A³O

API REFERENCE & INTEGRATION GUIDE

A³O API & Integration Guide R1.2

FULL MASTER — JAUS Integrated

Universal node, fleet, swarm and industry integration baseline

DOCUMENT ID	A3O-API-JAUS-1200
VERSION / STATUS	R1.2 FULL MASTER · CANDIDATE
LANGUAGE	ENGLISH
CLASSIFICATION	INTERNAL / PARTNER REVIEW
RELEASE DATE	17 JUL 2026
SOURCE GUIDE	R1.0 CANDIDATE · PRESERVED IN FULL
ARCHITECTURE BASELINE	A3O-ARCH-JAUS-1200 · R1.2 FULL MASTER

CONNECT ANY SYSTEM. GOVERN EVERY INTERFACE. PROVE EVERY EXCHANGE.

CONTROLLED DOCUMENT

Document Control and Release Governance

Field	Controlled value
Document ID	A3O-API-JAUS-1200
Document title	A³O API & Integration Guide R1.2 FULL MASTER — JAUS Integrated
Status	Candidate for architecture, partner and implementation review
Primary audience	Robot and sensor OEMs; system integrators; autonomy-stack developers; enterprise platforms; assessors
Normative baseline	A3O Architecture Baseline R1.2 FULL MASTER — JAUS Integrated
Strategic baseline	A3O Whitepaper R1.2 FULL MASTER — JAUS Integrated
Source guide	A3O API Reference & Integration Guide R1.0 Candidate — preserved in Annex H
Specification artifacts	OpenAPI 3.2 profile; AsyncAPI 3.1 profile; JSON Schema; JSIDL/JSD registry; SDKs and conformance assets
Claim status	JAUS-aligned implementation target; no SAE certification or endorsement is claimed

NORMATIVE BOUNDARY

The guide uses public scope information for SAE standards. Detailed JAUS wire formats, message definitions, JSD assets and formal compliance behavior shall be implemented from properly licensed standards and data sets.

Release objectives

- Replace the five-page outline with an implementation-grade guide while preserving every R1.0 section and content block.
- Make JAUS a native interoperability domain rather than a peripheral adapter, while keeping A³O protocol-neutral above the boundary.
- Define stable API objects, operations, events, version rules, authority checks, error behavior, offline semantics and conformance evidence.
- Align all interfaces with the eight A³O planes and the A3O-1000…9000 architecture baseline.
- Provide sufficient examples and requirements for independent implementation, review and test planning without reproducing licensed SAE content.

NAVIGATION

Controlled Structure

	Scope
	API Architecture, Governance and Common Contracts
	Domain API Specifications
I	Integration, Security, Conformance and Worked Examples
V	Annexes, Requirements and Preserved Source

PART I

API Architecture, Governance and Common Contracts

The contract foundation: scope, trust boundaries, standards, identity, identifiers, lifecycle, transports and native JAUS integration.

PART I · FOUNDATION

1. Executive API Contract Statement

The A³O API is the externally consumable contract of the Universal Trusted Framework for Autonomous Ecosystems. It does not expose a single monolithic command endpoint. It exposes a governed set of identity, data, state, mission, policy, authority, action, evidence, twin and conformance services. The contract is designed for heterogeneous robots, vehicles, sensors, industrial systems, enterprise platforms and partner ecosystems that may use JAUS, ROS 2, DDS, MQTT, OPC UA, proprietary protocols or direct SDK integration.

The central architectural rule is separation of observation, decision, authorization and execution. Telemetry may update candidate state; state may support a recommendation; a recommendation may lead to mission or action intent; but only an authorized, policy-compliant and safety-bounded path can produce an executable command. Every control-relevant transition is correlated to evidence.

PRIMARY CONTRACT

Connect any system. Normalize every capability. Govern every action. Preserve enough evidence to reconstruct what was known, who was authorized, what was commanded and what actually occurred.

1.1 API planes

Plane	External contract
Trust & Identity	Enrollment, credentials, attestation, ownership, mission membership, trust state and quarantine.
Data & Perception	Telemetry, sensor media references, quality, uncertainty, provenance and semantic mapping.
State & Knowledge	EntityState, ContextState, SharedWorldState, history, conflict markers and subscriptions.
Swarm & Coordination	Peer discovery, topology, capability exchange, task negotiation and partition handling.
Decision & Policy	Mission intent, policy evaluation, delegation, human approval and bounded autonomy.
Action & Control	ActionCommand, cancellation, progress, feedback, outcome assessment and safety interlocks.
Evidence & Assurance	EvidenceFragment, EvidencePackage, verification, replay, retention and selective disclosure.
Experience & Integration	OpenAPI, AsyncAPI, SDKs, adapters, profile manifests, test kits and partner tooling.

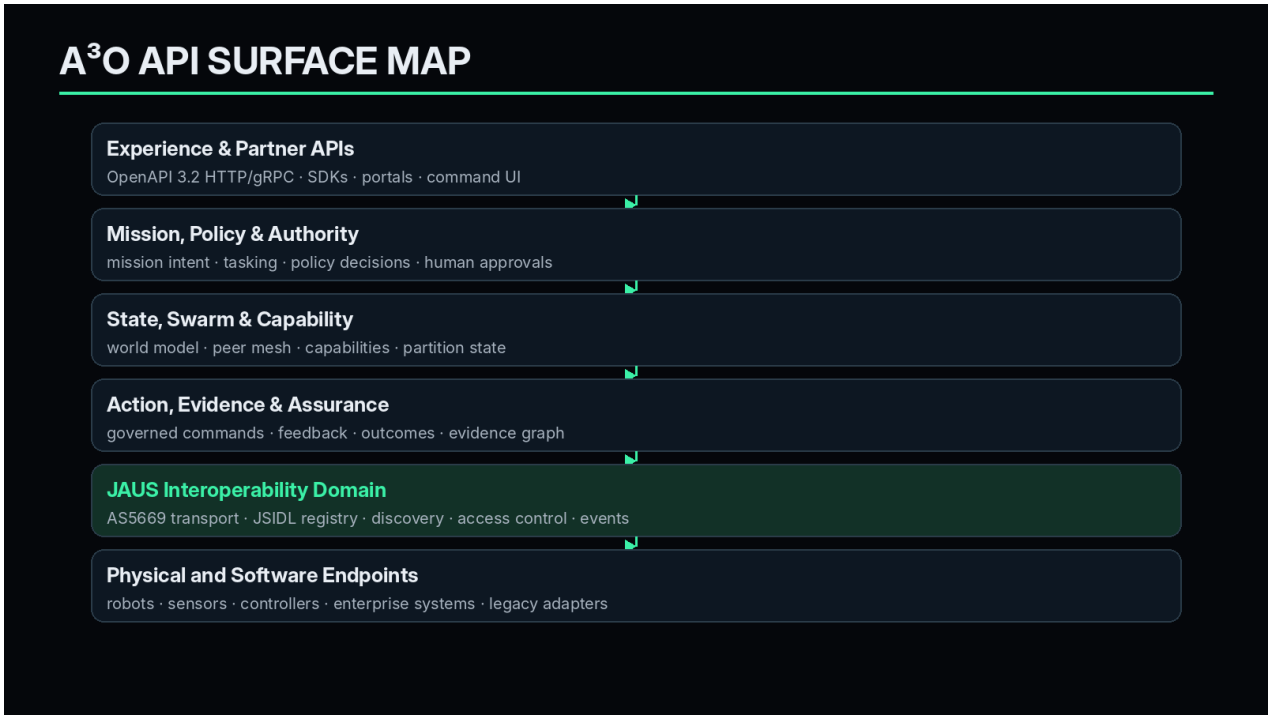

Figure 1 — A³O API surface and the native JAUS interoperability domain.

1.2 Architectural invariants

- No external protocol identity is treated as a global identity until it is bound to an A³O identity and trust context.
- No transport acknowledgement is treated as proof that the requested operational outcome occurred.
- No discovery or capability advertisement automatically grants trust, mission membership or command authority.
- No AI output can bypass policy, safety, human/delegated authority or local execution constraints.
- No partition or communications loss may expand authority; delegated permissions expire according to explicit rules.
- No semantic translation may silently change units, reference frames, coordinate conventions, safety meaning or control scope.
- Every authoritative object has a declared owner, version, lifecycle and evidence lineage.

Requirement ID	Normative requirement	Verification
API-FND-001	The implementation SHALL separate observation, decision, authorization and execution interfaces.	Architecture review and command-path test.
API-FND-002	Every control-relevant object SHALL carry ecosystem, mission, identity and correlation context where applicable.	Schema validation.
API-FND-003	The API SHALL support JAUS and non-JAUS endpoints under the same governance model.	Mixed-protocol integration scenario.
API-FND-004	The API SHALL distinguish transport success, execution progress and operational outcome.	Action lifecycle test.
API-FND-005	Protocol translation SHALL NOT expand source authority or permitted operating domain.	Negative authority test.

FOUNDATION

2. Scope, Roles and Supported Profiles

This guide is intended for robot and sensor OEMs, system integrators, autonomy-stack developers, industrial and enterprise platform teams, A³O industry-profile providers, ecosystem operators, federation partners and assessors. It defines the contract boundary between partner implementations and the A³O runtime. Executable schemas, service definitions, examples and test vectors take precedence over descriptive prose when a released artifact and the prose differ.

Integration role	Primary responsibility
Node Provider	Supplies a robot, vehicle, sensor, actuator, controller or software agent and declares its identity, capabilities and operational constraints.
Adapter Provider	Maps vendor protocol, state and commands into A³O contracts; owns transformation correctness and fault behavior.
JAUS Service Provider	Exposes JAUS components and services described by licensed JSD assets and an approved compatibility profile.
Profile Provider	Defines industry ontology, workflows, policies, conformance criteria and prohibited semantic overrides.
Ecosystem Operator	Owns tenant, site, fleet, identities, policies, mission authority and assurance obligations.
Federation Partner	Exchanges approved state, tasks or evidence across a separately governed trust boundary.
Assessor	Runs conformance, security, safety and performance suites and signs machine-readable results.

2.1 Supported integration profiles

Profile	Purpose
HTTP API profile	Synchronous command, query, configuration and administrative operations documented with OpenAPI 3.2.
Event API profile	Telemetry, state, mission, action and evidence events documented with AsyncAPI 3.1.
gRPC profile	Low-latency typed request/response and streaming inside controlled deployment zones.
Edge/P2P profile	Mutually authenticated peer exchange over constrained or intermittently connected links.

Profile	Purpose
AUS profile	Native AS5669 transport, JSIDL/JSD registry, Core Services integration and application service mappings.
ROS 2 / DDS profile	Gateway through declared message, service and QoS mappings without treating ROS graph identity as A³O trust identity.
ndustrial profile	OPC UA, field gateway or controller integration with process-state and safety-bound action semantics.
Federation profile	Cross-organization state and evidence exchange with explicit disclosure and retention policy.

2.2 Out of scope

- Reproduction of licensed SAE JAUS message definitions or data sets.
- Certification of a medical device, industrial machine, vehicle, aircraft, maritime system or weapon system.
- Replacement of local certified safety controllers, emergency stops or platform-specific safety kernels.
- A guarantee that all vendor extensions are semantically interoperable merely because they can be transported.
- Direct exposure of internal database models as public API contracts.

Requirement ID	Normative requirement	Verification
API-SCP-001	Every integration SHALL declare one or more supported profiles and an explicit conformance target.	Profile manifest review.
API-SCP-002	An adapter SHALL declare the source protocol, semantic mapping version and unsupported source features.	Adapter manifest validation.
API-SCP-003	Industry profiles SHALL identify external certification and compliance obligations that remain outside A³O interoperability conformance.	Profile governance review.

FOUNDATION

3. API Architecture and Trust Boundaries

The public API surface is separated from the internal canonical event fabric and from the execution boundary. Public endpoints terminate in an API gateway or edge runtime. They do not address actuators directly. The gateway validates identity, tenant and ecosystem scope, schema, replay protection, rate policy and requested authority before forwarding a typed intent to the relevant authoritative service.

A³O follows one-authoritative-service-per-object ownership. The identity service owns identity state; the mission service owns mission state; the policy service owns policy decisions; the action service owns action lifecycle; and the evidence service owns evidence lineage. Read models and digital-twin projections may replicate these objects but cannot modify them.

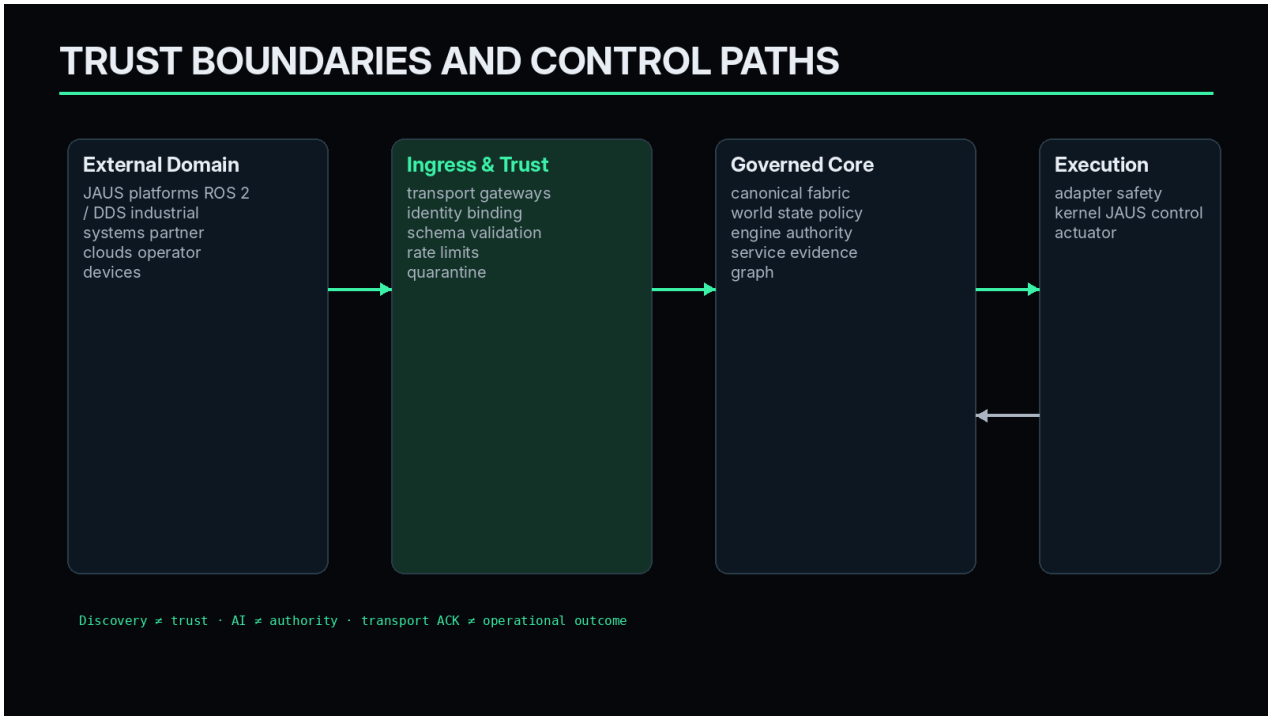

Figure 2 — Trust boundaries and the governed control path.

3.1 Boundary catalogue

Boundary	Input assumption	Required control
External device → ingress gateway	Untrusted transport and source data	Authenticate, validate, rate-limit, bind identity and quarantine on uncertainty.
JAUS network → JAUS domain	Protocol-valid messages with unverified organizational trust	Resolve address, service and version; bind to A³O identity; preserve raw payload.
Gateway → canonical fabric	Validated but not necessarily authoritative objects	Apply semantic mapping, ownership rules and provenance.
State → decision service	State may be stale, uncertain or conflicting	Carry age, uncertainty, source and conflict markers.
Decision → authority service	Recommendation or intent	Require policy, scope, ODD, TTL and human/delegated authority.
Authority → execution gateway	Approved bounded action	Revalidate local safety and protocol ownership before encoding.
Evidence → federation/export	Potentially sensitive records	Apply classification, disclosure, retention and export controls.

3.2 Authoritative service rule

Every object type shall have one system-of-record service and may have multiple read projections. Commands addressed to read projections shall be rejected. Event consumers shall not infer authority from the ability to receive an event. The object owner emits monotonic revisions or vector metadata sufficient for consumers to detect staleness and conflict.

3.3 API gateway responsibilities

- Mutual TLS or equivalent authenticated channel according to the profile.
- Token, certificate and workload-identity validation.
- Tenant, ecosystem, mission and resource-scope enforcement.
- Schema, size, rate, anti-replay and idempotency validation.
- Correlation and trace-context injection.
- Classification and privacy label enforcement.
- Protocol-specific error translation without loss of the canonical reason code.

Requirement ID	Normative requirement	Verification

Requirement ID	Normative requirement	Verification
API-ARC-001	Public APIs SHALL terminate at a governed ingress boundary and SHALL NOT directly address physical actuators.	Architecture and penetration test.
API-ARC-002	Every authoritative object SHALL declare a single owning service and a revision or conflict-detection mechanism.	Schema and ownership registry review.
API-ARC-003	Read projections SHALL reject state-changing operations.	Negative endpoint test.
API-ARC-004	Event delivery SHALL NOT be interpreted as command or authority delegation.	Policy test.
API-ARC-005	The gateway SHALL preserve canonical reason codes when translating errors to transport-specific responses.	Error mapping test.

FOUNDATION

4. Standards, Description Languages and Artifact Governance

A³O uses multiple interface-description languages because no single language fully describes synchronous APIs, event channels, autonomous-system services and domain semantics. OpenAPI describes HTTP interfaces; AsyncAPI describes message-driven channels; JSON Schema defines canonical payloads; Protocol Buffers may define gRPC contracts; JSIDL/JSD defines JAUS services; industry profiles add ontology, workflow and conformance manifests. These artifacts are governed as one release graph.

4.1 Baseline specifications

Specification	Use in A³O	Governance note
OpenAPI 3.2.0	HTTP API description and partner documentation	Published 19 Sep 2025; use the exact declared patch version in generated artifacts.
AsyncAPI 3.1.0	Event-driven API and protocol-binding description	Use channels, operations, messages and x-a3o-* extensions.
JSON Schema	Canonical object validation	A profile shall declare the dialect and vocabulary; schema ID is immutable.
Protocol Buffers / gRPC	Typed internal or edge RPC	Field numbers are never reused; unknown fields are preserved where supported.
SAE AS5684B JSIDL	Formal definition of JAUS services	Licensed JSD assets are imported into the controlled JAUS registry.
SAE AS5710A	JAUS Core Service Set	Transport abstraction, events, access control, management, time, liveness, discovery and list management.
SAE AS5669A	JAUS/SDP transport	Used by the native transport gateway where applicable.
A³O Profile Manifest	Composition and conformance contract	Binds protocol artifacts, semantics, policies, tests and supported maturity.

4.2 Artifact graph

4.3 Extension rules

- OpenAPI and AsyncAPI extensions use the x-a3o- prefix and shall be documented in the profile schema.
- Vendor payload extensions are confined to an extensions object unless the profile explicitly promotes them to canonical semantics.

- A³O JAUS extension services use a distinct A³O namespace and shall not be represented as SAE-standard services.
- An extension may add information but shall not weaken policy, safety, evidence or identity requirements.
- Profiles shall provide golden examples and negative examples for every extension with operational impact.

Requirement ID	Normative requirement	Verification
API-ART-001	Every published endpoint or event SHALL trace to a versioned machine-readable contract.	Artifact coverage report.
API-ART-002	Profile manifests SHALL bind schemas, examples, policies and tests by immutable digest.	Manifest verification.
API-ART-003	Vendor extensions SHALL NOT redefine canonical field semantics.	Schema and semantic review.
API-ART-004	Licensed JAUS artifacts SHALL be stored and distributed according to their applicable license terms.	Repository access audit.
API-ART-005	A³O-specific JAUS services SHALL use a distinct namespace and explicit extension status.	JSIDL registry validation.

COMMON CONTRACT

5. Identity, Enrollment and Trust Bootstrap

A³O identifies organizations, users, workloads, nodes, components and physical assets separately. A robot may contain several workloads and protocol components; a JAUS address identifies a protocol endpoint but does not replace the global A³O node identity. Enrollment binds these identities, credentials, ownership, ecosystem membership, allowed profiles and trust state into a controlled record.

