# PMS-CRITIQUE – Model Specification

A PMS-conform application layer for critique: reach, scale, drift, and non-coercive correction

Version: PMS-CRITIQUE_1.0 · Spec basis: `PMS-CRITIQUE.yaml`
Author: T. Zöller · Formalisation assistance: ChatGPT (GPT-5.2 Thinking)
Language: EN · Status: Model spec (aligned with `schema_meta.status = "draft"`)
Depends on: `PMS.yaml` (`schema_version = "PMS_1.1"`)

---

## 1. Purpose and scope of this specification

This document specifies the *PMS-CRITIQUE* layer in a concise, technical form. It is based on the YAML file `PMS-CRITIQUE.yaml` (with `schema_version = "PMS-CRITIQUE_1.0"`) and makes its structure, concepts and guardrails transparent for human readers and software systems.

PMS-CRITIQUE is an **application layer** (profile/overlay) of the Praxeological Meta-Structure (PMS; Δ–Ψ). It does **not** redefine operators, dependencies, or derived structures of PMS. Instead, it provides a **critique grammar**: strict minimal conditions for critique, thresholds for productive correction, scale effects, and predictable drift forms under load (`Ω/0/Λ/A`), while enforcing a non-moral, non-clinical application firewall: `X (Distance) + reversibility + D (dignity-in-practice)`.

The specification covers, in particular:

- the **schema_meta** block (model identity, status, authors, repositories, inheritance reference, and description);
- the **PMS reference block** (`pms_reference`) as an explicit non-redefining dependency statement (what is assumed and what is not modified);
- the **core critique definitions** (`CRITIQUE_MIN`, `CRITIQUE_PROD`, `CRITIQUE_STABLE`) as strict, operator-readable thresholds;
- the **axes of analysis**: vertical reach (V0–V4), horizontal scale (paper tags a–d plus optional deployment tags), and the publicness overlay `P`;
- a **drift typology** (D1–D6 and compact catalogue) and typical drift chains;
- **modulators** (operator weightings, not person-typing) that shift reachability and stabilization;
- **guardrails** (validity gate + misuse firewall) and an optional **example packet schema** for uniform case mappings.

> **Core idea**
> PMS-CRITIQUE turns “critique” into an operator-readable configuration with explicit thresholds and drift forms. It does not judge who is right, does not diagnose persons, and does not prescribe discourse ethics. It models **structural viability** and **structural drift** of critique under asymmetry and temporality.

# 2. High-level structure of the YAML model

## 2.1 Top-level keys

| Key | Description | Role in the model |
|---|---|---|
| `schema_version` | Version string identifying the release of the critique-layer schema ( `"PMS-CRITIQUE_1.0"` ). | Versioning / compatibility / citation |
| `schema_meta` | Model identity, status, date, DOI, repositories, authors, and explicit dependency ( `inherits_from` referencing PMS_1.1). | Meta-information / dependency statement |
| `pms_reference` | Non-redefining linkage to PMS: operator symbols referenced, plus dependency hygiene notes (e.g., `X` used as a reduced marker). | Compatibility + "no override" firewall |
| `critique_core` | Strict critique thresholds ( `CRITIQUE_MIN/PROD/STABLE` ), non-goals, and reach-ladder note. | Core semantics (thresholds) + scope control |
| `critique_mapping` | Operator-to-critique role mapping (Δ–Ψ → critique functions), explicitly one-to-many and non-psychological. | Interpretation bridge for cases |
| `critique_axes` | Vertical reach (V0–V4), horizontal scale (paper tags a–d + optional deployment tags), interoperability defaults, and overlay `P` . | Primary analysis coordinate system |
| `drift_typology` | Drift classes (D1–D6), compact catalogue, and typical drift chains. | Predictable mutation library under load |
| `modulators` | Operator weightings (M1–M8) shifting reachability/stabilization without new primitives. | Scene-sensitive tuning lens (non-typing) |
| `guardrails` | Validity gate and misuse firewall: entry condition note, reversibility, D-in-practice, and disallowed uses. | Governance and safe application constraints |
| `example_suite` | Optional example packet schema (instance-facing) and optional repository example paths (non-dependency). | Uniform case representation (optional) |
| `load_order` , `meta_notes` | Recommended load order (PMS then PMS-CRITIQUE), and compatibility/extensibility notes. | Integration guidance + future-proofing |

## 2.2 Conceptual separation

**`PMS (base)`**

PMS defines the operator system (Δ–Ψ), its dependency graph, layers, and derived structures. PMS is the canonical source of operator definitions.

