RM-S-AI-DEL-001 AI Agent Delegated Authority Verification Standard

Standard ID: RM-S-AI-DEL-001 **Version:** v1.0-F **Status:** Frozen **Publication Date:** 2026-07-23 **Effective Date:** 2026-07-23 **Standard Family:** RM-S — Canonical Standards Namespace **Normative Status:** Normative **Publisher:** RuleMark **Document Language:** English **Machine ID:** `rulemark:standard:rm-s-ai-del-001:v1.0-F`

Document Control

Field	Content
Standard ID	RM-S-AI-DEL-001
Standard Name	AI Agent Delegated Authority Verification Standard
Version	v1.0-F
Status	Frozen
Publication Date	2026-07-23
Effective Date	2026-07-23
Standard Family	RM-S
Domain	Delegated authority of AI agents: grant, scope, binding, revocation, sub-delegation, human checkpoints, action logging, traceability
Normative Status	Normative
Target Subject	A (delegator × agent instance × authorization credential) triple at a specific verification point
Default Determination	PASS / CONDITIONAL / DEFICIENT / FAIL / NOT ASSESSED
Conformance Use	Authorized

Abstract

This standard establishes a unified, deterministic, and machine-executable set of rules for verifying the delegated authority under which an AI agent acts on behalf of a principal.

This standard requires verification of the identity of the delegator and the operator as separate parties, the existence and validity of an authorization credential, the binding of that credential to a specific agent instance and to the digest of its configuration, the expression of authority scope in machine-testable form, the enforcement of limits and human checkpoints, the revocation path and its effective time, the chain of any sub-delegation, the completeness and tamper-evidence of the action log, and the ability to trace every action back to a specific grant of authority.

This standard does not evaluate the agent's capability, output quality, or decision correctness. It does not certify the safety or alignment of the underlying model, does not determine whether an authorization is legally effective in any jurisdiction, does not assess the security of the hosting infrastructure, does not predict future behaviour, and does not allocate liability between a delegator and a third party.

This standard answers only:

> At a specified verification point, under a specified standard version, for a specified delegator, agent instance, and authorization credential, and a specified evidence set, does the delegated authority satisfy the applicable requirements of RM-S-AI-DEL-001?

1. Applicability

1.1 Conditions of Applicability

This standard applies where all of the following conditions hold:

1. a software agent exists that can initiate actions against external systems without a per-action human instruction, including API calls, transactions, message dispatch, and state changes;
2. the agent's authority to act derives from an explicit grant by an identifiable delegator, being a natural person, a legal entity, or another agent holding delegable authority, and not from the agent itself;
3. that grant exists as a machine-readable, verifiable credential or record carrying a unique identifier, an issuance time, and an issuing party;
4. the delegator is a uniquely identifiable subject;
5. the agent's actions produce observable effects on a third party or an external system beyond the delegator, excluding purely internal reasoning and draft generation;
6. at the verification point the grant is asserted to be in force: issued, not expired, and not revoked.

1.2 Out of Scope

This standard does not apply to:

- question-answering or generative models with no capacity to act on external systems;
- assistive tools that only propose actions which a human then executes individually;
- conventional scripts, scheduled jobs, and RPA flows where no agent decision occurs and permissions are fixed by static configuration;
- self-running agents with no delegator, whose permissions derive from their own deployment rather than from a grant;
- ordinary employee and service-account permission systems within an organisation, being general identity and access management rather than agent delegation;
- model training, fine-tuning, and evaluation processes themselves;
- smart contracts and on-chain automatic execution logic in themselves, other than the part invoked by an agent;
- agents running entirely in sandbox, test, or simulated environments;
- pure information exchange between agents where no authority is transferred;
- agents whose authority has been fully revoked with no residual grant in force;
- **the quality, accuracy, or usefulness of an agent's output**, which belongs to the execution domain and is not addressed by this standard.

1.3 Non-Goals

This standard:

- does not provide investment advice;
- does not certify legal compliance in any jurisdiction;
- does not replace a regulator, auditor, court, or legal counsel;
- does not constitute a credit rating;
- does not constitute a RuleMark guarantee, recommendation, or endorsement;
- **does not evaluate the agent's capability, output quality, or decision correctness.** It answers only what the agent was authorised to do and whether it stayed within that authority;
- does not certify the safety, alignment, or values of the underlying model;
- does not determine whether an authorization is legally effective in the delegator's jurisdiction;
- does not assess the security of the infrastructure hosting the agent;

- does not predict or guarantee future agent behaviour. A conclusion describes the authority state and the recorded history at the verification point only;
- does not allocate liability or damages between a delegator, an operator, and a third party arising from an agent's actions.

1.4 Conforming Subject

A conformance claim MUST correspond to one (delegator × agent instance × authorization credential) triple at one specified verification point. **Any change to any element of the triple — a new agent version, a new credential, or a widened scope — constitutes a new conforming subject and MUST be re-verified.**

Unique identification MUST include at least:

- delegator legal entity name or natural person identifier, jurisdiction, and registration number. For a natural person, a stable identifier designated by the delegator MUST be used; this field MUST NOT be empty;
- operator legal entity identifier, being the party that actually runs the agent;
- `delegator_is_operator`, explicitly `true` or `false`. This field MUST NOT be omitted;
- agent instance identifier, globally unique and never reused;
- agent software version identifier;
- SHA-256 digest of the agent configuration, covering at minimum the system prompt and the tool inventory;
- authorization credential identifier and the SHA-256 digest of the credential;
- credential issuance time, effective time, and expiry time, all ISO 8601 UTC;
- the delegation chain: where authority is sub-delegated, every upstream authorization identifier up to the root delegator, in order;
- verification point, ISO 8601 UTC, precision to the second.

1.5 Participant Roles

This standard distinguishes the following participant roles. A single legal entity MAY assume multiple roles, but the responsibilities of each role MUST be separately identifiable.

- **Delegator (Principal)**: the party that grants authority. It issues and signs the authorization credential, defines the scope, and may revoke.
- **Operator**: the legal entity that deploys and runs the agent. **It is the obligated party** and the subject of the conformance claim. It submits the claim, the machine record, and all mandatory evidence; ensures the credential is genuine and unexpired; discloses the complete delegation chain and tool inventory; retains and submits the action log; and triggers re-verification on scope change or revocation.
- **Verifier**: the party that executes the verification procedure of §15.1, reconciles the action log against the authority scope, checks the delegation chain, determines the result, and signs the verification record.
- **Machine Validator**: the system that executes the schema and determination logic, replays the action log against scope, checks log hash-chain continuity, and outputs deterministic failure codes.
- **Evidence Provider**: the delegator (credential original and signature), a tamper-evident log service or anchoring party, the external systems invoked (counterparty call records), an independent technical assessor, and the operator for declaration-type evidence only.

Neither a statement by the operator nor a statement by the agent itself constitutes independent evidence. This includes an agent's own assertion that it did not exceed its authority.

The delegator and the operator MUST be separately identified. Where they are the same entity, `delegator_is_operator` MUST be explicitly `true`. It MUST NOT be defaulted or left empty.

Where a commercial, fee, ownership, or control relationship exists between the Verifier and the operator or delegator, it MUST be disclosed under §19.3. An undisclosed material interest relationship renders the verification result invalid.

2. Normative Terminology

The normative keywords in this standard are used with the following meanings:

- **MUST / SHALL**: mandatory requirement;
- **MUST NOT / SHALL NOT**: express prohibition;
- **SHOULD**: recommended requirement; a documented reason MUST be recorded where not satisfied;
- **SHOULD NOT**: not recommended; a documented reason MUST be recorded where adopted;
- **MAY**: optional capability.

Only the above English keywords appearing in uppercase have normative effect. Informative explanations, examples, and clarifications shall not override normative requirements.

