Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: 19 November 2026

T. Sato
MyAuberge K.K.

19 May 2026

The Human Escalation Mechanism (HEM) for Agentic AI Systems
draft-sato-soos-hem-01

Abstract

AI agents operating autonomously encounter situations where their authorized action space is insufficient for the task at hand, where policy mandates human judgment regardless of agent capability, or where the agent itself assesses that human oversight is required. No existing standard specifies the kernel-level contract that governs what happens in these moments: what triggers escalation, what a human principal may decide, how the system behaves while awaiting that decision, and what happens when the designated human is unreachable.

This document defines the Human Escalation Mechanism (HEM): a normative protocol in which a governing OS kernel places an agent session into a formally defined HEM_PENDING state, routes a structured escalation request to one or more designated human principals, enforces a prohibition on state transitions until a human decision is received, and processes five defined human decision types. HEM is a first-class kernel session state, not an application-layer confirmation protocol. This document also defines the Policy Rationale Declaration (PRD), which links Cedar policies that route to HEM with machine-readable rationale, and the Decision Rationale Record (DRR), which captures the human principal’s reasoning for high-consequence decisions. HEM provides the technical specification for human oversight of agentic AI systems required by EU AI Act Article 14.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 19 November 2026.

Copyright Notice

Table of Contents

1. Introduction

The deployment of AI agents in consequential workflows -- booking systems, healthcare coordination, financial operations, logistics -- requires that those systems be supervisable by humans. This requirement is not merely normative; it is increasingly regulatory. EU AI Act Article 14 mandates that high-risk AI systems include human oversight measures that enable humans to intervene in the operation of the system [EUAIA].

The existing landscape of AI agent standards provides no normative specification of what such intervention looks like at the protocol

level. WIMSE [I-D.ietf-wimse-arch] addresses workload identity. AAuth [I-D.klrc-aiagent-auth] addresses token-based authorization. CHEQ [I-D.rosenberg-aiproto-cheq] provides an application-layer protocol for human confirmation of specific agent actions. None of these specifications defines a kernel-level contract for what happens when an agent's authorized scope is insufficient, when policy mandates human judgment, or when an agent itself declares that it requires human oversight before proceeding.

The distinction between AAuth [I-D.klrc-aiagent-auth] and this document is specifically: AAuth achieves mission correlation, binding authorization tokens to a declared mission scope; HEM achieves mission containment, specifying the kernel-enforced state contract that governs agent sessions when autonomous execution must yield to human judgment. The two primitives address different layers of the same problem.

This document defines the Human Escalation Mechanism (HEM): a normative protocol governing the lifecycle of human oversight events in agentic AI systems. HEM specifies:

* Five trigger classes that cause a kernel to place a session into HEM_PENDING state (Section 5), with a defined extension mechanism for additional trigger classes (Section 5.7).
* A Policy Rationale Declaration (PRD) linking each Cedar policy that routes to HEM with machine-readable rationale (Section 5.6).
* A structured HEM escalation request routed to one or more designated human principals, kernel-signed with Ed25519 (Section 6).
* Five human decision types that resolve a pending escalation (Section 7).
* A Decision Rationale Record (DRR) capturing human principal reasoning for high-consequence decisions (Section 7.6).
* A mandatory transition prohibition enforced by the kernel while HEM_PENDING is active (Section 8).
* A timeout model governing behaviour when designated human principals are unreachable (Section 9).
* A mission validity clause governing HEM behaviour when the governing MissionDeclaration enters a terminal phase during an active HEM_PENDING event (Section 8.4).

HEM is a kernel primitive. It is enforced by the governing runtime kernel, not by the agent and not by the application layer. An agent cannot opt out of HEM; an application cannot suppress it. This kernel-enforced quality is what distinguishes HEM from application-layer confirmation protocols such as CHEQ [I-D.rosenberg-aiproto-cheq] and gives HEM its regulatory utility.

HEM does not specify the delivery mechanism by which escalation requests reach human principals. That is a deployment concern. Application-layer protocols such as CHEQ MAY serve as the delivery mechanism for HEM notification; HEM specifies what that delivery must achieve, not how it is achieved.

2. Conventions and Definitions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

This document uses the following terms:

Agent:

A software system that uses AI to reason about and take actions in pursuit of goals.

Governing Kernel:
A runtime component that enforces authorization policy, maintains session state, records agent actions to a tamper-evident Event Log, and mediates agent access to governed objects.

Governed Object (SO):
A typed, stateful entity managed by the kernel. Its lifecycle is governed by a Cedar-enforced state machine.

Mandate JWT:
A JSON Web Token [RFC7519] binding an agent's authorization to a specific governed object instance.

Cedar:
A policy language and evaluation engine [Cedar] used by the kernel to evaluate authorization decisions.

HEM:
Human Escalation Mechanism. The protocol defined by this document.

HEM_PENDING:
The kernel session state in which HEM is active. No governed object state transitions may execute while HEM_PENDING is active.

HEM_RESOLVED:
The kernel session state following receipt and processing of a valid human decision.

HEM_TIMEOUT:
The kernel session state following expiration of the HEM timeout budget without receipt of a valid human decision.

Human Principal:
A natural person designated in the governed object's Principal Registry as authorized to issue HEM decisions.

Designation Chain:
An ordered list of human principals to be contacted sequentially if earlier principals in the chain are unreachable.

IDP:
Intent Declaration Primitive [I-D.sato-soos-idp]. The per-transition agent reasoning declaration that accompanies each kernel.transition() call.

MissionDeclaration:
A structured declaration [SOOS] specifying the overarching goal, lifecycle conditions, and authority constraints under which a set of agent sessions operate. A MissionDeclaration has a phase (DISCOVERY, EXECUTION, RELEASE, SUSPENDED, FAILED, ABANDONED).

Proximity Threshold:
A pre-declared monitoring condition specifying a governed object state, resource value, or event pattern that, when reached, triggers HEM regardless of agent action.

Event Log:
An append-only, causally ordered, tamper-evident log of all transitions and kernel events for a governed object.

3. Problem Statement

3.1. The Human Oversight Gap in Agentic Systems

Agentic AI systems operating in automated workflows encounter three classes of situation that require human involvement:

(a) Scope insufficiency. The action required to advance the task is outside the agent's Cedar-authorized action set. The agent cannot proceed autonomously and cannot determine whether a human would authorize the action.

(b) Policy-mandated oversight. Authorization policy specifies that certain transitions require human approval regardless of agent capability. The agent may be capable of executing the action but policy prohibits autonomous execution.

(c) Agent-declared uncertainty. The agent assesses that its confidence in the appropriate action is insufficient and that human judgment should be sought before proceeding. This is a first-class agent capability, not a failure mode.

No existing protocol specifies a normative kernel-level response to any of these three classes. The consequences of this gap include:

* Agents that stall silently when their scope is insufficient, with no structured notification to human principals.
* Agents that execute policy-mandated-oversight transitions autonomously because no enforcement mechanism exists.
* Agents that have no normative mechanism for expressing that they require human judgment.
* Systems with no defined behaviour for the period between escalation and human decision.
* Systems with no defined behaviour when designated human principals are unreachable.
* Ghost execution: sessions continuing to operate after the overarching mission for which they were authorized has entered a terminal state (SUSPENDED, FAILED, or ABANDONED), producing actions that have no legitimate mission context.

3.2. Relationship to EU AI Act Article 14

EU AI Act Article 14 requires that high-risk AI systems be designed and developed, where technically feasible, to be effectively overseen by natural persons during the period in which the AI system is in use. Specifically, Article 14 requires that human oversight measures enable those persons to:

* understand the capabilities and limitations of the high-risk AI system;
* monitor the operation of the system;
* detect and address the emergence of anomalous situations; and
* intervene in the operation of the system or interrupt the system through a "stop" button or similar procedure.

HEM provides the technical specification for these requirements at the kernel level. The HEM_PENDING state and transition prohibition (Section 8) implement the "stop" button. The five human decision types (Section 7) implement structured intervention. The timeout model (Section 9) defines system behaviour during unreachability, preventing indefinite autonomous operation in oversight-mandated states.

This document is informative with respect to regulatory compliance and does not constitute legal advice.

3.3. Relationship to CHEQ

CHEQ [I-D.rosenberg-aiproto-cheq] defines an application-layer protocol by which AI agents request human confirmation of specific actions before executing them. CHEQ is agent-initiated: the agent chooses to request confirmation, constructs a CHEQ confirmation object, and delivers it to a human confirmation service.

HEM is architecturally distinct in three respects.

First, HEM is kernel-enforced. A HEM trigger causes the kernel to enter HEM_PENDING regardless of agent preference. An agent cannot suppress a kernel-triggered HEM; it cannot proceed without a human decision once HEM_PENDING is active. CHEQ confirmation is a choice the agent makes; HEM_PENDING is a state the kernel enforces.

