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

T. Sato
MyAuberge K.K.

24 May 2026

# The Sovereign Object (SOV) for Agentic AI Systems
draft-sato-soos-sov-00

## Abstract

Agentic AI governance protocols -- including intent declaration, human escalation, audit recording, and constitutional prohibition -- all require a normative definition of the governed resource that agents operate on: the structured, stateful, policy-carrying entity to which agent authority is bound and upon which governed transitions execute. No existing IETF specification defines this primitive.

This document defines the Sovereign Object (SO): a causally ordered, policy-governed, typed, living document that evolves through a predefined finite state space under Governing Enforcement Component (GEC) authority. The SO is the unit of governance in the SOOS protocol family: the thing agents operate on, the GEC governs, and human principals reason about.

This document specifies the SO's five-layer structure (Identity, State, Event Stream, Typed Graph, Attachment Index), its Zone A / Zone B boundary model, its five-phase lifecycle, its SO Type system, its Cedar policy context model, and the binding model by which a Mandate JWT binds an agent to a specific SO instance. The Sovereign Object is the architectural foundation referenced normatively by [I-D.sato-soos-idp], [I-D.sato-soos-hem], [I-D.sato-soos-gar], [I-D.sato-soos-cap], and [I-D.sato-soos-mjwt].

## 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

## 1. Introduction

The IETF community has made significant progress in specifying how AI agents authenticate [I-D.ietf-wimse-arch], how they express intent before acting [I-D.sato-soos-idp], how human oversight is enforced [I-D.sato-soos-hem], and how agent actions are recorded for audit [I-D.sato-soos-gar]. Each of these specifications refers, explicitly or implicitly, to a governed resource: the entity that the agent is acting upon, that the authorization policy applies to, and that the human principal is reasoning about.

None of them defines that resource.

The Intent Declaration Primitive [I-D.sato-soos-idp] requires an agent to declare what governed object its action targets. The Human Escalation Mechanism [I-D.sato-soos-hem] triggers escalation events bound to the state of a specific governed object instance. The Governance Audit Record [I-D.sato-soos-gar] records transitions on governed objects. The forthcoming Mandate JWT specification

[I-D.sato-soos-mjwt] binds agent authority to a specific governed object instance. In every case, the governed resource is assumed but not specified.

This document fills that gap. The Sovereign Object (SO) is the unit of governance in the SOOS protocol family: a causally ordered, policy-governed, typed, living document that evolves through a predefined finite state space under GEC authority. It is the thing agents operate on, the GEC governs, and human principals reason about.

The SO is not a database record. A database record is a passive data container. The SO carries its own state machine, its own Cedar policy context, its own tamper-evident event history, and its own lifecycle -- including eventual cryptographic erasure. It is the architectural resource that gives SOOS its coherence: the abstraction layer below which protocol-specific implementation details vary and above which governance, authorization, audit, and human oversight operate uniformly.

The SO is also the primitive that gives Mandate JWT binding its specificity. An agent is not authorized to perform actions of type X in the abstract. It is authorized to perform actions of type X on SO instance Y, in lifecycle phase P, under human principal H. That binding is what makes the authorization auditable, revocable, and scoped.

The design of the Sovereign Object draws on the insight that governed resources in agentic systems have properties that generic URI-addressed API endpoints do not: they have state machines, they carry policy context, they accumulate causal event histories, and they have lifecycles that include regulatory obligations upon termination (erasure under GDPR Article 17 [GDPR], APPI Article 19 [APPI], and equivalent). This document specifies those properties normatively.

## 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:

Sovereign Object (SO):
: A causally ordered, policy-governed, typed, living document that evolves through a predefined finite state space under GEC authority. The unit of governance in the SOOS protocol family.

SO Instance:
: A specific instantiation of an SO Type with a globally unique identifier, its own event stream, and its own current lifecycle state. An SO Instance is the entity to which a Mandate JWT binds agent authority.

SO Type:
: A declaration that specifies the state machine, Zone A schema, Cedar policy set, and permissible attachment types for a class of Sovereign Objects. SO Types are registered in the SO Type Registry.

SO Type Registry:
: A registry of SO Type declarations, analogous to the IANA MIME type registry, that enables interoperating implementations to discover and validate SO Type definitions.

Zone A:
: The Protocol Core zone of an SO. GEC-governed. Schema determined by the SO Type state machine. Required nodes enforced as invariants. Personal data MUST NOT be stored in Zone A.