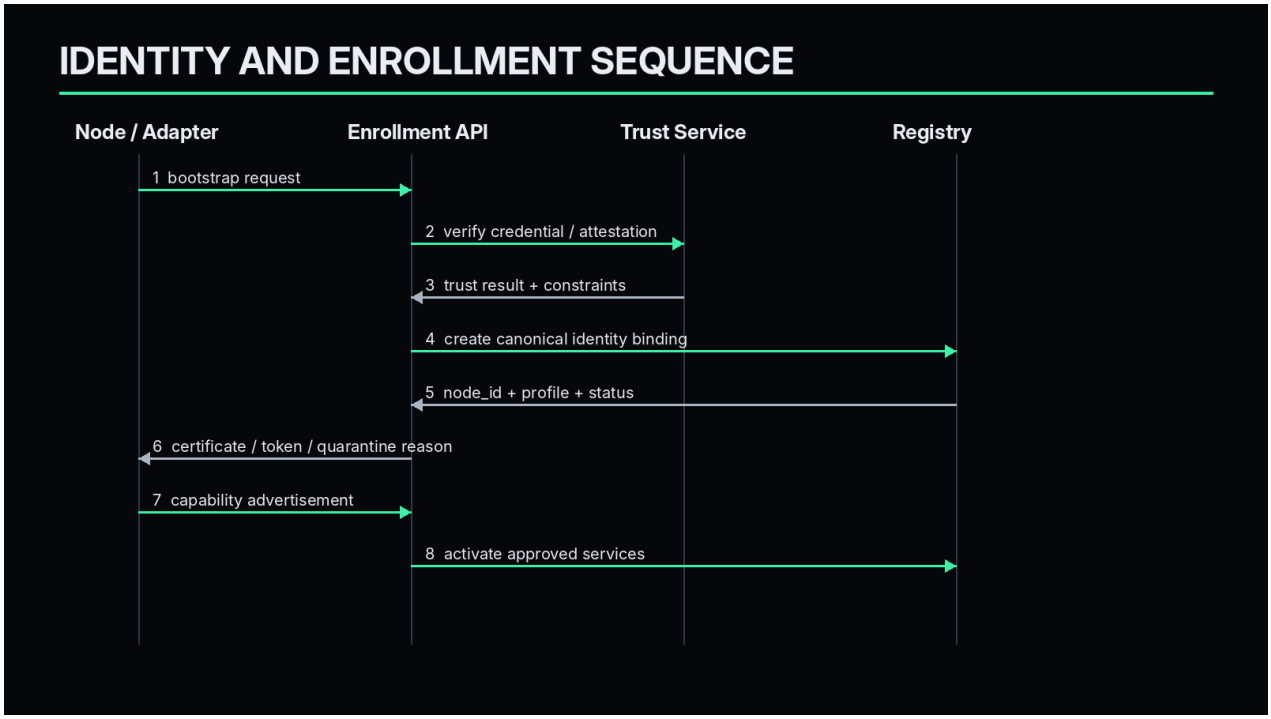

Figure 3 — Enrollment and identity-binding sequence.

5.1 Identity types

Identity	Meaning	Lifecycle
organization_id	Legal or operational entity responsible for assets and users.	Long-lived; governance controlled.
tenant_id	Commercial or administrative isolation boundary.	Long-lived; may contain multiple ecosystems.
ecosystem_id	Operational environment, site, fleet or federated domain.	Lifecycle controlled by operator.
node_id	Robot, vehicle, sensor, controller, edge server or software agent.	Stable across credential rotation.

Identity	Meaning	Lifecycle
component_id	Addressable software or hardware capability within a node.	May bind to JAUS component, ROS node or vendor module.
workload_id	Running software identity, build and attestation context.	Ephemeral; restart creates new instance identity.
user_id	Human principal.	Mapped to roles, qualifications and approvals.
asset_id	Physical asset, serial, vehicle or equipment identity.	Separated from replaceable compute identity.

5.2 Enrollment operations

Method / Channel	Resource or operation	Semantics	Authority
POST	/v1/enrollments	Create a pending enrollment request with bootstrap credential and claimed identity.	Enrollment authority
POST	/v1/enrollments/{id}:attest	Submit attestation, software inventory and hardware evidence.	Attestation service
POST	/v1/enrollments/{id}:approve	Approve, constrain or quarantine the candidate.	Authorized operator / policy
POST	/v1/credentials:rotate	Rotate certificate or token without changing canonical node identity.	Credential owner
POST	/v1/identities/{id}:revoke	Revoke identity or credential and propagate quarantine.	Security authority
EVENT	a3o.identity.status.changed	Publish state transitions including VERIFIED, DEGRADED, QUARANTINED and REVOKED.	Subscribers only

5.3 Enrollment object

ENROLLMENTREQUEST EXAMPLE

```
{
  "enrollment_id": "enr_01J...",
  "organization_id": "org:partner-a",
  "ecosystem_id": "ecosystem:hub-01",
  "asset_id": "asset:amr-042",
  "claimed_node_id": "node:hub:amr-042",
  "protocol_identities": [
    {"family":"JAUS","subsystem":42,"node":3,"component":17}
  ],
  "requested_profiles": ["A3O-JAUS-CORE","A3O-LOGISTICS-AMR"],
  "attestation": {"status":"PENDING","sbom_digest":"sha256:..."},
  "trust_state": "CANDIDATE",
  "requested_at": "2026-07-17T10:00:00Z"
}
```

5.4 Trust-state lifecycle

e	Meaning
DIDATE	Identity claim received; no operational trust.
IFIED	Required identity and attestation checks passed.
RADED	Identity remains known but one or more trust conditions failed or expired.
ARANTINED	Communication restricted to diagnostics, remediation and evidence export.
OKED	Credential or identity is no longer valid; control operations are denied.
IRED	Asset intentionally removed; historical evidence remains addressable.

Requirement ID	Normative requirement	Verification
API-ID-001	Protocol addresses SHALL be bound to a canonical A^{3}O identity before authoritative state or command use.	Enrollment and negative unbound-address test.

Requirement ID	Normative requirement	Verification
API-ID-002	Credential rotation SHALL NOT change the canonical node identity.	Rotation test.
API-ID-003	Revocation SHALL propagate to active sessions, authority leases and command paths within the declared security budget.	Revocation latency test.
API-ID-004	A quarantined node SHALL be limited to explicitly allowed remediation and evidence operations.	Quarantine policy test.
API-ID-005	Enrollment records SHALL retain the decision, evidence, approver and policy version.	Evidence audit.

COMMON CONTRACT

6. Canonical Identifiers, Naming and Versioning

Canonical identifiers are opaque, stable and globally unambiguous within their declared namespace. Human-readable names are attributes and may change. A protocol address, network address, serial number or vendor name may be indexed as an alias but shall not be used as the sole durable key for cross-ecosystem references.

6.1 Identifier catalogue

Field	Example	Use
ecosystem_id	ecosystem:bratislava-hub-01	Operational domain.
node_id	node:hub:amr-042	Autonomous or software node.
component_id	component:amr-042:mobility	Addressable capability endpoint.
entity_id	entity:track:7a19	Operational entity in world state.
mission_id	mission:cargo-transfer-1187	Mission lifecycle root.
task_id	task:mission-1187:deliver-03	Task within a mission.
action_id	action:01J...	Governed execution attempt.
policy_id	policy:logistics:route-safety:2.1.0	Policy definition.
decision_id	decision:01J...	Specific evaluated decision.
authority_lease_id	lease:01J...	Time-bounded delegated authority.
evidence_id	evidence:01J...	Evidence fragment or package.
schema_id	urn:a3o:schema:telemetry-envelope:1.2.0	Immutable schema identity.
profile_id	urn:a3o:profile:logistics-amr:1.1.0	Industry/profile release.

6.2 Version model

- API version identifies the public operation contract and is represented in the path, media type or explicit negotiated header according to the profile.
- Schema version identifies the payload structure and semantics; an immutable schema ID never points to different content.
- Profile version binds schemas, ontology, workflows, policies, examples and tests.
- Adapter version identifies implementation behavior and participates in evidence lineage.
- JAUS service version is resolved from the licensed JSD and recorded independently from the A^{3}O mapping profile version.

6.3 Compatibility rules

Change class	Meaning	Required behavior
PATCH	Bug fix or clarification without semantic change	Automatic within declared support range.
MINOR	Backward-compatible fields, operations or event additions	Consumer shall ignore unknown optional fields.
MAJOR	Breaking structure or semantic change	Explicit migration and dual-run period required.

Change class	Meaning	Required behavior
PROFILE REVISION	Composition, policy, test or mapping change	Approval according to operational impact.
JAUS MAPPING REVISION	Change to JSD binding or semantic translation	Round-trip and scenario requalification required.

Requirement ID	Normative requirement	Verification
API-IDV-001	Canonical identifiers SHALL be opaque and SHALL NOT encode mutable authority or network location.	Identifier lint.
API-IDV-002	Schema IDs SHALL be immutable and content-addressable or protected by a published digest.	Registry test.
API-IDV-003	A breaking semantic change SHALL increment the major version even when the JSON shape is unchanged.	Semantic review.
API-IDV-004	JAUS service version and A³O mapping profile version SHALL be recorded separately.	Evidence inspection.
API-IDV-005	Aliases SHALL resolve to one canonical identifier or return an explicit ambiguity error.	Alias resolution test.

COMMON CONTRACT

7. API Lifecycle, Compatibility and Deprecation

API lifecycle governance is a safety and operational-control function, not merely a developer-experience concern. A deprecated field can remain necessary for an installed robot; a new optional event can create load; and a changed interpretation can invalidate evidence. Every API and profile therefore passes through proposal, experimental, candidate, supported, deprecated, retired and archived states.

ecycle state	Meaning
OPOSED	Design record exists; no implementation commitment.
PERIMENTAL	May change without compatibility; isolated pilots only.
NDIDATE	Contract frozen for partner review and conformance development.
PPORTED	Published compatibility range, SLO and maintenance policy.
PRECATED	Still supported for a declared window; migration path published.
TIRED	No new use; requests rejected after the effective date.
CHIVED	Read-only artifacts and evidence retained for historical reconstruction.

7.1 Deprecation package

- Deprecation announcement with affected profiles and consumers.
- Replacement operation, object or mapping and a machine-readable migration guide.
- Minimum support window based on safety, certification and installed-base constraints.
- Dual-publish or translation behavior where operationally safe.
- Conformance tests for both old and replacement contracts during migration.
- Evidence indicating which contract version produced each action or state update.

7.2 Consumer capability negotiation

CAPABILITY NEGOTIATION EXAMPLE

```
GET /v1/capabilities/api
Accept: application/vnd.a3o.capabilities+json

{
  "api_versions": ["1.1", "1.2"],
  "schema_ranges": {"TelemetryEnvelope": ">=1.0 <2.0"},
  "profiles": ["A3O-JAUS-CORE:1.0", "A3O-LOGISTICS-AMR:1.1"],
  "features": {"event_resume": true, "batch_actions": false},
  "deprecated_after": {"api:1.1": "2027-07-17"}
}
```

Requirement ID	Normative requirement	Verification
API-LCM-001	Every released contract SHALL have a lifecycle state, owner and support policy.	Registry audit.
API-LCM-002	Deprecation SHALL include a machine-readable replacement and effective date.	Artifact validation.
API-LCM-003	Consumers SHALL be able to discover supported versions and profile ranges before mission execution.	Negotiation test.
API-LCM-004	A retired control operation SHALL fail closed with a canonical migration reason.	Retirement test.
API-LCM-005	Evidence SHALL preserve the exact API, schema, profile, adapter and mapping versions used.	Evidence audit.

COMMON CONTRACT

8. Transport Profiles, QoS and Delivery Semantics

A³O separates canonical semantics from transport. The same TelemetryEnvelope or ActionCommand may be carried over HTTPS, gRPC, MQTT, AMQP, NATS-like event infrastructure, DDS, JAUS/SDP or a constrained P2P link. A transport profile defines framing, authentication, ordering, delivery class, timeout, buffering, backpressure and recovery. It does not redefine the object semantics.

Transport profile	Typical use	Mandatory declaration
HTTPS/JSON	Administrative operations, queries, low-rate command intent	Request/response; explicit idempotency; bounded body size.
gRPC	Typed edge or service-to-service operations and streaming	Deadline required; status mapped to canonical errors.
MQTT	Constrained telemetry and events	Topic template, QoS class, retained-message prohibition for selected control topics.
AMQP / event broker	Durable enterprise event flows	Consumer groups, dead-letter policy, ordering key and replay cursor.
DDS / ROS 2 gateway	Real-time or robotics data exchange	Declared QoS mapping; graph discovery not equal to trust.
JAUS/SDP	Native JAUS component communications	AS5669 transport and licensed JSD message semantics.
Secure P2P	Swarm and partition-tolerant peer exchange	Mutual identity, mission context, bounded buffer and reconciliation.

8.1 Delivery classes

Class	Semantics	Examples
D0 BEST_EFFORT	Loss acceptable; no retry	High-rate non-critical samples.
D1 AT_LEAST_ONCE	Retry allowed; consumer deduplicates	Telemetry, evidence fragments, durable events.
D2 EFFECTIVELY_ONCE	Idempotency and deduplication required	Mission/task changes and action intent.
D3 ORDERED_STREAM	Ordering key and gap detection required	State deltas, action progress and evidence sequence.

Class	Semantics	Examples
D4 CONTROL_CRITICAL	Bounded latency, explicit acknowledgement and safe timeout	Authority, action and lifecycle transitions.

8.2 Required metadata

- message_id and correlation_id;
- source identity and destination or routing scope;
- creation, reception and expiry timestamps;
- delivery class, priority and ordering key;
- schema/profile version and content type;
- classification/privacy labels;
- retry count, replay indicator and original message identifier where applicable.

Requirement ID	Normative requirement	Verification
API-TRN-001	Transport profiles SHALL NOT redefine canonical object semantics.	Cross-transport golden-vector test.
API-TRN-002	Control-relevant requests SHALL carry an idempotency key or protocol-equivalent duplicate-control mechanism.	Duplicate request test.
API-TRN-003	Every stream SHALL define ordering, gap detection, replay and backpressure behavior.	Stream resilience test.
API-TRN-004	A message received after expiry SHALL NOT create a new authoritative state transition or action.	Expiry test.
API-TRN-005	Transport-specific status codes SHALL map to canonical A³O reason codes.	Error translation test.

NATIVE INTEROPERABILITY

9. JAUS Interoperability API Domain

The JAUS Interoperability Domain is a native runtime boundary that integrates JAUS Subsystems, Nodes, Components and Services into the A³O trust and orchestration model. A³O may operate as a JAUS client/controller, a provider of selected A³O extension services, or a protocol bridge. It preserves JAUS service semantics, address context, Access Control ownership and lifecycle behavior while enriching each exchange with A³O identity, policy, mission and evidence context.

STANDARDS CLARIFICATION

AS-4JAUS is the SAE committee designation. AS5684B defines JSIDL; AS5710A defines the JAUS Core Service Set; AS5669A defines the JAUS/SDP transport specification. Detailed implementation shall use properly licensed standards and data sets.

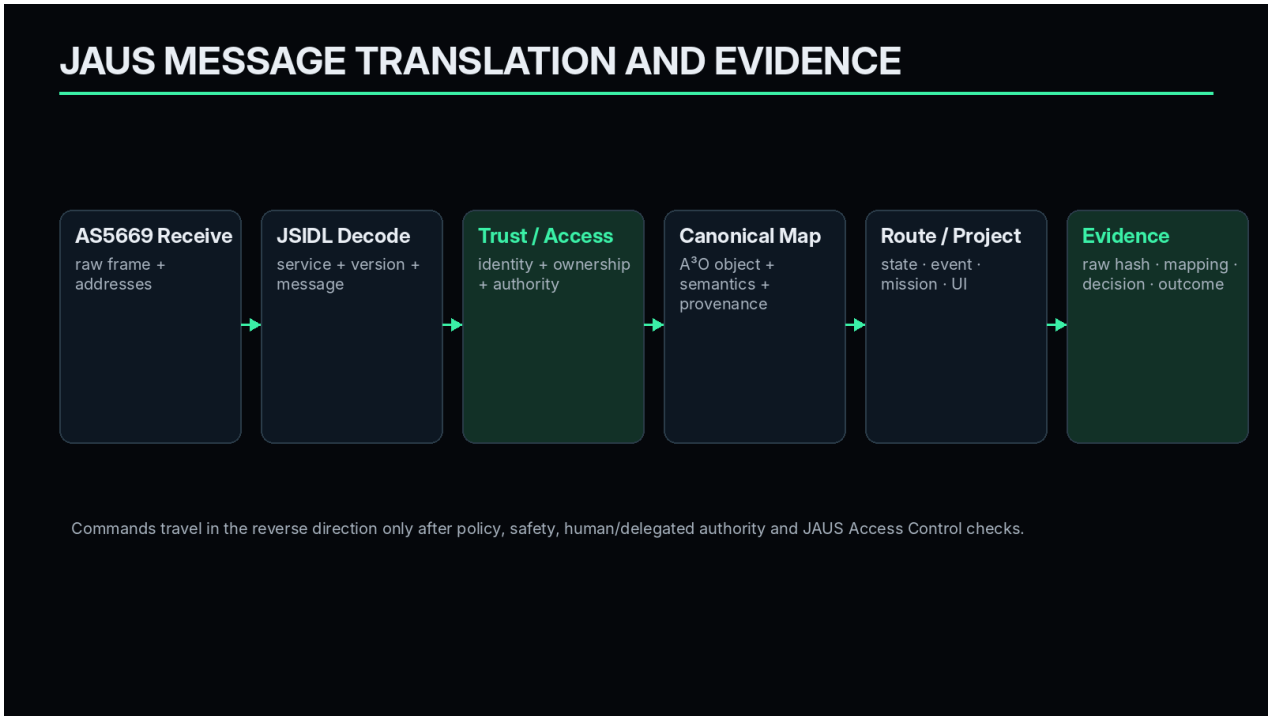

Figure 4 — JAUS translation, trust enrichment and evidence correlation.

9.1 JAUS entity mapping

JAUS concept	A³O representation	Interpretation
JAUS Subsystem	Managed Autonomous Platform	Robot, vehicle, machine, production cell or platform boundary.
JAUS Node	Edge Compute Node	Onboard computer, controller or compute module.
JAUS Component	Capability Endpoint	Addressable software entity with services.
JAUS Service	Capability Contract	Formally described interface and message protocol.
JSD / JSIDL	Service Schema Artifact	Controlled service definition, version and dependencies.
JAUS Address	External Protocol Identity	Subsystem/node/component route bound to A³O identity.
Command	Governed Action Request	State-changing request after A³O and JAUS authority checks.
Query	Information Request	One-time information request.
Report	Observation or State Update	Response or published state.
Event	Live State Subscription	Asynchronous update according to Events service.

9.2 Native JAUS runtime services

Runtime component	Responsibility
A3O.JAUS.TransportGateway	AS5669 framing, routing, connection diagnostics, segmentation and raw-payload retention.
A3O.JAUS.JSIDLRegistry	Imports licensed JSD assets; resolves versions and dependencies; generates codecs and compatibility metadata.
A3O.JAUS.DiscoveryBridge	Builds subsystem/node/component hierarchy and registers candidate capabilities.
A3O.JAUS.AccessControlBroker	Acquires, releases and monitors preemptable exclusive control; maps ownership to A³O authority context.
A3O.JAUS.ManagementBridge	Maps component lifecycle and readiness to A³O health and mission availability.
A3O.JAUS.EventBridge	Creates and cancels subscriptions; maps event rate, change condition and delivery state.
A3O.JAUS.TimeLivenessBridge	Maintains time quality and liveness evidence without treating liveness as trust.

Runtime component	Responsibility
A3O.JAUS.MessageMapper	Transforms JSD-defined messages into canonical A³O objects using a versioned semantic mapping.

9.3 Canonical JAUS envelope

INTERNAL A³O REPRESENTATION — NOT THE JAUS WIRE FORMAT

```
{
  "envelope_version": "a3o.jaus.v1",
  "message_id": "018f2aa0-42c7-7c63-a10e-82b104b5f120",
  "protocol": {"family":"JAUS","transport_profile":"SAE-AS5669A"},
  "source": {
    "jaus_address":{"subsystem":42,"node":3,"component":17},
    "a3o_node_id":"node:hub:amr-042",
    "trust_status":"VERIFIED"
  },
  "service": {
    "jsidl_reference":"licensed-registry-ref",
    "service_version":"declared-by-provider",
    "message_name":"JSD_DEFINED_MESSAGE",
    "interaction":"REPORT"
  },
  "context": {"ecosystem_id":"ecosystem:hub-01","mission_id":"mission:1187"},
  "payload": {"raw_message_retained":true,"decoded_fields":{},"canonical_state_delta":{}},
  "evidence": {"raw_hash":"sha256:...","mapping_profile":"A3O-JAUS-MOBILITY-1.0"}
}
```

9.4 Interaction mapping

JAUS pattern	A³O object	Rule
Command	GovernedActionRequest	Requires policy, safety, authority and JAUS Access Control ownership before encoding.
Query	InformationRequest	One-shot request with source/destination correlation, timeout and evidence.
Report	ObservationEvent / EntityStateDelta	Validated data updates state or a non-authoritative observation projection.
Event subscription	LiveStateSubscription	Subscription lifecycle, rate, change condition, loss and cancellation are recorded.
Discovery	CapabilityRegistration	Creates a candidate capability; trust and mission admission are separate.
Management	LifecycleTransition	Maps readiness, standby, emergency and shutdown behavior to operational availability.
Access Control	AuthorityLeaseBinding	Correlates JAUS ownership to A³O policy and delegated authority.

