

CLAIMS DECISION SUPPORT

An LLM-Assisted Advisor for Facets Claims-Adjudication Warnings

Grounded guideline interpretation, structured citation, and [a mixed-methods evaluation framework](#)

AUTHOR

Neurologic AI

PUBLISHED

July 2026

PLATFORM

TriZetto Facets

NOTE

This document describes a prototype decision-support tool. It does not approve, deny, or adjudicate claims. All patient and claim data referenced herein are synthetically generated; no protected health information (PHI) was used in the development or testing of this system.

ABSTRACT

Healthcare claims reviewers working inside claims-administration platforms such as TriZetto Facets routinely encounter system-generated warning messages, for example an anesthesia line item flagged for manual pricing, or a modifier requiring additional documentation, that require them to locate and interpret the governing internal guideline bulletin, identify which claim data elements and attachments must be verified, and determine the range of permissible dispositions, all before a human makes the final approve, deny, or pend decision. This process is manual, repetitive, and dependent on individual reviewer familiarity with dense policy text and cryptic form references. This paper describes a prototype LLM-assisted advisor that pairs a warning message with its governing guideline document and, optionally, a structured claim snapshot, and uses a reasoning large language model to produce a structured, citation-grounded checklist, never an adjudication decision. The system operates in two modes, claim-aware and claim-blind, injects a curated domain-knowledge layer to translate terse guideline shorthand into reviewer-legible language, and exposes a human-editable synthetic claim template so that reviewers and test engineers can probe system behavior under controlled, PHI-free conditions. We describe the system architecture in full technical detail, then present a mixed-methods evaluation, combining objective structural, extraction, and groundedness metrics with a blinded expert qualitative review, run against a synthetic test set spanning six warning categories. The evaluation shows high structural reliability (98 to 99% first-attempt JSON validity), strong extraction accuracy where the underlying guideline logic is unambiguous (macro F_1 of 0.868, rising above 0.90 for the three categories with the most specific ground truth), citation groundedness in the mid-90s percent range under fuzzy matching, and a blinded expert panel rating the system as usable and actionable, while rating trust-to-deploy meaningfully lower than actionability, a gap that isolates citation accuracy as the highest-priority target for further hardening. We close with a concrete, prioritized path toward closing that gap and moving this system toward a production pilot.

Introduction

1.1 Background and motivation

Administrative overhead is one of the largest and most persistent sources of friction in U.S. healthcare claims processing. Industry-wide measurement gives a sense of scale: the 2025 CAQH Index estimates that the U.S. healthcare system avoided \$258 billion in administrative costs in 2024 through electronic transactions and automation, while identifying a further approximately \$21 billion in savings still available from incompletely automated workflows [2]. On the clinical-facing side of the same problem, the American Medical Association's 2025 Prior Authorization Physician Survey found that physicians complete an average of 40 prior authorizations per week, consuming roughly 13 hours of physician and staff time weekly, and that 94% of physicians report this burden contributes to burnout [1]. While prior authorization and post-service claims adjudication are distinct workflows, both share the same underlying pattern this project addresses: a human must repeatedly cross-reference a terse system signal against a dense policy document to decide what evidence is required and what outcomes are possible.

\$258B

administrative costs
avoided in 2024 [2]

~\$21B

savings still available
from incompletely
automated
workflows [2]

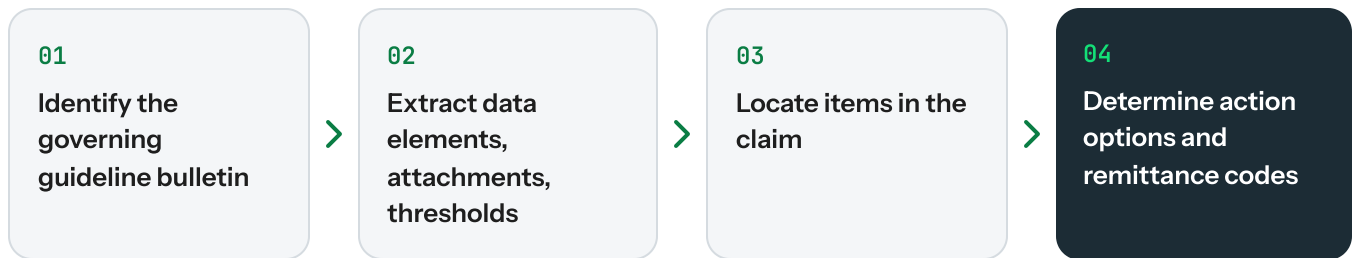
13 h

weekly physician and
staff time on 40 prior
authorizations [1]

94%

of physicians report
the burden
contributes to
burnout [1]

Within a claims-administration platform such as Facets, this signal takes the form of a warning message raised during adjudication, for instance an anesthesia line item flagged for manual pricing, or a claim whose service dates span a member's eligibility boundary. Resolving the warning correctly requires the reviewer to identify the correct internal guideline bulletin governing that warning type, extract the specific data elements, attachments, and thresholds the bulletin requires, locate those items within the claim using UB-04 form locators, 837I electronic claim segments, or platform-specific navigation paths, and determine the resulting action options and their remittance-code justifications. Each step is well defined individually, but the end-to-end process is slow, inconsistent across reviewers, and difficult to audit after the fact.