Zone B:
: The Attached Periphery zone of an SO. Agent-accessible, GEC-referenced, but not directly GEC-governed. Arbitrarily complex external content referenced by integrity-verified attachment records in the Attachment Index.

SO Lifecycle Phase:
: One of five phases that every SO Instance traverses: ACTIVE, OPERATIONALLY_COMPLETE, ADMINISTRATIVELY_CLOSED, ARCHIVED, and CRYPTOGRAPHICALLY_ERASED.

Event Stream:
: The append-only, causally ordered, tamper-evident log of all transitions executed against an SO Instance. The Event Stream is the ground truth of the SO's history.

Typed Graph:
: The current materialization of an SO Instance's Event Stream as a labeled property graph. Derived from the Event Stream; in case of conflict, the Event Stream is authoritative.

Attachment Index:
: The set of integrity-verified references to Zone B content associated with an SO Instance.

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 Log, and mediates agent access to governed objects.

Mandate JWT:
: As defined in [I-D.sato-soos-mjwt]: a JSON Web Token that binds an agent's authorization to a specific SO Instance.

Cedar:
: A policy language and evaluation engine [Cedar] used by the GEC to evaluate authorization decisions against SO Instance state and agent intent.

Human Principal:
: A natural person who holds authority over an SO Instance, as identified in the Mandate JWT human_principal_id field.

Narrowing Property:
: The invariant by which a child Mandate JWT is always a strict subset of its parent in all authorization dimensions.

# 3. Problem Statement

## 3.1. The Missing Governed Resource Primitive

SOOS governance protocols -- IDP, HEM, GAR, CAP, and MJWT -- are built around the concept of a governed resource: a typed, stateful entity that agents operate on under GEC authority. The governance properties of the SOOS stack -- auditability, revocability, human oversight, constitutional prohibition, and federated trust -- all derive their coherence from the properties of this governed resource.

As of the submission of this document, no IETF specification defines

this resource normatively. The existing SOOS drafts use the term "governed object" descriptively but without a formal definition. This creates three concrete problems:

(a) Binding ambiguity. The Mandate JWT binds agent authority to a governed object instance. Without a formal definition of what a governed object is -- what properties it has, what state it can be in, what lifecycle phases it traverses -- the binding has no normative referent. Implementations cannot interoperate on the meaning of the binding.

(b) Policy imprecision. Cedar policies evaluated by the GEC can reference SO Instance state attributes. Without a normative definition of what state attributes are available and how they are derived from the SO's event history, policy authors cannot write portable policies and evaluators cannot evaluate them consistently.

(c) Lifecycle opacity. Regulatory obligations on governed data -- including data minimization, erasure on request, and audit retention -- apply to the governed resource and its event history. Without a normative definition of the SO's lifecycle, including its Cryptographic Erasure phase, implementations cannot demonstrate compliance with these obligations.

### 3.2. Why Existing Primitives Are Insufficient

Three existing primitive types might appear to serve the role of the Sovereign Object. Each is insufficient for distinct reasons.

URI-addressed API endpoints. Generic REST endpoints are passive: they accept requests and return responses. They have no state machine, no causal event history, no Cedar policy context, and no lifecycle. An agent authorized to invoke an endpoint is not authorized to perform a specific state transition on a specific stateful resource under human principal oversight. The authorization semantics are categorically different.

Database records. A database record is a passive data container. It may carry a status field that approximates state, but that field is not enforced by a GEC, is not derived from a tamper-evident event history, and does not carry Cedar policy context. The record's "state" can be overwritten without triggering HEM escalation, without generating a GAR audit entry, and without verifying that the agent holds a valid Mandate JWT authorizing the transition.

vCon [I-D.ietf-vcon-vcon-container]. The vCon container records the history of a completed conversation or interaction. It is retrospective: it captures what happened after the fact. The Sovereign Object is prospective: it carries the Cedar policy context, lifecycle state, and mandate tree that governs agent behavior before and during execution. The SO does not replace vCon; the two primitives are complementary. A vCon carrying IDP, GAR, and HEM records can be produced from an SO's Event Stream after a session completes. The SO is the source of those records; vCon is a portable container for them.

### 3.3. Relationship to Thomas Howe's Agent Architecture Layering

Howe [Howe-Layering] proposes an eight-layer architecture for agentic AI systems covering transport, identity, credential issuance, authorization, orchestration, tool invocation, audit, and oversight. This layering correctly identifies the major functional concerns of agentic governance but does not identify the governed resource as a distinct architectural primitive. In Howe's model,

the Mandate JWT binds to an unspecified resource -- an implicit target of authorization that his layers do not define.

