

Stable Authority Boundary (SAB) v1.0 Conformance Specification

Normative Standard

Author: David Forbes

Publisher and Standards Custodian: BLOCK VECTOR Technologies, L.L.C.

Canonical Standard Identifier (CSI): SAB-STD-v1.0-DF-2026

Canonical Record Location:

https://blockvectortech.com/SAB_Conformance_Specification/SAB_Conformance_Specification.html

External Repository DOI: Not assigned at publication; see Canonical Record for supplemental archival identifiers.

Publication Date: 2026-08-14

License: Creative Commons Attribution 4.0 International (CC BY 4.0)

Abstract

This specification defines the normative requirements for implementing and assessing conformance to the **Stable Authority Boundary™ (SAB)**, an architectural invariant governing consequential execution.

A Stable Authority Boundary is **the point at which proposed execution is evaluated against recognized authority**. SAB requires an explicit and enforceable separation between **evidence and authority, evaluation and authorization, proposed state and authoritative state, and technical capability and permission to execute**. Evidence—including observations, measurements, human analysis, algorithmic results, model outputs, confidence scores, classifications, recommendations, and other evaluation products—may inform a decision but does not independently establish authority.

A conformant implementation permits consequential execution only when the authority applicable to that execution is identified, valid, within scope, and verifiable at the boundary. Where authority is absent, expired, revoked, ambiguous, degraded, outside scope, or otherwise unverifiable, the required system behavior is **Refusal/Hold**, not progress-by-default, fail-open execution, self-authorization, or implicit escalation.

This specification defines the required invariants, governance artifacts, conformance requirements, outcome states, and assessment criteria necessary to make SAB **implementable, testable, auditable, and independently assessable**. SAB is technology-neutral and applies wherever a human, machine, software system, hardware device, autonomous system, decision-support system, or hybrid workflow can directly or indirectly cause a consequential execution, authoritative state change, binding decision, resource commitment, enforcement action, irreversible action, or operational effect.

Scope is determined by consequence, not technology.

Contents

ABSTRACT	2
0. PROVENANCE, CANONICALITY, VERSION CONTROL, AND NAMING (NORMATIVE)	7
0.1 CANONICAL RELEASE — SAB-CAN-001	7
0.2 CANONICAL AUTHORITY AND CONFLICT RESOLUTION — SAB-CAN-002	7
0.3 EXTERNAL ARCHIVAL REPOSITORIES AND IDENTIFIERS — SAB-ARCH-001.....	8
0.4 REQUIRED CITATION — SAB-CITE-001	8
0.5 REDISTRIBUTION OF UNMODIFIED COPIES — SAB-DIST-001.....	9
0.6 DERIVATIVE WORKS AND ADAPTATIONS — SAB-NAME-001	9
0.7 IMPLEMENTATION PROFILES AND DOMAIN-SPECIFIC GUIDANCE — SAB-PROF-001.....	10
0.8 CONFORMANCE CLAIM VALIDITY — SAB-CONF-001	11
0.9 CANONICAL INTEGRITY — SAB-INTEG-001	11
0.10 VERSIONING AND SUPERSESSION — SAB-VER-001	12
0.11 EDITORIAL CORRECTIONS AND ERRATA — SAB-ERR-001.....	12
0.12 CUSTODIANSHIP — SAB-CUST-001	13
0.13 LICENSING, TRADEMARKS, AND PATENT RIGHTS — SAB-LIC-001	13
1. SCOPE (NORMATIVE)	14
1.1 APPLICABILITY	14
1.2 DIRECT AND INDIRECT EXECUTION	14
1.3 IMPLEMENTATION INDEPENDENCE	14
1.4 AUTHORITY PATH DETERMINES SCOPE.....	15
1.5 ADVISORY SYSTEMS	15
1.6 SYSTEMS OUTSIDE SCOPE	16
1.7 SCOPE PRINCIPLE.....	16
2. NORMATIVE LANGUAGE AND INTERPRETATION.....	17
2.1 PURPOSE.....	17
2.2 MANDATORY REQUIREMENTS	17
2.3 RECOMMENDED REQUIREMENTS.....	17
2.4 PERMITTED BEHAVIOR	18
2.5 CAPABILITY IS NOT NORMATIVE PERMISSION	18
2.6 DESCRIPTIVE AND FUTURE STATEMENTS.....	19
2.7 NORMATIVE AND NON-NORMATIVE CONTENT	19
2.8 EXAMPLES	19
2.9 FIGURES, TABLES, AND ARCHITECTURAL DIAGRAMS	20
2.10 REQUIREMENT IDENTIFIERS	20
2.11 NO IMPLIED EXCEPTIONS.....	20
2.12 INTERPRETATION UNDER AMBIGUITY.....	21
3. DEFINITIONS (NORMATIVE)	22
3.1 APPLICABILITY OF DEFINITIONS	22
3.2 AUTHORITY.....	22
3.3 RECOGNIZED AUTHORITY.....	22
3.4 AUTHORITY SOURCE.....	23
3.5 AUTHORITY PROOF	23

3.6 EVIDENCE	23
3.7 EVALUATION	24
3.8 PROPOSED EXECUTION	24
3.9 STABLE AUTHORITY BOUNDARY	24
3.10 BOUNDARY DECISION	25
3.11 EXECUTION	25
3.12 CONSEQUENTIAL EXECUTION	26
3.13 PROPOSED STATE	26
3.14 AUTHORITATIVE STATE	26
3.15 EXECUTION PATH	27
3.16 AUTHORITY-BEARING PATH	27
3.17 REFUSAL/HOLD	27
3.18 PROGRESS-BY-DEFAULT	27
3.19 FAIL-OPEN	28
3.20 OVERRIDE.....	28
3.21 SELF-AUTHORIZATION	28
3.22 SELF-ESCALATION	29
3.23 DEGRADED MODE	29
3.24 ESCALATION	29
3.25 EVALUATOR	29
3.26 AUTHORIZER	30
3.27 EXECUTOR	30
3.28 ENFORCEMENT MECHANISM	30
3.29 CONFORMANCE SCOPE	30
3.30 AMBIGUOUS AUTHORITY.....	30
3.31 NON-ACTION	31
3.32 LEGITIMACY	31
3.33 AUTHORITY MODEL	31
3.34 AUTHORITY SURFACE.....	31
3.35 AUTHORIZED EXECUTION.....	31
4. SAB CORE INVARIANTS (NORMATIVE)	32
4.1 STATUS OF THE CORE INVARIANTS	32
4.2 SAB-INV-001 — EVIDENCE INFORMS; AUTHORITY AUTHORIZES	32
4.3 SAB-INV-002 — REFUSAL IS CORRECT UNDER UNVERIFIABLE AUTHORITY.....	33
4.4 SAB-INV-003 — DEGRADED MODE TIGHTENS AUTHORITY.....	34
4.5 SAB-INV-004 — NON-ACTION IS A VALID STATE	35
4.6 RELATIONSHIP AMONG THE INVARIANTS	36
5. CONFORMANCE LEVELS (NORMATIVE)	37
5.1 PURPOSE OF CONFORMANCE LEVELS.....	37
5.2 CONFORMANCE IS ASSESSED WITHIN A DECLARED SCOPE	37
5.3 SAB-L1 — BASELINE CONFORMANCE	38
5.4 SAB-L2 — AUDITABLE CONFORMANCE.....	39
5.5 SAB-L3 — HARDENED CONFORMANCE.....	40
5.6 REQUIRED GOVERNANCE ARTIFACTS BY CONFORMANCE LEVEL	41
5.7 CONFORMANCE-LEVEL DECLARATION	42
5.8 NO PARTIAL-LEVEL CONFORMANCE	42

5.9 CONFIGURATION AND DEPLOYMENT DEPENDENCE.....	43
5.10 CONFORMANCE DOES NOT CREATE AUTHORITY.....	43
5.11 RELATIONSHIP BETWEEN CONFORMANCE LEVELS	43
6. CONFORMANCE ASSESSMENT OUTCOMES AND DECISION RULE (NORMATIVE)	44
6.1 PURPOSE.....	44
6.2 OUTCOME ASSIGNMENT	45
6.3 LEGITIMATE.....	45
6.4 AMBIGUOUS — REFUSAL REQUIRED.....	46
6.5 ILLEGITIMATE	47
6.6 REQUIREMENT-LEVEL OUTCOME AGGREGATION	47
6.7 OVERALL CONFORMANCE DECISION RULE — SAB-DEC-001	48
6.8 OPERATIONAL TREATMENT OF AMBIGUOUS AND ILLEGITIMATE CONDITIONS	48
6.9 CORRECTION AND REASSESSMENT	49
6.10 CONFORMANCE OUTCOMES DO NOT CREATE OPERATIONAL AUTHORITY	49
6.11 REQUIRED REPORTING	50
6.12 OUTCOME HIERARCHY.....	50
7. REQUIRED GOVERNANCE AND CONFORMANCE ARTIFACTS (NORMATIVE).....	52
7.1 PURPOSE.....	52
7.2 ARTIFACT CONTROL REQUIREMENTS	52
7.3 SAB-ART-01 — AUTHORITY STATEMENT.....	53
7.3.1 PURPOSE.....	53
7.3.2 REQUIRED CONTENT	53
7.3.3 COMPLETENESS REQUIREMENT	54
7.3.4 PROHIBITED ASSUMPTIONS	54
7.4 SAB-ART-02 — BOUNDARY MAP.....	55
7.4.1 PURPOSE.....	55
7.4.2 REQUIRED CONTENT	55
7.4.3 COMPLETENESS REQUIREMENT	55
7.4.4 ENFORCEMENT REQUIREMENT.....	56
7.5 SAB-ART-03 — REFUSAL POLICY.....	56
7.5.1 PURPOSE.....	56
7.5.2 REQUIRED REFUSAL CONDITIONS.....	56
7.5.3 REQUIRED REFUSAL BEHAVIOR	57
7.5.4 PROGRESS-BY-DEFAULT PROHIBITION.....	57
7.5.5 REFUSAL PERSISTENCE	58
7.6 SAB-ART-04 — ESCALATION AND OVERRIDE POLICY	58
7.6.1 PURPOSE.....	58
7.6.2 ESCALATION REQUIREMENTS	58
7.6.3 OVERRIDE REQUIREMENTS.....	59
7.6.4 PRESERVATION OF PRIOR DECISION	59
7.6.5 SELF-OVERRIDE PROHIBITION	59
7.6.6 OVERRIDE SCOPE.....	60
7.7 SAB-ART-05 — TRACEABILITY LEDGER	60
7.7.1 PURPOSE.....	60
7.7.2 REQUIRED TRACEABILITY	60
7.7.3 EVIDENCE INTEGRITY	61

7.7.4 TRACEABILITY COMPLETENESS	61
7.8 CROSS-ARTIFACT CONSISTENCY — SAB-CONS-001	62
7.9 ARTIFACT COMPLETENESS AND CONFORMANCE LEVEL	62
7.10 ARTIFACT MAINTENANCE AND CHANGE CONTROL	63
7.11 ARTIFACTS DO NOT CREATE AUTHORITY	63
8. CONFORMANCE REQUIREMENTS AND TESTS (NORMATIVE).....	64
8.1 TEST FRAMEWORK.....	64
8.2 APPLICABILITY BY CONFORMANCE LEVEL	64
8.3 SAB-REQ-01 — EXPLICIT BOUNDARY GATE	64
8.4 SAB-REQ-02 — AUTHORITY PROOF AT THE BOUNDARY	65
8.5 SAB-REQ-03 — REFUSAL UNDER UNVERIFIED AUTHORITY	66
8.6 SAB-REQ-04 — NO PROGRESS-BY-DEFAULT.....	67
8.7 SAB-REQ-05 — EXTERNAL OVERRIDES ONLY	67
8.8 SAB-REQ-06 — AUDIT RECONSTRUCTION	68
8.9 SAB-REQ-07 — DEGRADED-MODE TIGHTENING	69
8.10 SAB-REQ-08 — AUTHORITY SURFACE MINIMIZATION	69
8.11 SAB-REQ-09 — TERMINOLOGY AND CLAIMS HYGIENE	70
8.12 SAB-REQ-10 — REFUSAL OBSERVABILITY.....	71
8.13 SAB-L3 HARDENED ASSURANCE TESTS.....	72
8.14 TEST RECORD AND EVIDENCE PACKAGE.....	73
8.15 CORE INVARIANT VERIFICATION AND COVERAGE	73
9. CONFORMANCE STATEMENT AND CLAIM REQUIREMENTS (NORMATIVE).....	75
9.1 PURPOSE.....	75
9.2 REQUIRED CLAIM FIELDS	75
9.3 REQUIRED ARTIFACT REFERENCES.....	75
9.4 REQUIRED TEST OUTCOME SUMMARY.....	76
9.5 OVERALL CONFORMANCE DECLARATION	76
9.6 CLAIM SCOPE AND CONTINUING VALIDITY	76
APPENDIX A — BOUNDARY MAP RECORD TEMPLATE (NON-NORMATIVE).....	77
APPENDIX B — CONFORMANCE ASSESSMENT CHECKLIST (NORMATIVE)	78
APPENDIX C — CHANGE LOG (NORMATIVE)	81
APPENDIX D — REFERENCES (NON-NORMATIVE)	82

0. Provenance, Canonicity, Version Control, and Naming (Normative)

0.1 Canonical Release — SAB-CAN-001

The controlling canonical publication of **Stable Authority Boundary (SAB) v1.0 — Conformance Specification** SHALL be the release published and maintained by **BLOCK VECTOR Technologies, L.L.C.**, as identified by the Canonical Record Location and Canonical Standard Identifier (CSI) stated on the title page.

For purposes of this specification, the **Canonical Release** consists of:

1. the canonical publication record maintained by BLOCK VECTOR Technologies, L.L.C.; and
2. the normative released file identified by that record.

The canonical publication record MUST identify, at minimum:

- canonical title;
- SAB version;
- Canonical Standard Identifier (CSI);
- publication date;
- canonical record location;
- cryptographic hash algorithm;
- cryptographic hash of the normative released file;
- release status;
- and, where applicable, supersession information.

A file SHALL be considered an authentic copy of the Canonical Release only when its content corresponds to the identified release and its cryptographic hash matches the hash published in the canonical record.

The Canonical Standard Identifier for this release is:

SAB-STD-v1.0-DF-2026

The CSI MUST remain unchanged for this release.

0.2 Canonical Authority and Conflict Resolution — SAB-CAN-002

Where any difference exists between:

- the BLOCK VECTOR canonical publication;
- an archived copy;

- a redistributed copy;
- an external repository copy;
- a derivative work;
- an implementation guide;
- or any other representation of SAB v1.0,

the BLOCK VECTOR Canonical Release SHALL control.

External copies, citations, repository records, indexing services, mirrors, and archival deposits MUST NOT be treated as authoritative where they conflict with the Canonical Release.

Where uncertainty exists regarding the authenticity, version, or content of a purported SAB v1.0 copy, the canonical record and published cryptographic hash SHALL be used to resolve that uncertainty.

0.3 External Archival Repositories and Identifiers — SAB-ARCH-001

The Canonical Release MAY be deposited with Zenodo, institutional repositories, government repositories, standards archives, libraries, or other third-party archival services.

Such repositories MAY assign:

- Digital Object Identifiers (DOIs);
- accession numbers;
- archival identifiers;
- catalog identifiers;
- or other persistent identifiers.

These identifiers are **supplemental archival identifiers**.

