Change Control and Versioning Policy

ARI Institutional Coordinator

Aurora Research Initiative (ARI)

Change Control and Versioning Policy (v1.0.0)

Author: Shawn C. Wright
Affiliation: Waveframe Labs — Independent Open-Science Research Entity
ORCID: 0009-0006-6043-9295
Creation Date: 2025-11-27
Concept DOI: https://doi.org/10.5281/zenodo.17743096

Change Control and Versioning Policy (v1.0.0)

This document defines the **binding governance rules** for versioning, release control, and institutional change management across the Aurora Research Initiative (ARI), Aurora Workflow Orchestration (AWO), CRI-CORE, Waveframe Labs, and all Case Studies.
It establishes a **dual-versioning model** and mandates **global release tagging** for every version change, ensuring reproducibility, auditability, and long-term institutional stability.

---

1. Purpose

The purpose of this policy is to:

- prevent uncontrolled evolution of any subsystem
- enforce version discipline across all institutional layers
- define authoritative rules for version increments
- ensure every change is traceable and documented
- bind AWO and CRI evolution to ARI governance
- enable independent subsystem evolution without coupling
- maintain long-term reproducibility and provenance

This is a constitutional governance document and is binding for all contributors.

---

2. Dual Versioning Model

The Aurora ecosystem uses **independent versioning** for each subsystem, plus a **required global release tag** for every change.

2.1 Subsystem Versions

Each subsystem maintains its own semantic version:

- `ARI_VERSION`
- `AWO_VERSION`
- `CRI_VERSION`
- `LABS_VERSION` (optional)
- `CASE_STUDY_VERSION` (per project)

Each follows MAJOR.MINOR.PATCH.

2.2 Global Ecosystem Release Tag (Required)

Every version change **must** produce a unified global tag in the format:

```
ARI-x.y.z_AWO-a.b.c_CRI-d.e.f_LABS-l.m.n_CSNAME-u.v.w
```

This tag becomes the canonical provenance record for the entire ecosystem.

The tag is required **regardless of which subsystem changed**.

---

3. What Constitutes a Version Change

3.1 ARI Version Changes

Trigger MAJOR or MINOR bumps for: - governance doctrine changes
- constraint modifications
- metadata or provenance rules
- role definitions
- institutional policy updates

PATCH for clarifications only.

3.2 AWO Version Changes

Trigger MAJOR or MINOR bumps for: - workflow logic changes
- structural modifications
- metadata extraction behavior
- reasoning chain architecture
- enforcement of new method rules

PATCH for documentation-only fixes.

3.3 CRI Version Changes

Trigger MAJOR or MINOR bumps for: - deterministic engine changes
- identity/attestation logic
- integrity validation behavior
- execution environment capture rules

PATCH for non-executable changes or comments.

3.4 Labs Version Changes

Trigger for: - new demos
- engineering utilities
- non-governance infrastructure

3.5 Case Study Version Changes

Each case study is versioned independently.

---

4. Change Classes

4.1 Breaking Change

A change that: - invalidates prior assumptions
- alters subsystem guarantees
- impacts reproducibility or determinism
- modifies governance or subsystem contracts

Requires:
- MAJOR bump
- governance log entry
- IC approval
- new global release tag

4.2 Non-Breaking Change

Adds new features without violating constraints.

Requires:
- MINOR bump
- governance log entry
- new global release tag

4.3 Clarification Change

Documentation-level updates with **no impact** on logic or execution.

Requires:
- PATCH bump
- governance log entry
- new global release tag

4.4 Deprecated / Retired

Subsystems, documents, or versions may be retired but must remain accessible.

Requires:
- MAJOR or MINOR bump
- deprecation notice
- governance log entry

---

5. Approval Workflow

All version changes require:

1. **ARI Institutional Coordinator approval**
2. **Governance log entry**
3. **Provenance verification**
4. **Metadata normalization (if applicable)**
5. **Doc Guard pass**
6. **Creation of a Global Ecosystem Release Tag**

No subsystem may approve its own version changes.

---

6. Release Cycle Requirements

Before any release:

- metadata must be valid and normalized
- provenance must be verified
- deterministic execution must pass for CRI changes
- reasoning-chain auditability must pass for AWO changes
- integrity hashes must be regenerated
- changelogs must be updated
- no silent modifications permitted

Releases are not allowed without complete compliance.

---

7. Special Rules

- ARI changes force **all** subsystems to re-anchor via a global release tag.
- AWO may not trigger ARI version changes.
- CRI may not alter AWO or ARI versioning.
- Labs changes cannot modify upstream subsystem versions.

- Case Studies may version themselves independently.
- No circular dependencies permitted.

---

8. Deprecation & Retirement Policy

Artifacts may be deprecated but must remain:

- accessible
- reproducible
- traceable
- versioned

Retirement requires explicit governance documentation.

---

9. Revision Rules

All revisions require:

1. ARI IC approval
2. Governance log entry
3. Version increment
4. Backward linkage
5. Rationale included

No silent revisions permitted.