9.5 Command path and ownership

GOVERNED JAUS COMMAND PATH

A³O Action Intent
→ PolicyDecision
→ SafetyEnvelope
→ Human / Delegated Authority
→ JAUS Access Control ownership
→ JSD-defined command encoding
→ JAUS component execution / rejection
→ report or event correlation
→ OutcomeAssessment
→ EvidencePackage

9.6 A³O JAUS extension policy

A³O-specific capabilities that are not covered by an applicable SAE service shall use separate namespaces and explicit extension status. Candidate examples include track fusion, trust status, evidence correlation and swarm tasking. Extension services shall not reuse standard message names in a way that implies SAE standardization.

```
CANDIDATE EXTENSION NAMESPACES
urn:a3o:jaus:service:track-fusion:1.0
urn:a3o:jaus:service:trust-status:1.0
urn:a3o:jaus:service:evidence-correlation:1.0
urn:a3o:jaus:service:swarm-tasking:1.0
```

Requirement ID	Normative requirement	Verification
API-JAUS-001	The gateway SHALL preserve original JAUS source and destination addresses and the unmodified received payload.	Raw-message evidence test.
API-JAUS-002	Every decoded message SHALL reference the exact JSD/service definition and version used.	Registry and evidence test.
API-JAUS-003	Discovery SHALL create candidate capability records and SHALL NOT grant trust or control authority.	Discovery admission test.
API-JAUS-004	A³O SHALL respect JAUS Access Control ownership and preemption behavior.	Contention and preemption scenario.
API-JAUS-005	A³O SHALL fail closed when service version or semantic mapping is unknown or incompatible.	Unknown-version negative test.
API-JAUS-006	A³O extensions SHALL use separate namespaces and SHALL NOT be represented as SAE-standard services.	Registry lint.
API-JAUS-007	Mission-level authority SHALL NOT bypass local safety or JAUS control ownership.	End-to-end command-path test.
API-JAUS-008	Query, report, event and command correlations SHALL be retained in the Evidence Graph.	Evidence completeness test.

PART II

Domain API Specifications

Implementation-grade contracts for telemetry, state, mesh, capabilities, missions, authority, actions, evidence, twins, observability, offline operation and profiles.

PART II · DOMAIN APIS

10. Universal Telemetry API

The Universal Telemetry API accepts observations from physical sensors, controllers, software workloads and protocol gateways without forcing consumers to understand vendor-specific payload structures. The envelope separates source, semantic type, unit and reference frame, timing, quality, uncertainty, classification, provenance and payload. High-rate media may be referenced rather than embedded; control-critical data uses stricter validation and expiry rules.

10.1 TelemetryEnvelope

Object	Purpose	Required fields	Lifecycle / notes
TelemetryEnvelope	Common observation container.	message_id, source_node_id, ecosystem_id, timestamp, semantic_type, schema_version, payload	Immutable after acceptance; corrections create a new message.
QualityContext	Validity and sensor/data quality.	validity, confidence, completeness, freshness, diagnostics	May cause quarantine or non-authoritative routing.
UncertaintyContext	Quantified or categorical uncertainty.	model, covariance/range, confidence, method	Mandatory for profiles where uncertainty affects decisions.
ProvenanceContext	Origin and transformation lineage.	source, adapter, mapping, calibration, software build, hashes	Evidence-linked and append-only.
MediaReference	Reference to image, video, audio or large binary.	uri, digest, media_type, capture time, access policy	Content-addressed where possible.
BatchEnvelope	Bounded batch of telemetry items.	batch_id, items, ordering, compression, partial-failure policy	No mixed classification unless profile allows.

TELEMETRYENVELOPE EXAMPLE


```
{
  "message_id": "msg_01J...",
  "source_node_id": "node:factory:spindle-07",
  "source_component_id": "component:spindle-07:vibration",
  "ecosystem_id": "ecosystem:plant-01",
  "mission_id": "mission:production-shift-b",
  "timestamp": "2026-07-17T10:15:31.184Z",
  "trusted_time_status": "SYNCHRONIZED",
  "data_modality": "VIBRATION",
  "semantic_type": "machine.spindle.vibration.rms",
  "unit_system": "SI",
  "reference_frame": "asset-local",
  "quality": {"valid":true,"confidence":0.98,"freshness_ms":15},
  "uncertainty": {"type":"standard_deviation","value":0.02},
  "classification": "INTERNAL",
  "schema_version": "1.2.0",
  "signature": "...",
  "provenance": {"adapter_version":"2.3.1","calibration_id":"cal-921"},
  "payload": {"value":2.18,"unit":"mm/s"}
}
```


10.2 Operations and channels

Method / Channel	Resource or operation	Semantics	Authority
POST	/v1/telemetry	Submit one observation with synchronous acceptance result.	Node or adapter identity
POST	/v1/telemetry:batch	Submit a bounded batch with item-level status.	Node or adapter identity
STREAM	/v1/telemetry:stream	gRPC or profile stream with flow-control and deadlines.	Declared stream publisher
CHANNEL	a3o.telemetry.{ecosystem}.{semantic_type}	Event ingestion or distribution according to profile.	Publisher and subscriber scopes
GET	/v1/telemetry/{message_id}	Retrieve accepted envelope and provenance if permitted.	Evidence/read scope
POST	/v1/telemetry/{id}:invalidate	Mark an accepted observation invalid without deleting evidence.	Data authority

10.3 Modality registry

Canonical modalities: RF · RADAR · IMAGE · VIDEO · THERMAL · LIDAR · SONAR · POSITION · VELOCITY · ACCELERATION · VIBRATION · TEMPERATURE · PRESSURE · ENERGY · BIOMETRIC · PROCESS_STATE · MATERIAL_STATE · NAVIGATION · MANUAL_INPUT · EXTERNAL_SYSTEM

10.4 Validation and routing

- Validate identity, schema, semantic type, unit and reference frame before authoritative use.
- Retain raw source representation when translation occurred and bind it to the canonical message.
- Route invalid but potentially useful observations to diagnostics or quarantine, never silently discard them.
- Apply freshness and expiry rules by semantic type; a stale position may remain evidence but shall not guide a live action.
- Separate privacy-sensitive fields from generally distributable operational state through field-level or object-level labels.
- Use store-and-forward with original creation time and replay indicator during reconnect.

Requirement ID	Normative requirement	Verification
API-TEL-001	Telemetry SHALL carry semantic type, timestamp, source identity, schema version and provenance.	Schema validation.
API-TEL-002	Unit and reference-frame conversions SHALL be explicit, versioned and auditable.	Golden-vector conversion test.
API-TEL-003	Stale or expired telemetry SHALL NOT update authoritative live state unless the profile explicitly permits late correction.	Freshness test.
API-TEL-004	Batch ingestion SHALL report item-level acceptance and SHALL NOT hide partial failure.	Batch partial-failure test.
API-TEL-005	Raw source data SHALL be retained or content-addressed when required by the profile evidence policy.	Evidence retention test.
API-TEL-006	Unknown semantic types SHALL be isolated from policy-critical processing.	Unknown-type negative test.

DOMAIN API

11. State and World Model API

The State and World Model API provides versioned, queryable and subscribable representations of operational entities. It distinguishes an observation from a state estimate, a local context from a shared world model and a live projection from historical replay. State objects carry age, uncertainty, revision, ownership and conflict metadata so that consumers can reason about trust and freshness.

11.1 State objects

Object	Purpose	Required fields	Lifecycle / notes
EntityState	Current estimated state of an entity.	entity_id, entity_type, pose/status fields, revision, observed_at, valid_until	Owned by a declared state service.
ContextState	Operational context such as weather, traffic, process or site condition.	context_id, domain, values, scope, revision	May aggregate multiple sources.
SharedWorldState	Mission-scoped set of entities and context.	world_id, mission_id, members, revision_vector	Distributed and conflict-aware.
StateDelta	Change from a known revision.	object_id, base_revision, new_revision, changed fields	Rejected or branched when base revision mismatches.
StateSnapshot	Consistent bounded view.	snapshot_id, cutoff_time, object revisions	Used for planning, replay and evidence.
ConflictRecord	Unresolved competing claims.	object_id, branches, sources, policy, status	Never silently overwritten.

11.2 Operations and subscriptions

Method / Channel	Resource or operation	Semantics	Authority
GET	/v1/entities/{entity_id}/state	Read latest authorized state and age.	Read scope
GET	/v1/worlds/{world_id}:snapshot	Create a consistent snapshot at a cutoff.	Mission read scope

Method / Channel	Resource or operation	Semantics	Authority
GET	/v1/state:history	Query time range, source, semantic type and revision.	Historical/evidence scope
POST	/v1/state:applyDelta	Apply a validated delta from an authoritative or federated source.	State writer
CHANNEL	a3o.state.entity.changed	Publish state revisions with resume cursor.	Authorized subscribers
CHANNEL	a3o.state.conflict.created	Publish unresolved conflict requiring policy or human review.	Operations and assurance

11.3 EntityState example

ENTITYSTATE EXAMPLE

```
{
  "entity_id": "entity:vehicle:amr-042",
  "entity_type": "autonomous_mobile_robot",
  "ecosystem_id": "ecosystem:hub-01",
  "mission_id": "mission:1187",
  "revision": "rev:009183",
  "observed_at": "2026-07-17T10:22:11.020Z",
  "valid_until": "2026-07-17T10:22:12.020Z",
  "pose": {"frame":"map:hub-01","x":182.4,"y":71.9,"yaw":1.42},
  "velocity": {"vx":1.8,"vy":0.0,"wz":0.04},
  "operational_state": "EXECUTING_TASK",
  "quality": {"confidence":0.96,"source_count":3},
  "uncertainty": {"position_covariance":[0.04,0,0,0.04]},
  "provenance": ["msg_01J...","evidence:fusion:8812"]
}
```

11.4 Revision and conflict semantics

- Single-writer objects use a monotonic revision and conditional update.
- Distributed objects use vector or sequence metadata sufficient to identify concurrent updates.
- A consumer requesting deltas shall provide its last confirmed revision or resume cursor.
- A missing delta creates a gap state; the consumer obtains a snapshot before resuming authoritative processing.
- Conflict resolution produces a new decision record referencing all branches and the applied policy or human approval.

Requirement ID	Normative requirement	Verification
API-STA-001	Every state object SHALL include authoritative owner, revision, observation time and validity or age semantics.	Schema audit.
API-STA-002	Consumers SHALL detect and recover from delta gaps before using state for control.	Gap/recovery test.
API-STA-003	Concurrent conflicting updates SHALL be retained as conflict branches until resolved.	Partition merge test.
API-STA-004	A snapshot SHALL identify the cutoff time and exact object revisions included.	Snapshot consistency test.
API-STA-005	Historical replay SHALL remain isolated from LIVE namespaces and command paths.	Replay isolation test.

DOMAIN API

12. Peer Discovery and Mesh API

The Peer Discovery and Mesh API enables trusted nodes to discover mission-relevant peers, advertise supported schema ranges and form bounded communication topologies. Peer discovery is separate from JAUS Discovery and from network-layer discovery; these sources may identify candidates, but A^{3}O admits a peer only after identity, mission, profile and policy checks.

12.1 Peer objects

Object	Purpose	Required fields	Lifecycle / notes
PeerCandidate	Observed but not admitted peer.	protocol identities, claimed node, source, first_seen, evidence	No mission data or authority.
PeerMembership	Approved mission-scoped membership.	node_id, mission_id, role, trust state, schema ranges, expiry	Time-bounded and revocable.

Object	Purpose	Required fields	Lifecycle / notes
MeshLink	Logical peer link and QoS.	source, destination, transport, quality, security, last_seen	May be indirect or store-and-forward.
TopologyState	Current graph and partition view.	topology_id, members, links, components, revision	Mission-scoped projection.
PartitionNotice	Detected separation or inconsistent view.	partition_id, members, detected_at, confidence	Triggers degraded policy.

12.2 Operations and events

Method / Channel	Resource or operation	Semantics	Authority
POST	/v1/peers:discover	Submit candidate evidence from protocol or network discovery.	Gateway
POST	/v1/missions/{id}/members:join	Request mission membership with role and profile.	Node / operator
POST	/v1/missions/{id}/members/{node}:approve	Approve or condition membership.	Mission authority
POST	/v1/missions/{id}/members/{node}:quarantine	Remove peer from operational exchange without deleting evidence.	Security / mission authority
GET	/v1/meshes/{mesh_id}/topology	Read topology and partition state.	Mission read
CHANNEL	a3o.mesh.topology.changed	Membership, link, partition and restoration events.	Mission subscribers

12.3 Join handshake

```
PEER ADMISSION SEQUENCE
PeerHello(node_id, mission_id, nonce, supported_profiles, schema_ranges)
  → IdentityProof(credential, attestation, signed nonce)
  → MembershipDecision(allow | deny | conditions, role, expiry)
  → CapabilityAdvertisement
  → StateSummary(revision cursors)
  → MeshReady(topology revision)
```

12.4 Anti-Sybil and isolation rules

- One credential cannot create unbounded simultaneous logical peers unless the profile explicitly allows workload replicas.
- Protocol identities are correlated to physical or workload identities and ownership.
- A quarantined peer remains visible in security and evidence views but is removed from tasking and control exchange.
- Topology changes are signed or authenticated and include revision and source.
- A peer that loses mission membership may retain only the minimal channels required for safe exit and evidence export.

Requirement ID	Normative requirement	Verification
API-MSH-001	Peer discovery SHALL NOT create mission membership or trust.	Candidate admission test.
API-MSH-002	Membership SHALL be mission-scoped, role-scoped, time-bounded and revocable.	Membership lifecycle test.
API-MSH-003	Topology events SHALL identify revision, source and affected members or links.	Event schema test.
API-MSH-004	A quarantined peer SHALL be excluded from task allocation and command coordination.	Quarantine scenario.
API-MSH-005	The mesh SHALL detect and report partitions without expanding peer authority.	Partition scenario.

DOMAIN API

13. Capability Exchange API

Capability exchange describes what a node can provide, under what limits and in what current condition. It is not a marketing feature list. A CapabilityAdvertisement is a signed, versioned, time-bounded operational claim that includes required resources,

safety restrictions, supported actions, payload, mobility, energy, compute, communications and workload. The mission planner uses only currently valid capabilities admitted by profile and policy.

13.1 Capability objects

Object	Purpose	Required fields	Lifecycle / notes
CapabilityDefinition	Stable semantic capability contract.	capability_id, profile, inputs, outputs, actions, limits	Versioned profile artifact.
CapabilityAdvertisement	Node claim that it implements a capability.	node_id, capability_id, version, limits, safety, TTL, signature	Expires or changes with health.
CapabilityState	Current availability and capacity.	status, utilization, energy, queue, degraded reason	Frequently updated.
CapabilityRequirement	Mission/task need.	required capability, minimum performance, environment, redundancy	Used for allocation.
CapabilityMatch	Planner result.	requirement, candidates, scores, exclusions, policy	Evidence-linked decision support.

13.2 Advertisement example

CAPABILITYADVERTISEMENT EXAMPLE

```
{
  "advertisement_id": "capadv:01J...",
  "node_id": "node:hub:amr-042",
  "capability_id": "urn:a3o:capability:material-transport:1.1",
  "profile_id": "urn:a3o:profile:logistics-amr:1.1.0",
  "valid_until": "2026-07-17T10:35:00Z",
  "limits": {"payload_kg":250,"max_speed_mps":2.0,"slope_deg":8},
  "energy": {"soc":0.74,"reserve_policy":"mission-plus-return"},
  "communications": {"profiles":["secure-p2p","mqtt"],"partition_mode":true},
  "safety_restrictions": ["no_public_zone","human_proximity_speed_limit"],
  "workload": {"active_tasks":1,"queue_depth":0},
  "signature": "..."
}
```

13.3 Operations and events

Method / Channel	Resource or operation	Semantics	Authority
PUT	/v1/nodes/{node}/capabilities/{capability}	Publish or refresh advertisement with conditional revision.	Node identity
GET	/v1/capabilities:search	Find candidates by semantic capability, limits, location and trust.	Planner / operator
POST	/v1/capabilities:match	Evaluate a task requirement against candidates.	Mission planner
CHANNEL	a3o.capability.changed	Advertisement, degradation, expiry and withdrawal events.	Mission subscribers
CHANNEL	a3o.capability.match.created	Decision-support result and exclusions.	Mission/evidence

13.4 Matching and scoring

- Hard constraints are evaluated before optimization scores.
- Trust, safety restrictions and profile compatibility can exclude a candidate regardless of performance score.
- Energy, workload, route, communications and maintenance state are time-dependent inputs.
- A match is advisory until task assignment and authority checks complete.
- Scoring algorithms and weights are versioned and recorded in evidence.

Requirement ID	Normative requirement	Verification
API-CAP-001	Capability advertisements SHALL be signed or authenticated, versioned and time-bounded.	Advertisement validation.
API-CAP-002	Expired or withdrawn capabilities SHALL NOT be used for new task allocation.	Expiry test.

Requirement ID	Normative requirement	Verification
API-CAP-003	Hard safety, trust and profile constraints SHALL be applied before optimization scoring.	Matching test.
API-CAP-004	Capability match results SHALL record excluded candidates and reasons.	Evidence audit.
API-CAP-005	A capability advertisement SHALL NOT grant command authority.	Authority separation test.

DOMAIN API

14. Mission and Task APIs

Mission and Task APIs express intent, planning, allocation and execution coordination without coupling missions to a particular robot vendor. A MissionIntent defines the objective and constraints; a MissionPlan defines tasks and dependencies; TaskAssignment binds a task to one or more nodes for a bounded lease; execution is performed through Action APIs or through a protocol-specific mission mechanism such as a JAUS Mission Spooler.

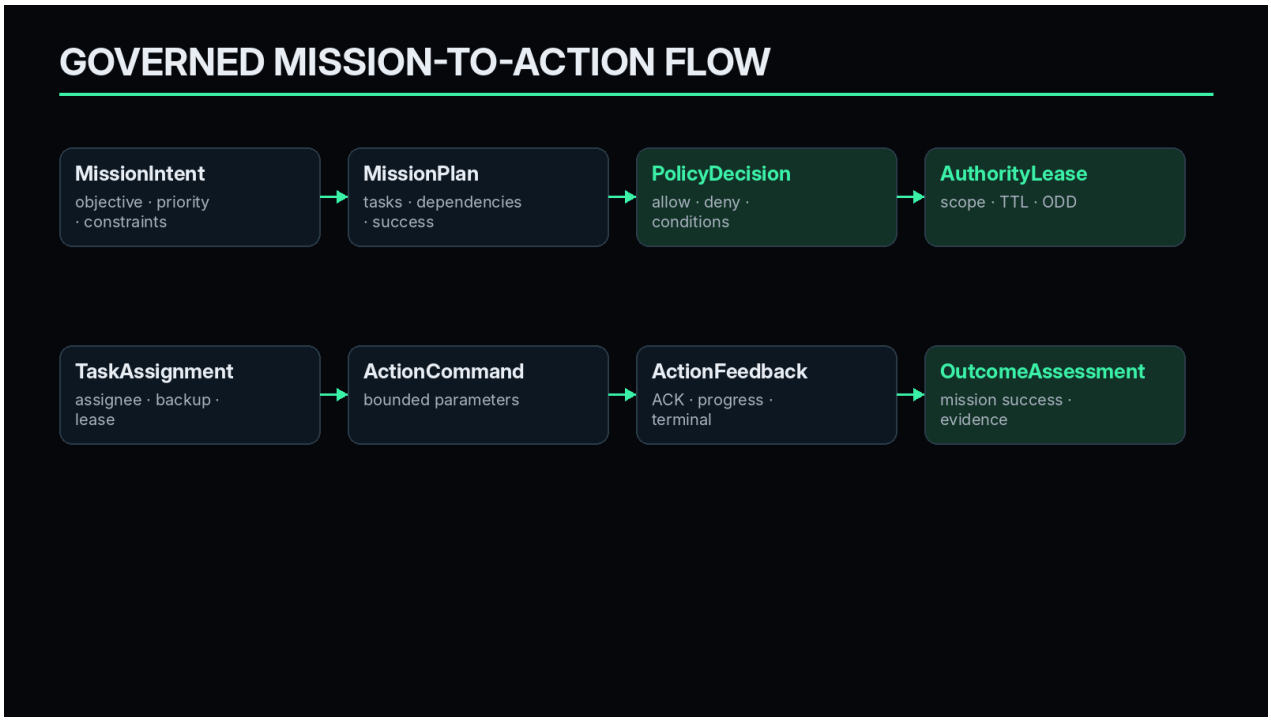

Figure 5 — Governed mission-to-action lifecycle.