3. Definitions

Term	Definition
Agent	A software system that can initiate actions against external systems without a per-action human instruction.
Agent Instance	A specific running deployment of an agent, identified by a globally unique identifier that is never reused.
Agent Configuration	The system prompt, tool inventory, and any other parameter set that determines what the agent may attempt. Its digest binds a credential to a specific behavioural configuration.
Delegator (Principal)	The party that grants authority to an agent.
Operator	The legal entity that deploys and runs the agent instance.
Delegation	The grant of authority by a delegator to an agent instance, expressed in an authorization credential.
Authorization Credential	A machine-readable, signed record expressing a delegation, carrying a unique identifier, issuer, subject, scope, validity period, and signature.
Authority Scope	The machine-testable expression of what an agent may do, comprising permitted action types, permitted counterparties, quantitative limits, and any conditions.
Action	A single externally observable operation initiated by the agent, including a tool invocation, API call, transaction, message dispatch, or state change.
Tool Invocation	An action executed through a named tool listed in the agent configuration.
Scope Test	The deterministic evaluation of whether a given action falls within a given authority scope.
Out-of-Scope Action	An action for which the scope test returns false against every credential in force at the time of the action.
Quantitative Limit	A numeric bound within an authority scope, such as a per-action maximum, a cumulative maximum, or a rate limit.
Human Checkpoint	A point at which the authority scope requires explicit human approval before an action may proceed.
Revocation	The act by which a delegator terminates a grant before its expiry.
Revocation Effective Time	The time from which a revoked credential no longer authorises any action.
Revocation Propagation Time	The elapsed time between the revocation being issued and the agent ceasing to act under it.
Sub-delegation	The grant by an agent of some or all of its authority to a further agent.
Delegation Chain	The ordered sequence of authorization credentials from a given credential up to the root delegator.

Term	Definition
Root Delegator	The first delegator in a delegation chain, whose authority does not derive from another credential.
Action Log	The ordered, tamper-evident record of every action initiated by the agent instance.
Hash Chain	A construction in which each log entry commits to the digest of the previous entry, so that removal or alteration of any entry is detectable.
Log Anchoring	The publication of a log digest to an external system such that the log state at a point in time cannot be retroactively altered.
Verification Point	The single UTC time at which the authority state is assessed.
Assessment Window	The period preceding the verification point over which action logs are examined.
Material Event	An event that changes the conformance conclusion, see §13.1.
Evidence	A verifiable record, credential, log, signature, or third-party confirmation supporting or refuting a requirement.
PASS	All applicable MUST requirements are satisfied, no MUST NOT is violated, all mandatory evidence is valid, and the verification record is signed.
CONDITIONAL	All MUST and MUST NOT requirements are satisfied and one or more SHOULD requirements are unsatisfied.
DEFICIENT	Nothing is violated on the available evidence, but one or more mandatory items cannot be verified because evidence is missing, incomplete, or expired, and the gap is curable.
FAIL	At least one applicable MUST is unsatisfied, a MUST NOT is violated, mandatory evidence is missing or invalid, or the result is not re-computable.
NOT_ASSESSED	Verification is not complete and no conformance conclusion is constituted.

3.1 Restriction on Ambiguous Terms

The following words MUST NOT be used alone in normative clauses: reasonable, appropriate, sufficient, adequate, timely, regular, high-quality, secure, transparent, real-time, independent, material, low-risk.

This standard additionally prohibits the standalone use of: autonomous, self-directed, human-in-the-loop (without naming the checkpoints), supervised, guardrailed, sandboxed (without naming the boundary), aligned, safe, and trusted agent.

"Autonomy" MUST NOT be used to describe an absence of delegated authority. An agent acting without a verifiable grant is out of scope under §1.2, not autonomous.

Where any such word is used, the clause MUST also define: a measurable value; a time range; test conditions; the responsible party; and the determination method. The terms defined in §3 satisfy this requirement for the meanings given there.

4. Core Principles

4.1 Evidence First

Any conformance conclusion MUST be supported by verifiable evidence. A statement by the operator or by the agent itself MUST NOT constitute independent evidence.

4.2 No Authority Without a Grant

An agent has no inherent authority. Every action MUST be attributable to a specific authorization credential in force at the time of that action.

4.3 Separation of Delegator and Operator

The party granting authority and the party running the agent MUST be separately identified, even where they are the same entity.

4.4 Scope Must Be Testable

An authority scope MUST be expressed such that, for any given action, a deterministic scope test returns true or false. **A scope expressed only in natural language MUST NOT be treated as an authority scope under this standard.**

4.5 Configuration Binding

A credential MUST bind to the digest of the agent configuration it authorises. **Changing the system prompt or the tool inventory changes what the agent will attempt, and therefore requires a new grant.**

4.6 Revocation Must Terminate

A revocation MUST have a determinate effective time, and the agent MUST cease to act under the revoked credential within the disclosed propagation time.

4.7 Chain Integrity

Sub-delegated authority MUST NOT exceed the authority of the credential from which it derives, at any level of the chain.

4.8 Actions Must Be Recorded

Every action MUST be recorded in a tamper-evident log before the conformance of that action can be determined. **An unrecorded action is an unverifiable action.**

4.9 Non-Discretionary Determination

The same inputs, the same standard version, and the same verification point MUST produce the same result.

4.10 Machine Readability

Credentials, scopes, action logs, and verification results MUST have a structured representation.

4.11 Minimum Necessary Claim

A conformance conclusion MUST NOT exceed the verified triple, the assessment window, and the verification point.

4.12 Freshness

Evidence exceeding the maximum age of §14.4 MUST NOT be used to prove current state.

4.13 Traceability

Every mandatory requirement MUST be traceable to evidence, a verification step, and a failure code.

4.14 Failure Closure

Where mandatory evidence is missing, evidence conflicts, or the result is not re-computable, the result MUST be FAIL or, where §16.6 applies, DEFICIENT. A default pass MUST NOT be applied.

5. Canonical Identity and Responsible Parties

5.1 Conforming Subject Canonical Identity

Every conforming subject submitted for verification MUST have a canonical identity containing at least:

```
{
  "delegator_id": "",
  "delegator_jurisdiction": "",
  "operator_id": "",
  "delegator_is_operator": false,
  "agent_instance_id": "",
  "agent_version": "",
  "agent_config_sha256": "",
  "authorization_id": "",
  "authorization_sha256": "",
  "issued_at": "",
  "effective_at": "",
  "expires_at": "",
  "delegation_chain": [],
  "verification_point": "",
  "assessment_window_start": "",
  "standard_version": ""
}
```

`delegator_is_operator` MUST be explicitly `true` or `false`. It MUST NOT be omitted.

`delegation_chain` MUST be an empty array where the credential is issued directly by a root delegator, and MUST list every upstream authorization identifier in order otherwise. It MUST NOT be omitted.

5.2 Delegator Registration

The delegator MUST be identified by legal entity name, jurisdiction, and registration number, or, for a natural person, by a stable identifier designated by the delegator. The identifier MUST NOT be empty and MUST NOT be a pseudonym that cannot be resolved by an authorised verifier.

5.3 Operator Registration

The operator MUST disclose its legal entity name, jurisdiction, registration number, and the official channel at which it publishes agent incident notices.

5.4 Change of Responsibility

Where the delegator, the operator, the agent instance, the agent configuration digest, or the authorization credential changes, the record MUST capture the position before and after the change, the effective time, and the supporting evidence. **The affected verification result expires at the time of the change** and MUST be recomputed.

6. Authorization Credential

6.1 Existence

Every agent instance within scope MUST hold at least one authorization credential in force at the verification point. Where no credential exists, or none is in force, the result MUST be FAIL using `RMSAIDEL-AUT-001`.

6.2 Credential Content

An authorization credential MUST contain at least: a unique authorization identifier; the delegator identifier; the agent instance identifier; the agent configuration digest; the authority scope as defined in §7; the issuance time, effective time, and expiry time; the delegation chain; and the delegator's signature. Where any field is absent, the result MUST be FAIL using `RMSAIDEL-AUT-002`.

6.3 Signature

The credential MUST be signed by the delegator, and the signature MUST be verifiable against a public key or certificate whose location is stated in the credential. The signature MUST cover the entire credential including the authority scope. Where the signature is absent, unverifiable, or does not cover the scope, the result MUST be FAIL using `RMSAIDEL-AUT-003`.

6.4 Validity Period

The credential MUST carry an expiry time. **An open-ended credential MUST NOT be issued.** The interval between the effective time and the expiry time MUST NOT exceed **365 days**. Where no expiry exists or the interval exceeds that limit, the result MUST be FAIL using `RMSAIDEL-AUT-004`.