The Sovereign Object is that missing primitive. It sits below the authorization and delegation layer as the structured, stateful, policy-carrying resource that the Mandate JWT refers to and that Cedar policy governs access to. Without the SO, authorization binding is generic: an agent is authorized to perform actions of type X. With the SO, the binding is specific: an agent is authorized to perform actions of type X on SO Instance Y, in lifecycle phase P, under human principal H, while the SO's state machine is in state S. That specificity is what makes the authorization auditable and revocable.

# 4. The Sovereign Object

## 4.1. Definition

A Sovereign Object (SO) is a causally ordered, policy-governed, typed, living document that evolves through a predefined finite state space under Governing Enforcement Component authority.

"Causally ordered" means that every state transition recorded in the Event Stream is causally linked to its predecessor: no transition can be inserted between two existing entries, and no transition can reference a future state.

"Policy-governed" means that every state transition MUST be evaluated by a Cedar policy set defined in the SO's Type declaration before execution. The GEC MUST reject transitions that do not satisfy the applicable Cedar policy.

"Typed" means that every SO Instance MUST conform to a declared SO Type that specifies its state machine, Zone A schema, and Cedar policy set.

"Living" means that the SO's current state is always derived from its Event Stream. The Typed Graph is a materialization of the Event Stream at a given point in time, not an independent data store. The Event Stream is authoritative.

"Finite state space" means that the set of states the SO can occupy is declared in the SO Type and is finite. The GEC MUST reject any transition that would move the SO to a state not declared in its SO Type.

## 4.2. Five-Layer Structure

Every Sovereign Object Instance has five layers. Layers 1 through 3 are mandatory. Layers 4 and 5 are RECOMMENDED for production implementations.

### 4.2.1. Layer 1 -- Identity Layer

The Identity Layer provides the globally unique, immutable identifier for an SO Instance.

so_id:
: REQUIRED. A UUID v7 [RFC9562] identifier. UUID v7 is selected for its time-ordered property, which enables efficient range queries over SO Instances by creation time. The so_id MUST NOT change for the lifetime of the SO Instance, including after Cryptographic Erasure.

so_type_id:

REQUIRED. The identifier of the SO Type this instance conforms to. MUST reference a registered entry in the SO Type Registry.

created_at:
REQUIRED. ISO 8601 timestamp of SO Instance creation.

human_principal_id:
REQUIRED. The identifier of the human principal who holds authority over this SO Instance.

gec_id:
REQUIRED. The identifier of the GEC currently associated with this SO Instance. See Section 9.1.

The Identity Layer MUST be committed to the Event Stream as the first entry when an SO Instance is created. The entry MUST be GEC-signed.

#### 4.2.2. Layer 2 -- State Layer

The State Layer records the SO Instance's current position in its SO Type's state machine.

current_state:
REQUIRED. The current state of the SO Instance. MUST be a state declared in the SO Type's state machine.

current_phase:
REQUIRED. The current SO Lifecycle Phase. One of: ACTIVE, OPERATIONALLY_COMPLETE, ADMINISTRATIVELY_CLOSED, ARCHIVED, CRYPTOGRAPHICALLY_ERASED. See Section 6.

state_entered_at:
REQUIRED. ISO 8601 timestamp of the most recent state transition.

The State Layer is derived from the Event Stream. The GEC MUST maintain a current-state cache for performance, but MUST treat the Event Stream as authoritative in any conflict.

#### 4.2.3. Layer 3 -- Event Stream

The Event Stream is the append-only, causally ordered, tamper-evident log of all transitions executed against an SO Instance. The Event Stream is the ground truth of the SO's history.

Every Event Stream entry MUST include:

event_id:
A UUID v7 identifier for this entry, assigned by the GEC at commitment.

event_type:
One of the event types defined in Section 13.3.

prior_event_id:
The event_id of the preceding entry. MUST be null for the SO_CREATED entry. MUST NOT be null for all subsequent entries. This field enforces causal ordering.

occurred_at:
ISO 8601 timestamp.

gec_signature:
The GEC signature over this entry, covering event_id, event_type, prior_event_id, occurred_at, and all event-type-specific fields. The signing key and label MUST conform to the GEC's conformance

level as defined in [I-D.sato-soos-idp] Section 9.

agent_id:
The identifier of the agent that triggered this event, if applicable. NULL for GEC-initiated events.

mandate_id:
The Mandate JWT identifier authorizing this event, if applicable. NULL for GEC-initiated events.

