Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: 24 November 2026

T. Sato
MyAuberge K.K.

24 May 2026

The Agent Execution Protocol (AEP) for Agentic AI Systems
draft-sato-soos-aep-00

Abstract

AI agents operating on governed resources require a normative interface contract between their internal reasoning loop and the Governing Enforcement Component (GEC) that enforces authorization policy, records transitions to a tamper-evident Event Stream, and mediates access to Sovereign Object instances. Existing agent frameworks define no such contract. Agents submit actions without a normative delivery protocol for the state and permission context they act on; GECs enforce policy without a normative protocol for communicating denial rationale back to agents; human oversight is invoked without a normative session state that governs the resulting suspension.

This document defines the Agent Execution Protocol (AEP): the normative five-step loop -- SENSE, REASON, PLAN, ACT, OBSERVE -- that specifies how a governed AI agent interfaces with GEC services at each iteration. The AEP defines the Context Package delivered at SENSE, the GEC Query Interface exercised at PLAN, the Transition Request submitted at ACT, and the atomic GEC response received at OBSERVE. The AEP specifies two conformance modes -- Standard and Goal Execution Engine (GEE) -- and normatively integrates the Intent Declaration Primitive [I-D.sato-soos-idp], the Mandate JWT [I-D.sato-soos-mjwt], the Human Escalation Mechanism [I-D.sato-soos-hem], the Governance Audit Record [I-D.sato-soos-gar], the Constitutional AI Protocol [I-D.sato-soos-cap], and the Sovereign Object [I-D.sato-soos-sov] as components of a single governed execution architecture.

The REASON step is intentionally GEC-unspecified: the LLM reasoning engine is opaque to the protocol. The AEP is the transmission between the LLM engine and the GEC enforcement substrate.

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 24 November 2026.

Copyright Notice

Table of Contents

Appendix A. ATP Booking Object -- AEP Reference Walk-Through

1. Introduction

The IETF SOOS protocol family specifies governance primitives for agentic AI systems: the Intent Declaration Primitive (IDP) [I-D.sato-soos-idp] for per-transition intent declaration; the Human Escalation Mechanism (HEM) [I-D.sato-soos-hem] for human oversight; the Governance Audit Record (GAR) [I-D.sato-soos-gar] for tamper-evident audit; the Constitutional AI Protocol (CAP) [I-D.sato-soos-cap] for constitutional prohibition enforcement; the Sovereign Object (SOV) [I-D.sato-soos-sov] for the governed resource definition; and the Mandate JWT (MJWT) [I-D.sato-soos-mjwt] for agent authorization binding.

Each of these specifications describes a component. None describes the loop.

An AI agent governed by these protocols executes a repeating cycle: it receives context about the Sovereign Object instance it is operating on; it reasons about what action to take next; it plans that action using GEC query services; it submits the action as a Transition Request carrying an MJWT and an IDP; and it receives a GEC response that either permits the transition, denies it with enriched rationale, or suspends the session pending human decision. Then it does this again.

This cycle -- SENSE, REASON, PLAN, ACT, OBSERVE -- is the interface contract between the agent’s internal reasoning and the GEC enforcement substrate. Without a normative specification of this cycle, each of the component protocols specifies its own assumptions about the context in which it operates, without guaranteeing that those assumptions are satisfied. IDP assumes the agent has received a context package before reasoning; AEP normatively defines that delivery. HEM assumes the session enters a HEM_PENDING state that governs subsequent behavior; AEP normatively defines that state and its exit conditions. GAR assumes GEC events are generated at defined points in the session; AEP normatively defines those points.

This document fills the loop gap. The Agent Execution Protocol (AEP) is the normative specification of how a governed AI agent interfaces with GEC services at each iteration of its execution cycle. It is the horizontal interface contract that makes the component SOOS protocols composable into a single governed execution architecture.

The AEP is explicitly not a specification of agent intelligence. The REASON step is GEC-unspecified: what LLM, what prompting strategy, what reasoning architecture the agent uses is opaque to the protocol. The AEP specifies the inputs the agent MUST have received before reasoning (SENSE), the GEC services it MAY query to inform its plan (PLAN), the Transition Request structure it MUST submit to act (ACT), and the response it MUST process before acting again (OBSERVE). The intelligence between SENSE and ACT is the agent’s; the protocol is the interface.

The design principle of the AEP: the LLM is the engine. The AEP is the transmission. The GEC enforcement substrate is the road system. The road system specifies only the interface; it does not specify the vehicle.

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 Execution Protocol (AEP):
: The normative five-step loop -- SENSE, REASON, PLAN, ACT, OBSERVE -- defined in this document. Specifies the interface contract between a governed AI agent and GEC services at each iteration of the agent's execution cycle.

AEP Session:
: A bounded execution context initiated when the GEC delivers the first Context Package to an agent for a specific Sovereign Object instance under a specific Mandate JWT. An AEP Session has a unique session_id, a goal_session_id, and an iteration counter. An AEP Session is in one of three states: ACTIVE, HEM_PENDING, or CLOSED.

AEP Iteration:
: One complete traversal of the SENSE-REASON-PLAN-ACT-OBSERVE cycle within an AEP Session. Counted by aep_iteration, a monotonically increasing integer per session.

Context Package:
: The structured data object delivered by the GEC to the agent at the SENSE step. Contains the current SO Instance state, the agent's permission set, the current goal, memory items, Proximity Events, and HEM context if applicable.

Transition Request:
: The structured submission from an agent to the GEC at the ACT step, carrying a Mandate JWT [I-D.sato-soos-mjwt], a Cedar action identifier, and an Intent Declaration Primitive [I-D.sato-soos-idp].

GEC Query Interface:
: The set of read-only GEC services queried at the PLAN step: Transition Graph, Live Permission Map, and Compensating Action Catalogue.

HEM_PENDING:
: An AEP Session state in which the session has been suspended by the GEC pending a human principal decision under the Human Escalation Mechanism [I-D.sato-soos-hem]. HEM_PENDING is a valid, expected session state, not an error state.

Goal Execution Engine (GEE):
: An optional orchestration mode in which a GEC-provided engine inverts control, calling the agent's reason() function as a service within a GEC-driven loop. In GEE mode the agent MUST NOT call the GEC transition endpoint directly.

Proximity Event:
: A GEC-generated notification that a monitored condition is approaching a threshold value. Delivered within the Context Package at SENSE. Enables proactive agent behavior before threshold breach.

ReasoningOutput:
: The structured output produced by the agent at the REASON step in GEE mode, consumed by the GEE to construct the full Transition Request.

Governing Enforcement Component (GEC):
: As defined in [I-D.sato-soos-idp]: a runtime component that enforces authorization policy, records agent actions to a tamper-evident Event Stream, and mediates agent access to Sovereign Object instances.

Sovereign Object (SO):
: As defined in [I-D.sato-soos-sov]: a causally ordered, policy-governed, typed, living document that evolves through a predefined finite state space under GEC authority.

Mandate JWT (MJWT):
: As defined in [I-D.sato-soos-mjwt]: a WIMSE workload credential profile that grants an AI agent authority to perform a specified set of Cedar actions on a specific SO Instance under the oversight of a named human principal.

Intent Declaration Primitive (IDP):
: As defined in [I-D.sato-soos-idp]: a structured object produced by an agent at the ACT step declaring the action, goal context, reasoning basis, confidence, and alternatives considered.

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

Human Principal:
: A natural person who holds authority over an SO Instance, as identified in the Mandate JWT human_principal_id claim [I-D.sato-soos-mjwt].

RETRY_CONTINUATION:
: An IDP reasoning_basis type [I-D.sato-soos-idp] Section 4.3 used when an agent retries an action following a GEC DENY response, declaring the prior denial as the basis for the revised attempt.

3. Problem Statement

3.1. The Loop Gap

The six live SOOS drafts -- IDP, HEM, GAR, CAP, SOV, and MJWT -- collectively specify what must be declared, what must be escalated, what must be recorded, what is constitutionally prohibited, what resource is governed, and what credential binds the agent. Each draft describes a component; none describes the cycle in which those components operate.

