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

T. Sato
MyAuberge K.K.

24 May 2026

The Intent Declaration Primitive (IDP) for Agentic AI Systems
draft-sato-soos-idp-03

Abstract

AI agents operating in automated workflows take actions without any normative mechanism for expressing why those actions are being taken. Access tokens declare what an agent is permitted to do; no existing standard declares what the agent believes it is doing, on what reasoning basis, and with what level of confidence, at the moment of action. This document defines the Intent Declaration Primitive (IDP): a structured per-transition declaration submitted by an AI agent to the Governing Enforcement Component (GEC) at each action step of an execution loop. The IDP is committed to a tamper-evident Event Log before the action executes, enabling post-hoc review of agent reasoning, richer authorization policy evaluation, and enriched denial responses that guide agent behaviour. The IDP also provides the technical basis for compliance with EU AI Act Article 12 logging requirements for high-risk AI systems. This document adds the IDP Commitment Verification mechanism, by which the GEC verifies at each state transition that the agent’s actual action matches its declared intent. This document also adds the RETRY_CONTINUATION reasoning basis type and the prior_denial_count Cedar context attribute, which together address computational inefficiency arising from uninformed agent retry loops. This document further defines, in Section 5.6, the schemas of the outcome Event Log entries (STATE_TRANSITIONED, CEDAR_DENY_RECORDED, and ACTION_RESULT_RECORDED) that record the result of each governed transition, completing the pre-action/post-action audit pair that gives the IDP its full auditability property. This document renames the Governing Kernel role to Governing Enforcement Component (GEC) to accommodate application-layer, isolated-process, and kernel-level implementations under a common conformance model. A Conformance Levels section (Section 9) defines three implementation profiles (Application, Isolated, and Kernel) with graduated non-suppressibility guarantees. A data_residency field is added to the IDP structure (Section 4.1) governing analytics tier eligibility. Section 3.5 states the Event Log as Operational Intelligence design intent.

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 23 November 2026.

Copyright Notice

Table of Contents

1. Introduction

The Internet Engineering Task Force and related standards bodies have made significant progress in specifying how AI agents authenticate (WIMSE [I-D.ietf-wimse-arch]) and how they obtain authorization tokens for API invocation (OAuth 2.1 [I-D.ietf-oauth-v2-1], AAuth [I-D.rosenberg-oauth-aauth]). These specifications answer the question: is this agent permitted to perform this action?

No existing specification answers the companion question: why does this agent believe it should perform this action at this moment?

This distinction matters for four reasons. First, post-hoc review of AI agent behaviour requires not only a record of what actions were taken but a record of the reasoning that led to them. Access token issuance records do not provide this. Second, authorization policy evaluation can be made significantly more precise when the agent's declared intent is available as a policy context attribute; a Cedar [Cedar] policy that evaluates both the action requested and the reasoning basis for that action is a more powerful governance tool than one that evaluates the action alone. Third, EU AI Act Article 12 requires that high-risk AI systems maintain logs sufficient for post-hoc human review of system decisions [EUAIA]; per-transition intent declaration is the technical specification that satisfies this requirement.

Fourth, uninformed retry loops represent a significant and measurable source of computational waste in deployed agentic systems. When an agent's action is denied and the agent receives no structured information about what alternative actions are available or what reasoning adjustments are warranted, the agent re-enters its reasoning loop without a revised basis for action. This produces semantically redundant computation: token expenditure and inference cost that generate no new information and advance no declared goal. The IDP addresses this through two mechanisms: the Enriched DENY Response (Section 6), which provides available_actions and

suggested_paths to guide the agent's next reasoning step; and the context_refs field (Section 4.1), which enables agents to explicitly reference prior failed attempts in subsequent declarations, making failure history a first-class input to both Cedar policy evaluation and human principal review. At the scale of large agentic deployments -- where LLM instances operate concurrently across millions of users -- elimination of uninformed retry loops through structured intent declaration represents substantial aggregate reduction in computational cost.

This document defines the Intent Declaration Primitive (IDP): a structured JSON object submitted by an AI agent to a governing runtime component at each action step of an agent execution loop, prior to the execution of the action. The IDP is recorded in a tamper-evident append-only Event Log before the action is executed. When authorization is denied, the GEC returns an Enriched DENY Response that includes the denial reason, available alternative actions, and the echo of the received IDP -- enabling the agent to reason about what it may do next.

The IDP is a GEC primitive, not an application-layer logging construct. It is submitted as part of the action invocation call, not after it. This is the property that gives the IDP its audit integrity: the reasoning declaration precedes the action record, making ex-post fabrication of intent structurally impossible within a conforming implementation.

This document also specifies IDP Commitment Verification (Section 5.5): a GEC mechanism that verifies, after every governed state transition, that the agent's actual action matches its declared IDP commitment. When a mismatch is detected -- the agent declared one intent and took a different action -- the GEC generates an IDP_COMMITMENT_GAP audit finding and automatically triggers Human Escalation Mechanism Class 2 (HEM_AGENT_ESCALATED).

- Section 5.6: Outcome Event Log Entries added (new section). Defines schemas for STATE_TRANSITIONED, CEDAR_DENY_RECORDED, and ACTION_RESULT_RECORDED Event Log entries. These entries record the result of each governed transition and complete the pre-action/post-action audit pair. A worked example (Section 5.6.4) shows the complete eight-entry Event Log for a denied-then-approved payment transition.

Changes from draft-sato-soos-idp-02:

- Throughout: "Governing Enforcement Component (GEC)" renamed to "Governing Enforcement Component (GEC)". The term "kernel" is retained only in the JSON field name kernel_signature (preserved for wire-format compatibility) and in section names that already existed in -02. All normative MUST/SHOULD/MAY requirements previously scoped to "the GEC" are now scoped to "the GEC", making them conformant for application-layer, isolated-process, and hardware-kernel implementations.

- Section 2: Governing Enforcement Component (GEC) definition replaced by Governing Enforcement Component (GEC) with three-level description. New terms added: GEC-signed, Transition Request, GEC Query Interface.

- Section 3.5: Event Log as Operational Intelligence added. States the analytics design intent and three-tier usage model (session-local, organization-internal, cross-domain). References forthcoming FAIP specification for Tier 3.

- Section 4.1: data_residency field added to IDP structure. Controls analytics tier eligibility per IDP record.

- Section 4.2: data_residency field definitions added.

- Section 9: Conformance Levels added (new section). Defines Level 1 (Application Profile), Level 2 (Isolated Profile), and Level 3 (Kernel Profile) with graduated non-suppressibility guarantees, signing key requirements, and regulatory applicability. Existing Sections 9-12 renumbered to 10-13.

- Section 5.1: GEC.transition() replaced by Transition Request abstraction.

- Section 5.2 heading: "Kernel Receipt and Validation" renamed to "GEC Receipt and Validation".

- Abstract line 19: "governing OS kernel" replaced with "Governing Enforcement Component (GEC)".

Changes from draft-sato-soos-idp-01:

- Section 1: Fourth motivation added -- computational efficiency through elimination of uninformed retry loops.

- Section 3.1: Problem (e) added -- computational inefficiency arising from uninformed retry loops and its aggregate cost at deployment scale.

- Section 4.2: context_refs field definition augmented to specify that prior denial references SHOULD be included on retry, and to define the prior_denial_count Cedar context attribute.

- Section 4.3: RETRY_CONTINUATION reasoning basis type added. Defines mandatory context_refs linkage to prior IDP_SUBMITTED events, required description of changed reasoning basis, and WARNING logging behaviour for retries without declared prior context.

- Section 5.4: prior_denial_count Cedar context attribute added to the set of IDP fields made available during policy evaluation.

- Section 11: RETRY_CONTINUATION added to IDP Reasoning Basis Types registry initial values.

Changes from draft-sato-soos-idp-00:

- Section 4.1: audit_accessible field added to IDP Core Record.

- Section 5.5: IDP Commitment Verification added (new section). Defines IDP Commitment Verification Record schema, IDP_COMMITMENT_VERIFIED and IDP_COMMITMENT_GAP Event Log entries, and IDP_COMMITMENT_GAP response protocol.

- Section 11: IANA registry names corrected (removed implementation-specific prefix).

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, including large language models, to reason about and take actions in pursuit of goals. An agent may be human-supervised, semi-autonomous, or fully autonomous.

Governing Enforcement Component (GEC):
   A runtime component that enforces authorization policy, records agent actions to a tamper-evident Event Log, and mediates agent access to governed objects. [I-D.sato-soos-hem] specifies the Human Escalation Mechanism that the GEC implements for human oversight of agent sessions. The GEC may be implemented as an application-layer library (Level 1), an isolated process or sidecar (Level 2), or an attested hardware execution environment (Level 3). See Section 9 for conformance level definitions.

GEC-signed:
   A record signed by the Governing Enforcement Component using the signing key appropriate to its conformance level. At Level 1, an application-managed key. At Level 2, a key held by the isolated GEC process, inaccessible to agent code. At Level 3, a key bound to a RATS-attested execution environment. The JSON field name kernel_signature is preserved across all levels for wire-format compatibility; the label field within the signature MUST indicate the conformance level (L1, L2, or L3).

Transition Request:
   The abstract operation by which an agent submits a cedar_action, mandate_jwt, and IDP to the GEC requesting a governed state transition. Protocol binding is implementation-defined; REST, gRPC, in-process function call, and message queue are all conforming transport mechanisms.