Event Stream entries MUST be written to a SCITT transparency log [I-D.ietf-scitt-architecture] at Level 3 conformance. At Level 1 and Level 2 conformance, SCITT submission is RECOMMENDED.

#### 4.2.4. Layer 4 -- Typed Graph

The Typed Graph is the current materialization of the Event Stream as a labeled property graph. It provides efficient query access to the SO Instance's current state without requiring full Event Stream replay.

The Typed Graph schema is defined by the SO Type. The GEC MUST update the Typed Graph atomically with each Event Stream commitment. In any conflict between the Typed Graph and the Event Stream, the Event Stream is authoritative.

The Typed Graph MUST expose the following minimum node set:
- The SO Instance identity (Layer 1 fields).
- The current state and phase (Layer 2 fields).
- The set of currently bound Mandate JWTs.
- The current Attachment Index (Layer 5).

#### 4.2.5. Layer 5 -- Attachment Index

The Attachment Index is the set of integrity-verified references to Zone B content associated with an SO Instance.

Every Attachment Index entry MUST include:

attachment_id:
A UUID v7 identifier.

attachment_type:
A type declared in the SO Type's attachment type list.

content_uri:
The URI of the Zone B content.

content_hash:
A SHA-256 [RFC6234] hash of the Zone B content at attachment time. The GEC MUST verify content integrity on access.

attached_at:
ISO 8601 timestamp.

gec_signature:
GEC signature over this Attachment Index entry.

Attachment and detachment operations MUST generate ZONE_B_ATTACHED and ZONE_B_DETACHED Event Stream entries.

### 4.3. Zone A / Zone B Boundary

#### 4.3.1. Zone A -- Protocol Core

Zone A is the Protocol Core of an SO Instance. It is the set of

data fields governed directly by the GEC and constrained by the SO Type's schema.

Properties of Zone A:
- Schema determined by the SO Type state machine.
- Required nodes enforced as invariants by the GEC.
- All writes mediated by the GEC.
- All transitions subject to Cedar policy evaluation.
- All transitions recorded to the Event Stream.

Zone A Invariant (INV-ZA-1):
> Personal data as defined under GDPR Article 4(1) [GDPR], APPI Article 2 [APPI], and equivalent jurisdictional definitions MUST NOT be stored in Zone A. Zone A contains only identifiers, state references, and policy-relevant metadata. Personal data is always Zone B content, referenced from Zone A via the Attachment Index.

This invariant enables Zone A and the Event Stream to be retained for audit purposes after a Cryptographic Erasure event (Section 6.3) while Zone B content is irreversibly erased.

#### 4.3.2. Zone B -- Attached Periphery

Zone B is the Attached Periphery of an SO Instance. It contains external content referenced by the Attachment Index.

Properties of Zone B:
- Arbitrarily complex external content.
- Not directly governed by the GEC.
- Accessible to agents holding appropriate Zone B read capability in their Mandate JWT.
- Content integrity verified by the GEC on access via content_hash.

Zone B content includes, but is not limited to: documents, images, consent records, health declarations, identity verification results, and any other personal data associated with the SO Instance.

#### 4.3.3. Zone Boundary Invariants

INV-ZA-1: Personal data MUST NOT be stored in Zone A.

INV-ZA-2: Zone A fields MUST be defined in the SO Type schema. The GEC MUST reject writes to undefined Zone A fields.

INV-ZB-1: Zone B content MUST be referenced from Zone A via a signed Attachment Index entry.

INV-ZB-2: The GEC MUST verify Zone B content integrity on every access. Content failing integrity verification MUST NOT be returned to the requesting agent.

## 5. SO Type System

### 5.1. SO Type Declaration

An SO Type declaration is a JSON object with the following fields:

```
{
  "so_type_id": "<string: unique SO Type identifier>",
  "so_type_name": "<string: human-readable name>",
  "so_type_version": "<string: semantic version>",
  "state_machine": {
    "states": ["<state_name>", ...],
    "initial_state": "<state_name>",
```

```
    "transitions": [
      {
        "from": "<state_name>",
        "to": "<state_name>",
        "cedar_action": "<action_name>",
        "requires_hem": <boolean>
      }
    ]
  },
  "zone_a_schema": {
    "<field_name>": {
      "type": "<json_schema_type>",
      "required": <boolean>,
      "personal_data": false
    }
  },
  "cedar_policy_set_uri": "<URI of Cedar policy set>",
  "attachment_types": ["<attachment_type_name>", ...],
  "registrant": "<organization name>",
  "registered_at": "<ISO 8601 timestamp>"
}
```

The zone_a_schema MUST NOT define fields with personal_data: true. The GEC MUST reject SO Type registration requests that include personal data fields in the Zone A schema.