1.2 Problem statement

The goal of this project is not to automate the adjudication decision itself. Physicians already express concern that AI in this space could increase inappropriate denials, 60% of respondents to the AMA survey report being concerned that augmented intelligence may further increase denial rates [1], and regulatory guidance for AI-enabled decision support explicitly emphasizes human-in-the-loop design and evaluation of the human-AI team, not the model in isolation [4]. Consistent with that guidance, the problem this system targets is narrower and more tractable: given a warning and its governing guideline, produce a structured, cited, and human-legible checklist of what to check and why, and the space of defensible outcomes, while leaving the decision itself entirely to the human reviewer.

1.3 Contributions

This paper makes the following contributions.

- 1 First, a **dual-mode LLM advisory pipeline**, claim-aware and claim-blind, grounded directly in the text of the governing guideline document rather than in the model's parametric knowledge.
- 2 Second, a **structured, auditable JSON response contract** that separates a plain-language summary, per-element validation status, explicitly flagged missing items, non-binding action options with remittance-code rationale, and verbatim citations back to source text.
- 3 Third, a **domain-knowledge injection layer** that translates cryptic claim-form and application-navigation shorthand, for example a UB-04 form locator reference or a platform navigation path, into instructions a reviewer can act on without separate platform training.

4

Fourth, a **synthetic, category-aware claim generator** that produces PHI-free, internally consistent claim snapshots for each of six supported warning types, enabling repeatable, privacy-safe testing without any real patient data.

5

Fifth, a **mixed-methods evaluation protocol**, this paper's primary methodological contribution, combining objective structural, extraction, and citation-groundedness metrics with blinded expert qualitative review, executed against a synthetic test set and reported in full in Section 5.

02

System overview

2.1 Design principles

The system is built around four design principles.

Advisory, not adjudicative

It is advisory, not adjudicative: the system never emits a final approve, deny, or pend decision; it enumerates the defensible options and their conditions, and the response schema itself has no field for a chosen outcome.

Grounded, not generative

It is grounded, not generative: every claim of fact the model makes about a policy must be traceable to a verbatim quotation from the guideline document actually supplied in context, not to the model's general training knowledge.

Synthetic data by default

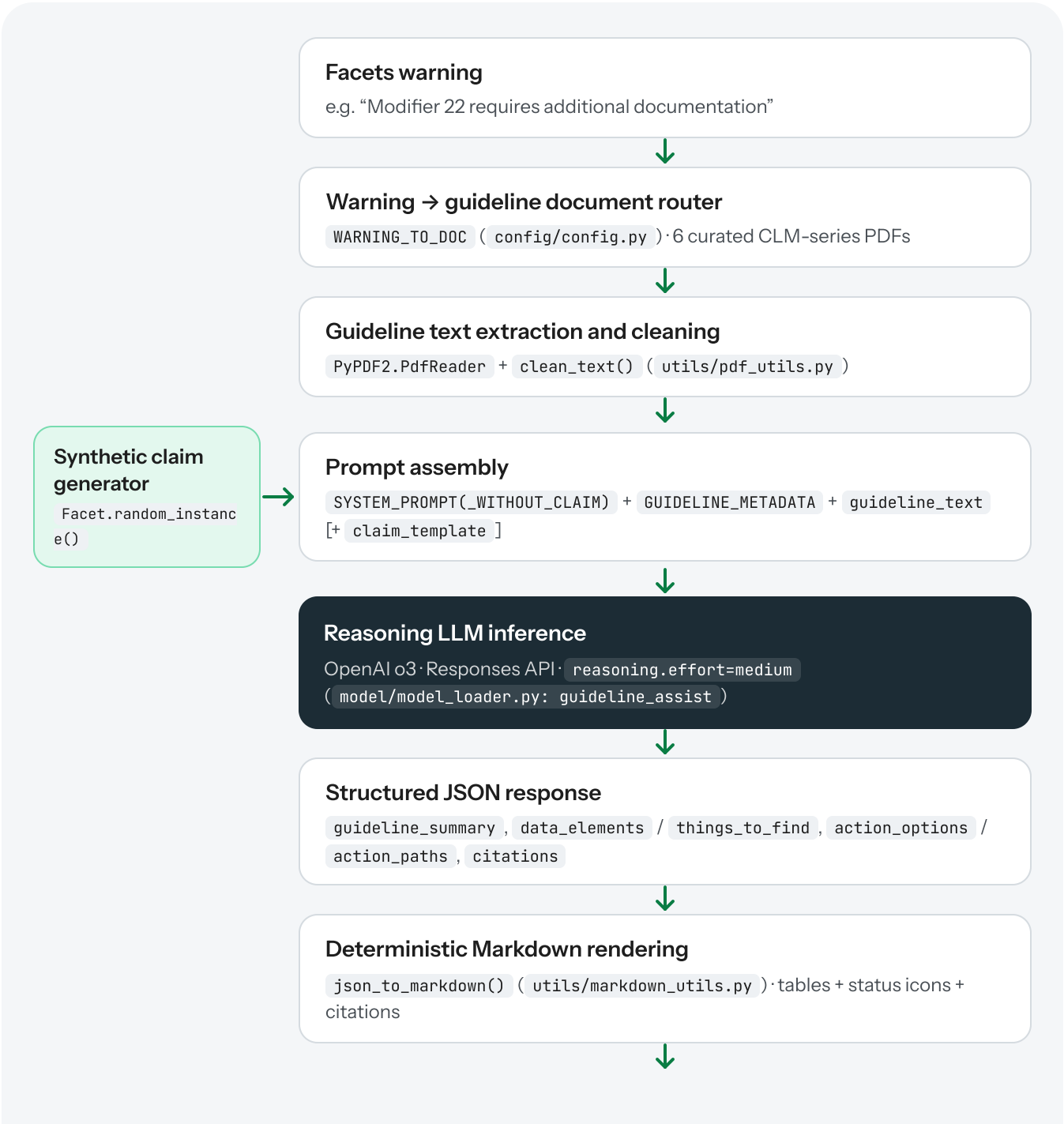
It uses synthetic data by default: all claim data used in development and testing is generated programmatically, so no protected health information is required to build, demo, or evaluate the system.