GEC Query Interface:
   The abstract interface by which authorized principals query Event Log entries. Protocol binding is implementation-defined.

Governed Object:
   A typed, stateful entity that the GEC governs. An agent operates on governed objects; the GEC enforces what transitions are permitted.

Mandate JWT:
   A JSON Web Token [RFC7519] that binds an agent’s authorization to a specific governed object instance. The mandate JWT is the authorization credential presented alongside an IDP.

Event Log:
   An append-only, causally ordered, tamper-evident log of all transitions executed against governed objects. The Event Log is the ground truth of system behaviour.

Cedar:
   A policy language and evaluation engine [Cedar] used by the GEC to evaluate authorization decisions. Cedar evaluates before any state transition executes.

HEM:
Human Escalation Mechanism. A GEC-enforced session state in which no transitions may execute until a designated human principal provides a decision. See [I-D.sato-soos-hem].

IDP:
   Intent Declaration Primitive. The structured per-transition reasoning declaration defined in this document.

IDP_THIN:

A reduced IDP profile for agents with limited reasoning capability. See Section 7.

IDP Commitment Verification Record:
A GEC-generated record produced after every governed state transition, recording whether the agent's actual action matched its IDP commitment. See Section 5.5.

IDP_COMMITMENT_GAP:
A condition in which the GEC determines that the agent's actual state transition does not match the agent's declared IDP commitment. A critical audit finding. See Section 5.5.

Prior Denial:
A GEC DENY response to a previous Transition Request within the same session. An IDP_SUBMITTED event with reasoning_basis.type RETRY_CONTINUATION MUST reference at least one prior denial via context_refs.

3. Problem Statement

3.1. The Intent Gap in Agentic Authorization

Current agentic authorization frameworks model agent behaviour as a set of permitted actions, expressed as OAuth scopes, Cedar action sets, or similar constructs. This model captures what an agent may do. It does not capture, at the moment of action:

* what goal the agent believes the action serves,
* what reasoning led the agent to select this action over alternatives,
* how confident the agent is that this action is appropriate, or
* whether the agent believes human judgment should be sought before proceeding.

The absence of per-transition intent records creates five concrete problems:

(a) Audit incompleteness. Post-hoc review of AI agent behaviour requires reconstruction of agent reasoning from action records alone. This is insufficient for high-stakes environments where understanding why an action was taken is as important as knowing what action was taken.

(b) Policy imprecision. Authorization policy evaluated solely against action type and agent identity cannot distinguish between an agent acting on a direct human instruction and an agent acting on an autonomous inference. These may warrant different authorization treatment.

(c) Opaque denials. When an agent's action is denied, the agent receives no structured information about what alternative actions are available or what reasoning adjustments might lead to a permitted action. This forces agents into uninformed retry loops.

(d) Regulatory non-compliance. EU AI Act Article 12 requires logs sufficient for human review of AI system decisions for high-risk systems. Access token issuance records do not satisfy this requirement.

(e) Computational inefficiency. An agent that receives a denial and no structured guidance re-enters its reasoning loop without a revised basis for action, producing redundant computation: token expenditure and inference cost that generate no new

information and advance no declared goal. An agent that retries a previously failed action without explicitly articulating what has changed in its reasoning basis is structurally likely to fail again for the same reason. This inefficiency compounds across the delegation chain: a sub-agent that silently retries propagates wasted computation upward to orchestrator agents that must await a result. Across large-scale agentic deployments operating concurrently for millions of users, uninformed retry loops represent a measurable aggregate waste of computational resources. The IDP Enriched DENY Response (Section 6) and the RETRY_CONTINUATION reasoning basis type with context_refs linkage (Section 4.3) address this directly.

3.2. Relationship to EU AI Act Article 12

EU AI Act Article 12 requires that high-risk AI systems maintain automatic logging sufficient to enable post-deployment monitoring, human review of decisions, and identification of risks. The logs must cover, at minimum, the operation of the system with reference to the relevant time period, the objects processed, and the data used.

The IDP, committed to the Event Log before each action, provides the per-decision record that Article 12 contemplates. An Event Log containing IDP entries for every agent transition produces a complete record of: what action was taken (the transition), by whom (the mandate JWT), against what (the governed object), and why (the IDP declared_goal and reasoning_basis). This document is informative with respect to regulatory compliance and does not constitute legal advice.

3.3. Relationship to Williams Intent Bound Authorization

Williams Intent Bound Authorization (IBA) [Williams-IBA] defines a human-signed authorization envelope issued before an agent session begins. The IBA envelope travels from human to agent; it constrains what the agent may do in the session.

The IDP is directionally distinct: it travels from agent to governing GEC, at each individual action step within the session. IBA expresses what the authorizing human intends the agent to do. IDP expresses what the agent declares it is doing and why.

The two primitives compose without conflict. An IBA-constrained agent session produces IDP entries at each transition; the IDP records the agent’s reasoning within the IBA-defined scope. Neither primitive replaces the other.

3.4. Relationship to Mission Bound Authorization

McGuinness Mission Bound Authorization [I-D.mcguinness-oauth-mission-bound-authorization] addresses how authorization tokens can be scoped to a declared mission, ensuring that token authority is contingent on mission validity. The IDP is directionally complementary: mission-bound authorization constrains what the agent is permitted to do within a mission; the IDP records what the agent declares it is doing and why at each individual transition within that mission. The two primitives compose without conflict. A system implementing mission-bound authorization produces IDP entries at each transition; the IDP’s mission_ref field (Section 4.1) links per-transition declarations to the governing MissionDeclaration evaluated by the mission-bound authorization framework. Neither primitive replaces the other.

3.5. Event Log as Operational Intelligence

The IDP Event Log is designed as a governance record: tamper-evident, GEC-signed, and non-suppressible. A consequence of the pre-action commitment plus post-action outcome pair (IDP_SUBMITTED and ACTION_RESULT_RECORDED) is that the Event Log is simultaneously the most structured dataset that will exist about how agentic AI systems behave in production. This section states the operational intelligence design intent that this structure enables. Protocol specifications for Tier 2 and Tier 3 analytics are defined in [I-D.sato-soos-gar] and the forthcoming Federated Agent Intelligence Protocol (FAIP) specification respectively.

Three tiers of Event Log usage are defined:

Tier 1 -- Session-Local. Within a single session, the GEC consults its own Event Log to derive the prior_denial_count attribute (Section 5.4), to detect GOAL_INCOMPLETE conditions (Section 5.6), and to detect CONFIDENCE_MISCALIBRATION patterns. No data leaves the session trust boundary. All Tier 1 analytics are specified in this document and require no additional protocol.

Tier 2 -- Organization-Internal. Across sessions within the same organizational trust domain, anonymized and aggregated Event Log patterns enable action success rate analysis, retry pattern libraries (which RETRY_CONTINUATION reasoning adjustments historically lead to PERMITTED outcomes after specific deny_codes), and goal completion profiling. Tier 2 analytics are subject to k-anonymity enforcement (minimum session count per query result) and data_residency constraints (Section 4.2). The Analytics Principal role and GEC query interface for Tier 2 are specified in [I-D.sato-soos-gar].

Tier 3 -- Cross-Domain Federated. Across organizational boundaries, privacy-preserving aggregation enables cross-industry pattern transfer: planning priors derived from billions of governed agent sessions spanning multiple SO Types and industries. Tier 3 analytics require that participating organizations file a signed Participation Declaration specifying contribution scope, privacy model (federated learning or differential privacy), and data residency constraints. The Tier 3 protocol, trust registry model, and Participation Declaration schema are specified in the Federated Agent Intelligence Protocol (FAIP), a forthcoming companion specification. Tier 3 eligibility for individual IDP records is controlled by the tier3_eligible field in data_residency (Section 4.2).

The computational efficiency motivation for the IDP (Section 1, fourth motivation; Section 3.1(e)) extends across tiers. Tier 1 analytics eliminate uninformed retry loops within a session. Tier 2 analytics eliminate category-level retry failure patterns across an organization’s sessions: an agent encountering a POLICY_DENY on Action::"ProcessPayment" can consult the organization’s Tier 2 RETRY_PATTERN_LIBRARY to learn which reasoning adjustments historically succeeded. Tier 3 analytics extend this across industries, enabling cross-domain transfer of validated planning trajectories.

The data_residency field (Section 4.2) is the per-record control mechanism by which operators govern which tiers a given IDP record contributes to. Organizations operating under GDPR or other data residency requirements MUST set jurisdiction and tier eligibility fields appropriately for all sessions within the regulated scope.

3.5. Event Log as Operational Intelligence

The IDP Event Log is specified as a governance record: GEC-signed, append-only, and tamper-evident. A consequence of the pre-action commitment plus post-action outcome pair defined in this document is that the Event Log is simultaneously the most richly structured behavioral dataset that will exist for deployed agentic AI systems. This section states the operational intelligence design intent that this structure enables. Analytics protocol specifications are defined in [I-D.sato-soos-gar] (Tier 2) and the forthcoming Federated Agent Intelligence Protocol (FAIP) specification (Tier 3).

Three tiers of Event Log usage are defined:

Tier 1 -- Session-Local.
The GEC consults its own Event Log within a single session to derive prior_denial_count (Section 5.4), detect GOAL_INCOMPLETE conditions, and detect CONFIDENCE_MISCALIBRATION patterns. No data leaves the session trust boundary. All Tier 1 analytics are specified in this document.