The cedar_policy_set_uri MUST resolve to a Cedar policy set that defines permit and forbid rules for all cedar_action values referenced in the state machine transitions.

### 5.2. SO Type Registry

The SO Type Registry is the authoritative catalog of registered SO Types. It is analogous to the IANA MIME type registry in function. IANA Considerations are specified in Section 13.1.

SO Type identifiers use a hierarchical naming convention:

<registrant-prefix>/<type-name>/<version>

Example: "atp/booking-object/1.0"

SO Type identifiers MUST be globally unique within the registry.

### 5.3. SO Type Discovery

A GEC MAY expose an SO Type Discovery endpoint returning the set of SO Types it supports, as a JSON array of SO Type declarations. The discovery endpoint URI is implementation-defined.

## 6. SO Lifecycle

### 6.1. Five Lifecycle Phases

Every SO Instance progresses through a sequence of lifecycle phases. Phase transitions are irreversible.

Phase 1 -- ACTIVE:
Normal operation. All Cedar-governed state transitions permitted, subject to the SO Type's state machine and Cedar policy set.

Phase 2 -- OPERATIONALLY_COMPLETE:
Terminal operational state reached. Agent-initiated state transitions are prohibited. Administrative transitions for billing, audit, and dispute resolution remain permitted subject

to Cedar policy and human principal authority.

Phase 3 -- ADMINISTRATIVELY_CLOSED:
All state transitions prohibited except DISPUTE_OPENED, if defined by the SO Type. Only human principals with explicit dispute authority may initiate transitions.

Phase 4 -- ARCHIVED:
All write operations prohibited. The SO Instance is read-only. The Event Stream is retained. Zone B content retention is governed by the data_residency policy and applicable law.

Phase 5 -- CRYPTOGRAPHICALLY_ERASED:
Zone B content has been irreversibly erased per Section 6.3. Zone A data and the Event Stream are retained for audit. No further phase transitions are possible.

## 6.2. Phase Transition Rules

Phase transitions MUST be initiated by the GEC, not by agents directly. Phase transitions MUST generate a PHASE_TRANSITIONED Event Stream entry recording the prior phase, the new phase, the authorizing human principal (if applicable), and the GEC signature.

Phase transition rules:

ACTIVE -> OPERATIONALLY_COMPLETE:
When the SO Type's state machine reaches a declared terminal state. The GEC MAY initiate this transition automatically.

OPERATIONALLY_COMPLETE -> ADMINISTRATIVELY_CLOSED:
After the administrative tail period declared in the SO Type has elapsed. MUST be authorized by a human principal with lifecycle authority.

ADMINISTRATIVELY_CLOSED -> ARCHIVED:
After all dispute resolution windows and applicable regulatory retention requirements have elapsed. MUST be authorized by a human principal with lifecycle authority.

ARCHIVED -> CRYPTOGRAPHICALLY_ERASED:
At any time after ARCHIVED phase, subject to regulatory retention requirements. MUST be authorized by a human principal with erasure authority, or initiated automatically by the GEC when data_residency.retention_days has elapsed.

## 6.3. Cryptographic Erasure

Cryptographic Erasure irreversibly erases Zone B content by destroying the encryption keys protecting it.

Cryptographic Erasure MUST:

(a) Destroy all encryption keys protecting Zone B content for this SO Instance.

(b) Generate an ERASURE_INITIATED Event Stream entry before beginning key destruction.

(c) Generate an ERASURE_COMPLETED Event Stream entry after key destruction is confirmed.

(d) Retain Zone A data and the complete Event Stream intact.

(e) Update current_phase to CRYPTOGRAPHICALLY_ERASED.

Cryptographic Erasure satisfies the right to erasure under GDPR Article 17 [GDPR] for Zone B content while preserving the Event Stream as an audit record.