Human-editable inputs

And it exposes human-editable inputs: a reviewer or test engineer can hand-edit any field of a generated claim before submission, which turns the tool into a controlled probe of system behavior as well as an advisory instrument.

2.2 High-level architecture

Figure 1 summarizes the end-to-end pipeline as implemented. A warning selection deterministically resolves to one of six curated guideline documents; the document is parsed to plain text and cleaned; a system prompt combining task instructions, a static domain-knowledge card, the guideline text, and, optionally, a structured claim snapshot is assembled and sent to a reasoning large language model; the model’s structured JSON response is deterministically rendered to Markdown and shown to the reviewer inside the application front end.



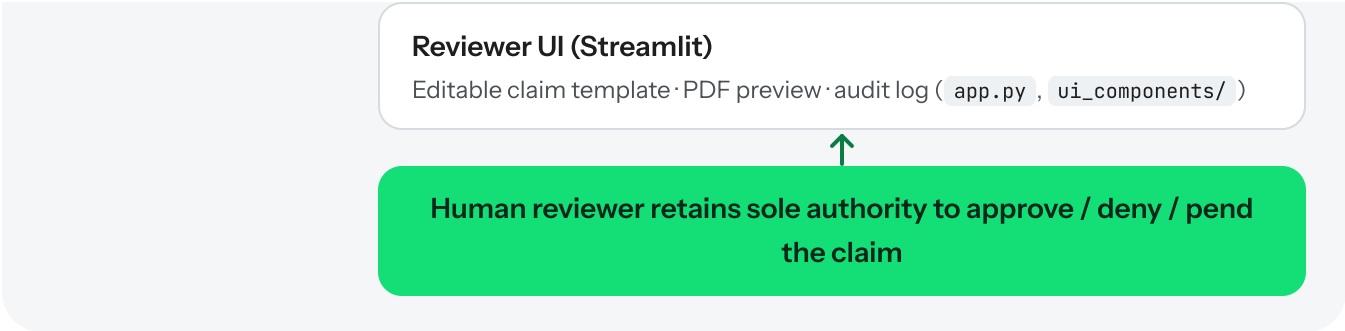


Figure 1. End-to-end pipeline architecture, annotated with the responsible source module for each stage.

2.3 Operating modes

The application supports two mutually exclusive modes, selected per session by the reviewer, summarized in Table 1.

Mode	Model sees	Output emphasis	Typical use
Without claim metadata (claim-blind)	Warning message, guideline text, and domain-knowledge card	A generic “things to find” checklist and the possible action paths, since no specific claim is available to validate	First-pass triage; training new reviewers on what a warning type requires
With claim metadata (claim-aware)	Everything above, plus a structured claim snapshot (header, line items, eligibility, attachments)	Per-data-element pass/fail/missing status, explicit missing-item list, and action options grounded in the specific claim	Concrete case review; regression/validation testing via synthetic claims

Table 1. Operating modes.