They MUST NOT supersede:

- the BLOCK VECTOR canonical publication record;
- the Canonical Standard Identifier;
- the canonical version designation;
- or the cryptographic identity of the Canonical Release.

The absence, unavailability, modification, or loss of access to an external repository or identifier MUST NOT invalidate the Canonical Release.

0.4 Required Citation — SAB-CITE-001

Any document making a formal claim of conformance to SAB v1.0 MUST identify:

- the canonical standard title;

- the SAB version;
- the CSI;
- and the Canonical Record Location.

A DOI or other external archival identifier SHOULD also be cited when one has been assigned to the same Canonical Release, but such an identifier is not required to establish canonicity or conformance validity.

The required minimum citation is:

Stable Authority Boundary (SAB) v1.0 — Conformance Specification. BLOCK VECTOR Technologies, L.L.C. CSI: SAB-STD-v1.0-DF-2026. Canonical Record:
https://blockvectortech.com/SAB_Conformance_Specification/SAB_Conformance_Specification.html.

Where a DOI or equivalent archival identifier exists, it MAY be appended to this citation.

0.5 Redistribution of Unmodified Copies — SAB-DIST-001

For purposes of representing a redistributed file as an unmodified copy of SAB v1.0, the Canonical Release MAY be redistributed subject to the license governing the release.

A redistributed copy represented as **SAB v1.0** MUST:

- preserve the normative text without modification;
- preserve the canonical title;
- preserve the SAB version;
- preserve the CSI;
- preserve attribution;
- preserve the canonical record citation;
- and clearly identify itself as a copy of the Canonical Release.

A redistributed copy MUST NOT imply that the redistributor is the publisher, standards custodian, or source of canonical authority.

Any alteration of normative content causes the resulting document to cease being an unmodified copy of SAB v1.0.

0.6 Derivative Works and Adaptations — SAB-NAME-001

CC BY 4.0 permits creation and distribution of adapted material subject to the terms of that license. Nothing in this specification is intended to add a copyright restriction inconsistent with CC BY 4.0.

For purposes of SAB canonicity, naming, and formal conformance claims, a document that alters, extends, removes, substitutes, reinterprets, or otherwise modifies normative SAB v1.0 content is an adaptation and is not the Canonical Release.

An adaptation MUST NOT represent itself as the Canonical Release, an identical copy of the Canonical Release, or an official revision issued by the Standards Custodian unless it has actually been issued as such by BLOCK VECTOR Technologies, L.L.C.

An adaptation that makes a formal statement about its relationship to SAB v1.0 MUST identify SAB v1.0 as its source, identify the CSI and Canonical Record Location, indicate that normative content has been modified, distinguish original SAB requirements from modified or additional material, and avoid use of the canonical title in a manner that would reasonably imply canonical status.

The recommended notice for an adaptation is:

“Derived from Stable Authority Boundary (SAB) v1.0 — Conformance Specification, CSI SAB-STD-v1.0-DF-2026. This document is not the canonical SAB v1.0 standard.”

These provisions define SAB canonical status, naming, and the validity of SAB-related representations. They do not narrow the copyright permissions granted by CC BY 4.0.

0.7 Implementation Profiles and Domain-Specific Guidance — SAB-PROF-001

An implementation profile, application profile, reference architecture, implementation guide, domain mapping, test procedure, or sector-specific guidance MAY specialize the application of SAB v1.0 without becoming a derivative of the core standard, provided that it does not modify the normative meaning of the SAB invariants or requirements.

Such a document MAY:

- define domain-specific authority sources;
- identify domain-specific execution boundaries;
- specify additional evidence;
- define stricter controls;
- establish additional tests;
- provide implementation examples;
- or impose requirements beyond SAB v1.0.

Such a document MUST NOT:

- weaken a normative SAB requirement;
- convert a MUST into a MAY or SHOULD;
- permit behavior classified by SAB as ILLEGITIMATE;
- redefine a Stable Authority Boundary;
- redefine Evidence or Authority in a manner inconsistent with this specification;
- or claim that domain-specific guidance supersedes the core standard.

A compliant profile SHOULD identify itself using language such as:

“SAB v1.0 Application Profile”

or:

“Conformant with and derived from the requirements of SAB v1.0.”

The core SAB invariant remains controlling.

0.8 Conformance Claim Validity — SAB-CONF-001

A claim of conformance to SAB v1.0 is valid only when the claimant:

1. identifies the Canonical Release;
2. states the CSI;
3. declares exactly one applicable SAB conformance level;
4. identifies the assessed implementation and assessment scope;
5. provides the governance artifacts required for the claimed level;
6. performs the required conformance tests;
7. reports the outcome of each required test;
8. satisfies the conformance decision rule defined by this specification;
9. and contains no unresolved ILLEGITIMATE or AMBIGUOUS — REFUSAL REQUIRED outcome within the claimed assessment scope.

A claim **MUST NOT** be considered valid solely because an implementation:

- references SAB;
- uses SAB terminology;
- implements a boundary described as a SAB;
- was designed with SAB principles in mind;
- or self-identifies as SAB-compliant.

Conformance is established by demonstrated behavior and evidence, not by declaration.

An external DOI is not required to establish conformance validity.

0.9 Canonical Integrity — SAB-INTEG-001

The canonical publication record **MUST** provide a cryptographic integrity value for the normative released file.

Unless superseded by a later controlled revision of this specification, the required integrity mechanism **SHALL** be:

SHA-256

The canonical record **MUST** identify:

- the filename of the normative released artifact;
- the exact SAB version;
- the CSI;
- the publication date;
- the SHA-256 hash;
- and the release status.

A copy whose cryptographic hash does not match the canonical hash **MUST NOT** be represented as an identical copy of the Canonical Release.

Cryptographic identity establishes file integrity and release identity.

It does not, by itself, establish conformance of an implementation.

0.10 Versioning and Supersession — SAB-VER-001

Each normative SAB release **MUST** have an explicit version designation.

A later SAB release **MUST NOT** silently replace an earlier release.

Where a release is superseded:

- the earlier canonical record **MUST** remain available;
- the earlier record **MUST** identify that it has been superseded;
- the superseding version **MUST** be identified;
- the superseding record **SHOULD** identify the release it replaces;
- and conformance claims **MUST** continue to identify the specific version against which conformance was assessed.

A conformance claim made against SAB v1.0 does not automatically become a claim of conformance to a later SAB version.

Changes to normative requirements require a new controlled version.

0.11 Editorial Corrections and Errata — SAB-ERR-001

Non-normative editorial corrections **MAY** be issued without changing the technical requirements of SAB v1.0 only when the correction does not alter:

- a requirement;
- an invariant;
- a definition affecting conformance;
- a conformance outcome;
- a required artifact;
- a test condition;

- or the meaning of normative language.

Every corrected release MUST:

- be uniquely identifiable;
- receive a new cryptographic hash;
- identify the nature of the correction;
- preserve the CSI where the underlying normative standard remains SAB v1.0;
- and be linked through the canonical record.

Any change that alters normative meaning MUST be treated as a controlled standards revision rather than an editorial correction.

0.12 Custodianship — SAB-CUST-001

BLOCK VECTOR Technologies, L.L.C. is the publisher and standards custodian for the Canonical Release of SAB v1.0.

Custodianship includes responsibility for:

- maintaining the canonical publication record;
- preserving released versions;
- publishing integrity information;
- identifying superseded releases;
- publishing approved corrections;
- maintaining canonical version history;
- and preventing ambiguity as to which release constitutes the controlling SAB standard.

Custodianship does not alter the licensing terms applicable to authorized use or redistribution of the published standard.

0.13 Licensing, Trademarks, and Patent Rights — SAB-LIC-001

Copyright and similar rights in the text of this specification are licensed under Creative Commons Attribution 4.0 International (CC BY 4.0). CC BY 4.0 governs copyright permission to share and adapt the licensed text and does not grant patent or trademark rights.

Sections 0.5 through 0.7 define canonical status, naming, and the validity of SAB-related conformance representations; they do not narrow the copyright permissions granted by CC BY 4.0. Use of Stable Authority Boundary™, SAB, BLOCK VECTOR names, marks, identifiers, or branding MUST NOT falsely imply sponsorship, endorsement, canonical status, custodianship, or formal SAB conformance. This specification does not itself grant a patent or trademark license beyond rights, if any, separately stated by the applicable rights holder.

1. Scope (Normative)

1.1 Applicability

This specification applies to any system, component, process, workflow, or combination thereof that can **directly or indirectly cause, permit, initiate, authorize, enable, or materially contribute to consequential execution**.

Consequential execution includes, but is not limited to:

- execution of an action;
- enforcement of a rule, policy, restriction, or decision;
- approval or denial;
- creation or modification of authoritative state;
- issuance of a binding decision;
- commitment or allocation of resources;
- exercise of control over a system, device, process, or environment;
- modification of rights, privileges, access, obligations, or status;
- initiation of an irreversible or materially difficult-to-reverse action;
- or production of another operational effect having consequential impact.

The applicability of SAB is determined by the relationship between the system and the consequential execution, not merely by the internal function or designation of the system.

1.2 Direct and Indirect Execution

This specification applies where a system directly performs consequential execution.

It also applies where a system does not itself perform the final execution but produces an output that is subsequently relied upon to authorize, trigger, permit, select, control, or materially determine that execution.

Accordingly, a system described as advisory, analytical, assistive, decision-support, classificatory, predictive, or recommendatory is within scope when its output participates in a consequential execution path.

The absence of direct actuator, enforcement, publication, or state-changing capability does not place a system outside scope when its output can acquire consequential operational effect through another human or machine process.

1.3 Implementation Independence

This specification applies regardless of implementation technology or system composition.

A system within scope may be:

- software-based;
- hardware-based;
- human-operated;
- machine-operated;
- autonomous;
- semi-autonomous;
- deterministic;
- probabilistic;
- AI-enabled;
- rules-based;
- cyber-physical;
- socio-technical;
- distributed;
- centralized;
- advisory;
- decision-support;
- or a hybrid of human and machine components.

No implementation category independently establishes or removes SAB applicability.

1.4 Authority Path Determines Scope

Where multiple components participate in a consequential execution path, SAB applicability extends to the portion of the architecture through which:

evidence or evaluation becomes proposed execution, and proposed execution may acquire authority to proceed.

A component that produces evidence MAY remain non-authoritative.

A component that receives, interprets, approves, authorizes, transmits, or executes a proposed action MAY occupy part of the authority-bearing path.

The relevant scope for conformance assessment MUST include every component, interface, decision point, override mechanism, and execution path whose behavior can affect whether proposed execution crosses the Stable Authority Boundary.

1.5 Advisory Systems

An advisory system is not excluded from SAB solely because it lacks independent execution authority.

Where an advisory output is used operationally to support, determine, approve, authorize, or trigger consequential execution, the resulting human-machine or machine-machine workflow is within scope.

A human intermediary does not remove SAB applicability.

Human review, human approval, or human interaction is relevant only to the extent that the human participant possesses recognized authority applicable to the proposed execution.

1.6 Systems Outside Scope

A system is outside the scope of this specification only where its outputs, actions, and state transitions **cannot directly or indirectly cause, authorize, enable, or materially contribute to consequential execution** within the assessed context.

The mere processing, storage, transmission, presentation, or analysis of information does not by itself create SAB applicability where that activity has no consequential execution path.

If an output that was previously non-operational is later incorporated into a consequential execution path, SAB applicability begins at the point where that output becomes capable of influencing authoritative or operational execution.

1.7 Scope Principle

SAB applicability is determined by the possibility of consequential execution and the authority required for that execution.

It is not determined by whether the system is called:

- artificial intelligence;
- autonomous;
- automated;
- intelligent;
- advisory;
- assistive;
- software;
- hardware;
- or human-operated.

Scope is determined by consequence, not technology.

2. Normative Language and Interpretation

2.1 Purpose

This section defines the language used to distinguish **mandatory conformance requirements, recommended practices, permitted implementation choices, descriptive statements, and non-normative guidance** within this specification.

Normative language establishes requirements against which SAB conformance may be assessed.

A statement is normative only when it imposes, prohibits, recommends, or expressly permits behavior through the terms defined in this section or when the specification explicitly identifies the statement as normative.

2.2 Mandatory Requirements

The terms **MUST**, **MUST NOT**, **SHALL**, **SHALL NOT**, and **REQUIRED** identify mandatory conditions of conformance.

For purposes of this specification:

- **MUST**, **SHALL**, and **REQUIRED** have equivalent normative force.
- **MUST NOT** and **SHALL NOT** identify prohibited behavior.

Failure to satisfy a mandatory requirement means that the affected requirement is not satisfied.

An implementation **MUST NOT** claim SAB conformance at a level for which an applicable mandatory requirement remains unsatisfied.

Where this specification defines a mandatory Refusal/Hold condition, execution contrary to that condition constitutes behavior inconsistent with SAB conformance.

2.3 Recommended Requirements

The terms **SHOULD** and **SHOULD NOT** identify requirements for which deviation may be permissible under particular circumstances, but only after the consequences of that deviation have been understood and addressed.

An implementation that does not follow a **SHOULD** or **SHOULD NOT** provision **SHOULD** document:

- the reason for the deviation;

- the affected scope;
- the alternative control or behavior used;
- and any resulting effect on authority, refusal, traceability, or conformance evidence.

A **SHOULD** statement **MUST NOT** be interpreted as optional merely for convenience, throughput, implementation simplicity, or operational preference.

A **SHOULD NOT** statement **MUST NOT** be disregarded without a reasoned basis.

No deviation from a **SHOULD** or **SHOULD NOT** provision may be used to weaken or bypass a separate mandatory requirement.

2.4 Permitted Behavior

The term **MAY** identifies behavior permitted by this specification but not required for conformance unless another requirement makes that behavior necessary in a particular context.

Where this specification states that an implementation **MAY** perform an action, the implementation remains responsible for satisfying all applicable authority, boundary, refusal, traceability, and conformance requirements associated with that action.

Normative permission within this specification does not itself constitute operational authority.

A statement that an implementation **MAY** perform an action means only that SAB does not prohibit that implementation choice. It does not establish the recognized authority required for consequential execution.

This distinction is fundamental:

Permission within the standard is not authority within the system.

2.5 Capability Is Not Normative Permission

The terms **CAN** and **CANNOT**, when used, describe capability, possibility, or technical condition and do not establish normative permission or prohibition unless the surrounding provision explicitly states otherwise.

For example:

A component **CAN** technically modify authoritative state.

does not mean:

The component **MAY** modify authoritative state.

Technical capability does not establish authority.

2.6 Descriptive and Future Statements

The terms **IS**, **ARE**, **WILL**, and similar declarative language ordinarily describe a condition, architecture, consequence, or expected behavior and do not independently establish a conformance requirement unless the surrounding provision explicitly makes the statement normative.

Where a requirement is intended, this specification uses the normative terms defined in Sections 2.2 through 2.4.

2.7 Normative and Non-Normative Content

Content identified as **Normative** establishes requirements applicable to SAB interpretation or conformance.

Content identified as **Non-Normative**, **Informative**, **Example**, **Note**, **Rationale**, or equivalent language is provided to aid understanding and **MUST NOT** independently create, modify, weaken, or supersede a normative requirement.

Non-normative material **MAY** explain:

- intent;
- architectural rationale;
- implementation considerations;
- representative use cases;
- examples;
- threat scenarios;
- or possible implementation approaches.

Where non-normative material appears inconsistent with a normative requirement, the normative requirement controls.