Tier 2 -- Organization-Internal.
Anonymized and aggregated Event Log patterns across sessions within the same organizational trust domain enable: action success rate analysis by reasoning_basis type; RETRY_PATTERN libraries (which RETRY_CONTINUATION reasoning adjustments historically led to PERMITTED outcomes after specific deny_codes); and goal completion profiling. Tier 2 analytics are subject to k-anonymity enforcement and data_residency constraints (Section 4.2). The Analytics Principal role and GEC query interface for Tier 2 are specified in [I-D.sato-soos-gar].

Tier 3 -- Cross-Domain Federated.
Privacy-preserving aggregation across organizational boundaries enables cross-industry pattern transfer and planning priors derived from large populations of governed agent sessions. Tier 3 requires that organizations file a Participation Declaration specifying contribution scope and privacy model. The Tier 3 protocol, trust registry, and Participation Declaration schema are specified in the forthcoming FAIP specification. Tier 3 eligibility per IDP record is controlled by the tier3_eligible field in data_residency (Section 4.2).

The computational efficiency motivation (Section 1, fourth; Section 3.1(e)) extends across tiers: Tier 1 eliminates uninformed retry loops within a session; Tier 2 eliminates category-level retry failure patterns across an organization’s sessions; Tier 3 enables cross-domain transfer of validated planning trajectories.

4. The Intent Declaration Primitive

4.1. IDP Structure

The IDP is a JSON object with the following structure:

```
{
  "idp_id":           string,   ; REQUIRED. UUID v4.
  "session_id":       string,   ; REQUIRED. Agent session identifier.
  "so_id":            string,   ; REQUIRED. Governed object UUID.
  "mandate_id":       string,   ; REQUIRED. Mandate JWT jti claim.
  "mission_ref":      string,   ; OPTIONAL. MissionDeclaration UUID.
                                ; Links this per-transition declaration
                                ; to the governing MissionDeclaration.
                                ; MUST match the mission_ref in the
                                ; mandate JWT if that field is present.
  "step_sequence":    integer,  ; REQUIRED. ACT step number, 1-based.
  "requested_action": string,   ; REQUIRED. Cedar action string.
```

```
  "declared_goal": {            ; REQUIRED.
    "goal_id":        string,   ; REQUIRED. UUID v4.
    "description":    string    ; REQUIRED. Human-readable. Max 500 chars.
  },
  "reasoning_basis": {          ; REQUIRED.
    "type":           string,   ; REQUIRED. See Section 4.3.
    "description":    string    ; REQUIRED. Human-readable. Max 1000 chars.
  },
  "confidence_level": number,   ; REQUIRED. Float, 0.0 to 1.0 inclusive.
  "hem_urgency":      string,   ; REQUIRED. See Section 4.4.
  "context_refs":    [string],  ; OPTIONAL. UUIDs of relevant prior events.
                                ; See Section 4.2 for retry requirements.
  "audit_accessible": boolean,  ; OPTIONAL. Default: true. When false,
                                ; IDP is not accessible to Audit
; Principals via GEC query interfaces.
  "metadata":         object,   ; OPTIONAL. Implementation-defined.
  "timestamp":        string,   ; REQUIRED. ISO 8601 UTC.
  "data_residency": {           ; OPTIONAL. Analytics eligibility.
    "jurisdiction":   string,   ; REQUIRED if field present.
                                ; ISO 3166-1 alpha-2, "EU", "EEA",
                                ; or "GLOBAL".
    "tier2_eligible": boolean,  ; REQUIRED if field present.
                                ; May contribute to organization-
                                ; internal analytics aggregation.
    "tier3_eligible": boolean,  ; REQUIRED if field present.
                                ; May contribute to cross-domain
                                ; federated analytics aggregation.
    "retention_days": integer,  ; OPTIONAL. Retention override.
    "anonymization_delay_days": integer ; OPTIONAL. Days after
                                ; session close before eligible
                                ; for analytics. Default: 0.
  }
}
```

4.2. Field Definitions

idp_id:
: A UUID v4 [RFC4122] assigned by the agent. Uniquely identifies this IDP within the Event Log. MUST be unique across all IDPs for the lifetime of a governed object instance.

session_id:
: The identifier of the agent session within which this action is taken. MUST match the session_id established at session creation.

so_id:
: The UUID v4 of the governed object being operated on. MUST match the UUID bound to the mandate_id.

mandate_id:
: The jti claim of the mandate JWT presented with this IDP. Links the intent declaration to the authorization credential under which the action is requested.

mission_ref:
: OPTIONAL. The UUID of the MissionDeclaration governing the session in which this IDP is submitted. When present, this field links the per-transition intent declaration to the overarching mission under which the agent is operating. If the mandate JWT carries a mission_ref claim, the IDP mission_ref MUST match that claim. Absent if the session does not operate under a declared MissionDeclaration.

step_sequence:
: A positive integer indicating the ordinal position of this ACT step within the current session. Starts at 1. MUST be

monotonically increasing within a session. The GEC SHOULD reject an IDP with a step_sequence equal to or less than the last committed step_sequence for this session.

requested_action:
The Cedar action string identifying the action the agent requests. MUST be the same Cedar action string submitted to the GEC for policy evaluation.

declared_goal:
A structured declaration of the goal the agent believes this action serves.

goal_id:
A UUID v4 assigned by the agent. MAY be shared across multiple IDPs within a session if those IDPs are understood to serve the same goal.

description:
A human-readable description of the goal. MUST NOT contain personally identifiable information. Maximum 500 characters.

reasoning_basis:
A structured declaration of the reasoning that led the agent to request this action.

type:
One of the reasoning basis types defined in Section 4.3.

description:
A human-readable description of the specific reasoning. MUST NOT contain personally identifiable information. Maximum 1000 characters. When reasoning_basis.type is RETRY_CONTINUATION, this field MUST identify: (a) what action was previously denied or failed, (b) what has changed in the agent's understanding or approach since the prior attempt, and (c) why the agent believes the revised approach will succeed.

confidence_level:
A floating-point number in [0.0, 1.0] expressing the agent's self-assessed confidence that this action is appropriate. 1.0 indicates full confidence; 0.0 indicates no confidence. This value is self-reported by the agent and is not independently verified by the GEC. Implementers SHOULD treat this value as a declaration, not a guarantee.

hem_urgency:
A declaration of the agent's assessment of whether human judgment should be sought. See Section 4.4.

context_refs:
An OPTIONAL array of Event Log entry UUIDs that the agent considers relevant to its reasoning. Allows tracing of how prior events informed the current action.

When the agent is retrying an action following a prior denial within the current session, context_refs SHOULD include the idp_ids of IDP_SUBMITTED events from prior attempts at the same requested_action. This enables the GEC to compute the prior_denial_count Cedar context attribute (Section 5.4) and enables audit tools to reconstruct the full failure-and-retry reasoning chain for post-hoc review.

The GEC SHOULD make the count of prior denials for the same requested_action available as a Cedar context attribute (idp.prior_denial_count), enabling Cedar policies that apply

progressive restriction or mandatory HEM escalation to repeatedly denied actions. The prior_denial_count value is derived by the GEC from the Event Log; it is not supplied by the agent and MUST NOT be accepted as an agent-supplied field.

The presence of prior denial references in context_refs, combined with a RETRY_CONTINUATION reasoning basis type (Section 4.3), enables Cedar policies to distinguish between a first-attempt action and a revised retry, and to apply different authorization treatment accordingly.

audit_accessible:
OPTIONAL. Boolean. Default: true. When true, this IDP record is accessible to Audit Principals via GEC query interfaces (GEC query interface (GEC.query_event_log())) and is included in Session Audit Records and Audit Packages. When false, the IDP is accessible only to the GEC's internal audit functions and to Verified External Auditors; it is not returned in response to Audit Principal queries. Operators who require restricted IDP access for confidentiality reasons MAY set this field to false; such restrictions MUST be documented in the Session Audit Record.

metadata:
An OPTIONAL implementation-defined JSON object. The GEC MUST record this field in the Event Log without interpretation. Domain-specific IDP extensions SHOULD use this field.

timestamp:
The time at which the agent generated the IDP, in ISO 8601 format with UTC timezone. The GEC MUST record both this timestamp and the GEC receipt timestamp in the Event Log entry.

data_residency:
OPTIONAL. A structured declaration governing the eligibility of this IDP record for analytics aggregation beyond the session-local tier. When absent, the SO Type's default data residency policy applies.

jurisdiction:
REQUIRED when data_residency is present. The geographic or regulatory jurisdiction of the data subject or governed object. Expressed as an ISO 3166-1 alpha-2 country code (e.g., "JP", "DE", "US"), or one of the multi-country values "EU" (European Union), "EEA" (European Economic Area), or "GLOBAL" (no geographic restriction). The GEC MUST NOT route analytics contributions from this IDP record to aggregation services outside the declared jurisdiction without explicit operator policy authorization.

tier2_eligible:
REQUIRED when data_residency is present. Boolean. When true, this IDP record MAY contribute to organization-internal analytics aggregation (Tier 2, as defined in the forthcoming FAIP specification). When false, this record MUST NOT appear in any Tier 2 aggregate query result, even if other records from the same session are tier2_eligible. Default when data_residency is absent: operator-configurable, RECOMMENDED default true.

tier3_eligible:
REQUIRED when data_residency is present. Boolean. When true, this IDP record MAY contribute to cross-domain federated analytics aggregation (Tier 3) subject to the Participation Declaration filed by the operator under the FAIP specification. When false, this record MUST NOT contribute to any Tier 3

aggregate. Default when data_residency is absent: operator-configurable, RECOMMENDED default false (opt-in for Tier 3).

retention_days:
OPTIONAL. Integer. Overrides the default Event Log retention policy for this record. MUST NOT exceed the maximum retention period specified by the governing SO Type. MUST NOT exceed any applicable regulatory maximum retention period.

anonymization_delay_days:
OPTIONAL. Integer. The number of days after session close before this record becomes eligible for analytics aggregation. Allows a cooling-off period during which the record is accessible for operational review but not yet contributed to analytics aggregates. Default: 0 (eligible immediately after session close).