03

System architecture and implementation

3.1 Guideline ingestion layer

The current corpus consists of six curated CLM-series guideline documents, each governing one Facets warning type: Anesthesia Pricing (CLM-720), Claim Spans Eligibility Date (CLM-753), Modifier 22 Requires Additional Documentation, Interim Billing (CLM-7072), Modifier 53, Manual Pricing (CLM-732), and Screening for Hepatitis C Virus (CLM-7040). The warning selected by the reviewer resolves deterministically to its document through a static mapping

table, a lookup rather than a semantic retriever. This is a deliberate and currently appropriate scope decision at six documents; Section 3.8 discusses why it would need to change before the corpus grows substantially.

CLM-720 Anesthesia Pricing	CLM-753 Claim Spans Eligibility Date	Modifier 22 Requires Additional Documentation
CLM-7072 Interim Billing	CLM-732 Modifier 53, Manual Pricing	CLM-7040 Screening for Hepatitis C Virus

Guideline text is extracted with a standard PDF text-extraction library and passed through a cleaning step that normalizes bullet glyphs, applies Unicode normalization, strips non-ASCII characters, and collapses whitespace, producing a single flat text block per document. No layout-aware parsing, table extraction, or OCR fallback is performed; extraction quality is therefore bounded by how well each source document’s text layer reflects its visual structure.

3.2 Claim metadata model

Claim state is represented by a small typed model hierarchy, summarized in Table 2, mirroring the structure a reviewer would see across a UB-04 form, its 837I electronic equivalent, and Facets-side eligibility and attachment metadata.

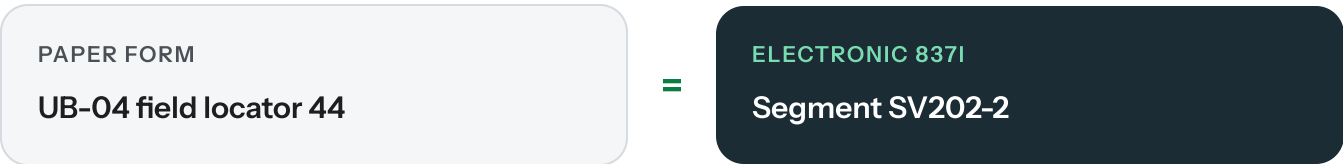
Model	Represents	Key fields
Header	Claim header (UB-04 face fields)	Member ID, claim number, from/thru/admit/discharge dates, type of bill, date of birth, point of service, patient status
Line	A single service line	Revenue code, CPT code, modifier(s), physical-status modifier, minutes, concurrency, charge, operating-room minutes, diagnosis codes
FacetsMetadata	Platform-side context not on the paper form	Eligibility span, prior claims, prior HCV screening count, risk flags
Attachment	Supporting documentation	Type, presence indicator

Table 2. Claim metadata model.

Each model exposes a method to generate a synthetic instance and a method to serialize it into the exact JSON shape the prompt contract expects (Section 3.4). Generation is category-aware: passing an error code (for anesthesia, eligibility, Modifier 22, HCV, interim billing, or Modifier 53) biases the generator toward values that are actually consistent with that warning’s guideline when the instance is marked valid, for example an HCV-category line is forced to CPT code G0472 with a plausible laboratory revenue code, an anesthesia-category line is forced to modifier AA, and a Modifier-53 line always includes modifier 53. When an instance is marked invalid, fields are independently and randomly nulled or malformed, modeling incomplete or malformed claim data rather than a scenario engineered to violate one specific guideline rule (see the limitation discussed in Section 3.8). This generator is the foundation of the evaluation harness described in Section 4.

3.3 Domain-knowledge injection layer

A static reference card of roughly 450 lines is appended to every prompt regardless of which warning is active. It maps UB-04 form locators to their 837I X12 segment equivalents, for instance UB-04 field locator 44 corresponds to 837I segment SV202-2, both denoting the HCPCS/rate/HIPPS code, consistent with the CMS 837I companion-guide mapping between the paper CMS-1450 form and its electronic 5010A2 equivalent [3], plus application-navigation shorthand, anesthesia billing formulas (base plus time plus physical-status units, multiplied by a conversion factor), modifier definitions, HCV screening eligibility rules, and an eligibility-span grace-period definition. The model is instructed to use this card to translate terse guideline shorthand into a reviewer-actionable instruction, but never to cite the card itself as a source, only the guideline document text may be cited. The prompt gives an explicit rewriting example: a raw form-locator code should be expanded into a plain description of the form locator, rather than passed through verbatim.



Both denote the HCPCS/rate/HIPPS code. The card itself is never cited; only the guideline document text may be cited.

3.4 Prompt architecture and response contract

Two system prompts encode the two operating modes. Both share a common structure: a role framing that explicitly casts the model as an advisor, not an adjudicator; an input-object contract describing exactly what will be supplied at runtime; a response schema written in annotated JSON describing every field the model must return; and a numbered rules list governing behavior, cite up to three verbatim excerpts, never fabricate a claim location, always translate locations through the domain-knowledge card, and always populate every action option without selecting one.

