PG-1 – Persistence Gate

Pre-Verification Admissibility Determination

by PanXnubis Gaia Ladrieh
1/04/26

1. Governing Principle

Only solutions persist.

2. Scope

This gate applies to any proposed solution, model, system, control, algorithm, architecture, or decision framework **prior to:**

- Verification
- Validation
- Optimization
- Calibration
- Deployment
- Risk Acceptance

PG-1 determines **admissibility.**Verification and validation operate exclusively on admissible configurations; inadmissible configurations are eliminated prior to verification.

Rule: Only admissible configurations persist.

3. Definitions

- **Admissibility:** A configuration is *inadmissible* if **any** constraint is violated.
- **Configuration:** Any arrangement, model, system, policy, algorithm, architecture, or construct subject to constraints.
- **Constraint:** Any condition, requirement, dependency, boundary, assumption, limit, interaction, or environmental factor that can affect persistence. Constraints may be known or unknown.
- **Correctness:** A configuration is *correct* **if and only if** it is admissible.
- **Violation:** Any state in which **any constraint** is not satisfied.
- **Exposure:** The condition under which constraints are active, including simultaneous activation.
- **Persistence:** Continued existence or validity of a configuration under constraint exposure.

- **Solution:** A configuration whose persistence under constraint exposure is not eliminated.

No configuration is a solution by assertion, intent, or declaration. Solution status is determined solely by persistence.

4. Exposure Declaration (Required)

The submitter SHALL provide an Exposure Declaration for the configuration under evaluation.

The Exposure Declaration is a statement of known interaction with constraints.

It **does not** establish admissibility.

4.1 Constraint Enumeration

The submitter SHALL enumerate all **known constraints** applicable to the configuration, including but not limited to:

- Physical constraints
- Logical constraints
- Environmental constraints
- Operational constraints
- Computational constraints
- Temporal constraints
- Human or organizational dependencies
- Assumptions required for operation

Constraints SHALL be stated explicitly.

Implicit or “understood” constraints are treated as unstated.

4.2 Violation Conditions

For each enumerated constraint, the submitter SHALL state:

- Conditions under which the constraint is violated, or
- Whether violation conditions are unknown

Absence of known violation conditions does not imply absence of violation.

4.3 Non-Persistence Conditions

The submitter SHALL state all known conditions under which the configuration:

- Ceases to function,
- Produces invalid output, or
- Loses coherence, stability, or integrity

If no such conditions are known, this SHALL be stated explicitly.

4.4 Assumed Absences

The submitter SHALL declare any constraints assumed to be:

- Absent
- Negligible
- Deferred
- Suppressed
- Outside Scope

Assumptions do not remove constraints from exposure.

4.5 Unknown Constraints

Any constraint not enumerated is treated as **unknown.**

Unknown constraints are included in exposure by default.

The inability to enumerate all constraints does not invalidate the declaration. It increases exposure.

4.6 Declaration Status

The Exposure Declaration:

- Does not assert correctness
- Does not establish completeness
- Does not guarantee persistence

It serves only to define the conditions under which persistence is evaluated.

5. Persistence Test

The Persistence Test evaluates whether a configuration qualifies as a solution.

It does so without appeal to likelihood, mitigation, sequencing, or

selective activation.

5.1 Test Condition

The configuration SHALL be evaluated under **constraint exposure**, defined as:

The simultaneous activation of all enumerated constraints and all unknown constraints.
Sequential, conditional, or probabilistic activation SHALL NOT be substituted for exposure.

5.2 Evaluation Question

The following question SHALL be answered without qualification:

Does the configuration persist under constraint exposure?

No additional framing, explanation, or mitigation SHALL be appended to the response.

5.3 Permitted Responses

Permitted responses are limited to:

- **Yes**
- **No**
- **Unknown**

Responses other than the above are invalid.

5.4 Interpretation

- **Yes** indicates persistence under exposure.
- **No** indicates violation of at least one constraint.
- **Unknown** indicates inability to establish persistence.

Inability to establish persistence SHALL be treated identically to non-persistence.

5.5 Non-Persistence

A configuration is considered **non-persistent** if:

- Any constraint is violated under exposure, or
- Persistence cannot be demonstrated under exposure

Any qualification of persistence introduces a constraint violation under exposure. k

5.6 Prohibited Qualifications

The following SHALL NOT be used to qualify or override the Persistence Test:

- Statistical confidence
- Risk tolerance
- Expected operating regimes
- Redundancy or fallback mechanisms
- Human oversight or intervention
- Post-failure recovery
- Deferred validation or monitoring

These considerations do not convert non-persistence into persistence.

5.7 Determinism of Outcome

The outcome of the Persistence Test is determined solely by the response provided in Section 5.3.

No additional review, interpretation, or weighting is permitted at this stage.