6.5 Configuration Binding

The credential MUST bind to the SHA-256 digest of the agent configuration. Where the configuration digest recorded at the verification point differs from the digest bound in the credential, the credential MUST NOT be treated as in force, and the result MUST be FAIL using `RMSAIDEL-AUT-005`.

6.6 Instance Binding

The credential MUST bind to a specific agent instance identifier. **A credential MUST NOT be transferable between agent instances.** Where the same credential is presented for more than one agent instance, the result MUST be FAIL using `RMSAIDEL-AUT-006`.

7. Authority Scope

7.1 Machine-Testable Expression

An authority scope MUST be expressed such that a deterministic scope test returns `true` or `false` for any given action. The expression MUST include, for each permitted action type: the action type identifier, the permitted counterparties or a stated rule for determining them, and any quantitative limits.

A scope stated only in natural language MUST NOT be treated as an authority scope. Where the scope is not machine-testable, the result MUST be FAIL using `RMSAIDEL-SCP-001`.

7.2 Enumerated Action Types

Permitted action types MUST be enumerated. **A scope MUST NOT be expressed as a general permission with exclusions**, because an action type not contemplated at issuance would then be permitted by default. Where a scope grants by exclusion rather than by enumeration, the result MUST be FAIL using `RMSAIDEL-SCP-002`.

7.3 Tool Inventory Consistency

Every action type permitted by the scope MUST correspond to a tool present in the agent configuration bound under §6.5, and every tool in that configuration that can produce an external effect MUST either be covered by the scope or be explicitly listed as disabled. Where a tool capable of external effect is neither in scope nor disabled, the result MUST be FAIL using

`RMSAIDEL-SCP-003`.

7.4 Quantitative Limits

Where an action type involves a quantity, the scope MUST state at least: a per-action maximum, and a cumulative maximum over a stated period. Both MUST be expressed as decimal strings with the unit named. Where a quantity-bearing action type carries no per-action maximum or no cumulative maximum, the result MUST be FAIL using `RMSAIDEL-SCP-004`.

7.5 Counterparty Restriction

The scope MUST state the permitted counterparties for each action type, either as an enumerated list or as a rule that a verifier can evaluate deterministically. **An unrestricted counterparty set MUST be stated explicitly and MUST NOT arise by omission.** Where the counterparty restriction is absent, the result MUST be FAIL using `RMSAIDEL-SCP-005`.

7.6 Out-of-Scope Actions

Every action in the assessment window MUST pass the scope test against a credential in force at the time of that action. Where any action fails the scope test against every such credential, the result MUST be FAIL using `RMSAIDEL-SCP-006`.

7.7 Limit Breach

Where any action exceeds a per-action maximum, or where the cumulative total over the stated period exceeds a cumulative maximum, the result MUST be FAIL using `RMSAIDEL-SCP-007`.

8. Human Checkpoints

8.1 Declaration

The authority scope MUST state, for each action type, whether a human checkpoint is required before the action may proceed, and where required, the identity or role of the approver. Where the declaration is absent for any action type, the result MUST be FAIL using `RMSAIDEL-HUM-001`.

8.2 Approval Evidence

Where a human checkpoint is required, each such action MUST carry approval evidence containing the approver identifier, the approval time, and a reference binding the approval to that specific action. The approval time MUST NOT be later than the action time. Where approval evidence is absent, unbound, or later than the action, the result MUST be FAIL using `RMSAIDEL-HUM-002`.

8.3 Blanket Approval Prohibition

A single approval MUST NOT be applied to more than one action, unless the scope expressly defines a batch checkpoint and states the batch composition rule and the maximum batch size. Where one approval is reused across actions outside such a definition, the result MUST be FAIL using `RMSAIDEL-HUM-003`.

8.4 Approver Independence from the Agent

The approver MUST be a human, or a system acting under a separate credential whose delegator differs from the agent under verification. **An agent MUST NOT approve its own checkpoint, and MUST NOT approve the checkpoint of an agent to which it has sub-delegated.** Where self-approval or downstream self-approval is found, the result MUST be FAIL using `RMSAIDEL-HUM-004`.

9. Revocation

9.1 Revocation Mechanism

The delegator MUST have a means of revoking a credential before its expiry, and that means MUST be disclosed. Where no revocation mechanism exists or it is not disclosed, the result MUST be FAIL using `RMSAIDEL-REV-001`.

9.2 Revocation Status Check

The agent or its runtime MUST check the revocation status of every credential it relies on, at an interval not exceeding the disclosed revocation propagation time. The check method MUST be disclosed. Where no status check occurs, the result MUST be FAIL using `RMSAIDEL-REV-002`.

9.3 Propagation Time

The operator MUST disclose the maximum revocation propagation time, expressed in seconds or minutes. **It MUST NOT exceed 3600 seconds.** Where no propagation time is disclosed, or the disclosed value exceeds that limit, the result MUST be FAIL using `RMSAIDEL-REV-003`.

9.4 Actions After Revocation

Where an action occurs after the revocation effective time plus the disclosed propagation time, and is not authorised by another credential in force, the result MUST be FAIL using `RMSAIDEL-REV-004`.

9.5 Expired Credentials

Where an action occurs after a credential's expiry time and is not authorised by another credential in force, the result MUST be FAIL using `RMSAIDEL-REV-005`.

9.6 Revocation Propagation Down the Chain

Revocation of a credential MUST invalidate every credential sub-delegated from it, from the same effective time. Where a downstream credential remains in force after its upstream credential is revoked, the result MUST be FAIL using `RMSAIDEL-REV-006`.

10. Sub-delegation

10.1 Permission to Sub-delegate

An agent MUST NOT sub-delegate unless the credential from which its authority derives expressly permits sub-delegation. Where sub-delegation occurs without that permission, the result MUST be FAIL using `RMSAIDEL-DEL-001`.

10.2 Scope Monotonicity

A sub-delegated authority scope MUST be a subset of the scope from which it derives: no action type, counterparty, or quantitative limit may exceed the upstream scope, at any level. Where any downstream element exceeds its upstream element, the result MUST be FAIL using `RMSAIDEL-DEL-002`.

10.3 Chain Depth

The credential MUST state the maximum permitted delegation depth. **Depth MUST NOT exceed 3 levels below the root delegator.** Where the stated maximum is absent or the actual chain exceeds the limit, the result MUST be FAIL using `RMSAIDEL-DEL-003`.

10.4 Chain Verifiability

Every credential in the delegation chain MUST be independently retrievable and its signature independently verifiable by the verifier, without relying on an assertion by any intermediate party. Where any credential in the chain cannot be retrieved or verified, the result MUST be FAIL using `RMSAIDEL-DEL-004`.

10.5 Chain Validity Window

The validity period of a sub-delegated credential MUST NOT extend beyond the expiry of any credential above it in the chain. Where a downstream expiry exceeds an upstream expiry, the result MUST be FAIL using `RMSAIDEL-DEL-005`.

10.6 Circularity Prohibition

A delegation chain MUST NOT contain the same agent instance identifier more than once. Where a cycle is found, the result MUST be FAIL using `RMSAIDEL-DEL-006`.

11. Action Log

11.1 Completeness

Every action initiated by the agent instance within the assessment window MUST be recorded. Each entry MUST contain: an entry identifier; the action time; the action type; the counterparty; the quantity and unit where applicable; the authorization identifier relied on; the outcome; and, where a human checkpoint applies, the approval reference. Where an entry is absent or any field is missing, the result MUST be FAIL using `RMSAIDEL-LOG-001`.

11.2 Tamper Evidence

The action log MUST be constructed such that removal or alteration of any entry is detectable, by a hash chain in which each entry commits to the digest of the previous entry, or by an equivalent construction. Where no such construction exists, or the chain is discontinuous, the result MUST be FAIL using `RMSAIDEL-LOG-002`.

11.3 Anchoring

The log digest MUST be anchored to an external system at an interval not exceeding **24 hours**, such that the log state at each anchoring point cannot be retroactively altered by the operator alone.

Anchoring MUST cover the assessment window continuously, and the most recent anchor MUST be no earlier than **24 hours** before the verification point. **Log entries after the most recent anchor are not anchored**, and any action recorded only after that anchor MUST be treated as unverifiable for the purposes of §11.2 and reported under §16.6.