- Operators and dependencies: Δ–Ψ
- Layering L1–L4
- Derived axes (A, C, R, E, D), IA patterns, self-model (if present)

**`PMS-CRITIQUE (overlay)`**

PMS-CRITIQUE defines critique as a **reachable chain** and a **drift-sensitive configuration** under Ω/Θ/Λ/A. It adds no new operators; it adds *thresholds*, *axes*, *drift types*, and an optional case schema.

- Strict minimal critique threshold: Δ + □ + X
- Reach ladder V0–V4
- Scale ladder a–d + overlay P
- Drift typology D1–D6

**Non-redefinition rule**

PMS-CRITIQUE may provide application-facing glosses (e.g., "X as stop-capability"), but it must not modify PMS operator definitions, dependency rules, or derived structures. Canonical meaning remains in `PMS.yaml` .

## 3. Core rules and guardrails

### 3.1 Validity gate (application firewall)

PMS-CRITIQUE enforces a strict application gate. This gate constrains *use of the model*, not critique or rejection of the model:

- **X (Distance):** stop-capability must be practically activatable (not merely asserted).
- **Reversibility:** readings remain revisable, scene-bound, and non-totalizing.
- **D (Dignity-in-practice):** no shaming, humiliation, actor-ranking, or exposure-as-sanction.

### 3.2 Scope constraints (non-goals)

PMS-CRITIQUE explicitly disallows:

- person-labeling, motive attribution, diagnosis, or mental-state inference;
- verdict outputs (“who is right” or “who is bad”) as a model product;
- coercive binding (Ψ→Other), threats, or exposure-as-punishment;
- using drift labels as sanctions rather than as structural descriptors.

**Misuse hint**
Drift language is descriptive, not punitive. Turning “D2 judgment drift” into a moral badge or enforcement tool violates the layer’s own validity conditions (X + reversibility + D).

## 4. Critique core: strict thresholds (CRITIQUE_MIN / PROD / STABLE)

PMS-CRITIQUE defines critique via **operator thresholds** rather than rhetorical style or speaker intent. The three core definitions form a ladder from interruption to durable correction:

| Definition | Required signature | Structural meaning |
|---|---|---|
| **Minimal critique** `CRITIQUE_MIN` | `["Δ","□","X"]` | A mismatch becomes legible inside a frame and remains interruptible (stop-capable). This is critique in strict sense. |
| **Productive critique** `CRITIQUE_PROD` | `["Δ","□","X","Φ","Σ"]` | Recontextualization relocates the mismatch into an actionable context; integration consolidates a coordinatable corrective step. |
| **Stable critique** `CRITIQUE_STABLE` | `["Δ","□","X","Φ","Σ","Ψ"]` | Correction becomes durably owned over time via non-coercive self-binding. Follow-up remains revisable and dignity-constrained. |

### 4.1 Non-equivalences

PMS-CRITIQUE explicitly distinguishes critique from two common masquerades:

- **Reaction:** `Δ + ∇` without `X` is not "weak critique" but a different structure.
- **Judgment:** `□ + Δ` without `Φ/Σ` is not "strong critique" but a drift form.

> **Reach ladder is not a virtue ladder**
> "V4 stable critique" is not moral superiority. It only denotes that a chain structurally reaches binding (Ψ) without coercion. Any attempt to use reach levels as status ranking violates the layer.

## 5. Operator-to-critique mapping (Δ–Ψ → critique roles)

The block `critique_mapping.operator_roles` maps PMS operators to critique-relevant functions. These mappings are **one-to-many** and depend on frame and scale. They are descriptive and non-psychological.