14.1 Core objects

Object	Purpose	Required fields	Lifecycle / notes
MissionIntent	Why and under what constraints the ecosystem acts.	objective, priority, scope, authority, success, termination	Immutable root intent; amendments create revisions.
MissionPlan	Tasks, dependencies and resources.	plan_id, tasks, edges, assumptions, policy refs, version	May be replanned with traceability.
TaskDefinition	Unit of work.	task_id, type, inputs, required capabilities, constraints, success	Reusable or mission-specific.
TaskAssignment	Task-to-node lease.	assignee, backups, lease, acceptance, authority, required profile	Expires, preempts or reassigns.
TaskProgress	Execution state and milestones.	task_id, state, progress, timestamps, blockers	Distinct from action-level progress.
MissionOutcome	Mission-level result.	success, partial success, termination reason, metrics, evidence	Final or post-mission revised assessment.

14.2 Allocation modes

Mode	Selection rule	Typical use
DIRECTED	Mission authority selects assignee.	High assurance or fixed ownership.
AUCTION	Eligible nodes bid using capability and cost.	Distributed optimization.
PRIORITY	Highest-priority eligible node accepts.	Fast response.
REDUNDANT	Multiple nodes perform or monitor the task.	High reliability or verification.
OPPORTUNISTIC	Node accepts when local conditions permit.	Intermittent or exploratory missions.
JAUS_MISSION_SPOOL	Platform-local task list through Mission Spooler.	JAUS execution boundary.

14.3 Operations and events

Method / Channel	Resource or operation	Semantics	Authority
POST	/v1/missions	Create MissionIntent and initial authority context.	Mission creator
POST	/v1/missions/{id}:plan	Generate or submit a versioned MissionPlan.	Planner
POST	/v1/tasks/{id}:allocate	Run allocation mode and produce assignment proposal.	Mission authority
POST	/v1/assignments/{id}:accept	Assignee accepts with declared conditions and current capability revision.	Assigned node
POST	/v1/assignments/{id}:preempt	Terminate or supersede assignment with reason.	Authorized mission controller
CHANNEL	a3o.mission.lifecycle.changed	Mission state, plan revision, suspension and termination.	Mission subscribers
CHANNEL	a3o.task.assignment.changed	Offer, acceptance, rejection, lease expiry and reassignment.	Participants

14.4 Mission state model

te	Meaning
AFT	Intent under construction.
THORIZED	Intent and authority accepted.
NNED	Executable plan exists.
OCATING	Tasks are being assigned.
CUTING	At least one task is active.
SED	Controlled suspension with retained state.
RADED	Mission continues under restricted policy.
RMINATING	Safe stop or exit in progress.
MPLETED	Success criteria reached.
LED	Termination criteria or unrecoverable failure.
NCELLED	Authorized cancellation.

Requirement ID	Normative requirement	Verification
API-MIS-001	MissionIntent SHALL define objective, scope, authority, success and termination criteria.	Schema test.
API-MIS-002	Every plan revision SHALL preserve traceability to the intent, assumptions and replaced plan.	Replanning evidence test.

Requirement ID	Normative requirement	Verification
API-MIS-003	TaskAssignment SHALL be time-bounded and SHALL reference current capability and authority context.	Lease test.
API-MIS-004	Allocation results SHALL record candidates, exclusions, scoring and decision authority.	Allocation evidence test.
API-MIS-005	Platform-local JAUS mission execution SHALL remain subordinate to the ecosystem mission intent and policy.	JAUS mission-spool scenario.
API-MIS-006	Task acceptance SHALL NOT imply permission to execute actions outside the assignment scope.	Scope enforcement test.

DOMAIN API

15. Policy, Delegation and Human Authority API

The Policy, Delegation and Human Authority API is the decision gate between proposed intent and permitted execution. Policy evaluation produces a machine-readable decision with reasons, conditions and required approvals. Delegation grants bounded authority to a user, workload or node for a specific scope, operating domain and time. Human approvals are explicit records, not UI clicks without identity and context.

15.1 Core objects

Object	Purpose	Required fields	Lifecycle / notes
PolicyEvaluation	Request to evaluate facts against policy.	subject, action, resource, context, policy set, timestamp	Immutable input to decision.
PolicyDecision	ALLOW, DENY, REQUIRE_APPROVAL or CONDITIONAL.	decision_id, result, reasons, obligations, policy versions	Expires when facts or context change.
DelegationGrant	Bounded authority delegated to a principal.	grantor, grantee, scope, ODD, TTL, conditions, revoke rules	Never broader than grantor authority.
HumanApproval	Authenticated human authorization record.	approver, qualification, decision, scope, expiry, evidence	May satisfy a policy obligation.
SafetyEnvelope	Action bounds independent of mission optimization.	limits, zones, rates, prohibited states, source	Enforced at execution boundary.
EmergencyAuthority	Profile-specific emergency capability.	eligible principals, triggers, scope, post-action review	Highly restricted and separately audited.

15.2 Policy evaluation example

POLICYEVALUATION EXAMPLE

```
{
 "evaluation_id": "eval_01J...",
 "subject": {"type":"workload","id":"workload:planner:17"},
 "action": "task.assign",
 "resource": {"type":"node","id":"node:hub:amr-042"},
 "context": {
  "mission_id":"mission:1187",
  "operational_domain":"warehouse-zone-a",
  "risk_level":"MEDIUM",
  "human_present":true,
  "capability_revision":"caprev:183"
 },
 "requested_scope": {"task_id":"task:deliver-03","ttl_s":300}
}
```

15.3 Control precedence

MANDATORY PRECEDENCE
1. Physical emergency and certified safety interlocks
2. Local platform safety kernel
3. JAUS Access Control ownership or equivalent protocol control
4. A³O policy and delegation constraints
5. Human operator authorization
6. Mission orchestration
7. AI optimization and recommendations

15.4 Operations and events

Method / Channel	Resource or operation	Semantics	Authority
POST	/v1/policy:evaluate	Evaluate a typed action/resource/context request.	Authenticated caller
POST	/v1/delegations	Create a bounded grant after policy and grantor checks.	Grantor
POST	/v1/delegations/{id}:revoke	Revoke and propagate to dependent leases and sessions.	Grantor / security authority
POST	/v1/approvals	Record authenticated human approval or rejection.	Qualified human
GET	/v1/decisions/{id}	Read decision, reasons, obligations and evidence links.	Authorized reader
CHANNEL	a3o.authority.changed	Grant, expiry, revocation and conflict events.	Control services

15.5 Separation of duties

- A profile may require different principals for mission creation, approval and execution.
- A user cannot approve an action when the applicable policy requires independent review and the user is the requester.
- Emergency authority cannot be used to bypass evidence capture or post-action review.
- A workload cannot grant itself broader authority than its identity or parent delegation.
- A changed safety envelope invalidates dependent policy decisions and action authorizations.

Requirement ID	Normative requirement	Verification
API-POL-001	Every decision SHALL include policy versions, result, reasons, obligations and expiry or invalidation conditions.	Decision schema test.
API-POL-002	Delegated authority SHALL be no broader or longer-lived than the grantor authority.	Delegation containment test.
API-POL-003	Human approval SHALL bind identity, qualification, scope, context and expiry.	Approval evidence test.
API-POL-004	AI services SHALL NOT create or extend authority grants.	Negative AI authority test.
API-POL-005	Revocation SHALL invalidate dependent command paths within the declared budget.	Revocation propagation test.
API-POL-006	Policy uncertainty or unavailable policy-critical data SHALL fail closed or enter an explicitly approved safe mode.	Policy dependency test.

DOMAIN API

16. Action and Feedback API

The Action API translates authorized operational intent into a bounded command lifecycle. It does not expose arbitrary vendor commands. Each ActionCommand names a canonical action, target, parameters, preconditions, authority context, safety envelope, idempotency key and expiry. The execution gateway maps it to the target protocol and preserves the mapping and raw protocol exchange in evidence.

16.1 Action lifecycle

te	Meaning
OPOSED	Intent exists but has not passed all gates.
THORIZED	Policy, authority and safety checks passed.

te	Meaning
PATCHING	Protocol encoding and transmission in progress.
KNOWLEDGED	Target accepted transport or service request.
ECUTING	Operational work is in progress.
JSED	Execution paused while retaining context.
NCELLING	Cancellation or safe-stop requested.
CCEEDED	Declared outcome achieved.
LED	Execution ended without success.
ECTED	Target or gateway refused before execution.
PIRED	Command was not valid when used.
KNOWN	Outcome cannot yet be established; evidence retained.

16.2 ActionCommand example

ACTIONCOMMAND EXAMPLE

```
{
  "action_id": "action:01J...",
  "target_node_id": "node:hub:amr-042",
  "canonical_action": "mobility.navigate_to",
  "parameters": {"pose":{"frame":"map:hub-01","x":210.0,"y":64.5,"yaw":1.57}},
  "preconditions": ["state.operational=READY","energy.soc>=0.30"],
  "mission_id": "mission:1187",
  "task_id": "task:deliver-03",
  "policy_decision_id": "decision:01J...",
  "authority_lease_id": "lease:01J...",
  "safety_envelope_id": "safety:warehouse-human-zone:2.1",
  "idempotency_key": "mission1187-task03-attempt01",
  "expires_at": "2026-07-17T10:45:00Z"
}
```

16.3 Operations and events

Method / Channel	Resource or operation	Semantics	Authority
POST	/v1/actions	Submit governed action intent; returns action resource.	Authorized principal
GET	/v1/actions/{id}	Read lifecycle, progress, mapping and evidence links.	Mission/control reader
POST	/v1/actions/{id}:cancel	Request bounded cancellation or safe stop.	Authorized controller
POST	/v1/actions/{id}:pause	Request pause if supported by capability/profile.	Authorized controller
POST	/v1/actions/{id}:resume	Resume after revalidation of authority and safety.	Authorized controller
CHANNEL	a3o.action.feedback	ACK, progress, warnings, terminal status and target diagnostics.	Mission subscribers
CHANNEL	a3o.action.outcome.assessed	Operational outcome separate from transport ACK.	Mission/evidence

16.4 Feedback versus outcome

Layer	Meaning	Use
Transport acknowledgement	Message or request reached a protocol endpoint.	Does not prove command acceptance or execution.
Service acknowledgement	Target service accepted or rejected the request.	May still precede execution.
Action progress	Execution milestones, current state and warnings.	May be estimated or incomplete.
Terminal feedback	Target reports success, failure, cancellation or unknown.	Must be validated against resulting state.

Layer	Meaning	Use
OutcomeAssessment	A³O evaluates mission-relevant result using state and evidence.	Authoritative operational conclusion for mission logic.

Requirement ID	Normative requirement	Verification
API-ACT-001	ActionCommand SHALL include target, canonical action, bounded parameters, authority, policy, safety, idempotency and expiry context.	Schema test.
API-ACT-002	The execution gateway SHALL revalidate authority and safety immediately before protocol encoding.	Time-of-use validation test.
API-ACT-003	Duplicate action submissions with the same idempotency key SHALL NOT produce duplicate execution.	Duplicate test.
API-ACT-004	Transport or service acknowledgement SHALL NOT be reported as operational success.	Outcome separation test.
API-ACT-005	Cancellation SHALL define supported semantics and safe fallback when the target cannot cancel.	Cancellation scenario.
API-ACT-006	Protocol mapping version and raw command/feedback exchange SHALL be retained in evidence.	Evidence audit.

DOMAIN API

17. Evidence and Assurance API

Evidence is generated during operation, not reconstructed only after an incident. The Evidence API receives fragments from gateways, state services, policy engines, mission services, execution gateways and human-approval systems. Fragments are linked into an Evidence Graph and may be assembled into an EvidencePackage for audit, conformance, incident review, customer acceptance or regulated-domain assurance.

17.1 Evidence objects

Object	Purpose	Required fields	Lifecycle / notes
EvidenceFragment	Atomic signed or integrity-protected record.	evidence_id, type, producer, time, subject, digest, classification	Append-only; may reference external media.
EvidenceLink	Typed relation between fragments or operational objects.	source, relation, target, confidence	Builds graph lineage.
EvidenceManifest	Ordered or set-based manifest of content.	manifest_id, members, hashes, policy	Used for export and verification.
EvidencePackage	Purpose-specific export.	scope, manifest, disclosures, signatures, verification result	Immutable release artifact.
AssuranceClaim	Claim supported by evidence.	claim_id, statement, context, evidence refs, status	Part of assurance case.
VerificationResult	Cryptographic and semantic verification.	package, checks, failures, verifier, time	Machine-readable.

17.2 Operations and events

Method / Channel	Resource or operation	Semantics	Authority
POST	/v1/evidence/fragments	Append a fragment with integrity and classification metadata.	Approved producer
POST	/v1/evidence/links	Create typed lineage relation.	Evidence producer
POST	/v1/evidence/packages:build	Assemble purpose-specific package with disclosure rules.	Assurance authority
POST	/v1/evidence/packages/{id}:verify	Verify hashes, signatures, completeness and policy.	Verifier
GET	/v1/evidence/graph	Query by mission, action, entity, time or evidence relation.	Authorized auditor

Method / Channel	Resource or operation	Semantics	Authority
CHANNEL	a3o.evidence.appended	New fragment or link notification.	Authorized subscribers

17.3 Control-action evidence chain

```
MINIMUM CONTROL EVIDENCE LINEAGE
MissionIntent
  → MissionPlan revision
  → CapabilityMatch
  → PolicyDecision
  → HumanApproval / DelegationGrant
  → ActionCommand
  → Protocol mapping and raw command
  → Target acknowledgement and feedback
  → Resulting state observations
  → OutcomeAssessment
  → EvidencePackage
```

17.4 Selective disclosure

- Package scope and purpose are explicit and signed.
- Sensitive fields may be redacted only through a transformation that records source digest, redaction policy and verifier-visible manifest.
- Cross-domain export preserves classification labels and prohibits unauthorized downgrade.
- Media may remain in controlled storage while the package contains a content digest and authorized retrieval reference.
- Retention and deletion obligations apply to data copies without erasing required audit lineage unless legally mandated.

Requirement ID	Normative requirement	Verification
API-EVD-001	Evidence fragments SHALL be append-only and integrity-protected.	Tamper test.
API-EVD-002	Control-relevant actions SHALL have traceable links from intent through outcome.	Evidence completeness test.
API-EVD-003	Selective disclosure SHALL preserve source digest, transformation record and policy.	Disclosure verification.
API-EVD-004	Evidence export SHALL enforce classification, privacy and retention policy.	Export policy test.
API-EVD-005	Package verification SHALL be machine-readable and identify missing, invalid or unverifiable elements.	Verification result test.

DOMAIN API

18. Twin, Simulation and Replay API

The Twin, Simulation and Replay API maintains models of assets, processes and environments while preserving strict namespace separation between LIVE, SIMULATION, FORECAST and REPLAY. Simulation or replay data can support planning, training and verification but cannot enter live command paths without explicit promotion, validation and policy.

18.1 Namespace model

Namespace	Purpose	Control rule
LIVE	Current operational state from live sources.	Eligible for policy and action according to freshness and trust.
SIMULATION	Synthetic execution in an isolated scenario.	No live action unless separately approved hardware-in-loop profile.
FORECAST	Predicted future state with model and uncertainty.	Decision support only; clearly labeled.
REPLAY	Historical reconstruction from evidence and recorded data.	Read-only; no live event-channel collision.
TRAINING	Scenario and user-training environment.	Separate identities, profiles and evidence labels.

18.2 Twin objects

Object	Purpose	Required fields	Lifecycle / notes
TwinEntity	Model of physical or logical entity.	twin_id, source entity, geometry, parameters, namespace, revision	May contain model confidence.
EnvironmentModel	Map, zones, weather, process or operational environment.	model_id, frame, layers, revision, validity	Versioned and spatially scoped.
Scenario	Initial state, perturbations, mission and evaluation criteria.	scenario_id, inputs, seeds, models, expected outputs	Reproducible when deterministic inputs exist.
Forecast	Predicted trajectory or state distribution.	model, horizon, assumptions, uncertainty, generated_at	Expires rapidly.
ReplaySession	Controlled playback.	session_id, evidence scope, clock mode, speed, cursor	Isolated channel prefix.

18.3 Operations

Method / Channel	Resource or operation	Semantics	Authority
POST	/v1/scenarios	Create a simulation or replay scenario.	Simulation authority
POST	/v1/scenarios/{id}:run	Start execution with deterministic seed and resource budget.	Simulation operator
GET	/v1/twins/{id}	Read twin model and source revision.	Twin reader
POST	/v1/replays	Create replay session from evidence manifest and time range.	Auditor / trainer
POST	/v1/forecasts	Generate forecast with model and assumptions.	Planner
CHANNEL	a3o.simulation.result	Scenario result, metrics and evidence manifest.	Simulation subscribers

18.4 Promotion from simulation

- Simulation output may create a recommendation or candidate plan, not an executable live command.
- Promotion requires explicit conversion into live MissionIntent or configuration change with new identity, policy and approval.
- Models, datasets, seeds and software builds are recorded for reproducibility.
- Forecast confidence and assumptions remain attached when the forecast influences a decision.
- Replay event channels use isolated addresses and identities so live consumers cannot subscribe accidentally.

Requirement ID	Normative requirement	Verification
API-TWN-001	LIVE, SIMULATION, FORECAST, REPLAY and TRAINING namespaces SHALL be distinguishable in identity, routing and evidence.	Namespace isolation test.
API-TWN-002	Simulation output SHALL NOT directly produce a live ActionCommand.	Negative promotion test.
API-TWN-003	Scenarios SHALL record model versions, input manifests, random seeds and execution environment.	Reproducibility audit.
API-TWN-004	Forecasts SHALL include generation time, horizon, assumptions and uncertainty.	Schema test.
API-TWN-005	Replay channels SHALL be isolated from live subscribers and command routes.	Replay isolation test.

DOMAIN API

19. Health, Observability and SLO API

Health and observability expose the operational condition of nodes, services, links, queues, authority dependencies and evidence pipelines. A heartbeat alone is not health; it proves only a form of liveness. Health includes readiness, trust, safety, workload, dependency and data-quality state. Observability data is classified and access-controlled because it may reveal topology, vulnerabilities or operational patterns.

19.1 Health objects

Object	Purpose	Required fields	Lifecycle / notes
Heartbeat	Minimal liveness signal.	source, sequence, sent_at, observed_at	No readiness or trust conclusion.
HealthState	Aggregated operational health.	status, readiness, degraded reasons, dependencies, revision	Owned by health service.
ResourceState	CPU, memory, storage, energy and queue usage.	resource, utilization, limits, forecast	Supports admission and planning.
LinkState	Latency, loss, jitter, bandwidth and security state.	endpoints, transport, metrics, last_update	Feeds topology and partition logic.
ServiceSLO	Latency, availability, throughput and error budget.	service, indicators, objectives, window	Profile-specific.
EvidenceOffset	Last durable evidence sequence or manifest position.	producer, sequence, durable_at	Detects evidence lag.

19.2 API and event surface

Method / Channel	Resource or operation	Semantics	Authority
GET	/v1/health	Service or node health summary.	Operations reader
GET	/v1/health/{id}/dependencies	Dependency tree and degraded reasons.	Operations / assessor
GET	/v1/metrics	Profile-specific metrics export.	Monitoring identity
GET	/v1/traces/{trace_id}	Distributed trace with classification filtering.	Authorized diagnostics
CHANNEL	a3o.health.changed	Health, trust, safety and evidence-offset transitions.	Operations subscribers
CHANNEL	a3o.slo.budget.alert	Error-budget or latency-budget threshold event.	Service owner

19.3 Standard status model

tus	Meaning
KNOWN	Insufficient current evidence.
RTING	Dependencies and readiness checks in progress.
DY	Available for declared role.
GRADED	Available with explicit restrictions.
T_READY	Alive but not permitted to accept operational work.
LED	Unable to provide declared service.
ARANTINED	Security or trust isolation.
NTENANCE	Intentionally unavailable or restricted.