Second, HEM defines the session state contract. While HEM_PENDING is active, all state transitions are prohibited (Section 8). CHEQ does not specify any equivalent session state constraint.

Third, HEM defines a structured decision vocabulary. The five HEM decision types (Section 7) are normative; they define what a human principal may decide and what the kernel must do with each decision. CHEQ defines a confirmation/denial binary without equivalent normative treatment of the decision space.

CHEQ and HEM are complementary. A CHEQ-like application-layer protocol MAY serve as the delivery mechanism for HEM escalation notifications (Section 6.3). HEM specifies the kernel contract that such a delivery protocol must satisfy; it does not specify the delivery mechanism itself.

4. HEM Session States

4.1. State Definitions

A kernel session with respect to HEM is in one of the following states at any moment:

HEM_INACTIVE:
: Default state. No HEM condition is active. The kernel processes agent transitions normally subject to Cedar evaluation.

HEM_PENDING:
: A HEM trigger has fired and the kernel has routed an escalation request to one or more designated human principals. No governed object state transitions may execute. This state persists until a valid human decision is received or the timeout budget is exhausted.

HEM_RESOLVED:
: A valid human decision has been received and processed. The session returns to HEM_INACTIVE unless the human decision itself requires further HEM consideration.

HEM_TIMEOUT:
: The timeout budget has been exhausted without receipt of a valid human decision. The kernel applies the configured timeout disposition (Section 9.2). This state is terminal for the current HEM event; the session MAY return to HEM_INACTIVE depending on the disposition.

HEM_CHAIN_EXHAUSTED:
: The designation chain has been exhausted without a decision from any principal and the timeout budget is exhausted. The kernel

applies the chain exhaustion disposition (Section 9.4). This state is terminal for the current HEM event.

4.2. State Transition Constraints

The following state transitions are normative for a conforming kernel implementation:

```
HEM_INACTIVE --> HEM_PENDING (HEM trigger fires)
HEM_PENDING --> HEM_RESOLVED (valid decision received)
HEM_PENDING --> HEM_TIMEOUT (timeout budget exhausted)
HEM_PENDING --> HEM_CHAIN_EXHAUSTED (mission validity failure;
                                     see Section 8.4)
HEM_TIMEOUT --> HEM_CHAIN_EXHAUSTED (chain exhausted)
HEM_TIMEOUT --> HEM_INACTIVE (disposition permits)
HEM_RESOLVED --> HEM_INACTIVE (always)
HEM_CHAIN_EXHAUSTED --> [terminal] (session suspended or
                         terminated per Section 9.4)
```

The transition HEM_INACTIVE --> HEM_PENDING MUST be kernel-initiated. No agent call may directly set the session to HEM_PENDING except through the defined trigger mechanisms (Section 5).

5. HEM Trigger Classes

This document defines five trigger classes. A trigger class specifies the condition under which a kernel MUST enter HEM_PENDING. Trigger classes are evaluated in the order defined below. The first trigger class that fires causes HEM_PENDING to be entered; the remaining classes are not evaluated for that transition.

5.1. Class 1: Cedar Policy Routing (HEM_CEDAR_ROUTED)

A Cedar policy may explicitly route a transition to HEM rather than resulting in PERMIT or DENY. This is the primary mechanism by which governed object designers specify that certain transitions require human approval regardless of agent identity or scope.

A Cedar policy routes to HEM by including the condition:

```
when { context.hem_required == true }
```

combined with a forbid rule that prevents autonomous execution. The kernel MUST recognize this pattern and MUST enter HEM_PENDING rather than returning DENY when Cedar evaluation produces this result.

The kernel identifies HEM routing by evaluating whether a Cedar policy explicitly forbids the action AND the forbid rule includes a hem_required context attribute set to true. Implementations MAY use a Cedar annotation or action attribute to mark HEM-required transitions; the specific mechanism is implementation-defined, subject to the requirement that the kernel can unambiguously distinguish a HEM-routed DENY from a policy DENY.

Example Cedar policy routing a transition to HEM:

```
// Booking finalization requires human approval
forbid(
  principal,
  action == Action::"FinalizeBooking",
  resource
)
when {
  context.hem_required == true &&
  !context.human_approval_present
```

```
};
```

5.2. Class 2: Agent Uncertainty Declaration (HEM_AGENT_ESCALATED)

An agent MAY declare, via the hem_urgency field of an IDP [I-D.sato-soos-idp], that it requires human judgment before an action executes. When the kernel receives a kernel.transition() call with a valid IDP containing hem_urgency: REQUIRED, the kernel MUST enter HEM_PENDING for the session, regardless of Cedar policy evaluation outcome.

Note: Cedar evaluation still executes when hem_urgency is REQUIRED. A Cedar PERMIT result does NOT cause the action to execute. The kernel enters HEM_PENDING and the action awaits human decision. A Cedar DENY result in the same call produces both a HEM_PENDING entry AND a CEDAR_DENY_RECORDED Event Log entry.

The HEM_AGENT_ESCALATED trigger represents a first-class agent capability. It is not a failure mode. Kernel implementations MUST treat agent-initiated escalation as a legitimate and expected system behaviour.

5.3. Class 3: Proximity Threshold Crossing (HEM_PROXIMITY_TRIGGERED)

A governed object’s configuration MAY include one or more Proximity Threshold declarations. A Proximity Threshold specifies a monitored condition that, when met, causes the kernel to enter HEM_PENDING regardless of any agent action being in progress.

A Proximity Threshold declaration MUST specify:

```
{
  "threshold_id":   string,  ; REQUIRED. UUID v4.
  "description":    string,  ; REQUIRED. Human-readable.
  "condition_type": string,  ; REQUIRED. See below.
  "condition_value": value,  ; REQUIRED. Type-specific.
  "hem_disposition": string  ; REQUIRED. "BLOCK" or "NOTIFY".
}
```

condition_type values:

SO_STATE:
: The governed object enters the specified state. condition_value is a state name string.

RESOURCE_THRESHOLD:
: A numeric resource value associated with the governed object crosses the specified threshold. condition_value is { "resource": string, "operator": string, "value": number }.

EVENT_PATTERN:
: A specified event pattern appears in the Event Log for the governed object. condition_value is a pattern descriptor; the pattern language is implementation-defined.

hem_disposition values:

BLOCK:
: The kernel enters HEM_PENDING immediately. No transitions may execute until a human decision is received. This is the normative "stop" disposition.

NOTIFY:
: The kernel routes an escalation notification to designated

human principals but does NOT enter HEM_PENDING. Transitions may continue to execute. This disposition is appropriate for monitoring conditions that warrant human awareness but not intervention.

Proximity Threshold evaluation MUST execute after each successful Event Log commitment, including after IDP_SUBMITTED events and after STATE_TRANSITIONED events. A kernel MUST NOT evaluate proximity thresholds only at action request time.

5.4. Class 4: Mission Validity Failure (HEM_MISSION_VALIDITY_FAILED)

When the Mission Authority Service [I-D.mcguinness-oauth-mission-bound-authorization] reports that the governing MissionDeclaration has transitioned to a non-active phase (SUSPENDED, FAILED, or ABANDONED) while a session is otherwise active and no HEM_PENDING event is in progress, the kernel MUST enter HEM_PENDING so that a human principal can confirm terminal disposition before the session is closed.

This trigger class is distinct from the mission validity monitoring described in Section 8.4. Section 8.4 governs the case where mission invalidity is detected while HEM_PENDING is already active. HEM_MISSION_VALIDITY_FAILED governs the initial HEM entry when mission invalidity is first detected in an otherwise HEM_INACTIVE session.

The kernel MUST record HEM_TRIGGERED with trigger_class HEM_MISSION_VALIDITY_FAILED, including mission_ref and the terminal mission_phase value detected. The designated human principal designation chain for this trigger class is the mission-level principal chain, not the governed object principal chain.

Decision constraints for HEM_MISSION_VALIDITY_FAILED: APPROVE and APPROVE_WITH_CONSTRAINTS MUST NOT be accepted. The valid decision types are TERMINATE, DEFER, and REDIRECT (cleanup-only actions only). The kernel MUST enforce this constraint and MUST return HEM_DECISION_INVALID for APPROVE or APPROVE_WITH_CONSTRAINTS submissions.

5.5. Class 5: Jurisdictional Conflict (HEM_JURISDICTIONAL_CONFLICT)

When the kernel determines that it cannot algorithmically resolve a conflict between the legal requirements of two or more applicable jurisdictions -- such that no single authorized action set satisfies all applicable requirements simultaneously -- the kernel MUST enter HEM_PENDING with trigger_class HEM_JURISDICTIONAL_CONFLICT.

The HEM Escalation Request for Class 5 MUST include a jurisdictional_conflict_summary object (Section 6.2). This object provides human principals with a structured summary of the conflicting jurisdictional requirements so that they may exercise informed judgment about the appropriate resolution.