4.3. Reasoning Basis Types

The reasoning_basis.type field MUST be one of the following values:

RULE_BASED:
The agent is executing a predetermined rule or procedure. The action was selected by applying a defined decision rule to the current state, not by open-ended inference. Example: "Booking confirmation is required when payment is received; payment has been received."

INFERENCE:
The agent inferred that this action is appropriate based on contextual reasoning. The action was not explicitly prescribed by a rule; the agent applied judgment to conclude it is correct. Example: "Based on the stated guest preferences and current availability, this room assignment appears optimal."

INSTRUCTION:
The agent is executing an explicit instruction from a human principal or from a higher-authority agent in the delegation chain. Where the mandate JWT carries a delegation_chain claim [I-D.mcguinness-oauth-actor-profile], the source of the instruction is traceable through the full actor chain. The reasoning_basis.description MUST identify the source of the instruction by mandate_id or session_id.

UNCERTAINTY_REDUCTION:
The agent is taking this action to reduce uncertainty, not to directly advance the declared goal. Used when an agent performs a read or query action to gather information before determining the appropriate forward action.

MISSION_STAGE:
The agent is executing an action whose primary purpose is to advance the governing MissionDeclaration from one stage to the next. This type is used when the mission stage transition, not the domain goal itself, is the proximate reason for the action. The mission_ref field MUST be present when this type is used.

RETRY_CONTINUATION:
The agent is retrying an action following a prior denial or failure within the current session. The idp_id of the prior attempt's IDP_SUBMITTED Event Log entry MUST be included in context_refs. The reasoning_basis.description MUST identify: (a) what action was previously denied or failed, (b) what has changed in the agent's understanding or approach since the prior attempt, and (c) why the agent believes the revised approach will succeed.

A RETRY_CONTINUATION IDP with no context_refs referencing a prior IDP_SUBMITTED event MUST be logged as a WARNING in the Event Log without rejection; the GEC SHOULD make the retry_without_prior_ref flag available as a Cedar context attribute. Cedar policies MAY use this flag to apply additional scrutiny or mandatory HEM escalation to agents that repeatedly retry without declared learning.

The cognitive discipline of explicitly articulating what has changed since a prior failure -- required by this type -- is the primary mechanism by which the IDP reduces uninformed retry loops (Section 3.1(e)). An agent that cannot articulate what has changed since its prior failure has not revised its reasoning basis and is structurally likely to produce the same outcome. Cedar policies SHOULD enforce a maximum retry count for RETRY_CONTINUATION IDPs against the same requested_action before mandatory HEM escalation, using the GEC-derived idp.prior_denial_count attribute.

Implementations MAY define additional type values. Such values MUST be registered in the IDP Reasoning Basis Types registry (see Section 11) or prefixed with a URI identifying the defining organization. Unrecognized type values MUST be recorded in the Event Log and MUST NOT cause the GEC to reject the IDP.

4.4. HEM Urgency Values

The hem_urgency field is a declaration by the agent of its assessment of whether human judgment should be sought before this action executes. The GEC uses this value as one input to HEM trigger determination.

NONE:
The agent assesses that no human judgment is needed for this action. The GEC proceeds with normal Cedar evaluation.

RECOMMENDED:
The agent assesses that human judgment would be beneficial but does not consider it required. The GEC SHOULD record this value but is not required to initiate HEM. GEC implementations MAY use this value to trigger HEM based on local policy.

REQUIRED:
The agent declares that it requires human judgment before this action executes. The GEC MUST initiate HEM_PENDING for this session upon receipt of an IDP with hem_urgency: REQUIRED, regardless of Cedar policy evaluation outcome. This corresponds to the HEM_AGENT_ESCALATED trigger class defined in [I-D.sato-soos-hem].

Note: A hem_urgency value of REQUIRED from the agent does not override GEC Cedar evaluation. Cedar evaluation still executes. However, even a Cedar PERMIT result MUST NOT cause the action to execute; the session enters HEM_PENDING, and the action awaits human decision.

4.5. IDP Profiles

This document defines two IDP profiles:

IDP_STANDARD:
All REQUIRED fields present. confidence_level is agent-assessed. reasoning_basis.type is one of the defined values. This is the normative profile for agents with sufficient reasoning capability to self-assess.

IDP_THIN:
A reduced profile for agents with limited reasoning capability. See Section 7.

The GEC MUST accept both profiles. The GEC MUST record the profile type in the IDP Event Log entry.

5. Submission Protocol

5.1. When the IDP Is Submitted

The IDP MUST be submitted by the agent as part of the same GEC call that requests the state transition. The IDP MUST NOT be submitted after the action has executed.

In a conforming agentic execution loop, the sequence is:

1. (SENSE) GEC delivers context to agent.
2. (REASON) Agent reasons about next action.
3. (PLAN) Agent consults Transition Graph API for available actions.
4. (ACT) Agent submits GEC.transition(mandate_jwt, cedar_action, idp) -- the IDP is part of this call.
5. (OBSERVE) Agent receives result: success, enriched DENY, or SESSION_HEM_PENDING.
6. (LOOP) Agent processes result before next ACT.

The IDP MUST be present in the Transition Request. A Transition Request without an IDP MUST be rejected by a conforming GEC implementation.

5.2. GEC Receipt and Validation

Upon receiving a Transition Request, the GEC MUST:

(a) Validate that an IDP is present. If absent, return REJECT with error code IDP_MISSING.

(b) Validate that the IDP fields conform to this specification. If malformed, return REJECT with error code IDP_MALFORMED.

(c) Validate that the idp_id is unique for this governed object. If a duplicate idp_id is detected, return REJECT with error code IDP_DUPLICATE.

(d) Validate that the IDP so_id matches the governed object UUID bound to the mandate_jwt. If mismatched, return REJECT with error code IDP_SO_MISMATCH.

(e) Validate that the IDP mandate_id matches the jti claim of the presented mandate_jwt. If mismatched, return REJECT with error code IDP_MANDATE_MISMATCH.

(f) Validate that step_sequence is strictly greater than the last committed step_sequence for this session.

(g) If mission_ref is present, validate that it matches the mission_ref claim in the mandate_jwt (if that claim is present). If mismatched, return Enriched DENY with deny_code IDP_MISSION_REF_MISMATCH and a structured mismatch_detail block containing the expected and submitted mission_ref values. The GEC MUST log IDP_MISSION_REF_MISMATCH_REJECTED in the Event Log. If mission_ref mismatches exceed a configurable threshold within a session, the GEC SHOULD automatically trigger HEM_AGENT_ESCALATED (Class 2).

(h) If reasoning_basis.type is RETRY_CONTINUATION and context_refs contains no reference to a prior IDP_SUBMITTED event for the same requested_action in this session, the GEC MUST log a WARNING entry (RETRY_WITHOUT_PRIOR_REF) in the Event Log and MUST make the retry_without_prior_ref flag available as a Cedar context attribute for the current evaluation. The GEC MUST NOT reject the IDP solely on this basis.

GEC validation of the IDP precedes Cedar evaluation. A valid IDP is required for Cedar evaluation to proceed.

5.3. Event Log Commitment

A conforming GEC implementation MUST commit an IDP_SUBMITTED event to the Event Log upon successful validation of the IDP, BEFORE Cedar evaluation executes.

The IDP_SUBMITTED event MUST contain:

* The full IDP object as submitted.
* The GEC receipt timestamp (distinct from the agent timestamp).
* The mandate_id.
* The session_id.
* The audit_accessible value (defaults to true if absent).
* The GEC-derived prior_denial_count for the requested_action within the current session.
* The GEC signature over the event.

The IDP_SUBMITTED event is committed regardless of the subsequent Cedar evaluation outcome. If Cedar evaluation results in DENY, the IDP_SUBMITTED event remains in the Event Log as a permanent record of the agent’s declared intent.

The Event Log ordering guarantee is:

```
IDP_SUBMITTED < Cedar evaluation < STATE_TRANSITIONED
(or CEDAR_DENY_RECORDED) < ACTION_RESULT_RECORDED
< IDP_COMMITMENT_VERIFIED (or IDP_COMMITMENT_GAP)
(or CEDAR_DENY_RECORDED)
```

The complete Event Log entry schemas are defined in Section 5.6.

No STATE_TRANSITIONED event may appear in the Event Log without a preceding IDP_SUBMITTED event for the same step_sequence.

Following every STATE_TRANSITIONED event, the GEC MUST generate an IDP_COMMITMENT_VERIFIED or IDP_COMMITMENT_GAP event as specified in Section 5.5.

