



ManChine AI
TECHNOLOGY



RiCo

Runtime Integrity Control

RGF-01

Foundations of Runtime Governance

Defining the Core Concepts of
the Runtime Governance Discipline



HUMANITY FIRST



CONTINUITY ALWAYS



ARCHITECTURE MUST SURVIVE

A Foundational Publication of the
Runtime Governance Standards Family

VERSION 1.0 | MAY 2025

1. Why Runtime Governance?

For generations, governance has provided the structures through which organizations establish authority, define responsibilities, manage risk, and oversee the operation of critical systems. These principles remain fundamental and continue to guide the design, deployment, and oversight of technology.

As intelligent and increasingly autonomous systems become capable of operating continuously, however, governance faces a new challenge. Decisions made at design time cannot, by themselves, account for the changing conditions that arise throughout execution.

A system may remain technically functional while the authority under which it operates changes. The evidence supporting its actions may evolve. Operational conditions may shift beyond those originally anticipated. Policies may be revised, permissions may expire, and new constraints may emerge while execution continues.

These changes do not necessarily indicate a failure of governance. They reveal a limitation in governance models that primarily concentrate on planning, deployment, and periodic oversight rather than the continuous preservation of legitimate operation.

Runtime Governance is introduced to address this challenge.

It extends governance into the period of execution, addressing the governance responsibilities that arise while systems continue to operate.

The objective is not to replace existing governance frameworks, but to complement them.

As systems become increasingly autonomous and continuously active, governance must likewise become continuous.

Runtime Governance is the discipline established to meet that need.

2. What is Runtime Governance?

Runtime Governance is the discipline concerned with ensuring that systems remain legitimately governed throughout their operational lifetime.

Unlike traditional governance, which primarily establishes the conditions under which systems are designed, approved, deployed, and periodically reviewed, Runtime Governance addresses the governance responsibilities that continue while systems are executing.

It recognizes that execution is not a static event, but an ongoing process in which authority, operational conditions, evidence, policies, dependencies, and responsibilities may change over time.

The central objective of Runtime Governance is to preserve the legitimacy of system operation as those conditions evolve.

To achieve this, Runtime Governance establishes concepts for:

- maintaining legitimate authority throughout execution;
- preserving governance integrity as systems operate;
- evaluating the evidence that supports continued operation;
- sustaining governance continuity as conditions change.

Together, these concepts ensure that governance remains capable of influencing execution whenever necessary.

Runtime Governance is therefore not a product, platform, implementation, or individual standard.

It is a discipline that defines how governance extends beyond design and deployment into the continuous operation of systems.

The Runtime Governance Standards Family provides the normative specifications through which this discipline is expressed, while implementations such as RiCo demonstrate how those specifications may be realized in practice.

As intelligent systems become increasingly autonomous, adaptive, and continuously active, Runtime Governance provides the conceptual and normative foundation required to ensure that governance remains an active operational capability rather than a point-in-time activity.

3. The Foundational Concepts of Runtime Governance

Runtime Governance is founded upon a set of interdependent concepts that together explain how governance is established, exercised, demonstrated, preserved, and sustained throughout the operational lifetime of a system.

These concepts are not independent mechanisms. They operate collectively within the context of execution, where intelligent systems continue to make decisions, perform actions, and interact with changing operational conditions.

Execution is therefore not itself a foundational concept. It is the operational environment in which the concepts of Runtime Governance are continuously applied.

Together, these concepts establish the conceptual foundation upon which the Runtime Governance Standards Family is built.

3.1 Runtime Authority

Runtime Authority establishes the legitimate basis upon which a system is permitted to act throughout execution.

It defines the governance conditions under which authority is granted, maintained, constrained, delegated, suspended, or withdrawn as operational conditions evolve.

Without legitimate authority, execution may remain technically possible but no longer remain legitimately governed.

3.2 Runtime Integrity

Runtime Integrity preserves the consistency, trustworthiness, and governance state of a system throughout execution.

It ensures that governance responsibilities, authorized behavior, and operational constraints remain aligned with the conditions under which continued execution has been authorized.

Runtime Integrity concerns the preservation of legitimate governance during operation, rather than solely the prevention of technical compromise.

3.3 Runtime Evidence

Runtime Evidence provides the information necessary to demonstrate, evaluate, and justify governance decisions during execution.

It establishes the factual basis upon which governance claims may be assessed, verified, challenged, or confirmed.

Without evidence, governance cannot be meaningfully evaluated, and continued execution cannot be responsibly justified.

3.4 Runtime Continuity

Runtime Continuity preserves legitimate governance as operational conditions evolve.