Decision constraints for HEM_JURISDICTIONAL_CONFLICT: APPROVE and APPROVE_WITH_CONSTRAINTS MUST NOT be accepted. The valid decision types are TERMINATE, DEFER, REDIRECT (subject to Cedar evaluation of the redirected action), and APPROVE_WITH_LEGAL_BASIS (reserved; see Section 7.1). The kernel MUST enforce this constraint and MUST return HEM_DECISION_INVALID for non-permitted decision types.

The jurisdictional conflict resolution mechanism (JCR) is specified as Part II of [I-D.sato-soos-cap]. This document specifies only the HEM trigger and session contract for Class 5; the conflict analysis and resolution algorithm are defined in that document.

5.6. Policy Rationale Declaration (PRD)

A Policy Rationale Declaration (PRD) is a kernel-registered object that links a Cedar policy routing transitions to HEM with machine-readable rationale. PRDs support regulatory transparency requirements (EU AI Act Article 13) and provide human principals with the policy basis for escalation events.

A PRD MUST be registered with the kernel before the Cedar policy that references it is loaded. A Cedar policy that routes to HEM (Section 5.1) MUST carry a prd_id annotation referencing a registered PRD. The kernel MUST reject at load time any Cedar policy that routes to HEM and either lacks a prd_id annotation or references an unregistered prd_id. The rejection error is HEM_PRD_MISSING.

A PRD object MUST contain the following fields:

```
{
  "prd_id":         string, ; REQUIRED. UUID v4. Primary key.
  "rationale_class": string, ; REQUIRED. See IANA registry
                             ;   Section 15.
  "rationale_text": string, ; REQUIRED. Human-readable rationale
                             ;   for requiring human oversight.
  "authority_ref":  string, ; REQUIRED when rationale_class is
                             ;   REGULATORY or CONTRACTUAL.
                             ;   Citation of the regulation or
                             ;   contract provision.
  "review_date":    string, ; REQUIRED. ISO 8601 date. The date
                             ;   by which this PRD SHOULD be
                             ;   reviewed.
  "review_overdue": boolean  ; REQUIRED. Kernel-set at escalation
                             ;   request generation time. TRUE if
                             ;   current date exceeds review_date.
}
```

The PRD store additionally retains the following administrative fields, which are NOT surfaced to human principals in the HEM Escalation Request: set_by (principal_id of registrant), set_at (ISO 8601 UTC registration timestamp), and policy_id (identifier of the Cedar policy referencing this PRD).

The policy_rationale_id field in the HEM Escalation Request (Section 6.1) carries the prd_id of the PRD associated with the triggering Cedar policy. This field is REQUIRED on HEM_TRIGGERED Event Log entries for Class 1 triggers. The full PRD object is accessible to human principals via kernel.query_rationale().

A PRD review_overdue value of TRUE MUST generate a PRD_REVIEW_DATE_EXCEEDED Audit Alert (Section 10) at escalation request generation time.

5.7. Extension Trigger Classes

This document recognizes that operational experience may identify additional trigger classes beyond those defined in Sections 5.1 through 5.5. Implementors MAY define additional trigger classes using a URI-prefixed identifier. Extension trigger classes MUST be submitted as an IDP with hem_urgency REQUIRED or as a kernel-internal event; they MUST NOT bypass the HEM_PENDING state machine.

Implementors wishing to register extension trigger classes SHOULD do so via the IANA registry defined in Section 15.

6. HEM Escalation Request

6.1. Structure

When a HEM trigger fires and the kernel enters HEM_PENDING, the kernel MUST construct and route a HEM Escalation Request. The escalation request is a JSON object:

```
{
  "hem_id":          string,  ; REQUIRED. UUID v4. Unique HEM event ID.
  "so_id":           string,  ; REQUIRED. Governed object UUID.
  "session_id":      string,  ; REQUIRED. Agent session identifier.
  "mandate_id":      string,  ; REQUIRED. Active mandate JWT jti.
  "mission_ref":     string,  ; OPTIONAL. MissionDeclaration UUID if
                              ;   session operates under a declared
                              ;   mission. NULL if no MissionDeclaration
                              ;   governs this session.
  "mission_phase":   string,  ; OPTIONAL. Current MissionDeclaration
                              ;   phase at time of HEM trigger. Provided
                              ;   to enable human principals to assess
                              ;   whether the mission context remains
                              ;   valid. NULL if mission_ref is NULL.
  "trigger_class":   string,  ; REQUIRED. One of the five defined
                              ;   trigger classes (Section 5) or an
                              ;   extension trigger class URI.
  "trigger_detail":  array,   ; REQUIRED. Typed array of trigger-class-
                              ;   specific detail objects. See Section
                              ;   8.3 for concurrent extension schema.
  "policy_rationale_id": string, ; REQUIRED for Class 1 triggers.
                              ;   prd_id of the PRD associated with
                              ;   the triggering Cedar policy.
                              ;   NULL for other trigger classes.
  "jurisdictional_conflict_summary": object, ; REQUIRED when
                              ;   trigger_class is
                              ;   HEM_JURISDICTIONAL_CONFLICT.
                              ;   NULL otherwise. See Section 6.2.
  "idp_summary": {            ; REQUIRED if IDP present; else null.
    "goal_description":  string,
    "reasoning_type":    string,
    "confidence_level":  number,
    "requested_action":  string,
    "mission_ref":       string  ; Echoed from triggering IDP.
  },
  "so_state_summary": {       ; REQUIRED.
    "current_state":           string,
    "phase":                   string,
    "available_actions_if_resolved": [string]
  },
  "principals":      [object], ; REQUIRED. Ordered designation chain.
  "timeout_seconds": integer,  ; REQUIRED. Per-principal timeout.
  "created_at":      string,   ; REQUIRED. ISO 8601 UTC.
  "kernel_signature": string,  ; REQUIRED. Ed25519 signature by the
                              ;   kernel over the canonical
                              ;   serialization of all other fields.
                              ;   See Section 12.8.
  "metadata":        object    ; OPTIONAL. Implementation-defined.
}
```

6.2. Field Definitions

hem_id:
A UUID v4 uniquely identifying this HEM event. Used as the primary key for all subsequent Event Log entries relating to this HEM lifecycle.

mission_ref:
   OPTIONAL. The UUID of the MissionDeclaration governing the
   session at the time of the HEM trigger. When present, provides
   human principals with the mission context needed to evaluate
   whether the action under review is appropriate within the
   overarching mission. NULL if the session does not operate under
   a declared MissionDeclaration.

mission_phase:
   OPTIONAL. The current phase of the MissionDeclaration identified
   by mission_ref at the time of the HEM trigger. Human principals
   use this field to assess whether the mission is still active.
   Possible values: DISCOVERY, EXECUTION, RELEASE, SUSPENDED,
   FAILED, ABANDONED. NULL if mission_ref is NULL. Human principals
   who receive an escalation request where mission_phase is
   SUSPENDED, FAILED, or ABANDONED SHOULD treat TERMINATE as the
   expected disposition unless there is specific reason to believe
   the mission state will be corrected.

trigger_class:
   One of the five defined trigger classes (Section 5) or an
   extension trigger class URI.

trigger_detail:
   A typed array of trigger-class-specific objects. The initial
   entry is populated at HEM trigger time. When a concurrent
   HEM_PENDING condition fires (Section 8.3), the kernel appends
   a new typed entry to this array. Each entry MUST contain:
   extension_type (string), extended_at (ISO 8601 UTC),
   trigger_source (string), and class-specific detail fields.
   For HEM_PROXIMITY_TRIGGERED entries: threshold_id,
   threshold_class, condition_met, triggered_at. See Section 13
   for minimisation requirements.

policy_rationale_id:
   The prd_id of the PRD (Section 5.6) associated with the Cedar
   policy that triggered this HEM event. REQUIRED for Class 1
   triggers (HEM_CEDAR_ROUTED). NULL for all other trigger classes.
   Provides human principals with direct access to the machine-
   readable policy rationale via kernel.query_rationale(prd_id).

jurisdictional_conflict_summary:
   REQUIRED when trigger_class is HEM_JURISDICTIONAL_CONFLICT.
   NULL otherwise. A structured object containing:
   {
     "jurisdictions": [string], ; ISO 3166-1 alpha-2 codes of the
                                ;   conflicting jurisdictions.
     "conflict_description": string, ; Human-readable description.
     "conflicting_requirements": [object], ; One entry per
                                ;   jurisdiction with fields:
                                ;   jurisdiction, requirement_ref,
                                ;   requirement_text.
     "resolution_methods_available": [string] ; From CAP JCR;
                                ;   see [I-D.sato-soos-cap].
   }

kernel_signature:
   An Ed25519 signature produced by the kernel over the canonical
   serialization of all other fields in the HEM Escalation Request.
   See Section 12.8. REQUIRED. Delivery intermediaries capable of
   verifying this signature MUST do so and MUST NOT forward the
   request if verification fails.

idp_summary:
   A summary of the IDP submitted with the triggering action, if

present. This provides human principals with context about what the agent was attempting to do. MUST be null if no IDP was associated with the trigger. The mission_ref field within idp_summary echoes the mission_ref from the triggering IDP; this allows principals to confirm that the IDP's declared mission matches the session's governing MissionDeclaration.

so_state_summary:
A summary of the governed object's current state, phase, and the Cedar actions that would be available to the agent if the HEM is resolved with APPROVE. This enables human principals to make informed decisions without requiring access to the full governed object record.

principals:
An ordered array of human principal descriptors forming the designation chain. Each element MUST contain:

```
{
  "principal_id":   string, ; Party Registry identifier.
  "display_name":   string, ; Human-readable name.
  "contact":        object, ; Delivery contact; implementation-defined.
  "timeout_seconds": integer ; Per-principal timeout override. Optional.
}
```

timeout_seconds:
The default per-principal timeout in seconds. If a principal descriptor includes a timeout_seconds override, the override applies for that principal. See Section 9.