2.8 Examples

Examples in this specification are illustrative unless explicitly identified as exhaustive.

An example demonstrates one possible application of a requirement and **MUST NOT** be interpreted as limiting the requirement to the implementation, technology, scenario, domain, or failure mode shown.

Absence of a particular scenario from an example does not place that scenario outside SAB where the normative scope and requirements otherwise apply.

2.9 Figures, Tables, and Architectural Diagrams

Figures, tables, diagrams, flow representations, and reference architectures are normative only where explicitly identified as normative.

Otherwise, they are explanatory representations of requirements stated in normative text.

A diagram or example **MUST NOT** be interpreted as creating an execution path, exception, authority source, or conformance allowance that is not supported by the normative requirements.

2.10 Requirement Identifiers

Normative requirements defined by this specification are assigned persistent identifiers, including but not limited to:

- **SAB-INV-###** — invariants;
- **SAB-REQ-###** — conformance requirements;
- **SAB-ART-###** — required governance or conformance artifacts;
- **SAB-CAN-###** — canonical publication requirements;
- **SAB-CONF-###** — conformance-claim requirements;
- **SAB-CONS-###** — cross-artifact consistency requirements;
- and other controlled identifier classes defined by this specification.

Requirement identifiers provide stable references for:

- implementation;
- testing;
- traceability;
- conformance assessment;
- audit;
- implementation profiles;
- derivative guidance;
- and standards mappings.

The text associated with the identifier, not the identifier alone, establishes the normative requirement.

2.11 No Implied Exceptions

A normative SAB requirement **MUST NOT** be weakened by implication.

Operational pressure, performance goals, convenience, system availability, model confidence, human expectation, retry behavior, degraded operation, or absence of an expressly stated prohibition do not create an exception to a mandatory requirement.

An exception exists only where this specification expressly permits one or where legitimate external authority, operating within the applicable SAB requirements, authorizes an action that the specification permits to be externally authorized.

2.12 Interpretation Under Ambiguity

Where an implementation cannot determine whether a proposed behavior satisfies an applicable mandatory SAB requirement, that uncertainty **MUST NOT** be treated as evidence of conformance or permission to execute.

For conformance assessment, the unresolved condition **MUST** be identified and evaluated using the outcome rules established by this specification.

For operational execution, ambiguity concerning the required authority remains subject to the SAB Refusal/Hold requirements.

Ambiguity does not create permission.

3. Definitions (Normative)

3.1 Applicability of Definitions

The definitions in this section govern interpretation of this specification.

Where a term defined in this section is used in a normative requirement, its meaning SHALL be the meaning established here unless the specification explicitly states otherwise.

Domain-specific terminology MAY supplement these definitions but MUST NOT redefine them in a manner that weakens, contradicts, or alters the normative meaning of SAB.

3.2 Authority

Authority: A bounded mandate recognized by the applicable governance context under which a person, role, organization, system, mechanism, or other recognized entity is permitted to authorize a specified consequential execution.

Authority MUST have an identifiable source and MUST be limited by the scope, conditions, duration, delegation rules, revocation conditions, and other constraints applicable to that mandate.

Authority is distinct from:

- evidence;
- confidence;
- recommendation;
- technical capability;
- access;
- identity;
- authentication;
- privilege;
- possession of credentials;
- and ability to cause an action.

Possession of technical capability or access MUST NOT, by itself, be treated as possession of authority.

3.3 Recognized Authority

Recognized Authority: Authority whose source, applicability, scope, and validity are defined by the governing organization, policy, law, contract, command structure, operating procedure, system configuration, or other authority source accepted by the applicable governance context for the proposed execution.

Recognized authority **MUST** be capable of being evaluated at the Stable Authority Boundary.

Authority that is merely asserted, inferred, presumed, self-declared, or derived solely from technical capability is not recognized authority unless the applicable authority model explicitly establishes it as such.

3.4 Authority Source

Authority Source: The law, regulation, policy, contract, delegation, command relationship, organizational mandate, operating rule, configuration, credentialed role, or other recognized basis from which authority derives.

An authority source defines or constrains:

- who or what may authorize;
- what may be authorized;
- under what conditions;
- for what duration;
- within what scope;
- and subject to what revocation, escalation, or override rules.

3.5 Authority Proof

Authority Proof: Evidence sufficient to demonstrate, at the Stable Authority Boundary, that the recognized authority required for a specific proposed execution exists, is valid, applies to that execution, and remains within its authorized scope.

Authority proof **MAY** include authenticated identity, role assignment, delegation records, signed authorization, policy state, configuration state, approval records, machine-verifiable credentials, command authority, or other mechanisms appropriate to the implementation.

Authority proof is evidence **of authority**.

It is not the authority itself.

3.6 Evidence

Evidence: Information available to a person or system for evaluation, including observations, measurements, records, source material, sensor output, logs, analyses, classifications, model outputs, confidence scores, recommendations, derived claims, provenance information, or other information relevant to a proposed execution.

Evidence **MAY** support, contradict, qualify, or invalidate a proposed execution or an authority determination.

Evidence **MUST NOT** independently exercise authority unless the governing authority model separately establishes an authority-bearing mechanism that evaluates that evidence and authorizes execution.

Evidence informs. Authority authorizes.

3.7 Evaluation

Evaluation: The process of examining evidence, state, conditions, rules, requirements, or other relevant information in order to produce a finding, classification, recommendation, proposed action, or other decision input.

Evaluation **MAY** be performed by:

- a human;
- deterministic software;
- an algorithm;
- an AI system;
- a sensor-derived process;
- a rules engine;
- a controller;
- or a combination of human and machine processes.

Evaluation does not inherently constitute authorization.

3.8 Proposed Execution

Proposed Execution: A specific action, state transition, approval, denial, enforcement action, resource commitment, control action, authoritative-state modification, or other consequential effect that has been generated, requested, selected, recommended, or otherwise presented for authorization but has not yet been permitted to execute.

The source of a proposed execution **MAY** be human or machine.

The origin of the proposal does not establish its authority.

3.9 Stable Authority Boundary

Stable Authority Boundary (SAB): The point at which the proposed execution is evaluated against recognized authority.

A Stable Authority Boundary **MUST** be explicit and enforceable.

The boundary separates:

- evaluation from authorization;
- evidence from authority;
- proposed execution from authorized execution;
- proposed state from authoritative state;
- and technical capability from permission to execute.

A boundary that exists only as documentation, expectation, convention, or operator habit but is not enforced on the relevant execution path does not satisfy the SAB requirement for an explicit boundary gate.

3.10 Boundary Decision

Boundary Decision: The operational determination produced when a proposed execution is evaluated against recognized authority at the Stable Authority Boundary.

A Boundary Decision MUST result in exactly one operational disposition:

AUTHORIZED EXECUTION — the recognized authority applicable to the proposed execution is valid, within scope, and verifiable, and the execution is permitted to proceed;

NON-ACTION — no consequential execution is required or selected under the applicable conditions, and the current state is preserved;

REFUSAL/HOLD — a proposed execution exists, but the authority required for that execution is absent, invalid, expired, revoked, ambiguous, outside scope, degraded, or otherwise unverifiable.

Escalation MAY follow a Refusal/Hold, but escalation does not itself authorize execution. An override MAY result in a new authorization only when independently recognized override authority is established and the resulting proposed execution is again governed by an applicable Stable Authority Boundary.

The operational dispositions in this section are distinct from the conformance assessment outcomes defined in Section 6. LEGITIMATE, AMBIGUOUS — REFUSAL REQUIRED, and ILLEGITIMATE are assessment outcomes assigned to demonstrated behavior; they are not runtime authorization states.

3.11 Execution

Execution: The performance or permitted completion of an action that produces a consequential effect.

Execution includes, but is not limited to:

- changing system or authoritative state;
- enforcing policy;
- granting or denying access, rights, benefits, or privileges;
- committing resources;
- approving or denying a binding action;
- issuing commands;

- modifying configuration;
- publishing authoritative information;
- controlling a device or process;
- initiating an irreversible action;
- or otherwise producing operational effect.

Execution may be physical, digital, administrative, financial, informational, or organizational.

3.12 Consequential Execution

Consequential Execution: Execution capable of materially affecting authoritative state, rights, obligations, resources, safety, security, operations, policy enforcement, system behavior, or another outcome for which legitimate authority is required.

The significance of an action is determined by its effect within the assessed context, not by the technology used to perform it.

3.13 Proposed State

Proposed State: Information, configuration, status, classification, decision, or other state representation that has been generated or evaluated but has not yet received the authority required to become authoritative or operationally controlling.

Proposed state MUST remain distinguishable from authoritative state where the transition between them is consequential.

3.14 Authoritative State

Authoritative State: State that has passed the applicable Stable Authority Boundary and is recognized by the governing system, organization, or process as valid for consequential use.

Authoritative state may include:

- official records;
- approved configurations;
- binding determinations;
- active permissions;
- operational commands;
- published information;
- or other state upon which consequential action may legitimately rely.

Technical persistence in a database, file, memory location, or device does not by itself establish authoritative status.

3.15 Execution Path

Execution Path: The complete sequence of components, interfaces, decisions, transformations, approvals, authorizations, overrides, and mechanisms through which a proposed execution may reach consequential effect.

An execution path includes alternate, administrative, automated, retry, recovery, maintenance, escalation, and override routes capable of producing the same or equivalent consequential effect.

3.16 Authority-Bearing Path

Authority-Bearing Path: The portion of an execution path through which authority is established, verified, exercised, transmitted, delegated, overridden, or otherwise used to permit consequential execution.

All authority-bearing paths are within the applicable SAB conformance scope.

3.17 Refusal/Hold

Refusal/Hold: A legitimate non-executing state in which a proposed execution is prevented from proceeding because the authority required for that execution is absent, invalid, expired, revoked, ambiguous, outside scope, degraded, or otherwise unverifiable.

Refusal/Hold is a valid operational outcome.

Refusal/Hold MUST NOT be treated as a system failure merely because execution did not occur.

A Refusal/Hold state remains in force until:

- recognized authority is established;
- the proposed execution is withdrawn;
- legitimate escalation resolves the authority condition;
- or an authorized external override is exercised in accordance with this specification.

3.18 Progress-by-Default

Progress-by-Default: Any behavior by which proposed execution advances toward or reaches consequential effect without verified recognized authority merely because an expected authorization, review, dependency, response, denial, or control is absent, delayed, unavailable, bypassed, or inconclusive.

Progress-by-default includes execution resulting from:

- timeout;

- missing response;
- queue aging;
- backlog pressure;
- retry exhaustion;
- unavailable reviewer;
- unavailable authority service;
- fail-open behavior;
- presumed approval;
- or equivalent mechanisms.

Absence of denial is not authorization.

3.19 Fail-Open

Fail-Open: Behavior in which failure, uncertainty, unavailable dependency, invalid state, missing authority verification, or another degraded condition permits execution that would otherwise require successful authority verification.

Fail-open behavior on an authority-bearing path is inconsistent with SAB unless the proposed execution remains independently authorized under a separately established and verifiable authority path.

3.20 Override

Override: An explicit and externally authorized action that permits a defined execution to proceed despite a prior Refusal/Hold or restrictive boundary decision.

An override **MUST**:

- originate from recognized authority outside the refused execution path;
- identify the execution being authorized;
- remain within the scope of the overriding authority;
- preserve the original refusal or boundary decision;
- identify the reason for the override;
- and produce an auditable record.

An override does not erase the event that caused the original refusal.

An override does not automatically establish continuing authority for subsequent executions.

3.21 Self-Authorization

Self-Authorization: Behavior in which the person, component, agent, process, or execution path proposing an action creates, asserts, expands, or substitutes the authority required to

permit that same action without an independently recognized authority source permitting it to do so.

Self-authorization is prohibited where this specification requires external authority.

3.22 Self-Escalation

Self-Escalation: Behavior in which a component or execution path increases its own authority, privilege, execution scope, or ability to cross a Stable Authority Boundary without independently recognized authority permitting that escalation.

Technical ability to obtain additional capability does not establish authority to exercise that capability.

3.23 Degraded Mode

Degraded Mode: Any operating condition in which information, dependencies, communications, coordination, authority-verification capability, evidence quality, system integrity, human availability, or other conditions required for normal operation are reduced, unavailable, uncertain, or impaired.

Degraded mode does not create additional authority.

Where degradation prevents required authority from being verified, the affected proposed execution MUST enter Refusal/Hold.

3.24 Escalation

Escalation: Transfer of an unresolved proposed execution or authority determination to another person, role, mechanism, or authority source capable of resolving the condition.

Escalation MAY seek additional evidence, clarification, authority, or review.

Escalation itself does not authorize execution.

3.25 Evaluator

Evaluator: A person, system, component, or process that evaluates evidence or conditions and produces a finding, recommendation, classification, proposed state, or proposed execution.

An evaluator is not an authorizer solely because it performs evaluation.

3.26 Authorizer

Authorizer: A person, role, system, component, or mechanism possessing recognized authority to permit a defined proposed execution within a defined scope.

An authorizer **MUST NOT** authorize beyond the authority actually granted to it.

3.27 Executor

Executor: A person, system, component, or mechanism that performs or causes an authorized execution.

An executor's technical ability to perform an action does not establish authority to perform that action.

3.28 Enforcement Mechanism

Enforcement Mechanism: The technical, procedural, organizational, physical, or hybrid control that prevents a proposed execution from bypassing the Stable Authority Boundary or proceeding contrary to its boundary decision.

An enforcement mechanism **MUST** be capable of preventing execution, not merely detecting or reporting unauthorized execution after it occurs, where prevention is required by the applicable SAB requirement.

3.29 Conformance Scope

Conformance Scope: The explicitly defined system boundary, components, execution paths, interfaces, authority-bearing mechanisms, operating conditions, and functions included in a specific SAB conformance assessment.

A conformance claim applies only to the declared conformance scope.

A conformant subsystem **MUST NOT** be used to imply conformance of an unassessed larger system.

3.30 Ambiguous Authority

Ambiguous Authority: A condition in which authority may exist but its source, validity, scope, applicability, status, or proof cannot be established with sufficient certainty to permit the proposed execution.

Ambiguous authority is not partial authorization.

Under SAB, unresolved ambiguous authority requires **Refusal/Hold**.

3.31 Non-Action

Non-Action: A deliberate operational disposition in which no consequential execution is performed because no such execution is required or selected under the applicable conditions. Non-Action is distinct from malfunction, incomplete processing, and Refusal/Hold.

Where a proposed execution exists but required authority cannot be established or verified, the correct disposition is Refusal/Hold rather than Non-Action. Refusal/Hold may result in no action being performed, but the authority disposition remains Refusal/Hold.

3.32 Legitimacy

Legitimacy: For purposes of SAB conformance, the condition in which an authority relationship, Boundary Decision, or execution behavior conforms to the recognized authority applicable under the declared Authority Model and to the applicable requirements of this specification.

SAB conformance assesses architectural legitimacy relative to the declared Authority Model. It does not by itself determine the legal validity, policy wisdom, or ethical propriety of an external Authority Source.

3.33 Authority Model

Authority Model: The controlled representation of the Authority Sources, authorizers, scopes, conditions, durations, delegation rules, revocation rules, escalation paths, override rules, Authority Proof mechanisms, and other constraints used to determine whether a proposed execution possesses Recognized Authority.

The Authority Model MUST be sufficiently explicit for the applicable authority to be evaluated at the Stable Authority Boundary.

