



TECHNICAL SPECIFICATIONS | FIRST EDITION

IAF AUTOID REGISTRY DATA AND INTERFACE SPECIFICATION

Architecture, identifiers, data contracts, decision logic,
security boundaries and implementation acceptance.

PROPOSED FOR IAF ADOPTION

Version 1.0 | Prepared for IAF adoption

Persistent identities and configuration records
Evidence, status logic and controlled interfaces
Security, interoperability and implementation assurance

International Automotive Federation

English master text | 15 September 2026

iaf.com

Document control

This specification gives the development team a consistent logical model and interface design. The accompanying machine-readable kit implements and tests selected contracts and invariants. The complete production service and official-source integrations require the release work defined here.

Control	Value
Programme	IAF AUTOID
Document	IAF AUTOID Technical Specification First Edition Version 1.0
Status	Proposed for adoption; operational effect requires the recorded IAF authorization
Prepared	15 September 2026
Accountable owner	AUTOID Programme Manager
Publication authority	Presidential Council under the effective Constitution
Adoption record	Resolution reference, authorized scope and effective date recorded in the controlled master list
Review	At least annual maintenance; immediate review on material legal, safety or integrity change
Control	English master; printed and downloaded copies checked against the current controlled edition

Use of this document

This package defines an IAF scheme and its implementation. It does not represent existing regulatory delegation, external acceptance or a live database. Pilot and operational records must be distinguished. The accountable operator shall activate only the markets, categories and functions approved through the release gate.

Research and legal source note

The separate Research and Alignment Note lists the official sources reviewed on 15 September 2026 and the limits of source retrieval. It is informative background. Operational profiles require current, independently reviewed obligation matrices. The standard text is an original IAF scheme proposal; no external organization is represented as endorsing it.

Contents

Document control	2
Contents	3
01 Architecture and implementation boundary	4
02 Identity syntax and graph rules	4
03 Data dictionary and validation	5
04 State machine and effective result	7
05 Database transactions and persistence	9
06 API contract authorization and errors	10
07 Evidence signatures and integrity profile	12
08 Privacy boundaries and retention design	12
09 Performance resilience and observability	13
10 Acceptance tests and production completion	13
IAF document alignment register	15
Local reference validation	15

01 Architecture and implementation boundary

The target is an authoritative global model index with controlled regional collection and restricted vehicle-level records. The service has five trust zones: public verification, authenticated participant services, assessment and decision workflow, restricted evidence and identifier storage, and security administration with signing infrastructure. Only approved public projections may cross into anonymous access.

The first release uses a central transactional allocator and status authority. Regional nodes may collect and verify data under contract but cannot issue divergent current records. During a partition they may queue authenticated submissions; they must not publish new positive outcomes. Authority connectors are read-only unless an explicit integration agreement grants a different function.

The accompanying implementation kit is an executable reference for selected data and decision rules. It is not a deployed registry, a complete backend, a validated security product or an activated national profile. Its SQLite schema demonstrates referential integrity, immutable history and selected release gates. Production requires a transactional server database, authorization layer, encrypted evidence vault, key management, source connectors, monitoring, security testing and the operator release gate.

No blockchain is required. An append-only event ledger with controlled writes, independent audit exports, protected backups and verifiable signed extracts is the baseline. Public immutable storage of personal vehicle data is prohibited. Cryptographic integrity is one property of evidence; source authority and technical relevance remain separate checks.

02 Identity syntax and graph rules

The canonical entity ID is AID-`{class}`-`{UUIDv4}`, with uppercase AID and class code and lowercase UUID text. Example: AID-M-123e4567-e89b-42d3-a456-426614174000. It is an IAF scheme identifier, not a claim to a registered public URN namespace and not a replacement for any legal VIN or approval number. Cases, evidence, profiles, decisions and events use separate UUID references without masquerading as vehicle identities.

The parent chain is organization to model to variant to configuration. A vehicle has one immutable as-built configuration relationship and may have multiple dated current-configuration relationships. A model may map to multiple regulatory types, and one regulatory type may cover several model aliases or subdivisions. That mapping must carry territory, source and configuration applicability.