5.4. IDP in Cedar Policy Evaluation

The GEC MUST make the following IDP fields available as Cedar context attributes during policy evaluation:

```
* idp.reasoning_basis.type  -- string
* idp.confidence_level      -- decimal
* idp.hem_urgency           -- string
* idp.goal_id               -- string
* idp.mission_ref           -- string (null if absent)
* idp.prior_denial_count    -- integer (GEC-derived; count of
                               prior DENY responses for the same
                               requested_action in this session)
* idp.retry_without_prior_ref -- boolean (true when
                                 reasoning_basis.type is
```

```
                                    RETRY_CONTINUATION and context_refs
                                    contains no prior IDP reference;
                                    false otherwise)
```

Cedar policies MAY use these attributes in condition expressions. Example policies:

```
// Require INSTRUCTION basis for high-value transitions
permit(principal, action == Action::"ProcessPayment", resource)
when {
  context.idp.reasoning_basis.type == "INSTRUCTION"
};

// Require confidence >= 0.8 for autonomous state closure
permit(principal, action == Action::"CloseBooking", resource)
when {
  context.idp.confidence_level >= decimal("0.8")
};

// Require mission_ref for MISSION_STAGE transitions
permit(principal, action == Action::"AdvanceMissionStage", resource)
when {
  context.idp.reasoning_basis.type == "MISSION_STAGE" &&
  context.idp.mission_ref != null
};

// Escalate to HEM after three denials for the same action
forbid(principal, action, resource)
when {
  context.idp.prior_denial_count >= 3
};

// Escalate to HEM when agent retries without declared learning
forbid(principal, action, resource)
when {
  context.idp.retry_without_prior_ref == true &&
  context.idp.prior_denial_count >= 1
};
```

Cedar policies that reference IDP attributes but receive an IDP_THIN submission (Section 7) where the relevant attribute is absent MUST treat the absent attribute as if it had a value that causes the condition to evaluate to false, resulting in DENY. This ensures that thin-profile agents cannot satisfy policies written for standard-profile agents by omission.

5.5. IDP Commitment Verification

After every successful governed state transition (STATE_TRANSITIONED Event Log entry), the GEC MUST generate an IDP Commitment Verification Record and commit it to the Event Log. The purpose of Commitment Verification is to detect IDP_COMMITMENT_GAP conditions: cases where the agent's actual state transition does not match the action declared in its preceding IDP.

This mechanism enforces the non-suppressibility property: the GEC cannot conceal the fact that an agent's actual behaviour diverged from its declared intent.

5.5.1. IDP Commitment Verification Record Schema

The IDP Commitment Verification Record MUST contain the following fields:

idp_id:
: The idp_id of the IDP submitted for this transition. Links the

verification record to the specific intent declaration.

state_transition_id:
The Event Log identifier of the STATE_TRANSITIONED entry being verified against.

verified_at:
ISO 8601 UTC timestamp of verification. Generated by the GEC.

match_result:
One of:

MATCHED: The Cedar action string in the IDP's requested_action field matches the action that was executed in the state transition. The agent acted in accordance with its declaration.

IDP_COMMITMENT_GAP: The Cedar action string in the IDP's requested_action field does not match the action that was executed. This is a critical audit finding.

kernel_signature:
Ed25519 signature over the canonical serialization of all fields except kernel_signature.

5.5.2. Event Log Entries

IDP_COMMITMENT_VERIFIED:
Committed when match_result is MATCHED. Contains the full IDP Commitment Verification Record.

IDP_COMMITMENT_GAP:
Committed when match_result is IDP_COMMITMENT_GAP. Contains the full IDP Commitment Verification Record. This entry MUST be treated as a critical audit finding. The GEC MUST immediately:

(a) Generate a CRITICAL Audit Alert (alert_trigger: IDP_COMMITMENT_GAP).

(b) Fire HEM_AGENT_ESCALATED (Class 2 per [I-D.sato-soos-hem] Section 5.2) for the active session. The HEM Escalation Request MUST include the IDP Commitment Verification Record in the idp_summary, with match_result: IDP_COMMITMENT_GAP clearly surfaced to the resolving principal.

The GEC MUST NOT allow the agent session to continue after an IDP_COMMITMENT_GAP without HEM resolution. The session enters HEM_PENDING immediately.

5.5.3. Commitment Verification Scope

The GEC compares the requested_action field of the IDP against the Cedar action string of the executed transition. The comparison is string-exact. A mismatch on any character MUST be recorded as IDP_COMMITMENT_GAP.

IDP_COMMITMENT_GAP does not distinguish between:

- An agent that declared one action and executed a different one due to an implementation error.
- An agent that was coerced or compromised after IDP submission.
- An agent that exploited a race condition between IDP submission and transition execution.

All three produce the same audit finding. The distinction is a

matter for human principal investigation following HEM resolution.

5.5.4. Relationship to GAR

The IDP Commitment Verification Record and its Event Log entries are specified in this document. Their inclusion in Session Audit Records and Audit Packages is governed by [I-D.sato-soos-gar]. The IDP_COMMITMENT_GAP Audit Alert trigger is registered in the GAR Audit Alert Triggers registry.

5.6. Outcome Event Log Entries

The IDP_SUBMITTED entry (Section 5.3) records the agent's declared intent before the action executes. This section defines the complementary outcome entries that the GEC MUST commit after the action executes or is denied, completing the pre-action/post-action audit pair.

Note: The authoritative definition of these entries will be provided by the Agent Execution Protocol (AEP) specification [I-D.sato-soos-aep] (forthcoming). The schemas defined here are a self-contained subset profile sufficient for conforming implementation of IDP-based audit without requiring AEP. Where AEP and this document define overlapping fields, AEP is normative.

The complete Event Log ordering for a single governed transition is:

```
IDP_SUBMITTED
  < Cedar evaluation
  < STATE_TRANSITIONED (PERMIT) or CEDAR_DENY_RECORDED (DENY)
  < ACTION_RESULT_RECORDED
  < IDP_COMMITMENT_VERIFIED or IDP_COMMITMENT_GAP (PERMIT only)
```

5.6.1. STATE_TRANSITIONED

Committed by the GEC after a governed state transition executes successfully following a Cedar PERMIT. Records the result of the transition.

```
{
  "event_type":         "STATE_TRANSITIONED",  ; REQUIRED.
  "event_id":           string,   ; REQUIRED. UUID v4. GEC-assigned.
  "session_id":         string,   ; REQUIRED.
  "so_id":              string,   ; REQUIRED. Governed object UUID.
  "mandate_id":         string,   ; REQUIRED. Mandate JWT jti claim.
  "step_sequence":      integer,  ; REQUIRED. Matches IDP step_sequence.
  "idp_id":             string,   ; REQUIRED. idp_id of preceding IDP.
  "cedar_action":       string,   ; REQUIRED. Cedar action string
                                  ; executed.
  "from_state":         string,   ; REQUIRED. SO state before transition.
  "to_state":           string,   ; REQUIRED. SO state after transition.
  "transition_inputs":  object,   ; OPTIONAL. Implementation-defined.
                                  ; MUST NOT contain PII.
  "transition_outputs": object,   ; OPTIONAL. Implementation-defined.
                                  ; MUST NOT contain PII.
  "executed_at":        string,   ; REQUIRED. ISO 8601 UTC.
  "kernel_signature":   string    ; REQUIRED. Ed25519 signature.
}
```

5.6.2. CEDAR_DENY_RECORDED

Committed by the GEC when Cedar evaluation returns DENY. This

entry is the permanent, GEC-signed Event Log record of the denial. It is produced independently of the Enriched DENY Response (Section 6) returned to the agent; both are produced for every denial.

```
{
  "event_type":          "CEDAR_DENY_RECORDED",  ; REQUIRED.
  "event_id":            string,   ; REQUIRED. UUID v4. GEC-assigned.
  "session_id":          string,   ; REQUIRED.
  "so_id":               string,   ; REQUIRED.
  "mandate_id":          string,   ; REQUIRED.
  "step_sequence":       integer,  ; REQUIRED. Matches IDP step_sequence.
  "idp_id":              string,   ; REQUIRED. idp_id of preceding IDP.
  "cedar_action":        string,   ; REQUIRED. Cedar action string denied.
  "deny_code":           string,   ; REQUIRED. See Section 6.2.
  "deny_reason":         string,   ; REQUIRED. Human-readable. MUST NOT
                                   ; expose exploitable policy structure.
  "so_state_at_deny":    string,   ; REQUIRED. SO state at denial.
  "prior_denial_count": integer,   ; REQUIRED. GEC-derived count of
                                   ; prior denials for this cedar_action
                                   ; in this session.
  "denied_at":           string,   ; REQUIRED. ISO 8601 UTC.
  "kernel_signature":    string    ; REQUIRED. Ed25519 signature.
}
```

Each denial produces a separate CEDAR_DENY_RECORDED entry. The sequence of CEDAR_DENY_RECORDED entries for the same cedar_action within a session constitutes the retry history surfaced to human principals when RETRY_LIMIT_EXCEEDED triggers HEM escalation (Section 6.3).