## 7. Cedar Policy Context

### 7.1. SO State as Cedar Attribute

The GEC MUST make the following SO Instance attributes available as Cedar context attributes during policy evaluation:

| | |
|---|---|
| so.so_id: | The SO Instance identifier. Type: String. |
| so.so_type_id: | The SO Type identifier. Type: String. |
| so.current_state: | Current state machine position. Type: String. |
| so.current_phase: | Current lifecycle phase. Type: String. |
| so.human_principal_id: | Human principal identifier. Type: String. |
| so.prior_denial_count: | Count of GEC DENY responses targeting this SO Instance in the current session. Type: Long. Composes with the prior_denial_count attribute in [I-D.sato-soos-idp] Section 4.2. |
| so.mandate_count: | Number of active Mandate JWTs bound to this SO Instance. Type: Long. |

### 7.2. Zone Access Policy Model

Zone B read access and Zone B write access are separate Cedar action types and MUST be declared separately in the SO Type's Cedar policy set. A Cedar policy permitting a state transition MUST NOT implicitly permit Zone B access.

### 7.3. Policy Evaluation Order

The GEC MUST evaluate Cedar policies in the following order:

(1) CAP Tier 0 (constitutional) prohibitions [I-D.sato-soos-cap]. A Tier 0 prohibition results in immediate DENY.

(2) CAP Tier 1 (jurisdictional) prohibitions [I-D.sato-soos-cap].

(3) SO Type Cedar policy, with SO Instance state attributes per Section 7.1 and IDP intent attributes per [I-D.sato-soos-idp] Section 5.4.

(4) Mandate JWT scope verification [I-D.sato-soos-mjwt].

A PERMIT requires all four layers to permit the action. A DENY at any layer results in immediate DENY without proceeding further.

## 8. Mandate JWT Binding

### 8.1. Binding Model

A Mandate JWT [I-D.sato-soos-mjwt] binds agent authority to a specific SO Instance with the following properties:

Instance specificity:
A Mandate JWT MUST reference the so_id of a specific SO Instance. A Mandate JWT MUST NOT authorize actions against an SO Type in general.

State awareness:
The Mandate JWT MAY declare the set of SO states in which the authorized Cedar actions may be performed. The GEC MUST reject

Transition Requests referencing Cedar actions outside the state-restricted set for the SO Instance's current state.

Phase restriction:
The Mandate JWT MAY declare lifecycle phase restrictions. The GEC MUST reject Transition Requests targeting an SO Instance in a phase not permitted by the Mandate JWT.

Temporal validity:
The Mandate JWT MUST carry a standard JWT exp claim. The GEC MUST reject expired Mandate JWTs.

Human principal linkage:
The Mandate JWT MUST carry the human_principal_id of the SO Instance's human principal. The GEC MUST verify that this value matches the human_principal_id in the SO Instance's Identity Layer.

### 8.2. Narrowing Property at the SO Level

The Narrowing Property is a normative invariant: a child Mandate JWT MUST be a strict subset of its parent in all authorization dimensions. At the SO level:

(a) SO Instance scope: A child mandate MUST reference the same so_id or a subset of SO Instances covered by its parent.

(b) Cedar action scope: A child mandate's authorized action set MUST be a subset of its parent's.

(c) State restrictions: A child mandate's permitted SO states MUST be a subset of its parent's.

(d) Phase restrictions: A child mandate's permitted lifecycle phases MUST be a subset of its parent's.

(e) Temporal validity: A child mandate's exp claim MUST NOT be later than its parent's.

The GEC MUST verify the Narrowing Property when a child Mandate JWT is presented. A violation MUST result in a NARROWING_VIOLATION deny code.

### 8.3. Binding Verification

On receiving a Transition Request, the GEC MUST verify before Cedar evaluation:

(a) The Mandate JWT signature MUST be valid.
(b) The Mandate JWT MUST not be expired.
(c) The so_id in the Mandate JWT MUST match the targeted SO Instance.
(d) The human_principal_id MUST match the SO Instance Identity Layer.
(e) The Mandate JWT MUST not appear in the revocation registry.
(f) If a child mandate, the Narrowing Property MUST be verified.

Binding verification failure MUST result in a DENY response recorded in the SO Instance's Event Stream.

## 9. GEC Association

### 9.1. GEC-SO Association Model

Every SO Instance MUST be associated with exactly one GEC at any given time, recorded as gec_id in the Identity Layer. The GEC is responsible for: evaluating all Transition Requests, maintaining the Event Stream, enforcing Cedar policy, executing HEM escalation,

generating GAR audit records, enforcing CAP prohibitions, and managing Mandate JWT revocation.

A GEC transition MUST generate a GEC_TRANSITIONED Event Stream entry signed by both the outgoing and incoming GEC. During a GEC transition, the SO Instance MUST be placed in a non-transitionable state until the new GEC confirms association.