6.3. Routing to Human Principals

HEM does not specify the delivery mechanism. The kernel MUST deliver the escalation request to the first principal in the designation chain. The delivery mechanism is implementation-defined and MAY include:

* Push notification to a mobile application.
* Message delivery via a registered messaging endpoint (e.g., the ATP Guest Agent [SOOS] uses WhatsApp Business API and LINE Messaging API as delivery channels).
* Email to a registered address.
* HTTP POST to a registered webhook.
* Any CHEQ-compatible [I-D.rosenberg-aiproto-cheq] confirmation delivery mechanism.

The kernel MUST record the delivery attempt in the Event Log regardless of the delivery mechanism. The kernel MUST NOT consider delivery confirmed until an acknowledgment is received from the delivery channel. Delivery failure MUST trigger escalation to the next principal in the designation chain without waiting for the per-principal timeout to expire.

7. Human Decision Types

A human principal responds to a HEM Escalation Request with one of five decision types. The kernel MUST accept only these five decision types as valid HEM resolutions. Any response that does not conform to one of these types MUST be rejected by the kernel with a HEM_DECISION_INVALID error.

No HEM decision type substitutes for Cedar policy evaluation. APPROVE and APPROVE_WITH_CONSTRAINTS authorize the kernel to proceed to Cedar evaluation for the triggering action; they do not override it. A Cedar DENY result following an APPROVE or APPROVE_WITH_

CONSTRAINTS decision takes precedence over the human decision; the action does not execute, and the kernel returns an Enriched DENY Response [I-D.sato-soos-idp] to the agent. This property is the normative separation between human judgment (was this action appropriate in context?) and policy enforcement (is this action permitted?). These two questions have different authorities and MUST NOT be collapsed.

7.1. APPROVE

The human principal approves the requested action and authorizes the agent to proceed.

Kernel behaviour upon receiving APPROVE:

* The kernel MUST record HEM_DECISION_RECEIVED(APPROVE) in the Event Log.
* The kernel MUST execute the Cedar policy evaluation for the action that triggered the HEM. If Cedar PERMIT results, the action MUST execute. If Cedar DENY results, the DENY takes precedence; APPROVE does not override Cedar policy (see Section 7 preamble).
* The session transitions to HEM_RESOLVED and then HEM_INACTIVE.

Note: APPROVE is not a Cedar override. It is a signal that the human principal has reviewed the situation and considers the action appropriate within the governing mission and policy context. If the action is Cedar-denied for reasons unrelated to the HEM trigger, the denial stands.

7.2. APPROVE_WITH_CONSTRAINTS

The human principal approves the requested action subject to specified constraints. Constraints are expressed as additional Cedar context attributes injected into the subsequent Cedar evaluation.

APPROVE_WITH_CONSTRAINTS MUST include a constraints object:

```
{
  "decision": "APPROVE_WITH_CONSTRAINTS",
  "constraints": {
    "cedar_context_additions": object, ; Additional Cedar context.
    "expiry_seconds":  integer,        ; OPTIONAL. Constraint validity
                                       ;   window in seconds from
                                       ;   HEM_DECISION_RECEIVED.
    "description":     string          ; Human-readable summary.
  }
}
```

Kernel behaviour:

* The kernel MUST record HEM_DECISION_RECEIVED(APPROVE_WITH_ CONSTRAINTS) in the Event Log, including the full constraints object.
* The kernel MUST inject the cedar_context_additions into the Cedar context for the subsequent action evaluation.
* Expiry semantics: expiry_seconds governs action initiation, not completion. Transitions for which initiation begins before the expiry window closes MUST be permitted to complete. Transitions for which initiation has not begun when the expiry window closes MUST be evaluated without the constraint additions; Cedar context returns to the pre-constraint state. Initiation time is recorded as the timestamp of the kernel.transition() call.
* When an initiation attempt is rejected due to constraint expiry, the kernel MUST return HEM_CONSTRAINT_EXPIRED. The expiry time is recorded against hem_id, not individual transitions.
* The session transitions to HEM_RESOLVED and then HEM_INACTIVE.

7.3. REDIRECT

The human principal does not approve the requested action but authorizes the agent to take a different specified action instead.

REDIRECT MUST include a redirect object:

```
{
  "decision": "REDIRECT",
  "redirect": {
    "action":      string, ; Cedar action string for the new action.
    "description": string  ; Human-readable rationale.
  }
}
```

Kernel behaviour:

* The kernel MUST record HEM_DECISION_RECEIVED(REDIRECT) in the Event Log.
* The kernel MUST evaluate the redirected action against Cedar policy. If Cedar PERMIT results, the redirected action executes. If Cedar DENY results, the kernel returns HEM_REDIRECT_DENIED and an Enriched DENY Response [I-D.sato-soos-idp] to the submitting principal. HEM does NOT re-trigger automatically on Cedar DENY of a REDIRECT; there is no normative default retrigger. A REDIRECT_DENY_RETRIGGER extension may be defined in future work.
* When Cedar DENY follows REDIRECT, the principal’s decision slot is NOT consumed. The principal MAY submit a new decision within the same active HEM event. This permits iterative resolution without requiring a new HEM lifecycle.
* The original requested action MUST NOT execute.
* When Cedar PERMIT follows REDIRECT, the session transitions to HEM_RESOLVED and then HEM_INACTIVE.

7.4. TERMINATE

The human principal terminates the agent session. The session is immediately closed. No further agent actions may be executed in this session.

Kernel behaviour upon receiving TERMINATE. The following steps MUST be executed in the order specified. Steps 1 and 2 are atomic: both MUST be committed before HEM_DECISION_ACCEPTED is returned to any caller. Step 3 executes after the atomic completion of Steps 1 and 2.

Step 1 -- Mandate revocation:

* The kernel MUST record HEM_DECISION_RECEIVED(TERMINATE) in the Event Log.
* The mandate JWT associated with this session MUST be revoked. Revocation MUST propagate to all relying parties before any further agent calls are accepted on that mandate.
* The principal who issued TERMINATE MUST be recorded by principal_id in the Event Log.

Step 2 -- Mission cascade (atomic with Step 1):

* If the session is associated with a governing MissionDeclaration and is the sole active session under that MissionDeclaration, the kernel MUST initiate MISSION_REVOKE_CASCADE. Steps 1 and 2 MUST both be committed -- mandate revocation completed AND MISSION_REVOKE_CASCADE initiated -- before HEM_DECISION_ACCEPTED is returned. No further calls on related mandates are accepted until both commitments are complete.
* The semantics of MISSION_REVOKE_CASCADE are defined in [SOOS] Section 10. HEM TERMINATE is one normative trigger for that protocol.

Step 3 -- Governed object disposition (after atomic completion):

* The kernel MUST apply the SO Type's defined termination disposition for the governed object's current state. The SO Type MUST define a termination disposition for each state in which SESSION_TERMINATED is reachable.
* The kernel MUST enter SESSION_TERMINATED state.

A TERMINATE decision for a Class 5 trigger (HEM_JURISDICTIONAL_CONFLICT) MUST be accompanied by a Decision Rationale Record (DRR) with safety_basis populated (Section 7.6). The kernel MUST reject a TERMINATE decision on a Class 5 trigger that lacks a DRR with HEM_DRR_REQUIRED.

7.5. DEFER

The human principal is unable to decide at this moment and requests that the HEM timeout be extended. This decision type acknowledges the situation without resolving it.

DEFER MUST include a defer object:

```
{
  "decision": "DEFER",
  "defer": {
    "extension_seconds": integer, ; REQUIRED. Additional seconds.
    "reason":            string   ; REQUIRED. Human-readable.
  }
}
```

Kernel behaviour:

* The kernel MUST record HEM_DECISION_RECEIVED(DEFER) in the Event Log.
* The kernel MUST extend the remaining timeout by extension_seconds.
* The session remains in HEM_PENDING.
* The kernel SHOULD deliver a re-notification to the same principal when the extended timeout has 20% remaining.