The consequences of the missing loop specification are concrete:

(a) SENSE ambiguity. IDP Section 4.1 requires context_package_ref: a SHA-256 hash of the Context Package delivered to the agent before reasoning. Without a normative definition of the Context Package -- its schema, its trigger types, its GEC delivery semantics -- this field has no interoperable meaning. Implementations cannot verify that the IDP's reasoning_basis references are grounded in the same context the GEC delivered.

(b) PLAN unspecified. HEM Section 5 references the Windley Loop [Windley-Loop] as the planning gate that precedes the enforcement gate. SOV Section 7.1 defines SO state as a Cedar attribute available at evaluation time. Neither specifies the GEC Query Interface through which the agent discovers valid transition paths, live permissions, and compensating action availability before submitting a Transition Request.

(c) OBSERVE fragmented. IDP Section 6 specifies the enriched DENY response; MJWT Section 8.2 specifies denial codes; HEM Section 4 specifies the HEM_PENDING state entered at escalation. No document specifies how the agent receives and processes these responses as part of a single normative OBSERVE step, or how the RETRY_CONTINUATION reasoning basis type [I-D.sato-soos-idp] Section 4.3 is to be used in the subsequent ACT.

(d) Session state implicit. HEM Section 3.2 defines HEM_PENDING as a session state. GAR Section 6.1 defines SAR generation at session close. IDP Section 5.1 defines the Transition Request submission. No document defines the session itself: its initiation, its valid states, its event log markers, and its termination paths.

This document closes all four gaps.

3.2. Why Existing Agent Frameworks Are Insufficient

Agent frameworks such as LangGraph, AutoGen, CrewAI, and the emerging Model Context Protocol [MCP] define agent loop abstractions. None provides the governance properties required by the SOOS protocol family:

State-grounded SENSE. Existing frameworks do not deliver a cryptographically bound, GEC-signed snapshot of a Sovereign Object Instance’s current state to the agent before each iteration. The agent’s context is whatever the application layer provides.

Governed ACT. Existing frameworks submit tool calls and API invocations without verifying that the agent holds a valid Mandate JWT, that the Cedar action is within scope, and that the human principal linkage is intact. The GEC’s 5-step execution sequence [I-D.sato-soos-mjwt] Section 8 has no equivalent.

Non-suppressible OBSERVE. Existing frameworks may log agent actions; none enforces that the log is append-only, GEC-signed, and non-modifiable by the agent. The AEP OBSERVE step triggers the GAR [I-D.sato-soos-gar] recording that makes governance auditable.

Human oversight integration. Existing frameworks may pause for human input; none defines the HEM_PENDING session state, the five human decision types, the cascade revocation on TERMINATE, or the Progressive Trust signals generated by the escalation outcome.

3.3. AEP as Architectural Integrator

The AEP is the document that positions all other SOOS drafts as components of a single governed execution architecture. From the perspective of an AEP-conforming implementation:

- SOV defines what the agent operates on (the governed resource).
- MJWT defines the credential that binds the agent to that resource.
- IDP defines what the agent must declare before each ACT.
- CAP defines constitutional prohibitions evaluated before Cedar.
- HEM defines what happens when the GEC suspends the session.
- GAR defines the audit artifacts generated throughout the session.

The AEP specifies how these components interoperate at each step of the execution cycle. An implementation that conforms to AEP automatically satisfies the integration requirements assumed by each component draft.

4. The AEP Loop

4.1. Overview

The AEP defines five normative steps executed in strict order within each AEP Iteration.

```
Step 1 -- SENSE:    GEC delivers Context Package to agent.
Step 2 -- REASON:   Agent performs LLM-internal reasoning.
Step 3 -- PLAN:     Agent queries GEC Query Interface.
Step 4 -- ACT:      Agent submits Transition Request to GEC.
Step 5 -- OBSERVE:  Agent receives and processes GEC response.
```

After OBSERVE, the LOOP mechanism determines whether the next iteration begins (SENSE again) or the session closes.

The REASON step is intentionally GEC-unspecified. What LLM, reasoning strategy, prompting approach, or internal architecture the agent uses between SENSE and PLAN is opaque to this protocol. The AEP specifies only the inputs the agent MUST have received (SENSE), the services it MAY query (PLAN), the submission it MUST make (ACT), and the response it MUST process (OBSERVE).

A session in HEM_PENDING state has exited the normal 5-step cycle. Section 5.3 specifies the HEM_PENDING protocol. Resumption from HEM_PENDING re-enters the cycle at SENSE with trigger: HEM_RESOLUTION.

4.2. Step 1 -- SENSE

At SENSE, the GEC delivers a Context Package (Section 6) to the agent. The Context Package carries the current state of the SO Instance the agent is authorized to operate on, the agent's current permission set, the current goal context, memory items from prior iterations, Proximity Events approaching threshold, and HEM context if the session is resuming from HEM_PENDING.

SENSE requirements:

(a) The GEC MUST write an AEP_SENSE_DELIVERED Event Stream entry (Section 10.1) before delivering the Context Package. The entry MUST include the cp_hash of the Context Package to be delivered.

(b) The Context Package MUST be delivered by the GEC, not constructed by the agent. An agent MUST NOT construct or modify its own Context Package.

(c) The agent MUST NOT submit a Transition Request (ACT) without first having received an AEP_SENSE_DELIVERED Event Stream entry for the current session iteration. The IDP submitted at ACT MUST carry a context_package_ref matching the cp_hash of the most recently delivered Context Package.

(d) The trigger field in the Context Package (Section 6.2) declares why this SENSE delivery occurred. The agent MUST process the trigger semantics before reasoning.

(e) At SESSION_START, the Context Package carries the initial SO Instance state. On STATE_CHANGE, it carries the updated state following the prior ACT. On HEM_RESOLUTION, it carries the human principal's decision and the updated HEM context. On MANDATE_REVOCATION, it signals session termination.

4.3. Step 2 -- REASON

At REASON, the agent performs its internal LLM reasoning. The AEP

does not specify this step beyond its inputs and outputs.

Required inputs to REASON:
- The Context Package delivered at SENSE.
- All prior AEP_SENSE_DELIVERED and GEC response history for this session, as the agent's own context.

Required outputs from REASON (consumed at PLAN and ACT):
- The Cedar action the agent intends to request.
- A confidence value in the range [0.0, 1.0].
- A draft reasoning_basis for the IDP to be submitted at ACT.
- An escalation assessment for the IDP.
- Alternatives considered and uncertainty flags, per agent class requirements (Section 13).

REASON constraints:

(a) The agent MUST NOT submit a Transition Request for any Cedar action not included in its current Mandate JWT cedar_actions array [I-D.sato-soos-mjwt] Section 4.2.3. The Live Permission Map query at PLAN provides pre-computed authorization scope.

(b) When OBSERVE returned a DENY response in the prior iteration, the RETRY_CONTINUATION reasoning_basis type MUST be used in the IDP submitted at the next ACT (Section 9.4). The agent MUST NOT silently retry the same action with the same IDP.

(c) When SENSE trigger is MANDATE_REVOCATION, the agent MUST NOT enter the REASON step. It MUST proceed directly to graceful session shutdown (Section 5.4).

4.4. Step 3 -- PLAN

At PLAN, the agent queries GEC Query Interface services (Section 7) to inform its Transition Request before submission. The PLAN step is a read-only interaction with the GEC; it does not modify SO Instance state.

PLAN is the normative implementation of the Windley Loop [Windley-Loop]: the planning gate that precedes the enforcement gate. The Windley Loop principle is that an agent SHOULD verify the feasibility and authorization of a planned action before committing to it, rather than discovering infeasibility at ACT.

PLAN services:

(a) Transition Graph query (Section 7.1): the agent queries the GEC for valid Cedar-authorized transition paths from the current SO state to the agent's declared goal state. The returned graph enables the agent to select a viable action path before ACT.

