Renderer Specification v1

1. Purpose

The Renderer is responsible for producing the **final visual output of captions** based on instructions from the Presentation Policy Engine.

It executes:

- Text layout and segmentation
- Typography and styling
- Positioning and safe-area compliance
- Timing and synchronisation
- Motion and transition behaviour

The Renderer must **strictly follow Policy Engine instructions** and must not independently modify caption behaviour.

---

2. Scope

The Renderer is responsible for:

- Rendering caption text on screen
- Applying layout and line-breaking decisions
- Enforcing typography constraints
- Positioning captions within safe regions
- Executing timing and display duration
- Handling transitions and updates
- Applying approved enhancements

The Renderer is **not responsible for decision-making**.

---

3. Core Design Principles

Rendering must not depend on non-deterministic factors such as frame timing jitter or external UI state.

3.1 Policy Authority

The Renderer must execute instructions exactly as provided by the Policy Engine.

3.2 Deterministic Output

Given the same input and instructions, the Renderer must produce identical output.

3.3 Layout Stability

Rendering must avoid unnecessary movement, reflow, or flicker.

3.4 Safe Area Enforcement

Captions must remain within defined safe regions at all times.

3.5 Accessibility Preservation

Rendering must preserve readability, contrast, and visual clarity.

---

4. Inputs

The Renderer receives:

- Final caption text (segmented into lines)
- Start and end times
- Layout instructions
- Positioning coordinates or regions
- Styling parameters (font size, colour, etc.)
- Enhancement instructions (if approved)

---

5. Outputs

The Renderer produces:

- On-screen captions
- Frame-by-frame rendering updates
- Rendering-related decision logs

---

6. Layout System

6.1 Line Count

- Maximum lines must follow policy constraints (typically 2 lines)

6.2 Line Length

- Must respect readability thresholds
- Must avoid overflow and clipping

6.3 Line Breaking

- Must follow segmentation provided by Policy Engine
- Must not perform uncontrolled automatic wrapping

6.4 Alignment

- Default: centred or left-aligned based on profile
- Must remain consistent within a sequence

---

7. Typography

7.1 Font Size

- Must not exceed authored size
- May be reduced if required for layout safety

7.2 Font Style

- Must follow profile and policy rules
- Italics and emphasis must be applied only when approved

7.3 Colour

- Must preserve contrast

- Must respect speaker colour rules
- Must not conflict with readability constraints

---

8. Positioning

8.1 Safe Region

Captions must remain within defined safe areas:

- Avoid faces and mouths
- Avoid UI overlays
- Avoid clipping at edges

8.2 Default Position

- Bottom-centre (or profile-defined)

8.3 Dynamic Positioning

- Only allowed when explicitly enabled (e.g. immersive mode)
- Must include motion constraints

8.4 Vertical Adjustment

- Must shift upward if lower region is obstructed
- Must maintain consistent positioning when possible

---

9. Timing and Synchronisation

9.1 Display Timing

- Must match Policy Engine output
- Must not anticipate or lag beyond defined thresholds

9.2 Frame Accuracy

- Rendering must align with frame timing

9.3 Update Behaviour

- Caption updates must be smooth
- Avoid flicker or abrupt changes

---

10. Motion and Transitions

10.1 Stability

- Captions must remain stable unless movement is required

10.2 Movement Constraints

- Motion must be minimal and controlled
- Must not impair readability

10.3 Transition Types

Allowed transitions include:

- fade-in / fade-out
- position shift (limited)
- progressive reveal (if approved)

---

11. Enhancement Rendering

11.1 Application

Enhancements must only be applied if:

- approved by Policy Engine
- within defined constraints

11.2 Types

- emotion-based styling
- semantic sound indicators
- reactive typography (limited)
- spatial positioning

11.3 Constraints

Enhancements must not:

- cause reflow
- reduce readability
- introduce instability
- conflict with speaker clarity

---

12. Reflow Prevention

12.1 Definition

Reflow occurs when text layout changes unexpectedly during display.

12.2 Requirements

- Must not reflow during active display
- Scaling must be pre-calculated
- Layout must remain stable

---

13. Collision Handling

13.1 Detection

The Renderer must detect:

- overlap with UI elements
- overlap with important visual content

13.2 Resolution

- reposition caption
- reduce size
- apply fallback positioning

---

14. Fallback Rendering

Fallback must be triggered when:

- layout cannot be safely resolved
- enhancements introduce instability
- constraints cannot be satisfied

14.1 Fallback Modes

- static bottom-centre
- reduced font size
- no enhancements

---

15. Integration with Policy Engine

The Renderer must:

- accept all instructions from Policy Engine
- not override or reinterpret decisions
- log any rendering constraints encountered

The Renderer must not introduce any behaviour not explicitly defined by the Policy Engine, even if such behaviour would improve layout or aesthetics.

---

16. Integration with Decision Logs

The Renderer must log:

- final layout decisions
- positioning adjustments

- enhancement application
- fallback triggers

---

17. Performance Requirements

- Must support real-time rendering
- Must avoid frame drops
- Must maintain consistent timing accuracy

---

18. Testing Requirements

Tests must verify:

- layout correctness
- timing accuracy
- stability (no flicker or reflow)
- safe region compliance
- correct enhancement rendering

---

19. Claude Integration

Claude can use Renderer outputs to:

- analyse layout correctness
- detect visual conflicts
- validate enhancement safety
- suggest improvements

---

20. Future Extensions

- multi-display rendering (WallSpace integration)
- projection mapping support
- VR / AR spatial rendering
- adaptive typography

Renderer Specification v1

🔚 Summary

The Renderer:

- Executes all visual behaviour
- Enforces layout and display constraints
- Applies approved enhancements
- Maintains stability and readability
- Produces the final user-facing output