It recognizes that authority, evidence, policies, dependencies, and environmental conditions may change while execution continues.

Its purpose is to ensure that governance remains effective throughout those changes without compromising the legitimacy of continued operation.

4. Relationships Among the Foundational Concepts

The foundational concepts of Runtime Governance do not operate independently. They exist as a connected governance system in which each concept contributes to the preservation of legitimate operation throughout execution.

Understanding Runtime Governance therefore requires more than defining individual concepts. It requires understanding how those concepts interact as operational conditions evolve.

Runtime Authority establishes the legitimacy of action.

Runtime Evidence provides the basis upon which governance decisions may be justified.

Runtime Integrity preserves the governed state throughout execution.

Runtime Continuity ensures that governance remains effective as conditions evolve.

The precise architectural relationships among Runtime Authority, Runtime Evidence, Runtime Integrity, and Runtime Continuity are developed through the Runtime Governance Standards Family. This publication introduces those relationships conceptually without prescribing a complete architectural model.

Several fundamental questions guide that architecture:

- How does Runtime Authority establish and constrain legitimate execution?
- What evidence is necessary to justify the continuation, modification, or termination of authority?
- How does Runtime Evidence support the preservation of Runtime Integrity?
- In what manner does Runtime Integrity contribute to Runtime Continuity?
- How do these concepts collectively enable Runtime Governance throughout execution?

These questions represent architectural relationships rather than isolated definitions. Their answers emerge through the interaction of the foundational concepts rather than through any single concept alone.

Accordingly, the Runtime Governance Standards Family should be understood as an evolving body of work that progressively formalizes these relationships through normative specifications.

This foundational publication establishes the conceptual basis from which those specifications develop.

5. Why Standards?

Foundational concepts explain *why* a discipline exists and identify the essential ideas upon which it is built.

Concepts alone, however, are not sufficient to achieve consistent implementation, interoperability, or long-term governance.

As Runtime Governance matures, its concepts must be expressed in a manner that is clear, repeatable, verifiable, and capable of being applied consistently across organizations, technologies, and implementations.

This is the purpose of standards.

Standards transform foundational concepts into common specifications that define terminology, responsibilities, governance requirements, architectural relationships, and publication practices. They establish a shared basis for understanding and enable independent implementations to operate according to the same normative expectations.

The Runtime Governance Standards Family fulfills this role.

Each publication within the Standards Family addresses a specific aspect of the discipline while remaining grounded in the foundational concepts established by Runtime Governance.

Together, these standards create a coherent, versioned, and interoperable body of knowledge that enables Runtime Governance to be understood, implemented, evaluated, and maintained consistently over time.

Standards do not replace the foundational concepts. They formalize them.

In doing so, they provide the common language and normative structure through which Runtime Governance can be applied across diverse systems while preserving architectural consistency and long-term continuity.

6. Relationship to Implementations

The Runtime Governance Standards Family establishes the normative foundation of the Runtime Governance discipline. It defines the principles, terminology, governance requirements, and architectural guidance that provide a common basis for understanding and implementing Runtime Governance.

Implementations apply these standards within operational systems.

An implementation is responsible for realizing the requirements and principles established by the standards in a manner appropriate to its intended environment. Multiple implementations may exist, each employing different architectures, technologies, or operational approaches while remaining consistent with the published standards.

RiCo (Runtime Integrity Control) is an example of such an implementation.

RiCo demonstrates how the principles and specifications of the Runtime Governance Standards Family may be realized within an operational environment. It is an implementation and reference architecture, not a replacement for the standards themselves.

The standards define *what* Runtime Governance requires.

Implementations determine *how* those requirements are realized in practice.

This distinction preserves the independence of both.

The Standards Family may continue to evolve through formally published revisions, while implementations may innovate independently, provided they remain consistent with the published standards upon which they are based.

By maintaining this separation, the Runtime Governance discipline remains implementation-independent, encourages interoperability, and enables multiple conforming implementations to emerge over time.

Conclusion

Runtime Governance introduces a discipline for governing systems throughout execution as operational conditions evolve.

This publication establishes the conceptual foundation of that discipline by defining its purpose, introducing its foundational concepts, and explaining the role of standards and implementations within a coherent architectural framework.

The Runtime Governance Standards Family builds upon these foundations through normative specifications that formalize the discipline, while independent implementations demonstrate how those specifications may be realized in practice.

As increasingly autonomous systems become integral to society, Runtime Governance provides a foundation upon which governance can remain continuous, trustworthy, and capable of adapting to change.

The discipline begins with concepts.

Its long-term success depends upon their consistent realization through standards, implementations, and responsible stewardship.