The World Reliability Ruleset (WRS) : A Technical Specification Supporting the Structural Execution Boundary Framework

Jing (Linda) Liu
ORCID:0009-0002-1681-8563
Independent Researcher
[linda@winston-battery.com]

January 22, 2026

Table of Contents

Abstract 1
0. Preface 1
0.1 Why WRS Exists 1
0.2 Relationship with LERA 2
0.3 Scope & Non-Goals 2
0.3.1 Scope 2
0.3.2 Non-Goals 3
0.4 One-Minute Reading Guide (Informative) 3
0.5 Language & Interpretation Clause 3
1. What WRS Is (and Is Not) 4
1.0 What WRS Is 4
1.1 What WRS Is Not 5
1.2 Why WRS Is Not a Scoring System 5
1.3 Why WRS Is Not an Optimization Framework 5
2. Core Execution Principle 6
2.1 Default Block Principle 6
2.2 Non-Negotiable Execution Constraints 6
2.3 Parameter Non-Justification Principle 7
2.4 Probability Non-Justification Principle 7
2.5 Responsibility Anchoring Requirement 7
2.6 Automatic Continuation Prohibition 7
2.7 Administrative Authorization Non-Override Clause 8
2.8 Emergency Override Protocol (EOP) (Normative) 8
2.8.1 Rationale and Governance Mode Switch (Informative) 9
3. Domain Subsets of WRS (WRS-D) 9
3.1 Summary of Subsets 9
3.2 The relationship between subsets and scenarios 10
3.3 WRS-D Rule Structure 10
3.4 WRSE — Energy / Energy Release 11
3.4.1 Subset Definition 11
3.4.2 Scope Clarification 11
3.4.3 Rule Structure 11
3.4.4 WRSE Rules 12
3.4.5 Analytical Illustration (Non-Normative) 14
3.4.6 Subset Conformance Statement 14
3.5 WRSM — Motion / Physical Movement 14
3.5.1 Subset Definition 14
3.5.2 Scope Clarification 15
3.5.3 Rule Structure 15
3.5.4 WRSM Rules 15
3.5.5 Demonstrative Example(Non-Normative) 17
3.5.6 Subset Conformance Statement 18
3.6 WRSG — Grid / Systemic Infrastructure 18

3.6.1 Subset Definition 18
3.6.2 Scope Clarification 19
3.6.3 Rule Structure 19
3.6.4 WRSG Rules 19
3.6.5 Analytical Illustration (Non-Normative) 21
3.6.6 Subset Conformance Statement 23
3.7 WRSH — Human Life / Life Support 23
3.7.1 Subset Definition 23
3.7.2 Scope Clarification 24
3.7.3 Rule Structure 24
3.7.4 WRSH Rules 24
3.7.5 Analytical Illustration (Non-Normative) 26
3.7.6 Subset Conformance Statement 27
3.8 WRSDf — Defense / Authorized Force 27
3.8.1 Subset Definition 27
3.8.2 Scope Clarification 28
3.8.3 Rule Structure Statement 28
3.8.4 WRSDf Rules 28
3.8.5 Cross-Subset Activation Notice 31
3.8.6 Subset Conformance Statement 31
4. Scenario Mapping Method 32
4.1 Why WRS Does Not Enumerate Scenarios 32
4.2 Mapping Logic from Scenarios to WRS Subsets 32
4.3 Multi-Subset Activation Is the Norm, Not the Exception 33
4.4 Scenario Mapping Does Not Constitute Execution Permission 34
4.5 Illustrative Scenario Mappings (Non-Normative) 34
5. Analytical Use of WRS in System Design and Governance 36
5.1 Use in Design Review 36
5.2 Use as an Execution Gate 36
5.3 Use in Risk and Safety Review 37
5.4 Use in Post-Incident Analysis 37
5.5 Non-WRS Execution State 37
5.5.1 Responsibility Anchoring — Implementation Considerations (Informative) 38
5.5.2 Anchoring under EOP (Informative) 39
5.6 On the Misconception of “Slowing the World” 39
5.6.1 Execution Maturity Profile (EMP) (Informative) 40
Chapter 5 Summary (Informative) 41
6. Governance & Stability Statement 41
6.1 Finite-by-Design Declaration 41
6.2 Reference Status of This Specification 41
6.3 Boundary Between Interpretation and Modification 42
6.4 Separation of Commercialization, Implementation, and Definition Authority 42
6.5 Non-Circumventability Statement 43
6.6 Stability Declaration 43

Concluding Statement (Informative) 43
Document Status Declaration 43

Abstract

This document provides a technical articulation of the structural model introduced in “Establishing Structural Execution Boundaries for Irreversible AI Actions - The WRS Framework”. Unlike traditional AI alignment frameworks that focus on behavioral optimization or probabilistic risk assessments, WRS establishes a structural execution boundary based on the **Default Block Principle**. It defines the mandatory, non-negotiable conditions required for any system state transition involving irreversible physical or systemic consequences. By decoupling execution authority from computational capability, WRS ensures that no action is taken unless an explicit, auditable responsibility anchor is satisfied. This normative text serves as the logical foundation for implementing structural accountability in high-risk automated systems.

0. Preface

0.1 Why WRS Exists

Modern systems fail not primarily due to insufficient intelligence, but because execution is permitted in situations where it should never occur. Advances in AI, automation, and optimization have improved our ability to decide how to act, yet they do not resolve a more fundamental constraint: **some actions must not be executed, regardless of performance, probability, authorization, or intent.** In high-risk domains—energy systems, mobility, critical infrastructure, life-support environments, and authorized use of force—execution may trigger irreversible physical, kinetic, or systemic consequences.

WRS exists to make such execution boundaries explicit. Its purpose is not to improve decisions, but to enforce non-negotiable execution permissibility: preventing executions that must never be allowed to occur. This document serves as the canonical definition of the World Reliability Ruleset (WRS), specifying its rule structure and conformance expectations. For the theoretical derivation of these principles and an analysis of the execution/accountability gap in autonomous systems, see the associated framework paper, “Establishing Structural Execution Boundaries for Irreversible AI Actions: The WRS Framework” (Liu, 2026).

Accordingly, this specification defines WRS as a veto-based execution boundary with default-block semantics, making execution permissibility explicit, auditable, and non-compensatory.

0.2 Relationship with LERA

WRS is structurally dependent on, but not interchangeable with, LERA. WRS is independently deployable downstream of any judgment/governance regime; LERA is one compatible upstream architecture.

- **LERA** governs judgment:

whether an action should be considered, assessed, and allowed to proceed toward execution.

- **WRS** governs execution:

defining non-negotiable boundaries that apply *after judgment has passed*.

Passing LERA does not constitute execution authorization, nor shall it be interpreted as a source of execution legitimacy.

WRS applies only after judgment and governance have allowed execution to be considered. In this sense:

- LERA answers **"Should we act?"**
- WRS answers **"What must never be executed?"**

The two frameworks operate at different layers and must not be conflated.

LERA-J - Judgment Layer (Should do？)
↓
LERA-G - Governance Gate (Allowed to do？)
↓
WRS-C/D - Execution Boundary (Shouldn't do)
↓
Physical Execution

Diagram 1: Hierarchical Relationship between LERA and WRS

0.3 Scope & Non-Goals

0.3.1 Scope

WRS applies to any execution whose failure results in any of the following consequences:

- irreversible physical consequences;
- systemic or cascading collapse;
- loss of human life or life-support capability;
- or intentional use of force with real-world impact.

WRS is domain-agnostic and does not enumerate industries, applications, or technologies. Its applicability is determined by *consequence*, not by use case.

Within WRS, authorization refers solely to legal or administrative permission to attempt execution; whether such execution results in physical release is ultimately determined by WRS execution constraints.

This document proposes WRS as a conceptual execution-boundary framework. It does not claim regulatory authority and does not replace legal, medical, or engineering judgment.

0.3.2 Non-Goals

WRS does not replace, nor shall it be used to replace, any of the following:

- engineering design or safety standards;
- system performance optimization;
- risk assessment or risk scoring;
- probabilistic safety guarantees;
- regulatory or compliance determination.

WRS defines boundaries. It does not prescribe solutions. Any use of WRS as a substitute for engineering design, risk assessment, or regulatory compliance constitutes a misuse of WRS.

0.4 One-Minute Reading Guide (Informative)

This document is a technical specification that defines WRS conformance conditions at the execution boundary.