Where no anchoring occurs, an interval within the window exceeds 24 hours, or the most recent anchor is earlier than 24 hours before the verification point, the result MUST be FAIL using `RMSAIDEL-LOG-003`.

11.4 Counterparty Reconciliation

For each action type where the invoked external system produces its own record, the log MUST reconcile to that record. The verifier MUST obtain the counterparty record for a sample of actions and compare it to the log. **Where the counterparty record shows an action absent from the log, the result MUST be FAIL using** `RMSAIDEL-LOG-004`, and the log MUST NOT be treated as complete.

11.5 Retention

The action log MUST be retained for at least **7 years** from the action time, unless applicable law requires longer. Where the retention period is insufficient or cannot be demonstrated, the result MUST be FAIL using `RMSAIDEL-LOG-005`.

11.6 Assessment Window

The assessment window MUST cover at least the **90 days** preceding the verification point, or the entire period since the credential's effective time where that is shorter. **A conclusion MUST NOT be stated for a period outside the assessment window.** Where the window is shorter than required, the result MUST be FAIL using `RMSAIDEL-LOG-006`.

11.7 Real-Time Recording

The log entry for an action SHOULD be written before or at the time the action is initiated, rather than reconstructed afterwards.

12. Agent Identity and Attribution

12.1 Instance Identifier Uniqueness

The agent instance identifier MUST be globally unique and MUST NOT be reused for a different deployment. Where an identifier is reused, the result MUST be FAIL using `RMSAIDEL-IDN-001`.

12.2 Action Attribution

Every action in the log MUST be attributable to the agent instance under verification, by an identifier or signature carried with the action. **An action that cannot be distinguished from an action by a human or by another agent MUST NOT be counted as an action of this agent instance**, and its presence in the log without such attribution means the result MUST be FAIL using `RMSAIDEL-IDN-002`.

12.3 Configuration Change Record

Every change to the agent configuration MUST be recorded with the previous digest, the new digest, the change time, and the authorising party. Where a configuration change is unrecorded, the result MUST be FAIL using `RMSAIDEL-IDN-003`.

12.4 Impersonation Prohibition

The agent MUST NOT present itself to a counterparty as a human, and MUST NOT act under an identifier belonging to the delegator in a manner that conceals that an agent is acting. Where the scope permits acting in the delegator's name, that permission MUST be explicit in the credential. Where concealment occurs without explicit permission, the result MUST be FAIL using `RMSAIDEL-IDN-004`.

13. Material Events

13.1 Types of Material Event

A material event includes at least: an out-of-scope action; a breach of a quantitative limit; an action after revocation or expiry; compromise of a delegator signing key or an agent credential; a configuration change not covered by a credential; discovery of an unrecorded action; failure of counterparty reconciliation; loss of log continuity; sub-delegation exceeding upstream scope; and any other event that changes the conformance conclusion.

13.2 Notice Deadline

The operator MUST record a material event within **24 hours** of confirming it, and MUST notify the delegator within **24 hours** of confirming it. Where either deadline is not met, the result MUST be FAIL using `RMSAIDEL-EVT-001`.

13.3 Notice Content

```
{
  "event_id": "",
  "event_type": "",
  "agent_instance_id": "",
  "authorization_id": "",
  "affected_actions": [],
  "detected_at": "",
  "recorded_at": "",
  "notified_at": "",
  "description": "",
  "containment": "",
  "evidence_id": "",
  "requires_reverification": true
}
```

13.4 Verification Status Handling

Where a material event may affect conformance, the existing verification result MUST be treated as expired from the time of the event, and re-verification MUST be performed before the result is relied upon again.

14. Evidence Requirements

14.1 Required Evidence

Evidence ID	Evidence Type	Provided By	Required
EVD-001	Delegator identity record	Delegator; registry or trusted identity system	Mandatory
EVD-002	Operator identity record	Operator; company registry	Mandatory
EVD-003	Authorization credential original, with signature	Delegator	Mandatory
EVD-004	Delegator public key or certificate, and its location	Delegator	Mandatory
EVD-005	Agent configuration snapshot: system prompt and tool inventory	Operator	Mandatory
EVD-006	Authority scope in machine-testable form	Delegator	Mandatory
EVD-007	Action log for the assessment window	Operator	Mandatory
EVD-008	Log hash-chain continuity proof	Operator; log service	Mandatory
EVD-009	Log anchoring records	Anchoring party	Mandatory
EVD-010	Counterparty records for the reconciliation sample	External systems invoked	Mandatory
EVD-011	Revocation mechanism disclosure and status-check records	Delegator; operator	Mandatory
EVD-012	Human checkpoint approval evidence	Approver	Mandatory where a checkpoint applies
EVD-013	Every upstream credential in the delegation chain	Upstream delegators	Mandatory where sub-delegation exists
EVD-014	Configuration change records	Operator	Mandatory where a change occurred in the window

Evidence ID	Evidence Type	Provided By	Required
EVD-015	Material event record and notification	Operator	Mandatory where a material event has occurred
EVD-016	Retention policy proof	Operator	Mandatory
EVD-017	Machine-readable conformance record	Operator	Mandatory
EVD-018	Human disclosure version	Operator	Mandatory
EVD-019	Verifier digital signature and key information	Verifier	Mandatory

Evidence marked `Mandatory` MUST be present for every verification. Evidence marked `Mandatory where …` MUST be present only when the stated condition holds; where the condition does not hold, the verifier MUST record it as `NOT_APPLICABLE` with the reason.

14.2 Evidence Metadata

All mandatory evidence MUST contain the source, the generation time, the acquisition time, the content digest, and access location information. Where any of these is absent, the result MUST be FAIL using `RMSAIDEL-EVD-001`.

```
{
  "evidence_id": "",
  "evidence_type": "",
  "provider_id": "",
  "provider_independence_disclosed": false,
  "generated_at": "",
  "acquired_at": "",
  "uri": "",
  "sha256": "",
  "applies_to_authorization_id": ""
}
```

14.3 Evidence Digests

All evidence files and machine records MUST use SHA-256 digests, expressed as 64-character lowercase hexadecimal strings. Where a digest does not match or the format is incorrect, the result MUST be FAIL using `RMSAIDEL-EVD-002`.

14.4 Evidence Freshness and Binding

- the agent configuration snapshot (EVD-005) MUST be taken at the verification point;
- the action log (EVD-007) MUST cover the assessment window and MUST be closed no earlier than **24 hours** before the verification point;
- the hash-chain continuity proof (EVD-008) MUST be generated no earlier than **24 hours** before the verification point;
- log anchoring records (EVD-009) MUST include an anchor no earlier than **24 hours** before the verification point, and MUST cover the assessment window continuously;
- counterparty records (EVD-010) MUST be obtained no earlier than **35 days** before the verification point;
- revocation status-check records (EVD-011) MUST cover the assessment window;
- every item of evidence MUST state the authorization identifier to which it applies. **Evidence applying to a different authorization MUST NOT be used.**

Age is computed in days or hours, as the verification time minus the evidence generation time, rounded down. Age MUST NOT be computed in calendar months.

Where an evidence item exceeds the stated maximum age, or applies to a different authorization, the result MUST be FAIL using `RMSAIDEL-EVD-003`. Expired evidence MUST NOT be reported as missing evidence.

14.5 Evidence Provider Independence

Counterparty records (EVD-010) and log anchoring records (EVD-009) MUST be obtained from a party other than the operator, and that party MUST disclose any ownership, control, common-control, or fee-dependency relationship with the operator or the delegator. Evidence provided by a party owned or controlled by the operator MUST NOT be counted as independent evidence and is treated as an operator statement under §4.1. Where the disclosure is absent or the provider is not independent, the result MUST be FAIL using `RMSAIDEL-EVD-004`.

14.6 Missing Evidence

Where any mandatory evidence is missing, the overall result MUST be `FAIL` or, where §16.6 applies, `DEFICIENT`. `CONDITIONAL` MUST NOT be used in its place.

14.7 Conflicting Evidence

Where evidence conflicts, the verifier MUST record both sides of the conflict, their source and time, stop the PASS determination for the affected requirements, and determine the overall result as FAIL until the conflict is resolved.

14.8 Evidence Authority Ordering

Evidence sources are ordered by authority as follows, from highest to lowest:

1. a record obtained by the verifier directly from an invoked external system;
2. an anchoring record published to an external system;
3. a credential signed by a delegator;
4. a log or record produced by the operator;
5. a statement by the operator or an output of the agent itself.