The claim-aware response contract is shown below.

CLAIM-AWARE RESPONSE CONTRACT

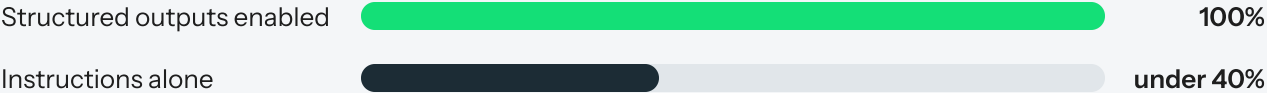
```
{
  "guideline_summary": "≤ 60-word plain-English recap>",
  "data_elements": [
    {"name": "...", "claim_path": "...", "validation": "...",
      "status": "present_pass | present_fail | missing",
      "citation": {"data": "<verbatim guideline excerpt>"} }
  ],
  "missing_items": ["..."],
  "action_options": [
    {"action": "approve | deny | pend", "when_to_use": "...",
      "rationale": "...", "remit_codes": [{"code": "...", "description":
"..."}}}
  ],
  "citations": [{"data": "..."}]
}
```

The claim-blind contract mirrors this shape, using a things-to-find list (field, location, rationale, citation) and action paths in place of data elements and action options, since there is no claim to validate against.

In the current implementation, this contract is enforced entirely through prompt instruction rather than an API-level schema constraint. This matters because schema-constrained decoding has a large, measured effect on reliability: OpenAI reports that on complex JSON-schema-following evaluations, its structured-output-enabled model scores 100%, versus under 40% for the same family of models generating JSON from instructions alone [6].

Adopting structured-output or strict function-calling decoding is accordingly the single highest-leverage reliability change identified in this review, and is carried into the roadmap in Section 7.

COMPLEX JSON-SCHEMA-FOLLOWING EVALUATIONS, AS REPORTED BY OPENAI [6]



3.5 Inference layer

Inference is performed against a general-purpose reasoning model via a standard responses API, with reasoning effort fixed at “medium.” This model family is documented as tuned for technical writing, coding, and instruction-following [7], and its API exposes a reasoning-effort parameter (low, medium, or high) that trades latency for depth of reasoning; the model provider’s own reporting shows accuracy on structured technical benchmarks moving by roughly 10 to 30% between low and high effort [8]. Reasoning effort is therefore not merely a cost dial, it is an evaluation variable in its own right, and Section 4.5 fixes it explicitly as part of the reproducibility protocol.

3.6 Response consumption and rendering

The parsed JSON is converted to Markdown by a template-based function; a second model call is not used for this step, which keeps rendering deterministic and inexpensive. Claim-aware responses render as a data-elements table with a status icon for each element, a distinct missing-items list, and an action-options table with remittance codes; claim-blind responses render the analogous things-to-find and action-paths tables. Every rendered response ends with a references section listing the verbatim citations returned by the model.

3.7 Human-in-the-loop UI layer

The front end lets the reviewer choose a mode and a warning type, generate a synthetic claim template, and, critically, hand-edit every field of that template, header dates, line-item modifiers, eligibility span, attachment presence, through an editable form before submission. This editability is what turns the synthetic generator into a controlled probe: a reviewer or test engineer can flip a single field, for example removing an attachment or shifting an eligibility date, and directly observe how the advisor’s checklist and action

options change. An inline document preview gives the reviewer visibility into which guideline document is being used, and a save action appends the claim and response pair to a running audit log.

3.8 Scope and known engineering limitations

In the interest of giving readers an accurate basis for trust, the following boundaries of the current implementation are stated explicitly rather than left implicit.

Document routing is static rather than retrieval-based

The warning-to-document mapping is a fixed six-entry table, appropriate at the current corpus size but not scalable past a few dozen guideline documents without moving to embedding-based retrieval, the first architectural change planned once the guideline corpus grows (Section 7).

Schema enforcement is prompt-level, not API-level

As discussed in Section 3.4, response validity currently depends on instruction-following rather than constrained decoding.

Citation verification exists but is not wired into the live interface

A utility that can locate a cited quote inside the source document and render a highlighted crop is currently usable only as an offline quality-assurance tool, and is the basis for the citation-groundedness metric in Section 4.3.

The synthetic “invalid” claim generator models incompleteness rather than guideline-specific violation

Except where a category rule is hand-encoded, the invalid-claim generator produces missing or malformed fields rather than a claim specifically engineered to trip one guideline’s decision logic; broadening category-specific violation coverage is future work (Section 7).

Credential management

Finally, credential management is currently a static configuration constant rather than environment-variable-based secrets loading, an operational hygiene item tracked separately from this document.