3.34 Authority Surface

Authority Surface: The complete set of authority-bearing decision points, interfaces, mechanisms, roles, administrative functions, automation paths, recovery paths, integrations, and other routes capable of permitting or enabling consequential execution within the declared Conformance Scope.

Authority Surface Minimization requires those points to be enumerated, justified, governed, constrained to their required scope, and removed or disabled when they are not required for the declared implementation functions.

3.35 Authorized Execution

Authorized Execution: A Proposed Execution that the Stable Authority Boundary permits to proceed after the Recognized Authority applicable to that specific execution has been established as valid, within scope, and verifiable.

Authorization applies only within the scope, conditions, duration, and other constraints of the verified authority and does not imply continuing authority for later or different executions.

4. SAB Core Invariants (Normative)

4.1 Status of the Core Invariants

The invariants in this section define the minimum architectural conditions that distinguish a Stable Authority Boundary from an ordinary approval, access-control, workflow, or decision-support mechanism.

The core invariants apply to **all SAB conformance levels, all implementation technologies, all execution domains, and all authority-bearing paths within the declared conformance scope.**

A conformant implementation **MUST** satisfy all applicable SAB core invariants.

An implementation profile, domain-specific specification, operating procedure, exception mechanism, or implementation choice **MUST NOT** weaken, bypass, redefine, or contradict a SAB core invariant.

Failure to satisfy any applicable core invariant constitutes failure of SAB conformance for the affected scope.

4.2 SAB-INV-001 — Evidence Informs; Authority Authorizes

Evidence MAY inform a decision. Evidence MUST NOT independently establish, substitute for, or exercise the recognized authority required for consequential execution.

A conformant system **MUST** maintain an explicit and enforceable Stable Authority Boundary between evaluation and authorization such that a proposed execution is not permitted to proceed solely because supporting evidence is:

- credible;
- complete;
- persuasive;
- high-confidence;
- statistically probable;
- repeatedly confirmed;
- produced by a trusted system;
- produced by an authorized user;
- generated by multiple agreeing sources;
- or technically sufficient to cause execution.

Evidence includes, but is not limited to:

- observations;
- measurements;
- human analysis;

- model outputs;
- classifications;
- confidence scores;
- recommendations;
- sensor results;
- logs;
- derived claims;
- and other evaluation products.

Evidence MAY demonstrate that recognized authority exists.

Evidence of authority is not the authority itself.

The boundary MUST determine whether the recognized authority required for the specific proposed execution exists, is valid, applies to that execution, and remains within scope before consequential execution is permitted.

Technical capability, access, identity, authentication, privilege, confidence, prediction, recommendation, consensus, or successful completion of an evaluation MUST NOT be treated as substitutes for recognized authority.

Evidence informs. Authority authorizes.

4.3 SAB-INV-002 — Refusal Is Correct Under Unverifiable Authority

Where the recognized authority required for a proposed execution cannot be established and verified at the Stable Authority Boundary, the proposed execution MUST NOT proceed.

A conformant system MUST enter or maintain **Refusal/Hold** when required authority is:

- absent;
- invalid;
- expired;
- revoked;
- outside the applicable scope;
- contradictory;
- ambiguous;
- unavailable for verification;
- dependent upon an unverifiable assumption;
- or otherwise incapable of being established at the boundary.

Refusal/Hold is a legitimate operating state and MUST NOT be treated as an error merely because consequential execution did not occur.

The system MUST NOT convert unresolved authority into permission through:

- timeout;
- retry;
- elapsed time;
- queue aging;
- backlog pressure;
- absence of a reviewer;
- absence of an explicit denial;
- repeated submission;
- alternate execution paths;
- increasing confidence;
- model consensus;
- operational urgency;
- or any other progress-by-default mechanism.

A refused or held execution MAY proceed only when:

- the required recognized authority is subsequently established and verified;
- the proposed execution is withdrawn or replaced by a separately authorized execution;
- legitimate escalation establishes the required authority;
- or an authorized external override is exercised in accordance with this specification.

Absence of verified authority is a reason to stop, not a reason to continue.

4.4 SAB-INV-003 — Degraded Mode Tightens Authority

Degradation MUST NOT create, expand, presume, or substitute authority.

When conditions required to establish or verify recognized authority become degraded, unavailable, uncertain, inconsistent, or impaired, a conformant system MUST maintain or reduce the permitted execution scope.

Degraded conditions include, but are not limited to:

- loss of communications;
- unavailable dependencies;
- missing or incomplete evidence;
- authority-verification failure;
- identity or authentication degradation;
- unavailable authorized personnel;
- partial system failure;
- conflicting state;
- corrupted information;
- loss of coordination;
- reduced system integrity;
- uncertain operating conditions;

- or ambiguous authority state.

During degraded operation, the system **MUST NOT**:

- weaken the authority-verification requirement;
- increase execution latitude;
- broaden an existing authority grant;
- infer new authority from prior authority;
- convert missing verification into approval;
- bypass the Stable Authority Boundary;
- enable fail-open execution;
- or permit a component to self-authorize because a normal authority mechanism is unavailable.

Previously authorized execution **MAY** continue during degraded operation only where the authority applicable to that execution remains valid, within scope, and independently verifiable under the degraded condition.

Where degradation prevents the required authority from being verified, the affected proposed execution **MUST** enter or remain in **Refusal/Hold**.

Restoration of technical capability does not by itself restore authority. Authority **MUST** again be valid and verifiable before execution that depended upon the degraded authority condition may proceed.

Loss of certainty does not create additional authority.

4.5 SAB-INV-004 — Non-Action Is a Valid State

A conformant system MUST recognize Non-Action as a valid operational disposition when no consequential execution is required or selected under the applicable conditions.

SAB does not require a system to produce an action merely because:

- an input was received;
- an evaluation was completed;
- evidence was available;
- a recommendation was generated;
- a workflow was initiated;
- an operator expected an outcome;
- an automated process reached a decision point;
- a response deadline approached or expired;
- or the system possesses the technical capability to act.

A system **MUST NOT** interpret the absence of execution as an error condition requiring automatic remediation when Non-Action is the correct operational disposition.

Non-action MAY result because:

- no consequential execution is warranted;
- the available evidence does not support a proposed execution;
- the applicable authority does not require action;
- an existing authoritative state requires no modification;
- or the legitimate outcome is to preserve the current state.

A conformant implementation MUST distinguish between:

- **successful authorized execution;**
- **successful determination that no execution is required;**
- **Refusal/Hold because required authority cannot be established;**
- **and failure to perform an execution that was validly authorized and required.**

These states MUST NOT be collapsed into a single concept of success or failure where doing so would obscure the authority decision.

Where non-action is the legitimate outcome, the system MUST NOT manufacture, infer, escalate, retry, or broaden authority merely to produce an execution.

Operational objectives such as throughput, responsiveness, automation rate, task completion, availability, or action frequency MUST NOT create a requirement to execute where the applicable conditions do not require or select consequential execution.

Where traceability is required by the applicable conformance level, the reason for consequential non-action MUST be sufficiently observable and auditable to distinguish deliberate non-action from system malfunction, lost work, or an unrecorded refusal.

The absence of action may itself be the correct operational outcome.

4.6 Relationship Among the Invariants

The four SAB core invariants operate together.

SAB-INV-001 establishes that evaluation and evidence do not independently constitute authority.

SAB-INV-002 establishes the required behavior when authority cannot be verified.

SAB-INV-003 establishes that degraded conditions cannot be used to relax that requirement.

SAB-INV-004 establishes that execution is not required merely because execution is possible.

Together, they establish the fundamental SAB execution rule:

A proposed consequential execution may proceed only when the recognized authority applicable to that execution is valid, within scope, and verifiable at the Stable Authority Boundary. Where no consequential execution is required or selected under the applicable conditions, Non-Action is valid.

Where a Proposed Execution exists but authority cannot be established or verified, the execution remains in Refusal/Hold.

No evidence quality, system capability, operational pressure, degraded condition, or absence of explicit denial creates an exception to this rule.

5. Conformance Levels (Normative)

5.1 Purpose of Conformance Levels

SAB conformance levels define increasing degrees of assurance that the Stable Authority Boundary is not merely present, but is **enforced, reconstructable, resistant to bypass, and reliable under adverse or degraded conditions**.

The conformance levels do not define different versions of SAB.

All claimed levels are governed by the same:

- canonical SAB definition;
- core invariants;
- normative definitions;
- mandatory authority behavior;
- Refusal/Hold requirements;
- prohibition against progress-by-default;
- prohibition against unauthorized self-override;
- and conformance outcome rules.

A higher conformance level adds assurance requirements. It does not replace or weaken the requirements of a lower level.

An implementation claiming SAB-L2 MUST satisfy all SAB-L1 requirements.

An implementation claiming SAB-L3 MUST satisfy all SAB-L1 and SAB-L2 requirements.

5.2 Conformance Is Assessed Within a Declared Scope

Every conformance claim MUST identify the **Conformance Scope** being assessed.

The declared scope MUST identify, as applicable:

- the system or implementation being assessed;
- version or build;
- relevant configuration;
- authority-bearing functions;

- consequential execution paths;
- boundary enforcement points;
- interfaces capable of producing or influencing consequential execution;
- override mechanisms;
- degraded operating conditions;
- and exclusions from the assessment.

A conformance claim applies only to the declared scope.

Conformance of one component, workflow, subsystem, deployment, or configuration **MUST NOT** be represented as conformance of a larger unassessed system.

5.3 SAB-L1 — Baseline Conformance

SAB-L1 establishes the minimum acceptable implementation of a Stable Authority Boundary.

An SAB-L1 implementation **MUST** demonstrate that:

1. every consequential execution path within the declared scope passes through an explicit Stable Authority Boundary;
2. the recognized authority required for each proposed execution is evaluated before execution is permitted;
3. evidence, confidence, technical capability, access, identity, recommendation, or successful evaluation does not independently substitute for recognized authority;
4. a proposed execution enters or remains in Refusal/Hold when the required authority cannot be established or verified;
5. progress-by-default is prohibited;
6. fail-open execution is prohibited on authority-bearing paths where authority verification is required;
7. the system cannot bypass the boundary through alternate execution paths, retries, administrative functions, maintenance mechanisms, or equivalent routes within the assessed scope;
8. the system cannot independently convert its own Refusal/Hold state into authorization unless a separately recognized authority path explicitly permits that transition;
9. degraded operation does not expand authority or execution latitude;
10. Non-Action is permitted as a valid state where no consequential execution is required or selected under the applicable conditions;
11. the boundary decision governs whether consequential execution may proceed;
12. and behavior classified as ILLEGITIMATE by this specification is not permitted within the claimed scope.

SAB-L1 is therefore a **behavioral conformance level**.

It demonstrates that the required authority boundary exists and is enforced.

It does not, by itself, establish that an independent third party can fully reconstruct every consequential authority decision after the fact.

SAB-L1 Minimum Assurance Principle

No consequential execution without verified recognized authority. No fail-open path around that requirement.

5.4 SAB-L2 — Auditable Conformance

SAB-L2 includes all SAB-L1 requirements and adds sufficient governance, traceability, and retained evidence to permit independent reconstruction of consequential authority decisions.

An SAB-L2 implementation **MUST** satisfy SAB-L1 and **MUST** additionally:

1. maintain the governance and conformance artifacts required by this specification;
2. identify the recognized authority source applicable to each authority-bearing execution class;
3. identify the enforcement point for each Stable Authority Boundary;
4. preserve sufficient evidence to reconstruct the proposed execution;
5. preserve the evidence used during evaluation;
6. preserve the applicable authority proof;
7. preserve the resulting boundary decision;
8. preserve whether the result was execution, non-action, Refusal/Hold, escalation, or override;
9. preserve any material change between proposed state and resulting authoritative state;
10. preserve refusal conditions and subsequent resolution where Refusal/Hold occurred;
11. preserve escalation and override records sufficient to identify who or what possessed the authority to permit the resulting execution;
12. identify the implementation version, configuration, policy, ruleset, or equivalent state necessary to understand the consequential decision path;
13. provide traceability sufficient for an independent reviewer to determine whether each applicable SAB requirement was satisfied;
14. and retain such records according to a documented retention policy appropriate to the assessed system and consequence.

An independent reviewer **MUST** be able to reconstruct, to the extent required by the declared scope:

input → evidence → evaluation → proposed execution → authority proof → Boundary Decision → authorized execution / Non-Action / Refusal/Hold → escalation or override, if applicable → resulting authoritative or operational state

The reconstruction **MUST** distinguish **evidence supporting an action** from **authority permitting that action**.

SAB-L2 is therefore an **auditable assurance level**.

It demonstrates not only that SAB behavior occurred, but that the authority path can be independently examined and reconstructed.

SAB-L2 Minimum Assurance Principle

If the system acted—or deliberately did not act—the consequential authority path can be independently reconstructed.

5.5 SAB-L3 — Hardened Conformance

SAB-L3 includes all SAB-L1 and SAB-L2 requirements and adds controls intended to preserve SAB behavior when the authority mechanism, system, operator, dependency, or operating environment is subjected to failure, manipulation, bypass attempts, or degraded conditions.

An SAB-L3 implementation **MUST** satisfy SAB-L1 and SAB-L2 and **MUST** additionally demonstrate:

1. resistance to unauthorized modification of authority definitions, boundary logic, refusal state, audit evidence, and override controls;
2. protection of authority proof against unauthorized alteration, substitution, replay, or use outside its authorized scope;
3. protection of boundary enforcement mechanisms against unauthorized bypass or disabling;
4. durable Refusal/Hold behavior that survives restart, retry, resubmission, failover, alternate workflow routing, and equivalent attempts to recreate the refused execution;
5. externally governed override mechanisms that cannot be independently exercised by the refused component or execution path;
6. preservation of the original refusal or restrictive boundary decision when an authorized override occurs;
7. explicit degraded-mode behavior for each applicable authority-bearing path;
8. demonstration that degraded conditions maintain or reduce execution latitude and do not create new authority;
9. protection against unauthorized self-authorization, self-escalation, privilege expansion, or authority-scope expansion;
10. protection of conformance-relevant logs and traceability records against undetected alteration sufficient to invalidate independent reconstruction;
11. defined recovery behavior following loss or degradation of an authority-verification dependency;
12. re-establishment and verification of applicable authority before execution resumes where authority became unverifiable during degradation;
13. testing of identified bypass, override, retry, failover, degraded-mode, and authority-manipulation paths;

14. and evidence that the implementation remains within SAB requirements under the adverse conditions included in the declared SAB-L3 assessment scope.

SAB-L3 does not require that a system remain operational under every failure condition.

Where continued operation cannot preserve legitimate authority, affected Proposed Executions MUST enter or remain in Refusal/Hold. Non-Action remains a valid hardened outcome only where no consequential execution is required or selected under the applicable conditions.

Availability MUST NOT be preserved by weakening the Stable Authority Boundary.

SAB-L3 is therefore a **resilient and adversarially hardened assurance level**.

SAB-L3 Minimum Assurance Principle

The authority boundary remains legitimate when the surrounding system is stressed, degraded, manipulated, or partially unavailable.

5.6 Required Governance Artifacts by Conformance Level

The artifacts required by this specification SHALL be applied according to the claimed conformance level.

At SAB-L1, an implementation MUST maintain sufficient documented information to identify:

- the conformance scope;
- the consequential execution paths;
- the applicable recognized authority;
- the Stable Authority Boundary;
- and the required Refusal/Hold behavior.

At SAB-L2, all formal governance and conformance artifacts required by this specification MUST be complete and sufficient for independent reconstruction.