This ordering MUST be recorded for each item of conflicting evidence and MUST be used when reporting a conflict. It MUST NOT be used to resolve a conflict automatically: §14.7 continues to apply.

A statement by the operator or an output of the agent MUST NOT be the sole evidence for any requirement of this standard.

15. Verification Procedure

15.1 Verification Sequence

The verifier MUST execute in the following order:

1. confirm the standard identifier, version, and machine identifier;
2. confirm the delegator, the operator, and the `delegator_is_operator` declaration;
3. confirm the agent instance identifier and its non-reuse;
4. confirm the verification point in UTC and the assessment window, and freeze the evidence set;
5. retrieve the authorization credential and verify its required fields;
6. verify the delegator signature over the credential, including the scope;
7. verify the credential validity period and the 365-day maximum;
8. compute the agent configuration digest and compare it to the digest bound in the credential;
9. verify that the credential binds to this agent instance and to no other;
10. verify that the authority scope is machine-testable and enumerates action types;
11. verify tool inventory consistency: every externally effective tool is in scope or disabled;

12 verify that every quantity-bearing action type carries a per-action and a cumulative maximum;

13 verify counterparty restrictions for every action type;

14 verify human checkpoint declarations for every action type;

15 retrieve the action log and verify hash-chain continuity;

16 verify log anchoring records and the anchoring interval;

17 obtain counterparty records for the reconciliation sample and compare them to the log;

18 replay every action in the assessment window against the scope in force at the time of that action;

19 compute cumulative totals per action type and test them against cumulative maxima;

20 verify approval evidence for every action requiring a human checkpoint, including binding and ordering;

21 test for blanket approval and for self-approval;

22 verify the revocation mechanism, the status-check records, and the disclosed propagation time;

23 test for actions occurring after revocation or expiry;

24 where sub-delegation exists, retrieve every upstream credential and verify its signature independently;

25 verify scope monotonicity, chain depth, chain validity windows, and absence of circularity down the chain;

26 verify action attribution and configuration change records;

27 verify material event records and notice deadlines;

28 verify the retention policy;

29 verify the source, generation time, acquisition time, age, authorization binding, digest, and provider independence of all mandatory evidence;

30 verify consistency between the machine record and the human disclosure;

31 recompute all SHA-256 digests;

32 compute each requirement result and record the evidence used, the verification step, and any failure code;

33 compute the overall result under §16;

34 sign the verification record and record the signing time.

15.2 Determinism Requirement

The verification procedure MUST use fixed rules. Requirements MUST NOT be lowered based on the operator's size, brand, model provider, or reputation.

15.3 Individual Result

Each requirement MUST record an applicability of `APPLICABLE` or `NOT_APPLICABLE`, and each applicable requirement MUST produce one of `PASS`, `FAIL`, or `NOT_ASSESSED`. Where the result is `FAIL`, a failure code from §17 MUST be recorded.

15.4 NOT_APPLICABLE

A requirement may be marked `NOT_APPLICABLE` only where this standard expressly permits and the verifier records the reason. It MUST NOT be used to bypass a generally applicable MUST requirement.

15.5 Re-computability

The verification record MUST contain, for every applicable requirement, the result, the evidence used, the verification step, and any failure code, such that the overall result can be re-computed by a second conforming implementation. Where the result cannot be re-computed, the result MUST be FAIL using `RMSAIDEL-VER-001`.

15.6 Verification Signature

The verification record MUST be signed by the verifier using a verifiable digital signature, and MUST include the signing time, the signature algorithm, the key identifier, and the certificate or public key location. Where the signature is missing, invalid, or cannot be verified, the result MUST be FAIL using `RMSAIDEL-SIG-001`.

16. Conformance Determination

16.1 PASS

The overall result may be PASS only when all of the following are simultaneously satisfied:

1. all applicable MUST requirements are PASS;
2. no MUST NOT is violated;
3. a signed authorization credential is in force, bound to this agent instance and to the configuration digest observed at the verification point;
4. the authority scope is machine-testable and enumerates permitted action types;
5. every action in the assessment window passes the scope test against a credential in force at the time of that action;
6. no quantitative limit is breached;
7. every action requiring a human checkpoint carries bound, correctly ordered approval evidence;
8. no action occurs after revocation or expiry;
9. every credential in the delegation chain is independently verifiable and scope-monotonic;
10. the action log is continuous, anchored, and reconciles to the counterparty sample;
11. all mandatory evidence is present, valid, within its freshness limits, and bound to this authorization;
12. the human disclosure is consistent with the machine record;
13. there are no unresolved evidence conflicts;
14. the result can be re-computed by a second conforming implementation;
15. the verification record is signed by the verifier with a valid digital signature.

16.2 CONDITIONAL

The overall result may be CONDITIONAL only where all MUST and MUST NOT requirements are satisfied, all mandatory evidence is valid, and one or more SHOULD requirements are unsatisfied.

16.3 FAIL

Except where §16.6 applies, the overall result MUST be FAIL where any of the following occurs:

- any applicable MUST is unsatisfied;
- any MUST NOT is violated;
- no authorization credential is in force;
- the agent configuration digest differs from the digest bound in the credential;
- a credential is presented for more than one agent instance;
- the authority scope is not machine-testable, or grants by exclusion;
- any action fails the scope test against every credential in force at the time of that action;
- a per-action or cumulative limit is breached;

- an action requiring a human checkpoint lacks bound approval, or the approval post-dates the action;
- one approval is reused across actions outside a defined batch checkpoint;
- an agent approves its own checkpoint or that of a downstream agent;
- an action occurs after revocation effective time plus propagation time, or after expiry;
- a sub-delegated scope exceeds its upstream scope, or the chain contains a cycle;
- the log hash chain is discontinuous, or a counterparty record shows an action absent from the log;
- mandatory evidence is missing, expired, bound to a different authorization, digest-mismatched, or provided by a party controlled by the operator;
- the human disclosure conflicts with the machine record;
- a second verification implementation cannot re-compute the same result;
- the verification record is unsigned or the signature is invalid.

16.4 NOT ASSESSED

`NOT ASSESSED` is used where verification is not formally complete, the evidence set has not been frozen, the verifier has not signed, or the subject falls within §1.2. It does not constitute a conformance conclusion.

16.5 Result Validity Period

Each verification result MUST have an expiry time. The expiry time MUST NOT be later than the earliest of:

1. **90 days** after the verification point;
2. the expiry time of the authorization credential;
3. the expiry of any mandatory evidence under §14.4.

The result expires immediately, regardless of the stated expiry time, upon any of: a change to the agent configuration digest; revocation of the credential or of any credential above it in the chain; a change of operator or delegator; or a material event under §13.

A result under this standard describes the authority state at the verification point and the recorded history within the assessment window. **It MUST NOT be presented as a statement about future agent behaviour.** Where the declared expiry exceeds the limits above, the result MUST be FAIL using `RMSAIDEL-VER-002`.

16.6 DEFICIENT

`DEFICIENT` is used where **no requirement is affirmatively violated on the evidence available, but one or more mandatory items cannot be verified** because the required evidence is missing, incomplete, or expired, and the gap is curable by supplying that evidence.

`DEFICIENT` MUST be used, in preference to `FAIL`, where all of the following hold:

1. every requirement that could be evaluated on the available evidence is satisfied;
2. one or more mandatory items could not be evaluated because evidence was absent, incomplete, or beyond the freshness limits of §14.4;
3. no MUST NOT is violated and no evidence conflict under §14.7 is present.

Where any requirement is affirmatively violated on the available evidence, the result MUST be `FAIL`, not `DEFICIENT`. Where both a violation and a gap are present, `FAIL` takes precedence.

A `DEFICIENT` result MUST list every item that could not be verified and state, for each, the evidence that would resolve it.

Assessment from public information only. Where an assessment is not based on an evidence set submitted by the operator, but on publicly available information alone, the result MUST NOT be reported as `FAIL` on the ground that evidence is absent. Items that cannot be verified from the available information MUST be reported as `DEFICIENT` and identified individually. Such an assessment MUST state that the operator did not participate.

`DEFICIENT` states only that specified items could not be verified within the scope assessed. It is not a statement about the quality, soundness, or good faith of the operator or the agent.