All graph edges carry source, `effective_from`, optional `effective_to`, `recorded_at` and the approving event. Model parent cycles, a variant with an organization as its direct parent, a vehicle pointing to a model instead of a configuration, and cross-tenant unauthorized edges must be rejected. A component or software edge may be many-to-many and must state installed, compatible, `tested_with` or `replaced_by` rather than an ambiguous association.

A changed product baseline creates a new configuration. A correction to its description creates a new record revision that explains the correction and keeps original history. Merging duplicate records never deletes the losing ID; it becomes a restricted alias with the decision and canonical destination. A merge must not merge private owner data or erase contradictory vehicle evidence.

Entity classes

Entity	Code	Meaning	Minimum distinguishing data
organization	AID-O	Legal operator, manufacturer, applicant or issuer	Registration reference, jurisdiction, legal roles, identity evidence
model	AID-M	Stable engineering family	Manufacturer, engineering code, aliases, family boundary
variant	AID-V	Technical subdivision	Parent model, selection rule, category-specific attributes
configuration	AID-C	Frozen technical baseline	Parent variant, immutable manifest, hardware and software references
vehicle	AID-U	Individual physical vehicle	Protected VIN binding, as-built configuration, lifecycle relationships
component	AID-P	Part type, batch or instance	Granularity, supplier, part code, revision and serial if applicable
software	AID-S	Identified release artifact	Supplier, release/build, digest and compatible hardware
facility	AID-F	Physical production or assessment site	Operator, location at appropriate precision, authorized function

03 Data dictionary and validation

Dates in machine records use UTC RFC 3339 timestamps ending in Z. The API shall reject timestamps without a timezone. Source dates known only to day precision use a separate date field and precision flag; the application must not invent an exact observation time. Effective time may precede recording time but the reason for late entry shall be recorded.

Unknown, absent and not applicable are distinct. Null is accepted only for fields whose definition permits it. An empty string shall not stand for unknown validity. Unit-sensitive numeric values shall include an approved unit and conditions; strings such as 500 km shall not replace a structured quantity in the machine record.

External identifiers shall be unique only within their lawful issuer and scheme context. A protected VIN index uses keyed HMAC rather than an unsalted public hash; the full value is encrypted separately. Key rotation preserves lookup through controlled reindexing and logs. The reference database stores only a synthetic lookup token, never an actual VIN.

The public JSON schema is deliberately narrower than the internal dictionary. It contains model and scoped assertion summaries, not evidence files, VINs, owners or access tokens. Additional fields are rejected at every nested level. The validation suite tests both schema structure and semantic rules; syntax validation alone is insufficient.

Core fields

Field	Type	When required	Validation and meaning
entity_id	string	Required	Immutable class-coded UUID; validated against kind
kind	enum	Required	organization model variant configuration vehicle component software facility
record_revision	integer	Required	Positive monotonically increasing number for an entity
schema_version	string	Required	Major and minor transport schema version
manufacturer_id	AID-O	Model and related scopes	Verified legal manufacturer; not just brand name
parent_id	AUTOID	Variant and configuration	Allowed hierarchy only; immutable without a correction event
identity_state	enum	Required	draft verified disputed suspended retired
effective_at	UTC timestamp	Required	When the fact or event became effective
recorded_at	UTC timestamp	Required	When accepted into the registry; never backdated to conceal delay
source_ref	UUID or controlled reference	Required for verified fields	Evidence or verified source supporting the field
classification	enum	Required	public restricted confidential highly_restricted
steward_id	AID-O	Required	Organization accountable for updates
external_identifier	structured value	When applicable	Scheme issuer original normalized value and context; access-restricted
configuration_manifest	structured immutable payload	Configuration	Hardware software boundaries options units and range exclusions
digest	algorithm plus value	Evidence and manifest	SHA-256 baseline; original bytes and canonical JSON are distinct digest targets
jurisdiction_profile_id	string	Assessment	Controlled profile and version; inactive profiles cannot issue claims
assessment_state	enum	Assessment	not_assessed in_review verified refused suspended withdrawn superseded