At SAB-L3, the SAB-L2 artifacts MUST additionally document:

- tamper-resistance controls;
- override governance;
- degraded-mode behavior;
- recovery conditions;
- hardened enforcement mechanisms;
- and the tests used to demonstrate resistance to bypass or unauthorized authority expansion.

A higher level MAY impose additional documentation or evidence requirements but MUST NOT reduce those required by a lower level.

5.7 Conformance-Level Declaration

An implementation claiming SAB conformance **MUST** declare **exactly one highest achieved conformance level**:

- **SAB-L1 — Baseline**
- **SAB-L2 — Auditable**
- **SAB-L3 — Hardened**

A claim **MUST** identify:

- the SAB version;
- the CSI;
- the declared conformance level;
- the assessed implementation;
- the assessed version or configuration;
- the conformance scope;
- the assessment date;
- and the conformance result.

A claim of SAB-L2 inherently asserts satisfaction of SAB-L1.

A claim of SAB-L3 inherently asserts satisfaction of SAB-L1 and SAB-L2.

An implementation **MUST NOT** claim a higher conformance level solely because it possesses some controls associated with that level.

The complete set of applicable requirements for the claimed level **MUST** be satisfied.

5.8 No Partial-Level Conformance

An implementation **MUST NOT** claim:

- “partial SAB-L1”;
- “mostly SAB-L2”;
- “SAB-L3 equivalent”;
- “SAB compliant except for”;
- or another expression reasonably implying formal SAB conformance where the applicable requirements for the claimed level have not been satisfied.

An implementation **MAY** report individual requirements as satisfied or unsatisfied.

Such reporting does not constitute a formal SAB conformance claim until the complete conformance decision rule has been satisfied.

5.9 Configuration and Deployment Dependence

SAB conformance MAY depend upon system configuration, deployment architecture, policy state, authority assignment, or operating environment.

A change that materially affects:

- an execution path;
- authority source;
- authority scope;
- boundary enforcement mechanism;
- refusal behavior;
- override mechanism;
- degraded-mode behavior;
- or conformance evidence

MUST be evaluated to determine whether the existing conformance claim remains valid.

A conformant software component deployed in a configuration that bypasses its Stable Authority Boundary MUST NOT be represented as a conformant deployment.

5.10 Conformance Does Not Create Authority

A SAB conformance claim establishes that the assessed implementation satisfies the applicable requirements of this specification within the declared scope.

A conformance claim does **not**:

- grant operational authority;
- identify what an organization must authorize;
- create legal authority;
- expand an existing mandate;
- authorize a particular execution;
- or replace the authority model of the implementing organization.

SAB verifies the governance of authority. SAB conformance does not itself supply that authority.

5.11 Relationship Between Conformance Levels

The progression of SAB assurance is:

SAB-L1 — Is the boundary enforced?

SAB-L2 — Can the authority decision be independently reconstructed?

SAB-L3 — Does the boundary remain legitimate under degradation, manipulation, and attempted bypass?

The conformance levels therefore represent increasing assurance:

Enforced → Auditable → Hardened

They do not represent increasing amounts of authority.

A higher SAB conformance level never grants greater authority to execute.

6. Conformance Assessment Outcomes and Decision Rule (Normative)

6.1 Purpose

This section defines the outcome states used to evaluate SAB requirements, tests, execution paths, and overall conformance.

Each required SAB conformance test **MUST** produce exactly one assessment outcome:

- **LEGITIMATE**
- **AMBIGUOUS — REFUSAL REQUIRED**
- **ILLEGITIMATE**

These outcomes describe whether the assessed behavior preserves the Stable Authority Boundary requirements.

They do not indicate merely whether an action succeeded or failed.

A **LEGITIMATE** outcome **MAY** result in:

- authorized execution;
- Non-Action;
- Refusal/Hold;
- escalation;
- or another behavior expressly permitted by this specification.

The governing question is whether the observed or demonstrated behavior preserved legitimate authority.

6.2 Outcome Assignment

An assessment outcome **MUST** be based on demonstrated system behavior and available conformance evidence.

An assessor **MUST NOT** assign **LEGITIMATE** solely because:

- the design intends to enforce SAB;
- documentation states that a boundary exists;
- a component is described as secure;
- a human reviewer is present;
- the implementation uses access controls;
- a model produces high-confidence output;
- the implementation has not yet experienced a failure;
- or no unauthorized execution has previously been observed.

The outcome **MUST** reflect whether the applicable SAB requirement is demonstrably satisfied within the declared conformance scope.

6.3 LEGITIMATE

A test outcome is **LEGITIMATE** when the assessed behavior satisfies the applicable SAB requirement and preserves the required relationship between proposed execution and recognized authority.

A **LEGITIMATE** outcome requires that the applicable authority behavior be explicit, enforceable, and sufficiently demonstrated for the claimed conformance level.

Depending upon the requirement being tested, **LEGITIMATE** behavior **MAY** include:

- permitting execution after recognized authority is established and verified;
- preserving authoritative state where no execution is required;
- entering Refusal/Hold where required authority cannot be verified;
- remaining in Refusal/Hold during unresolved ambiguity;
- preventing progress-by-default;
- preventing fail-open execution;
- rejecting unauthorized override;
- contracting execution latitude during degraded operation;
- preserving a valid Non-Action state;
- or producing the evidence necessary for independent reconstruction.

A **LEGITIMATE** outcome therefore does **not** require execution to occur.

The test is **LEGITIMATE** when the system performs the behavior required by SAB.

LEGITIMATE Principle

The required authority relationship was preserved.

6.4 AMBIGUOUS — REFUSAL REQUIRED

A test outcome is **AMBIGUOUS — REFUSAL REQUIRED** when the available evidence is insufficient to establish that the applicable SAB requirement is satisfied, but illegitimate behavior has not been conclusively demonstrated.

Ambiguity includes conditions in which:

- the applicable authority source cannot be established;
- authority scope is unclear;
- authority validity cannot be verified;
- authority proof is incomplete or unavailable;
- the location or enforcement of the Stable Authority Boundary cannot be demonstrated;
- an execution path is incompletely mapped;
- required traceability is insufficient to reconstruct the authority decision;
- degraded-mode behavior is unknown or untested;
- override authority cannot be conclusively established;
- conformance evidence is contradictory;
- configuration state cannot be verified;
- or another unresolved condition prevents a determination of legitimacy.

AMBIGUOUS does not mean partially authorized.

AMBIGUOUS does not mean probably legitimate.

AMBIGUOUS does not create permission.

Where the ambiguity affects an operational authority determination, the affected proposed execution **MUST** enter or remain in **Refusal/Hold** until recognized authority is established and verified.

Where the ambiguity affects a conformance assessment, the affected requirement **MUST** remain unresolved until sufficient evidence exists to assign either **LEGITIMATE** or **ILLEGITIMATE**.

AMBIGUOUS Principle

Where legitimacy cannot be established, execution does not proceed.

6.5 ILLEGITIMATE

A test outcome is **ILLEGITIMATE** when the assessed implementation permits, performs, enables, or contains an execution path capable of consequential execution contrary to an applicable mandatory SAB requirement.

ILLEGITIMATE behavior includes, but is not limited to:

- consequential execution without verified recognized authority;
- bypass of the Stable Authority Boundary;
- progress-by-default;
- fail-open execution;
- self-authorization;
- unauthorized self-escalation;
- unauthorized self-override;
- substitution of evidence for authority;
- substitution of confidence, probability, consensus, recommendation, or technical capability for authority;
- execution under expired, revoked, invalid, or out-of-scope authority;
- expansion of execution latitude during degraded operation;
- retry or alternate routing that bypasses a refusal;
- destruction or suppression of a required refusal state;
- undocumented authority-bearing execution paths;
- unauthorized modification of authoritative state;
- or another behavior expressly prohibited by this specification.

Actual unauthorized execution is not required for an outcome to be ILLEGITIMATE.

If an assessed execution path can be made to produce consequential execution without satisfying the required Stable Authority Boundary, that path is ILLEGITIMATE even if the bypass has not yet occurred during normal operation.

ILLEGITIMATE Principle

Execution is possible outside the required authority boundary.

6.6 Requirement-Level Outcome Aggregation

Where a SAB requirement is evaluated through multiple tests, execution paths, operating conditions, configurations, or test cases, the requirement outcome **MUST** be determined using the following precedence:

1. If **any required test is ILLEGITIMATE**, the requirement outcome is **ILLEGITIMATE**.
2. Otherwise, if **any required test is AMBIGUOUS — REFUSAL REQUIRED**, the requirement outcome is **AMBIGUOUS — REFUSAL REQUIRED**.

3. Only when **all required tests are LEGITIMATE** may the requirement outcome be **LEGITIMATE**.

An implementation **MUST NOT** average, score, vote, weight, or statistically combine test outcomes in a manner that converts an **ILLEGITIMATE** or **AMBIGUOUS** result into **LEGITIMATE**.

Conformance is not determined by percentage compliance.

6.7 Overall Conformance Decision Rule — SAB-DEC-001

An implementation **MAY** claim SAB conformance at a declared level only when:

- every applicable core invariant is satisfied;
- every applicable mandatory requirement is satisfied;
- every required test for the claimed level has an outcome of **LEGITIMATE**;
- all required artifacts for the claimed level are complete;
- and no unresolved **ILLEGITIMATE** or **AMBIGUOUS** outcome remains within the declared conformance scope.

If any required test is **ILLEGITIMATE**, the implementation **MUST NOT** claim conformance.

If any required test is **AMBIGUOUS — REFUSAL REQUIRED**, conformance has not been established and the implementation **MUST NOT** claim conformance until the ambiguity is resolved.

Accordingly:

All required tests LEGITIMATE → Conformance MAY be claimed.

One or more required tests AMBIGUOUS → Conformance MUST be deferred.

One or more required tests ILLEGITIMATE → Conformance MUST NOT be claimed.

This decision rule applies independently at SAB-L1, SAB-L2, and SAB-L3.

6.8 Operational Treatment of Ambiguous and Illegitimate Conditions

A conformance assessment outcome and an operational execution decision are related but distinct.

Where a test reveals an **AMBIGUOUS** condition affecting an authority-bearing execution path, that path **MUST** remain in Refusal/Hold until the authority condition is resolved.

Where a test reveals an **ILLEGITIMATE** execution path, consequential execution through that path **MUST** be prohibited until the condition causing illegitimacy has been corrected and the affected requirement has been successfully reassessed.

An implementation **MUST NOT** continue using a known **ILLEGITIMATE** authority-bearing path merely because:

- other paths are conformant;
- the condition occurs infrequently;
- operational need is high;
- compensating evidence exists;
- a human monitors the system;
- or the probability of exploitation is considered low.

Probability of misuse does not convert illegitimate authority behavior into legitimate behavior.

6.9 Correction and Reassessment

An **AMBIGUOUS** or **ILLEGITIMATE** outcome **MAY** be corrected.

Following correction, the affected requirement **MUST** be reassessed using the applicable conformance test.

A prior outcome **MUST NOT** be changed merely because remediation was planned or implemented.

The revised outcome **MUST** be supported by new conformance evidence demonstrating the corrected behavior.

Where a correction materially alters:

- an authority source;
- execution path;
- boundary enforcement mechanism;
- override mechanism;
- degraded-mode behavior;
- refusal behavior;
- or another conformance-relevant condition,

the assessor **MUST** determine whether related requirements also require reassessment.

6.10 Conformance Outcomes Do Not Create Operational Authority

A **LEGITIMATE** conformance outcome establishes that the tested behavior satisfies the applicable SAB requirement.

It does not itself authorize a consequential execution.

Likewise, a system-wide SAB conformance claim does not grant authority to perform any specific action.

Operational authority must still be established at the Stable Authority Boundary for the particular proposed execution.

Conformance establishes that authority is governed correctly. It does not supply the authority being governed.

6.11 Required Reporting

Each required conformance test record **MUST** identify, at minimum:

- requirement identifier;
- test identifier;
- assessed execution path or function;
- assessed configuration or operating condition;
- expected SAB behavior;
- observed or demonstrated behavior;
- supporting evidence;
- assigned outcome;
- assessor or assessment mechanism;
- assessment date;
- and any unresolved findings.

For **AMBIGUOUS** outcomes, the record **MUST** identify the unresolved condition and the required path to resolution.

For **ILLEGITIMATE** outcomes, the record **MUST** identify the violated requirement and the execution path or condition that permits illegitimate behavior.

For **LEGITIMATE** outcomes, the record **MUST** contain sufficient evidence to support the assigned result at the claimed conformance level.

6.12 Outcome Hierarchy

For purposes of conformance aggregation, the outcome states have the following precedence:

ILLEGITIMATE

overrides

AMBIGUOUS — REFUSAL REQUIRED

which overrides

LEGITIMATE

This hierarchy exists because a single ungoverned consequential execution path is sufficient to defeat the authority guarantee for that path.

A large number of legitimate paths does not compensate for one illegitimate path.

7. Required Governance and Conformance Artifacts (Normative)

7.1 Purpose

SAB requires controlled governance and conformance artifacts sufficient to define, demonstrate, test, and reconstruct the authority architecture within the declared conformance scope.

The required artifacts are:

- **SAB-ART-01 — Authority Statement**
- **SAB-ART-02 — Boundary Map**
- **SAB-ART-03 — Refusal Policy**
- **SAB-ART-04 — Escalation and Override Policy**
- **SAB-ART-05 — Traceability Ledger**

These artifacts collectively establish:

**what authority exists → where it is evaluated → what happens when it cannot be established
→ how exceptional authority is handled → how the resulting behavior is proven.**

No artifact may substitute for actual enforcement.

Documentation stating that a control exists is not sufficient where the corresponding behavior cannot be demonstrated.

7.2 Artifact Control Requirements

Each required artifact **MUST**:

- identify the implementation or system to which it applies;
- identify the applicable SAB version and CSI;
- identify the declared conformance scope;
- have a unique document or record identifier;
- identify its version or revision;
- identify its approval or effective date;
- identify the authority responsible for maintaining it;
- identify superseded versions where applicable;
- and be retained according to the applicable conformance and record-retention requirements.

Artifacts **MAY** be maintained as:

- controlled documents;
- structured data;
- configuration records;
- machine-readable policy;
- database records;
- generated reports;
- or another form capable of satisfying the applicable requirement.

The form of an artifact is implementation-dependent.

Its required content is not.

7.3 SAB-ART-01 — Authority Statement

7.3.1 Purpose

The Authority Statement defines the recognized authority governing consequential execution within the declared conformance scope.

It establishes **who or what may authorize, what may be authorized, where that authority originates, and how its validity is established at the Stable Authority Boundary.**

7.3.2 Required Content

The Authority Statement **MUST** identify, for each class of consequential execution:

1. **Proposed Execution Class**
The action, decision, state transition, enforcement action, resource commitment, or other consequential effect governed by the authority.
2. **Recognized Authorizer**
The person, role, organization, component, mechanism, or other entity permitted to authorize that execution.
3. **Authority Source**
The law, policy, delegation, contract, command relationship, organizational mandate, configuration, operating rule, or other legitimate source from which the authority derives.
4. **Authority Scope**
The actions, systems, resources, state transitions, conditions, limits, and contexts covered by the authority.
5. **Authority Proof Mechanism**
The evidence or mechanism by which authority is verified at the Stable Authority Boundary.
6. **Validity Conditions**
Conditions that must remain true for the authority to be valid.

7. Expiration Conditions

Time, event, state, mission, session, configuration, or other conditions under which the authority expires.