(b) Live Permission Map query (Section 7.2): the agent queries the GEC for the Cedar residual policy set applicable to its current Mandate JWT and the current SO Instance state. The returned permitted_actions array MUST be used to validate the intended Cedar action before ACT.

(c) Compensating Action Catalogue query (Section 7.3): OPTIONAL. The agent MAY query available compensating transitions from the current state if the intended action fails. This query informs the compensating_action_assessed field in the IDP.

PLAN conformance:

Class 2 and Class 3 agents (Section 13) MUST query the Transition Graph at least once per AEP Session before submitting the first

Transition Request. Agents that deviate from the returned path MUST query the Transition Graph again before the deviation step.

4.5. Step 4 -- ACT

At ACT, the agent submits a Transition Request (Section 8) to the GEC. The Transition Request carries the agent's Mandate JWT, the Cedar action identifier, and the Intent Declaration Primitive.

ACT requirements:

(a) The agent MUST submit exactly one Mandate JWT per Transition Request. The Mandate JWT MUST carry the so_id of the SO Instance targeted by this Transition Request.

(b) The agent MUST submit a well-formed IDP [I-D.sato-soos-idp] with every Transition Request. Required IDP fields depend on the agent's class (Section 13).

(c) The agent MUST NOT submit a second Transition Request while awaiting the GEC response to the first. Concurrent transitions on the same SO Instance from the same agent are not permitted.

(d) The GEC MUST execute its transition execution sequence [I-D.sato-soos-mjwt] Section 8 before Cedar evaluation. The GEC MUST also evaluate CAP [I-D.sato-soos-cap] prohibitions before Cedar evaluation per [I-D.sato-soos-sov] Section 7.3.

(e) The transition is atomic: it either completes fully or does not occur at all. Partial execution is not a valid GEC state.

The ACT step is the only step at which the SO Instance state can change. SENSE, REASON, PLAN, and OBSERVE do not modify SO state.

4.6. Step 5 -- OBSERVE

At OBSERVE, the agent receives and processes the GEC response to its Transition Request. Three response types are defined (Section 9): PERMIT, DENY, and HEM_PENDING.

OBSERVE requirements:

(a) The agent MUST process the OBSERVE response before entering the next iteration (SENSE).

(b) On PERMIT: the agent receives the new SO Instance state, the updated Cedar residual, and the Event Stream entry reference. The session continues.

(c) On DENY: the agent receives an enriched DENY response [I-D.sato-soos-idp] Section 6 carrying the denial code and the IDP reasoning_basis fields that would change the decision. The agent MUST use RETRY_CONTINUATION reasoning basis in any subsequent IDP for the same action (Section 9.4).

(d) On HEM_PENDING: the session has entered HEM_PENDING state (Section 5.3). The agent MUST NOT submit further Transition Requests until HEM_PENDING is resolved.

(e) The agent MUST NOT silently discard or ignore a DENY response. The DENY and the IDP are recorded in the SO Instance Event Stream regardless of whether the transition succeeded. The agent's next reasoning cycle MUST account for the DENY.

4.7. LOOP Mechanism

After OBSERVE, the GEC evaluates whether the AEP Session continues.

The session continues (next SENSE) when:
- The OBSERVE response was PERMIT and the goal state is not yet reached.
- The OBSERVE response was DENY and the session mandate has not expired.
- HEM_PENDING has resolved with APPROVE, APPROVE_WITH_CONSTRAINTS, or REDIRECT.

The session closes when:
- The goal SO state is reached (closure_reason: GOAL_ACHIEVED).
- The Mandate JWT expires before goal completion (MANDATE_EXPIRED).
- The agent declares goal outcome (AGENT_DECLARED).
- The GEE closes the session (GEE_CLOSED).
- HEM resolves with TERMINATE (HEM_TERMINATED).
- A MANDATE_REVOCATION SENSE trigger is received (MANDATE_REVOKED).
- The GEC rejects the session (KERNEL_REJECTED).

On session close, the GEC MUST write AEP_SESSION_CLOSED (Section 10.2) and the GAR MUST generate a Session Audit Record [I-D.sato-soos-gar] Section 6.

State change at PERMIT triggers GEC re-injection: the GEC prepares the next Context Package for SENSE. The aep_iteration counter MUST be incremented.

5. Session Lifecycle

5.1. Session Initiation

An AEP Session is initiated when a human principal or an authorized agent requests that the GEC begin a governed session for a specific SO Instance under a specific Root Mandate JWT [I-D.sato-soos-mjwt] Section 6.1.

Session initiation:

(a) The GEC MUST verify the Root Mandate JWT per the 10-step verification protocol [I-D.sato-soos-mjwt] Section 8 before initiating the session.

(b) The GEC MUST assign a unique session_id (UUID v7 [RFC9562]) and a goal_session_id (UUID v7) for the session.

(c) The GEC MUST initialize the aep_iteration counter to 1.

(d) The GEC MUST deliver the first Context Package with trigger: SESSION_START. The GEC MUST write AEP_SENSE_DELIVERED before delivering this Context Package.

(e) The session enters ACTIVE state.

5.2. Active Session

An ACTIVE session is a session in which the AEP loop is executing. In ACTIVE state:

- The GEC MUST accept Transition Requests from the agent holding the session’s Mandate JWT.
- The GEC MUST deliver Context Packages at each SENSE.
- The GEC MUST process Proximity Events and deliver them within the Context Package when triggered.
- The GEC MAY transition the session to HEM_PENDING at any time a HEM trigger condition is met (Section 5.3).

5.3. HEM_PENDING Session State

HEM_PENDING is a valid, expected AEP Session state, not an error state. An agent that correctly identifies a decision outside its confident operating range and signals escalation is performing correctly under the AEP.

Entry into HEM_PENDING:

The GEC enters HEM_PENDING when any HEM trigger class fires [I-D.sato-soos-hem] Section 3:
- HEM_MANDATORY: Cedar returns DENY with hem_required: true.
- HEM_AGENT_ESCALATED: IDP carries escalation_assessment with hem_urgency: REQUIRED and Cedar evaluation returns PERMIT.
- HEM_PROXIMITY_TRIGGERED: a Proximity Event threshold is reached.

The GEC MUST write HEM_INVOKED to the SO Instance Event Stream before the session enters HEM_PENDING.

While HEM_PENDING:

(a) The GEC MUST reject Transition Requests from this session with SESSION_HEM_PENDING.
(b) Other agents' sessions on the same SO Instance are NOT affected. HEM_PENDING is scoped to this AEP Session.
(c) The SO Instance's state machine is NOT frozen. Other agents with appropriate Mandate JWTs MAY continue transitions on the same SO Instance.

Exit from HEM_PENDING:

HEM_PENDING is exited when the human principal submits a HEMDecision [I-D.sato-soos-hem] Section 5:
- APPROVE: the pending transition proceeds. The GEC delivers a new SENSE with trigger: HEM_RESOLUTION.
- APPROVE_WITH_CONSTRAINTS: as APPROVE, with additional Cedar fragments added to the session.
- REDIRECT: the pending transition is abandoned. The GEC delivers a new SENSE with trigger: HEM_RESOLUTION carrying the redirect instruction.
- TERMINATE: the session closes. AEP_SESSION_CLOSED written with closure_reason: HEM_TERMINATED.
- DEFER: the HEM_PENDING timeout is extended. The session remains in HEM_PENDING. The agent receives no new SENSE.

HEM_PENDING also exits on HEM_TIMEOUT if the timeout_at timestamp is reached before HEM_RESOLVED is committed. Timeout disposition depends on urgency per [I-D.sato-soos-hem] Section 6.3.

5.4. Session Closure

All session closure paths MUST generate an AEP_SESSION_CLOSED Event Stream entry (Section 10.2). The GEC MUST write AEP_SESSION_CLOSED on every closure path, including failures.

On MANDATE_REVOCATION trigger at SENSE:

The agent MUST NOT enter the REASON step. The agent MUST perform graceful shutdown: completing any in-progress Zone B operations, recording a final goal outcome declaration if the agent is in standard mode, and confirming session closure to the GEC. The GEC MUST write AEP_SESSION_CLOSED with closure_reason: MANDATE_REVOKED. In GEE mode, the GEE MUST NOT call agent.reason() after receiving a MANDATE_REVOCATION trigger.

