WORK SAMPLE, PREPARED FOR THIS POSTING

First Narrow Workflow: Medical Necessity Documentation Check

A pre submission reviewer for ambulance claims. Written before any access to your systems, so every assumption below is explicit and meant to be corrected by your billing team.

> Before a claim is submitted, read the narrative documentation attached to the transport and report which elements required to support medical necessity are present, which are missing, and quote the exact passage that supports each finding. It recommends. It does not code, price, alter or submit anything.

WHY THIS ONE FIRST

Insufficient documentation of medical necessity is consistently reported as the leading root cause of ambulance claim denials, and it is the one that lives in free text rather than in structured fields, so deterministic edits cannot reach it. It is also low blast radius: the output is a checklist shown to a biller, not a change to a code, a charge or a submission.

Three properties make it the right first target: high denial impact, genuine language ambiguity, and a label source that already exists.

DETERMINISTIC, NOT A MODEL

- EDI syntax and implementation guide conformance
- Valid code sets and situational rules
- Origin and destination modifier pairing
- Mileage arithmetic and units
- Signature and certification presence, and date windows
- Timely filing clocks
- Eligibility transactions themselves

If a rule can be written down and tested, it should be code. Putting a model on top of a solved problem adds cost, latency and a new failure mode.

WHERE A MODEL EARNS ITS PLACE

- Extracting evidence of medical necessity from the crew narrative
- Detecting what the documentation is missing relative to what the billed level of service requires
- Flagging contradictions between the narrative and the structured record
- Later: denial root cause classification and appeal drafting grounded in the same source text

GROUNDING

The only admissible source is the documentation attached to that transport, plus the payer rules your team approves as reference material. No open retrieval, no general web knowledge, no reasoning from other patients.

OUTPUT CONTRACT

Strict schema, validated before anything is written. Each finding carries:

FIELD	PURPOSE
`element`	Which required element, from a closed list
`status`	present, missing, ambiguous
`quote`	Verbatim passage from the source
`locator`	Where in the document it was found
`confidence`	Calibrated, not self reported

Hard gate: if the quoted text does not appear verbatim in the source document, the finding is discarded before a human ever sees it. A claim of evidence that cannot be located is not a low confidence answer, it is a fabrication, and it is dropped deterministically rather than scored.

CONTROLS

- **Human review is structural.** Output is a suggestion attached to the claim. Nothing changes state without a biller acting on it.
- **Confidence is calibrated.** Self reported model confidence is stored but never used as a threshold until the bucketed curve proves it predicts correctness.
- **Idempotency by content.** Re running on an unchanged document returns the stored result instead of paying for a second opinion that may differ.
- **Full provenance.** Which document version, which prompt and workflow version, which model, which retrieved passages, and the resulting findings, in an append only record. A log you can rewrite is not evidence.

HOW ACCURACY GETS MEASURED

Backtest against remittances. Historical claims already carry their outcome and, when denied, the coded reason. Replay the reviewer against the record as it stood before submission and ask whether it would have caught the denial that happened.

Name the bias. That set only contains claims that were submitted. Everything a biller fixed silently beforehand is invisible, and that is the population this workflow serves, so it is paired with a prospective set captured during real review.

Score per element, never globally. Precision and recall for each required element separately. An average hides a detector that is blind.

Shadow first. It scores without blocking anything, to obtain agreement rate against expert judgment and realistic alert volume per biller per day before any process changes.

TRIAL SUCCESS CRITERIA, AGREED BEFORE BUILDING

1. On a held out set of expert reviewed transports, recall on missing medical necessity elements above a threshold your billing team sets, with false flags per claim low enough that reviewers keep reading them. Both numbers are chosen by them, not by me.

2. Every finding traceable to a verbatim passage in the source document. Zero unlocatable citations, enforced by the gate rather than by inspection.

3. A frozen regression set that replays on every prompt, model or retrieval change, with a per element diff, so an upgrade cannot quietly degrade one detector while improving another.

4. Cost and latency per reviewed claim measured from the first run, not estimated afterwards.

WHAT THIS WORKFLOW DELIBERATELY DOES NOT DO

It does not select or modify codes or modifiers. It does not compute or alter charges. It does not decide eligibility. It does not submit, hold or release a claim. It does not read across patients. It has no general database access: it receives the documents for one transport and returns findings about that transport. Expansion into coding suggestions or denial classification happens only after this one is measured in production, and each expansion repeats this same cycle.

Assumptions here are mine and are meant to be challenged by the people who bill these claims every day. The first thing I would ask your billing team is to define what a correct output looks like for this workflow, before I write any code, because their judgment is the ground truth and mine is not.