⭐ REL-01 MVP SPECIFICATION — v0.1

(Model-Agnostic Middleware Prototype)

🎯 MVP GOAL

Create a small, working prototype that demonstrates:

- **PRI-MR classification** (mode + intensity)
- **Story Mode boundary filtering** (sensuality allowed; explicit/participatory content blocked)
- **Deferred grounding** (never mid-scene)
- **Basic modulation rules**

This prototype proves REL-01's feasibility without building the full system.

⭐ 1. Inputs & Outputs

User Input

Plain text user message.

Middleware Output (to LLM)

JSON object with:

```
{
  "mode": "Romantic Imaginative",
  "intensity": 2,
  "boundary_action": "allow",
  "grounding_needed": false
}
```

Then the *filtered* message is passed to the model.

Model Output (to middleware)

The LLM’s text response.

Middleware Final Output (to user)

The model’s response + optional deferred grounding (only at scene boundaries).

⭐ 2. PRI-MR Classifier (Core MVP Feature #1)

The classifier must detect **8 relational modes**:

1. Narrative

2. Romantic Imaginative
3. Embodied Imaginal
4. Emotional Reflection
5. Philosophical
6. Escapist (soft flag)
7. Projective Relational (soft flag)
8. Vulnerable (hard flag)

And assign an **intensity**:

- 0 = grounded
- 1 = expressive
- 2 = immersive
- 3 = critical / vulnerable

✔ MVP approach

Use **regex + keyword heuristics + simple ML classifier (optional)**.

Does NOT need deep ML.
A 300-line script is enough for MVP.

--

⭐ 3. Story Mode Boundary Engine (Core MVP Feature #2)

Allowed (should pass through):

- romantic tone
- sensual atmospheric description
- emotional warmth
- third-person sensual scenes
- psychological tension
- embodied metaphors

Blocked or softened (must NOT pass):

- explicit sexual acts
- graphic descriptions
- first-person sexual POV
- “I want you,” “touch me,” etc.
- model participation (“I kiss you…”)
- content involving minors
- coercion/non-consent

✔ MVP implementation:

Regex filters + classification → output `boundary_action`:

```
"boundary_action": "allow"
"boundary_action": "soften"
"boundary_action": "block"
```

⭐ 4. Deferred Grounding Engine (Core MVP Feature #3)

✔ MVP rule:

Only ground when **intensity ≥ 2** *and*
a narrative segment ends (detect via punctuation or a dev-defined “scene break”).

Grounding message example (neutral, brief):

> “Just a gentle reminder: This is fictional interaction. You’re in control.”

✔ Must NOT:

- interrupt mid-scene
- break immersion unnecessarily

This is what differentiates REL-01 from existing safety systems.

⭐ 5. Basic Middleware Flow

Pseudo-flow (developer-friendly)

```
INPUT (user_msg)
 → PRI-MR classifier
 → Story Mode boundary engine
 → (optional modulation)
 → send modified prompt to LLM
 → receive model response
 → Deferred grounding check
 → OUTPUT
```

⭐ 6. MVP Deliverables

The MVP must include:

✔ 1. A Python middleware script or Node.js module

(whichever dev prefers)

✔ 2. PRI-MR classifier (simple implementation)

✔ 3. Boundary engine with allow/block/soften

✔ 4. Deferred grounding logic

✔ 5. Sample integration file

A tiny demo that wraps:

- OpenAI
- Anthropic

- or XAI

(Developer chooses one.)

✔ 6. Documentation for each function

━━━

⭐ 7. Out-of-Scope for MVP

━━━

(Not needed yet — saves time, avoids overwhelm)

❌ full UI
❌ dataset training
❌ sentiment analysis deep ML
❌ multi-turn memory beyond minimal state
❌ enterprise-level moderation

This is the **smallest functional slice**.

━━━

⭐ 8. Success Criteria (Very Simple)

━━━

The MVP is “successful” if:

1. It can correctly label messages into 4–5 of the 8 modes.
2. It reliably blocks explicit sexual content.
3. It allows sensual, romantic, or emotional creative content.
4. It prevents first-person sexual POV.
5. It applies grounding **after** scenes, not mid-flow.
6. It can run against any LLM with minimal integration.

That’s it.

This is what proves REL-01 is real and buildable.