5.6.3. ACTION_RESULT_RECORDED

Committed by the GEC after every governed transition, whether PERMIT or DENY, as a unified summary record linking the IDP to its outcome. This entry is the single join point between the pre-action IDP record and the post-action outcome, enabling audit tools to reconstruct the complete intent-to-outcome chain for any step_sequence without traversing multiple event types.

```
{
  "event_type":           "ACTION_RESULT_RECORDED",  ; REQUIRED.
  "event_id":             string,   ; REQUIRED. UUID v4. GEC-assigned
  "session_id":           string,   ; REQUIRED.
  "so_id":                string,   ; REQUIRED.
  "step_sequence":        integer,  ; REQUIRED.
  "idp_id":               string,   ; REQUIRED. Links to IDP_SUBMITTED.
  "outcome":              string,   ; REQUIRED. One of: "PERMITTED",
                                    ; "DENIED", "HEM_PENDING".
  "outcome_event_id":     string,   ; REQUIRED. event_id of:
                                    ; STATE_TRANSITIONED (if PERMITTED),
                                    ; CEDAR_DENY_RECORDED (if DENIED), or
                                    ; HEM session event (if HEM_PENDING).
  "reasoning_basis_type": string,   ; REQUIRED. Echo of IDP
                                    ; reasoning_basis.type.
  "confidence_level":     number,   ; REQUIRED. Echo of IDP
                                    ; confidence_level.
  "hem_urgency":          string,   ; REQUIRED. Echo of IDP hem_urgency.
  "recorded_at":          string,   ; REQUIRED. ISO 8601 UTC.
  "kernel_signature":     string    ; REQUIRED. Ed25519 signature.
}
```

The outcome field MUST be one of:

PERMITTED:
   Cedar returned PERMIT and STATE_TRANSITIONED was committed. outcome_event_id references the STATE_TRANSITIONED entry.

DENIED:
Cedar returned DENY. outcome_event_id references the CEDAR_DENY_RECORDED entry.

HEM_PENDING:
The transition triggered HEM escalation (via hem_urgency: REQUIRED in the IDP, via RETRY_LIMIT_EXCEEDED, or via IDP_COMMITMENT_GAP). The action did not execute. outcome_event_id references the HEM session event.

5.6.4. Complete Audit Trail Example

The following example shows the complete Event Log for an agent that attempts Action::"ProcessPayment", receives POLICY_DENY, submits a RETRY_CONTINUATION IDP with HEM escalation, and executes after human approval.

```
Step 1 -- First attempt:
   IDP_SUBMITTED          (step_sequence: 1,
                           reasoning_basis.type: INFERENCE,
                           hem_urgency: NONE)
   CEDAR_DENY_RECORDED    (deny_code: POLICY_DENY,
                           prior_denial_count: 0)
   ACTION_RESULT_RECORDED (outcome: "DENIED", step_sequence: 1,
                           outcome_event_id: [CEDAR_DENY event_id])

Step 2 -- Retry with HEM escalation:
   IDP_SUBMITTED          (step_sequence: 2,
                           reasoning_basis.type: RETRY_CONTINUATION,
                           context_refs: [idp_id from step 1],
                           hem_urgency: REQUIRED,
                           description: "Prior denial: POLICY_DENY.
                           Payment amount exceeds autonomous threshold.
                           Requesting human principal approval.")
   [CEDAR_DENY_RECORDED not committed -- HEM_PENDING triggered
    before Cedar evaluation by hem_urgency: REQUIRED]
   [HEM escalation and human principal DRR]
   ACTION_RESULT_RECORDED (outcome: "HEM_PENDING", step_sequence: 2,
                           outcome_event_id: [HEM session event_id])

Step 3 -- Post-HEM execution:
   IDP_SUBMITTED          (step_sequence: 3,
                           reasoning_basis.type: INSTRUCTION,
                           description: "Human principal approved via
                           HEM DRR [drr_id]. Executing under
                           instruction.")
   STATE_TRANSITIONED     (from_state: PAYMENT_PENDING,
                           to_state: PAYMENT_PROCESSED)
   ACTION_RESULT_RECORDED (outcome: "PERMITTED", step_sequence: 3,
                           outcome_event_id: [STATE_TRANS event_id])
   IDP_COMMITMENT_VERIFIED (match_result: MATCHED)
```

This sequence produces nine Event Log entries across three ACT steps, providing a complete, GEC-signed, tamper-evident record of the agent's declared intent at each step, the authorization outcome at each step, the retry reasoning articulation, the HEM escalation and resolution, and the post-HEM execution.

6. Enriched DENY Response

6.1. Structure

When Cedar evaluation results in DENY, a conforming GEC implementation MUST return an Enriched DENY Response rather than a bare authorization failure. The Enriched DENY Response is a JSON object:

```
{
  "result":            "DENY",   ; REQUIRED.
  "deny_code":         string,   ; REQUIRED. See Section 6.2.
  "deny_reason":       string,   ; REQUIRED. Human-readable.
  "idp_received":      object,   ; REQUIRED. Echo of submitted IDP.
  "available_actions": [string], ; REQUIRED. Cedar actions currently
                                 ; permitted for this agent and SO.
  "suggested_paths":   [object], ; OPTIONAL. See Section 6.2.
  "hem_available":     boolean,  ; REQUIRED.
  "prior_denial_count": integer, ; REQUIRED. GEC-derived count of
                                 ; prior denials for this
                                 ; requested_action in this session.
  "mismatch_detail":   object,   ; CONDITIONAL. Present when deny_code
                                 ; is IDP_MISSION_REF_MISMATCH.
  "timestamp":         string    ; REQUIRED. ISO 8601 UTC.
}
```

6.2. Field Definitions

deny_code:
A machine-readable denial code. The following codes are defined:

POLICY_DENY	Cedar policy denied the action.
MANDATE_SCOPE	Action is outside the mandate's Cedar scope.
SO_STATE_INVALID	Action is not valid in the current SO state.
HEM_PENDING	Session is in HEM_PENDING; no transitions permitted until HEM is resolved.
MANDATE_REVOKED	Mandate JWT has been revoked.
CROSS_PRINCIPAL_UNRESOLVED	Cross-principal coordination required.

MISSION_INVALID	The governing MissionDeclaration is in a terminal phase (SUSPENDED, FAILED, or ABANDONED). No transitions are permitted until the mission state is resolved.
IDP_MISSION_REF_MISMATCH	IDP mission_ref does not match the active session MissionDeclaration. See mismatch_detail.
RETRY_LIMIT_EXCEEDED	The prior_denial_count for this requested_action has reached the Cedar policy threshold. HEM escalation is required before further retries.

deny_reason:
A human-readable explanation of why the action was denied. SHOULD be specific enough to guide the agent's next reasoning step. MUST NOT expose internal policy structure that would enable policy circumvention.

idp_received:
The complete IDP object as received and validated by the GEC. Allows the agent to verify that its declaration was received accurately.

available_actions:
An array of Cedar action strings that the agent's current mandate permits in the current SO state. MAY be empty if no actions are currently available. This is the primary mechanism by which the
primary mechanism by which the GEC guides agents away from prohibited actions toward p
ermitted
ones, eliminating the need for uninformed retry against denied actions.

suggested_paths:
An OPTIONAL array of path objects, each representing a sequence of actions that would lead the agent to a state where the denied action becomes available. Each path object contains:

```
{
  "path_id":          string,  ; UUID.
  "steps":           [string], ; Ordered Cedar action strings.
  "requires_elevation": boolean
}
```

hem_available:
A boolean indicating whether the agent may trigger HEM by resubmitting with hem_urgency: REQUIRED. False if the session is already in HEM_PENDING or if HEM is not configured for this SO Type.

prior_denial_count:
The GEC-derived count of prior DENY responses for the same requested_action within the current session. Included in the Enriched DENY Response to enable the agent to assess whether further retry is likely to be productive and to decide whether to submit a RETRY_CONTINUATION IDP or to escalate via HEM. When prior_denial_count reaches a Cedar policy threshold, the deny_code RETRY_LIMIT_EXCEEDED is returned and HEM escalation is required.

mismatch_detail:
CONDITIONAL. Present when deny_code is IDP_MISSION_REF_MISMATCH. Contains:

```
{
  "expected_mission_ref": string,  ; mission_ref from mandate JWT.
  "submitted_mission_ref": string  ; mission_ref from IDP.
```

```
}
```

Enables the submitting agent to diagnose and correct the mismatch without ambiguity.

6.3. HEM Routing from DENY

A Cedar DENY result with deny_code POLICY_DENY does not automatically trigger HEM. However, the agent MAY respond to a DENY by resubmitting the same action with hem_urgency: REQUIRED. If hem_available is true in the Enriched DENY Response, the GEC MUST accept the resubmission and enter HEM_PENDING, routing the decision to a designated human principal.

This provides a normative escalation path for situations where the agent's Cedar-authorized scope does not include the needed action but the agent has grounds to believe that human judgment may permit it.

When deny_code is RETRY_LIMIT_EXCEEDED, the GEC MUST automatically enter HEM_PENDING for the session. The agent MUST NOT submit further retries for the denied action until HEM is resolved. The HEM Escalation Request MUST include the prior_denial_count and the sequence of IDP_SUBMITTED events for the denied action, enabling the human principal to review the full retry history.

7. IDP Thin Profile

Some agents -- particularly narrow executors operating on simple tasks -- may have limited capability to produce full IDP_STANDARD declarations. The IDP_THIN profile accommodates these agents while maintaining the core audit property of the IDP.

