Trusted Execution Environment,” Jetson Linux Developer Guide, release 38.4. Documentation.
- [33] NVIDIA, “Firmware TPM,” Jetson Linux Developer Guide, release 38.4. Documentation.
- [34] NVIDIA, “Memory Encryption,” Jetson Linux Developer Guide, release 38.2. Documentation.
- [35] NVIDIA, “Secure Boot,” Jetson Linux Developer Guide, release 36.4.4. Documentation.
- [36] OpenID Foundation, “OpenID for Verifiable Presentations 1.0,” Final specification, 9 July 2025. Specification.

- [37] NVIDIA, “Inside NVIDIA Halos for Robotics: A Full-Stack Functional Safety System for Physical AI,” NVIDIA Technical Blog, 22 June 2026. Technical article.
- [38] NVIDIA, “Holoscan SDK Overview,” SDK documentation. Documentation.
- [39] NVIDIA, “Isaac Sim,” developer documentation. Overview.
- [40] NVIDIA, “Isaac Lab,” project documentation. Documentation.
- [41] NVIDIA, “Cosmos Introduction,” platform documentation. Documentation.
- [42] NVIDIA, “OSMO,” platform overview and FAQ. Overview.
- [43] NVIDIA, “Halos Outside-In Safety OSS Safety Notice,” project repository. Safety notice.
- [44] NVIDIA, “Hardware-in-the-Loop Testing,” Halos Outside-In documentation. Documentation.
- [45] NVIDIA, “NVIDIA Halos Safety Assurance Framework for Evaluation (SAFE),” technical brief, “Introduction” and “Our Architecture.” Introduction. Architecture.
- [46] W3C, “Verifiable Credentials JSON Schema Specification,” Candidate Recommendation Draft, 4 February 2025. Dated draft.
- [47] H. Birkholz, D. Thaler, M. Richardson, N. Smith and W. Pan, “Remote ATtestation procedureS (RATS) Architecture,” IETF RFC 9334, January 2023. RFC.
- [48] NVIDIA, “Secure Storage,” Jetson Linux Developer Guide, release 38.4. Documentation.
- [49] OP-TEE Project, “Secure Storage,” documentation, version 4.4.0. Documentation.