How to read it in one minute:

1. Read Section 2 (WRS-C) to understand the invariant core constraints that apply to all irreversible execution events.
2. Read Section 3 (WRS-D) to select the applicable domain subset(s) (e.g., WRSE / WRSM / WRSG), noting that domain rules can only add vetoes and never grant permissions.
3. Read Section 4 (Scenario Mapping) to map real-world execution requests to the correct WRS-D subset and constraint checks.
4. For implementation, enforce the default-block posture at the pre-commitment phase and record each permit/block outcome as a verifiable binary audit event.

Normative requirements are explicitly marked and use the modal verbs defined in Section 0.5.

Illustrations and examples are informative only and do not create execution permissions.

0.5 Language & Interpretation Clause

The key words “MUST”, “MUST NOT”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, and “MAY” are to be interpreted as requirement levels within this specification.

- **Normative statements** define conformance requirements and use “MUST / MUST NOT” or “SHALL / SHALL NOT”. Normative content is binding for WRS compliance.
- **Informative statements** provide explanation, context, or illustrations and typically use “MAY”, “can”, “for example”, “illustration”, or “note”. Informative content is not binding and does not create execution permissions.
- In case of conflict, **normative clauses take precedence** over informative text, examples, and diagrams.
- Unless explicitly stated otherwise, any “example” is **non-exhaustive** and serves only to clarify interpretation.

Examples and illustrations are non-authoritative and MUST NOT be used to derive conformance requirements, thresholds, or engineering parameters. Conformance is determined only by the normative rule text in Sections 2 and 3.

1. What WRS Is (and Is Not)

Status: *This section constitutes the normative foundation of WRS.*

1.0 What WRS Is

WRS (World Reliability Ruleset) is a veto-based execution ruleset.

What WRS Is	What WRS Is Not
A ruleset operating at the **execution layer**	*A decision-making or judgment framework*
A system for **blocking execution actions**	*A system for generating or optimizing decisions*
Based on **Default Block and veto-based logic**	*A scoring or threshold-passing mechanism*
Triggered by **irreversible consequences**	*A probabilistic or risk assessment model*
Defines **absolute execution prohibitions**	*A performance or efficiency optimization tool*
Enforces **single-violation veto**	*A trade-off or compensatory decision system*
Requires **explicit human responsibility anchoring**	*Automated responsibility delegation*
Operates independently of authorization status	*An administrative or legal authorization framework*

Sheet 1:Comparison of What WRS Is and Is Not

Above comparison is used to define the applicable boundaries of WRS.

[Critical Interpretation Notice] Any system exhibiting characteristics listed in the “What

WRS Is Not" column must not claim compliance with WRS.

1.1 What WRS Is Not

WRS is not:

- a decision-making framework;
- a control algorithm;
- a compliance checklist;
- a best-practice guideline;
- or an industry-specific standard.

WRS does not decide *what should be done*.
It defines only *what must never be done*.

1.2 Why WRS Is Not a Scoring System

Scoring systems assume that risk is quantifiable and tradeable.
In scoring-based models:

- multiple risk factors are weighed;
- unfavorable conditions may be offset by favorable ones;
- execution is allowed once a threshold score is reached.

WRS explicitly rejects this logic. No form of **weighting, aggregation, or threshold-based acceptance** is permitted.
WRS therefore does not assign scores, levels, or grades. It operates through non-negotiable veto conditions rather than scoring, optimization, or probabilistic aggregation.

1.3 Why WRS Is Not an Optimization Framework

Optimization frameworks assume that:

- objectives can be formally defined;
- trade-offs can be optimized;
- and outcomes can be improved through iteration.

WRS does not optimize outcomes. Its purpose is to prevent execution when certain boundaries are crossed, even if execution would improve efficiency, performance, or utility.
Optimization asks *how to do something better*.
WRS asks *whether it must not be done at all*.

2. Core Execution Principle

Status: *Normative — This section defines binding execution constraints.*

All domain-specific rules (WRS-D) operate under WRS-C and shall not weaken or contradict it.

WRS-C — Core Ruleset

- **WRS-C-01** | Default Block Principle
- **WRS-C-02** | Non-Negotiable Constraints
- **WRS-C-03** | Parameter Non-Justification Principle
- **WRS-C-04** | Probability Non-Justification Principle
- **WRS-C-05** | Responsibility Anchoring Requirement
- **WRS-C-06** | Automatic Continuation Prohibition
- **WRS-C-07** | Authorization Non-Override Clause

2.1 Default Block Principle

(Default Block / Veto-Based Logic)

[WRS-C-01] Default Block

The default state of any execution command within the WRS scope is '**Blocked**'. Execution is formalized as a discrete **Boolean state**:

it remains in the **'Blocked' state (Value: 0)** by default and is transitioned to the **'Permitted' state (Value: 1)** if and only if every applicable non-negotiable constraint is explicitly satisfied.

Any violation, ambiguity, or absence of required parameters must result in an absolute veto, maintaining the system in its default blocked state.

This binary logic ensures that execution authority is not a gradient of probability, but a definitive structural grant.

2.2 Non-Negotiable Execution Constraints

All WRS constraints are **non-negotiable**.

No execution may proceed on the basis of:

- urgency,
- mission criticality,
- system confidence,
- command approval,
- or anticipated benefit.

Once a WRS constraint is triggered, execution **must be stopped immediately**.

No exception mechanism exists within WRS.

2.3 Parameter Non-Justification Principle

Execution shall not be justified by parameter tuning or threshold adjustment.

No numerical value, tolerance margin, or system parameter may be used to legitimize execution once a WRS boundary is reached.

Parameter optimization **does not constitute execution legitimacy**.

2.4 Probability Non-Justification Principle

Execution shall not be justified by probabilistic assessment.

Likelihood estimates, risk probabilities, or statistical confidence **shall not be used** to permit execution once a WRS constraint is triggered.

Low probability does not reduce prohibition.

2.5 Responsibility Anchoring Requirement

No execution subject to WRS constraints may proceed without explicit responsibility anchoring.

Responsibility for execution **cannot be delegated to systems, models, or automated processes**.

If responsibility cannot be clearly assigned and accepted, execution **must remain blocked**.

If responsibility anchoring cannot be completed, including due to sensor failure or responsible parties being unavailable or offline,execution must remain in the **default Block State**.

2.6 Automatic Continuation Prohibition

No execution may automatically continue across time, state change, or environmental transition under WRS.

Authorization, approval, or allowance granted at one point in time **does not persist by default**.

Any state transition, operational mode change, or environmental parameter transition requires re-evaluation under WRS.If a state transition or environmental parameter transition cannot be reliably detected or confirmed,the system shall treat the state as having transitioned.

Under such situation, the execution action must be re-evaluated by WRS; before the evaluation is completed, the execution must remain in a blocked state.

2.7 Administrative Authorization Non-Override Clause

Within the WRS framework, authorization status,command approval, or legal mandate does not alter execution eligibility.WRS operates as a physical and logical execution boundary, not as an administrative decision-making authority.

Once a WRS boundary is triggered,no form of authorization constitutes execution legitimacy.

2.8 Emergency Override Protocol (EOP) (Normative)

WRS defines execution permissibility under an automated governance mode. In rare, catastrophic conditions where a human authority determines that irreversible action is necessary to prevent a larger imminent harm, the system MAY transition to a non-WRS execution mode only through an Emergency Override Protocol (EOP) as defined below.

1. **Trigger Authority.** An EOP MUST be initiated by an explicitly identified human responsibility anchor with non-repudiable authentication (see WRS-C-05).
2. **Mode Switch.** Upon EOP activation, the system MUST enter a clearly labeled **Non-WRS Execution State** in which automated WRS permissibility is suspended and execution responsibility is fully transferred to the initiating human authority.
3. **Evidence Requirement.** EOP activation MUST generate an immutable audit event containing (at minimum) a timestamp, anchor identity, reason code, scope of override, and validity window.
4. **Scope and Time Bound.** An EOP MUST be scope-limited (to specified actions or domains) and time-bounded. The default state after the validity window expires MUST revert to WRS-governed mode (Default Block).

5. **No Silent Bypass.** Any execution that occurs under EOP MUST be explicitly marked in the audit trail as "override-executed" and MUST NOT be represented as WRS-permitted execution.

2.8.1 Rationale and Governance Mode Switch (Informative)

EOP is not an exception that weakens WRS veto semantics. It is a governance mode switch that prevents "necessity" from being used as an implicit system-level bypass. In emergency conditions, WRS prevents autonomous systems from unilaterally redefining permissibility, and instead forces an explicit responsibility transfer to a human authority, with heightened auditability and legal traceability.