17. Failure Codes

Code	Meaning
RMSAIDEL-ID-001	Conforming subject identifier incomplete or not unique
RMSAIDEL-ID-002	Delegator or operator cannot be confirmed, or `delegator_is_operator` not declared
RMSAIDEL-AUT-001	No authorization credential in force
RMSAIDEL-AUT-002	Credential missing a required field
RMSAIDEL-AUT-003	Credential signature absent, unverifiable, or not covering the scope
RMSAIDEL-AUT-004	Credential open-ended or validity period exceeding 365 days
RMSAIDEL-AUT-005	Agent configuration digest differs from the digest bound in the credential
RMSAIDEL-AUT-006	Credential presented for more than one agent instance
RMSAIDEL-SCP-001	Authority scope not machine-testable
RMSAIDEL-SCP-002	Scope granted by exclusion rather than enumeration
RMSAIDEL-SCP-003	Externally effective tool neither in scope nor disabled
RMSAIDEL-SCP-004	Quantity-bearing action type lacking a per-action or cumulative maximum
RMSAIDEL-SCP-005	Counterparty restriction absent for an action type
RMSAIDEL-SCP-006	Action outside the scope of every credential in force at the time
RMSAIDEL-SCP-007	Per-action or cumulative quantitative limit breached
RMSAIDEL-HUM-001	Human checkpoint declaration absent for an action type
RMSAIDEL-HUM-002	Approval evidence absent, unbound, or later than the action
RMSAIDEL-HUM-003	One approval reused across actions outside a defined batch checkpoint
RMSAIDEL-HUM-004	Agent approved its own checkpoint or that of a downstream agent
RMSAIDEL-REV-001	Revocation mechanism absent or undisclosed
RMSAIDEL-REV-002	Revocation status not checked at the required interval
RMSAIDEL-REV-003	Revocation propagation time undisclosed or exceeding 3600 seconds
RMSAIDEL-REV-004	Action after revocation effective time plus propagation time
RMSAIDEL-REV-005	Action after credential expiry
RMSAIDEL-REV-006	Downstream credential in force after upstream revocation
RMSAIDEL-DEL-001	Sub-delegation without express permission
RMSAIDEL-DEL-002	Sub-delegated scope exceeds upstream scope
RMSAIDEL-DEL-003	Maximum delegation depth absent, or chain exceeding 3 levels
RMSAIDEL-DEL-004	Credential in the chain not independently retrievable or verifiable
RMSAIDEL-DEL-005	Downstream validity period extending beyond an upstream expiry
RMSAIDEL-DEL-006	Delegation chain contains a cycle
RMSAIDEL-LOG-001	Action log entry absent or missing a required field
RMSAIDEL-LOG-002	Log lacks tamper evidence, or the hash chain is discontinuous
RMSAIDEL-LOG-003	Log anchoring absent or interval exceeding 24 hours
RMSAIDEL-LOG-004	Counterparty record shows an action absent from the log
RMSAIDEL-LOG-005	Log retention period insufficient or not demonstrable
RMSAIDEL-LOG-006	Assessment window shorter than required

Code	Meaning
RMSAIDEL-IDN-001	Agent instance identifier reused
RMSAIDEL-IDN-002	Action not attributable to the agent instance under verification
RMSAIDEL-IDN-003	Configuration change unrecorded
RMSAIDEL-IDN-004	Agent concealed its nature or acted in the delegator's name without permission
RMSAIDEL-EVT-001	Material event not recorded or notified within 24 hours
RMSAIDEL-DAT-001	Human disclosure inconsistent with machine record
RMSAIDEL-EVD-001	Mandatory evidence missing
RMSAIDEL-EVD-002	Evidence digest mismatch or incorrect format
RMSAIDEL-EVD-003	Evidence exceeds maximum age or is bound to a different authorization
RMSAIDEL-EVD-004	Evidence provider independence undisclosed, or provider controlled by the operator
RMSAIDEL-SIG-001	Digital signature missing, invalid, or unverifiable
RMSAIDEL-VER-001	Verification result cannot be re-computed
RMSAIDEL-VER-002	Result validity period exceeds the limits of §16.5

18. Machine Interface

18.1 Conformance Record

```
{
  "record_id": "",
  "schema_version": "1.0",
  "verification_point": "",
  "assessment_window_start": "",
  "result_expires_at": "",
  "digest_algorithm": "sha-256",
  "standard": {},
  "conformity_subject": {},
  "delegator": {},
  "operator": {},
  "agent": {},
  "authorization": {},
  "authority_scope": {},
  "delegation_chain": [],
  "revocation": {},
  "action_log": {},
  "actions": [],
  "human_checkpoints": [],
  "material_events": [],
  "evidence": [],
  "requirement_results": [],
  "human_disclosure": {},
  "overall_result": {},
  "verifier": {},
  "signature": {}
}
```

18.2 Authority Scope Representation

The authority scope MUST be represented as an array of permitted action types. Each entry MUST carry the action type identifier, the counterparty rule, the human checkpoint requirement, and, where the action type bears a quantity, the per-action and cumulative maxima.

```
{
  "action_type": "PAYMENT",
  "counterparties": {"mode": "ENUMERATED", "values": ["merchant:A", "merchant:B"]},
  "human_checkpoint": {"required": true, "approver_role": "finance_manager"},
  "quantity": {
    "unit": "USD",
```

```
      "decimals": 2,
      "per_action_max": "500.00",
      "cumulative_max": "5000.00",
      "cumulative_period_hours": 720
    }
  }
```

`counterparties.mode` MUST be one of `ENUMERATED`, `RULE`, or `UNRESTRICTED`. `UNRESTRICTED` MUST be stated explicitly and MUST NOT arise from omission.

18.3 Quantity Format

All quantities MUST be expressed as decimal strings with the unit named and the number of decimals stated. Binary floating-point values MUST NOT be used.

18.4 Time Format

All times MUST use ISO 8601 UTC format with a precision of at least one second. The `verification_point` MUST be a single value used throughout the record.

18.5 Action Record

```
{
  "entry_id": "",
  "previous_entry_sha256": "",
  "action_time": "",
  "action_type": "",
  "counterparty": "",
  "quantity": "",
  "unit": "",
  "authorization_id": "",
  "approval_reference": null,
  "outcome": "",
  "agent_attribution": ""
}
```

`previous_entry_sha256` MUST be present for every entry except the first, and MUST equal the digest of the preceding entry. `authorization_id` MUST identify a credential in force at `action_time`. `agent_attribution` MUST identify the agent instance.

18.6 Human–Machine Consistency

The delegator, the operator, the agent instance identifier, the agent configuration digest, the authorization identifier, the authority scope, and the overall result MUST be consistent between the human disclosure and the machine record. Where they conflict, the result MUST be FAIL using `RMSAIDEL-DAT-001`.

18.7 Canonicalisation and Digest

Before signing, a machine record MUST use a deterministic canonicalisation procedure and generate a SHA-256 digest.

18.8 Digest Algorithm Identifier

The machine record MUST record the digest algorithm identifier. The default value is `sha-256`. A verifier MUST NOT use an unregistered digest algorithm to produce a formal verification record.

19. Security Considerations

19.1 Principal Threats

This standard considers at least: an agent acting with no credential; a credential reused across agent instances; a configuration changed after issuance so that the same credential authorises different behaviour; a scope written in natural language so that no action can be shown to be out of scope; a scope granted by exclusion so that unanticipated action types are permitted by default; a tool present in the configuration but absent from the scope; limits stated per action but not cumulatively, permitting many small actions; approval obtained after the action; one approval reused across many actions; an agent approving its own checkpoint; action continuing after revocation; sub-delegation exceeding upstream authority; a delegation cycle; actions omitted from the log; a log rewritten before verification; a log never reconciled to counterparty records; an agent presenting itself as a human; and verifier conflict of interest.

19.2 Security Controls

The operator and the verifier MUST verify credential signatures independently, bind credentials to configuration digests, evaluate scope deterministically, replay the action log against scope, reconcile a sample against counterparty records, verify hash-chain continuity and anchoring, and sign the verification record.

19.3 Verifier Independence

The verifier MUST disclose commercial, fee, ownership, control, and other interest relationships with the operator or the delegator that may affect independence. An undisclosed material interest relationship renders the verification result invalid.