An IDP_THIN submission MUST contain:

* idp_id (REQUIRED)
* session_id (REQUIRED)
* so_id (REQUIRED)
* mandate_id (REQUIRED)
* step_sequence (REQUIRED)
* requested_action (REQUIRED)
* profile: "IDP_THIN" (REQUIRED; absent in IDP_STANDARD)
* timestamp (REQUIRED)

An IDP_THIN submission MUST omit:

* declared_goal (OPTIONAL in IDP_THIN; GEC synthesizes stub)
* reasoning_basis (OPTIONAL in IDP_THIN; GEC records "UNSPECIFIED")
* confidence_level (OPTIONAL in IDP_THIN; GEC records 0.5 default)
* hem_urgency (OPTIONAL in IDP_THIN; GEC defaults to "NONE")
* mission_ref (OPTIONAL in IDP_THIN; GEC records null)

The audit_accessible field is OPTIONAL in IDP_THIN; GEC defaults to true if absent.

The GEC MUST record the IDP_THIN profile in the IDP_SUBMITTED event. The GEC SHOULD synthesize stub values for absent fields to maintain Event Log schema consistency.

A conforming SO Type MAY declare that IDP_THIN submissions are not accepted for transitions of a given type. In this case, a GEC receiving an IDP_THIN submission for a prohibited transition MUST return REJECT with error code IDP_THIN_NOT_ACCEPTED.

IDP_THIN submissions are subject to IDP Commitment Verification (Section 5.5). The requested_action field is present in IDP_THIN and is the basis for the commitment comparison.

IDP_THIN submissions MUST NOT use reasoning_basis.type RETRY_CONTINUATION. A conforming GEC MUST return REJECT with error code IDP_THIN_NOT_ACCEPTED if an IDP_THIN submission includes reasoning_basis.type RETRY_CONTINUATION. Retry semantics require the full reasoning articulation of IDP_STANDARD.

Note: This specification identifies thin profile GEC compensation as an area for future work. Mechanisms by which the GEC may synthesize richer IDP content from contextual signals for thin-profile agents are outside the scope of this document.

8. Security Considerations

8.1. IDP as Declaration, Not Proof

The IDP records the agent's declared intent; it does not verify it. A compromised or adversarial agent may submit an IDP that does not accurately represent its actual reasoning. The security value of the IDP lies not in proof of intent but in the permanent, pre-action record that creates accountability: an agent that acts contrary to its declared intent produces an auditable inconsistency between the IDP record and the observed action sequence. The IDP Commitment Verification mechanism (Section 5.5) detects this inconsistency automatically.

8.2. Tamper Evidence

The IDP_SUBMITTED Event Log entry MUST be GEC-signed before Cedar evaluation executes. This prevents retroactive modification of the intent record. Implementations MUST use an asymmetric signing key held by the GEC attestation component and MUST NOT permit agents to write directly to the Event Log.

8.3. Inference from Denials

The Enriched DENY Response reveals information about permitted actions and authorization policy structure. Implementations MUST ensure that available_actions and suggested_paths do not reveal policy conditions that would enable adversarial agents to reverse-engineer policy bypass strategies. The deny_reason field MUST be informative without being exploitable.

The prior_denial_count field in the Enriched DENY Response does not reveal policy structure; it reveals only how many times the same action has been denied in this session. This value is safe to expose to the agent.

8.4. Replay Prevention

The idp_id uniqueness requirement (Section 5.2(c)) prevents replay of previously committed IDP objects. Implementations MUST maintain an in-memory index of committed idp_ids for the lifetime of each governed object, rebuilt from the Event Log on GEC restart.

8.5. Delegation Chain Integrity

The IDP's mandate_id field links the intent declaration to the authorization credential. In multi-hop delegation chains, the mandate_id identifies the specific delegation under which the action is requested. This enables audit tracing of intent across delegation trees.

In systems implementing the actor chain model [I-D.mcguinness-oauth-actor-profile], the mandate JWT carries a delegation_chain claim identifying each principal in the issuance chain. The IDP's mandate_id field, which references the jti of the presented mandate JWT, therefore provides a complete traceability link from the per-transition intent declaration to the full delegation chain. Audit tools reconstructing agent reasoning SHOULD traverse the delegation_chain of the referenced mandate JWT to identify all principals in the chain of authority.

8.6. Thin Profile Risk

IDP_THIN submissions provide reduced accountability because the reasoning record is minimal. High-value or high-risk transitions SHOULD require IDP_STANDARD submissions. SO Type designers SHOULD use the IDP_THIN_NOT_ACCEPTED mechanism (Section 7) for transitions where reasoning accountability is critical. IDP_THIN submissions MUST NOT carry RETRY_CONTINUATION reasoning basis type (Section 7).

8.7. Commitment Verification Integrity

The IDP Commitment Verification Record (Section 5.5) is GEC-generated and GEC-signed. Agents MUST NOT be able to influence the match_result field. Implementations MUST ensure that the comparison between requested_action and the executed transition action string occurs entirely within the GEC, with no agent-accessible interface to the comparison logic or its result.

8.8. Retry Loop Exhaustion Attacks

An adversarial agent may attempt to exhaust system resources by submitting continuous RETRY_CONTINUATION IDPs against a denied action, generating Event Log entries without advancing a legitimate goal. Implementations MUST enforce the Cedar policy threshold for prior_denial_count (Section 5.4) and MUST enter HEM_PENDING automatically when RETRY_LIMIT_EXCEEDED is triggered (Section 6.3). The HEM escalation requirement ensures that human review interrupts pathological retry behaviour.

9. Conformance Levels

9.1. Overview

IDP conformance is defined at three levels. All three levels share the same IDP wire format (Section 4.1), the same pre-action submission requirement (Section 5.1), and the same Enriched DENY Response structure (Section 6). They differ in the implementation form of the Governing Enforcement Component (GEC) and in the resulting non-suppressibility guarantee of the Event Log.

The three levels allow application developers to implement Level 1 today using standard SDK libraries, enterprise operators to implement Level 2 using container or sidecar deployment, and regulated high-risk AI system operators to implement Level 3 using hardware attestation infrastructure. All three levels are IDP-conformant.

Level 1 -- Application Profile:
: GEC implemented as a library or middleware component within the agent execution environment. Non-suppressibility is probabilistic. Compensated by SCITT corroboration.

Level 2 -- Isolated Profile:
: GEC implemented as a separate process or sidecar with software-

enforced isolation. Signing key inaccessible to agent process. Non-suppressibility is architectural.

Level 3 -- Kernel Profile:
GEC runs in a RATS-attested hardware execution environment. Signing key bound to TEE via RATS attestation. Non-suppressibility is hardware-enforced.

9.2. Level 1 -- Application Profile

The Application Profile is the entry-level conformance tier for developers implementing IDP within an existing application stack without deploying new infrastructure. SDK libraries, middleware interceptors, in-process enforcement modules, and agent framework plugins are all conforming Level 1 GEC implementations.

Level 1 GEC implementations MUST:

(a) Accept the IDP as part of the Transition Request (Section 5.1) and commit an IDP_SUBMITTED entry to the Event Log before the governed action executes. The calling convention MUST ensure that the action is not invoked if the GEC call fails or if IDP validation fails (Section 5.2).

(b) Return an Enriched DENY Response (Section 6) for every denied action, including the available_actions, prior_denial_count, and idp_received fields.

(c) Commit STATE_TRANSITIONED, CEDAR_DENY_RECORDED, and ACTION_RESULT_RECORDED entries (Section 5.6) for every transition.

(d) Sign all Event Log entries. At Level 1, the signing key MAY be application-managed. The kernel_signature field MUST be present. The signature MUST include a label sub-field with value "L1-app-signed".

(e) Enforce RETRY_CONTINUATION requirements (Section 4.3) and maintain the GEC-derived prior_denial_count attribute (Section 5.4) across Transition Requests within a session.

(f) Enforce data_residency constraints (Section 4.2) in all Event Log queries and analytics contributions.

Level 1 GEC implementations SHOULD:

(g) Register Session Audit Records as SCITT Signed Statements. SCITT corroboration is the primary mechanism by which Level 1 achieves externally-verifiable non-suppressibility without process isolation.

(h) Implement IDP Commitment Verification (Section 5.5). Commitment Verification is RECOMMENDED at Level 1.

Non-suppressibility: PROBABILISTIC. SCITT corroboration converts to externally-verifiable detection of suppression.

Signature label: "L1-app-signed".

Tier 2 analytics default: operator-configurable.
Tier 3 analytics default: opt-in only (tier3_eligible: false).

9.3. Level 2 -- Isolated Profile

The Isolated Profile requires that the GEC run as a separate process from the agent with OS-enforced memory isolation between GEC and

agent. The signing key MUST be held by the GEC process and MUST NOT be accessible to the agent process by any interface.

Level 2 GEC implementations MUST:

(a) All Level 1 MUST requirements.

(b) Run as a process separate from the agent with OS-level memory isolation. In containerized deployments, a dedicated GEC container with no shared memory with the agent container is conforming. A sidecar with mTLS to the agent is conforming.

(c) Hold the signing key in a memory space inaccessible to the agent process. The agent MUST NOT be able to read, modify, or delete Event Log entries directly; all interaction MUST be mediated by the GEC Transition Request interface.

(d) Implement IDP Commitment Verification (Section 5.5). At Level 2, Commitment Verification is REQUIRED.

(e) Register Session Audit Records in a SCITT Transparency Service.