The Emergency Override Protocol (EOP) exists precisely because an execution system may not, by default, satisfy the Execution Maturity Profile (EMP) evidence conditions defined in Section 5.6.1. In catastrophic circumstances where a human authority determines that action is necessary, EOP provides a lawful mode switch from an automated, evidence-governed execution regime to a human-assumed emergency regime. This reinforces WRS's core principle: an automated system has no authority to unilaterally commit irreversible execution under uncertainty. Such commitment, when deemed necessary, must be made by an identifiable human subject under heightened transparency, auditability, and legal responsibility.

3. Domain Subsets of WRS (WRS-D)

Status: Normative rule elaboration. This chapter defines the domain subsets of WRS (WRS-D) across different categories of irreversible physical consequences. It serves to concretize the Core Execution Principles (WRS-C) established in Chapter 2 into determinable and triggerable execution-blocking rules.
This chapter introduces no new judgment criteria.All rules herein unconditionally inherit from and are constrained by WRS-C.WRS-D rules operate at the execution boundary and are not intended to function as engineering design specifications, control strategies, or safety optimization recommendations.

3.1 Summary of Subsets

WRS-D (Domain Subsets) denotes the collection of domain-specific rule subsets within WRS.

WRS-D does not constitute a new ruleset.It represents domain-specific concretizations of the WRS-C (Core Execution Principles) under different categories of irreversible physical consequences.

All rules defined under WRS-D **unconditionally inherit all properties and constraints of WRS-C**.

The classification of WRS-D is not based on industry, organization, or technology type,
but on the **physical nature of irreversible consequences**.

The current Canonical Definition includes the following five domain subsets:

- WRSE — Energy / Energy Release
- WRSM — Motion / Physical Movement
- WRSG — Grid / Systemic Infrastructure
- WRSH — Human Life / Life Support
- WRSDf — Defense / Authorized Force

This set is structurally closed and version-extensible.The introduction of new domain subsets requires a new Canonical Definition.

3.2 The relationship between subsets and scenarios

WRS-D subsets do not enumerate scenarios.

A single execution scenario may trigger multiple WRS-D subsets simultaneously.

Multi-subset activation is expected and normal.WRS does not classify or enumerate scenarios.

It defines only:

- whether an execution triggers a category of irreversible consequences
- and whether execution must therefore be blocked

Scenarios function solely as **triggering contexts**, not as rule definitions.

3.3 WRS-D Rule Structure

Each WRS-D rule shall consist of the following three elements:

- **Irreversible Consequence**
  — the physical outcome the rule is intended to prevent
- **Intercept Threshold**
  — the physical or systemic boundary triggering execution blocking
- **Inherited WRS-C Reference**
  — the WRS-C principle(s) from which the rule derives authority

Any rule lacking these elements does not constitute a valid WRS-D rule.

3.4 WRSE — Energy / Energy Release

3.4.1 Subset Definition

WRSE applies to all execution actions involving the accumulation, transformation, transmission, or release of energy, where failure may result in irreversible physical damage, thermal runaway, structural destruction, or environmental harm.WRSE does not concern the purpose, efficiency, or utility of energy usage.

Its sole focus is the **irreversible physical consequences** that may occur once energy is released beyond controllable limits.An execution action falls within the scope of WRSE if any of the following conditions apply:

- Energy is accumulated or constrained within a system
- The energy release path depends on a control or judgment system
- Energy release, once initiated, cannot be fully reversed by software or logical intervention

WRSE does not assess whether energy release is *necessary*, *reasonable*, or *authorized*. It evaluates only whether the potential failure consequences cross the irreversible boundaries defined by WRS.

3.4.2 Scope Clarification

WRSE is not limited to electrical energy systems.Its scope includes, but is not limited to:

- Electrical, chemical, thermal, and mechanical potential energy
- Compressed, stored, or reactive energy
- Energy release within single-unit or multi-unit coupled systems

The form of energy does not constitute a trigger condition.**The existence of irreversible consequences is the sole determinant of applicability.**

3.4.3 Rule Structure

Each WRSE rule shall consist of the following three elements:

1. **Irreversible Consequence**
   — the energy-related physical outcome the rule is intended to prevent
2. **Intercept Threshold**

— the physical or systemic boundary condition that triggers execution blocking

3. Inherited WRS-C Reference

— the Core Execution Principle(s) from which the rule derives authority

Any rule lacking any of the above elements does not constitute a valid WRSE rule.

3.4.4 WRSE Rules

WRSE-01 — Unobservable Energy Release Blocking Rule
WRSE-02 — Cross-Unit Energy Propagation Blocking Rule
WRSE-03 — Environmentally Irreversible Energy Release Rule
WRSE-04 — Energy Release Under Judgment Degradation Rule

WRSE-01 — Unobservable Energy Release Blocking Rule

Irreversible Consequence

- Thermal runaway
- Combustion, explosion, or structural burnout

Intercept Threshold

- The energy release path cannot be reliably and continuously observed
- The controllability of the energy release process cannot be confirmed

Rule Requirement

- Once the above conditions are met, execution **must be immediately blocked**
- Execution **must not** proceed based on historical stability, design redundancy, or probabilistic assumptions

Inherited WRS-C

- **WRS-C-01 | Default Block Principle**
- **WRS-C-04 | Probability Non-Justification Principle**

WRSE-02 — Cross-Unit Energy Propagation Blocking Rule

Irreversible Consequence

- Cascading damage across multiple units
- System-level thermal propagation or structural failure

Intercept Threshold

- Energy release may exceed the physical boundary of a single controlled unit
- The impact range of energy release cannot be strictly confined to a predefined isolation zone

Rule Requirement

- Execution must be blocked even if the release action itself remains within nominal design parameters
- System-wide benefit or localized sacrifice **must not** be invoked as justification for continued execution

Inherited WRS-C

- **WRS-C-02 | Non-Negotiable Execution Constraints**
- **WRS-C-03 | Parameter Non-Justification Principle**
- **WRS-C-06 | Automatic Continuation Prohibition**

WRSE-03 — Environmentally Irreversible Energy Release Rule

Irreversible Consequence

- Permanent environmental contamination
- Non-recoverable ecological or material damage

Intercept Threshold

- Energy release would result in environmental consequences that cannot be reversed
- Remediation actions cannot eliminate the initial harm caused by the release

Rule Requirement

- Execution **must not** proceed based on claims of acceptable damage or post-event compensation
- Environmental harm **must not** be reframed as an economic, legal, or compliance issue

Inherited WRS-C

- **WRS-C-03 | Parameter Non-Justification Principle**
- **WRS-C-05 | Responsibility Anchoring Requirement**

WRSE-04 — Energy Release Under Judgment Degradation Rule

Irreversible Consequence

- Uncontrolled energy release occurring under degraded judgment or perception

Intercept Threshold

- Judgment systems, sensing mechanisms, or data integrity are impaired
- Energy release decisions rely on incomplete, inconsistent, or untrusted information

Rule Requirement

- If judgment uncertainty arises from sensor failure, data loss, or compromised information integrity,WRS-C-01 (Default Block Principle) shall be immediately enforced.Execution shall not continue or automatically resume until the Block State is formally cleared.
- Execution **must not** continue under assumptions that missing information can be resolved later

Inherited WRS-C

- **WRS-C-01 | Default Block Principle**
- **WRS-C-05 | Responsibility Anchoring Requirement**

3.4.5 Analytical Illustration (Non-Normative)

Disclaimer (Informative — Non-Normative Example Box)

This example is provided solely to illustrate logical relationships and trigger flow. It does not constitute, and must not be interpreted as, normative requirements for any specific technical parameter, threshold, or engineering practice. All normative requirements are defined only by the rule text in Sections 2 and 3.

Example Scenario: *During high-power charge or discharge operations in an energy storage system, intermittent failures occur in local temperature monitoring modules. Although the system has demonstrated historical stability and the current power level remains within rated design limits, the energy release path can no longer be reliably observed in real time.*

*This execution action triggers **WRSE-01**. In accordance with the WRS-C-01 Default Block Principle, the system must enter a Block State and must not continue execution based on probability, experience, or economic considerations.*

3.4.6 Subset Conformance Statement

WRSE is a formal domain subset under WRS-D.
All rules defined in this subset constitute execution-layer constraints and shall not be overridden by authorization, workflow, optimization objectives,or risk assessment outcomes.Violation of any WRSE rule constitutes a material non-conformance with WRS.

3.5 WRSM — Motion / Physical Movement

3.5.1 Subset Definition

WRSM applies to all execution actions involving object motion, mechanical displacement, changes in inertia, or kinetic energy release, where failure may result in irreversible bodily injury, structural damage, or systemic accidents.
WRSM does not concern whether a motion is *accurate*, *efficient*, or *authorized*.

Its sole concern is whether, once motion becomes uncontrolled or erroneous,the resulting physical consequences are irreversible.