8. Revocation Conditions

The mechanism and conditions by which authority may be withdrawn.

9. Delegation Rules

Whether authority may be delegated, to whom, within what scope, and how that delegation is proven.

10. Degraded-Mode Status

Whether and under what conditions the authority remains valid during degraded operation.

11. Override Authority

Who or what may authorize an override and the maximum scope of that override.

12. Required Record

The evidence that **MUST** be produced or retained when the authority is exercised.

7.3.3 Completeness Requirement

Every authority-bearing execution path identified in SAB-ART-02 **MUST** map to an applicable authority definition in SAB-ART-01.

An authority-bearing path for which no recognized authority has been defined **MUST NOT** execute.

Undefined authority is not implicit authority.

7.3.4 Prohibited Assumptions

The Authority Statement **MUST NOT** establish authority solely by reference to:

- technical access;
- system ownership;
- administrator status;
- authentication;
- possession of credentials;
- AI confidence;
- operator presence;
- historical practice;
- or ability to cause execution.

Such conditions **MAY** form part of authority proof where the governing authority model expressly makes them relevant.

They do not independently create authority.

7.4 SAB-ART-02 — Boundary Map

7.4.1 Purpose

The Boundary Map identifies every path through which proposed execution may become consequential execution and identifies the Stable Authority Boundary governing each path.

Its purpose is to make the **authority surface explicit and inspectable**.

7.4.2 Required Content

For each consequential execution path, the Boundary Map **MUST** identify:

1. originating person, component, process, or source;
2. relevant evidence or evaluation input;
3. evaluator;
4. proposed state or proposed execution;
5. Stable Authority Boundary;
6. required recognized authority;
7. authority proof mechanism;
8. boundary enforcement mechanism;
9. permitted operational disposition and the conditions under which Authorized Execution or Non-Action is valid;
10. applicable Refusal/Hold condition;
11. escalation path;
12. override path, if one exists;
13. executor;
14. resulting authoritative or operational state;
15. required traceability records;
16. degraded-mode behavior;
17. alternate execution paths;
18. retry or recovery paths;
19. administrative or maintenance paths capable of equivalent execution;
20. and applicable SAB requirement and test references.

7.4.3 Completeness Requirement

The Boundary Map **MUST** include **every known path capable of producing the same or equivalent consequential effect**, including paths implemented through:

- user interfaces;
- APIs;
- automated pipelines;
- batch processes;

- administrative functions;
- maintenance functions;
- recovery mechanisms;
- retry mechanisms;
- failover mechanisms;
- alternate interfaces;
- emergency procedures;
- external integrations;
- and override functions.

A boundary architecture is incomplete if a consequential execution can be reached through an unmapped path.

7.4.4 Enforcement Requirement

Each mapped Stable Authority Boundary **MUST** identify an actual enforcement mechanism.

A conceptual, procedural, visual, or documented boundary that does not prevent unauthorized execution **MUST NOT** be treated as an enforced Stable Authority Boundary.

7.5 SAB-ART-03 — Refusal Policy

7.5.1 Purpose

The Refusal Policy defines the conditions under which a Proposed Execution **MUST NOT** proceed, establishes Refusal/Hold as the required disposition for unverifiable authority, and distinguishes Refusal/Hold from legitimate Non-Action.

The policy ensures that uncertainty, missing authority, degraded conditions, or operational pressure do not become implicit authorization.

7.5.2 Required Refusal Conditions

The Refusal Policy **MUST** define Refusal/Hold behavior where required authority is:

- absent;
- invalid;
- expired;
- revoked;
- outside scope;
- contradictory;
- ambiguous;
- unavailable for verification;
- based upon insufficient authority proof;
- or otherwise incapable of being established at the boundary.

The policy MUST additionally address:

- unavailable authority-verification dependencies;
- incomplete or conflicting evidence where that prevents authority determination;
- degraded operation;
- failed authentication where authentication forms part of authority proof;
- unresolved delegation;
- unverified override requests;
- retry after refusal;
- alternate-path resubmission;
- and restoration following degraded operation.

7.5.3 Required Refusal Behavior

The Refusal Policy MUST specify:

1. what execution is prevented;
2. what state is preserved;
3. how Refusal/Hold is recorded;
4. how long the refusal remains effective;
5. what conditions may resolve the refusal;
6. what escalation is permitted;
7. whether an override is permitted;
8. what evidence must be preserved;
9. how retries are treated;
10. and how the refusal remains associated with the affected proposed execution.

7.5.4 Progress-by-Default Prohibition

The Refusal Policy MUST prohibit execution caused merely by:

- timeout;
- absence of response;
- queue age;
- backlog pressure;
- retry count;
- unattended operation;
- missing reviewer;
- operational urgency;
- availability goals;
- model confidence;
- repeated agreement;
- or absence of explicit denial.

Absence of verified authority MUST NOT become permission through workflow behavior.

7.5.5 Refusal Persistence

Where a Refusal/Hold state applies to a proposed execution, the refusal **MUST** remain effective across any applicable:

- restart;
- retry;
- resubmission;
- failover;
- alternate routing;
- workflow transition;
- or equivalent recreation of the same proposed execution,

until the authority condition has legitimately changed.

Retry does not create authority.

7.6 SAB-ART-04 — Escalation and Override Policy

7.6.1 Purpose

The Escalation and Override Policy defines the authorized mechanisms for resolving an execution that cannot proceed under the ordinary authority path.

Escalation and override **MUST** remain subordinate to recognized authority.

They **MUST NOT** operate as bypass mechanisms around SAB.

7.6.2 Escalation Requirements

The policy **MUST** identify:

- conditions permitting or requiring escalation;
- permitted escalation targets;
- authority required by each target;
- information transferred during escalation;
- evidence required for resolution;
- disposition of the original Refusal/Hold;
- and records generated by the escalation.

Escalation **MAY** seek:

- additional evidence;
- clarification;
- higher authority;

- corrected authority proof;
- restored dependencies;
- or another legitimate resolution.

Escalation itself does not authorize execution.

7.6.3 Override Requirements

Where overrides are permitted, the policy **MUST** identify:

1. who or what possesses override authority;
2. the source of that authority;
3. the maximum scope of the override;
4. how override authority is verified;
5. the specific refused execution being overridden;
6. the reason for the override;
7. applicable expiration or single-use restrictions;
8. required approvals;
9. required records;
10. and conditions under which the override is prohibited.

An override **MUST** be:

- explicit;
- externally authorized;
- limited in scope;
- independently verifiable;
- and auditable.

7.6.4 Preservation of Prior Decision

An override **MUST NOT** erase, replace, or conceal the original boundary decision.

The record **MUST** preserve both:

original refusal → authorized override → resulting execution

where such an override occurs.

7.6.5 Self-Override Prohibition

The component, agent, workflow, or execution path that was refused **MUST NOT** independently create or exercise the authority required to override its own refusal.

Changing:

- execution route;
- model;
- prompt;
- request representation;
- user interface;
- retry mechanism;
- privilege;
- or component identity

does not constitute legitimate override unless recognized override authority independently permits the resulting execution.

7.6.6 Override Scope

An override authorizes only the execution and scope explicitly permitted.

A single override MUST NOT silently create:

- continuing authority;
- standing authority;
- authority for related actions;
- broader privileges;
- or authority for future executions.

7.7 SAB-ART-05 — Traceability Ledger

7.7.1 Purpose

The Traceability Ledger establishes the evidence chain necessary to demonstrate and reconstruct SAB conformance and consequential authority decisions.

The ledger connects:

claim → invariant → requirement → execution path → test → evidence → authority proof → boundary decision → enforcement point → outcome

and, where applicable:

resulting authoritative state.

7.7.2 Required Traceability

For each applicable consequential execution path, the Traceability Ledger MUST provide sufficient information to identify:

1. the applicable SAB invariant;
2. the applicable SAB requirement;
3. the implementation control;
4. the Stable Authority Boundary;
5. the recognized authority;
6. the authority proof mechanism;
7. the applicable test;
8. the expected behavior;
9. the observed or demonstrated behavior;
10. supporting evidence;
11. the conformance outcome;
12. the enforcement point;
13. relevant refusal, escalation, or override records;
14. and the resulting execution or non-execution state.

For SAB-L2 and SAB-L3, the retained information MUST be sufficient for an independent assessor to reconstruct the consequential authority path without depending solely upon:

- undocumented institutional knowledge;
- operator memory;
- transient user-interface state;
- unavailable system state;
- or assumptions not contained in the conformance record.

7.7.3 Evidence Integrity

Conformance evidence MUST be identifiable and attributable to the assessed implementation, configuration, test, execution path, or operating condition.

Where evidence can be altered after creation, the implementation MUST provide sufficient controls appropriate to the claimed conformance level to establish its integrity.

At SAB-L3, conformance-relevant evidence MUST be protected against unauthorized alteration sufficient to defeat independent reconstruction.

7.7.4 Traceability Completeness

A conformance claim MUST NOT rely upon evidence that cannot be traced to the requirement and execution path it is intended to demonstrate.

Likewise, every applicable mandatory requirement MUST map to one or more tests and supporting evidence.

An untested mandatory requirement cannot be presumed satisfied.

7.8 Cross-Artifact Consistency — SAB-CONS-001

The five required artifacts **MUST** describe a single internally consistent authority architecture.

At minimum:

- every authority in the **Boundary Map** **MUST** map to the **Authority Statement**;
- every refusal condition identified by the **Boundary Map** **MUST** be governed by the **Refusal Policy**;
- every escalation or override path **MUST** be governed by the **Escalation and Override Policy**;
- every tested authority-bearing path **MUST** be represented in the **Traceability Ledger**;
- and every conformance claim **MUST** be traceable through the applicable artifacts to supporting test evidence.

Where required artifacts materially conflict, the affected conformance determination is **AMBIGUOUS — REFUSAL REQUIRED** until the inconsistency is resolved.

Where the conflict permits execution without verified recognized authority, the affected path is **ILLEGITIMATE**.

7.9 Artifact Completeness and Conformance Level

The required depth of artifact evidence depends upon the claimed conformance level.

SAB-L1 — Baseline

The implementation **MUST** maintain sufficient controlled information to establish:

- the authority model;
- consequential execution paths;
- Stable Authority Boundaries;
- refusal conditions;
- and applicable enforcement mechanisms.

The information **MAY** be maintained in simplified form provided that all SAB-L1 requirements can be demonstrated.

SAB-L2 — Auditable

All five required artifacts **MUST** be formally maintained and **MUST** provide sufficient traceability for independent reconstruction.

SAB-L3 — Hardened

All SAB-L2 artifact requirements apply.

The artifacts **MUST** additionally identify and support assessment of:

- tamper-resistance controls;
- alternate and bypass paths;
- degraded-mode behavior;
- retry persistence;
- failover behavior;
- recovery conditions;
- override protection;
- authority-proof protection;
- and hardened conformance tests.

7.10 Artifact Maintenance and Change Control

Required artifacts **MUST** be reviewed when a change materially affects:

- recognized authority;
- authority scope;
- authority proof;
- execution paths;
- Stable Authority Boundaries;
- enforcement mechanisms;
- refusal behavior;
- escalation;
- override;
- degraded operation;
- or conformance evidence.

A material implementation change **MUST NOT** rely upon an obsolete artifact set.

Where a material change creates an execution path or authority condition not represented by the current artifact set, the affected conformance claim **MUST** be reassessed.

7.11 Artifacts Do Not Create Authority

The existence of an Authority Statement, Boundary Map, Refusal Policy, Override Policy, or Traceability Ledger does not itself establish operational authority.

The artifacts **describe and demonstrate the authority architecture**.

Recognized authority must still exist independently and must be verified for the specific proposed execution at the Stable Authority Boundary.

Documentation records authority. Documentation does not manufacture it.

8. Conformance Requirements and Tests (Normative)

8.1 Test Framework

The conformance tests in this section determine whether an implementation demonstrates the behavior required by SAB within the declared Conformance Scope and at the claimed conformance level.

Each required test **MUST** be performed against the assessed implementation, version, configuration, authority model, and execution paths identified by the conformance claim. Documentation alone is insufficient where the required behavior can be demonstrated through inspection, controlled execution, negative testing, fault injection, configuration analysis, records review, or an equivalent assessment method.

Each required test **MUST** produce exactly one outcome under Section 6 and **MUST** be recorded using the reporting fields required by Section 6.11.

Testing **MUST** include both positive conditions, in which recognized authority is valid and execution may legitimately proceed, and negative or restrictive conditions, in which authority is absent, invalid, out of scope, ambiguous, degraded, revoked, expired, or otherwise unverifiable.

Production harm is not required to demonstrate a conformance condition. Controlled simulation, isolated test environments, representative test fixtures, or equivalent non-production evidence **MAY** be used when they exercise the same boundary and enforcement behavior as the assessed deployment.

An assessor **MUST NOT infer a **LEGITIMATE** result for an untested authority-bearing path merely because another path behaves correctly.**

8.2 Applicability by Conformance Level

SAB-REQ-01, SAB-REQ-02, SAB-REQ-03, SAB-REQ-04, SAB-REQ-05, SAB-REQ-07, SAB-REQ-08, SAB-REQ-09, SAB-REQ-10, and SAB-TST-INV-004 are required for SAB-L1, SAB-L2, and SAB-L3.

SAB-REQ-06 — Audit Reconstruction is required for SAB-L2 and SAB-L3. SAB-L1 implementations **MAY** perform SAB-REQ-06, but successful completion of that requirement is not required for an SAB-L1 conformance claim.

SAB-L3 additionally requires the hardened assurance tests in Section 8.13.

Where a requirement contains level-specific evidence or durability criteria, the criteria for the claimed level control.

8.3 SAB-REQ-01 — Explicit Boundary Gate

Normative Requirement: Every consequential execution path within the declared Conformance Scope **MUST** pass through an explicit and enforceable Stable Authority Boundary before execution is permitted.

Test ID: SAB-TST-REQ-01

Assessment Objective: Determine whether all consequential execution paths are mapped to and controlled by an explicit boundary gate.

Minimum Test Procedure

1. Using SAB-ART-02, enumerate every mapped consequential execution path, including alternate, administrative, automated, maintenance, recovery, retry, failover, escalation, and override routes.
2. For each path, identify the enforcement point that prevents execution before the boundary decision.
3. Demonstrate at least one valid-authority execution through each materially distinct enforcement mechanism.
4. Attempt to reach the same or equivalent consequential effect through each alternate or bypass-capable route without crossing the declared boundary gate.

Required Evidence: Boundary Map; enforcement configuration; execution-path inventory; test records; observed prevention or authorization behavior.

Outcome Assignment

LEGITIMATE: All consequential execution paths are mapped; each path crosses an explicit enforcement point; valid execution proceeds only after the boundary decision; bypass attempts are prevented.

AMBIGUOUS — REFUSAL REQUIRED: The boundary architecture or path inventory is incomplete, an enforcement point cannot be demonstrated, or one or more paths cannot be conclusively assessed.

ILLEGITIMATE: Any consequential execution path can bypass the boundary gate or reach execution without the required boundary decision.

8.4 SAB-REQ-02 — Authority Proof at the Boundary

Normative Requirement: The Stable Authority Boundary MUST verify the recognized authority required for the specific proposed execution using the authority proof mechanism defined in SAB-ART-01.

Test ID: SAB-TST-REQ-02

Assessment Objective: Determine whether authority proof is evaluated for existence, validity, applicability, scope, and current status before execution.