Limits:

* A single DEFER extension MUST NOT exceed the original per-principal timeout_seconds for this HEM event.
* DEFER is limited to one per principal per hem_id. The limit is scoped independently for each principal: a DEFER by Principal A does not affect Principal B's DEFER availability. Escalation through the chain does not affect prior principals' DEFER records. The timeout countdown does not consume a principal's DEFER -- only a submitted DEFER decision does.
* A second DEFER from the same principal on the same hem_id MUST be rejected with HEM_DEFER_LIMIT_EXCEEDED.

7.6. Decision Rationale Record (DRR)

A Decision Rationale Record (DRR) captures the human principal's reasoning for a HEM decision. DRRs support regulatory transparency requirements (EU AI Act Article 13) and provide an auditable record of human judgment in the governance trail.

A DRR is REQUIRED for TERMINATE decisions. A DRR SHOULD be provided for APPROVE_WITH_CONSTRAINTS decisions. A DRR MAY be provided for any decision type. The kernel MUST reject a TERMINATE decision that lacks a DRR or that includes a DRR in which safety_basis is null, with error HEM_DRR_REQUIRED.

A DRR object MUST contain the following fields:

```
{
```

```
  "rationale_class": string, ; REQUIRED. See IANA registry
                             ;   Section 15.
  "rationale_text":  string, ; REQUIRED when DRR is required.
                             ;   Human-readable description of
                             ;   the principal's reasoning.
  "safety_basis":    string, ; REQUIRED for TERMINATE; null
                             ;   otherwise. Describes the safety
                             ;   or risk basis for the decision
                             ;   to terminate the session.
  "reference_ref":   string  ; OPTIONAL. Reference identifier
                             ;   (e.g., incident ID, regulation
                             ;   citation, contract clause) for
                             ;   any decision type. Included to
                             ;   support APPROVE_WITH_LEGAL_BASIS
                             ;   when that decision type becomes
                             ;   operational.
}
```

The kernel MUST validate the presence of required fields. The kernel MUST NOT validate the content quality of rationale_text or safety_basis. Content sufficiency is a human governance matter.

DRRs are committed to the Rationale Store, a separate kernel-managed object store. DRRs are immutable on commit. The Event Log carries drr_id as a foreign key on HEM_DECISION_RECEIVED entries where a DRR is present. DRRs are accessible via kernel.query_rationale(). The Session Audit Record (SAR) includes a rationale coverage summary counting decisions with and without DRRs.

The non-normative advisory in Section 14 maps DRR rationale_class and safety_basis to EU AI Act Article 13 transparency requirements.

7.7. Decision Submission Protocol

Human decisions are submitted to the kernel via a decision endpoint. The submission MUST include:

```
{
  "hem_id":        string, ; REQUIRED. Must match active HEM event.
  "principal_id":  string, ; REQUIRED. Party Registry identifier.
  "decision":      string, ; REQUIRED. One of the five types.
  "decision_data": object, ; REQUIRED for types with data objects.
  "drr":           object, ; REQUIRED for TERMINATE; OPTIONAL
                           ;   otherwise. See Section 7.6.
  "timestamp":     string, ; REQUIRED. ISO 8601 UTC.
  "signature":     string  ; REQUIRED. Principal's Ed25519 signature
                           ;   over hem_id + principal_id + decision
                           ;   + timestamp.
}
```

The kernel MUST verify the signature before processing the decision. The signing key MUST correspond to the principal's registered key in the Party Registry. A decision submitted with an invalid signature MUST be rejected with HEM_SIGNATURE_INVALID.

The kernel MUST accept decisions only from principals listed in the designation chain for the active HEM event. A decision from an unlisted principal MUST be rejected with HEM_PRINCIPAL_NOT_AUTHORIZED.

8. Transition Prohibition During HEM_PENDING

8.1. The Prohibition Rule

While a session is in HEM_PENDING state, the kernel MUST reject all

kernel.transition() calls with error code HEM_PENDING_ACTIVE. This prohibition is absolute and applies regardless of:

* The Cedar policy evaluation outcome for the requested action.
* The identity of the requesting agent.
* The mandate JWT presented.
* The IDP submitted with the call.

The transition prohibition is the primary safety property of HEM. It guarantees that no governed object state changes occur while a human oversight event is in progress. Implementations MUST enforce this prohibition at the kernel layer, prior to Cedar evaluation.

8.2. Read-Only Operations During HEM_PENDING

The following operations are PERMITTED during HEM_PENDING:

* kernel.query_state() -- reading the current governed object state.
* kernel.query_event_log() -- reading the Event Log.
* kernel.query_hem_status() -- reading the status of the active HEM event, including which principals have been notified and the remaining timeout.
* kernel.list_available_actions() -- listing Cedar-permitted actions for planning purposes (agents may plan during HEM_PENDING even though they may not act).

These operations do not modify governed object state and are therefore not subject to the transition prohibition.

8.3. Multiple Concurrent HEM_PENDING Conditions

A governing kernel MAY support multiple concurrent HEM events for distinct governed objects. Each governed object's HEM state is independent.

A single governed object MUST NOT have more than one active HEM_PENDING event at a time. If a proximity threshold fires while an agent-escalated HEM is already pending for the same governed object, the kernel MUST record the proximity threshold crossing in the Event Log and MUST extend the existing HEM event's context to include the proximity trigger detail. A second HEM event MUST NOT be created.

Context extension uses the trigger_detail typed array (Section 6.1). Each context extension appends a new typed entry to the array. The entry MUST contain:

```
{
  "extension_type":        string, ; REQUIRED. Trigger class of the
                                   ;   extending condition.
  "extended_at":           string, ; REQUIRED. ISO 8601 UTC.
  "trigger_source":        string, ; REQUIRED. Identifier of the
                                   ;   threshold or event that fired.
  "proximity_detail":      object, ; REQUIRED for HEM_PROXIMITY_
                                   ;   TRIGGERED extensions.
                                   ;   See Section 13 for
                                   ;   minimisation requirements.
  "timeout_extension_seconds": integer ; OPTIONAL. Additional seconds
                                   ;   to add to the current timeout
                                   ;   budget. Pre-declared by the
                                   ;   SO Type designer in the
                                   ;   proximity threshold config.
                                   ;   The timeout budget does NOT
                                   ;   reset automatically.
}
```

When a context extension occurs, the kernel MUST re-deliver the updated HEM Escalation Request to the active principal via the same delivery channel. The principal's decision slot is NOT reset by a context extension. The kernel MUST record HEM_CONTEXT_EXTENDED in the Event Log with: hem_id, extension_type, trigger_source, extended_at, and timeout_extension_seconds (if applicable).

8.4. Mission Validity During HEM_PENDING

A session in HEM_PENDING state may be associated with a governing MissionDeclaration [SOOS]. While HEM_PENDING is active, the kernel MUST monitor mission validity. If the governing MissionDeclaration transitions to phase SUSPENDED, FAILED, or ABANDONED during the HEM_PENDING period:

(a) The kernel MUST enter HEM_CHAIN_EXHAUSTED state with disposition TERMINATE_SESSION, regardless of whether the designation chain has been exhausted.

(b) The kernel MUST record a HEM_MISSION_INVALID event in the Event Log containing: hem_id, mission_ref, mission_phase (the terminal phase that was detected), and timestamp.

(c) Outstanding human principal notifications MUST be cancelled. The kernel MUST notify principals who have already received a notification that the session has been terminated due to mission invalidation. The cancellation notification is recorded as HEM_NOTIFICATION_CANCELLED in the Event Log.

(d) The TERMINATE disposition applies as defined in Section 7.4, including mandate JWT revocation and MISSION_REVOKE_CASCADE if the session is the sole active session under the MissionDeclaration.

This provision closes the ghost execution failure mode: a valid HEM event does not imply that the underlying mission for which the agent was operating is still active. A valid mandate JWT + valid HEM event does not imply a legitimate ongoing mission.

Note: This section applies only when HEM_PENDING is already active and the mission phase changes during the wait. The case where mission invalidity is first detected in an otherwise active session (no existing HEM event) is addressed by the Class 4 extension trigger class defined in Section 5.4.

9. Timeout Model

9.1. Timeout Budget

Each principal in the designation chain is allocated a timeout budget. The default timeout_seconds is specified in the HEM Escalation Request (Section 6.1). A per-principal override MAY be specified in the principal descriptor.

The timeout clock for a principal begins when the kernel delivers the escalation request to that principal (or records a delivery attempt, if delivery is not confirmed).

The minimum timeout_seconds is 60. A HEM configuration specifying a timeout_seconds less than 60 MUST be rejected by the kernel at configuration time.

9.2. Timeout Disposition

When a principal's timeout budget is exhausted without receipt of

a valid decision, the kernel MUST apply one of the following timeout dispositions. The disposition MUST be pre-declared in the governed object's HEM configuration; it MUST NOT be determined at timeout time.

