ARCHITECTURE NOTES · RESOURCE

The four signals every AI content program should log from day one

A diagnostic companion. The four signals that tell you which decision broke when a draft underperforms, named in order: brief completeness, voice gate findings, quality reviewer score, reviewer behaviour.

Companion to: The four signals every AI content program should log from day one.

When a draft is bad, the operator needs to know inside thirty seconc

Was the brief thin. Did the voice gate catch it. Did the model regress. Did the reviewer skip a step. The article makes the point that generic LLM logging answers none of those questions. It tells you which API call returned what. The four signals tell you which brand decision broke. This deck walks each signal in working form, with the failure mode it catches and the action it implies.

- For content leads and AI program owners past the first thirty days of a rollout.
- Each signal maps one-to-one to a decision in the production pipeline.
- Without the four signals, the operator is swapping models and praying.
- No invented metrics. The article is short. This deck stays inside it.

The four signals, in the order the article ranks them

01

1. Brief completeness

Score every brief against the schema before the draft runs. The article frame: a 7 of 10 brief produces a 7 of 10 draft. Brief score is the first thing the operator should see when a draft underperforms.

02

2. Voice gate findings

Every output is scanned for forbidden vocabulary, brand-direction violations, and AI-pattern leaks. The findings are logged. A voice gate that passed a draft that should have failed is a signal in itself.

03

3. Quality reviewer score

The score the reviewer assigned to the output, plus the rationale. The number on its own is not enough. The trend across drafts is what carries the signal.

04

4. Reviewer behaviour

Note length, time-per-review, rejection rate. The signal that catches reviewer fatigue. A reviewer skipping a step shows up here before it shows up anywhere else.

Generic LLM logging versus the four signals

What generic LLM logging tells you

- Which API endpoint was called.
- Which model version handled the request.
- How many tokens went in and came out.
- How long the request took.
- Whether the call succeeded or errored.

What the four signals tell you

- Whether the brief that fed the draft was complete.
- What the voice gate caught and what it let through.
- How the reviewer scored the output and why.
- Whether the reviewer is still doing the work or fatigued.
- Which decision broke when the draft underperforms.

The diagnostic chain, signal by signal

01 **Brief score below threshold**
If brief completeness is the failing signal, the work is upstream of the model. Fix the brief schema before touching anything else.

02 **Voice gate passed a bad draft**
If the voice gate let through an output that should have been caught, the gate needs recalibration against the current model behaviour.

03 **Voice gate caught a draft the reviewer approved**
If the gate flagged something and the reviewer overrode it, the contract between gate and reviewer is unclear. Tighten the rule.

04 **Reviewer score dropping over time**
A reviewer score curve that bends downward is a model regression or a brief decay signal, not necessarily a reviewer signal.

05 **Reviewer behaviour shifting**
Shorter notes, faster reviews, lower rejection rate. The signal that catches reviewer fatigue before the program collapses.

Wiring the four signals from day one

- **Score every brief against the schema before the draft runs.**
  Log the score and the missing fields. Make it visible to the operator on the draft page.
- **Run every output through the voice gate scanner.**
  Log the findings: forbidden vocabulary, brand-direction violations, AI-pattern leaks. Pass and fail are both logged.
- **Capture the quality reviewer score and rationale.**
  A number alone is not enough. The rationale is what carries the diff across weeks.
- **Track reviewer behaviour as a first-class signal.**
  Note length, time-per-review, rejection rate. Trended weekly.
- **Surface all four on a single draft-level view.**
  When a draft underperforms, the operator should see the four signals together, not in four different tools.
- **Re-baseline the voice gate after every model release.**
  Scanners catch what they were calibrated against. A new model release shifts the baseline.

Anti-patterns to avoid when instrumenting

What goes wrong

- Logging only at the API layer and calling it instrumentation.
- Capturing reviewer score but not reviewer behaviour.
- Treating the voice gate as a binary pass-fail with no findings log.
- Scoring briefs without surfacing the score on the draft page.
- Re-using the same voice gate calibration after a model release.

What works instead

- Log at the brand-decision layer, not the API layer.
- Trend reviewer note length, time, and rejection rate weekly.
- Log every voice gate finding, not just the verdict.
- Surface brief score on the draft page so the operator sees it first.
- Re-baseline the voice gate against current model behaviour.

Signals to watch on a working content program

BRIEF COMPLETENESS

Schema-scored

Average score across the last thirty briefs. Trend it weekly.

VOICE GATE FINDINGS

Logged + trended

Pass rate, forbidden-term hits, AI-pattern leaks. Tracked across model releases.

QUALITY REVIEWER SCORE

Scored + rationale

Number alone is not enough. The rationale carries the diff over time.

REVIEWER BEHAVIOUR

Length / time /

The signal that catches reviewer fatigue first.

NEXT STEP

Without the four signals, you are swapping models and praying

Wire brief completeness, voice gate findings, quality reviewer score, and reviewer behaviour from day one. Then when a draft underperforms, the operator knows

Read the full architecture note ->