Field	Type	When required	Validation and meaning
legal_source_state	enum	Assessment	not_checked reported_valid reported_restricted reported_expired reported_withdrawn
source_route	enum	Assessment	authority_approval manufacturer_self_certification individual_approval exemption
source_checked_at	UTC timestamp or null	Assessment	Actual successful check; null means not checked
source_fresh_until	UTC timestamp or null	Assessment	Approved maximum source age; independent of approval expiry
review_due_at	UTC timestamp	Assessment	Case-maintenance due time
source_valid_until	UTC timestamp or null	When stated by source	No fabricated expiry for indefinite approvals
currency	enum	Assessment	current review_due stale; derived from timing and events
safety_alert_state	enum	Public projection	none_reported open resolved; no guarantee of comprehensive coverage
blocking_restriction	boolean	Public projection	Authorized restriction preventing a current positive result
decision_ref	UUID	Verified assessment	Signed or otherwise authenticated independent decision
relying_party_notice	event relationship	Material status change	Recipient scope delivery status retries and acknowledgement

04 State machine and effective result

The effective verification result is calculated per assertion, market profile and exact configuration. It is not a model-wide approval flag. The API returns the source states and a computed result with reason codes and as_of time. A model with several markets returns several independently calculated results.

Apply conditions in this order: service cannot establish trustworthy current state gives UNAVAILABLE; identity disputed, suspended or retired, a blocking restriction, a suspended or withdrawn assessment, or a source reported restricted, expired or withdrawn gives RESTRICTED; an unverified draft identity gives UNAVAILABLE; no requested assessment gives IDENTITY_ONLY; a refused, in-review or superseded assessment gives NOT_CURRENT; a verified assessment without its decision reference gives UNAVAILABLE; an exact configuration mismatch gives NOT_CURRENT; missing legal-source verification or impossible source-check timing gives UNAVAILABLE; stale

source, overdue review or expired source validity gives NOT_CURRENT; an open nonblocking safety alert gives QUALIFIED_ALERT; only all remaining conditions give CURRENT.

CURRENT means a current AUTOID scheme statement for the specified scope. It is never serialized as approved_worldwide or permission_to_register. QUALIFIED_ALERT must display the alert and its impact prominently. none_reported means no alert in the checked source coverage, not that no defect exists anywhere.

Restoration from suspension goes through in_review and a new independent decision. The historical assertion remains auditable. A withdrawn statement cannot be reactivated; a new assertion may reference it only after fresh assessment. External legal-source dates and IAF review dates remain separate.

Controlled transitions

Domain	Transition	Authority	Precondition
Identity	draft to verified	Registrar plus independent identity review	Legal authority and object boundary authenticated
Identity	verified to disputed or suspended	Authorized identity or duty function	Conflict or protective restriction with reason
Identity	suspended or disputed to verified	Independent reassessment decision	Conflict resolved and restrictions closed
Identity	verified to retired	Authorized lifecycle decision	Historic ID retained; dependent safety records preserved
Assessment	not_assessed to in_review	Case manager	Scope and plan accepted
Assessment	in_review to verified or refused	Independent decision maker	All applicable requirements checked; reasoned outcome
Assessment	verified to suspended	Duty or decision function	Credible integrity or conformity concern
Assessment	suspended to in_review	Case manager after authorized reopening	Fresh reassessment; no direct automatic restoration
Assessment	verified or suspended to withdrawn	Independent decision maker	Established loss of support; historical record retained
Assessment	verified to superseded	Independent replacement decision	Successor and exact scope recorded
External source	any reported status to another	Authenticated source check	Private opinion cannot modify authority facts
Currency	current to review_due or stale	Deterministic clock or event rule	Deadline elapsed or source feed outside maximum age

05 Database transactions and persistence

The table inventory describes the production logical model. The accompanying SQLite reference implements a subset for entity hierarchy, profile, case rows, evidence references, decisions, assertions and immutable events. The remaining production stores must be implemented by the selected development team; the inventory must not be confused with a completed backend.

A publish transaction validates actor authority, expected revision, independent decision, requirement closure, approved profile and immutable configuration; commits the decision and events; and enqueues the projection update in the same transaction. The reader sees either the old safe state or the new authorized state. A public projection shall not be updated ahead of its signed decision.

Use optimistic concurrency through expected revision and ETag. A stale writer receives a conflict or precondition failure and must reread. Idempotency keys are scoped to authenticated issuer and tenant, retained for at least the supported retry window, and bound to the payload digest. A duplicate command must not allocate a second entity.