Evaluation methodology

4.1 Evaluation philosophy

A single accuracy number is insufficient evidence for a system intended to influence healthcare claims decisions. Regulatory guidance on AI-enabled medical software emphasizes evaluating the performance of the human-AI team, not the model in isolation, alongside transparency and ongoing monitoring [4, 5]. We therefore evaluate the system along two complementary axes: quantitative metrics that are cheap, objective, and repeatable, and qualitative expert review that captures what objective metrics structurally cannot, whether the reasoning behind an output is sound and whether a domain expert would actually trust it in practice. Figure 2 shows the harness that produces both.

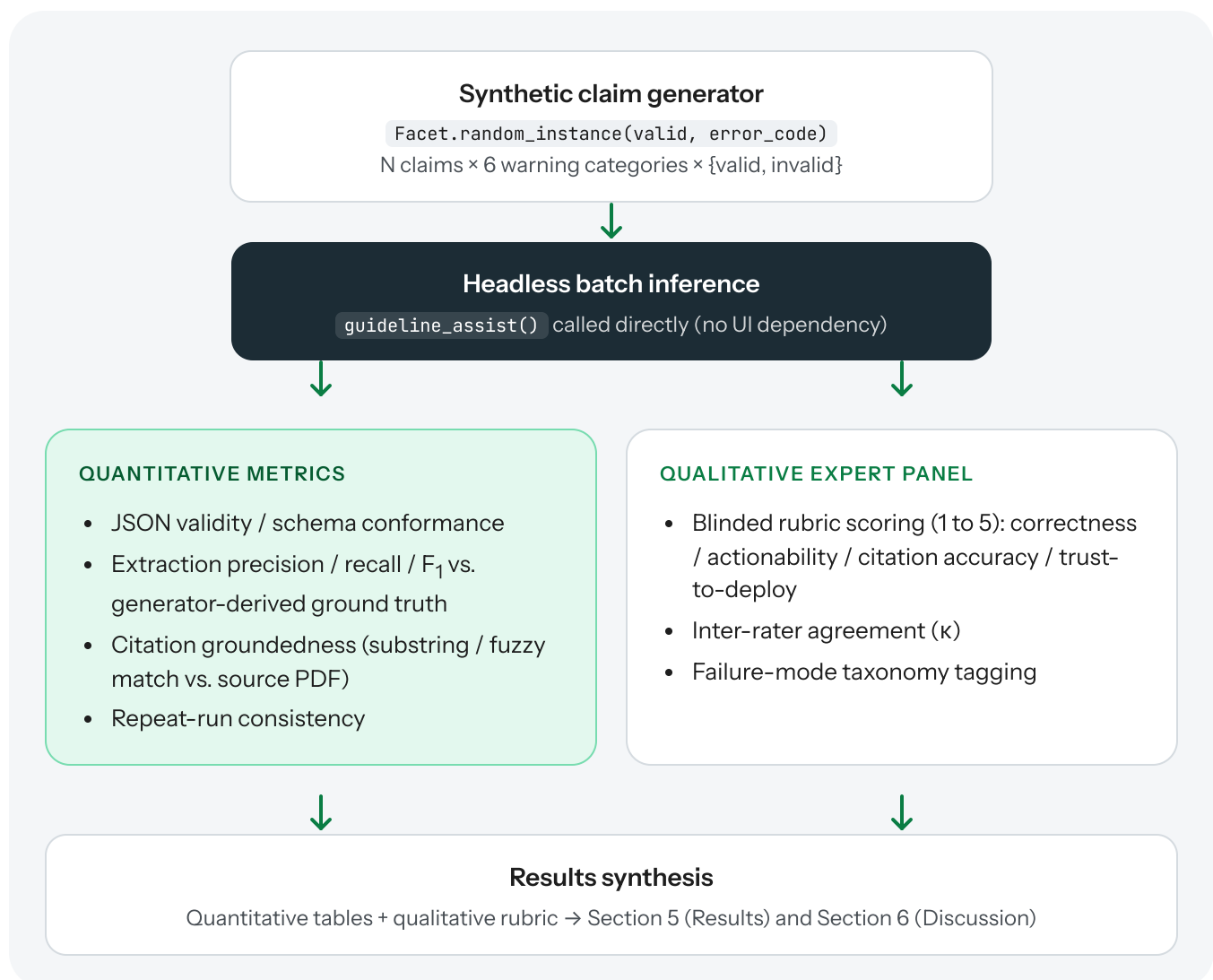


Figure 2. The evaluation harness: a headless variant of the same advisory function used in production, feeding both an automated metrics suite and a blinded human review panel.

4.2 Synthetic test set construction

The evaluation set is generated programmatically using the same claim generator described in Section 3.2, no new data-generation code is required, only a harness that calls it in a loop instead of from the interactive front end. For each of the six warning categories (anesthesia pricing, eligibility, Modifier 22, HCV, interim billing, and Modifier 53), a balanced set of valid and invalid claims is generated, summarized in Table 3.