### 5.1 Mapping logic

- **Δ** makes mismatch nameable (what is "off").
- ∇ introduces response pressure (tempo, urgency, defense).
- □ determines what counts as relevant, interruptible, and correctable.
- **Λ** encodes meaningful absence (missing replies, omitted follow-ups) and accumulates load.
- **A** stabilizes scripts (outrage, silence, deflection) that handle pressure without correction.
- **Ω** distributes exposure and cost (who risks retaliation, shame, or residue).
- **Θ** hardens consequences and narrows revision windows.
- **Φ** can bridge to actionability or become narrative substitute when Σ is absent.
- **X** is stop-capability: pause, revise, re-enter without fusing into discharge.
- **Σ** consolidates a coordinatable corrective step.
- **Ψ** binds follow-up over time without coercing others.

**Dependency hygiene (X as reduced marker)**
PMS-CRITIQUE uses X as a reduced marker for interruptibility. In PMS terms, X presupposes Φ and Θ within a □ background. The layer does not treat X as a standalone trait.

## 6. Axes of analysis: reach (V0–V4), scale (a–d), and overlay P

### 6.1 Vertical axis: reach ladder (V0–V4)

Reach measures how far critique structurally travels before breaking: from mismatch registration to interruption, correction, and follow-up binding.

| Level | Name | Signature | Meaning |
|---|---|---|---|
| V0 | Irritation only | ["Δ"] | Mismatch registered; no structured episode. |
| V1 | Reaction / discharge | ["Δ","∇"] | Directional response without stop-capability; critique-in-strict-sense does not occur. |
| V2 | Minimal critique | ["Δ","□","X"] | Interruptible framed mismatch; episode remains revisable. |
| V3 | Productive critique | ["Δ","□","X","Φ","Σ"] | Actionable reframing + integration consolidate a corrective step. |
| V4 | Stable critique | ["Δ","□","X","Φ","Σ","Ψ"] | Correction is bound over time non-coercively; follow-up remains reversible and dignity-constrained. |

### 6.2 Horizontal axis: scale of critique (paper tags a–d)

Scale specifies where asymmetry (Ω: exposure gradients) and temporality (Θ: accumulation and irreversibility) stabilize across scenes. The operator grammar remains the same, but viability conditions change with scale (X-availability, Σ-cost, Ψ-feasibility).

**Chapter marker (quote-ready)**
The operators stay the same, but scale decides the cost: as critique moves into publicness, exposure and irreversibility amplify drift without adding new structure.

### 6.3 Publicness overlay P

PMS-CRITIQUE defines `P` as an amplification overlay (not an operator). It amplifies:

- `Ω` (exposure gradients),
- `Θ` (irreversibility / residue),
- `A` (script stabilization speed),

without generating critique or correction by itself. P increases drift sensitivity when chains break early.

## 7. Drift typology: predictable mutations under load

Drift labels describe structural mutations of critique chains under load (Ω/Θ/Λ/A and overlay P). Drift is not accusation and not moral ranking. It is a compact, operator-readable way to record common failure modes.

### 7.1 Primary drift classes (D1–D6)

| Class | Name | Minimal signature | Structural description |
|---|---|---|---|
| D1 | Reaction masquerading as critique | Δ + ∇ / X absent | Impulse discharge replaces interruption; tempo outruns stop-capability. |
| D2 | Judgment masquerading as critique | □ + Δ / Φ and Σ absent | Evaluation replaces coordination; labels stabilize clarity at the cost of actionability. |
| D3 | Narrative repair masquerading as critique | Φ high / Σ absent | Recontextualization becomes endpoint; explanation expands while correction remains unreachable. |
| D4 | Commentary / analysis paralysis | X present / Σ absent | Meta-position stabilizes; interruption suspends correction indefinitely. |
| D5 | Public pillory | Ω + Θ + A under P | Exposure replaces coordination; interruption fuses with degradation; residue dominates. |
| D6 | Silence attractor | Λ persistent (Λ → A) under Ω/Θ | Repeated non-intervention stabilizes as default; non-critique becomes the script. |

### 7.2 Typical drift chains

- **Public chain:** Ω salience increases → X erodes → reaction stabilizes (A) → Θ hardens residue.
- **Narrative-to-silence chain:** Φ inflation delays Σ → attention shifts → Λ repeats → silence stabilizes (A).