The production database must enforce foreign keys, unique allocation, allowed type hierarchy and immutable configuration manifests. Application authorization is also required; database constraints do not determine whether the actor has authority. Administrative emergency changes require dual control, a reason and retrospective review. All retained originals require backup and recoverable encryption keys.

Production table inventory

Store	Key and core relationship	Purpose
entities	id PK; kind; steward; created_at	Persistent identity; never deleted or reassigned
entity_revisions	entity_id plus revision PK	Append-only descriptive and identity-state snapshots
relationships	edge UUID; parent; child; kind; effective dates	Typed hierarchy and lifecycle edges with provenance
profiles	id plus version; territory; route; active	Versioned applicability package and sources
profile_requirements	profile version plus requirement key	Applicability, legal source, method and evidence rules
cases	case UUID; applicant; requested scope	Workflow record with configuration and profile versions
case_requirements	case plus control or legal row	Applicable status, evidence, result and closure
evidence	evidence UUID; issuer; digest; source class	Original evidence metadata and protected object-store pointer
evidence_links	case row plus evidence UUID	Many-to-many coverage and applicability rationale

Store	Key and core relationship	Purpose
decisions	decision UUID; case; assessor; reviewer; decider	Append-only outcome; distinct-role and completeness gate
assertions	assertion UUID; config; profile; decision	Scoped claim independent of source and identity state
assertion_events	assertion plus sequence	Append-only status, source checks and effective times
external_identifiers	scheme issuer token plus entity	Restricted lookup; actual values in encrypted vault
lifecycle_events	event UUID; subject; source; effective time	Recall, update, remedy, production and retirement
issuer_trust	issuer plus scope and key validity	Trusted evidence types, permitted roles and revoked keys
idempotency	tenant issuer plus key	Payload digest and exact response; conflicting replay rejected
outbox	event UUID; payload digest; delivery state	Transactional status propagation and recipient notification
audit_events	event UUID; actor; action; reason; recorded_at	Privileged actions and evidence access; protected export

06 API contract authorization and errors

The bundled OpenAPI contract fully defines the public model-read and verification-read interfaces and the authenticated application-create request. The other operations in this inventory are requirements for later implementation and are not represented as working endpoints. No API server URL in the package is an active IAF service; example.invalid is used deliberately.

Use short-lived access tokens with audience and issuer validation and scopes such as case:submit, evidence:read, review:write and decision:write. Authorization must additionally check tenant, assigned case, object classification and conflicts. A scope string alone is not sufficient. Full VINs and secret tokens must not appear in GET query strings or ordinary access logs.

Responses shall include correlation_id, schema_version, generated_at, source_checked_at where relevant, and explicit result reason codes. Public responses must not disclose whether a protected VIN or private entity exists. Apply rate limits and enumeration detection. Evidence upload services must protect against malware, oversized archives and server-side requests to untrusted internal URLs.

Error meanings: 400 malformed input; 401 unauthenticated; 403 not authorized; 404 not found or intentionally concealed; 409 identity, revision or idempotency conflict; 412 failed If-Match; 422 semantic or profile validation failure; 429 rate limit; 503 trustworthy

service state unavailable. Do not return an empty 200 response that consumers may interpret as a pass.

Webhooks use signed payloads, timestamp and event ID, an allowlisted recipient, bounded retries and dead-letter review. Delivery is at least once; recipients must deduplicate. Every notice contains the affected scope and a verification reference, not unnecessary personal evidence. A missed delivery does not excuse a stale public status.

API inventory

Method	Path	Access	Meaning
GET	/v1/models/{id}	Public	Allowlisted model summary with separate market assertions
GET	/v1/assertions/{id}/verification	Public	Computed result, scope, source states, freshness and reason codes
POST	/v1/cases	Applicant scope	Create a scoped application with idempotency key
POST	/v1/cases/{id}/evidence	Assigned case	Quarantined upload registration; no automatic trust
POST	/v1/cases/{id}/review	Independent reviewer	Record authenticated review and closure verification
POST	/v1/cases/{id}/decisions	Independent decision maker	Submit signed scoped outcome with expected revision
POST	/v1/entities/{id}/changes	Authorized steward	Propose change; cannot overwrite immutable baseline
POST	/v1/vehicle-lookups	Authorized purpose	Restricted exact-match lookup; do not expose VIN in URL
GET	/v1/profiles/{id}/versions/{version}	Permitted profile access	Controlled requirements and applicability metadata
POST	/v1/incidents	Authenticated reporter	Urgent report receipt; separate triage and status authorization
GET	/v1/events	Authorized subscription	Cursor-based scoped lifecycle changes
POST	/v1/exports	Authorized purpose	Auditable asynchronous portable export
GET	/v1/trust/keys	Public	Permitted verification keys and revocation metadata