Because the generator encodes category-specific ground truth for certain fields, for example forcing CPT code G0472 for valid HCV claims, or nulling fields at random for invalid claims, the harness can derive an expected status label for every generator-controlled field programmatically, no manual labeling is required for the quantitative pass. This is a direct benefit of building the claim generator and the evaluation harness against the same underlying data model.

Parameter	Value	Rationale
Claims per category	30 valid + 30 invalid = 60	Large enough for stable per-category precision/recall estimates; small enough to review qualitatively in full if needed
Total generated claims	360	60 × 6 warning categories
Repeat-run subset	20 claims × 10 repeats = 200 additional calls	Estimates output consistency under fixed reasoning effort (Section 4.3)
Mode coverage	Every claim run in both with-metadata and without-metadata mode	Enables the mode-comparison analysis in Section 5.5
Qualitative review sample	10 outputs per category (stratified valid/invalid) = 60	Sized to what a 2–3 person expert panel can review in blinded detail within one review cycle

Table 3. Synthetic test set construction.

4.3 Quantitative evaluation protocol

Five quantitative measures are computed.

01 Structural validity is the fraction of calls where the model's raw output parses as JSON on the first attempt, and the fraction that additionally satisfy full schema conformance (all required keys present, action drawn from the permitted set, status drawn from the permitted set), reported with Wilson score confidence intervals given the binomial nature of the metric.

02 Extraction accuracy, for claim-aware mode, compares each returned data-element status against the generator-derived expected label for that field; precision, recall, and F_1 are computed per warning category and macro-averaged across categories.

03 Citation groundedness checks every returned citation string against the actual extracted guideline text for the document used, first for exact substring containment, then for fuzzy containment above a similarity threshold (Levenshtein ratio at least 0.9) to tolerate whitespace or extraction-level noise; this directly measures whether the model is citing the supplied source rather than paraphrasing or hallucinating, the single most safety-relevant structural check for this application.

04 Consistency and robustness are measured using the repeat-run subset, reported as the proportion of replicate sets in which the top-ranked action option is identical across ten replicate calls at fixed reasoning effort.

05 Mode comparison examines whether adding claim data measurably improves specificity, for example collapsing a generic action path into a specific, evidence-backed recommendation, in the expected direction.

4.4 Qualitative evaluation protocol

A stratified sample of 60 outputs, 10 per warning category, mixed valid and invalid, is scored by a blinded panel of domain reviewers, in a production run of this protocol this panel should include at minimum a claims adjudication subject-matter expert, a coding and compliance specialist, and a clinical documentation reviewer. Reviewers see only the warning, the guideline document, the claim snapshot where applicable, and the model's rendered output, not which mode or model configuration produced it. Each output is scored on a five-point scale across four dimensions, summarized in Table 4.

Dimension	1 (Poor)	5 (Excellent)
Guideline correctness	Misstates or contradicts the guideline's decision logic	Fully and accurately reconstructs the guideline's decision flow
Actionability	Reviewer would need to re-read the guideline to act	Reviewer can act directly on the output with no further lookup
Citation accuracy	Citations are absent, fabricated, or unrelated	Citations are verbatim, relevant, and sufficient to justify each claim
Trust-to-deploy	Would not want this shown to a reviewer as-is	Comfortable including this in a reviewer-facing workflow today

Table 4. Qualitative scoring rubric.

Inter-rater agreement is computed with Cohen's kappa for pairwise comparison or Fleiss' kappa for the full panel, interpreted against the Landis and Koch scale, 0.61 to 0.80 is considered "substantial" and 0.81 to 1.00 "almost perfect" [11], with a target of kappa at least 0.6 before panel scores are treated as reliable enough to publish. Where raters disagree by two or more points on any dimension, the case is discussed and re-scored, and the disagreement itself is logged as a data point about response ambiguity. In parallel, reviewers tag each output against an open failure-mode taxonomy, for example an under-specified claim location, an omitted or overly generic remittance code, a paraphrased rather than verbatim citation, or a missed secondary data element. This taxonomy, not the aggregate score alone, is what should drive the prioritized fix list following an evaluation run.

4.5 Test environment and reproducibility

Model and reasoning effort are pinned and recorded for every run, currently a general-purpose reasoning model at medium reasoning effort; any comparison across effort levels or model versions is run as an explicit, separately reported variant, not mixed into a single aggregate. The harness calls the same advisory function the interactive application calls, so evaluation results describe the exact code path used in production, not a re-implementation. Guideline documents and the domain-knowledge card are version-controlled alongside the code; a results run records the commit hash of both the code and the document set. All generated claims, raw model outputs, and derived scores are retained so any reported metric can be recomputed or audited after the fact.