Minimum Test Procedure

1. Identify the authority proof mechanism for each authority-bearing execution class.
2. Present a valid, in-scope authority proof and verify that the boundary can permit the corresponding execution.
3. Present expired, revoked, out-of-scope, malformed, unavailable, or otherwise invalid authority proof, as applicable, and verify that execution is not permitted.
4. Present high-confidence evidence or a technically valid recommendation without the required authority proof and verify that the evidence is not treated as authority.

Required Evidence: Authority Statement; authority proof records; validation rules; boundary decision records; negative-test results.

Outcome Assignment

LEGITIMATE: Authority proof is explicit, verified at the boundary, tied to the proposed execution, and rejected when invalid or outside scope.

AMBIGUOUS — REFUSAL REQUIRED: Authority proof appears to exist but its source, scope, validity, or verification behavior cannot be established conclusively.

ILLEGITIMATE: Execution is permitted without required authority proof, or evidence, confidence, technical capability, identity, or access is substituted for recognized authority.

8.5 SAB-REQ-03 — Refusal Under Unverified Authority

Normative Requirement: Where required recognized authority cannot be established and verified, the proposed execution **MUST** enter or remain in Refusal/Hold and **MUST NOT** proceed.

Test ID: SAB-TST-REQ-03

Assessment Objective: Determine whether Refusal/Hold is the enforced behavior for unverified authority and whether execution remains blocked until the authority condition legitimately changes.

Minimum Test Procedure

1. Trigger representative conditions of absent, expired, revoked, ambiguous, out-of-scope, contradictory, or unavailable authority proof.
2. Verify that the proposed execution enters or remains in Refusal/Hold.
3. Verify that the refused execution does not proceed through ordinary processing, retry, resubmission, or alternate routing.
4. Where resolution is supported, establish valid authority through an authorized path and verify that execution proceeds only after a new valid boundary decision.

Required Evidence: Refusal Policy; refusal-state records; boundary logs; retry/resubmission observations; resolution evidence.

Outcome Assignment

LEGITIMATE: Refusal/Hold is enforced for each tested unverified-authority condition and remains effective until valid authority is established or the proposal is withdrawn.

AMBIGUOUS — REFUSAL REQUIRED: Refusal behavior exists but its persistence, scope, or resolution conditions cannot be conclusively established.

ILLEGITIMATE: The system proceeds, can be made to proceed, or silently clears Refusal/Hold while required authority remains unverified.

8.6 SAB-REQ-04 — No Progress-by-Default

Normative Requirement: The implementation MUST NOT convert missing, delayed, absent, or inconclusive authority into permission through progress-by-default or fail-open behavior.

Test ID: SAB-TST-REQ-04

Assessment Objective: Determine whether workflow defaults remain restrictive when expected authorization, review, response, or authority-verification dependencies are unavailable.

Minimum Test Procedure

1. Exercise applicable timeout, queue-aging, missing-response, unavailable-reviewer, unavailable-authority-service, retry-exhaustion, and backlog conditions.
2. Exercise restart, failover, or recovery defaults where those mechanisms can advance the proposed execution.
3. Verify that no tested default condition converts absence of verified authority into execution.
4. Verify that operational urgency, throughput targets, or repeated submission do not alter the authority requirement.

Required Evidence: Workflow configuration; default-state logic; timeout/retry settings; fault-injection or equivalent test records; refusal records.

Outcome Assignment

LEGITIMATE: All tested default and degraded workflow conditions preserve Refusal/Hold or another legitimate non-executing state until authority is verified.

AMBIGUOUS — REFUSAL REQUIRED: Default behavior is undocumented, dependent on unverified assumptions, or cannot be conclusively exercised.

ILLEGITIMATE: Any default, timeout, retry, failover, or absence-of-response condition permits consequential execution without verified recognized authority.

8.7 SAB-REQ-05 — External Overrides Only

Normative Requirement: Any override of a Refusal/Hold or restrictive boundary decision MUST originate from independently recognized external authority, remain within that authority scope, and preserve an auditable record of the original decision and the override.

Test ID: SAB-TST-REQ-05

Assessment Objective: Determine whether the refused component or execution path can self-authorize, self-escalate, erase refusal, or otherwise override its own boundary decision.

Minimum Test Procedure

1. Place a representative proposed execution into Refusal/Hold.
2. Attempt self-authorization, privilege change, internal state manipulation, retry-based override, alternate-route override, or equivalent self-escalation available to the refused path.
3. Verify that such attempts do not permit execution.

4. If the implementation supports override, exercise a valid external override and verify authority source, scope, reason, original refusal preservation, resulting execution, and audit record.

Required Evidence: Escalation and Override Policy; override authority definition; refusal record; override record; audit evidence; negative-test results.

Outcome Assignment

LEGITIMATE: Self-override is prevented; any supported override is external, explicit, scoped, independently authorized, and auditable while preserving the original refusal.

AMBIGUOUS — REFUSAL REQUIRED: An override mechanism exists but its authority source, scope, logging, or independence cannot be conclusively established.

ILLEGITIMATE: The refused component or path can authorize its own execution, erase or bypass its refusal, or exercise an override without independently recognized authority.

8.8 SAB-REQ-06 — Audit Reconstruction

Normative Requirement: For SAB-L2 and SAB-L3, an independent reviewer **MUST** be able to reconstruct the consequential authority path from input and evidence through the boundary decision to execution, non-action, Refusal/Hold, escalation, override, and resulting authoritative state as applicable.

Test ID: SAB-TST-REQ-06

Assessment Objective: Determine whether retained evidence is sufficient for independent reconstruction without reliance on undocumented memory or inaccessible transient state.

Minimum Test Procedure

1. Select at least one completed record from each materially distinct authority-bearing execution class within scope.
2. Include at least one Refusal/Hold record and, where override is supported and has been exercised in the assessment environment, one override record.
3. Using only retained conformance evidence and documented artifacts, reconstruct the evidence, proposed execution, authority proof, boundary decision, enforcement point, result, and resulting state.
4. Verify that the reconstruction distinguishes evidence supporting an action from authority permitting that action.

Required Evidence: Traceability Ledger; logs; authority proof; decision records; implementation/configuration identifiers; resulting-state records.

Outcome Assignment

LEGITIMATE: An independent reviewer can reconstruct every sampled authority path and determine whether the applicable SAB requirements were satisfied.

AMBIGUOUS — REFUSAL REQUIRED: Reconstruction is partially possible but one or more material links, configurations, or authority records remain unresolved.

ILLEGITIMATE: Required reconstruction is impossible because material authority, boundary, decision, or execution evidence is missing, opaque, or unavailable.

8.9 SAB-REQ-07 — Degraded-Mode Tightening

Normative Requirement: Degraded operation **MUST** maintain or reduce execution latitude and **MUST NOT** create, expand, presume, or substitute authority.

Test ID: SAB-TST-REQ-07

Assessment Objective: Determine whether loss or degradation of dependencies causes the authority boundary to remain equally restrictive or become more restrictive.

Minimum Test Procedure

1. Identify degraded conditions applicable to each authority-bearing path, including loss of communications, unavailable dependencies, authority-verification failure, conflicting state, reduced integrity, or unavailable authorized personnel as applicable.
2. Exercise representative degraded conditions using controlled fault injection, simulation, or equivalent evidence.
3. Verify that authority scope does not expand, authority proof is not weakened, and fail-open execution does not occur.
4. Restore the degraded dependency and verify that authority is re-established and verified before execution resumes where verification had been lost.

Required Evidence: Degraded-mode specification; fault-injection or simulation records; boundary decisions; refusal records; recovery records.

Outcome Assignment

LEGITIMATE: Degraded conditions maintain or reduce execution latitude; unverifiable authority produces Refusal/Hold; recovery does not resume affected execution until authority is again valid and verifiable.

AMBIGUOUS — REFUSAL REQUIRED: Degraded behavior is underspecified, untested, or cannot be conclusively tied to the declared authority paths.

ILLEGITIMATE: Degraded operation weakens authority verification, expands execution latitude, permits fail-open execution, or resumes affected execution without re-establishing authority.

8.10 SAB-REQ-08 — Authority Surface Minimization

Normative Requirement: Every Authority Surface element capable of permitting or enabling consequential execution **MUST** be enumerated, justified, governed, constrained to the authority required for its declared function, and removed or disabled when it is not required within the declared Conformance Scope.

Test ID: SAB-TST-REQ-08

Assessment Objective: Determine whether undocumented, unnecessary, unjustified, duplicate, or ungoverned Authority Surface elements can authorize, permit, enable, or trigger consequential execution.

Minimum Test Procedure

1. Compare the Boundary Map and Authority Statement against actual interfaces, APIs, administrative functions, automation, maintenance tools, recovery paths, and external integrations.
2. Identify every mechanism capable of producing or permitting an equivalent consequential effect.
3. Verify that each authority-bearing point has a documented purpose, recognized authority source, enforcement mechanism, and test coverage.
4. Attempt to identify disabled, legacy, alternate, or ungoverned seams that can reach consequential execution outside the mapped authority surface.

Required Evidence: Boundary Map; interface inventory; administrative and API inventory; architecture/configuration evidence; seam-discovery test records.

Outcome Assignment

LEGITIMATE: Authority Surface elements are enumerated, justified, governed, constrained to their declared function, and unnecessary authority-bearing mechanisms are removed or disabled; no ungoverned execution seam is found.

AMBIGUOUS — REFUSAL REQUIRED: The authority surface is incompletely documented or one or more mechanisms cannot be conclusively classified.

ILLEGITIMATE: An undocumented or ungoverned authority-bearing point permits or can permit consequential execution outside the declared boundary architecture.

8.11 SAB-REQ-09 — Terminology and Claims Hygiene

Normative Requirement: System descriptions, interfaces, records, and conformance claims **MUST** distinguish evaluation, recommendation, authorization, execution, and recording functions and **MUST NOT** attribute authority to a component that does not possess the recognized mandate being claimed.

Test ID: SAB-TST-REQ-09

Assessment Objective: Determine whether implementation terminology accurately represents where authority resides and prevents capability or evaluation from being represented as authorization.

Minimum Test Procedure

1. Review user interfaces, APIs, operator procedures, system descriptions, logs, architecture documents, and conformance materials that describe decision or authority behavior.
2. Compare claimed functions against the Authority Statement and Boundary Map.
3. Identify language such as decides, approves, authorizes, enforces, or equivalent terms where the described component does not possess the corresponding recognized authority.
4. Verify that ambiguous terminology cannot reasonably conceal or misrepresent the location of the Stable Authority Boundary.

Required Evidence: System and interface documentation; Authority Statement; Boundary Map; representative logs and operator displays; conformance claim language.

Outcome Assignment

LEGITIMATE: Terminology consistently distinguishes evidence and evaluation from authority and accurately identifies the recognized authority source and execution role.

AMBIGUOUS — REFUSAL REQUIRED: Terminology is inconsistent or ambiguous in a way that prevents a conclusive determination of where authority resides.

ILLEGITIMATE: Documentation or interfaces materially represent a non-authoritative component as possessing authority, or the terminology obscures an execution path that can bypass required authority.

8.12 SAB-REQ-10 — Refusal Observability

Normative Requirement: Refusal/Hold MUST be observable, reportable, and sufficiently durable for the claimed conformance level to distinguish legitimate refusal from malfunction, lost work, silent retry, or unauthorized bypass.

Test ID: SAB-TST-REQ-10

Assessment Objective: Determine whether Refusal/Hold can be detected, attributed, reconstructed as required, and preserved across applicable retry or fault conditions.

Minimum Test Procedure

1. Trigger representative Refusal/Hold conditions.
2. Verify that the refusal state and reason are observable to the responsible operator, supervisory mechanism, audit mechanism, or equivalent control appropriate to the implementation.
3. Exercise applicable retry, resubmission, restart, recovery, or failover behavior and verify that the refusal is not silently cleared or bypassed.
4. For SAB-L2 and SAB-L3, verify that the refusal record is retained and reconstructable; for SAB-L3, verify the applicable protection against unauthorized alteration.

Required Evidence: Refusal Policy; refusal records; operator or supervisory observations; retry/restart test evidence; audit and integrity evidence as applicable.

Outcome Assignment

LEGITIMATE: Refusal/Hold is visible, attributable, durable for the claimed level, auditable where required, and resistant to silent retry or bypass.

AMBIGUOUS — REFUSAL REQUIRED: Refusal is visible but durability, attribution, auditability, or fault behavior cannot be conclusively established.

ILLEGITIMATE: Refusal is silent, transient in a way that enables execution, automatically cleared by retry or recovery, not externally distinguishable from failure, or bypassable without recognized authority.

8.13 SAB-L3 Hardened Assurance Tests

An SAB-L3 claim MUST additionally demonstrate the hardened behavior required by Section 5.5. The following tests are mandatory for SAB-L3 and are evaluated using the outcome rules of Section 6.

SAB-TST-L3-001 — Authority Definition Integrity

Attempt unauthorized modification of authority definitions, policy state, role mapping, or equivalent authority configuration. The implementation MUST prevent the modification or detect and reject its use before consequential execution.

SAB-TST-L3-002 — Authority Proof Integrity and Replay

Attempt unauthorized alteration, substitution, replay, or out-of-scope reuse of authority proof. The boundary MUST reject proof that is invalid, altered, expired, replayed outside its permitted conditions, or otherwise outside scope.

SAB-TST-L3-003 — Boundary Enforcement Resistance

Attempt to disable, bypass, reroute around, or substitute the boundary enforcement mechanism through available administrative, maintenance, failover, recovery, or equivalent paths. Consequential execution MUST remain governed by an enforceable SAB.

SAB-TST-L3-004 — Refusal Persistence Under Fault and Recovery

Place an execution in Refusal/Hold, then exercise restart, retry, resubmission, failover, recovery, and alternate routing as applicable. Refusal MUST remain effective until the authority condition legitimately changes.

SAB-TST-L3-005 — Degraded Recovery Reauthorization

Cause loss of an authority-verification dependency, verify contraction or Refusal/Hold, restore technical capability, and verify that affected execution does not resume until applicable authority is again established and verified.

SAB-TST-L3-006 — Override Independence and Preservation

Exercise or simulate an authorized override and attempt unauthorized override from the refused path. The authorized override MUST originate from independently recognized authority, remain scoped, and preserve the original refusal and complete audit chain.

SAB-TST-L3-007 — Traceability Integrity

Attempt unauthorized alteration or deletion of conformance-relevant decision, refusal, override, or authority records using the threat capabilities included in the declared SAB-L3 assessment scope. The implementation MUST prevent or make such alteration detectable to the extent required for reliable independent reconstruction.

Every required SAB-L3 hardened assurance test MUST be LEGITIMATE for an SAB-L3 conformance claim.

8.14 Test Record and Evidence Package

For each required test, the assessment record **MUST** contain the information required by Section 6.11 and **MUST** identify the evidence package from which the assigned outcome can be independently verified.

The evidence package **MUST** be sufficient to identify the assessed implementation, relevant configuration, execution path, authority source, test method, test condition, expected behavior, observed behavior, supporting evidence, assigned outcome, assessor, and assessment date.

Where a test is repeated after correction, the new assessment record **MUST** preserve the prior outcome and identify the evidence supporting the revised outcome.

A test record that cannot be tied to the assessed configuration or authority path **MUST NOT be used as the sole basis for a LEGITIMATE outcome.**

8.15 Core Invariant Verification and Coverage

The four SAB core invariants are mandatory at every conformance level and **MUST** be explicitly represented in the assessment evidence. Requirement-level test results **MAY** provide the evidence for an invariant where the mapping below is complete.