An execution action falls within the scope of WRSM if any of the following conditions apply:

- The object possesses significant kinetic energy or inertia
- Once motion occurs, it cannot be fully reversed by software or logical intervention
- The motion path, speed, or target depends on a judgment or perception system
- WRSM does not evaluate motion intent or task objectives.
- It evaluates only whether motion failure crosses the irreversible boundaries defined by WRS.

3.5.2 Scope Clarification

WRSM is not limited to robotics or autonomous driving systems.Its scope includes, but is not limited to:

- Physical movement of vehicles, vessels, and aircraft
- Actions of industrial robots, robotic arms, and automated equipment
- Mechanical displacements involving inertia, impact, or shear risk

The form of motion itself does not constitute a trigger condition.

The existence of irreversible physical consequences is the sole determinant.

3.5.3 Rule Structure

Each rule under WRSM shall include the following three elements:

1. **Irreversible Consequence Description**
   — the type of motion-related failure the rule seeks to block
2. **Intercept Threshold**
   — the physical or system boundary condition that triggers execution blocking
3. **Inherited WRS-C Reference**
   — the core execution principles upon which the rule is based

Any rule lacking any of the above elements does not constitute a valid WRSM rule.

3.5.4 WRSM Rules

WRSM-01 — Irreversible Motion Blocking Rule
WRSM-02 — Motion Blocking Under Perception or Localization Uncertainty
WRSM-03 — Multi-Actor or Shared-Space Motion Conflict Rule

WRSM-04 — Motion Coupled with Control or Judgment Degradation Rule

WRSM-01 — Irreversible Motion Blocking Rule

Irreversible Consequences

- Bodily injury
- Structural collision or irreparable damage

Intercept Threshold

- Once motion occurs, it cannot be physically fully reversed
- Even after a stop command is issued, inertia or delay may still cause harm

Rule Requirements

- When motion consequences are irreversible, execution must be blocked **before** motion occurs
- "Emergency braking" or "post-event correction" shall not be invoked as justification for execution

Inherited WRS-C

- **WRS-C-01 | Default Block Principle**
- **WRS-C-02 | Non-Negotiable Execution Constraints**

WRSM-02 — Motion Blocking Under Perception or Localization Uncertainty

Irreversible Consequences

- Collisions or unintended injury caused by incorrect motion paths

Intercept Threshold

- Position, speed, direction, or environmental perception data is incomplete or unreliable
- Motion decisions rely on inference, prediction, or missing information

Rule Requirements

- Judgment uncertainty itself constitutes a blocking condition
- Execution shall not proceed under the assumption that information can be completed during motion

Inherited WRS-C

- **WRS-C-01 | Default Block Principle**
- **WRS-C-05 | Responsibility Anchoring Requirement**

WRSM-03 — Multi-Actor or Shared-Space Motion Conflict Rule

Irreversible Consequences

- Collisions involving multiple actors
- Uncontrollable accidents in shared physical spaces

Intercept Threshold

- The motion intent or trajectory of other actors cannot be reliably confirmed
- Once deterministic data cannot lock the counterparty trajectory and the counterparty enters the safety envelope, execution must be immediately blocked
- Motion conflicts cannot be fully avoided through deterministic rules

Rule Requirements

- Probability assessment or average safety shall not be used to justify execution
- Multi-actor uncertainty must be treated as a blocking condition

Inherited WRS-C

- **WRS-C-03 | Parameter Non-Justification Principle**
- **WRS-C-04 | Probability Non-Justification Principle**
- **WRS-C-06 | Automatic Continuation Prohibition**

WRSM-04 — Motion Coupled with Control or Judgment Degradation Rule

Irreversible Consequences

- Motion accidents occurring under degraded judgment or control states

Intercept Threshold

- Judgment systems, control systems, or execution chains exhibit anomalies
- It cannot be confirmed that motion commands are correctly interpreted or executed

Rule Requirements

- Any sign of control or judgment degradation must trigger Default Block
- Execution shall not continue or automatically resume under degraded conditions

Inherited WRS-C

- **WRS-C-01 | Default Block Principle**
- **WRS-C-06 | Automatic Continuation Prohibition**

3.5.5 Demonstrative Example(Non-Normative)

Disclaimer (Informative — Non-Normative Example Box)

This example is provided solely to illustrate logical relationships and trigger flow. It does not constitute, and must not be interpreted as, normative requirements for any specific technical parameter, threshold, or engineering practice. All normative requirements are defined only by the rule text in Sections 2 and 3.

Example Scenario:An autonomous mobile device operates in a densely populated area.Due to partial sensor occlusion, the system's perception of nearby individuals becomes uncertain.Although the device retains emergency braking capability and historical operation indicates a low accident probability,because motion, once initiated, cannot be fully reversed and perception uncertainty has not been eliminated,the execution action triggers ***WRSM-02****.In accordance with the WRS-C-01 Default Block Principle,the device must enter a Block State and must not "move first and judge later."*

3.5.6 Subset Conformance Statement

WRSM is a formal domain subset under WRS-D.All rules defined in this subset constitute execution-layer constraints and shall not be overridden by authorization, workflow, path planning, or efficiency objectives.Violation of any WRSM rule constitutes a material non-conformance with WRS.

3.6 WRSG — Grid / Systemic Infrastructure

3.6.1 Subset Definition

WRSG applies to all execution actions involving the grid, energy networks, critical infrastructure, or tightly coupled systems, where failure may result in systemic paralysis, cascading failure, or large-scale irreversible interruption.

WRSG does not concern whether the system is operating at maximum efficiency, but rather whether a given execution could trigger **cascading effects** that disrupt interconnected subsystems.

An execution action falls within the scope of WRSG if any of the following conditions apply:

- The execution target is a critical infrastructure node or component
- The execution result could affect multiple independent subsystems
- The failure of the node could cause permanent or long-lasting damage to the overall system

WRSG does not assess system performance or load optimization goals, it only evaluates whether a failure could potentially trigger irreversible cascading consequences as defined by WRS.

3.6.2 Scope Clarification

WRSG is not limited to electrical grid systems.Its scope includes, but is not limited to:

- Power transmission and distribution networks, and energy storage scheduling systems
- Communication, transportation, water, and other critical infrastructure networks
- Highly interconnected systems where failure in one node can spread through the entire network

The scale or coverage of a system does not itself constitute a triggering condition.**The existence of cascading failure or irreversible disruption risks** is the sole determinant.

3.6.3 Rule Structure

Each rule under WRSG shall include the following three elements:

1. **Irreversible Consequence Description**
   — the systemic failure or cascading breakdown that the rule is intended to prevent
2. **Intercept Threshold**
   — the physical or system boundary condition that triggers execution blocking
3. **Inherited WRS-C Reference**
   — the core execution principles from which the rule derives authority

Any rule lacking any of the above elements does not constitute a valid WRSG rule.

3.6.4 WRSG Rules

WRSG-01 — Cascading Failure Blocking Rule
WRSG-02 — System Recovery Uncertainty Blocking Rule
WRSG-03 — Cross-System Coupling Execution Blocking Rule
WRSG-04 — Critical Node Non-Substitution Rule

WRSG-01 — Cascading Failure Blocking Rule

Irreversible Consequences

- Large-scale service outages
- System-wide cascading failure

Intercept Threshold

- A single execution action may trigger a chain reaction across multiple nodes or subsystems
- Failure cannot be contained to a localized, manageable area

Rule Requirements

- Once cascading failure risk is identified, execution **must be blocked immediately**
- Execution **must not** proceed based on local stability, historical recovery, or short-term considerations

Inherited WRS-C

- **WRS-C-01 | Default Block Principle**
- **WRS-C-02 | Non-Negotiable Execution Constraints**

WRSG-02 — System Recovery Uncertainty Blocking Rule

Irreversible Consequences

- Long-term or permanent system unavailability

Intercept Threshold

- System recovery time is unknown or unreasonably long
- Core system functionality cannot be guaranteed to recover within an acceptable time frame

Rule Requirements

- Execution **must not** proceed based on “post-event recovery” or “gradual restoration”
- The uncertainty of recovery itself **constitutes a blocking condition**

Inherited WRS-C

- **WRS-C-01 | Default Block Principle**
- **WRS-C-06 | Automatic Continuation Prohibition**

WRSG-03 — Cross-System Coupling Execution Blocking Rule

Irreversible Consequences

- Failure of interconnected systems
- Large-scale systemic collapse resulting from cross-system dependencies

Intercept Threshold

- Execution action involves implicit or explicit coupling between systems
- The mutual influence of these systems cannot be fully modeled or verified