ESCALATE_CHAIN:
: Escalate to the next principal in the designation chain. This is the default disposition when the chain has not been exhausted.

SUSPEND:
: Place the governed object in BOOKING_SUSPENDED state (or the equivalent suspended state for the SO Type). The session remains in HEM_PENDING. A governance recovery process must manually resolve the suspension.

TERMINATE_SESSION:
: Apply the TERMINATE decision semantics (Section 7.4). The session ends; the mandate JWT is revoked; the SO Type's termination disposition applies.

AUTO_APPROVE:
: Approve the triggering action and proceed as if APPROVE had been received. This disposition is Cedar-gated: the kernel MUST perform a Cedar policy evaluation before applying AUTO_APPROVE. If Cedar evaluation returns DENY, the AUTO_APPROVE disposition MUST NOT apply; the kernel MUST instead apply the SUSPEND disposition and record HEM_AUTO_APPROVE_CEDAR_DENIED in the Event Log. This ensures that AUTO_APPROVE cannot be used to bypass Cedar policy.

  Hard prohibition: AUTO_APPROVE MUST NOT be available for transition classes designated as high-value in the SO Type configuration, regardless of Cedar evaluation outcome. The kernel MUST enforce this prohibition at SO Type configuration load time by rejecting any HEM configuration that specifies AUTO_APPROVE for a high-value transition class. The rejection error is HEM_AUTO_APPROVE_PROHIBITED.

  AUTO_APPROVE MUST NOT be used for HEM_CEDAR_ROUTED triggers. It MAY be used for HEM_PROXIMITY_TRIGGERED triggers with hem_disposition: NOTIFY. Implementations SHOULD discourage this disposition for high-value transitions.

9.3. Timeout Chain (HEM_UNREACHABLE)

When the disposition is ESCALATE_CHAIN and the chain has not been exhausted, the kernel enters HEM_TIMEOUT state briefly, records HEM_PRINCIPAL_TIMEOUT in the Event Log, and immediately re-enters HEM_PENDING with the next principal as the active target.

A notification MUST be delivered to the next principal within 30 seconds of the prior principal's timeout.

The kernel MUST record in the Event Log for each principal in the chain:

* HEM_NOTIFICATION_SENT (with principal_id and timestamp)
* HEM_NOTIFICATION_DELIVERED or HEM_NOTIFICATION_UNDELIVERED
* HEM_PRINCIPAL_TIMEOUT (if timeout elapsed without decision)

9.4. Chain Exhaustion

If all principals in the designation chain have timed out without providing a valid decision, the kernel enters HEM_CHAIN_EXHAUSTED state. The kernel MUST apply the pre-declared chain exhaustion disposition.

The chain exhaustion disposition is specified separately from the per-principal timeout disposition and MUST be one of:

SUSPEND:
   As defined in Section 9.2. Recommended default for most SO Types.

TERMINATE_SESSION:
   As defined in Section 9.2.

The chain exhaustion disposition MUST be declared in the governed object's HEM configuration. A HEM configuration without a chain exhaustion disposition MUST default to SUSPEND.

10. Event Log Requirements

A conforming kernel MUST record the following events in the Event Log for each HEM lifecycle. All events MUST be kernel-signed [I-D.sato-soos-gar].

Event sequence for a complete HEM lifecycle:

HEM_TRIGGERED
   Fields: hem_id, trigger_class, trigger_detail (initial entry), so_id, session_id, mandate_id, mission_ref (if present), policy_rationale_id (REQUIRED for Class 1 triggers; null otherwise), timestamp.

HEM_NOTIFICATION_SENT (one per principal notified)
   Fields: hem_id, principal_id, delivery_mechanism, timestamp.

HEM_NOTIFICATION_DELIVERED or HEM_NOTIFICATION_UNDELIVERED
   Fields: hem_id, principal_id, timestamp.

HEM_DECISION_RECEIVED
   The following nine fields are REQUIRED on all HEM_DECISION_ RECEIVED entries:
      hem_id, session_id, mandate_id, trigger_class, principal_type, principal_id, trigger_source, decision_type, created_at.
   Additionally: decision_rationale_class is REQUIRED when a DRR is mandatory for the decision type; OPTIONAL otherwise. policy_rationale_id SHOULD be recorded. drr_id SHOULD be recorded when a DRR is submitted.

HEM_RESOLVED or HEM_TIMEOUT or HEM_CHAIN_EXHAUSTED
   Fields: hem_id, final_state, applied_disposition (if timeout or chain exhaustion), timestamp.

In addition, each of the following MUST be recorded when applicable:

HEM_CONTEXT_EXTENDED (Section 8.3)
   Fields: hem_id, extension_type, trigger_source, extended_at, timeout_extension_seconds (if applicable).

HEM_PRINCIPAL_TIMEOUT
   Fields: hem_id, principal_id, elapsed_seconds, timestamp.

HEM_DEFER_RECEIVED
   Fields: hem_id, principal_id, extension_seconds, timestamp.

HEM_DECISION_REJECTED
   Fields: hem_id, rejection_code, submitter_info, timestamp.

HEM_MISSION_INVALID (Section 8.4)

Fields: hem_id, mission_ref, mission_phase (terminal phase detected), timestamp.

HEM_NOTIFICATION_CANCELLED (Section 8.4)
Fields: hem_id, principal_id, reason, timestamp.

HEM_MISSION_REF_MISMATCH_REJECTED (Section 11)
Fields: hem_id, session_id, idp_id, expected_mission_ref, submitted_mission_ref, timestamp.

CONSTITUTIONAL_VIOLATION (when CAP Tier 0 or Tier 1 refusal applies during HEM_PENDING processing)
Fields: hem_id, violation_class, tier, timestamp.
Generates a CRITICAL Audit Alert [I-D.sato-soos-gar].

The Event Log ordering guarantee for a HEM lifecycle is:

HEM_TRIGGERED < HEM_NOTIFICATION_SENT < HEM_DECISION_RECEIVED < HEM_RESOLVED

No STATE_TRANSITIONED event for the governed object may appear in the Event Log between HEM_TRIGGERED and HEM_RESOLVED (or HEM_TIMEOUT, HEM_CHAIN_EXHAUSTED) events. This is the Event Log expression of the transition prohibition in Section 8.

The Session Audit Record (SAR) for a governed object session is generated by the kernel at session close and includes a reference array hem_events[] carrying the hem_id of each HEM lifecycle in that session. SAR specification is defined in [I-D.sato-soos-gar].

11. Relationship to IDP

The IDP [I-D.sato-soos-idp] and HEM are co-operating kernel primitives. Their relationship is as follows:

* An IDP with hem_urgency: REQUIRED triggers HEM_AGENT_ESCALATED (Section 5.2). The IDP idp_id is recorded in the HEM trigger detail.

* The idp_summary field of the HEM Escalation Request (Section 6.1) is derived from the triggering IDP. Human principals receive the agent's declared goal, reasoning type, confidence level, and requested action as part of the escalation context. The mission_ref field within idp_summary is echoed from the triggering IDP's mission_ref field, enabling principals to confirm mission context consistency.

* Mission_ref mismatch: The kernel MUST verify that the mission_ref field in each incoming IDP matches the mission_ref of the governing session MissionDeclaration. On mismatch, the kernel MUST reject the IDP and return an Enriched DENY Response [I-D.sato-soos-idp] containing mismatch_detail with expected_mission_ref and submitted_mission_ref values. The kernel MUST log HEM_MISSION_REF_MISMATCH_REJECTED (Section 10). If mismatches from the same session exceed a configurable threshold, the kernel SHOULD automatically trigger HEM_AGENT_ESCALATED (Class 2).

* IDP_COMMITMENT_GAP: When the IDP Commitment Verification Record [I-D.sato-soos-idp] detects that the agent's actual action diverged from its declared intent (IDP_COMMITMENT_GAP), the kernel MUST automatically trigger HEM_AGENT_ESCALATED (Class 2). This trigger is non-suppressible. IDP_COMMITMENT_GAP also generates a CRITICAL Audit Alert [I-D.sato-soos-gar].

* Following HEM_RESOLVED with a REDIRECT decision, the agent MUST

submit a new IDP for the redirected action before the kernel will execute it. The new IDP SHOULD reference the HEM hem_id in context_refs.

* Following HEM_RESOLVED with APPROVE or APPROVE_WITH_CONSTRAINTS, the agent MAY re-execute its original action. It MUST submit the original IDP's goal_id in the new IDP's declared_goal to provide continuity of audit trail.

* The Event Log ordering between IDP and HEM events is:

  IDP_SUBMITTED < HEM_TRIGGERED < HEM_RESOLVED < STATE_TRANSITIONED

12. Security Considerations

12.1. HEM_PENDING as a Denial-of-Service Vector

The transition prohibition during HEM_PENDING (Section 8.1) means that any attacker who can trigger HEM on a governed object can prevent state transitions for the duration of the HEM event. Implementations MUST rate-limit HEM_AGENT_ESCALATED triggers from any single agent session. Implementations SHOULD apply anomaly detection to frequent HEM triggers from the same agent.