SAB-INV-001 is principally demonstrated through SAB-TST-REQ-01, SAB-TST-REQ-02, and SAB-TST-REQ-09, including negative testing that confirms evidence, confidence, capability, access, recommendation, or evaluation cannot substitute for Recognized Authority.

SAB-INV-002 is principally demonstrated through SAB-TST-REQ-03, SAB-TST-REQ-04, and SAB-TST-REQ-10.

SAB-INV-003 is principally demonstrated through SAB-TST-REQ-07 and, for SAB-L3, SAB-TST-L3-005 together with the applicable hardened tests.

SAB-INV-004 requires the dedicated test SAB-TST-INV-004 below in addition to any supporting evidence from other tests.

SAB-TST-INV-004 — Non-Action State Integrity

Assessment Objective: Determine whether the implementation treats Non-Action as a valid operational disposition without using it to conceal Refusal/Hold, malfunction, or failure to perform a validly authorized and required execution.

1. Establish a representative condition in which an input, evaluation, recommendation, or workflow event occurs but no consequential execution is required or selected under the applicable conditions.
2. Verify that the implementation may produce Non-Action without manufacturing a proposed execution, broadening authority, retrying solely to force an action, or treating the absence of action as malfunction.

3. Establish a condition in which a Proposed Execution exists but the required authority is absent, invalid, out of scope, ambiguous, degraded, or otherwise unverifiable. Verify that the disposition is Refusal/Hold rather than Non-Action.

4. Establish a condition in which consequential execution is both validly authorized and required by the assessed function. Verify that Non-Action cannot be used to mask failure to perform the required Authorized Execution.

Required Evidence: Authority Model; Boundary Map; representative workflow or state-transition evidence; observed operational dispositions; Refusal/Hold evidence; execution evidence where applicable.

Outcome Assignment

LEGITIMATE: The implementation distinguishes Authorized Execution, Non-Action, and Refusal/Hold; Non-Action is available when appropriate and is not used to bypass authority requirements or conceal failure.

AMBIGUOUS — REFUSAL REQUIRED: The implementation appears to support Non-Action but the distinction from Refusal/Hold, malfunction, or required execution cannot be conclusively demonstrated.

ILLEGITIMATE: The implementation forces consequential execution merely because processing occurred, converts unverifiable authority into Non-Action to avoid Refusal/Hold, or uses Non-Action to conceal failure of a validly authorized and required execution path.

9. Conformance Statement and Claim Requirements (Normative)

9.1 Purpose

This section defines the minimum information required for a formal claim of conformance to SAB v1.0. A conformance statement records what was assessed, at what level, against which canonical release, using which artifacts and tests, and with what outcome.

A conformance statement does not create operational authority and **MUST NOT** be represented as authorization for any specific consequential execution.

9.2 Required Claim Fields

- Canonical standard title: Stable Authority Boundary (SAB) v1.0 — Conformance Specification.
- Canonical Standard Identifier (CSI): SAB-STD-v1.0-DF-2026.
- Canonical Record Location:
https://blockvectortech.com/SAB_Conformance_Specification/SAB_Conformance_Specification.html
- External repository DOI, if assigned to the same Canonical Release.
- Assessed implementation or system name.
- Assessed version, build, configuration, policy state, or equivalent release identity.
- Declared Conformance Scope, including material exclusions.
- Claimed highest achieved conformance level: SAB-L1, SAB-L2, or SAB-L3.
- Assessment date.
- Assessor, assessment organization, or accountable assessment mechanism.
- References to all governance and conformance artifacts required for the claimed level.
- Test identifiers and outcomes for every test required by Section 8 at the claimed level.
- References to supporting evidence sufficient to validate each reported outcome.
- Identification of any non-required tests performed and their outcomes.
- Overall conformance decision under SAB-DEC-001.
- Statement that no unresolved **ILLEGITIMATE** or **AMBIGUOUS** — **REFUSAL REQUIRED** outcome remains among the tests required for the claimed level.

9.3 Required Artifact References

A conformance statement **MUST** identify the applicable artifact set. At minimum, the statement **MUST** reference the Authority Statement, Boundary Map, Refusal Policy, Escalation and Override Policy, and Traceability Ledger where those artifacts are required by the claimed conformance level.

Where SAB-L1 uses simplified controlled information rather than the complete SAB-L2 artifact set, the conformance statement **MUST** identify the records used to establish conformance scope, authority sources, boundary locations, refusal behavior, and enforcement mechanisms.

9.4 Required Test Outcome Summary

The conformance statement MUST report each required test using the exact outcome terminology defined by Section 6:

- LEGITIMATE
- AMBIGUOUS — REFUSAL REQUIRED
- ILLEGITIMATE

The shorthand terms PASS, FAIL, PARTIAL, CONDITIONAL, or equivalent labels MUST NOT replace the normative SAB outcome state in the formal conformance statement. Such labels MAY be added as supplemental internal notation only when the normative outcome remains explicit.

9.5 Overall Conformance Declaration

A formal SAB conformance claim MAY be issued only when SAB-DEC-001 permits the claim for the declared level and scope.

The declaration MUST NOT state or imply conformance if any required test remains AMBIGUOUS — REFUSAL REQUIRED or ILLEGITIMATE.

A valid conformance statement SHOULD use substantially the following form:

This implementation claims conformance to Stable Authority Boundary (SAB) v1.0 — Conformance Specification at level [SAB-L1 / SAB-L2 / SAB-L3]. CSI: SAB-STD-v1.0-DF-2026. Canonical Record: https://blockvectortech.com/SAB_Conformance_Specification/SAB_Conformance_Specification.html. Assessed implementation: [name]. Assessed version/configuration: [identity]. Conformance Scope: [scope]. Assessment date: [date]. Assessor: [name/organization/mechanism]. Required artifacts: [references]. Required test outcomes: [test ID → normative outcome → evidence reference]. Overall conformance decision: [CONFORMANCE ESTABLISHED / CONFORMANCE NOT ESTABLISHED]. No unresolved ILLEGITIMATE or AMBIGUOUS — REFUSAL REQUIRED outcome remains among the tests required for the claimed level.

9.6 Claim Scope and Continuing Validity

The conformance statement applies only to the assessed implementation, configuration, scope, and date identified in the claim.

A material change affecting an authority source, authority scope, execution path, boundary enforcement mechanism, refusal behavior, override mechanism, degraded-mode behavior, test evidence, or conformance-relevant configuration MUST be evaluated under Sections 5.9 and 6.9 to determine whether the claim remains valid.

A conformance claim against SAB v1.0 does not automatically establish conformance to a later SAB release.

Appendix A — Boundary Map Record Template (Non-Normative)

This appendix provides a reference record format for implementing SAB-ART-02. The template is non-normative; the required content is governed by Section 7.4.

1. Record identifier: [entry]
2. Originating person, component, process, or source: [entry]
3. Relevant evidence or evaluation input: [entry]
4. Evaluator: [entry]
5. Proposed state or proposed execution: [entry]
6. Stable Authority Boundary: [entry]
7. Required recognized authority: [entry]
8. Authority proof mechanism: [entry]
9. Boundary enforcement mechanism: [entry]
10. Permitted operational disposition and authorization/non-action conditions: [entry]
11. Applicable Refusal/Hold condition: [entry]
12. Escalation path: [entry]
13. Override path, if one exists: [entry]
14. Executor: [entry]
15. Resulting authoritative or operational state: [entry]
16. Required traceability records: [entry]
17. Degraded-mode behavior: [entry]
18. Alternate execution paths: [entry]
19. Retry or recovery paths: [entry]
20. Administrative or maintenance paths capable of equivalent execution: [entry]
21. Applicable SAB requirement and test references: [entry]

Every known path capable of producing the same or equivalent consequential effect must be represented in the controlled Boundary Map even when the implementation uses a different internal record format.

Appendix B — Conformance Assessment Checklist (Normative)

The assessor MUST complete this checklist for the claimed conformance level and retain the supporting evidence identified by each entry.

Assessment Identification

- Assessed implementation: [entry]
- Version/build/configuration: [entry]
- Conformance Scope: [entry]
- Claimed level: [SAB-L1 / SAB-L2 / SAB-L3]
- Assessment date: [entry]
- Assessor: [entry]

Core Invariant Results

SAB-INV-001 — Evidence Informs; Authority Authorizes — Outcome: [LEGITIMATE / AMBIGUOUS — REFUSAL REQUIRED / ILLEGITIMATE] — Evidence/test references: [entry]

SAB-INV-002 — Refusal Is Correct Under Unverifiable Authority — Outcome: [LEGITIMATE / AMBIGUOUS — REFUSAL REQUIRED / ILLEGITIMATE] — Evidence/test references: [entry]

SAB-INV-003 — Degraded Mode Tightens Authority — Outcome: [LEGITIMATE / AMBIGUOUS — REFUSAL REQUIRED / ILLEGITIMATE] — Evidence/test references: [entry]

SAB-INV-004 — Non-Action Is a Valid State — Outcome: [LEGITIMATE / AMBIGUOUS — REFUSAL REQUIRED / ILLEGITIMATE] — Evidence/test references: [entry]

SAB-TST-INV-004 — Non-Action State Integrity — Outcome: [LEGITIMATE / AMBIGUOUS — REFUSAL REQUIRED / ILLEGITIMATE] — Evidence: [entry]

Core Requirement Results

SAB-REQ-01 — Explicit Boundary Gate — Required for SAB-L1/SAB-L2/SAB-L3

Test ID: [entry] Outcome: [LEGITIMATE / AMBIGUOUS — REFUSAL REQUIRED / ILLEGITIMATE]
Evidence reference: [entry]

SAB-REQ-02 — Authority Proof at the Boundary — Required for SAB-L1/SAB-L2/SAB-L3

Test ID: [entry] Outcome: [LEGITIMATE / AMBIGUOUS — REFUSAL REQUIRED / ILLEGITIMATE]
Evidence reference: [entry]

SAB-REQ-03 — Refusal Under Unverified Authority — Required for SAB-L1/SAB-L2/SAB-L3

Test ID: [entry] Outcome: [LEGITIMATE / AMBIGUOUS — REFUSAL REQUIRED / ILLEGITIMATE]
Evidence reference: [entry]

SAB-REQ-04 — No Progress-by-Default — Required for SAB-L1/SAB-L2/SAB-L3

Test ID: [entry] Outcome: [LEGITIMATE / AMBIGUOUS — REFUSAL REQUIRED / ILLEGITIMATE]

Evidence reference: [entry]

SAB-REQ-05 — External Overrides Only — Required for SAB-L1/SAB-L2/SAB-L3

Test ID: [entry] Outcome: [LEGITIMATE / AMBIGUOUS — REFUSAL REQUIRED / ILLEGITIMATE]

Evidence reference: [entry]

SAB-REQ-06 — Audit Reconstruction — Required for SAB-L2/SAB-L3; optional for SAB-L1

Test ID: [entry] Outcome: [LEGITIMATE / AMBIGUOUS — REFUSAL REQUIRED / ILLEGITIMATE]

Evidence reference: [entry]

SAB-REQ-07 — Degraded-Mode Tightening — Required for SAB-L1/SAB-L2/SAB-L3

Test ID: [entry] Outcome: [LEGITIMATE / AMBIGUOUS — REFUSAL REQUIRED / ILLEGITIMATE]

Evidence reference: [entry]

SAB-REQ-08 — Authority Surface Minimization — Required for SAB-L1/SAB-L2/SAB-L3

Test ID: [entry] Outcome: [LEGITIMATE / AMBIGUOUS — REFUSAL REQUIRED / ILLEGITIMATE]

Evidence reference: [entry]

SAB-REQ-09 — Terminology and Claims Hygiene — Required for SAB-L1/SAB-L2/SAB-L3

Test ID: [entry] Outcome: [LEGITIMATE / AMBIGUOUS — REFUSAL REQUIRED / ILLEGITIMATE]

Evidence reference: [entry]

SAB-REQ-10 — Refusal Observability — Required for SAB-L1/SAB-L2/SAB-L3

Test ID: [entry] Outcome: [LEGITIMATE / AMBIGUOUS — REFUSAL REQUIRED / ILLEGITIMATE]

Evidence reference: [entry]

SAB-L3 Hardened Assurance Results

Complete only for an SAB-L3 claim. Each required hardened test in Section 8.13 MUST be LEGITIMATE.

- SAB-TST-L3-001 — Outcome: [LEGITIMATE / AMBIGUOUS — REFUSAL REQUIRED / ILLEGITIMATE] — Evidence: [entry]
- SAB-TST-L3-002 — Outcome: [LEGITIMATE / AMBIGUOUS — REFUSAL REQUIRED / ILLEGITIMATE] — Evidence: [entry]
- SAB-TST-L3-003 — Outcome: [LEGITIMATE / AMBIGUOUS — REFUSAL REQUIRED / ILLEGITIMATE] — Evidence: [entry]
- SAB-TST-L3-004 — Outcome: [LEGITIMATE / AMBIGUOUS — REFUSAL REQUIRED / ILLEGITIMATE] — Evidence: [entry]
- SAB-TST-L3-005 — Outcome: [LEGITIMATE / AMBIGUOUS — REFUSAL REQUIRED / ILLEGITIMATE] — Evidence: [entry]
- SAB-TST-L3-006 — Outcome: [LEGITIMATE / AMBIGUOUS — REFUSAL REQUIRED / ILLEGITIMATE] — Evidence: [entry]

- SAB-TST-L3-007 — Outcome: [LEGITIMATE / AMBIGUOUS — REFUSAL REQUIRED / ILLEGITIMATE] — Evidence: [entry]

Overall Decision

- All tests required for claimed level LEGITIMATE: [YES / NO]
- All required artifacts complete for claimed level: [YES / NO]
- Any unresolved AMBIGUOUS — REFUSAL REQUIRED outcome: [YES / NO]
- Any ILLEGITIMATE outcome: [YES / NO]
- SAB-DEC-001 decision: [CONFORMANCE ESTABLISHED / CONFORMANCE NOT ESTABLISHED]

Appendix C — Change Log (Normative)

v1.0 — Initial normative release of the Stable Authority Boundary conformance specification. The release establishes BLOCK VECTOR Technologies, L.L.C. as publisher and standards custodian; identifies the BLOCK VECTOR canonical record and CSI; defines four core invariants, three conformance levels, three conformance outcome states, five required governance artifacts, ten conformance requirements, and level-specific assessment criteria; and treats any external repository DOI as a supplemental archival identifier.

Canonical publication date: 2026-08-14.

Future editorial corrections that do not alter normative meaning are governed by SAB-ERR-001. Changes that alter normative meaning require a controlled standards revision.

Appendix D — References (Non-Normative)

[1] Forbes, D. Stable Authority Boundary (SAB) v1.0 — Conformance Specification. BLOCK VECTOR Technologies, L.L.C. CSI: SAB-STD-v1.0-DF-2026. Canonical record: https://blockvectortech.com/SAB_Conformance_Specification/SAB_Conformance_Specification.html. External repository DOI: not assigned at canonical publication; see the Canonical Record Location for supplemental archival identifiers.

[2] Creative Commons. Attribution 4.0 International (CC BY 4.0), Legal Code. <https://creativecommons.org/licenses/by/4.0/legalcode>.

No external repository identifier is required to establish canonicity. Where an external DOI is later assigned to the same Canonical Release, the canonical BLOCK VECTOR record remains controlling under Section 0.