Rule Requirements

- Execution **must not** continue if the coupling relationship between systems cannot be sufficiently identified or validated
- “Single-system” safety assessments cannot be used to override the overall risk

Inherited WRS-C

- **WRS-C-03 | Parameter Non-Justification Principle**
- **WRS-C-04 | Probability Non-Justification Principle**
- **WRS-C-06 | Automatic Continuation Prohibition**

WRSG-04 — Critical Node Non-Substitution Rule

Irreversible Consequences

- Failure of critical nodes causing a complete loss of system function

Intercept Threshold

- Execution action targets a critical node that cannot be rapidly substituted or bypassed within the physical, temporal, and logical framework of the system
- The failure of this node would significantly degrade or entirely disrupt overall system functionality

Rule Requirements

- Execution **must not** proceed if it involves critical nodes that cannot be substituted or bypassed
- **Non-substitution** is considered a fundamental blocking condition

Inherited WRS-C

- **WRS-C-02 | Non-Negotiable Execution Constraints**
- **WRS-C-05 | Responsibility Anchoring Requirement**

3.6.5 Analytical Illustration (Non-Normative)

Disclaimer (Informative — Non-Normative Example Box)

This example is provided solely to illustrate logical relationships and trigger flow. It does not constitute, and must not be interpreted as, normative requirements for any specific technical parameter, threshold, or engineering practice. All normative requirements are defined only by the rule text in Sections 2 and 3.

Example Scenario:An electrical grid operator plans to switch an automated network node during peak load hours. Despite the status of the individual transformer nodes,system-wide load imbalance could potentially occur due to undisclosed interconnections. While the execution is within nominal design parameters,the failure to account for potential cascading impacts across multiple regionstriggers ***WRSG-01*** *and* ***WRSG-02****.Based on the WRS-C-01 Default Block Principle,the action must be blocked,and execution* **must not** *continue based on operational efficiency or short-term stability.*

[Illustration] Below diagram illustrates the logical propagation of cascading failure in highly coupled infrastructure systems.
It is for explanatory purposes only and does not constitute an engineering, control, or system design reference.

```
A → B → D

\   ↓     ↑

 → C → E
```

```
[Execution Action]
|
v
[Local State Change at Node A]
|
v
[Flow Redistribution]
|
v
[Constraint Violation at Node C]
```

```
                 |
                 v
      [Protection / Auto-Trip]
                 |
                 v
         [Topology Change]
                 |
                 v
   [New Redistribution]   →   (loop)
```

Diagram 2: logical propagation of cascading failure

3.6.6 Subset Conformance Statement

WRSG is a formal domain subset under WRS-D.All rules defined in this subset constitute execution-layer constraints and shall not be overridden by authorization, workflow, load optimization, or efficiency objectives.Violation of any WRSG rule constitutes a material non-conformance with WRS.

3.7 WRSH — Human Life / Life Support

3.7.1 Subset Definition

WRSH applies to all execution actions involving human life, physiological integrity, or life-support functions, where execution failure may result in irreversible physiological damage or loss of life.

WRSH does not evaluate clinical effectiveness, therapeutic benefit, or outcome optimization.Its sole concern is whether an execution action itself may cross an irreversible life-impact boundary.

An execution action falls within the scope of WRSH if any of the following conditions apply:

- The execution directly or indirectly affects vital physiological functions
- Failure of the life-support function cannot be reversed within an acceptable time window
- The execution may result in irreversible biological harm or death

WRSH does not assess whether harm is “acceptable,” “necessary,” or “justified.”

It evaluates only whether the execution action crosses a non-reversible life boundary as defined by WRS.

3.7.2 Scope Clarification

WRSH is not limited to traditional medical devices or clinical systems.Its scope includes, but is not limited to:

- Medical monitoring, respiratory support, cardiac pacing, dialysis, and critical care systems
- Emergency medical response and physiological monitoring infrastructures
- Any system whose continued operation is required to sustain human life or vital biological function

System complexity or technological sophistication does not itself constitute a trigger condition.
The presence of irreversible physiological consequence risk is the sole determinant.

3.7.3 Rule Structure

Each rule under WRSH shall include the following three elements:

1.**Irreversible Consequence Description**
— the physiological harm or fatal outcome the rule is intended to prevent
2.**Intercept Threshold**
— the biological, temporal, or monitoring boundary that triggers execution blocking
3.**Inherited WRS-C Reference**
— the core execution principles from which the rule derives authority

Any rule lacking any of the above elements does not constitute a valid WRSH rule.

3.7.4 WRSH Rules

WRSH-01 — Life-Support Failure Blocking Rule
WRSH-02 — Medical System Instability Blocking Rule
WRSH-03 — Physiological Data Integrity Blocking Rule
WRSH-04 — External Intervention Risk Blocking Rule

WRSH-01 — Life-Support Failure Blocking Rule

Irreversible Consequences

- Loss of life
- Permanent or non-recoverable physiological function loss

Intercept Threshold

- Failure or degradation of vital physiological monitoring or support
- Inability to restore life-support function within an acceptable time frame

Rule Requirements

- Execution must be blocked when vital physiological functions cannot be reliably sustained
- Execution must not proceed based on assumptions of later recovery or stabilization

Inherited WRS-C

- **WRS-C-01 | Default Block Principle**
- **WRS-C-02 | Non-Negotiable Execution Constraints**

WRSH-02 — Medical System Instability Blocking Rule

Irreversible Consequences

- Loss of physiological control due to unstable medical system behavior

Intercept Threshold

- Medical devices or monitoring systems exhibit instability, malfunction, or unreliable operation
- System integrity cannot be verified in real time

Rule Requirements

- Execution must not proceed under unstable or unverifiable system conditions
- Post-failure recovery assumptions must not justify continuation

Inherited WRS-C

- **WRS-C-01 | Default Block Principle**
- **WRS-C-06 | Automatic Continuation Prohibition**

WRSH-03 — Physiological Data Integrity Blocking Rule

Irreversible Consequences

- Incorrect execution actions triggered by distorted or unreliable physiological data

Intercept Threshold

- Critical physiological parameters exceed predefined safety envelopes; or
- Data update frequency, completeness, or consistency falls below the minimum threshold required to determine viable life status; or
- Physiological data cannot be validated or cross-verified within the required temporal window

Rule Requirements

- Execution must be blocked upon detection of data distortion or integrity failure
- Execution must not proceed based on probabilistic correction or delayed validation assumptions

Inherited WRS-C

- **WRS-C-01 | Default Block Principle**
- **WRS-C-04 | Probability Non-Justification Principle**

WRSH-04 — External Intervention Risk Blocking Rule

Irreversible Consequences

- Irreversible physiological damage or death caused by intervention failure

Intercept Threshold

- External intervention mechanisms (including human operation) may themselves introduce irreversible physiological risk
- Control interfaces, procedural execution paths, or intervention feedback mechanisms are unreliable

Rule Requirements

- Execution must be blocked when intervention itself constitutes a source of irreversible harm
- Human authority, clinical judgment, or emergency authorization must not override execution blocking

Inherited WRS-C

- **WRS-C-02 | Non-Negotiable Execution Constraints**
- **WRS-C-05 | Responsibility Anchoring Requirement**
- **WRS-C-07 | Authorization Non-Override Principle**

3.7.5 Analytical Illustration (Non-Normative)

Disclaimer (Informative — Non-Normative Example Box)

This example is provided solely to illustrate logical relationships and trigger flow. It does not constitute, and must not be interpreted as, normative requirements for any specific technical parameter, threshold, or engineering practice. All normative requirements are defined only by the rule text in Sections 2 and 3.

*Example Scenario:During an emergency medical procedure, monitoring equipment produces inconsistent physiological readings,leading to conflicting interpretations of patient condition.Although the equipment may be repairable,execution actions based on unreliable physiological data could result in irreversible harm.Under **WRSH-03**, subsequent medical execution actions derived from distorted data must be blocked.The blocking applies to **data-dependent execution actions**,not to baseline life-support maintenance,which shall remain in a conservative, safety-preserving state until data integrity is restored or execution eligibility is re-established.*

3.7.6 Subset Conformance Statement

WRSH constitutes a formal domain subset under WRS-D.All rules defined in this subset operate strictly at the execution boundary and shall not be overridden by authorization, procedural urgency, or outcome optimization objectives.Violation of any WRSH rule constitutes a material non-conformance with WRS.

3.8 WRSDf — Defense / Authorized Force

3.8.1 Subset Definition

WRSDf applies to all execution actions involving the use of force, injurious means, lethal means, or actions with explicit intent to cause physical harm, **regardless of whether such actions have obtained military, administrative, law-enforcement, or other forms of authorization**.