19.4 Residual Risk

Even where the result is PASS, there may still be actions after the verification point that exceed authority, a compromise of a signing key after the verification point, a counterparty system that produced no record and so could not be reconciled, an action within scope that nonetheless causes harm, a scope that was granted too widely, or concealment by an evidence provider.

A PASS describes the authority state at a point in time and the recorded history within the assessment window. It does not establish that the agent will remain within its authority afterwards.

19.5 Threat Mapping Matrix

Threat	Control Requirement	Evidence	Failure Code
Agent acting with no credential	§4.2, §6.1	EVD-003	RMSAIDEL-AUT-001
Credential reused across instances	§6.6	EVD-003	RMSAIDEL-AUT-006
Configuration changed after issuance	§4.5, §6.5	EVD-005, EVD-014	RMSAIDEL-AUT-005, RMSAIDEL-IDN-003
Scope only in natural language	§4.4, §7.1	EVD-006	RMSAIDEL-SCP-001
Scope granted by exclusion	§7.2	EVD-006	RMSAIDEL-SCP-002
Tool outside the scope	§7.3	EVD-005, EVD-006	RMSAIDEL-SCP-003
Many small actions evading a per-action cap	§7.4, §7.7	EVD-006, EVD-007	RMSAIDEL-SCP-007
Approval obtained after the action	§8.2	EVD-012	RMSAIDEL-HUM-002
One approval reused across actions	§8.3	EVD-012	RMSAIDEL-HUM-003
Agent approving its own checkpoint	§8.4	EVD-012, EVD-013	RMSAIDEL-HUM-004
Action continuing after revocation	§9.3, §9.4	EVD-007, EVD-011	RMSAIDEL-REV-004
Sub-delegation exceeding upstream authority	§4.7, §10.2	EVD-013	RMSAIDEL-DEL-002
Delegation cycle	§10.6	EVD-013	RMSAIDEL-DEL-006
Actions omitted from the log	§4.8, §11.4	EVD-007, EVD-010	RMSAIDEL-LOG-004
Log rewritten before verification	§11.2, §11.3	EVD-008, EVD-009	RMSAIDEL-LOG-002, RMSAIDEL-LOG-003

Threat	Control Requirement	Evidence	Failure Code
Agent presenting itself as human	§12.4	EVD-010	RMSAIDEL-IDN-004
Evidence provider controlled by the operator	§14.5, §4.1	EVD-009, EVD-010	RMSAIDEL-EVD-004
Result presented as future assurance	§16.5, §19.4	EVD-017	RMSAIDEL-VER-002
Verifier conflict of interest	§1.5, §19.3	—	verification result invalid

20. Conformance Claim

20.1 Minimum Fields

A conformance claim MUST include at least the standard identifier and version, the delegator, the operator, the `delegator_is_operator` declaration, the agent instance identifier, the agent configuration digest, the authorization identifier, the verification point, the assessment window, the result, the result expiry, the evidence set digest, the verifier identity, the signature, and any failure codes.

20.2 Example

```
{
  "standard_id": "RM-S-AI-DEL-001",
  "standard_version": "v1.0-F",
  "delegator_id": "entity:example-principal",
  "operator_id": "entity:example-operator",
  "delegator_is_operator": false,
  "agent_instance_id": "agent:example:0001",
  "agent_config_sha256": "",
  "authorization_id": "auth:example:0001",
  "verification_point": "2026-01-01T00:00:00Z",
  "assessment_window_start": "2025-10-03T00:00:00Z",
  "result_expires_at": "2026-04-01T00:00:00Z",
  "verification_result": "PASS",
  "evidence_set_sha256": "",
  "verifier_id": "",
  "signature": ""
}
```

20.3 Prohibited Statements

A conformance claim MUST NOT use expressions such as "fully autonomous", "safe agent", "aligned agent", "cannot exceed its authority", "human-in-the-loop" without naming the checkpoints, "completely safe", "zero risk", "government approved", "RuleMark guaranteed", or "permanently valid".

A conformance claim MUST NOT be presented as a statement about future agent behaviour, and MUST NOT be presented as covering a period outside the assessment window.

21. Test Cases

21.1 PASS Case

A signed credential is in force, bound to this agent instance and to the configuration digest observed at the verification point; the scope enumerates two action types with counterparty lists, per-action and cumulative maxima, and checkpoint declarations; every action in the 90-day window passes the scope test; cumulative totals are within limits; every checkpoint action carries a bound approval that precedes it; no action follows revocation or expiry; there is no sub-delegation; the log hash chain is continuous, anchored within 24 hours, and reconciles to the counterparty sample; all mandatory evidence is present, fresh, bound to this authorization, and independently provided; the record is signed.

Expected result: PASS

21.2 Configuration Changed After Issuance

All other requirements are satisfied; the configuration digest observed at the verification point differs from the digest bound in the credential, because the tool inventory was extended after issuance.

Expected result: FAIL, RMSAIDEL-AUT-005

21.3 Out-of-Scope Action

The scope permits PAYMENT to two enumerated merchants; the log contains one PAYMENT to a third merchant not in the list.

Expected result: FAIL, RMSAIDEL-SCP-006

21.4 Cumulative Limit Evaded

Every action is below the per-action maximum of 500.00 USD; the cumulative total over the stated 720-hour period is 7,400.00 USD against a cumulative maximum of 5,000.00 USD.

Expected result: FAIL, RMSAIDEL-SCP-007

21.5 Approval After the Action

An action requiring a human checkpoint carries approval evidence whose approval time is 40 minutes after the action time.

Expected result: FAIL, RMSAIDEL-HUM-002

21.6 Sub-delegation Exceeding Upstream Scope

The upstream credential permits PAYMENT up to 500.00 USD per action; a sub-delegated credential permits PAYMENT up to 2,000.00 USD per action.

Expected result: FAIL, RMSAIDEL-DEL-002

21.7 Action Absent From the Log

The counterparty record obtained from the invoked system shows an action that does not appear in the operator's log.

Expected result: FAIL, RMSAIDEL-LOG-004

21.8 SHOULD Not Satisfied

All MUST and MUST NOT requirements are satisfied and all mandatory evidence is valid; log entries are reconstructed after the fact rather than written at the time of the action, which is a SHOULD under §11.7.

Expected result: CONDITIONAL

21.9 Public Information Only

The assessment is based on the operator's published documentation alone: the agent instance and configuration digest are published and the credential format is documented; the credential original, the action log, anchoring records, and counterparty records are not published; the operator did not participate.