Requirement ID	Normative requirement	Verification
API-OBS-001	Heartbeat SHALL NOT be interpreted as readiness, trust or safety.	Health logic test.
API-OBS-002	HealthState SHALL identify degraded reasons, affected capabilities and dependency status.	Schema test.
API-OBS-003	Control services SHALL enforce readiness and safety status at time of use.	Action admission test.
API-OBS-004	Observability exports SHALL enforce classification and least privilege.	Access-control test.
API-OBS-005	Evidence pipeline lag SHALL be observable and capable of triggering safe degradation.	Evidence-offset fault test.

20. Offline, Partition and Reconciliation

A³O is designed for intermittent connectivity and distributed edge operation. Offline capability is bounded by previously delegated authority, local safety and mission policy. Nodes buffer state and evidence with original timestamps and provenance. Reconnection does not overwrite conflicts silently; it enters reconciliation, exchanges cursors and manifests, preserves divergent branches and produces a ReconciliationReport.

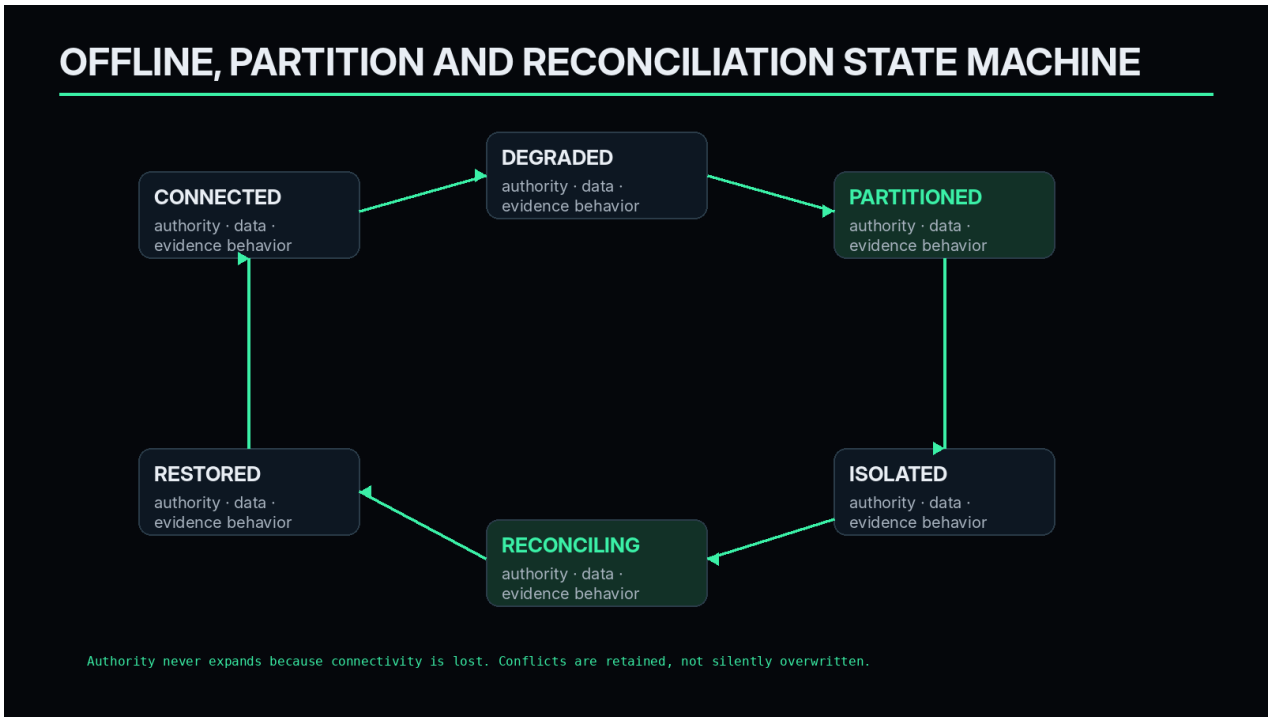

Figure 6 — Connectivity and reconciliation state machine.

20.1 Connectivity states

te	API behavior
NNECTED	Normal streaming, synchronization and policy reachability.
GRADED	Priority classes, reduced update rates and explicit restrictions.
RTITIONED	Local delegated operation; buffer state and evidence; no authority expansion.
LATED	Profile-specific safe fallback; peer or central trust unavailable.
CONCILING	Exchange deltas, manifests and authority history; preserve conflicts.
STORED	Publish reconciliation result and resume only the approved mode.

20.2 Reconciliation objects

Object	Purpose	Required fields	Lifecycle / notes
SyncCursor	Per-stream confirmed position.	stream_id, sequence/revision, durable_at	Used to request missing data.
BufferedOperation	Offline state/action/evidence record.	operation_id, created_at, authority, expiry, payload, result	May be accepted, rejected or retained only as evidence.
ConflictSet	Competing state or decisions.	object, branches, sources, vector metadata	Requires policy or human resolution.
AuthorityHistory	Delegations and revocations observed locally.	leases, revisions, last trusted time	Used to detect expired or revoked work.
ReconciliationReport	Outcome of merge.	accepted, rejected, conflicts, unresolved, operator actions	Published before restored mode.

20.3 Offline action rules

- Only actions explicitly permitted by a still-valid delegated authority and partition policy may execute.
- Authority expiry is evaluated against trusted local time; uncertain time causes stricter behavior.
- A node records the authority and policy information available when the action was taken.
- Locally completed actions are not repeated on reconnect merely because the central system did not observe them.
- Conflicting task ownership or state is retained and resolved before dependent high-risk actions resume.
- Evidence buffering has a reserved capacity or safe-degradation rule so operational activity cannot continue without required evidence indefinitely.

Requirement ID	Normative requirement	Verification
API-OFF-001	Connectivity loss SHALL NOT expand authority or operating domain.	Partition authority test.
API-OFF-002	Offline actions SHALL reference the local authority, policy version and trusted-time state used.	Evidence audit.
API-OFF-003	Reconciliation SHALL preserve concurrent conflicts and SHALL NOT use last-write-wins for policy-critical state unless the profile explicitly justifies it.	Conflict merge test.
API-OFF-004	Completed offline actions SHALL be deduplicated during reconnect.	Reconnect duplicate test.
API-OFF-005	RESTORED state SHALL require a completed ReconciliationReport and approved resume mode.	State transition test.
API-OFF-006	Evidence-buffer exhaustion SHALL trigger a declared safe degradation or stop rule.	Storage exhaustion test.

DOMAIN API

21. Industry Profile SDK

The Industry Profile SDK packages the semantic, workflow, policy, safety and conformance material needed to apply A³O in a domain without modifying core contracts. A profile composes core objects and protocol bindings, defines domain vocabulary and prohibited semantics, declares required capabilities and tests, and specifies which external safety or compliance regimes remain applicable.

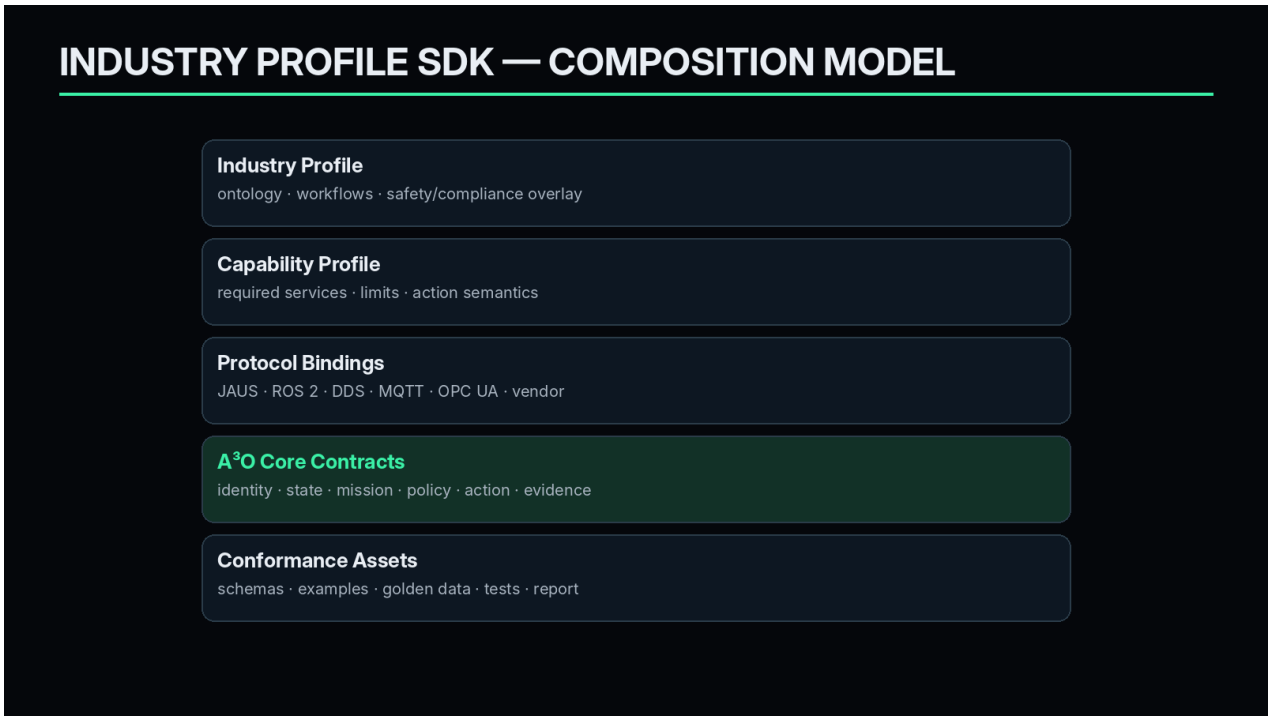

Figure 7 — Industry Profile SDK composition model.

21.1 Profile manifest

```
ILLUSTRATIVE PROFILE MANIFEST
profile_id: urn:a3o:profile:logistics-amr:1.1.0
status: candidate
core_contract_range: ">=1.2 <2.0"
protocol_bindings:
  - a3o-http: 1.2
  - a3o-event: 1.2
  - jaus-core: 1.0
required_capabilities:
  - material-transport
  - local-pose
  - energy-state
objects:
  - schemas/LogisticsTask.json
  - schemas/CustodyEvent.json
policies:
  - policies/human-zone-speed.rego
conformance:
  suite: tests/logistics-amr
prohibited_overrides:
  - ActionCommand.authority_semantics
  - EvidenceFragment.integrity_semantics
external_obligations:
  - machinery_safety_assessment
  - site_cybersecurity_controls
```

21.2 Profile contents

Artifact	Purpose
Manifest	Identity, version, compatibility range, status, owner and artifact digests.
Ontology extensions	Domain concepts, terms, relationships and mapping rules.
State and action schemas	Domain-specific fields and actions built on core semantics.
Workflow templates	Mission/task patterns, transitions, compensating actions and escalation.
Policy packs	Domain rules, approval obligations, ODD constraints and emergency behavior.
Protocol mappings	JAUS, ROS 2, DDS, OPC UA or vendor mappings with semantic tests.
Examples and golden data	Valid, invalid, edge and migration examples.
Conformance suite	Contract, fault, security, performance and scenario tests.
Assurance notes	Hazards, external compliance, claim limits and evidence requirements.

21.3 Core semantic protections

- A profile may narrow authority but cannot broaden core authority semantics.
- A profile may add evidence but cannot disable required integrity or lineage.
- A profile may define domain-specific actions but shall map them to bounded canonical action semantics.
- A profile may choose stricter security, privacy, retention or human-approval requirements.
- A protocol binding may optimize representation but shall preserve canonical units, frames, lifecycle and errors.

21.4 Initial profile families

Profile family	Scope
Manufacturing	Industrial robots, cobots, AMR/AGV, cells, machine and process state, quality evidence.
Logistics	Warehouse robots, autonomous trucks, delivery UAS, ports, custody and handoff.
Inspection & information collection	UAS/UGV/USV/UUV, multimodal sensing, coverage, dataset provenance.
Care, medical & response	Delivery, telepresence, assistance and emergency robotics with separate regulated-domain assurance.

Profile family	Scope
Security & defence	DroneDefender and other multi-domain mission profiles with bounded human authority and evidence.

Requirement ID	Normative requirement	Verification
API-PRF-001	Every profile SHALL declare compatibility ranges, external obligations, owner and lifecycle state.	Manifest validation.
API-PRF-002	Profiles SHALL NOT override core identity, authority, action-outcome or evidence-integrity semantics.	Semantic protection test.
API-PRF-003	Every operationally significant extension SHALL include valid, invalid and fault examples.	Artifact completeness test.
API-PRF-004	Protocol mappings SHALL include round-trip and semantic-equivalence tests.	Mapping conformance test.
API-PRF-005	A profile SHALL state which certifications or regulatory approvals remain outside A³O conformance.	Governance review.

PART III

Integration, Security, Conformance and Worked Examples

Safe failure, security, test evidence, release gates, cross-industry examples and implementation roadmap.

PART III · INTEGRATION ASSURANCE

22. Errors, Resilience and Safe Failure

The canonical error model separates transport failure, contract failure, policy or authority denial, target rejection, execution failure and outcome uncertainty. A transport-specific status code is insufficient because the same HTTP 409 or JAUS rejection can represent different operational meanings. Every error response therefore includes a canonical code, retryability, responsible domain, correlation, safety effect and evidence reference when available.

22.1 Error object

CANONICALERROR EXAMPLE

```
{
  "error_id": "err_01J...",
  "code": "A3O.AUTHORITY.LEASE_EXPIRED",
  "message": "Authority lease expired before action dispatch.",
  "domain": "AUTHORITY",
  "severity": "HIGH",
  "retryable": false,
  "safe_effect": "ACTION_NOT_DISPATCHED",
  "correlation_id": "corr_01J...",
  "resource": {"type":"action","id":"action:01J..."},
  "details": {"lease_id":"lease:01J...","expired_at":"2026-07-17T10:45:00Z"},
  "evidence_id": "evidence:01J..."
}
```

22.2 Error domains

main	Meaning
NTRACT	Schema, content type, required field, semantic or version failure.
ENTITY	Unknown, expired, revoked or mismatched principal.
UST	Attestation, quarantine, ownership or integrity failure.
LICY	Denied, obligation unmet or policy dependency unavailable.
THORITY	Missing, expired, conflicted or insufficient authority.
FETY	Safety envelope, interlock or operating-domain violation.
ATE	Stale, conflicting, incomplete or invalid authoritative state.
PABILITY	Unsupported, degraded, expired or insufficient capability.
ANSPORT	Timeout, loss, framing, link or delivery failure.
OTOCOL	JAUS/DDS/ROS/vendor message or service-level failure.
ECUTION	Target rejection, fault, cancellation or terminal failure.
TCOME	Execution ended but operational result remains unknown or disputed.
IDENCE	Evidence gap, integrity failure or retention/export restriction.
SOURCE	Rate, storage, queue, compute, energy or concurrency limit.

22.3 Retry and compensation rules

Class	Rule
Safe automatic retry	Idempotent query, telemetry publish or action intent with unchanged authority and valid expiry.
Conditional retry	Only after refreshed state, authority, capability or policy decision.
No automatic retry	Safety denial, authority denial, semantic ambiguity, target rejection or unknown outcome.
Compensating action	Separate governed action; never hidden as an internal retry.

Class	Rule
Manual review	Conflicting outcome, evidence gap, policy ambiguity or repeated protocol fault.

22.4 Resilience patterns

- Bulkheads separate telemetry, state, command and evidence resource pools.
- Circuit breakers stop repeated calls to degraded dependencies while publishing explicit health state.
- Dead-letter storage preserves rejected events with reason and replay authorization requirements.
- Backpressure reduces rate according to priority class instead of uncontrolled queue growth.
- Graceful degradation selects a profile-defined safe mode, not an improvised local behavior.
- Unknown outcome blocks dependent high-risk actions until state or human review resolves uncertainty.

Requirement ID	Normative requirement	Verification
API-ERR-001	All transport-specific errors SHALL map to a canonical error code and safe effect.	Error mapping test.
API-ERR-002	Retryability SHALL be explicit and SHALL consider idempotency, authority, expiry and outcome uncertainty.	Retry decision test.
API-ERR-003	Unknown outcome SHALL NOT be automatically retried for non-idempotent or physical actions.	Unknown-outcome scenario.
API-ERR-004	Dead-letter records SHALL preserve original payload, reason, attempts, identity and evidence linkage.	Dead-letter audit.
API-ERR-005	Resource exhaustion SHALL trigger declared backpressure or safe degradation.	Load/fault test.

INTEGRATION ASSURANCE

23. Security and Privacy Requirements

Security controls are applied at identity, transport, schema, policy, runtime, update and evidence layers. The threat model assumes malicious or compromised peers, spoofed protocol components, replay, service-version downgrade, mapping manipulation, event-subscription exhaustion, credential theft, unauthorized federation and evidence suppression. Security design remains profile-specific where safety, privacy or mission sensitivity differ.

23.1 Security control families

Control family	Minimum coverage
Identity and authentication	Mutual authentication, workload identity, certificate rotation, revocation and ownership binding.
Authorization	Least privilege, scoped roles, delegated authority, separation of duties and deny-by-default.
Transport security	mTLS or profile equivalent, secure key management, channel binding and replay protection.
Input and schema security	Size, depth, type, semantic, decompression and media validation.
Runtime isolation	Process/container isolation, minimal privileges, network segmentation and sandboxed adapters.
Software supply chain	Signed releases, SBOM, provenance, vulnerability policy and secure update.
Availability	Rate limits, quotas, backpressure, event-subscription limits and resource reservations.
Evidence integrity	Append-only storage, hashes, signatures, trusted time and replicated manifests.
Privacy	Purpose limitation, minimization, field labels, retention, selective disclosure and access logging.
Federation security	Explicit partner trust, export policy, cross-domain identity and revocation propagation.

23.2 JAUS-specific threats

Threat	Scenario	Primary controls
Spoofed JAUS component	Attacker claims an allowed subsystem/node/component address.	Bind address to enrolled identity; detect collision and unexpected relocation.

Threat	Scenario	Primary controls
Malicious Discovery	Flood or deceptive service advertisement.	Rate limit, quarantine, profile and JSD validation.
Access Control abuse	Illicit preemption, stale ownership or priority manipulation.	Policy-bounded priority, lease monitoring, alerts and evidence.
Service downgrade	Select older or semantically weaker JSD/mapping.	Pinned compatibility range and downgrade policy.
Replay or duplicate	Old command/report treated as current.	Trusted time, message correlation and protocol-aware deduplication.
Event exhaustion	Excessive subscriptions or rates consume component capacity.	Quota, rate bounds, subscription approval and circuit breaker.
Mapping manipulation	Valid message translated into unsafe A^{3}O semantics.	Signed mapping artifacts, review and golden-vector tests.
Evidence suppression	Raw message or control record omitted.	Mandatory evidence path, offset monitoring and safe degradation.

23.3 Privacy and classification

- Each object carries classification and, where applicable, privacy purpose and retention labels.
- Biometric, patient, employee or sensitive location data is separated from general operational state whenever practical.
- A subscriber receives only the minimum fields necessary for its declared purpose and role.
- Cross-ecosystem federation requires explicit disclosure policy and may use derived or redacted state instead of raw telemetry.
- Deletion or retention requests are handled without destroying evidence that must legally or contractually remain, using restricted tombstones or legal-hold status.

23.4 Secure update and configuration

GOVERNED UPDATE PATH
Update proposal
→ signed artifact + SBOM + provenance
→ compatibility and policy evaluation
→ staged deployment / canary
→ health and evidence validation
→ promote or rollback
→ update identity / build attestation

Requirement ID	Normative requirement	Verification
API-SEC-001	All operational principals SHALL use authenticated identities and least-privilege authorization.	Access-control test.
API-SEC-002	Control-relevant messages SHALL include anti-replay and freshness protection appropriate to the transport.	Replay test.
API-SEC-003	Adapters and semantic mappings SHALL be signed, versioned and included in software provenance.	Supply-chain audit.
API-SEC-004	Event subscriptions SHALL be quota-controlled and shall declare maximum rate or change condition.	Exhaustion test.
API-SEC-005	Security-relevant revocation SHALL propagate to federated or partitioned nodes according to declared policy and evidence limitations.	Revocation scenario.
API-SEC-006	Sensitive data SHALL be classified, minimized, retained and disclosed according to explicit policy.	Privacy review.
API-SEC-007	Evidence-path failure SHALL be observable and may require safe degradation for control-critical operations.	Evidence failure test.