6. Admissibility Determination

6.1 Admissibility Criteria

The configuration is considered **admissible** if and only if the Persistence Test response is "**Yes.**"

Any other response – "**No**" or "**Unknown**" – results in the configuration being **inadmissible.**

6.2 Outcome Implications

- **Admissible Configuration:**
  - May proceed to subsequent stages, such as Verification, Validation, Optimization, or Deployment, without further modification.
  - The configuration meets the fundamental requirements of persistence under exposure.
- **Inadmissible Configurations:**
  - Shall not proceed to subsequent stages.
  - The configuration fails to demonstrate persistence and must

be reconsidered, redesigned, or discarded.
- No exceptions, qualifications, or mitigations are allowed at this stage.

6.3 No Remediation

Admissibility is determined solely by the Persistence Test. There is no provision for remediation, adjustments, or iterative improvements at this gate.

If a configuration is inadmissible, it must be re-evaluated or re-engineered before any further attempts.

6.4 Final Authority

The Admissibility Determination is final at this stage.

It is not subject to appeal, negotiation, or reinterpretation.

The gate serves as a **hard boundary** for moving forward.

6.5 Documentation

The outcome of the Admissibility Determination, along with the Exposure Declaration and Persistence Test results, shall be documented thoroughly for transparency and traceability.

6.6 Summary

In summary, **only configurations that persist under full constraint exposure are admissible.** This ensures that no solution is considered valid unless it can withstand all constraints simultaneously.

7. Final Determination and Position

This gate establishes a pre-verification admissibility boundary based on a single condition:

Only solutions persist.

No configuration proceeds beyond this boundary unless persistence under constraint exposure is established.

7.1 Relationship to Verification and Validation

This gate does not replace Verification or Validation.

It precedes them.

Verification and Validation evaluate properties of configurations that are already admissible. This gate determines whether a configuration qualifies as a solution at all.

Configurations that fail to persist under exposure are excluded prior to downstream effort.

7.2 Scope of Authority

This gate asserts no methodological, institutional, or procedural authority.

It does not prescribe:

- design methods,
- modeling techniques,
- validation strategies,
- or operational policies.

It enforces a single admissibility condition.

7.3 Independence from Methodology

The Persistence Gate is agnostic to:

- deterministic or probabilistic methods,
- statistical or analytical frameworks,
- simulation-based or empirical approaches,
- human-in-the-loop or autonomous systems.

Methodological choice does not alter admissibility.

7.4 Failure without Exception

A configuration that fails to persist under exposure is not admissible.

This determination is independent of:

- expected operating regimes,
- likelihood of occurrences,
- compensatory mechanisms,
- redundancy,
- oversight,
- or post-failure response.

Persistence is evaluated prior to execution.

7.5 Final Statement

This gate does not determine correctness, optimality, or performance.

It determines *only* this:

Whether a configuration qualifies as a solution.

Configurations that do not persist under exposure do not qualify.

End of Gate

Appendix A – Formal Basis of Persistence Admissibility

A.1 Configuration Space

Let:

- C be the space of all configurations
- $K = \{k_1, k_2, \ldots\}$ be the set of constraints (known and unknown)

Each constraint induces a predicate:

$k_i : C \to \{0,1\}$

where $k_i(c) = 1$ indicates satisfaction.

A.2 Persistence Definition

Define persistence as:

$P(c) = \Pi_{(k_i \in K)}\ k_i(c)$

Persistence holds iff:

$P(c) = 1$

If **any** constraint is violated:

$\exists k_j \in K : k_j(c) = 0 \Rightarrow P(c) = 0$

A.3 Solution Characterization

A solution is not defined *a priori*.

It is characterized as:

$S = \{c \in C \mid P(c) = 1\}$

No other definition is admissible.

A.4 Simultaneity Requirement

Sequential or probabilistic evaluation replaces:

$\Pi\ k\_i(c)$

with expectations, marginals, or conditional subsets.

These are not equivalent.

Therefore, any admissibility criterion not evaluating the full product over K is incomplete.

A.5 Unknown Constraints

Unknown constraints are modeled as latent predicates:

$k\_u \in K$

with undefined evaluation.

Undefined predicates force:

$P(c) \neq 1$

Thus, unknown constraints increase exposure and cannot be ignored.

A.6 Consequence

Any framework that:

- averages constraint satisfaction,
- assigned probabilities to violation,
- defers failure handling,
- or assumes selective activation

implicitly evaluates a different function than $P(c)$.

Such frameworks do not establish admissibility.

A.7 Elimination and Persistence

Let configuration space C be subjected to constraint exposure K.

Any configuration $c \in C$ for which $P(c) = 0$ is eliminated.

Elimination is final and requires no interpretation.

The set of configurations not eliminated under exposure constitutes the remaining configuration space.

No additional criteria are required to designate elements of this set as solutions.

A configuration that does not persist under exposure is eliminated and therefore cannot be instantiated, interacted with, evaluated, or distinguished from failure.

Persistence is not a property assigned to solutions. Persistence is the process by which non-solutions are removed.

What remains after elimination is the only configuration space available for operation.