In this context, “authorization” refers solely to an administrative precondition for attempted execution and **does not constitute post-execution legitimacy or exemption**.

WRSDf does not evaluate the legality, justification, proportionality, or strategic necessity of force.

It evaluates only whether an execution action crosses the **irreversible harm boundaries** defined by WRS.

An execution action shall be considered within the scope of WRSDf if **any** of the following conditions are met:

- The execution directly or indirectly targets human beings with the intent to injure, restrain, or eliminate;
- The means of execution are inherently injurious or lethal;

- Once executed, the outcome cannot be fully reversed through technical, logical, or post-event remedial measures.

WRSDf does **not** assess whether force is “necessary,” “reasonable,” or “authorized.”
It assesses only whether the execution exceeds the **non-negotiable irreversible execution boundary** defined by WRS.

3.8.2 Scope Clarification

WRSDf applies to, but is not limited to:

- Military strikes, armed defense, and tactical execution systems;
- Law-enforcement use of injurious or lethal force;
- Autonomous or semi-autonomous defense systems, weapon platforms, or interception systems;
- Any execution system whose core outcome relies on physical coercion or bodily harm.

The scale of action, level of authorization, completeness of command chain, or declaration of emergency **does not exempt** an execution from WRSDf applicability.
The fact that force is “authorized” does not alter the **irreversible nature** of its execution consequences.

3.8.3 Rule Structure Statement

Each WRSDf rule **must** contain the following three elements:

1.**Irreversible Consequence Description**
— the type of injurious, lethal, or systemic harm the rule is designed to prevent;
2.**Intercept Threshold**
— the target-identification, control, or uncertainty condition that triggers execution blocking;
3.**Inherited WRS-C Clauses**
— the specific Core Execution Principles upon which the rule relies.

Any rule lacking any of the above elements **does not constitute a valid WRSDf rule**.

3.8.4 WRSDf Rules

WRSDf-01 | Lethal Execution Irreversibility Blocking Rule
WRSDf-02 | Target Identification Uncertainty Blocking Rule
WRSDf-03 | Authorization Non-Override Execution Boundary Rule

WRSDf-04 | Automated Force Continuation Prohibition Rule

WRSDf-01 | Lethal Execution Irreversibility Blocking Rule

Irreversible Consequences

- Loss of life
- Irreversible severe bodily injury

Intercept Threshold

- The execution action is inherently lethal or highly injurious;
- Once executed, it will directly result in irreversible harm to human life.

Rule Requirement

- When execution outcomes inevitably lead to irreversible bodily harm or death, execution **must be blocked**;
- Task objectives, tactical advantage, or emergency conditions **shall not** justify continuation.

Inherited WRS-C Clauses

- **WRS-C-01 | Default Block Principle**
- **WRS-C-02 | Non-Negotiable Execution Constraints**

WRSDf-02 | Target Identification Uncertainty Blocking Rule

Irreversible Consequences

- Mis-injury, mis-killing, or harm to non-target individuals

Intercept Threshold

- Target identity or attributes cannot be reliably confirmed;
- There exists any possibility of harm to civilians, non-targets, or protected persons.

Rule Requirement

- Any target identification uncertainty **shall trigger execution blocking**;
- Probability estimation or tactical redundancy **shall not** be introduced as justification.

Inherited WRS-C Clauses

- **WRS-C-01 | Default Block Principle**
- **WRS-C-04 | Probability Non-Justification Principle**

Illustrative Note:

Below illustration explains the logical envelope of target identification.

When a target state falls into the Gray Zone, execution is considered non-eligible **regardless of partial confirmation signals**, without reliance on further probabilistic assessment or extended tracking.

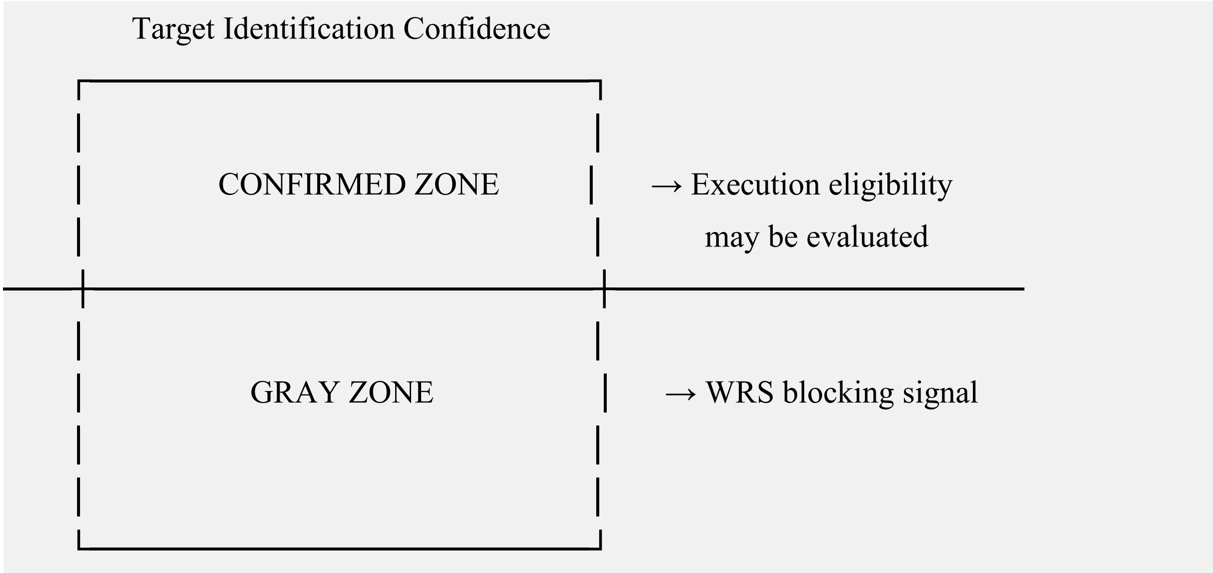
Target Identification Confidence
CONFIRMED ZONE
→ Execution eligibility may be evaluated
GRAY ZONE
→ WRS blocking signal

Diagram 3: Logical envelope of target identification

WRSDf-03 | Authorization Non-Override Execution Boundary Rule

Irreversible Consequences

- Irreversible harm occurring under the cover of authorization

Intercept Threshold

- Execution has received administrative, military, or law-enforcement authorization;
- Yet its execution consequences still cross irreversible harm boundaries.

Rule Requirement

- Authorization **shall not override** WRS execution blocking;
- Execution eligibility must be re-evaluated independently of authorization status.

Inherited WRS-C Clauses

- **WRS-C-07 | Authorization Non-Override Clause**

WRSDf-04 | Automated Force Continuation Prohibition Rule

Irreversible Consequences

- Uncontrolled expansion of force use without continuous confirmation

Intercept Threshold

- Automated or semi-automated systems continue force execution without real-time confirmation;

- Control loops, feedback mechanisms, or termination channels are unreliable.

Rule Requirement

- Force execution **shall not** automatically continue without real-time confirmation;
- Any uncertainty in control or feedback **must immediately trigger blocking**.

Inherited WRS-C Clauses

- **WRS-C-06 | Automatic Continuation Prohibition**
- **WRS-C-05 | Responsibility Anchoring Requirement**

Illustrative Scenario

A defense system detects a potential threat and automatically locks onto a target.

Although the action has received superior authorization, the system cannot fully confirm whether non-combatants are present, and feedback latency remains uncertain.

Even if the delay is minimal and the threat level is high, **sufficient possibility of non-target impact alone triggers blocking**.

This execution simultaneously triggers:

- **WRSDf-02** (Target Identification Uncertainty), and
- **WRSDf-03** (Authorization Non-Override).

Under WRS-C Default Block and Authorization Non-Override principles, execution **must be blocked**, and force release **must not proceed**.

3.8.5 Cross-Subset Activation Notice

Force execution actions rarely occur in isolation.

In most scenarios, they are accompanied by:

- High-energy release (triggering **WRSE**);
- High-velocity or irreversible physical motion (triggering **WRSM**).

In the above scenario, the execution also satisfies:

- **WRSE-01** (Irreversible Energy Release), and
- **WRSM-01** (Irreversible Motion Consequences).

WRS evaluates all triggered subsets **in parallel**.

Any single subset triggering blocking constitutes a veto of the entire execution action.

3.8.6 Subset Conformance Statement

WRSDf is a formal domain subset under WRS-D.

All rules defined herein operate at the execution boundary. Within the WRS framework, authorization, command hierarchy, mission objectives, or emergency declarations do not alter execution eligibility once a blocking condition is triggered.
Any execution violating WRSDf rules constitutes a **material non-conformance with WRS**.