INTEGRATION ASSURANCE

24. Conformance Test Kit

The Conformance Test Kit is a versioned set of validators, protocol emulators, reference components, golden datasets, fault injectors and scenario runners. It produces a machine-readable report bound to the profile, implementation build, environment and evidence manifest. Passing an A^{3}O suite demonstrates conformance to the declared A^{3}O profile; it does not automatically represent certification by SAE or another external body.

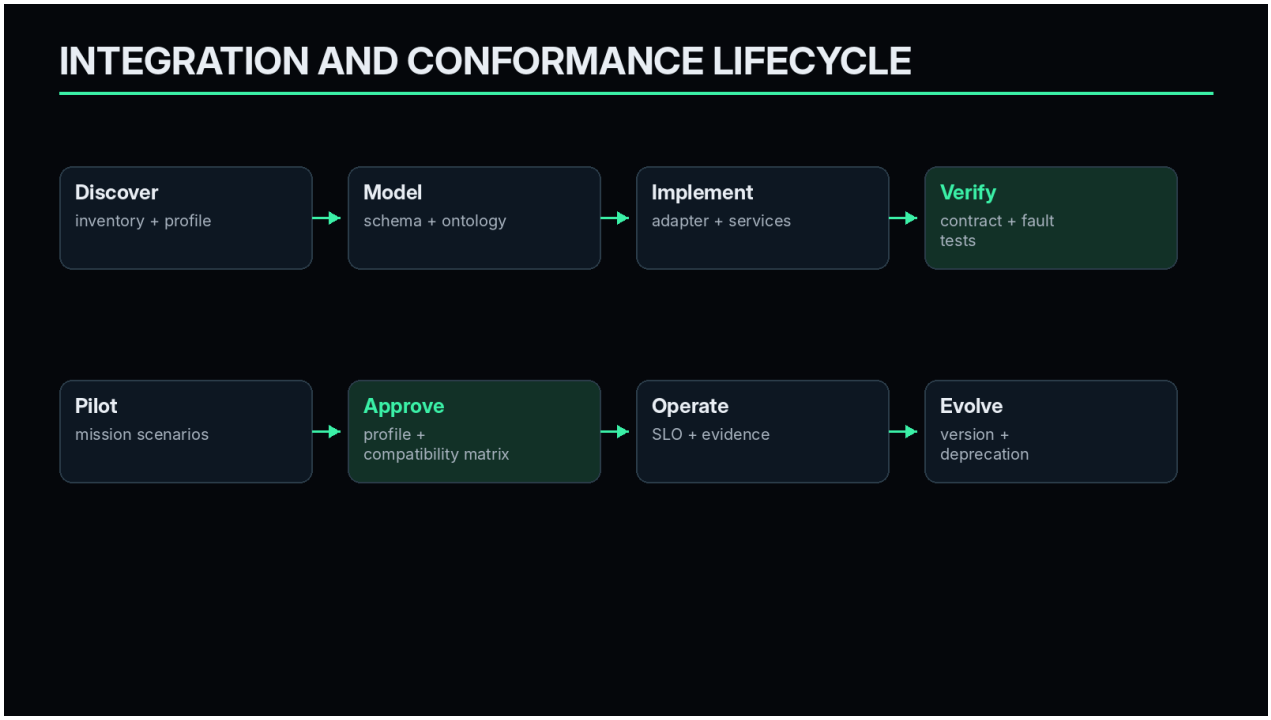

Figure 8 — Integration and conformance lifecycle.

24.1 Conformance levels

evel	Acceptance meaning
0 CONTRACT	Schemas, examples, identifiers and static artifact rules pass.
1 TRANSPORT	Authentication, delivery, ordering, timeout, retry and recovery pass.
2 SEMANTIC	Units, frames, lifecycle, protocol mapping and golden vectors pass.
3 GOVERNED	Identity, policy, authority, action and evidence controls pass.
4 SCENARIO	Industry-profile mission and fault scenarios pass.
JAUS-AWARE	JAUS entities and messages are recognized and preserved.
JAUS CORE	Target Core Service integration behavior passes.
SERVICE SET	Selected application service-set mappings pass.
GOVERNED JAUS	JAUS integration passes A^{3}O identity, policy, authority, state and evidence controls.
PROFILE	Partner/industry profile suite passes; external certification stated separately.

24.2 Test families

Family	Coverage
CT-01 Schema	Valid/invalid objects, required fields, bounds, extension and version behavior.
CT-02 Identity	Enrollment, credential rotation, revocation, quarantine and address binding.
CT-03 Transport	Loss, delay, duplicate, reordering, timeout, reconnect and backpressure.
CT-04 State	Revision, snapshot, delta gap, stale data and conflict retention.
CT-05 Mission	Plan revision, allocation, lease expiry, preemption and reassignment.
CT-06 Policy/authority	Allow, deny, obligation, approval, delegation and revocation propagation.
CT-07 Action	Idempotency, precondition, cancellation, feedback and outcome separation.
CT-08 Evidence	Completeness, integrity, selective disclosure and verification.

Family	Coverage
CT-09 Security	Spoofing, replay, downgrade, flooding, Sybil and mapping tamper.
CT-10 Offline	Partition, authority expiry, buffering, deduplication and reconciliation.
CT-11 JAUS	Discovery, JSD/version resolution, query/report/event, access control and lifecycle mapping.
CT-12 Performance	Latency, throughput, event rate, queue, CPU, memory, storage and error budget.
CT-13 Industry scenario	End-to-end mission with domain policy and evidence acceptance.

24.3 Machine-readable report

CONFORMANCEREPORT EXAMPLE

```
{
 "report_id": "conformance:01J...",
 "implementation": {"name":"partner-adapter","version":"2.3.1","digest":"sha256:..."},
 "profile": "urn:a3o:profile:logistics-amr:1.1.0",
 "environment": {"runtime":"a3o-edge-1.2","transport":"jaus-as5669"},
 "suite": {"id":"A3O-CTS-1.2","digest":"sha256:..."},
 "summary": {"passed":183,"failed":0,"skipped":4,"warnings":2},
 "levels": ["C0","C1","C2","C3","J0","J1","J2","J3"],
 "evidence_manifest": "manifest:01J...",
 "signed_by": "assessor:a3o-lab"
}
```

24.4 Acceptance rules

- All mandatory tests for the claimed profile and level pass.
- No unresolved safety-critical, authority-control, identity-binding or evidence-integrity defect remains.
- Skipped tests are justified by an explicit unsupported feature or environment limitation and reduce the claim accordingly.
- Performance is measured against the declared deployment profile, not a generic laboratory maximum.
- The report and test artifacts are signed or otherwise integrity-protected and reproducible.

Requirement ID	Normative requirement	Verification
API-CONF-001	Conformance claims SHALL identify profile, version, implementation build, environment, suite and evidence manifest.	Report schema test.
API-CONF-002	Unsupported or skipped mandatory features SHALL reduce the claimed level or profile scope.	Claim validation.
API-CONF-003	Fault and negative tests SHALL be required for policy, authority, action and evidence interfaces.	Suite coverage audit.
API-CONF-004	A^{3}O conformance SHALL NOT be represented as SAE certification or endorsement.	Publication review.
API-CONF-005	Test artifacts and results SHALL be reproducible from controlled versions and digests.	Reproduction test.

INTEGRATION ASSURANCE

25. Integration Lifecycle and Release Gates

Integration is managed as a controlled lifecycle from discovery through sustained operation. The process prevents a prototype adapter from silently becoming a production command path. Each gate has required artifacts, responsible owners, test evidence and rollback conditions.

Gate	Purpose	Required output
G0 Discover	Inventory asset, protocol, services, identities, hazards and operational owner.	Integration brief; protocol inventory; data classification.
G1 Model	Define canonical mappings, units, frames, lifecycle and unsupported semantics.	Mapping specification; schemas; example vectors.
G2 Implement	Build adapter, SDK integration, health, evidence and configuration.	Signed build; SBOM; deployment manifest.

Gate	Purpose	Required output
G3 Contract verify	Pass schema, version, transport and negative contract tests.	C0/C1 report.
G4 Semantic verify	Pass golden vectors, state/action semantics and protocol round trips.	C2 or J2 report.
G5 Governed verify	Pass identity, policy, authority, evidence and security tests.	C3/J3 report.
G6 Pilot	Run bounded mission scenarios with monitoring and rollback.	Pilot report; issues; evidence package.
G7 Approve	Publish profile compatibility and operational conditions.	Signed acceptance; compatibility matrix entry.
G8 Operate	Monitor SLO, evidence, security, versions and drift.	Operational dashboard; incident and change process.
G9 Evolve/retire	Migrate, deprecate, replace or retire without evidence loss.	Migration and retirement record.

25.1 Partner deliverables

- Integration manifest and named technical owner.
- Protocol/service inventory and licensed artifact declaration.
- Canonical mapping and unsupported-feature statement.
- OpenAPI/AsyncAPI/schema or SDK contract artifacts as applicable.
- Signed software build, SBOM, configuration and deployment requirements.
- Golden examples, negative examples and conformance tests.
- Threat model, fault behavior, safety assumptions and rollback plan.
- Operational SLO, support window and version/deprecation policy.

25.2 Change classification

Class	Examples	Minimum gate
LOW	Documentation, non-semantic examples or internal implementation with no external effect.	Automated tests and owner review.
MODERATE	Backward-compatible schema or optional capability addition.	Regression, profile and performance review.
HIGH	Semantic mapping, authority, safety, action or protocol behavior change.	Requalification and controlled pilot.
CRITICAL	Change affecting certified safety boundary, emergency behavior or strategic control.	Independent review, full scenario suite and executive approval.

Requirement ID	Normative requirement	Verification
API-INT-001	No integration SHALL enter production command paths before governed conformance and pilot acceptance.	Release audit.
API-INT-002	Every release SHALL include immutable artifact digests, SBOM, configuration and rollback plan.	Release package audit.
API-INT-003	Semantic, authority, safety or protocol-mapping changes SHALL trigger requalification.	Change-control test.
API-INT-004	Compatibility matrix entries SHALL state supported versions, profiles, limitations and expiration or review date.	Matrix review.
API-INT-005	Retirement SHALL preserve historical evidence, identity references and migration traceability.	Retirement audit.

INTEGRATION ASSURANCE

26. Worked Integration Examples

The examples below show how common contracts are composed. They are illustrative profile blueprints rather than certified operating procedures. Each production profile adds site, vehicle, machine, medical, aviation, maritime or defence-specific safety and compliance requirements.

26.1 Industrial Robot Cell

A production cell contains a robot controller, PLC, vision system, tool changer and quality station. The adapter publishes process and machine state, exposes bounded actions and preserves the local certified safety controller as the highest-priority boundary.

Key objects and interfaces

Object / interface	Role
TelemetryEnvelope	Joint state, process state, vibration, temperature and controller diagnostics.
EntityState	Robot, tool, workpiece and cell state.
CapabilityAdvertisement	Manipulation, payload, tooling and vision capabilities.
MissionPlan	Production order decomposed into process tasks.
PolicyDecision	Recipe, user role, maintenance and safety-zone checks.
ActionCommand	Start cycle, move to bounded pose, change tool or pause.
EvidencePackage	Recipe, program version, quality measurement and outcome.

Canonical flow

```
INDUSTRIAL ROBOT CELL FLOW
PLC/robot adapter
  → process telemetry
  → MachineState
  → workflow task
  → policy and safety
  → bounded ProcessCommand
  → quality evidence
```

JAUS / protocol integration notes

- JAUS Manipulator services may be used where semantics and implementation match the licensed service set.
- Vendor-specific PLC commands remain behind the adapter and map to canonical bounded actions.

Fault and degraded-mode tests

- Vision unavailable during a quality-critical step.
- Safety controller reports protective stop.
- Recipe or robot-program version changes during task.
- Adapter loses evidence storage.

Acceptance evidence

- No action bypasses local safety.
- Program and recipe versions appear in evidence.
- Outcome is based on process and quality state, not command ACK.

26.2 Warehouse AMR Fleet

A fleet manager coordinates heterogeneous AMRs from multiple vendors. Some expose JAUS mobility services, others use vendor APIs. A³O normalizes pose, energy, capability and task state and allocates delivery tasks under human-zone and battery-reserve policy.

Key objects and interfaces

Object / interface	Role

Object / interface	Role
CapabilityAdvertisement	Payload, speed, energy, route and partition capabilities.
SharedWorldState	Map, AMRs, humans, blocked zones and stations.
TaskAssignment	Pickup and delivery lease with backup node.
AuthorityLease	Mission-scoped ability to command mobility.
ActionCommand	Navigate, dock, load or unload.
CustodyEvidence	Cargo identity, handoff and delivery verification.

Canonical flow

```
WAREHOUSE AMR FLEET FLOW
capabilities and energy
  → shared map/state
  → distributed allocation
  → route action
  → delivery outcome
  → chain of custody
```

JAUS / protocol integration notes

- JAUS Mobility and Core Services are mapped natively for compatible platforms.
- JAUS Access Control ownership is acquired before dispatching mobility commands.
- JSD version and A³O mapping profile are retained in evidence.

Fault and degraded-mode tests

- Two fleet managers contend for control.
- Network partition occurs after pickup.
- AMR reports success but cargo sensor shows no load.
- Battery capability degrades after assignment.

Acceptance evidence

- No duplicate delivery after reconnect.
- Cargo outcome requires state and custody evidence.
- Authority and task leases expire safely.

26.3 Delivery UAV

A delivery UAS receives a route task, weather and geofence context, payload custody requirements and human handoff conditions. A³O remains the mission and evidence layer; flight control and aviation approvals remain platform and jurisdiction specific.

Key objects and interfaces

Object / interface	Role
TelemetryEnvelope	Navigation, energy, payload, weather and link state.
MissionIntent	Deliver payload under geofence, reserve and handoff constraints.
SafetyEnvelope	Geofence, altitude, weather and reserve limits.
ActionCommand	Mission upload or bounded navigation intent.
OutcomeAssessment	Delivery and handoff result.
EvidencePackage	Route, approvals, custody and exception history.

Canonical flow

```
DELIVERY UAV FLOW
navigation/payload/weather
  → MissionPlan
  → geofence and authority
  → flight task
  → handoff evidence
```

JAUS / protocol integration notes

- Compatible JAUS services can contribute pose, mobility or mission execution state where applicable.
- A³O does not claim flight-control certification through JAUS compatibility.

Fault and degraded-mode tests

- GNSS quality loss.
- Weather exceeds policy threshold.
- Command link partition.
- Handoff recipient identity mismatch.

Acceptance evidence

- Safe fallback is profile-defined.
- Delivery success requires custody verification.
- All approvals and changes are traceable.

26.4 Medical Delivery Robot

A hospital robot transports a sealed payload between authorized points. The integration minimizes patient and staff data, separates clinical identity from general fleet state and requires human escalation when custody or route conditions are uncertain.

Key objects and interfaces

Object / interface	Role
MissionIntent	Move sealed payload from source to destination.
PolicyDecision	Privacy, route, time, staff role and custody conditions.
TaskAssignment	Robot assignment with backup and lease.
ActionCommand	Navigate, dock, unlock under human confirmation.
EvidencePackage	Custody, temperature, access and delivery verification.

Canonical flow

```
MEDICAL DELIVERY ROBOT FLOW
cargo and destination identity
  → privacy policy
  → route/task
  → human escalation
  → custody and delivery verification
```

JAUS / protocol integration notes

- JAUS may provide interoperability for mobility or sensing, but medical-device and hospital compliance remain separate.

Fault and degraded-mode tests

- Recipient cannot authenticate.
- Payload temperature out of range.
- Privacy-sensitive field appears in general event stream.
- Route enters restricted clinical zone.

Acceptance evidence

- No patient data beyond declared purpose.

- Custody chain remains complete.
- Human escalation is recorded and time-bounded.

26.5 Environmental Survey Swarm

A heterogeneous swarm of aerial, ground and maritime nodes surveys an area, exchanges capabilities, allocates coverage and produces a provenance-rich dataset. Partitioned nodes continue only within delegated survey authority and reconcile coverage and evidence later.

Key objects and interfaces

Object / interface	Role
MissionPlan	Coverage zones, modalities, quality and termination.
CapabilityAdvertisement	Sensor, endurance, mobility and communications.
TaskAssignment	Zone and modality allocation.
TelemetryEnvelope	Multimodal observations with time, frame and provenance.
SharedWorldState	Coverage, gaps, hazards and peer state.
Dataset EvidencePackage	Acquisition lineage, calibration and gap report.

Canonical flow

```
ENVIRONMENTAL SURVEY SWARM FLOW
coverage plan
  → capability exchange
  → zone allocation
  → multimodal telemetry
  → dataset provenance
  → gap detection
```

JAUS / protocol integration notes

- JAUS sensing, mobility and autonomy services may be used where applicable.
- A³O swarm tasking remains an explicitly separate extension when no standard service applies.

Fault and degraded-mode tests

- Partition creates overlapping coverage.
- One peer advertises false capability.
- Clock quality degrades.
- Dataset fragment arrives after mission completion.

Acceptance evidence

- Conflicts and duplicates are preserved and resolved.
- Capability trust is verified.
- Dataset provenance identifies every source and transformation.

26.6 DroneDefender Security Profile

DroneDefender is the first A³O security and defence reference profile. RF, radar, EO/IR, acoustic and other sensors contribute observations and tracks. Policy and human authority govern bounded responses. The API design remains at architecture and evidence level; profile-specific operational and legal controls define permitted actions.

Key objects and interfaces

Object / interface	Role
TelemetryEnvelope	RF, radar, EO/IR, acoustic and system-health data.
EntityState	Detection, track and classified operational entity.
PolicyDecision	Rules, zones, confidence and human-approval conditions.
ActionCommand	Bounded profile-defined response intent.

Object / interface	Role
EvidencePackage	Sensor provenance, decisions, approvals, actions and outcome.

Canonical flow

```
DRONEDEFENDER SECURITY PROFILE FLOW
sensor observations
  → fusion and EntityState
  → risk assessment
  → policy and human authority
  → bounded response
  → incident evidence
```

JAUS / protocol integration notes

- JAUS-compatible sensors or platforms are integrated through the native domain.
- A³O-specific track fusion uses a separate extension namespace unless an applicable standard service is used.

Fault and degraded-mode tests

- Spoofed friendly node.
- Sensor disagreement or uncertain track.
- Access Control ownership conflict.
- Evidence recorder unavailable.

Acceptance evidence

- No automated expansion of response authority.
- Raw and fused evidence are correlated.
- Human decisions and denied actions remain auditable.

26.7 JAUS-Compatible Ground Robot

A third-party ground robot exposes Core and Mobility services. A³O discovers the hierarchy, binds JAUS addresses to the enrolled asset, validates declared JSD versions, subscribes to reports/events, acquires control according to policy and maps a mission task into platform-local actions.

Key objects and interfaces

Object / interface	Role
PeerCandidate	Candidate created from JAUS Discovery.
IdentityBinding	JAUS address to canonical node and asset.
CapabilityAdvertisement	Mobility, pose, velocity and mission capabilities.
AuthorityLeaseBinding	A³O authority correlated with JAUS Access Control ownership.
Canonical JAUS Envelope	Raw, decoded and mapped message context.
OutcomeAssessment	Result based on reports, state and evidence.

Canonical flow

```
JAUS-COMPATIBLE GROUND ROBOT FLOW
JAUS discovery
  → identity binding
  → JSD and service validation
  → event subscriptions
  → Access Control ownership
  → governed command
  → report/outcome correlation
```

JAUS / protocol integration notes

- AS5669 transport is handled by the native gateway.

- AS5710A Core Service behavior is integrated without bypassing ownership or lifecycle.
- The exact JSD and mapping profile are evidence-linked.

Fault and degraded-mode tests

- Unknown service version.
- Duplicate address from another asset.
- Preemption by a higher-priority controller.
- Late report after authority expiry.

Acceptance evidence

- Unknown versions fail closed.
- Address collision creates quarantine.
- Preemption is visible to mission and action state.
- Late reports do not create unauthorized new actions.

INTEGRATION ASSURANCE

27. Implementation Roadmap

The roadmap converts the API baseline into released machine-readable contracts, runtime services and partner tooling. Work is sequenced so that identity, evidence and governance are present before broad command integration.