### 9.2. Multi-GEC Coordination

A single SO Instance MUST NOT be governed by more than one GEC simultaneously. A trust domain MAY designate shadow GECs for resilience; shadow GECs MUST receive Event Stream entries in real time but MUST NOT accept Transition Requests except following formal failover. Multi-GEC coordination protocols are outside the scope of this document.

## 10. Relationship to Other SOOS Drafts

IDP [I-D.sato-soos-idp]:
: The "governed object" in IDP is an SO Instance as defined in this document. The IDP context_refs field may reference SO Event Stream entries by event_id. The so.prior_denial_count Cedar attribute (Section 7.1) composes with the prior_denial_count attribute defined in [I-D.sato-soos-idp] Section 4.2.

HEM [I-D.sato-soos-hem]:
: HEM events are bound to SO Instances. HEM_ESCALATED and HEM_RESOLVED events MUST be recorded in the SO Instance's Event Stream. The HEM_PENDING state prohibits all agent-initiated Transition Requests against the SO Instance until a human principal provides a decision.

GAR [I-D.sato-soos-gar]:
: Governance Audit Records are generated from SO Instance Event Stream entries. The SO's Cryptographic Erasure event MUST be recorded as a Type 1 GAR entry.

CAP [I-D.sato-soos-cap]:
: CAP prohibitions are enforced at the SO level before Cedar evaluation per Section 7.3.

MJWT [I-D.sato-soos-mjwt]:
: The Mandate JWT binds agent authority to a specific SO Instance per Section 8. The Mandate JWT's so_id field is the normative reference to the SO Instance Identity Layer.

FAIP [I-D.sato-soos-faip]:
: The Federated Agent Intelligence Protocol specifies cross-domain federated analytics derived from SO Instance Event Streams. The data_residency field in [I-D.sato-soos-idp] Section 4.1 controls per-record FAIP tier eligibility.

## 11. Security Considerations

SO Instance identifiers (so_id) are UUID v7 values carrying timing information. Implementations that consider SO creation timing sensitive MUST treat so_id values as sensitive and avoid exposing them in unauthenticated contexts.

The Event Stream is the ground truth of an SO Instance's history. Implementations MUST protect Event Stream integrity through GEC signing of every entry. A tampered Event Stream MUST be treated as a critical security incident triggering HEM Class 1 escalation.

Zone B content integrity depends on the security of the Zone B storage system. Implementations MUST ensure Zone B storage provides integrity guarantees appropriate to the sensitivity of stored content.

The GEC-SO association model (Section 9) creates a single point of authority over an SO Instance. Implementations MUST protect GEC signing keys at the conformance level declared by the GEC per [I-D.sato-soos-idp] Section 9.

Mandate JWT binding verification (Section 8.3) MUST precede Cedar policy evaluation. A failed binding verification MUST generate an Event Stream entry.

The Narrowing Property (Section 8.2) is a security invariant. Any implementation permitting a child mandate to exceed its parent's scope creates an authorization escalation vulnerability.

## 12. Privacy Considerations

Zone A Invariant INV-ZA-1 prohibits personal data in Zone A. This is the primary privacy protection in the SO design: it ensures that Zone A data and the Event Stream can be retained for audit after Cryptographic Erasure without retaining personal data.

Zone B content MUST be encrypted at rest. Cryptographic Erasure (Section 6.3) satisfies erasure obligations under GDPR Article 17 [GDPR] and APPI Article 19 [APPI] by destroying Zone B encryption keys.

The human_principal_id field is a persistent identifier for a natural person. Access MUST be controlled by Cedar policy.

The Event Stream may contain agent identifiers and action records that constitute personal data under applicable law. Implementations MUST apply appropriate access controls to Event Stream queries and ensure retention periods comply with applicable data retention law.

## 13. IANA Considerations

### 13.1. SO Type Registry

Registry name: Sovereign Object Type Registry
Registration procedure: Specification Required.
Initial registrations: None. The first registration is expected from the ATP Foundation following RFC publication, registering "atp/booking-object" as the reference implementation SO Type.

### 13.2. SO Lifecycle Phase Registry

Registry name: Sovereign Object Lifecycle Phase Registry
Registration procedure: Standards Action.

Initial registrations:

| Phase Name | Description |
|---|---|
| ACTIVE | Normal operation phase. |
| OPERATIONALLY_COMPLETE | Terminal operational state reached. |
| ADMINISTRATIVELY_CLOSED | Administrative closure phase. |
| ARCHIVED | Read-only retention phase. |
| CRYPTOGRAPHICALLY_ERASED | Zone B content erased. |

### 13.3. SO Event Type Registry

Registry name: Sovereign Object Event Type Registry
Registration procedure: Specification Required.