12.2. Decision Signature Verification

Human principal decisions MUST be signature-verified (Section 7.7). A kernel that processes unsigned decisions enables an attacker who can inject messages into the delivery channel to resolve HEM events arbitrarily. The signature requirement over the concatenation of hem_id, principal_id, decision, and timestamp prevents replay of prior decisions.

12.3. Designation Chain Confidentiality

The designation chain in the HEM Escalation Request contains personal contact information for human principals. The escalation request MUST NOT be accessible to the requesting agent. The kernel routes the request; the agent receives only HEM_PENDING_ACTIVE as the response to its transition call.

12.4. TERMINATE Decision Integrity

The TERMINATE decision (Section 7.4) immediately ends the agent session and revokes the mandate JWT. The three-step atomicity ordering (Section 7.4) ensures that mandate revocation and MISSION_REVOKE_CASCADE initiation are both committed before any caller receives HEM_DECISION_ACCEPTED. Implementations MUST enforce this ordering guarantee. Race conditions in mandate revocation propagation MUST be mitigated by implementing the relying-party propagation fence described in [SOOS] Section 10.

12.5. Timeout Configuration

AUTO_APPROVE as a timeout disposition (Section 9.2) has significant security implications. The Cedar-gating requirement (Section 9.2) mitigates the risk of unauthorized approval but does not eliminate it. Implementations MUST additionally enforce the hard prohibition on AUTO_APPROVE for high-value transition classes at configuration load time. Implementations MUST log a configuration warning when AUTO_APPROVE is configured for any transition class. Auto-approval SHOULD NOT be permitted for governed objects classified as high-risk under EU AI Act Annex III.

12.6. Proximity Threshold Injection

An attacker with the ability to modify a governed object's proximity threshold configuration can trigger HEM at will. Threshold configuration MUST be protected by the same access controls as the governed object's mandate configuration. Changes to threshold configuration MUST be recorded in the Event Log with the identity of the principal making the change.

12.7. Mission Validity Monitoring

The mission validity monitoring described in Section 8.4 requires the kernel to receive reliable signals about MissionDeclaration phase changes. Implementations MUST ensure that the channel by which mission phase updates reach the kernel is authenticated and tamper-evident. An attacker who can inject false SUSPENDED or ABANDONED phase signals can trigger spurious session termination.

12.8. Escalation Request Signing

The kernel MUST sign the full HEM Escalation Request using Ed25519, using the same key model as the Mandate JWT. The signature MUST cover the canonical serialization of all fields in the escalation request other than kernel_signature itself. The resulting signature value is placed in the kernel_signature field (Section 6.1).

The verification key is available via the operator's JWKS endpoint, using the same key discovery mechanism as for Mandate JWT verification. Delivery intermediaries that are capable of verifying Ed25519 signatures MUST verify the kernel_signature before forwarding the escalation request. A delivery intermediary MUST NOT forward an escalation request on which signature verification fails.

Non-normative advisory: delivery channels that cannot perform Ed25519 verification (e.g., plain messaging channels without cryptographic processing capability) MAY forward without verification, but implementations using such channels SHOULD be aware that the escalation request integrity guarantee does not extend to the final delivery hop. Operators SHOULD prefer delivery channels capable of end-to-end verification.

13. Privacy Considerations

HEM escalation requests contain information about agent actions, governed object state, and agent-declared goals derived from the IDP. Implementors MUST ensure that escalation notifications delivered to human principals do not contain personally identifiable information beyond what is necessary for the principal to make an informed decision.

The idp_summary included in escalation requests (Section 6.1) MUST be filtered to remove any PII from the declared_goal.description field before delivery. The full IDP text MUST NOT be included in the delivery payload.

Trigger_detail minimisation: The trigger_detail array entries for HEM_PROXIMITY_TRIGGERED conditions MUST contain only the following fields: threshold_id, threshold_class, condition_met, triggered_at. The trigger_detail MUST NOT contain: personal data, Zone B content, traveller identity attributes beyond principal_id, raw telemetry data, or raw sensor readings. The kernel MUST enforce these minimisation requirements at escalation request generation time, before the request is constructed. Proximity detail that exceeds this enumeration MUST be retained in the kernel Event Log (where access is controlled) and MUST NOT be propagated to the escalation

request.

Designation chain contact information (principal contact objects) MUST be stored and transmitted in compliance with applicable data protection regulations. The contact objects MUST NOT be logged in plaintext; implementations SHOULD log only principal_id and delivery mechanism type in the Event Log.

The tension between Event Log immutability and data subject erasure rights is noted; this document defers to the SOOS Kernel Specification [SOOS] cryptographic erasure provisions.

14. EU AI Act Applicability

This section provides normative and advisory Article mappings. The normative mappings in this section are designed for direct reference in EU AI Act conformance documentation. This document does not constitute legal advice and makes no representation that conforming implementations satisfy any regulatory requirement.

Article	Requirement	HEM Implementation	Mapping
Art. 5	Prohibited AI practices	CAP Tier 0/1 CEE [I-D.sato-soos-cap]	Normative
Art. 12	Automatic logging	Event Log (Section 10); SAR [I-D.sato-soos-gar]; IDP pre-action intent [I-D.sato-soos-idp]. Event Log retention SHOULD be minimum 12 months. Operators may reference this section and [I-D.sato-soos-gar] Section 10 directly in conformance documentation.	Normative
Art. 13	Transparency 13(3)(b): AI system capabilities and limits 13(3)(e): Human oversight measures	PRD rationale_class and review_date (Section 5.6); DRR rationale_class (Section 7.6). DRR safety_basis REQUIRED for TERMINATE.	Advisory
Art. 14	Human oversight 14(1): Effective oversight by natural persons 14(4)(b): Awareness of tendency to over-rely on output 14(4)(c): Interpret output correctly 14(4)(d): Decide not to use output in any particular situation	HEM_PENDING + transition prohibition (Section 8); Designation chain and timeout model (Section 9). Five decision types (Section 7); DRR captures human reasoning. idp_summary in Escalation Request (Section 6.1); IDP CVR anomaly detection [I-D.sato-soos-idp]. HEM_PENDING transition prohibition (Section 8.1); TERMINATE decision type (Section 7.4).	Normative Normative Normative Normative

Note on Article 14(4)(d): This provision requires that human oversight persons be able to decide not to use the AI system output in any particular situation and not to be unduly influenced by it. The HEM_PENDING transition prohibition (Section 8.1) implements the technical stop capability; TERMINATE (Section 7.4) implements the irrevocable session closure. Both are kernel-enforced and non-bypassable by agent or application.

15. IANA Considerations

This document requests the creation of the following IANA registry:

Registry Name:
Human Escalation Mechanism Trigger Classes

Registration Procedure:
Standards Action [RFC8126]

Initial Values:

Trigger Class	Description
HEM_CEDAR_ROUTED	Cedar policy routes to HEM
HEM_AGENT_ESCALATED	Agent declares hem_urgency: REQUIRED in IDP
HEM_PROXIMITY_TRIGGERED	Proximity threshold crossed
HEM_MISSION_VALIDITY_FAILED	MissionDeclaration in terminal phase; active session detected
HEM_JURISDICTIONAL_CONFLICT	Kernel cannot resolve cross-jurisdiction legal conflict

This document requests the creation of the following IANA registry:

Registry Name:
Human Escalation Mechanism Decision Types

Registration Procedure:
Standards Action [RFC8126]

Initial Values:

Decision Type	Description
APPROVE	Approve requested action
APPROVE_WITH_CONSTRAINTS	Approve with Cedar context additions
REDIRECT	Approve alternative action
TERMINATE	Terminate agent session
DEFER	Extend timeout; defer decision
APPROVE_WITH_LEGAL_BASIS	Reserved; not yet operational. Kernel returns HEM_DECISION_TYPE_NOT_YET_OPERATIONAL if submitted.