07 Evidence signatures and integrity profile

Retain SHA-256 digests of original evidence bytes and separately of frozen manifests. JSON signing uses a documented canonicalization profile and detached JWS with a restricted algorithm allowlist, initially ES256. Implementations must verify issuer, key ID, certificate or key trust, audience or intended purpose, payload digest, time and revocation. Algorithm none and client-selected untrusted keys are rejected.

The signed object contains statement ID, subject and configuration IDs, profile ID and version, declared scope, evidence references, decision reference, issue time, review due time and the exact claim text. It excludes private VINs and personal data unless the approved restricted use specifically requires them. A separate status service communicates later changes; a downloaded signature cannot prove the absence of later revocation.

Trust-list changes require independent approval. Key compromise triggers immediate key restriction, identification of affected signed objects and reissuance or review according to compromise timing. Preserve the original signed statement and the reason for its status. Revocation of a key today does not automatically prove that every earlier signature was fraudulent.

The reference kit does not implement signing or key management. Before production, obtain test vectors for canonicalization, signature verification, key rotation, revoked and expired keys, wrong issuer, changed payload, clock skew and unavailable status service. An application that simply hashes sorted Python JSON does not claim interoperable JCS signing.

08 Privacy boundaries and retention design

The public model index and restricted vehicle vault are separate services or strongly separated security domains. The public service receives only reviewed projections. Unit-level access requires a lawful purpose, an authenticated requesting party or a holder-authorized opaque token, and a minimal response policy. UUID unpredictability does not replace authorization.

Default public model data include manufacturer legal name, model aliases, declared configuration, market route, scoped assessment state, currency and relevant safety notice. Exclude owner identity, licence plates, exact vehicle whereabouts, uploaded identity documents, confidential technical drawings and unrestricted VINs. Redact original evidence before any permitted public excerpt.

Retention is approved per dataset, not hard-coded as keep forever. Preserve minimal non-personal identifier tombstones for uniqueness and continuity where lawful; retain technical evidence for the profile-defined lifecycle and claims period; delete personal data when its purpose and applicable obligations end. Backup deletion is controlled by bounded expiry and documented restoration filtering.

A linked battery passport retains its own lawful identifier, operator and access rules. AUTOID may reference it but does not automatically fulfil a statutory passport

obligation. Repurposing or remanufacturing may require a new external passport linked to the predecessor under the applicable rule. The AUTOID tombstone policy must not override a legal requirement for the external passport to cease after recycling.

09 Performance resilience and observability

Proposed initial sizing for design tests is one million public model and configuration records and one hundred million restricted vehicle records. These are capacity scenarios, not expected adoption or current registry volume. Partition event and vehicle stores by operationally appropriate keys while preserving global uniqueness, cross-market references and tenant authorization.

Initial service objectives are 99.9 percent monthly public verification availability, p95 response time below 750 milliseconds for cached public reads under an agreed load profile, authorized critical status propagation within 15 minutes, recovery time of four hours and recovery point of 15 minutes. A load-test report must state dataset size, concurrency, request mix, hardware, cache state and error rate.

Cache validity is bounded by both projection version and evidence freshness. Critical state events invalidate affected cached records; if invalidation is unreliable, the service suppresses current positives. Negative lookups must be short-lived to avoid hiding recently created restrictions or corrected identities.

Monitor allocation conflicts, unauthorized access attempts, status lag, event-processing errors, source outages, stale assertions, webhook failures, key revocation and backup verification. Alerts need owners and escalation; dashboards without action procedures do not meet the standard. Audit telemetry is itself classified and retained under a schedule.

10 Acceptance tests and production completion

Use the reference tests to verify the delivered machine contracts and selected decision/database invariants. They are not a substitute for live authorization, cryptographic, concurrency, penetration, performance, disaster recovery or privacy testing. The QA result identifies exactly which checks were executed on the local reference kit.