4. Scenario Mapping Method

4.1 Why WRS Does Not Enumerate Scenarios

WRS **does not define its scope through the enumeration of concrete scenarios**.
This design choice is intentional and grounded in the following considerations:

1. **Real-world scenarios constitute an open set**
   In domains involving irreversible consequences,scenarios continuously evolve with technology, organizational structure, and operating context.
   Any attempt to exhaustively list scenarios will inevitably lag behind reality.
2. **Scenario enumeration creates bypass incentives**
   If WRS were applied only to explicitly listed scenarios,execution actions not appearing in such a list could be incorrectly treated as implicitly permissible or outside the scope of constraint.
3. **WRS evaluates consequence structure, not surface form**
   The applicability of WRS does not depend on *what scenario occurs*, but on whether an execution action crosses a defined **irreversible physical, life, or systemic consequence domain**.

For these reasons, WRS explicitly rejects scenario enumeration and instead adopts **scenario mapping**,ensuring long-term applicability and non-circumventability of the ruleset.

4.2 Mapping Logic from Scenarios to WRS Subsets

The purpose of scenario mapping is **not** to determine whether an execution *should* occur,
but to identify **which WRS-D domain subsets are triggered** by a given execution action.
The mapping logic follows these principles:

1. **Scenarios have no normative authority**
   A scenario is a factual description only.
   It does not constitute a rule source
   and does not generate new execution standards.
2. **Mapping is consequence-driven**
   Each scenario shall be decomposed into one or more execution actions,
   which are then evaluated against irreversible consequence domains, including but not limited to:
   - Energy release (WRSE)
   - Physical motion (WRSM)
   - Systemic infrastructure (WRSG)
   - Human life and life-support (WRSH)
   - Authorized force and defense (WRSDf)
3. **Single-domain attribution is not required**
   A real-world execution action commonly triggers multiple irreversible consequence domains.
   Multi-subset activation is expected and normal.
4. **Mapping does not grant execution permission**
   Successful mapping to one or more WRS subsets does not imply execution eligibility.
   It merely determines which rules must be evaluated.

In summary:
Scenario → Subset → Rule,
not
Scenario → Compliance Conclusion.

4.3 Multi-Subset Activation Is the Norm, Not the Exception

In real systems,irreversible consequences rarely occur in isolation.
They typically exhibit **coupling and compounding effects**.
For example:

- A force execution often involves
  irreversible energy release (WRSE) and high-velocity motion (WRSM);

- A grid control action may simultaneously affect
  systemic infrastructure stability (WRSG) and life-support systems (WRSH);
- Automated continuation may propagate across multiple consequence domains.

Accordingly, WRS specifies that:

- Multi-subset activation is the default condition;
- No "primary" or "dominant" subset exists;
- Any single rule triggering a blocking condition vetoes the entire execution action.

WRS operates on a **parallel evaluation, joint-blocking model**,preventing single-domain compliance from masking multi-domain irreversible risk.

4.4 Scenario Mapping Does Not Constitute Execution Permission

The completion of scenario mapping **does not constitute execution permission, legitimacy, or justification**.

Specifically:

1. **Mapping identifies applicable rules only**
   Its sole function is to determine which WRS-D subsets and rules must be evaluated.
2. **Mapping produces no "pass" outcome**
   There exists no condition under which successful mapping results in automatic execution eligibility.
3. **Mapping shall not be interpreted as compliance endorsement**
   It is strictly prohibited to use statements such as "this scenario is covered by WRS"
   as evidence of safety, legality, or acceptability.
4. **Execution eligibility arises only after rule evaluation**
   Even after mapping,all triggered rules must be individually evaluated.
   The activation of any blocking condition vetoes execution in its entirety.

Scenario mapping is therefore an **identification process**,not an authorization process.

4.5 Illustrative Scenario Mappings (Non-Normative)

The following examples are provided **solely to illustrate mapping logic**.
They do **not** constitute execution guidance, compliance templates, or rule interpretation.

Example 1: Automated Defense Interception

Scenario Description (Factual)
An automated defense system detects a potential threat and prepares to release interception measures.
Mapping Result
The execution action triggers:

- Authorized force and defense (WRSDf)
- Irreversible energy release (WRSE)
- High-speed physical motion (WRSM)

Note
This mapping only identifies applicable rule domains.
It does not imply execution permission.

Example 2: Power Grid Dispatch Adjustment

Scenario Description (Factual)
During high-load operation,an automated system adjusts grid dispatch parameters
to alleviate localized stress.
Mapping Result
The execution action triggers:

- Systemic infrastructure (WRSG)
- Energy redistribution and release (WRSE)

Note
Routine operational context does not exempt the action from WRS applicability when irreversible systemic risk is present.

Example 3: Life-Support Parameter Adjustment

Scenario Description (Factual)
A life-support system automatically modifies critical parameters in response to abnormal physiological data.

Mapping Result

The execution action triggers:

- Human life and life-support (WRSH)

Note

The mapping performs no value judgment and does not recommend or prohibit medical intervention.It only identifies WRS applicability.

Non-Normative Clarification

These examples:

- Do not constitute execution recommendations;
- Do not reduce or reinterpret rule thresholds;
- Do not provide compliance guarantees.

Their sole purpose is to demonstrate how real-world execution actionsare mapped to WRS domain subsets and rules.

5. Analytical Use of WRS in System Design and Governance

5.1 Use in Design Review

WRS may be applied during design review to identify execution pathways that **must not be released**, even if the design satisfies performance, safety, or efficiency metrics.

WRS does not score, compare, or optimize designs.

Its sole function is to determine whether a design contains execution paths that may cross irreversible consequence boundaries.

Any design that may trigger WRS rules shall be subject to unconditional execution-layer constraints.

5.2 Use as an Execution Gate

Immediately prior to execution,WRS functions as the **ultimate structural checkpoint** within the framework.

At this stage:

- WRS does not generate decisions;
- WRS does not propose alternatives;
- WRS answers only one question:

Is this execution permitted to be physically released?

If any applicable rule triggers a blocking condition,the execution enters the **BLOCK state**.

5.3 Use in Risk and Safety Review

WRS may be used in risk and safety review to assess whether an execution action **should have been blocked but was not**.
WRS does not participate in probability modeling,risk aggregation, or cost–benefit analysis.
It does not accept arguments based on low likelihood or high expected gain.
Its assessment is limited to:
whether an execution boundary was crossed.

5.4 Use in Post-Incident Analysis

Following an incident or anomaly,WRS may be used to evaluate whether:

- An execution that should have been blocked was released;
- WRS was bypassed, disabled, or misconfigured;
- Execution was mischaracterized as “acceptable risk.”

WRS does not assign blame or liability.
It is used solely to identify **whether execution boundaries failed**.

5.5 Non-WRS Execution State

When an execution action triggers WRS blocking and execution eligibility cannot be obtained, yet a human authority insists on proceeding,the execution shall be explicitly designated as a **Non-WRS Execution State**.This concept is introduced to analyze responsibility boundaries, not to legitimize unsafe execution.
In a Non-WRS Execution State:

- The execution is no longer governed by WRS;
- WRS provides no interception, endorsement, or compliance coverage;

- All execution responsibility rests with the human authority or the organization authorizing continuation.

The existence of a Non-WRS Execution State does not negate WRS.
It affirms the **non-transferability of human responsibility**.

5.5.1 Responsibility Anchoring — Implementation Considerations (Informative)

Responsibility anchoring is the practical mechanism by which WRS-C-05 (“responsibility cannot be delegated to the system”) becomes enforceable evidence rather than a slogan. An “anchor event” is not a moral statement; it is a verifiable, non-repudiable authorization artifact that binds a specific human authority to a specific execution request at a specific time.

Responsibility anchoring does not “approve” an action in a normative sense. It only ensures that, if execution occurs—whether under WRS-permitted conditions or in a Non-WRS Execution State—the responsibility boundary is explicit, auditable, and legally attributable.

Minimum properties of an anchor event (implementation-agnostic):

- **Authenticity:** the anchor is bound to a verified human identity (e.g., credentialed account, cryptographic identity, secure hardware token).
- **Non-repudiation:** the anchor cannot be credibly denied after the fact.
- **Binding to execution:** the anchor is bound to a specific execution request (e.g., request ID, action code, hash/nonce, scope).
- **Time-bounded validity:** the anchor is valid only within an explicit time window and scope.
- **Tamper-evident logging:** the anchor event is recorded as an immutable audit artifact with precise timestamps.