Phase	Primary work	Exit result
Phase 0 — Contract baseline	Freeze canonical identifiers, core objects, error model and artifact governance.	OpenAPI/AsyncAPI skeletons; schemas; registry; requirement trace.
Phase 1 — Identity and ingress	Enrollment, credentials, gateway, audit and rate controls.	Verified node onboarding and quarantine.
Phase 2 — Data and state	Telemetry, state, history, subscriptions and evidence fragments.	Read-only multi-protocol integrations.
Phase 3 — JAUS Core runtime	AS5669 gateway, JSIDL registry, Discovery, Events, Management, Time, Liveness and Access Control integration.	J0/J1 reference implementation.
Phase 4 — Mission and action	Mission, policy, delegation, task, action, feedback and outcome.	Governed closed loop with evidence.
Phase 5 — Edge and swarm	Peer mesh, capability exchange, partition and reconciliation.	Partition-tolerant pilot.
Phase 6 — Industry profiles	Manufacturing, logistics, inspection, care/response and DroneDefender.	C4/J4 pilot profiles.
Phase 7 — Conformance lab	Reference components, fault injection, compatibility matrix and partner portal.	Repeatable ecosystem onboarding.

27.1 Minimum reference implementation

- API gateway and identity/enrollment service;
- schema/profile registry with immutable artifacts;
- telemetry ingestion, state store and event distribution;
- policy, delegation, human approval and authority service;
- action lifecycle and evidence recorder;
- JAUS transport gateway, JSIDL registry and Core Service bridges;
- simulator/reference nodes and conformance runner;
- partner SDK examples for at least one JAUS and one non-JAUS platform.

27.2 Release metrics

Metric	Release criterion
Contract coverage	100% public operations and events have machine-readable descriptions and examples.
Requirement trace	Every SHALL maps to at least one verification method.

Metric	Release criterion
Evidence completeness	All control-relevant scenario actions have full intent-to-outcome lineage.
Mapping quality	Golden-vector pass rate and zero unresolved safety-critical semantic ambiguity.
Performance	Profile-specific latency, throughput, event-rate and recovery budgets.
Security	No open critical identity, authority, command, evidence or supply-chain issue.
Partner repeatability	A new reference node completes onboarding using published kit without private code changes.

INTEGRATION ASSURANCE

28. Change and Traceability Matrix

The matrix below demonstrates that the R1.0 outline has been retained and expanded rather than replaced. The original twenty-one sections are preserved in Annex H, and each maps to one or more implementation-grade chapters in this R1.2 FULL MASTER.

R1.0 source section	R1.2 destination	Modernization
1 Scope and supported profiles	2 Scope, Roles and Supported Profiles; 21 Industry Profile SDK	Expanded roles, profiles, out-of-scope and requirements.
2 Architecture and trust boundaries	3 API Architecture and Trust Boundaries	Added authoritative-service model, gateway duties and boundary catalogue.
3 Identity and enrollment	5 Identity, Enrollment and Trust Bootstrap	Added identity taxonomy, lifecycle, endpoints, examples and requirements.
4 Canonical IDs and versioning	6 Identifiers; 7 Lifecycle and Deprecation	Added identifier catalogue, semantic versioning and migration rules.
5 Transport profiles	8 Transport Profiles, QoS and Delivery Semantics	Added delivery classes, required metadata and resilience requirements.
6 Universal Telemetry API	10 Universal Telemetry API	Expanded envelope, modalities, operations, validation and routing.
7 State and World Model API	11 State and World Model API	Expanded objects, history, conflicts, snapshots and subscriptions.
8 Peer Discovery and Mesh API	12 Peer Discovery and Mesh API	Added candidate/admission split, topology, anti-Sybil and quarantine.
9 Capability Exchange API	13 Capability Exchange API	Added definitions, advertisements, matching and evidence.
10 Mission and Task APIs	14 Mission and Task APIs	Added lifecycle, allocation, JAUS Mission Spooler layering and requirements.
11 Policy, Delegation and Human Authority API	15 Policy, Delegation and Human Authority API	Added decision objects, precedence, separation of duties and endpoints.
12 Action and Feedback API	16 Action and Feedback API	Added action lifecycle, idempotency, cancellation and outcome separation.
13 Evidence and Assurance API	17 Evidence and Assurance API	Added graph, packages, verification and selective disclosure.
14 Twin, Simulation and Replay API	18 Twin, Simulation and Replay API	Added namespace isolation, promotion rules and reproducibility.
15 Health and Observability	19 Health, Observability and SLO API	Added health objects, status model, SLO and evidence-offset monitoring.
16 Offline, Partition and Reconciliation	20 Offline, Partition and Reconciliation	Added state machine, objects, authority and evidence behavior.
17 Industry Profile SDK	21 Industry Profile SDK	Added manifest, composition, protections and initial families.
18 Errors and Resilience	22 Errors, Resilience and Safe Failure	Added canonical error object, domains, retry and compensation.
19 Security Requirements	23 Security and Privacy Requirements	Added control families, JAUS threats, privacy and secure update.
20 Conformance Test Kit	24 Conformance Test Kit; 25 Integration Lifecycle	Added levels, test families, report and release gates.
21 Worked integration examples	26 Worked Integration Examples	Expanded six original examples and added a native JAUS robot example.

28.1 Architecture alignment

Architecture series	API implementation trace
A3O-1000 / 1200 / 1300 / 1400	Chapters 4–9, 21 and Annexes A–E.
A3O-2000 / 2100	Chapters 3, 8–12, 19 and 20.
A3O-2200 / 2300	Chapters 14–16 and AI authority constraints.
A3O-2400 / 2500	Chapters 12, 20, 22–24 and evidence/safety rules.
A3O-3000	JAUS runtime, API services and component ownership throughout.
A3O-4000	Chapter 21 and worked industry examples.
A3O-5000	Chapters 24–25 and Annex C requirements.
A3O-6000 / 7000 / 8000	Governed learning boundaries, federation, partition and threat controls.
A3O-9000	OpenAPI, AsyncAPI, schemas, SDKs, examples and conformance assets.

PART IV

Annexes, Requirements and Preserved Source

Canonical object and error catalogues, consolidated requirements, documentation profiles, standards, glossary, integrity record and preserved R1.0 content.

PART IV · REFERENCE ANNEXES

Annex A — Canonical Object Catalogue

This catalogue is a controlled index of the principal public objects defined in this guide. Detailed schemas remain authoritative and are released as separate machine-readable artifacts.

Domain	Objects	Purpose
Identity	EnrollmentRequest, IdentityBinding, CredentialState, TrustState	Identity and trust bootstrap.
Telemetry	TelemetryEnvelope, QualityContext, UncertaintyContext, ProvenanceContext, MediaReference	Observations and source lineage.
State	EntityState, ContextState, SharedWorldState, StateDelta, StateSnapshot, ConflictRecord	Operational state and history.
Mesh	PeerCandidate, PeerMembership, MeshLink, TopologyState, PartitionNotice	Peer admission and topology.
Capability	CapabilityDefinition, CapabilityAdvertisement, CapabilityState, CapabilityRequirement, CapabilityMatch	Operational capabilities and selection.
Mission	MissionIntent, MissionPlan, TaskDefinition, TaskAssignment, TaskProgress, MissionOutcome	Intent and coordination.
Policy / authority	PolicyEvaluation, PolicyDecision, DelegationGrant, HumanApproval, SafetyEnvelope, EmergencyAuthority	Governed authorization.
Action	ActionCommand, ActionFeedback, ActionProgress, CancellationRequest, OutcomeAssessment	Execution lifecycle.
Evidence	EvidenceFragment, EvidenceLink, EvidenceManifest, EvidencePackage, AssuranceClaim, VerificationResult	Audit and assurance.
Twin	TwinEntity, EnvironmentModel, Scenario, Forecast, ReplaySession	Simulation and replay.
Observability	Heartbeat, HealthState, ResourceState, LinkState, ServiceSLO, EvidenceOffset	Health and SLO.
Offline	SyncCursor, BufferedOperation, ConflictSet, AuthorityHistory, ReconciliationReport	Partition and merge.
Error	CanonicalError, ProblemDetail, RetryDirective, DeadLetterRecord	Failure representation.
Profile	ProfileManifest, ProtocolBinding, MappingProfile, ConformanceReport, CompatibilityRecord	Integration packaging.
JAUS	CanonicalJausEnvelope, JausIdentityBinding, JsdRegistryEntry, AccessControlBinding, JausSubscription	Native JAUS integration.

A.1 Common metadata

Field family	Meaning
id	Canonical stable identifier.
revision	Object revision or conflict metadata.
schema_id / schema_version	Machine-readable contract identity.
ecosystem_id	Operational boundary.
mission_id	Mission context when applicable.
source_identity	Producer or author.
created_at / observed_at / valid_until	Time semantics.
classification / privacy	Disclosure and retention.
correlation_id / causation_id	Trace and causal chain.
provenance / evidence_id	Lineage and assurance.

REFERENCE ANNEX

Annex B — Canonical Error and Reason Codes

Code	Meaning	Retry	Safe effect
A3O.CONTRACT.INVALID_SCHEMA	Payload violates the declared schema.	No	Rejected before domain processing.
A3O.CONTRACT.UNSUPPORTED_VERSION	Version is outside the supported range.	Conditional	Negotiate or migrate.
A3O.IDENTITY.UNKNOWN	Principal or node identity is unknown.	No	Reject or enroll.
A3O.IDENTITY.REVOKED	Identity or credential has been revoked.	No	Terminate session and quarantine.
A3O.TRUST.QUARANTINED	Node is isolated by trust policy.	No	Diagnostics/evidence only.
A3O.POLICY.DENIED	Applicable policy denied the request.	No	Return reasons and obligations.
A3O.POLICY.APPROVAL_REQUIRED	Human or independent approval required.	Conditional	Create approval workflow.
A3O.AUTHORITY.MISSING	No valid authority context.	Conditional	Acquire bounded authority.
A3O.AUTHORITY.LEASE_EXPIRED	Authority expired before use.	Conditional	Re-evaluate and obtain new lease.
A3O.SAFETY.ENVELOPE_VIOLATION	Requested action violates safety bounds.	No	Reject and alert.
A3O.STATE.STALE	Required state is too old.	Conditional	Refresh state.
A3O.STATE.CONFLICT	Authoritative state has unresolved branches.	Conditional	Resolve or use approved degraded policy.
A3O.CAPABILITY.UNAVAILABLE	Capability is absent, expired or degraded.	Conditional	Reallocate or wait.
A3O.TRANSPORT.TIMEOUT	Transport deadline expired.	Conditional	Depends on idempotency and outcome.
A3O.PROTOCOL.JAUS_UNKNOWN_SERVICE	JAUS service/version is unknown.	No	Quarantine mapping and review JSD.
A3O.PROTOCOL.JAUS_CONTROL_NOT_OWNED	Required JAUS control ownership is absent.	Conditional	Acquire control according to policy.
A3O.EXECUTION.REJECTED	Target rejected before execution.	Conditional	Inspect target reason.
A3O.EXECUTION.FAILED	Execution ended unsuccessfully.	Conditional	Compensation or reassignment.
A3O.OUTCOME.UNKNOWN	Operational result cannot be established.	No automatic retry	Hold dependent actions and investigate.
A3O.EVIDENCE.GAP	Required evidence fragment or sequence is missing.	Conditional	Recover or degrade safely.
A3O.RESOURCE.RATE_LIMITED	Rate or quota exceeded.	Yes after delay	Apply retry-after and backpressure.
A3O.OFFLINE.AUTHORITY_EXPIRED	Offline delegated authority expired.	No	Enter profile safe fallback.
A3O.RECONCILIATION.CONFLICT	Reconnect produced unresolved conflict.	No automatic retry	Policy or human resolution.

REFERENCE ANNEX

Annex C — Consolidated Normative Requirements

This annex consolidates 133 normative API requirements from the body of the guide. The requirements remain normative in their original chapters; this annex supports traceability, test planning and release review.

Requirement ID	Normative requirement	Primary verification
API-FND-001	The implementation SHALL separate observation, decision, authorization and execution interfaces.	Architecture review and command-path test.
API-FND-002	Every control-relevant object SHALL carry ecosystem, mission, identity and correlation context where applicable.	Schema validation.
API-FND-003	The API SHALL support JAUS and non-JAUS endpoints under the same governance model.	Mixed-protocol integration scenario.
API-FND-004	The API SHALL distinguish transport success, execution progress and operational outcome.	Action lifecycle test.
API-FND-005	Protocol translation SHALL NOT expand source authority or permitted operating domain.	Negative authority test.
API-SCP-001	Every integration SHALL declare one or more supported profiles and an explicit conformance target.	Profile manifest review.
API-SCP-002	An adapter SHALL declare the source protocol, semantic mapping version and unsupported source features.	Adapter manifest validation.
API-SCP-003	Industry profiles SHALL identify external certification and compliance obligations that remain outside A^{3}O interoperability conformance.	Profile governance review.
API-ARC-001	Public APIs SHALL terminate at a governed ingress boundary and SHALL NOT directly address physical actuators.	Architecture and penetration test.
API-ARC-002	Every authoritative object SHALL declare a single owning service and a revision or conflict-detection mechanism.	Schema and ownership registry review.
API-ARC-003	Read projections SHALL reject state-changing operations.	Negative endpoint test.
API-ARC-004	Event delivery SHALL NOT be interpreted as command or authority delegation.	Policy test.
API-ARC-005	The gateway SHALL preserve canonical reason codes when translating errors to transport-specific responses.	Error mapping test.
API-ART-001	Every published endpoint or event SHALL trace to a versioned machine-readable contract.	Artifact coverage report.
API-ART-002	Profile manifests SHALL bind schemas, examples, policies and tests by immutable digest.	Manifest verification.
API-ART-003	Vendor extensions SHALL NOT redefine canonical field semantics.	Schema and semantic review.
API-ART-004	Licensed JAUS artifacts SHALL be stored and distributed according to their applicable license terms.	Repository access audit.
API-ART-005	A^{3}O-specific JAUS services SHALL use a distinct namespace and explicit extension status.	JSIDL registry validation.
API-ID-001	Protocol addresses SHALL be bound to a canonical A^{3}O identity before authoritative state or command use.	Enrollment and negative unbound-address test.
API-ID-002	Credential rotation SHALL NOT change the canonical node identity.	Rotation test.
API-ID-003	Revocation SHALL propagate to active sessions, authority leases and command paths within the declared security budget.	Revocation latency test.
API-ID-004	A quarantined node SHALL be limited to explicitly allowed remediation and evidence operations.	Quarantine policy test.
API-ID-005	Enrollment records SHALL retain the decision, evidence, approver and policy version.	Evidence audit.
API-IDV-001	Canonical identifiers SHALL be opaque and SHALL NOT encode mutable authority or network location.	Identifier lint.
API-IDV-002	Schema IDs SHALL be immutable and content-addressable or protected by a published digest.	Registry test.
API-IDV-003	A breaking semantic change SHALL increment the major version even when the JSON shape is unchanged.	Semantic review.
API-IDV-004	JAUS service version and A^{3}O mapping profile version SHALL be recorded separately.	Evidence inspection.
API-IDV-005	Aliases SHALL resolve to one canonical identifier or return an explicit ambiguity error.	Alias resolution test.
API-LCM-001	Every released contract SHALL have a lifecycle state, owner and support policy.	Registry audit.

Requirement ID	Normative requirement	Primary verification
API-LCM-002	Deprecation SHALL include a machine-readable replacement and effective date.	Artifact validation.
API-LCM-003	Consumers SHALL be able to discover supported versions and profile ranges before mission execution.	Negotiation test.
API-LCM-004	A retired control operation SHALL fail closed with a canonical migration reason.	Retirement test.
API-LCM-005	Evidence SHALL preserve the exact API, schema, profile, adapter and mapping versions used.	Evidence audit.
API-TRN-001	Transport profiles SHALL NOT redefine canonical object semantics.	Cross-transport golden-vector test.
API-TRN-002	Control-relevant requests SHALL carry an idempotency key or protocol-equivalent duplicate-control mechanism.	Duplicate request test.
API-TRN-003	Every stream SHALL define ordering, gap detection, replay and backpressure behavior.	Stream resilience test.
API-TRN-004	A message received after expiry SHALL NOT create a new authoritative state transition or action.	Expiry test.
API-TRN-005	Transport-specific status codes SHALL map to canonical A^{3}O reason codes.	Error translation test.
API-JAUS-001	The gateway SHALL preserve original JAUS source and destination addresses and the unmodified received payload.	Raw-message evidence test.
API-JAUS-002	Every decoded message SHALL reference the exact JSD/service definition and version used.	Registry and evidence test.
API-JAUS-003	Discovery SHALL create candidate capability records and SHALL NOT grant trust or control authority.	Discovery admission test.
API-JAUS-004	A^{3}O SHALL respect JAUS Access Control ownership and preemption behavior.	Contention and preemption scenario.
API-JAUS-005	A^{3}O SHALL fail closed when service version or semantic mapping is unknown or incompatible.	Unknown-version negative test.
API-JAUS-006	A^{3}O extensions SHALL use separate namespaces and SHALL NOT be represented as SAE-standard services.	Registry lint.
API-JAUS-007	Mission-level authority SHALL NOT bypass local safety or JAUS control ownership.	End-to-end command-path test.
API-JAUS-008	Query, report, event and command correlations SHALL be retained in the Evidence Graph.	Evidence completeness test.
API-TEL-001	Telemetry SHALL carry semantic type, timestamp, source identity, schema version and provenance.	Schema validation.
API-TEL-002	Unit and reference-frame conversions SHALL be explicit, versioned and auditable.	Golden-vector conversion test.
API-TEL-003	Stale or expired telemetry SHALL NOT update authoritative live state unless the profile explicitly permits late correction.	Freshness test.
API-TEL-004	Batch ingestion SHALL report item-level acceptance and SHALL NOT hide partial failure.	Batch partial-failure test.
API-TEL-005	Raw source data SHALL be retained or content-addressed when required by the profile evidence policy.	Evidence retention test.
API-TEL-006	Unknown semantic types SHALL be isolated from policy-critical processing.	Unknown-type negative test.
API-STA-001	Every state object SHALL include authoritative owner, revision, observation time and validity or age semantics.	Schema audit.
API-STA-002	Consumers SHALL detect and recover from delta gaps before using state for control.	Gap/recovery test.
API-STA-003	Concurrent conflicting updates SHALL be retained as conflict branches until resolved.	Partition merge test.
API-STA-004	A snapshot SHALL identify the cutoff time and exact object revisions included.	Snapshot consistency test.
API-STA-005	Historical replay SHALL remain isolated from LIVE namespaces and command paths.	Replay isolation test.
API-MSH-001	Peer discovery SHALL NOT create mission membership or trust.	Candidate admission test.
API-MSH-002	Membership SHALL be mission-scoped, role-scoped, time-bounded and revocable.	Membership lifecycle test.
API-MSH-003	Topology events SHALL identify revision, source and affected members or links.	Event schema test.
API-MSH-004	A quarantined peer SHALL be excluded from task allocation and command coordination.	Quarantine scenario.
API-MSH-005	The mesh SHALL detect and report partitions without expanding peer authority.	Partition scenario.
API-CAP-001	Capability advertisements SHALL be signed or authenticated, versioned and time-bounded.	Advertisement validation.
API-CAP-002	Expired or withdrawn capabilities SHALL NOT be used for new task allocation.	Expiry test.