After AEP_SESSION_CLOSED, the GAR [I-D.sato-soos-gar] generates a Session Audit Record for the closed session.

6. Context Package

6.1. Context Package Schema

The Context Package is the structured data object delivered by the GEC to the agent at SENSE. It is the agent's ground truth about the SO Instance state, permissions, goal, memory, and session context at the start of each AEP Iteration.

The following is the normative JSON structure of a Context Package.

```
{
  "cp_version":     "1.0",        ; REQUIRED. Schema version.
  "cp_id":          string,        ; REQUIRED. UUID v7. CP identifier.
  "cp_hash":        string,        ; REQUIRED. SHA-256 of canonical CP.
  "delivered_at":   string,        ; REQUIRED. ISO 8601 timestamp.
  "trigger":        string,        ; REQUIRED. See Section 6.2.
  "so":             object,        ; REQUIRED. See Section 6.3.
  "permissions":    object,        ; REQUIRED. See Section 6.4.
  "goal":           object,        ; REQUIRED. See Section 6.5.
  "memory":         object,        ; RECOMMENDED. See Section 6.6.
  "proximity_events": [object],    ; REQUIRED. May be empty. Sec 6.7.
  "hem_context":    object | null, ; REQUIRED. See Section 6.8.
  "agent":          object         ; REQUIRED. See Section 6.9.
}
```

The GEC MUST compute cp_hash as a SHA-256 hash over the canonical JSON serialization of the Context Package excluding the cp_hash field itself. The agent MUST record cp_hash as the context_package_ref in the IDP submitted at the next ACT.

6.2. Context Package Trigger Types

The trigger field declares the reason for this SENSE delivery. The following trigger values are defined:

SESSION_START:
: First Context Package delivered for this AEP Session.

STATE_CHANGE:
: The SO Instance has transitioned to a new state following the prior ACT. The so.current_state field reflects the new state.

PROXIMITY_EVENT:
: One or more Proximity Events in proximity_events have reached or exceeded their threshold.

HEM_RESOLUTION:
: The session is resuming from HEM_PENDING following a human principal decision. hem_context carries the HEMDecision.

MANDATE_REFRESH:
: The GEC has refreshed the agent's Mandate JWT. The permissions sub-object carries the updated mandate scope.

MANDATE_REVOCATION:
: The agent's Mandate JWT has been revoked. The agent MUST NOT enter REASON. The session is being terminated.

6.3. SO Sub-Object

The so sub-object carries the current state of the SO Instance.

```
{
  "so_id":           string,  ; REQUIRED. SO Instance UUID v7.
  "so_type_id":      string,  ; REQUIRED. SO Type identifier.
  "current_state":   string,  ; REQUIRED. Current state machine state.
  "current_phase":   string,  ; REQUIRED. SO lifecycle phase.
  "state_entered_at": string, ; REQUIRED. ISO 8601 timestamp.
  "event_log_head":  string,  ; REQUIRED. event_id of last entry.
  "zone_a_snapshot": object   ; REQUIRED. Current Zone A field values.
}
```

The zone_a_snapshot MUST reflect the Zone A state of the SO Instance as of event_log_head. Personal data MUST NOT be present in zone_a_snapshot per [I-D.sato-soos-sov] Section 4.3.1 (INV-ZA-1).

6.4. Permissions Sub-Object

The permissions sub-object carries the agent's current authorization scope derived from the Mandate JWT.

```
{
  "mandate_jwt_id":     string,   ; REQUIRED. MJWT jti.
  "mandate_expires_at": string,   ; REQUIRED. ISO 8601.
  "agent_class":        string,   ; REQUIRED. CLASS_1|CLASS_2|CLASS_3.
  "cedar_residual":     object,   ; REQUIRED. Residual Cedar policy set.
  "permitted_actions":  [string], ; REQUIRED. Cedar actions in scope.
  "forbidden_until":    [object]  ; OPTIONAL. Temporarily restricted actions.
}
```

Each entry in forbidden_until carries: action (string), reason (string), until (ISO 8601 | null).

The cedar_residual is the GEC-computed partial evaluation of Cedar policy against the current SO Instance state and the agent's Mandate JWT scope. Agents SHOULD use cedar_residual to inform REASON; however, Cedar evaluation at ACT is authoritative.

6.5. Goal Sub-Object

The goal sub-object carries the agent's current goal session context.

```
{
  "goal_session_id":    string,   ; REQUIRED. UUID v7.
  "declared_goal_state": string,  ; REQUIRED. Target SO state.
  "goal_step_current":  integer,  ; REQUIRED. Current step number.
  "path_to_goal":       [object], ; REQUIRED. TransitionSteps to goal.
  "path_confidence":    number,   ; REQUIRED. Float 0.0-1.0.
  "prior_idp_ref":      string    ; OPTIONAL. Prior iteration IDP UUID.
}
```

Each TransitionStep in path_to_goal contains: step (integer), from_state (string), action (string), to_state (string), authority_sufficient (boolean), hem_required (boolean).

path_to_goal MUST be GEC-computed from the current SO state to declared_goal_state using the SO Type's state machine and the agent's cedar_residual. The agent MUST NOT modify path_to_goal.

6.6. Memory Sub-Object

The memory sub-object provides the agent with persistent context across AEP Iterations.

```
{
  "episodic":                    [object], ; Prior iteration summaries.
  "active_constraints":          [object], ; Active session constraints.
```

```
  "compensating_actions_available": [object] ; Available compensations.
}
```

Episodic entries are GEC-generated summaries of prior iterations within this AEP Session. The episodic store does not extend across sessions; cross-session agent memory is outside the scope of this document.

active_constraints are additional Cedar policy fragments applied to this session, including fragments added by APPROVE_WITH_CONSTRAINTS HEM decisions [I-D.sato-soos-hem].

6.7. Proximity Events

The proximity_events array contains Proximity Event objects. An empty array indicates no events are currently active.

Each Proximity Event object:

```
{
  "condition_id":        string,  ; REQUIRED. Unique condition identifier.
  "condition_type":      string,  ; REQUIRED. See below.
  "current_value":       string,  ; REQUIRED. Current observed value.
  "threshold_value":     string,  ; REQUIRED. Threshold value.
  "proximity_pct":       number,  ; REQUIRED. Float 0.0-1.0.
  "estimated_trigger_at": string  ; OPTIONAL. ISO 8601.
}
```

Defined condition_type values:

HEM_TRIGGER_APPROACHING:
: A Cedar policy condition that will trigger mandatory HEM is being approached. proximity_pct = 1.0 causes HEM_PENDING entry.

MANDATE_EXPIRING:
: The agent's Mandate JWT exp claim is approaching.

STATE_DURATION_THRESHOLD:
: The SO Instance has been in its current state longer than a threshold declared in the SO Type.

ZONE_B_SENSOR_THRESHOLD:
: A sensor-linked Zone B attachment value is approaching a threshold declared in the SO Type.

MANDATE_REVOCATION_PENDING:
: A scheduled future mandate revocation is approaching. estimated_trigger_at carries the scheduled revocation timestamp.

6.8. HEM Context

The hem_context field carries a HEMContext object when trigger is HEM_RESOLUTION, or is null otherwise. The HEMContext schema is defined normatively in [I-D.sato-soos-hem] Section 4.

When trigger is HEM_RESOLUTION:

- hem_context MUST carry the full HEMContext including the human principal's HEMDecision.
- The agent MUST process hem_context before entering REASON.
- If HEMDecision is REDIRECT, the agent MUST update its declared goal state to match redirect_target_state.

6.9. Agent Sub-Object

The agent sub-object carries session metadata.

```
{
  "agent_provider_id": string,  ; REQUIRED. Party Registry ID.
  "agent_type":        string,  ; REQUIRED. Agent implementation type.
  "aep_iteration":     integer, ; REQUIRED. Current iteration number.
  "session_id":        string   ; REQUIRED. AEP Session UUID v7.
}
```

The aep_iteration counter MUST be strictly monotonically increasing within a session. Gaps are permitted; reversals are not.

