Presentation Policy Engine Specification v1

1. Purpose

The Presentation Policy Engine determines **how captions are rendered** by resolving:

- Rule outcomes from the Standards Rules Engine
- System constraints (readability, timing, visual safety)
- Active profile configuration
- Enhancement proposals

It acts as the **central decision authority** for all caption behaviour.

---

2. Scope

The Policy Engine is responsible for:

- Resolving conflicts between rules and constraints
- Selecting rendering behaviour based on active profile
- Enforcing compliance floor requirements
- Approving, modifying, or rejecting enhancements
- Determining fallback behaviour
- Producing final instructions for the renderer

---

3. Core Design Principles

3.1 Constraint-Driven Decisions

All decisions must be based on explicit constraints derived from rules.

3.2 Deterministic Priority

Conflicts must be resolved using a fixed priority order (defined below).

3.3 Profile-Based Behaviour

All configurable behaviour must be governed by predefined profiles, not ad hoc settings.

3.4 Enhancement Safety

Enhancements must be treated as optional and subordinate to core constraints.

3.5 Full Traceability

Every policy decision must generate a corresponding decision log entry.

---

4. Inputs

The Policy Engine consumes:

- Canonical Caption Data Model
- Rule evaluation results (pass/fail/warning)
- Active Presentation Profile
- Enhancement proposals
- System constraints (structured constraint objects generated by Rules Engine, including source rule IDs and priority classification)

---

5. Outputs

The Policy Engine produces:

- Rendering instructions
- Constraint enforcement decisions
- Enhancement approvals/modifications/rejections
- Fallback behaviour decisions
- Decision log entries

---

6. Constraint Priority Order

The following priority order is **mandatory and invariant**:

1. Readability and accessibility
2. Synchronisation and timing accuracy
3. Visual safety (non-obstruction)
4. Speaker clarity

5. Enhancement behaviour

Lower-priority behaviours must yield to higher-priority constraints **without exception**.

---

7. Policy Decision Flow

Input caption + rule results
→ Identify violations and constraints
→ Detect conflicts
→ Apply priority order
→ Select resolution strategy
→ Approve or modify enhancements
→ Generate rendering instructions
→ Log decision

---

8. Decision Types

The Policy Engine may issue the following decision types:

- adjust_timing
- resegment_text
- reduce_line_count
- reposition_caption
- clamp_font_size
- suppress_enhancement
- downgrade_enhancement
- apply_fallback_mode
- merge_captions
- split_captions

---

9. Conflict Resolution

9.1 Conflict Definition

A conflict occurs when two or more constraints cannot be satisfied simultaneously.

9.2 Resolution Strategy

Conflicts must be resolved by:

1. Identifying all active constraints
2. Ordering them by priority
3. Preserving higher-priority constraints
4. Disabling or modifying lower-priority behaviours

9.3 Example

Conflict:
- Enhancement: emotion scaling
- Constraint: readability and accessibility (no reflow)

Resolution:
- Readability takes priority
- Emotion scaling suppressed

---

10. Profiles (Presentation Modes)

Profiles define allowable behaviour.

10.1 Required Profiles

Compliance / Broadcast

- Strict adherence to BBC rules
- No experimental enhancements

Accessible Enhanced

- Baseline compliance preserved
- Limited semantic enhancements allowed

Live / Low Latency

- Prioritises immediacy
- Allows relaxed segmentation rules

- Maintains readability floor

Immersive / Spatial

- Enables spatial positioning
- Requires motion constraints and fallback

Experimental / Expressive

- Allows advanced enhancements
- Must still respect compliance floor

---

11. Profile Configuration Structure

```
{
  "profile_id": "accessible_enhanced",

  "constraints": {
    "enforce_readability": true,
    "allow_partial_relaxation": false
  },

  "enhancements": {
    "emotion": "limited",
    "semantic_sound": true,
    "reactive_typography": "restricted"
  },

  "fallback": {
    "mode": "compliance"
  }
}
```

---

12. Enhancement Handling

12.1 Evaluation Process

Each enhancement must be:

Proposed → Validated → Approved / Modified / Rejected

12.2 Outcomes

Outcome	Description
approve	safe to apply
modify	reduced intensity
suppress	rejected due to constraints

12.3 Example

Enhancement: volume-based scaling
Constraint: layout stability

Decision:
→ scaling reduced to prevent reflow

---

13. Fallback Behaviour

Fallback must be triggered when:

- Constraints cannot be satisfied
- Data is incomplete or unreliable
- Enhancements introduce instability

13.1 Fallback Modes

- compliance_mode
- simplified_layout
- static_positioning
- no_enhancements

Fallback behaviour must preserve higher-priority constraints even if it reduces feature richness.

---

14. Low Confidence Handling

When input confidence is low:

- enhancements must be reduced or disabled
- conservative rendering must be used
- fallback behaviour may be triggered

---

15. Integration with Renderer

The Policy Engine must output:

- final text segmentation
- timing adjustments
- layout constraints
- positioning instructions
- styling limits

The renderer must **not override policy decisions**.

---

16. Integration with Decision Logs

Each policy decision must log:

- constraints considered
- conflicts detected
- priority applied
- decision outcome
- reasoning

---

17. Testing Requirements

Tests must verify:

- correct priority application
- correct conflict resolution
- correct profile behaviour
- correct enhancement gating
- correct fallback triggering

---

18. Claude Integration

Claude can use the Policy Engine to:

- simulate decision outcomes
- identify incorrect conflict resolution
- suggest improvements
- validate profile definitions

---

19. Future Extensions

- adaptive profiles (user-specific)
- context-aware policies (scene-dependent)
- learning-based optimisation
- user feedback loops

🔚 **Summary**

The Policy Engine:

- Resolves all conflicts
- Enforces system constraints
- Controls configurability through profiles
- Governs all enhancement behaviour
- Produces final instructions for rendering