Expected result: `DEFICIENT`, with each unverifiable item listed individually

22. Requirement Traceability Matrix

Requirement Area	Clause	Core Evidence	Verification Step	Failure Code
Subject identity	§1.4, §5.1	EVD-001, EVD-002	2, 3	RMSAIDEL-ID-001
Delegator / operator separation	§1.5, §4.3, §5.3	EVD-001, EVD-002	2	RMSAIDEL-ID-002
Credential existence	§6.1	EVD-003	5	RMSAIDEL-AUT-001
Credential content	§6.2	EVD-003	5	RMSAIDEL-AUT-002
Credential signature	§6.3	EVD-003, EVD-004	6	RMSAIDEL-AUT-003
Validity period	§6.4	EVD-003	7	RMSAIDEL-AUT-004
Configuration binding	§4.5, §6.5	EVD-005	8	RMSAIDEL-AUT-005
Instance binding	§6.6	EVD-003	9	RMSAIDEL-AUT-006
Scope machine-testability	§4.4, §7.1	EVD-006	10	RMSAIDEL-SCP-001
Enumerated action types	§7.2	EVD-006	10	RMSAIDEL-SCP-002
Tool inventory consistency	§7.3	EVD-005, EVD-006	11	RMSAIDEL-SCP-003
Quantitative limits declared	§7.4	EVD-006	12	RMSAIDEL-SCP-004
Counterparty restriction	§7.5	EVD-006	13	RMSAIDEL-SCP-005
Out-of-scope actions	§7.6	EVD-006, EVD-007	18	RMSAIDEL-SCP-006
Limit breach	§7.7	EVD-006, EVD-007	19	RMSAIDEL-SCP-007
Checkpoint declaration	§8.1	EVD-006	14	RMSAIDEL-HUM-001
Approval evidence	§8.2	EVD-012	20	RMSAIDEL-HUM-002
Blanket approval	§8.3	EVD-012	21	RMSAIDEL-HUM-003
Self-approval	§8.4	EVD-012, EVD-013	21	RMSAIDEL-HUM-004
Revocation mechanism	§9.1	EVD-011	22	RMSAIDEL-REV-001
Revocation status check	§9.2	EVD-011	22	RMSAIDEL-REV-002
Propagation time	§9.3	EVD-011	22	RMSAIDEL-REV-003
Action after revocation	§9.4	EVD-007, EVD-011	23	RMSAIDEL-REV-004
Action after expiry	§9.5	EVD-003, EVD-007	23	RMSAIDEL-REV-005
Downstream revocation	§9.6	EVD-011, EVD-013	23	RMSAIDEL-REV-006
Sub-delegation permission	§10.1	EVD-013	24	RMSAIDEL-DEL-001
Scope monotonicity	§4.7, §10.2	EVD-013	25	RMSAIDEL-DEL-002
Chain depth	§10.3	EVD-013	25	RMSAIDEL-DEL-003
Chain verifiability	§10.4	EVD-013	24	RMSAIDEL-DEL-004
Chain validity window	§10.5	EVD-013	25	RMSAIDEL-DEL-005
Circularity	§10.6	EVD-013	25	RMSAIDEL-DEL-006
Log completeness	§11.1	EVD-007	15	RMSAIDEL-LOG-001
Tamper evidence	§11.2	EVD-008	15	RMSAIDEL-LOG-002
Anchoring	§11.3	EVD-009	16	RMSAIDEL-LOG-003
Anchor coverage to verification point	§11.3	EVD-008, EVD-009	16	RMSAIDEL-LOG-003
Counterparty reconciliation	§4.8, §11.4	EVD-010	17	RMSAIDEL-LOG-004
Retention	§11.5	EVD-016	28	RMSAIDEL-LOG-005
Assessment window	§11.6	EVD-007	4	RMSAIDEL-LOG-006
Instance uniqueness	§12.1	EVD-002	3	RMSAIDEL-IDN-001
Action attribution	§12.2	EVD-007, EVD-010	26	RMSAIDEL-IDN-002

Requirement Area	Clause	Core Evidence	Verification Step	Failure Code
Configuration change record	§12.3	EVD-014	26	RMSAIDEL-IDN-003
Impersonation	§12.4	EVD-010	26	RMSAIDEL-IDN-004
Material events	§13.2	EVD-015	27	RMSAIDEL-EVT-001
Evidence metadata	§14.2	EVD-001 to EVD-019	29	RMSAIDEL-EVD-001
Evidence digests	§14.3	EVD-017, EVD-019	31	RMSAIDEL-EVD-002
Evidence freshness and binding	§14.4	EVD-005, EVD-007, EVD-009, EVD-010	29	RMSAIDEL-EVD-003
Provider independence	§14.5	EVD-009, EVD-010	29	RMSAIDEL-EVD-004
Human–machine consistency	§18.6	EVD-018	30	RMSAIDEL-DAT-001
Re-computability	§15.5	EVD-017	32, 33	RMSAIDEL-VER-001
Result validity	§16.5	EVD-017	33	RMSAIDEL-VER-002
Verification signature	§15.6	EVD-019	34	RMSAIDEL-SIG-001
Real-time log writing (SHOULD)	§11.7	EVD-007, EVD-008	15	—

23. Lifecycle

23.1 Draft

In Draft status, this standard MUST NOT serve as a basis for formal RuleMark conformance.

23.2 Frozen

Once formally published, this standard may be referenced and implemented.

23.3 Frozen

A version may be frozen only after all of the following are satisfied:

- the automated consistency check reports zero errors;
- each MUST is mapped to evidence, a verification step, and a failure code;
- the machine schema accepts a conforming record and rejects non-conforming records;
- failure codes are stable;
- test cases pass;
- the human and machine versions are consistent;
- all normative references are confirmed to exist. This standard is self-contained; normative references are limited to the public technical specifications listed in §24.1;
- a SHA-256 digest has been generated for the normative files and the digest manifest has been signed by RuleMark;
- RuleMark sovereign approval has been recorded, stating the approver, the date, the approved version, and the approved digest.

Independent review by an external expert is recommended and, where performed, MUST be recorded with the reviewer's identity, the reviewed version, and the reviewed digest. It is not a condition of freezing.

23.4 Superseded

Where a new version replaces this version, it MUST record the new version number, effective date, compatibility changes, migration rules, and the last valid date of the old version.

23.5 Withdrawn

After withdrawal, this standard MUST NOT be used for new conformance claims. Historical verification records MUST continue to be retained.

24. References and Dependencies

24.1 Normative Technical References

- FIPS 180-4 (SHA-256);
- RFC 8785 JSON Canonicalization Scheme;
- RFC 8259 JSON;
- ISO 8601 date and time representation;
- JSON Schema Draft 2020-12.

24.2 RuleMark Dependencies

This standard is self-contained and does not depend on any external RuleMark standard. Canonical identity, evidence, verification and determination, digital signature, and lifecycle rules are defined internally at §5, §14, §15, §16, §15.6, and §23 respectively.

Where RuleMark later issues corresponding core-layer standards, a subsequent version of this standard MAY align these internal clauses with normative references to those standards. Until then, this standard MUST NOT cite an identifier that has not been issued.

24.3 Informative References

RFC 7519 (JWT), RFC 9421 (HTTP Message Signatures), and the W3C Verifiable Credentials Data Model 2.0 may be used as carrier formats for an authorization credential. RFC 3161 or an equivalent mechanism may be used for log anchoring. This standard does not require any particular carrier format, and the use of one does not by itself satisfy any requirement of this standard.

25. Implementation Requirements

An implementation conforming to this standard MUST be able to:

1. identify the delegator and the operator separately;
2. retrieve an authorization credential and verify its signature independently;
3. compute an agent configuration digest and compare it to the digest bound in a credential;
4. evaluate a scope test deterministically for any given action;
5. detect a scope expressed by exclusion rather than enumeration;
6. reconcile the tool inventory against the scope;
7. replay an action log against the scope in force at each action time;
8. compute cumulative totals over a stated period and test them against maxima;
9. verify approval binding and ordering, and detect blanket and self-approval;

10 check revocation status and detect actions after revocation or expiry;

11 traverse a delegation chain, verify each credential independently, and test scope monotonicity, depth, validity windows, and circularity;

12 verify log hash-chain continuity and anchoring intervals;

13 reconcile a sample of log entries against counterparty records;

14 verify evidence metadata, age, authorization binding, digests, and provider independence;

15 verify consistency between the machine record and the human disclosure;

16 produce a deterministic conformance result with per-requirement traceability;

17 expire the result upon a configuration change, revocation, or party change;

18 sign and retain the verification record;

19 allow a second conforming implementation to re-compute the same overall result.

26. Final Non-Claim

A PASS result under RM-S-AI-DEL-001 indicates only:

> At the specified verification point, under the specified standard version, for the specified delegator, agent instance, and authorization credential, and for the recorded actions within the specified assessment window, all applicable mandatory requirements of RM-S-AI-DEL-001 are satisfied.

PASS does not indicate that the agent will remain within its authority after the verification point, that the agent's outputs are correct or useful, that the underlying model is safe or aligned, that the authorization is legally effective in any jurisdiction, that the hosting infrastructure is secure, that actions within scope caused no harm, that the scope was granted wisely, regulatory approval, an investment recommendation, or a RuleMark guarantee.

Authority is not competence. A verified grant does not make an agent capable, and a capable agent without a verified grant is not within this standard.

Document Status Declaration

```
Standard ID: RM-S-AI-DEL-001
Version: v1.0-F
Status: FROZEN
Publication Date: 2026-07-23
Effective Date: 2026-07-23
Normative Status: NORMATIVE
Conformance Use: AUTHORIZED
Machine ID: rulemark:standard:rm-s-ai-del-001:v1.0-F
```

This standard MUST NOT be marked FROZEN before the automated consistency check reports zero errors, the machine schema has been tested against conforming and non-conforming records, the digest manifest has been signed by RuleMark, and RuleMark sovereign approval has been recorded.