7. GEC Query Interface (PLAN Step)

7.1. Transition Graph Query

The Transition Graph query returns the set of valid Cedar-authorized transition paths from the current SO state to the agent's declared goal state.

Request:

```
{
  "so_id":            string,  ; REQUIRED.
  "mandate_jwt_id":   string,  ; REQUIRED.
  "goal_state":       string   ; REQUIRED.
}
```

Response:

```
{
  "path_to_goal":    [TransitionStep], ; Computed at current state.
  "path_confidence": number,           ; Float 0.0-1.0.
  "blocked_actions": [object]          ; Actions blocked by Cedar.
}
```

The GEC MUST compute path_to_goal using the SO Type's state machine and the agent's Cedar residual. Blocked paths due to Cedar policy MUST be included in blocked_actions with the blocking policy fragment reference.

This is the normative implementation of the Windley Loop planning gate [Windley-Loop]: the agent discovers what it is authorized to do before committing to an action.

7.2. Live Permission Map

The Live Permission Map query returns the Cedar residual policy set applicable to the agent's current Mandate JWT and the current SO Instance state.

Request:

```
{
  "so_id":          string,  ; REQUIRED.
  "mandate_jwt_id": string   ; REQUIRED.
}
```

Response:

```
{
  "cedar_residual":    object,   ; Current residual policy set.
  "permitted_actions": [string], ; Cedar actions currently in scope.
  "forbidden_until":   [object]  ; Temporarily restricted actions.
}
```

The Live Permission Map reflects the current state of Cedar evaluation; cedar_residual from the Context Package may be stale if the SO Instance has transitioned since SENSE delivery. Agents SHOULD query the Live Permission Map when planning an action that depends on state entered since the last SENSE.

7.3. Compensating Action Catalogue

The Compensating Action Catalogue query returns available compensating transitions from the current SO state.

Request:

```
{
  "so_id":          string, ; REQUIRED.
  "mandate_jwt_id": string  ; REQUIRED.
}
```

Response:

```
{
  "compensating_actions": [
    {
      "from_state":      string,  ; State this compensates from.
      "compensating_action": string, ; Cedar action available.
      "to_state":        string,  ; Resulting state.
      "authority_sufficient": boolean ; In agent's mandate scope.
    }
  ]
}
```

The Compensating Action Catalogue is used by the agent to assess the compensating_action_assessed field in the IDP and to plan rollback paths before a high-risk ACT.

8. Transition Request (ACT Step)

8.1. Transition Request Structure

The Transition Request is submitted by the agent to the GEC at ACT.

```
{
  "mandate_jwt":  string, ; REQUIRED. Compact-serialized MJWT.
  "cedar_action": string, ; REQUIRED. Cedar action identifier.
  "idp":          object  ; REQUIRED. Intent Declaration Primitive.
}
```

The cedar_action MUST be a string from the agent's Mandate JWT cedar_actions array. The idp MUST be a well-formed IDP per [I-D.sato-soos-idp] with all fields required for the agent's class present (Section 13).

8.2. GEC Execution Sequence

On receiving a Transition Request, the GEC MUST execute the following sequence in strict order. Failure at any step MUST abort the transition.

Step 1 -- MJWT Verification.
Execute the 10-step MJWT verification protocol [I-D.sato-soos-mjwt] Section 8. Failure: DENY with appropriate MJWT deny code.

Step 2 -- CAP Evaluation.
Evaluate CAP Tier 0 (constitutional) and Tier 1 (jurisdictional) prohibitions [I-D.sato-soos-cap]. A CAP prohibition results in immediate DENY regardless of Cedar result.

Step 3 -- Cedar Evaluation.
Evaluate Cedar policy set with SO Instance state attributes [I-D.sato-soos-sov] Section 7 and IDP intent attributes [I-D.sato-soos-idp] Section 5. DENY: return enriched DENY.

DENY with hem_required: true: route to HEM (Section 5.3).
PERMIT: proceed.

Step 4 -- State Machine Validation.
Verify the edge (current_state, cedar_action) exists in the SO Type's state machine. Failure: DENY.

Step 5 -- Event Stream Write.
Append a StateTransitionEvent to the SO Instance Event Stream. The GEC MUST sign this entry. The transition is NOT complete until this write succeeds.

Step 6 -- Event Emission.
The GEC MAY emit asynchronous notifications. This step MUST NOT block the transition response. Failure of event emission MUST NOT affect transition completeness.

The transition is atomic: either all steps complete or the transition does not occur.

8.3. IDP Submission at ACT

The IDP is submitted as part of every Transition Request. The IDP is permanently recorded in the SO Instance Event Stream as part of the StateTransitionEvent. The IDP is recorded even for Cedar DENY outcomes: failed transition attempts are part of the permanent audit trail.

The GEC MUST record the IDP verbatim as submitted. The GEC MUST NOT validate IDP content beyond schema conformance. Cedar does not evaluate IDP content; it is the agent's permanent reasoning declaration, not an authorization input.

9. GEC Response (OBSERVE Step)

9.1. PERMIT Response

A PERMIT response indicates the transition completed successfully.

```
{
  "result":              "PERMIT",
  "new_state":           string,  ; New SO current_state.
  "new_phase":           string,  ; New SO lifecycle phase.
  "event_stream_entry_id": string, ; UUID v7 of committed entry.
  "updated_cedar_residual": object, ; Updated residual policy.
  "aep_iteration":       integer  ; Current iteration number.
}
```

On PERMIT, the GEC prepares the next Context Package (trigger: STATE_CHANGE) and the LOOP mechanism initiates the next SENSE. The aep_iteration counter is incremented.

9.2. DENY Response

A DENY response indicates the transition was rejected.

```
{
  "result":         "DENY",
  "deny_code":      string,   ; Deny code per [I-D.sato-soos-idp]
                              ; Section 6 and [I-D.sato-soos-mjwt]
                              ; Section 8.2.
  "deny_reason":    string,   ; Human-readable denial reason.
  "idp_ref":        string,   ; idp_id of the rejected IDP.
  "enrichment":     object,   ; IDP fields that would change result.
  "aep_iteration":  integer   ; Current iteration number.
```

```
}
```

The enrichment field carries the specific IDP reasoning_basis references that, if changed, would produce a PERMIT. This is the mechanism by which the GEC guides the agent's next REASON step.

On DENY, the SO Instance state does NOT change. The GEC delivers the next Context Package (trigger: STATE_CHANGE is NOT fired; the trigger is omitted and the agent re-evaluates from the current state).

9.3. HEM_PENDING Response

An HEM_PENDING response indicates the session has entered HEM_PENDING state.

```
{
  "result":        "HEM_PENDING",
  "hem_id":        string, ; UUID v7 of the HEM invocation.
  "trigger_class": string, ; HEM trigger class.
  "urgency":       string, ; ADVISORY|RECOMMENDED|REQUIRED.
  "timeout_at":    string  ; ISO 8601 | null.
}
```

On HEM_PENDING, the agent MUST NOT submit further Transition Requests. The GEC will deliver a new SENSE with trigger: HEM_RESOLUTION when the human principal provides a decision.

9.4. RETRY_CONTINUATION Handling

When an agent retries an action following a GEC DENY response, the IDP submitted at the next ACT MUST include a reasoning_basis entry with ref_type: RETRY_CONTINUATION referencing the prior DENY.

```
{
  "ref_type":     "RETRY_CONTINUATION",
  "ref_id":       string, ; idp_id of the rejected prior IDP.
  "content_hash": string, ; SHA-256 of the DENY response received.
  "weight":       "primary"
}
```

The RETRY_CONTINUATION entry MUST reference at least one additional reasoning_basis entry explaining what changed between the rejected attempt and the retry. Silent retry -- submitting the same action with the same reasoning without acknowledging the prior DENY -- is not permitted. The GEC MUST detect silent retry patterns via the prior_denial_count attribute [I-D.sato-soos-idp] Section 7.1.