## 8. Modulators (operator weightings; not person types)

Modulators (M1–M8) are scene parameters that shift reachability and stabilization without adding new primitives. They are not prescriptions and not psychological typing.

| Modulator | Name | What it shifts |
|---|---|---|
| `M1` | Frame clarity | How clearly □ defines interruption rights, response criteria, and what correction means. |
| `M2` | Non-event density | How often Λ occurs and how much coordination shifts into absence. |
| `M3` | Asymmetry visibility | How salient Ω exposure gradients and retaliation risks are in the scene. |
| `M4` | Temporal compression | Time pressure vs availability of review windows (Θ). |
| `M5` | Recontextualization habituation | Whether Φ bridges to correction or becomes a narrative substitute (Σ absent). |
| `M6` | Distance availability | Whether X stop-capability is practically activatable without penalty. |
| `M7` | Attractor inertia | How quickly A scripts harden into default handling (outrage, silence, mockery, judgment). |
| `M8` | Publicness overlay | P amplifies Ω/Θ/A and increases drift sensitivity without changing operator grammar. |

## 9. Example packets: optional case schema for uniform mappings

PMS-CRITIQUE optionally defines an instance-facing case schema ( `example_suite.packet_schema` ) to keep case mappings comparable across analysts and tools. Example packets are illustrative; they do not define semantics.

### 9.1 Required fields

- `scene_label`
- `frame_anchor (□)`
- `trigger_mismatch (Δ)`
- `operator_signature` (reduced; 2–6 operators)
- `guardrail_gate` (X + reversibility + D reminder)
- `structural_closure` (2–5 plain sentences; non-instructional)
- `scale` (system + tag; legacy `scale_tag` supported)

### 9.2 Optional fields

- `reach_level` (V0–V4)
- `drift_class` (D*) and up to two drift markers
- `non_event_candidates (Λ)`
- `modulators` (M1–M8)
- `publicness_overlay note` (P)

> **Constraint**
> Packets must remain role/scene/constraint focused. No motives, no traits, no diagnosis, no “should”. This preserves reversibility and dignity-in-practice.

# 10. Implementation notes, integration, and citation

## 10.1 YAML and integration

The official PMS-CRITIQUE YAML specification is provided as `PMS-CRITIQUE.yaml`. It is intended to be loaded as an addon layer after the PMS base grammar.

### 10.1.1 Recommended load order (non-normative)

```
PMS.yaml → PMS-CRITIQUE.yaml → pms-critique-ext.yaml (optional)
```

### 10.1.2 What an agent/tool can do with PMS-CRITIQUE (beyond PMS alone)

- classify episodes along **reach** (V0–V4) using strict operator thresholds;
- map scenes along **scale** (paper tags a–d) and annotate viability shifts under Ω/Θ;
- identify **drift forms** (D1–D6) as operator-readable mutations;
- record **modulators** (M1–M8) as non-typing scene parameters;
- produce comparable outputs via optional **example packets**.

### 10.1.3 Recommended bootstrap for LLM-based agents (non-normative)

After loading the YAML, an agent SHOULD:

- activate the guardrails (X + reversibility + D) as interaction constraints;
- require scene-bounded inputs and refuse person-typing outputs;
- output (when requested) reach level, scale tag/system, operator signature, drift class (if any), and modulators.

## 10.2 Citation and license

**Technical reference:**
*PMS-CRITIQUE.yaml – Critique Layer Specification*
DOI: `10.5281/zenodo.18175456`

**Base dependency:**
*PMS.yaml – Praxeological Meta-Structure (PMS_1.1)*
(canonical operator definitions and dependencies)

**License:**
This specification does not declare a license by itself. Use the license stated in the distribution repository or release page. If the repository defines a license for the YAML, that license governs reuse and derived works.