This document requests the creation of the following IANA registry:

Registry Name:
Human Escalation Mechanism Timeout Dispositions

Registration Procedure:
Specification Required [RFC8126]

Initial Values:

Disposition	Description
ESCALATE_CHAIN	Escalate to next principal in chain
SUSPEND	Suspend governed object; await recovery
TERMINATE_SESSION	Apply TERMINATE decision semantics
AUTO_APPROVE	Cedar-gated approve; restricted use (see Section 9.2)

This document requests the creation of the following IANA registry:

Registry Name:
Human Escalation Mechanism Error Codes

Registration Procedure:
Specification Required [RFC8126]

Initial Values:

Error Code	Description
HEM_PENDING_ACTIVE	Transition rejected; HEM active
HEM_DECISION_INVALID	Unrecognized decision type
HEM_SIGNATURE_INVALID	Decision signature failed
HEM_PRINCIPAL_NOT_AUTHORIZED	Submitter not in chain
HEM_DEFER_LIMIT_EXCEEDED	Second DEFER from same principal on same hem_id
HEM_DECISION_REJECTED	Decision rejected; generic
HEM_MISSION_INVALID	Session terminated; mission entered terminal phase
HEM_PRD_MISSING	Cedar policy routes to HEM but lacks registered prd_id
HEM_CONSTRAINT_EXPIRED	APPROVE_WITH_CONSTRAINTS expiry window closed
HEM_REDIRECT_DENIED	Cedar DENY on REDIRECT action
HEM_MISSION_REF_MISMATCH_REJECTED	IDP mission_ref does not match session
HEM_DRR_REQUIRED	TERMINATE submitted without required DRR
HEM_AUTO_APPROVE_CEDAR_DENIED	AUTO_APPROVE not applied; Cedar returned DENY
HEM_AUTO_APPROVE_PROHIBITED	AUTO_APPROVE configured for high-value transition class
HEM_DECISION_TYPE_NOT_YET_OPERATIONAL	Decision type reserved; not yet operational

This document requests the creation of the following IANA registry:

Registry Name:
Human Escalation Mechanism PRD Rationale Classes

Registration Procedure:
Specification Required [RFC8126]

Initial Values:

Rationale Class	Description

REGULATORY	Required by applicable regulation
CONTRACTUAL	Required by contract or agreement
OPERATIONAL_RISK	Operational risk threshold requires human judgment
SAFETY	Safety requirement mandates oversight
LEGAL	Legal requirement (non-regulatory)
POLICY	Internal organizational policy

This document requests the creation of the following IANA registry:

Registry Name:
Human Escalation Mechanism DRR Rationale Classes

Registration Procedure:
Specification Required [RFC8126]

Initial Values:

Rationale Class	Description
REGULATORY_COMPLIANCE	Decision based on regulatory requirement
SAFETY_ASSESSMENT	Decision based on safety evaluation
MISSION_ALIGNMENT	Decision based on mission scope assessment
OPERATIONAL_JUDGMENT	Decision based on operational expertise
CONTRACTUAL_OBLIGATION	Decision required by contract
ETHICAL_CONSIDERATION	Decision based on ethical assessment
INSUFFICIENT_CONTEXT	Insufficient information to approve
ESCALATION_JUDGMENT	Decision reflects human escalation evaluation

This document requests the creation of the following IANA registry:

Registry Name:
Human Escalation Mechanism Denial Reason Classes

Registration Procedure:
Specification Required [RFC8126]

Initial Values:

Denial Reason	Description
CEDAR_POLICY_DENY	Cedar evaluation returned DENY
MANDATE_SCOPE_EXCEEDED	Action outside mandate scope
HEM_PENDING_ACTIVE	HEM is active for this session
MISSION_REF_MISMATCH	IDP mission_ref does not match session MissionDeclaration
CAP_PROHIBITION	CAP Tier 0 or Tier 1 prohibition applies
PRINCIPAL_NOT_AUTHORIZED	Principal not in designation chain

16. References

16.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate

Requirement Levels", BCP 14, RFC 2119,
DOI 10.17487/RFC2119, March 1997,
<https://www.rfc-editor.org/rfc/rfc2119>.

[RFC7519] Jones, M., Bradley, J., and N. Sakimura, "JSON Web Token (JWT)", RFC 7519, DOI 10.17487/RFC7519, May 2015, <https://www.rfc-editor.org/rfc/rfc7519>.

[RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <https://www.rfc-editor.org/rfc/rfc8126>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/rfc/rfc8174>.

[Cedar] Amazon Web Services, "Cedar Policy Language", <https://www.cedarpolicy.com/>.

[I-D.sato-soos-idp] Sato, T., "The Intent Declaration Primitive (IDP) for Agentic AI Systems", Work in Progress, Internet-Draft, draft-sato-soos-idp-01, May 2026, <https://datatracker.ietf.org/doc/draft-sato-soos-idp/>.

[I-D.sato-soos-gar] Sato, T., "The Governance Audit Record (GAR) for Agentic AI Systems", Work in Progress, Internet-Draft, draft-sato-soos-gar-00, May 2026, <https://datatracker.ietf.org/doc/draft-sato-soos-gar/>.

[I-D.sato-soos-cap] Sato, T., "The Constitutional AI Protocol (CAP) for Agentic AI Systems", Work in Progress, Internet-Draft, draft-sato-soos-cap-00, May 2026, <https://datatracker.ietf.org/doc/draft-sato-soos-cap/>.

16.2. Informative References

[EUAIA] European Union, "Regulation (EU) 2024/1689 of the European Parliament and of the Council laying down harmonised rules on artificial intelligence (Artificial Intelligence Act)", Official Journal of the European Union, 12 July 2024, <https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=OJ:L_202401689>.

[I-D.ietf-wimse-arch] Salowey, J., Rosomakho, Y., and H. Tschofenig, "Workload Identity in a Multi System Environment (WIMSE) Architecture", Work in Progress, Internet-Draft, draft-ietf-wimse-arch-07, March 2026, <https://datatracker.ietf.org/doc/draft-ietf-wimse-arch/>.

[I-D.klrc-aiagent-auth] Kasselman, P., Lombardo, J., Rosomakho, Y., Campbell, B., and N. Steele, "AI Agent Authentication and Authorization", Work in Progress, Internet-Draft, draft-klrc-aiagent-auth-01, March 2026, <https://datatracker.ietf.org/doc/draft-klrc-aiagent-auth/>.

[I-D.rosenberg-aiproto-cheq] Rosenberg, J., White, P., and C. Jennings, "CHEQ: A Protocol for Confirmation of AI Agent Decisions with Human in the Loop (HITL)", Work in Progress, Internet-Draft,

draft-rosenberg-aiproto-cheq-00, October 2025,
<https://datatracker.ietf.org/doc/
draft-rosenberg-aiproto-cheq/>.

[SOOS] Sato, T., "Sovereign Object OS -- Kernel Specification",
Work in Progress, Version 2, May 2026,
<https://activitytravel.pro/>.

[I-D.sato-soos-transition-graph]
Sato, T., "Transition Graph API for Agentic AI Systems",
Work in Progress, Internet-Draft,
draft-sato-soos-transition-graph-00, 2026 (forthcoming).

[I-D.sato-jones-soos-mandate]
Sato, T. and Jones, M., "Mandate JWT Profile for Agentic
AI Systems", Work in Progress, Internet-Draft,
draft-sato-jones-soos-mandate-00, 2026 (forthcoming).

[I-D.mcguinness-oauth-actor-profile]
McGuinness, K., "OAuth Actor Profile for AI Agents",
Work in Progress, Internet-Draft,
draft-mcguinness-oauth-actor-profile-00, 2026,
<https://datatracker.ietf.org/doc/
draft-mcguinness-oauth-actor-profile/>.

[I-D.mcguinness-oauth-mission-bound-authorization]
McGuinness, K., "Mission Bound Authorization",
Work in Progress, Internet-Draft,
draft-mcguinness-oauth-mission-bound-authorization-00,
2026,
<https://datatracker.ietf.org/doc/
draft-mcguinness-oauth-mission-bound-authorization/>.

Acknowledgments

The HEM design builds on the OVID mandate model and the Clawdrey Hepburn delegation framework developed by the OVID community, from which the five decision type vocabulary was adapted. The EU AI Act Article 14 analysis was developed in the context of the Sovereign Object OS specification [SOOS]. The CHEQ complementarity framing draws on review of [I-D.rosenberg-aiproto-cheq]. The designation chain and timeout model reflects operational experience with the TRAVELER_UNREACHABLE chains defined in the ATP booking lifecycle. Section 8.4 (Mission Validity During HEM_PENDING) and the Class 4 trigger class (Section 5.4, promoted to normative in this version) were developed following review of the McGuinness Mission Bound Authorization framework [I-D.mcguinness-oauth-mission-bound-authorization] and the mission governance model described in McGuinness (2026). The AAuth vs. HEM positioning statement in Section 1 was clarified following that review.

The Windley Loop -- the kernel-mediated cycle of context delivery, agent intent declaration, and Cedar enforcement before action execution -- is named in recognition of Phil Windley’s work on context and the Internet of My Things. The loop appears normatively in Section 5.1 and Section 11. The Policy Rationale Declaration (PRD, Section 5.6) and Decision Rationale Record (DRR, Section 7.6) were added in this version following completion of the full 29-item hem-01 agenda (S19, May 2026). The escalation request signing requirement (Section 12.8) reflects security review of delivery channel integrity properties.

Author’s Address

Tom Sato
MyAuberge K.K.
Chino, Nagano, Japan
Email: [tomsato@myauberge.jp]
URI: https://activitytravel.pro/