Before go-live, execute the acceptance matrix below against the actual service and retain inputs, expected outputs, observed results, version, operator and reviewer. Every failed applicable launch requirement must be resolved. Tests should include malicious and accidental inputs, not only a successful application.

The operational deployment team must complete authority and source integration agreements, actual country profiles, production encryption and key management, legally reviewed retention schedules, provider appointments, service terms, incident duty coverage and independent readiness review. These are deployment dependencies with accountable owners, not hidden assumptions in the code.

A release manifest shall list standard, manual, checklist, schema and API versions, migrations, profile versions, authorized scope, remaining excluded modules and rollback plan. No document or code file shall be labelled as evidence that an authority has accepted AUTOID unless that acceptance exists in a verified instrument.

Production acceptance matrix

Area	Scenario	Required result
Identity	Wrong class, malformed UUID, duplicate and concurrent allocation	Reject invalid inputs; one stable result for authorized retry
Hierarchy	Wrong parent class, cycle, changed frozen manifest	Reject mutation; require controlled new configuration
Evidence	Missing original, forged approval, wrong issuer or expired scope	Hold case; no favourable default
Decision	Same assessor and decider; incomplete requirements; invalid profile	Reject positive decision
Status	Suspended source, stale data, future check time, past review due	No current positive display
Scope	Correct model but wrong variant or software configuration	Reject inherited claim
Recall	Open campaign and unit-only remedy	Keep campaign and other units open
Privacy	Cross-tenant read, VIN enumeration, nested unexpected public field	Deny access and leak no restricted data
Security	Tampered signature, unknown key, revoked key, replayed request	Reject or return explicit unavailable state
Availability	Feed outage, network partition, stale cache, recovery	Preserve restrictions; fail safely
Interoperability	Export, second parser, round-trip and schema migration	Preserve identifiers, meaning, history and access
Remedies	Appeal panel conflict and corrected decision propagation	Independent handling; old result remains historical

IAF document alignment register

The following source documents were available for alignment. Their formal effective status and any subsequent revisions shall be checked by document control before adoption. This register identifies specific dependencies; it does not assert that every IAF document has been exhaustively reconciled.

Source	Interface	Alignment action
IAF Constitution First Edition	Articles 18 19 23 24 and standards process	Council corporate authority; independent technical decisions; 7 30 60 calendar-day complaint periods; 30-day appeal filing and independent three-person panel
IAF Committees Guide Rev01	Clauses 8 9 10 and 12	Controlled pilot gates, consultation normally 60 days, independent review, separate provider memoranda
IAF-PR-026 Rev2	AutoID Issuance and Lifecycle Control	Retain identity-not-approval principle; expand into precise configuration and jurisdiction evidence rules
IAF-FR-032 and IAF-RG-015 Rev2	Scope issue change and identity register	Map existing fields to the richer model; preserve legacy identifiers as aliases
IAF-WI-005 Rev2	Identifier and verification checks	Use explicit AUTOID class syntax; reconcile old prefix or serial guidance before release
IAF-PR-014 Rev2	Appeals Handling	Align business-day targets and Board wording with constitutional calendar-day periods and current Council/panel authority
AUTOID website	Pilot and non-regulatory statements	Keep pilot label until release; distinguish identity from scoped evidence review; replace broad claims about VIN and homologation with accurate descriptions

Website alignment actions

Keep the development or pilot label until operational release. Add precise model, configuration and jurisdiction evidence descriptions. Replace the broad statement that a VIN has no technical meaning with a factual distinction: a VIN identifies a vehicle and can encode manufacturer and vehicle attributes, while AUTOID links additional controlled evidence and lifecycle records. Avoid describing homologation as merely limited transparency; explain the complementary role accurately. Publish no verification endpoint before it is implemented and controlled.

Local reference validation

The delivered reference kit was tested with 44 positive and negative tests covering schema validation, public-data exclusions, identifier class and UUID format, configuration mismatch, source freshness, independent decision gates, authenticated evidence, inactive profiles, immutable records and idempotent retries. All 44 passed in the local fixture environment. This result does not validate production security, legal applicability, live data feeds or global capacity.