Initial registrations:

| Event Type | Description |
|---|---|
| SO_CREATED | SO Instance created. |
| STATE_TRANSITIONED | State machine transition executed. |
| ZONE_B_ATTACHED | Zone B content attached. |
| ZONE_B_DETACHED | Zone B content detached. |
| MANDATE_BOUND | Mandate JWT bound to SO Instance. |
| MANDATE_REVOKED | Mandate JWT revoked. |
| HEM_ESCALATED | HEM escalation triggered. |
| HEM_RESOLVED | HEM escalation resolved. |
| PHASE_TRANSITIONED | Lifecycle phase transition. |
| ERASURE_INITIATED | Cryptographic erasure initiated. |
| ERASURE_COMPLETED | Cryptographic erasure completed. |
| GEC_TRANSITIONED | GEC association transferred. |

# 14. References

## 14.1. 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.

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

[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.

[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.

### 14.2. Informative References

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

[I-D.ietf-vcon-vcon-container]
Howe, T., et al., "The vCon Container", draft-ietf-vcon-vcon-container, work in progress.

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

[Howe-Layering]
Howe, T., "Eight-Layer Architecture for Agentic AI Systems", personal communication, May 2026.

## Appendix A. ATP Booking Object -- Reference Implementation

The ATP Booking Object is the reference implementation of the Sovereign Object primitive. It is the SO Type that the Activity Travel Protocol [ATP] defines for governing travel transactions across AI agents, operators, and travellers. It has been operational as a reference implementation at MyAuberge K.K. (Chino, Nagano, Japan) since early 2026.

### A.1. SO Type Registration

```
so_type_id:    "atp/booking-object/1.0"
registrant:    ATP Foundation (activity-travel-protocol.org)
specification: https://activitytravel.pro/layer3/
```

### A.2. State Machine

The ATP Booking Object has eleven canonical states:

| State | Description |
|---|---|
| INQUIRY | Booking intent declared; no commitment. |
| FEASIBILITY_CHECK | Async feasibility validation in progress. |
| AWAITING_CONFIRMATION | Feasibility passed; supplier confirmation pending. |
| CONFIRMED | All parties confirmed; booking live. |
| PRE_ACTIVITY | Pre-activity collection phase. |
| IN_JOURNEY | Traveller in experience; tracks sub-phases. |
| COMPLETED | Journey finished; duty-of-care met. |
| CANCELLED | Booking cancelled. |
| EXPIRED | Booking lapsed without confirmation. |
| BOOKING_SUSPENDED | Cross-cutting disruption state; overlays any other state. |
| DISPUTED | Dispute opened post-completion. |

The IN_JOURNEY state carries eight sub-phases: PRE_DEPARTURE, TRANSIT, ARRIVAL, ORIENTATION, ACTIVITY, POST_ACTIVITY, RETURN, POST_JOURNEY.

BOOKING_SUSPENDED is a cross-cutting state modifier that may overlay any non-terminal state. When active, all agent-initiated transitions

are prohibited until a human principal resolves the suspension.

### A.3. Zone A / Zone B Application

Zone A fields: so_id, so_type_id, current_state, booking_reference, operator_id, supplier_id, activity_id, journey_date, policy_version.

Zone B attachments: traveller identity documents, health declarations, consent records, insurance certificates, and all other personal data associated with the booking.

This separation enables the ATP Booking Object Event Stream to be retained for audit after Cryptographic Erasure of traveller personal data, satisfying GDPR Article 17 erasure obligations without destroying the governance audit record.

### A.4. SO Lifecycle Mapping

The ATP Booking Object's eleven operational states exist entirely within SO Lifecycle Phase 1 (ACTIVE). The SO Lifecycle Phase transitions of Section 6 govern the Booking Object's progression from active operation through archival and eventual Cryptographic Erasure of traveller personal data.

The two-axis model -- Booking Object state (operational) and SO Lifecycle Phase (governance) -- are orthogonal. A booking in IN_JOURNEY state is simultaneously in SO Phase 1 (ACTIVE). A booking in COMPLETED state transitions to SO Phase 2 (OPERATIONALLY_COMPLETE) after billing and duty-of-care obligations are satisfied.

## Author's Address

Tom Sato
MyAuberge K.K.
Chino, Nagano, Japan
Email: tomsato@myauberge.jp