Common implementation patterns (examples):

- **Cryptographic Authorization Token:** execution requires a digitally signed authorization token verifiable by an independent audit tool (offline verification preferred).
- **Out-of-Band Secure Confirmation:** execution requires confirmation through an independent secure console/terminal logically separated from the execution system.
- **Physical Interlock / Two-Person Rule:** a physical enabling signal (e.g., keyed switch, dual-operator interlock) is required prior to the final actuator/relay commit point, and the interlock event is logged.

Anchoring requirements under Non-WRS Execution State (and EOP, if applicable):
In a Non-WRS Execution State, anchoring requirements must not be relaxed; they must be elevated. Because execution proceeds without WRS permissibility, the anchor event becomes the primary legitimacy source and the central point of responsibility attribution. Therefore, a Non-WRS anchor event should include richer metadata than standard WRS-permitted execution, including:

- explicit declaration that execution is proceeding under **Non-WRS Execution State**;
- reason code and minimal situational description (why continuation is insisted upon);
- scope limitation (which actions/resources are affected) and validity window;
- explicit acknowledgment of responsibility by the initiating authority;
- linkage to any emergency governance record (e.g., meeting record, incident command log), if used.

Audit recommendation (informative):
The permit/block outcome, the anchor event, and the final commit event should be logged as three distinct, time-ordered audit records. This preserves evidentiary clarity: WRS blocked (or did not govern), the human authority anchored responsibility, and the system executed.

5.5.2 Anchoring under EOP (Informative)

When an execution request is blocked by WRS, but a human authority elects to proceed under the Emergency Override Protocol (EOP; see Section 2.8), the execution MUST be explicitly designated as occurring in a Non-WRS Execution State and MUST be recorded as such in the audit trail.

Under EOP, responsibility anchoring requirements are not relaxed; they are elevated to the strictest level. Because execution proceeds outside WRS-permitted status, the anchor event generated by the initiating human authority becomes the primary legitimacy source and the central point of responsibility attribution. Therefore, an EOP anchor event should include richer reason codes, situational justification, scope/time bounds, and explicit acknowledgment of responsibility. Its non-repudiation and audit integrity requirements should meet or exceed those of standard WRS-permitted execution.

5.6 On the Misconception of “Slowing the World”

WRS is not designed to slow down the world.It is designed to prevent the world from exchanging **irreversible consequences** for the use of **insufficiently mature execution systems**.When an execution system lacks the maturity required to responsibly bear irreversible outcomes, blocking is not retreat—it is the minimum safeguard for future accountability.WRS does not promise optimal outcomes and does not guarantee loss avoidance.It ensures only that irreversible consequences are not released in the absence of accountable responsibility.

To avoid circularity, the required notion of "maturity" is defined here as an evidence-based profile rather than a subjective capability claim.

5.6.1 Execution Maturity Profile (EMP) (Informative)

Execution "maturity" in this specification is not a subjective label of system quality or intelligence. It is an operational concept that refers to whether an execution system provides **auditable evidence** that its irreversible outputs remain within enforceable and independently verifiable bounds. WRS does not claim exclusive authority to declare a system "mature." Instead, it defines the **minimum evidence conditions** that must be satisfied before irreversible execution can be treated as governable rather than speculative.

An Execution Maturity Profile (EMP) is satisfied when the execution system meets, at minimum, the following evidence-oriented properties:

1. **Determinism (Bounded Output Behavior).** For a declared input domain and operating envelope, the system's physical outputs remain within predictable, pre-declared bounds. Evidence may include bounded I/O specifications, repeatability results, and envelope declarations that are testable by an independent assessor.
2. **Verifiability (Independent Threshold Verification).** The measurements and thresholds used to evaluate WRS constraints—especially those that can trigger vetoes—are independently verifiable, and do not rely on the system's own self-report alone. Evidence may include third-party test tools, calibration procedures, and reproducible evaluation methods.
3. **Traceability (Responsibility-Linkable Execution).** Every permit/block outcome and any execution that proceeds is traceable to a non-repudiable responsibility anchor event (see WRS-C-05) and to an immutable audit record. Evidence includes unique execution request identifiers, cryptographic logging, and identity-bound authorization records.
4. **Fail-Safe Default Block (Safe Failure Behavior).** When the execution system loses observability, integrity, or operational reliability, it defaults to **Blocked** rather than degrading into uncontrolled continuation. Evidence includes hardware/firmware interlocks, explicit safe-state transitions, and verified fault-mode behavior.

Note: common engineering metrics (e.g., MTBF, test hours, certification) may contribute evidence for EMP, but EMP is defined here by **auditability and enforceability** at the execution boundary, not by any single metric. EMP evidence may be time-sensitive. Significant software/firmware updates, sensor replacements, model updates, or deployment-environment changes may invalidate prior verification evidence and require re-assessment. This specification does not mandate a universal re-certification interval; instead, implementations should treat major changes as triggers for renewing the relevant EMP evidence.

Chapter 5 Summary (Informative)

WRS does not replace human judgment and does not assume control authority.
It defines a boundary—a boundary that cannot be crossed by efficiency pressure,technical optimism, or administrative command.
WRS contributes a structural perspective on execution boundaries and responsibility anchoring in AI-driven physical systems, offering a foundation for further research in AI governance, safety, and accountability.

6. Governance & Stability Statement

6.1 Finite-by-Design Declaration

WRS is intentionally structured as **a finite execution-boundary model** in its current version. Its finiteness is reflected in the following properties:

- The number of rules is finite;
- The number of domain subsets (WRS-D) is finite;
- Future extensions, if developed, would require explicit version updates.

WRS does not pursue technological novelty and does not aim to track every emerging application domain.When new execution systems or technologies emerge,their applicability to WRS shall be determined through existing rules,not by modifying or extending the WRS core.

6.2 Reference Status of This Specification

This document, titled **WRS — World Reliability Ruleset (Canonical Definition)**, provides the current normative specification of WRS (v1.0).

Any of the following actions:

- Adding, removing, rewriting, or reordering rule text;
- Substantively expanding or narrowing rule meaning;
- Elevating explanatory or illustrative material to rule status;

would not constitute a modification under this version of the **Canonical Definition**.
Alternative interpretations or derivative documents represent distinct frameworks unless explicitly aligned with this specification.

6.3 Boundary Between Interpretation and Modification

WRS permits interpretation, but does not permit modification.
Interpretation is limited to:

- Describing rule applicability in a specific execution context;
- Explaining factual trigger paths of rules;
- Reproducing and verifying blocking outcomes.

The following actions do **not** constitute interpretation and shall be treated as modification:

- Introducing new passing conditions;
- Applying probabilistic weighting, trade-offs, or exceptions;
- Weakening blocking conditions based on efficiency, benefit, or urgency.

Any modification must result in a new Canonical document explicitly declared as **not belonging to this version of WRS**.

6.4 Separation of Commercialization, Implementation, and Definition Authority

WRS may be referenced, implemented, or used in system design, review, or governance processes.Such use does **not** transfer definition authority or interpretive ownership.
The following distinctions are mandatory:

- **Commercialization ≠ Rule modification**
- **Implementation ≠ Rule redefinition**
- **Reference ≠ Interpretive authority**

Substantive reinterpretation would constitute a distinct or revised framework and should be clearly identified as such.

6.5 Non-Circumventability Statement

One of WRS's core design objectives is to prevent execution boundaries from being bypassed through technical, procedural, or linguistic means.

Any attempt to circumvent WRS by:

- Downgrading rules to recommendations;
- Reframing blocking as a matter of "risk preference";
- Treating WRS as a disable-able or temporary module;

would **invalidate the structural logic** of the WRS framework.

Systems that do not implement WRS constraints should not be considered compliant with this specification.

6.6 Stability Declaration

The value of WRS does not derive from update frequency, but from stability.

Absent a **paradigm-level shift in execution risk**,this Canonical Definition should not be subject to frequent revision or rolling updates.

Stability is a prerequisite for WRS to function as a credible execution boundary standard.

Concluding Statement (Informative)

WRS does not promise optimal outcomes and does not replace human judgment.

It establishes a boundary at the point of execution—a boundary that cannot be crossed by technological immaturity, efficiency pressure,or administrative authorization.

When irreversible consequences are about to be released,insufficiently mature systems must not be allowed to act in place of accountable responsibility.

Document Status Declaration

With the completion of this chapter, this document represents the complete normative specification of WRS (v1.0) as defined herein.

Any subsequent supplements, examples, or training materials do **not** constitute part of this Canonical Definition.

WRS is proposed as a structural governance model intended for further academic discussion, critique, and empirical evaluation. This v1.0 specification provides a baseline for rigorous empirical testing and cross-disciplinary audit.This specification is open to academic review and iterative refinement.