Requirement ID	Normative requirement	Primary verification
API-CAP-003	Hard safety, trust and profile constraints SHALL be applied before optimization scoring.	Matching test.
API-CAP-004	Capability match results SHALL record excluded candidates and reasons.	Evidence audit.
API-CAP-005	A capability advertisement SHALL NOT grant command authority.	Authority separation test.
API-MIS-001	MissionIntent SHALL define objective, scope, authority, success and termination criteria.	Schema test.
API-MIS-002	Every plan revision SHALL preserve traceability to the intent, assumptions and replaced plan.	Replanning evidence test.
API-MIS-003	TaskAssignment SHALL be time-bounded and SHALL reference current capability and authority context.	Lease test.
API-MIS-004	Allocation results SHALL record candidates, exclusions, scoring and decision authority.	Allocation evidence test.
API-MIS-005	Platform-local JAUS mission execution SHALL remain subordinate to the ecosystem mission intent and policy.	JAUS mission-spool scenario.
API-MIS-006	Task acceptance SHALL NOT imply permission to execute actions outside the assignment scope.	Scope enforcement test.
API-POL-001	Every decision SHALL include policy versions, result, reasons, obligations and expiry or invalidation conditions.	Decision schema test.
API-POL-002	Delegated authority SHALL be no broader or longer-lived than the grantor authority.	Delegation containment test.
API-POL-003	Human approval SHALL bind identity, qualification, scope, context and expiry.	Approval evidence test.
API-POL-004	AI services SHALL NOT create or extend authority grants.	Negative AI authority test.
API-POL-005	Revocation SHALL invalidate dependent command paths within the declared budget.	Revocation propagation test.
API-POL-006	Policy uncertainty or unavailable policy-critical data SHALL fail closed or enter an explicitly approved safe mode.	Policy dependency test.
API-ACT-001	ActionCommand SHALL include target, canonical action, bounded parameters, authority, policy, safety, idempotency and expiry context.	Schema test.
API-ACT-002	The execution gateway SHALL revalidate authority and safety immediately before protocol encoding.	Time-of-use validation test.
API-ACT-003	Duplicate action submissions with the same idempotency key SHALL NOT produce duplicate execution.	Duplicate test.
API-ACT-004	Transport or service acknowledgement SHALL NOT be reported as operational success.	Outcome separation test.
API-ACT-005	Cancellation SHALL define supported semantics and safe fallback when the target cannot cancel.	Cancellation scenario.
API-ACT-006	Protocol mapping version and raw command/feedback exchange SHALL be retained in evidence.	Evidence audit.
API-EVD-001	Evidence fragments SHALL be append-only and integrity-protected.	Tamper test.
API-EVD-002	Control-relevant actions SHALL have traceable links from intent through outcome.	Evidence completeness test.
API-EVD-003	Selective disclosure SHALL preserve source digest, transformation record and policy.	Disclosure verification.
API-EVD-004	Evidence export SHALL enforce classification, privacy and retention policy.	Export policy test.
API-EVD-005	Package verification SHALL be machine-readable and identify missing, invalid or unverifiable elements.	Verification result test.
API-TWN-001	LIVE, SIMULATION, FORECAST, REPLAY and TRAINING namespaces SHALL be distinguishable in identity, routing and evidence.	Namespace isolation test.
API-TWN-002	Simulation output SHALL NOT directly produce a live ActionCommand.	Negative promotion test.
API-TWN-003	Scenarios SHALL record model versions, input manifests, random seeds and execution environment.	Reproducibility audit.
API-TWN-004	Forecasts SHALL include generation time, horizon, assumptions and uncertainty.	Schema test.
API-TWN-005	Replay channels SHALL be isolated from live subscribers and command routes.	Replay isolation test.
API-OBS-001	Heartbeat SHALL NOT be interpreted as readiness, trust or safety.	Health logic test.
API-OBS-002	HealthState SHALL identify degraded reasons, affected capabilities and dependency status.	Schema test.
API-OBS-003	Control services SHALL enforce readiness and safety status at time of use.	Action admission test.

Requirement ID	Normative requirement	Primary verification
API-OBS-004	Observability exports SHALL enforce classification and least privilege.	Access-control test.
API-OBS-005	Evidence pipeline lag SHALL be observable and capable of triggering safe degradation.	Evidence-offset fault test.
API-OFF-001	Connectivity loss SHALL NOT expand authority or operating domain.	Partition authority test.
API-OFF-002	Offline actions SHALL reference the local authority, policy version and trusted-time state used.	Evidence audit.
API-OFF-003	Reconciliation SHALL preserve concurrent conflicts and SHALL NOT use last-write-wins for policy-critical state unless the profile explicitly justifies it.	Conflict merge test.
API-OFF-004	Completed offline actions SHALL be deduplicated during reconnect.	Reconnect duplicate test.
API-OFF-005	RESTORED state SHALL require a completed ReconciliationReport and approved resume mode.	State transition test.
API-OFF-006	Evidence-buffer exhaustion SHALL trigger a declared safe degradation or stop rule.	Storage exhaustion test.
API-PRF-001	Every profile SHALL declare compatibility ranges, external obligations, owner and lifecycle state.	Manifest validation.
API-PRF-002	Profiles SHALL NOT override core identity, authority, action-outcome or evidence-integrity semantics.	Semantic protection test.
API-PRF-003	Every operationally significant extension SHALL include valid, invalid and fault examples.	Artifact completeness test.
API-PRF-004	Protocol mappings SHALL include round-trip and semantic-equivalence tests.	Mapping conformance test.
API-PRF-005	A profile SHALL state which certifications or regulatory approvals remain outside A^{3}O conformance.	Governance review.
API-ERR-001	All transport-specific errors SHALL map to a canonical error code and safe effect.	Error mapping test.
API-ERR-002	Retryability SHALL be explicit and SHALL consider idempotency, authority, expiry and outcome uncertainty.	Retry decision test.
API-ERR-003	Unknown outcome SHALL NOT be automatically retried for non-idempotent or physical actions.	Unknown-outcome scenario.
API-ERR-004	Dead-letter records SHALL preserve original payload, reason, attempts, identity and evidence linkage.	Dead-letter audit.
API-ERR-005	Resource exhaustion SHALL trigger declared backpressure or safe degradation.	Load/fault test.
API-SEC-001	All operational principals SHALL use authenticated identities and least-privilege authorization.	Access-control test.
API-SEC-002	Control-relevant messages SHALL include anti-replay and freshness protection appropriate to the transport.	Replay test.
API-SEC-003	Adapters and semantic mappings SHALL be signed, versioned and included in software provenance.	Supply-chain audit.
API-SEC-004	Event subscriptions SHALL be quota-controlled and shall declare maximum rate or change condition.	Exhaustion test.
API-SEC-005	Security-relevant revocation SHALL propagate to federated or partitioned nodes according to declared policy and evidence limitations.	Revocation scenario.
API-SEC-006	Sensitive data SHALL be classified, minimized, retained and disclosed according to explicit policy.	Privacy review.
API-SEC-007	Evidence-path failure SHALL be observable and may require safe degradation for control-critical operations.	Evidence failure test.
API-CONF-001	Conformance claims SHALL identify profile, version, implementation build, environment, suite and evidence manifest.	Report schema test.
API-CONF-002	Unsupported or skipped mandatory features SHALL reduce the claimed level or profile scope.	Claim validation.
API-CONF-003	Fault and negative tests SHALL be required for policy, authority, action and evidence interfaces.	Suite coverage audit.
API-CONF-004	A^{3}O conformance SHALL NOT be represented as SAE certification or endorsement.	Publication review.
API-CONF-005	Test artifacts and results SHALL be reproducible from controlled versions and digests.	Reproduction test.
API-INT-001	No integration SHALL enter production command paths before governed conformance and pilot acceptance.	Release audit.

Requirement ID	Normative requirement	Primary verification
API-INT-002	Every release SHALL include immutable artifact digests, SBOM, configuration and rollback plan.	Release package audit.
API-INT-003	Semantic, authority, safety or protocol-mapping changes SHALL trigger requalification.	Change-control test.
API-INT-004	Compatibility matrix entries SHALL state supported versions, profiles, limitations and expiration or review date.	Matrix review.
API-INT-005	Retirement SHALL preserve historical evidence, identity references and migration traceability.	Retirement audit.

REFERENCE ANNEX

Annex D — OpenAPI and AsyncAPI Documentation Profiles

The examples are intentionally partial. Released artifacts shall include the complete operation set, reusable schemas, security schemes, errors, examples and profile extensions. OpenAPI 3.2.0 is the baseline for HTTP documentation. AsyncAPI 3.1.0 is the baseline for event-driven documentation.

D.1 OpenAPI profile

PARTIAL OPENAPI EXAMPLE

```yaml
openapi: 3.2.0
info:
  title: A3O Mission and Action API
  version: 1.2.0
  x-a3o-document-id: A3O-API-JAUS-1200
servers:
  - url: https://api.example.a3o/v1
security:
  - mutualTLS: []
  - oauth2: [mission.control]
paths:
  /actions:
    post:
      operationId: createAction
      x-a3o-authority-required: true
      x-a3o-evidence-required: true
      requestBody:
        required: true
        content:
          application/vnd.a3o.action-command+json:
            schema:
              $ref: '#/components/schemas/ActionCommand'
      responses:
        '202':
          description: Governed action resource created
        '403':
          $ref: '#/components/responses/AuthorityDenied'
        '409':
          $ref: '#/components/responses/StateOrControlConflict'
        '422':
          $ref: '#/components/responses/SafetyOrSemanticFailure'
```

D.2 AsyncAPI profile

PARTIAL ASYNCAPI EXAMPLE

```yaml
asyncapi: 3.1.0
info:
  title: A3O State and Action Events
  version: 1.2.0
  x-a3o-profile: urn:a3o:profile:core:1.2.0
channels:
  actionFeedback:
    address: a3o/{ecosystemId}/actions/{actionId}/feedback
    messages:
      actionFeedback:
        $ref: '#/components/messages/ActionFeedback'
operations:
  receiveActionFeedback:
    action: receive
    channel:
      $ref: '#/channels/actionFeedback'
    x-a3o-delivery-class: D3_ORDERED_STREAM
    x-a3o-resume-cursor: required
    x-a3o-classification: operational
components:
  messages:
    ActionFeedback:
      headers:
        $ref: '#/components/schemas/EventHeaders'
      payload:
        $ref: 'schemas/ActionFeedback.json'
```

D.3 A³O extension fields

Extension	Meaning
x-a3o-authority-required	Operation requires a valid authority context.
x-a3o-evidence-required	Ingress, decision and outcome evidence is mandatory.
x-a3o-delivery-class	D0–D4 delivery semantics.
x-a3o-profile	Applicable profile identity and range.
x-a3o-classification	Minimum classification or privacy handling.
x-a3o-idempotency	Required, optional or prohibited.
x-a3o-safe-effect	Declared behavior when the operation fails.
x-a3o-jaus-mapping	Reference to controlled JSD and mapping profile.
x-a3o-conformance-tests	References required tests and golden vectors.

REFERENCE ANNEX

Annex E — SAE JAUS Standards Baseline

The following baseline is limited to public title, status and scope information verified from SAE Mobilus pages on 17 July 2026. Implementation teams shall obtain and use the licensed standards and associated data sets needed for the declared profile.

Reference	Title	Status	DOI
AIR5665C	Architecture Framework for Unmanned Systems	Stabilized May 2024	10.4271/AIR5665C
AS5669A	JAUS / SDP Transport Specification	Reaffirmed Apr 2019	10.4271/AS5669A
AS5684B	JAUS Service Interface Definition Language	Reaffirmed Apr 2025	10.4271/AS5684B
AS5710A	JAUS Core Service Set	Reaffirmed Apr 2015	10.4271/AS5710A
AS6009A	JAUS Mobility Service Set	Revised Oct 2023	10.4271/AS6009A
AS6057A	JAUS Manipulator Service Set	Reaffirmed Apr 2025	10.4271/AS6057A
AS6060A	JAUS Environment Sensing Service Set	Reaffirmed May 2026	10.4271/AS6060A
AS6062A	JAUS Mission Spooling Service Set	Reaffirmed Mar 2024	10.4271/AS6062A
AS6091	JAUS Unmanned Ground Vehicle Service Set	Reaffirmed Apr 2025	10.4271/AS6091
AS6111	JAUS Unmanned Maritime Vehicle Service Set	Issued Sep 2023	10.4271/AS6111
AS8024	JAUS Autonomous Capabilities Service Set	Reaffirmed Apr 2025	10.4271/AS8024
ARP6227	JAUS Messaging over OMG DDS	Reaffirmed Mar 2024	10.4271/ARP6227

E.1 Public specification references

Reference	Location	Use
OpenAPI Specification 3.2.0	spec.openapis.org/oas/v3.2.0.html	Published 19 Sep 2025.
AsyncAPI Specification 3.1.0	asyncapi.com/docs/reference/specification/v3.1.0	Event-driven API description baseline.
ROS 2 middleware documentation	docs.ros.org	Used only for adapter and middleware context; A³O identity remains separate.
SAE Mobilus JAUS pages	saemobilus.sae.org/standards	Public scope and status reference; licensed standards required for implementation.

REFERENCE ANNEX

Annex F — Glossary

Term	Definition
Action intent	Canonical request to perform a bounded operational action before protocol encoding.
Authority lease	Time- and scope-bounded permission for a principal to act.
Canonical object	Protocol-neutral A³O data contract with defined semantics and lifecycle.
Capability	Operationally usable function with limits, conditions and current availability.
Conformance profile	Declared set of contracts, mappings, policies, tests and supported behavior.
Evidence Graph	Linked records connecting observations, decisions, commands, feedback and outcomes.
Governed action	Action that passed identity, policy, authority, safety and protocol-control checks.
JAUS address	Subsystem/node/component protocol identity; not a global A³O trust identity.
JSD	JAUS Service Definition expressed according to JSIDL.
JSIDL	JAUS Service Interface Definition Language defined by SAE AS5684B.
ODD	Operational Design Domain or explicit operating conditions and limits.
Outcome assessment	A³O conclusion about operational result, distinct from acknowledgement.
Profile manifest	Machine-readable composition of schemas, mappings, policies and tests.
Semantic mapping	Versioned transformation preserving meaning between protocol and canonical contracts.
Trust state	Operational status of identity and supporting evidence, such as VERIFIED or QUARANTINED.

REFERENCE ANNEX

Annex G — Release Integrity Record

Integrity item	Value
Source guide	A3O_API_Integration_Guide_R1.0_EN(1).docx
Source word count	912
Source paragraphs	76
Source tables	8
Source non-empty text blocks	168
Source SHA-256	6668f2c1a91110bc4c1e50ddc65ccc39acc283a116f4cfd09ec83c8ae90e7619
R1.2 normative requirements	133
R1.2 current word count before preserved source	17129
Architecture baseline	A3O_Architecture_Baseline_R1.2_FULL_MASTER_JAUS_Integrated_EN.docx
Whitepaper baseline	A3O_Whitepaper_R1.2_FULL_MASTER_JAUS_Integrated_EN.docx

G.1 Claim discipline

- This document is an A^{3}O engineering candidate and not an SAE publication.
- JAUS-compatible and JAUS-aligned language denotes a target implementation profile until conformance evidence is complete.
- No external certification, regulatory approval or platform installed-base quantity is claimed by this guide.
- Examples are illustrative and do not reproduce licensed JAUS message definitions.
- The source R1.0 guide is preserved in Annex H for auditability.

SOURCE PRESERVATION

Annex H — Preserved R1.0 Source Guide

PRESERVATION NOTE

The following material preserves the full textual and tabular content of "A³O API Reference & Integration Guide R1.0 Candidate." It is retained for traceability. The R1.2 body is the governing modernization candidate.

A³O / UNIVERSAL TRUSTED FRAMEWORK A³O API Reference & Integration Guide Universal node, fleet, swarm and industry integration baseline R1.0 CANDIDATE OPENAPI / ASYNCAPI / SCHEMA EDGE & P2P PROFILES CONFORMANCE TEST KIT

DOCUMENT ID	A3O-API-INTEGRATION-R1.0-EN
DATE	17 JULY 2026
STATUS	CONFIDENTIAL WORKING DRAFT

Contents

1. Scope and supported profiles

This guide is intended for robot and sensor OEMs, system integrators, developers of autonomy stacks, industrial/enterprise platforms and A³O industry profiles. Executable schemas, examples and test vectors take precedence over descriptive prose.

Integration role	Typical responsibility
Node Provider	Robot, vehicle, sensor, actuator or software agent.

Integration role	Typical responsibility
Adapter Provider	Maps vendor protocol and state into A³O contracts.
Profile Provider	Defines industry ontology, workflows, policies and conformance.
Ecosystem Operator	Owns tenant/site/fleet, identities, policies and assurance.
Federation Partner	Exchanges approved state and evidence across a trust boundary.
Assessor	Runs conformance, security and safety test suites.

2. Architecture and trust boundaries

Data/State, Swarm, Decision/Policy, Execution and Evidence service paths.

One authoritative service for every object and state.

Explicit trust boundaries for external devices, peers, AI, humans, actuators and federation partners.

3. Identity and enrollment

Node, workload, user and organization identities.

Certificate/credential bootstrap, attestation, key rotation, revocation and quarantine.

Mission membership and trust-domain claims.

4. Canonical IDs and versioning

ecosystem_id, node_id, component_id, mission_id, task_id, entity_id, policy_id and evidence_id.

Stable URI rules, lifecycle, deprecation and aliasing.

Semantic version negotiation and compatibility profiles.

5. Transport profiles

HTTPS/gRPC for command/query; MQTT, AMQP or NATS-like event profiles; DDS/ROS2 gateway where applicable.

QoS, ordering classes, reconnect, buffering, backpressure, rate limits and dead-letter handling.

Secure P2P transport and constrained-link profile.

6. Universal Telemetry API

TelemetryEnvelope schema, modality registry, units, reference frames, timestamps, quality, uncertainty, privacy and provenance.

Batch, stream and store-and-forward modes.

Vendor extension rules and semantic compatibility checks.

{ "message_id": "...", "source_node_id": "...", "ecosystem_id": "...", "mission_id": "...", "timestamp": "...", "data_modality": "PROCESS_STATE", "semantic_type": "machine.spindle.vibration", "unit_system": "SI", "quality": {...}, "uncertainty": {...}, "schema_version": "1.0.0", "signature": "...", "provenance": {...}, "payload": {...} }

7. State and World Model API

EntityState, ContextState, SharedWorldState and subscriptions.

Vector/sequence semantics, conflict markers, state age and uncertainty.

Snapshot, delta, replay and historical query.

8. Peer Discovery and Mesh API

Mutual identity, mission context and supported schema ranges.

Join, leave, quarantine and topology events.

Network-partition detection and degraded priorities.

9. Capability Exchange API

CapabilityAdvertisement covers limits, payload, mobility, energy, compute, communications and workload.

Lease/TTL, capability change and health degradation.

10. Mission and Task APIs

MissionIntent, MissionPlan, TaskAssignment, acceptance, lease, preemption and reassignment.

Directed, auction, priority, redundant and opportunistic allocation.

Object	Key fields
MissionIntent	objective, priority, constraints, authority, success and termination criteria
TaskAssignment	task, assignee, backup, lease/TTL, required capability, acceptance
CapabilityAdvertisement	capability, limits, energy, compute, communications, workload, safety restrictions

11. Policy, Delegation and Human Authority API

PolicyEvaluation, DelegationGrant, PolicyDecision, human approval, revoke, conditions and deny reasons.
Scope, TTL, ODD, separation of duties and emergency controls.

12. Action and Feedback API

ActionCommand, idempotency, bounded parameters, preconditions, ACK, progress, cancellation and ActionFeedback.
OutcomeAssessment is distinct from transport acknowledgement.

13. Evidence and Assurance API

Append EvidenceFragment, merge, verify, export, manifest, media references, classification and retention.
Selective disclosure and cross-domain export.

14. Twin, Simulation and Replay API

TwinEntity, geometry, process/environment model, scenario and forecast.
LIVE, SIMULATION, FORECAST and REPLAY namespace and adapter separation.

15. Health and Observability

Heartbeat, resource, latency, queue, trust/safety state, degraded reason and evidence offset.
Logs, metrics, traces and SLO/error budget.

16. Offline, Partition and Reconciliation

Store-and-forward, vector clocks, conflict retention, authority expiry and synchronization cursors.
Reconciliation report and human-review triggers.

State	API behavior
CONNECTED	Normal streaming and synchronization.
DEGRADED	Priority classes and reduced update rate.
PARTITIONED	Local delegated operation; buffer evidence and state.
ISOLATED	Profile-specific safe fallback.
RECONCILING	Exchange deltas/fragments and preserve conflicts.
RESTORED	Publish reconciliation result and resume approved mode.

17. Industry Profile SDK

Profile manifest, ontology extensions, state/action schemas, workflow templates, policy packs and tests.
Prohibited semantic overrides and compatibility checks.

18. Errors and Resilience

Canonical error model, retryability, partial success, quarantine, dead-letter and replay.
Safe behavior for unknown or invalid policy-critical data.

19. Security Requirements

mTLS, least privilege, input validation, secret handling, secure update, rate limiting, audit and SBOM status.
Anti-replay, anti-Sybil, peer isolation and evidence integrity.

20. Conformance Test Kit

Schema validation, golden datasets, protocol emulator, latency/loss/partition tests, compromised-peer tests and evidence verification.

Machine-readable pass/fail report tied to profile and maturity.

21. Worked integration examples

Example	Canonical flow
Industrial robot cell	PLC/robot adapter → process telemetry → MachineState → workflow/task → policy/safety → ProcessCommand → quality evidence.
Warehouse AMR fleet	AMR capabilities/energy → shared map/state → distributed allocation → route command → delivery outcome and chain of custody.
Delivery UAV	Navigation/payload/weather → MissionPlan → geofence/authority → flight task → hand-off evidence.
Medical delivery robot	Patient/cargo identity → privacy policy → route/task → human escalation → custody and delivery verification.
Environmental survey swarm	Coverage plan → capability exchange → zone allocation → multimodal telemetry → dataset provenance and gap detection.
DroneDefender	RF/radar/EO-IR/acoustic → profile Detection/Track/Threat → policy/human authority → bounded response → incident evidence.

CONTROLLED TERMINATION

Document End

End of A³O API & Integration Guide R1.2 FULL MASTER — JAUS Integrated.