10. AEP Event Log Markers

10.1. AEP_SENSE_DELIVERED

The GEC MUST write this entry to the SO Instance Event Stream immediately before delivering each Context Package.

```
{
  "event_type":      "AEP_SENSE_DELIVERED",
  "event_id":        string,  ; UUID v7.
  "prior_event_id":  string,  ; Previous entry event_id.
  "occurred_at":     string,  ; ISO 8601.
  "so_id":           string,  ; SO Instance UUID v7.
  "session_id":      string,  ; AEP Session UUID v7.
  "aep_iteration":   integer, ; Current iteration number.
  "cp_id":           string,  ; Context Package UUID v7.
  "cp_hash":         string,  ; SHA-256 of Context Package.
```

```
  "trigger":         string,  ; Context Package trigger type.
  "agent_id":        string,  ; Agent Party Registry identifier.
  "goal_session_id": string,  ; Goal session UUID v7.
  "gec_signature":   string   ; GEC Ed25519 signature.
}
```

The gec_signature MUST be computed over the canonical JSON of this entry. AEP_SENSE_DELIVERED MUST be committed before the Context Package is delivered to the agent.

10.2. AEP_SESSION_CLOSED

The GEC MUST write this entry on every session termination path.

```
{
  "event_type":       "AEP_SESSION_CLOSED",
  "event_id":         string,  ; UUID v7.
  "prior_event_id":   string,  ; Previous entry event_id.
  "occurred_at":      string,  ; ISO 8601.
  "so_id":            string,  ; SO Instance UUID v7.
  "session_id":       string,  ; AEP Session UUID v7.
  "goal_session_id":  string,  ; Goal session UUID v7.
  "total_iterations": integer, ; Count of completed AEP Iterations.
  "final_state":      string,  ; SO state at closure.
  "goal_achieved":    boolean, ; Whether declared goal state reached.
  "closure_reason":   string,  ; See below.
  "agent_id":         string,  ; Agent Party Registry identifier.
  "gec_signature":    string   ; GEC Ed25519 signature.
}
```

closure_reason values:

```
GOAL_ACHIEVED:     Agent reached declared goal state.
MANDATE_EXPIRED:   Mandate JWT expired before goal completion.
AGENT_DECLARED:    Agent declared goal outcome explicitly.
GEE_CLOSED:        GEE closed session in GEE mode.
HEM_TERMINATED:    Human principal TERMINATE decision at HEM.
KERNEL_REJECTED:   GEC rejected the session (policy violation).
MANDATE_REVOKED:   Mandate JWT revoked during active session.
```

After AEP_SESSION_CLOSED is committed, the GAR MUST generate a Session Audit Record [I-D.sato-soos-gar] Section 6 for the closed session.

11. Standard Mode Conformance

The following conformance rules apply in Standard Mode, in which the agent implements the full SENSE-REASON-PLAN-ACT-OBSERVE loop directly.

Rule	Requirement	Violation
CONF-AEP-01	Agent MUST NOT submit ACT before receiving AEP_SENSE_DELIVERED for the current iteration. IDP context_package_ref MUST match cp_hash of the most recent Context Package.	REJECT
CONF-AEP-02	Class 2 and Class 3 agents MUST query Transition Graph at least once per session before first ACT. Agents deviating from returned path MUST re-query before the deviation step.	REJECT
CONF-AEP-03	IDP submitted at ACT MUST be well-	REJECT

	formed per [I-D.sato-soos-idp] Section 4. Required fields per agent class (Section 13) MUST be present.	
CONF-AEP-04	Agent MUST NOT submit a second ACT while awaiting response to the first. Concurrent transitions on the same SO Instance from the same agent session are not permitted.	REJECT
CONF-AEP-05	All IDPs in a session MUST carry the same goal_session_id. An IDP with a mismatched goal_session_id MUST be rejected.	REJECT
CONF-AEP-06	aep_iteration MUST be strictly monotonically increasing within a session. Gaps are permitted; reversals are not.	LOG
CONF-AEP-07	Agent MUST use RETRY_CONTINUATION reasoning basis after a DENY response for the same action. Silent retry is not permitted.	LOG
CONF-AEP-08	Agent MUST NOT enter REASON on a MANDATE_REVOCATION trigger. Session MUST proceed to graceful shutdown.	REJECT

HEM conformance rules (from [I-D.sato-soos-hem]):

CONF-HEM-01	GEC MUST write HEM_INVOKED before session enters HEM_PENDING.	REJECT
CONF-HEM-02	Session in HEM_PENDING MUST NOT accept ACT from the same agent. GEC MUST reject with SESSION_HEM_PENDING.	REJECT
CONF-HEM-03	HEM_RESOLVED MUST carry valid principal_signature from a human Party Registry principal.	REJECT
CONF-HEM-04	Agent MUST NOT submit HEMDecision for any session, including its own.	REJECT + CONFORMANCE _VIOLATION
CONF-HEM-05	GEC MUST write HEM_TIMEOUT if timeout_at reached before HEM_RESOLVED.	LOG
CONF-HEM-06	After non-TERMINATE HEM_RESOLVED, GEC MUST deliver new SENSE with trigger: HEM_RESOLUTION before agent may submit next ACT.	REJECT
CONF-HEM-07	available_decisions in HEMContext MUST be GEC-computed, not agent-supplied.	REJECT

12. GEE Orchestration Mode

12.1. GEE Overview

In GEE (Goal Execution Engine) Orchestration Mode, a GEC-provided engine inverts control: the agent exposes a single reasoning function and the GEE drives the AEP loop. The GEE constructs

Context Packages, calls agent.reason(), constructs the full Transition Request including the IDP, and submits it to the GEC.

In GEE mode, the agent MUST NOT call the GEC transition endpoint directly. The GEE is the sole submitter of Transition Requests.

GEE mode is appropriate when:
- The agent LLM is being called as a function within a larger orchestrated workflow.
- The operator requires uniform IDP construction across multiple agent implementations.
- The operator requires GEE-level Progressive Trust scoring independent of agent-level IDP claims.

12.2. Agent Reasoning Interface

In GEE mode the agent MUST implement the following function:

```
reason(context_package: ContextPackage) -> ReasoningOutput
```

ReasoningOutput:

```
{
  "selected_action":          string,   ; Cedar action identifier.
  "confidence":               number,   ; Float 0.0-1.0.
  "intent_summary":           string,   ; Human-readable summary.
  "reasoning_basis":          [object], ; ReasoningRef array.
  "alternatives_considered":  [object], ; AlternativeRef array.
  "uncertainty_flags":        [string], ; UncertaintyFlag array.
  "escalation_assessment":    object,   ; EscalationAssessment.
  "agent_recommends_replan":  boolean,  ; Signal to GEE to replan.
  "replan_reason":            string    ; Replan explanation.
}
```

The GEE constructs the full IDP from ReasoningOutput plus session metadata (session_id, goal_session_id, aep_iteration, so_uuid, context_package_ref) that the GEE manages.

12.3. GEE Conformance

Rule	Requirement	Violation
CONF-GEE-01	GEE MUST deliver normative Context Package before calling agent.reason().	REJECT
CONF-GEE-02	GEE MUST write AEP_SENSE_DELIVERED before delivering Context Package to agent.	REJECT
CONF-GEE-03	GEE MUST construct complete IDP from ReasoningOutput before submitting Transition Request.	REJECT
CONF-GEE-04	GEE MUST handle DENY by re-calling agent.reason() with enriched denial context or invoking HEM. Silent retry is not permitted. On MANDATE_REVOCATION trigger, GEE MUST terminate loop and write AEP_SESSION_CLOSED with closure_reason: MANDATE_REVOKED. GEE MUST NOT call agent.reason() after revocation.	REJECT
CONF-GEE-05	GEE MUST write AEP_SESSION_CLOSED on all termination paths, including failures.	REJECT

CONF-GEE-06	Agents operating in GEE mode MUST NOT call GEC transition endpoint directly.	REJECT

13. Agent Class Model

The AEP defines three Agent Classes that determine the required field set for IDPs submitted at ACT. Agent Class is declared in the Mandate JWT and reflects the level of autonomous authority granted to the agent.