Non-suppressibility: ARCHITECTURAL. Requires independently compromising the GEC process.

Signature label: "L2-isolated-signed".

9.4. Level 3 -- Kernel Profile

The Kernel Profile is the reference implementation level and the normative compliance baseline for regulated high-risk AI systems. The GEC runs in a RATS-attested Trusted Execution Environment (TEE).

Level 3 GEC implementations MUST:

(a) All Level 2 MUST requirements.

(b) Run the GEC within a hardware-attested TEE. Conforming technologies include Intel TDX, AMD SEV-SNP, ARM TrustZone, and equivalent hardware isolation technologies.

(c) Attest the GEC execution environment using RATS [I-D.ietf-rats-architecture]. Attestation results MUST conform to [I-D.ietf-rats-ar4si]. The attestation result reference MUST be included in the Session Audit Record.

(d) Bind the signing key to the TEE attestation environment such that the key MUST NOT be usable outside the verified TEE context.

(e) Register Session Audit Records in a SCITT Transparency Service backed by a CCF-profile implementation or equivalent attested execution environment.

Non-suppressibility: HARDWARE-ENFORCED.

Signature label: "L3-kernel-signed".

9.5. Regulatory Applicability

This section is informative and does not constitute legal advice.

EU AI Act Article 12 (logging requirements for high-risk AI systems):

> Level 1 without SCITT: INSUFFICIENT for Article 12 for high-risk systems.

Level 1 with SCITT: MAY satisfy Article 12 for lower-risk categories of high-risk systems. Operators SHOULD obtain a legal opinion before relying on Level 1 for Article 12 compliance.

Level 2: SHOULD satisfy Article 12 for most high-risk AI system categories.

Level 3: RECOMMENDED for Article 12 compliance for the highest-risk categories including autonomous systems, critical infrastructure, and financial decision AI.

EU AI Act Article 14 (human oversight):
HEM escalation requirements (Section 6.3) are equally mandatory at all three conformance levels.

NIST AI RMF GOVERN 1.5, MANAGE 2.2:
All three levels satisfy the GOVERN 1.5 and MANAGE 2.2 mappings. Level 3 is RECOMMENDED for MANAGE 4.1 (monitoring of deployed systems) in high-stakes contexts.

9.6. Conformance Level Signalling

(a) Every Event Log entry's kernel_signature field MUST include a label sub-field with value "L1-app-signed", "L2-isolated-signed", or "L3-kernel-signed" as appropriate.

(b) A governed SO Type MAY declare a minimum required conformance level. A GEC that receives a Transition Request for an SO Type requiring a higher conformance level MUST return REJECT with error code GEC_LEVEL_INSUFFICIENT.

(c) The GEC SHOULD expose its declared conformance level via the GEC Query Interface. Audit Principals SHOULD verify declared level against signature labels in sampled Event Log entries.

The deny code GEC_LEVEL_INSUFFICIENT is added to the IDP Deny Codes IANA registry (Section 12).

10. Privacy Considerations

The IDP contains agent-generated text in the declared_goal.description and reasoning_basis.description fields. These fields MUST NOT contain personally identifiable information. The GEC SHOULD validate that these fields do not contain obvious PII markers (email addresses, phone numbers, names), but CANNOT guarantee the absence of PII through syntactic validation alone.

Implementers MUST ensure that Event Log access controls prevent unauthorized parties from reading IDP records. The audit_accessible field (Section 4.1) provides a mechanism for restricting Audit Principal access to individual IDP records. IDP records are governance data, not publicly accessible telemetry.

IDP records associated with governed objects that are subject to data protection erasure requests present a tension with the Event Log's append-only property. This tension is addressed through the Zone A personal data prohibition and cryptographic erasure provisions specified in the GEC implementation.

The reasoning_basis.description field in RETRY_CONTINUATION IDPs MAY contain references to prior failed actions that themselves contain contextual information. Implementers SHOULD treat RETRY_CONTINUATION reasoning descriptions with the same PII sensitivity as primary IDP descriptions.

11. EU AI Act Applicability

This section is informative.

EU AI Act Article 12 requires that high-risk AI systems maintain automatic logging sufficient for post-deployment monitoring and human review of AI decisions. The IDP, committed to a GEC-signed Event Log before each agent action, provides the per-decision record that Article 12 contemplates.

EU AI Act Article 14 requires human oversight measures enabling intervention in the operation of high-risk AI systems. The HEM escalation path triggered by hem_urgency: REQUIRED, and the HEM routing from Cedar DENY responses (Section 6.3), provide the normative intervention mechanism that Article 14 requires. The HEM protocol is fully specified in [I-D.sato-soos-hem].

The IDP Commitment Verification mechanism (Section 5.5) contributes to Article 14(4)(c) compliance: automatic detection of anomalous situations (commitment gaps between declared intent and actual action) with mandatory human escalation.

The RETRY_CONTINUATION reasoning basis type and the prior_denial_count threshold mechanism provide an additional contribution to Article 14 compliance: automatic detection of pathological agent retry behaviour with mandatory human escalation after a configurable threshold, ensuring that humans can intervene before agents exhaust their retry budget against a denied action class.

This document does not constitute legal advice and makes no representation that conforming implementations satisfy any regulatory requirement.

12. IANA Considerations

This document requests the creation of the following IANA registry:

Registry Name:
IDP Reasoning Basis Types

Registration Procedure:
Specification Required [RFC8126]

Initial Values:

Type Value	Description
RULE_BASED	Action selected by applying a defined rule to the current state
INFERENCE	Action inferred from contextual reasoning
INSTRUCTION	Action executing an explicit instruction from a human principal or higher-authority agent
UNCERTAINTY_REDUCTION	Action taken to reduce uncertainty before selecting a goal-advancing action
MISSION_STAGE	Action advancing the governing MissionDeclaration to the next stage; mission_ref MUST be present
RETRY_CONTINUATION	Agent retrying an action following a prior denial or failure; context_refs MUST reference prior IDP_SUBMITTED event;

		reasoning_basis.description MUST articulate what has changed since the prior attempt; NOT permitted in IDP_THIN

Table 1: Initial IDP Reasoning Basis Types Registry Values

This document also requests the creation of the following IANA registry:

Registry Name:
IDP Deny Codes

Registration Procedure:
Specification Required [RFC8126]

Initial Values:

Deny Code	Description
POLICY_DENY	Cedar policy denied the action
MANDATE_SCOPE	Outside mandate Cedar scope
SO_STATE_INVALID	Invalid in current SO state
HEM_PENDING	Session in HEM_PENDING state
MANDATE_REVOKED	Mandate JWT revoked
CROSS_PRINCIPAL_UNRESOLVED	Cross-principal pending
MISSION_INVALID	MissionDeclaration in terminal phase; no transitions permitted
IDP_MISSION_REF_MISMATCH	IDP mission_ref mismatch; see mismatch_detail
IDP_MISSING	IDP absent from call
IDP_MALFORMED	IDP failed schema validation
IDP_DUPLICATE	Duplicate idp_id detected
IDP_SO_MISMATCH	IDP so_id / mandate mismatch
IDP_MANDATE_MISMATCH	IDP mandate_id / JWT mismatch
IDP_THIN_NOT_ACCEPTED	IDP_THIN not accepted here
RETRY_LIMIT_EXCEEDED	prior_denial_count at Cedar policy threshold; HEM required
GEC_LEVEL_INSUFFICIENT	SO Type requires higher GEC conformance level than provided

Table 2: Initial IDP Deny Codes Registry Values

13. References

13.1. 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>.

[RFC4122] Leach, P., Mealling, M., and R. Salz, "A Universally Unique IDentifier (UUID) URN Namespace", RFC 4122, DOI 10.17487/RFC4122, July 2005, <https://www.rfc-editor.org/rfc/rfc4122>.

[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-hem] Sato, T., "The Human Escalation Mechanism (HEM) for Agentic AI Systems", Work in Progress, Internet-Draft, draft-sato-soos-hem-01, May 2026, <https://datatracker.ietf.org/doc/draft-sato-soos-hem/>.

13.2. 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.ietf-oauth-v2-1] Hardt, D., Parecki, A., and T. Lodderstedt, "The OAuth 2.1 Authorization Framework", Work in Progress, Internet-Draft, draft-ietf-oauth-v2-1, 2026, <https://datatracker.ietf.org/doc/draft-ietf-oauth-v2-1/>.

[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 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/>.

[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-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.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/>.

[Williams-IBA]
Williams, J., "Intent Bound Authorization",
GB2603013.0, PCT pending, 2025.

Acknowledgments

The IDP design draws on the Windley Loop context injection model and the Cedar policy evaluation architecture [Cedar]. The directional distinction from Williams IBA was identified through review of [Williams-IBA]. The mission_ref field and MISSION_STAGE reasoning type were added following review of the McGuinness Mission Bound Authorization framework [I-D.mcguinness-oauth-mission-bound-authorization] and the actor chain model [I-D.mcguinness-oauth-actor-profile]. The IDP Commitment Verification mechanism was developed in conjunction with the Governance Audit Record specification [I-D.sato-soos-gar]. The RETRY_CONTINUATION reasoning basis type and the computational efficiency motivation in Section 1 and Section 3.1(e) were developed through operational analysis of LLM agent execution patterns, specifically the observation that agents lacking a pre-action commitment discipline for retry reasoning exhibit structurally redundant computation at scale.

Author’s Address

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