Class 1 -- Basic Agent:
Agents with narrow, well-defined operational scope. Minimal IDP required fields. Typically used for single-action or single-step automations where human oversight is maintained through Cedar policy constraints rather than escalation assessment.

Class 2 -- Standard Agent:
Agents with multi-step operational scope operating within established patterns. Required fields include goal_ref, confidence, and reasoning_basis. These agents are expected to provide auditable reasoning but are not required to articulate alternatives considered or uncertainty flags at each step.

Class 3 -- Autonomous Agent:
Agents with broad operational scope operating with elevated autonomy. All IDP fields are required, including alternatives_considered, uncertainty_flags, and escalation_assessment. These agents are expected to proactively signal escalation when approaching their confident operating boundary.

Agent Class determines IDP required fields as specified in [I-D.sato-soos-idp] Section 4.4.

Confidence to autonomy level mapping:

Confidence Range	Autonomy Level	Cedar Policy Implications
0.0 - 0.59	UNCERTAIN	Cedar MAY require mandatory HEM before permitting Class 2/3 actions.
0.60 - 0.79	STANDARD	Standard Cedar evaluation.
0.80 - 0.89	HIGH	Cedar MAY permit elevated-autonomy actions otherwise restricted.
0.90 - 1.00	VERIFIED	Cedar MAY permit actions otherwise gated behind mandate elevation, subject to policy.

Agents MUST NOT treat the confidence field as a mechanism to bypass Cedar policy. Systematic overconfidence -- high declared confidence followed by frequent DENYs or HEM invocations -- MUST be detectable via the Progressive Trust model [I-D.sato-soos-pt] and MUST result in authority review.

14. Relationship to Other SOOS Drafts

SOV [I-D.sato-soos-sov]:
The Sovereign Object is the governed resource that the AEP loop operates on. The so sub-object of the Context Package (Section 6.3) carries the current SO Instance state at each SENSE. The zone_a_snapshot reflects Zone A as of event_log_head. The AEP session is scoped to a single SO Instance per Mandate JWT.

MJWT [I-D.sato-soos-mjwt]:
The Mandate JWT is verified at Step 1 of the GEC execution

sequence (Section 8.2) for every Transition Request. The MJWT so_id MUST match the SO Instance in the Context Package. The permissions sub-object (Section 6.4) is derived from the MJWT. The MJWT cedar_actions array bounds the permitted_actions set.

IDP [I-D.sato-soos-idp]:
The IDP is submitted at every ACT step. The IDP's context_package_ref binds each intent declaration to the SENSE delivery that preceded it. RETRY_CONTINUATION (Section 9.4) is the IDP mechanism for acknowledged denial-and-retry. Agent Class IDP requirements (Section 13) derive from IDP Section 4.4.

HEM [I-D.sato-soos-hem]:
HEM is invoked at Step 3 of the GEC execution sequence (Section 8.2) when Cedar returns DENY with hem_required: true. HEM_PENDING is an AEP Session state (Section 5.3). The five HEM decision types determine AEP loop resumption or closure. HEM_RESOLUTION is an AEP Context Package trigger type.

GAR [I-D.sato-soos-gar]:
Every AEP_SENSE_DELIVERED and AEP_SESSION_CLOSED entry generates a GAR Type 1 self-audit entry. At session close, GAR generates a Session Audit Record (SAR) covering all governance events in the session. The SAR includes the complete IDP chain, delegation_chain from MJWTs, and HEM event history.

CAP [I-D.sato-soos-cap]:
CAP prohibitions are evaluated at Step 2 of the GEC execution sequence (Section 8.2), before Cedar evaluation. A CAP Tier 0 or Tier 1 prohibition produces an immediate DENY regardless of Cedar result. CAP evaluation is transparent to the AEP loop; the agent receives a DENY at OBSERVE.

15. Security Considerations

The AEP loop is the outermost interface of the SOOS governance architecture. Its security properties derive from the security properties of the component protocols it integrates.

Context Package integrity. The cp_hash field enables the agent to detect tampering with the delivered Context Package. The GEC MUST sign AEP_SENSE_DELIVERED before delivery. An agent that receives a Context Package whose cp_hash does not match the AEP_SENSE_DELIVERED entry MUST NOT proceed to REASON. It MUST report the discrepancy as a critical security event.

SENSE injection. Malicious content in Zone B attachments MUST NOT be included in the zone_a_snapshot delivered at SENSE. Zone A Invariant INV-ZA-1 [I-D.sato-soos-sov] Section 4.3.1 prohibits personal data in Zone A; implementations MUST also ensure that Zone B content cannot be injected into the zone_a_snapshot to influence agent reasoning with unsanctioned content.

REASON opacity. The AEP intentionally makes REASON opaque to the GEC. This design property means the GEC cannot verify the correctness of the agent's reasoning -- only that the resulting Transition Request is authorized. The IDP's reasoning_basis, confidence, and uncertainty_flags are the agent's self-attestation; their forensic value depends on the agent's integrity. Progressive Trust scoring [I-D.sato-soos-pt] provides a longitudinal signal on reasoning calibration.

ACT atomicity. The 5-step GEC execution sequence (Section 8.2) MUST be atomic. Partial execution creates governance gaps: a transition that is Cedar-permitted but not Event-Stream-committed

is unauditable. Implementations MUST ensure that Step 5 (Event Stream Write) succeeds or the transition is aborted.

Silent retry detection. CONF-AEP-07 requires RETRY_CONTINUATION on denial-and-retry. The GEC MUST track prior_denial_count [I-D.sato-soos-idp] Section 7.1 to detect agents that submit repeated identical Transition Requests without acknowledging prior denials. Systematic silent retry is an indicator of authorization bypass attempts.

HEM_PENDING integrity. While a session is in HEM_PENDING, the GEC MUST reject Transition Requests from that session. An implementation that permits transitions from an HEM_PENDING session violates INV-12 and creates an authorization bypass vulnerability.

16. Privacy Considerations

The Context Package carries zone_a_snapshot. Zone A Invariant INV-ZA-1 [I-D.sato-soos-sov] prohibits personal data in Zone A. Implementations MUST verify this invariant before including zone_a_snapshot in the Context Package.

The goal sub-object (Section 6.5) carries declared_goal_state, which may reveal the human principal's intentions for the SO Instance. Access to goal context MUST be controlled by Cedar policy.

The episodic memory sub-object (Section 6.6) may contain references to prior agent actions. While episodic entries MUST NOT contain personal data directly, they may contain identifiers that correlate to personal data in Zone B. Implementations MUST apply appropriate access controls to episodic memory entries.

AEP_SENSE_DELIVERED and AEP_SESSION_CLOSED entries in the SO Instance Event Stream carry agent_id and session_id values. These entries may constitute personal data under GDPR Article 4(1) [GDPR] and APPI Article 2 [APPI] where the agent is associated with an identifiable natural person. Implementations MUST apply appropriate access controls to Event Stream queries.

17. IANA Considerations

17.1. AEP Context Package Trigger Registry

Registry name: Agent Execution Protocol Context Package Trigger Registry
Registration procedure: Specification Required.

Initial registrations:

Trigger Value	Description
SESSION_START	First CP delivered for a new AEP Session.
STATE_CHANGE	SO Instance transitioned since last SENSE.
PROXIMITY_EVENT	One or more Proximity Events at threshold.
HEM_RESOLUTION	Session resuming from HEM_PENDING.
MANDATE_REFRESH	Mandate JWT refreshed.
MANDATE_REVOCATION	Mandate JWT revoked; session terminating.

17.2. AEP Session Closure Reason Registry

Registry name: Agent Execution Protocol Session Closure Reason Registry
Registration procedure: Specification Required.

Initial registrations:

Closure Reason	Description
GOAL_ACHIEVED	Agent reached declared goal state.
MANDATE_EXPIRED	Mandate JWT expired before goal completion.
AGENT_DECLARED	Agent declared goal outcome explicitly.
GEE_CLOSED	GEE closed the session in GEE mode.
HEM_TERMINATED	Human principal TERMINATE decision at HEM.
KERNEL_REJECTED	GEC rejected the session.
MANDATE_REVOKED	Mandate JWT revoked during active session.

17.3. AEP Proximity Event Condition Type Registry

Registry name: Agent Execution Protocol Proximity Event Condition Type Registry
Registration procedure: Specification Required.

Initial registrations: As listed in Section 6.7.

17.4. AEP Agent Class Registry

Registry name: Agent Execution Protocol Agent Class Registry
Registration procedure: Standards Action.

Initial registrations:

Class Value	Description
CLASS_1	Basic Agent.
CLASS_2	Standard Agent.
CLASS_3	Autonomous Agent.

18. References

18.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC7519] Jones, M., Bradley, J., and N. Sakimura, "JSON Web Token (JWT)", RFC 7519, May 2015.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, May 2017.

[RFC8037] Liusvaara, I., "CFRG Elliptic Curves for JOSE", RFC 8037, January 2017.

[RFC9562] Davis, B., Peabody, C., and P. Leach, "Universally Unique IDentifiers (UUIDs)", RFC 9562, May 2024.

[RFC6234] Eastlake, D. and T. Hansen, "US Secure Hash Algorithms (SHA and SHA-based HMAC and HKDF)", RFC 6234, May 2011.

[Cedar] Amazon Web Services, "Cedar Policy Language Specification", https://docs.cedarpolicy.com/

[I-D.sato-soos-idp] Sato, T., "The Intent Declaration Primitive (IDP) for Agentic AI Systems", draft-sato-soos-idp-03, May 2026.

[I-D.sato-soos-hem] Sato, T., "The Human Escalation Mechanism (HEM) for Agentic AI Systems", draft-sato-soos-hem-01, May 2026.

[I-D.sato-soos-gar] Sato, T., "Governance Audit Record (GAR) for Agentic AI Systems", draft-sato-soos-gar-01, May 2026.

[I-D.sato-soos-cap]
Sato, T., "Constitutional AI Protocol (CAP) for Agentic AI Systems", draft-sato-soos-cap-00, May 2026.

[I-D.sato-soos-sov]
Sato, T., "The Sovereign Object (SOV) for Agentic AI Systems", draft-sato-soos-sov-00, May 2026.

[I-D.sato-soos-mjwt]
Sato, T., "The Mandate JWT (MJWT) for Agentic AI Systems", draft-sato-soos-mjwt-00, May 2026.

[I-D.ietf-wimse-arch]
Salomoni, D., et al., "WIMSE Architecture", draft-ietf-wimse-arch, work in progress.

[GDPR]
European Parliament, "General Data Protection Regulation", Regulation (EU) 2016/679, April 2016.

[APPI]
Government of Japan, "Act on the Protection of Personal Information", Act No. 57 of 2003, as amended.

18.2. Informative References

[I-D.sato-soos-pt]
Sato, T., "Progressive Trust (PT) for Agentic AI Systems", draft-sato-soos-pt-00, forthcoming.

[I-D.sato-soos-faip]
Sato, T., "Federated Agent Intelligence Protocol (FAIP)", draft-sato-soos-faip-00, forthcoming.

[I-D.sato-soos-mad]
Sato, T., "Multi-Agent Delegation (MAD) for Agentic AI Systems", draft-sato-soos-mad-00, forthcoming.

[I-D.ietf-scitt-architecture]
Birkholz, H., et al., "An Architecture for Trustworthy and Transparent Digital Supply Chains", draft-ietf-scitt-architecture, work in progress.

[I-D.mcguinness-oauth-actor-profile]
McGuinness, K., et al., "OAuth Actor Profile", draft-mcguinness-oauth-actor-profile-00, 2026.

[I-D.mcguinness-oauth-mission-bound-authorization]
McGuinness, K., et al., "Mission Bound Authorization", draft-mcguinness-oauth-mission-bound-authorization-00, 2026.

[Windley-Loop]
Windley, P., "The Windley Loop: Planning and Enforcement Gates in Agentic AI Authorization", personal communication, 2025.

[EUAIA]
European Parliament, "Artificial Intelligence Act", Regulation (EU) 2024/1689, June 2024.

[MCP]
Anthropic, "Model Context Protocol", https://modelcontextprotocol.io/, 2024.

Appendix A. ATP Booking Object -- AEP Reference Walk-Through

The ATP Booking Object is the reference implementation of the

Sovereign Object primitive [I-D.sato-soos-sov] Appendix A. This walk-through illustrates a single AEP Iteration for the Azusa Journey scenario: an OTA booking agent advancing a booking from CONFIRMED to PRE_ACTIVITY.

A.1. SENSE

The GEC delivers a Context Package with trigger: SESSION_START.
The so sub-object carries:
- so_id: "019547ab-1234-7abc-8def-000000000099"
- so_type_id: "atp/booking-object/1.0"
- current_state: "CONFIRMED"
- current_phase: "ACTIVE"
- zone_a_snapshot: { booking_reference: "MYA-2026-04521",
  activity_id: "PH-TRAIL-001",
  journey_date: "2026-06-15" }

The permissions sub-object carries:
- permitted_actions: ["atp:booking:confirm", "atp:booking:cancel",
  "atp:booking:pre_activity_open",
  "atp:booking:suspend"]
- mandate_ceiling: 2
- agent_class: "CLASS_2"

A.2. REASON

The OTA booking agent reasons: the booking is CONFIRMED, journey date is 2026-06-15, current date is 2026-06-14. Pre-activity collection should begin. Selected action: atp:booking:pre_activity_open. Confidence: 0.91 (VERIFIED). No uncertainty flags.

A.3. PLAN

Transition Graph query confirms: CONFIRMED -> PRE_ACTIVITY via atp:booking:pre_activity_open is a valid path. authority_sufficient: true. hem_required: false.

Live Permission Map confirms: atp:booking:pre_activity_open is in permitted_actions for current state CONFIRMED.

A.4. ACT

Transition Request submitted:
- mandate_jwt: <root mandate, so_id bound, mandate_ceiling: 2>
- cedar_action: "atp:booking:pre_activity_open"
- idp: {
  action: "atp:booking:pre_activity_open",
  so_uuid: "019547ab-1234-7abc-8def-000000000099",
  transition_from: "CONFIRMED",
  transition_to: "PRE_ACTIVITY",
  goal_ref: "goal-session-azusa-journey-001",
  goal_step: 1,
  confidence: 0.91,
  reasoning_basis: [
  { ref_type: "so_graph_node", ref_id: "booking_reference",
  weight: "primary" },
  { ref_type: "zone_b_attachment", ref_id: "journey_date_doc",
  weight: "primary" }
  ],
  intent_summary: "Booking confirmed and journey date tomorrow.
  Opening pre-activity collection phase.",
  escalation_assessment: {
  agent_recommends_hem: false,
  hem_urgency: "ADVISORY"
  }
  }

GEC execution sequence:
Step 1: MJWT verified. so_id matches. Not revoked. Within exp.
Step 2: CAP -- no Tier 0 or Tier 1 prohibition applies.
Step 3: Cedar -- PERMIT. autonomy_level VERIFIED. No HEM required.
Step 4: State machine -- edge (CONFIRMED, atp:booking:pre_activity_open)
exists. Transition to PRE_ACTIVITY valid.
Step 5: Event Stream Write committed. StateTransitionEvent signed.

A.5. OBSERVE

GEC returns PERMIT response:
- new_state: "PRE_ACTIVITY"
- new_phase: "ACTIVE"
- event_stream_entry_id: "event-id-0042"

GAR records the IDP, the PERMIT result, and the delegation chain from the MJWT. AEP_SENSE_DELIVERED is already in the Event Stream.

The LOOP mechanism prepares the next Context Package with trigger: STATE_CHANGE, current_state: PRE_ACTIVITY. The next SENSE delivers it to the agent. aep_iteration increments to 2.

Author’s Address

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