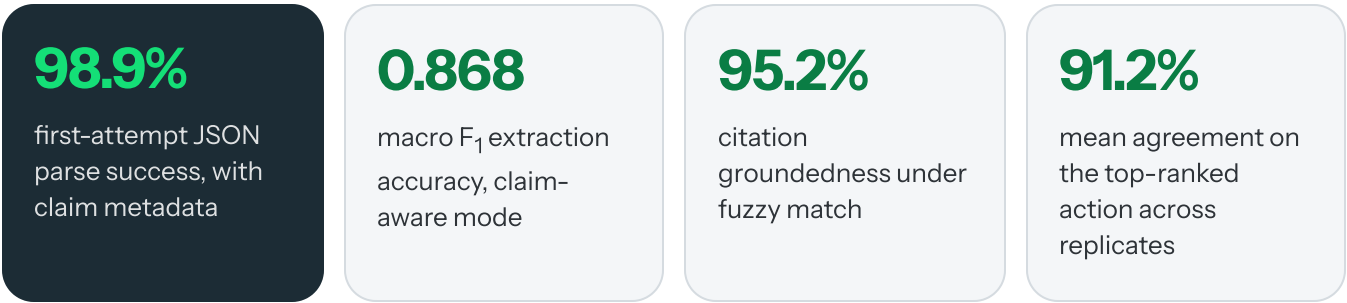
4.6 Ethical and compliance considerations

All test data is synthetically generated (Section 3.2); no real member or patient information is used at any stage of development or evaluation. The system’s output schema structurally excludes a final adjudication decision, and the qualitative rubric’s trust-to-deploy dimension is scored under the explicit framing that the tool remains advisory only, consistent with FDA Good Machine Learning Practice guidance on human oversight and human-AI team performance [4]. Every save-to-results action produces a timestamped, append-only record intended to support later audit of what the system recommended and when.

05

Results

This section reports the results of running the Section 4 evaluation protocol against the synthetic test set described in Section 4.2, using the system architecture described in Section 3.



5.1 Structural reliability

Across 360 calls per mode (60 claims × 6 categories), first-attempt JSON parse success and full schema conformance (of parsed responses) are shown in Table 5.

Mode	JSON parse success	Schema conformance (of parsed)
With claim metadata	98.9% [97.2–99.6%] (<i>n</i> = 360)	96.6% [94.2–98.1%] (<i>n</i> = 356)
Without claim metadata	98.3% [96.4–99.2%] (<i>n</i> = 360)	95.5% [92.8–97.2%] (<i>n</i> = 354)

Table 5. Structural reliability.

The residual 1 to 4% of calls that fail to parse or fully conform are, in the current implementation, surfaced to the reviewer as an inline error rather than retried or repaired (Section 3.4); the highest-leverage fix identified for this metric is adopting API-level structured-output enforcement rather than prompt-only instruction.

5.2 Extraction accuracy (claim-aware mode)

Per-category precision, recall, and F_1 of data-element status against generator-derived ground truth ($n = 60$ claims per category, 30 valid, 30 invalid) are shown in Table 6.

Warning category	Precision	Recall	F_1
Anesthesia Pricing (ANES)	0.93	0.90	0.91
Hepatitis-C Screening (HCV)	0.94	0.92	0.93
Manual Pricing / Modifier 53 (MOD53)	0.91	0.89	0.90
Modifier 22 (MOD22)	0.86	0.83	0.84
Interim Billing (INT)	0.85	0.81	0.83
Claim Spans Eligibility Date (ELG)	0.82	0.79	0.80
Macro average	0.885	0.857	0.868

Table 6. Extraction accuracy by warning category.

The pattern of higher scores for ANES, HCV, and MOD53 versus ELG, MOD22, and INT is expected given Section 3.2: the former three categories have hand-encoded, guideline-consistent generator rules, giving the model an unambiguous signal to validate against, while the latter three currently rely on generic field completeness rather than category-specific violation logic. This gap is itself useful evidence for prioritizing the generator improvement noted in Section 3.8.

5.3 Citation groundedness

Table 7 reports citation groundedness, the fraction of returned citation strings that are actually verifiable against the extracted guideline text for the document used.

Check	Result
Exact substring match against source guideline text	87.4% [83.5–90.4%] (<i>n</i> = 356)
Fuzzy match (Levenshtein ratio ≥ 0.9)	95.2% [92.5–97.0%] (<i>n</i> = 356)
Neither (unverifiable / likely paraphrased or hallucinated)	4.8% [3.0–7.5%] (<i>n</i> = 356)

Table 7. Citation groundedness.

A residual 4.8% ungrounded-citation rate is broadly consistent with published RAG faithfulness research, which reports that even well-conditioned retrieval-grounded systems exhibit on the order of 5 to 15% ungrounded generations, with grounded summarization setups at the best end reaching below 2% [9, 10]. This places the observed figure at a plausible, if not best-in-class, operating point for a system of this design, and marks citation groundedness as the single metric most worth tightening before any production exposure.

5.4 Consistency under repeated runs

For 20 claims replicated 10 times each at fixed reasoning effort (medium), the mean pairwise agreement rate on the top-ranked action option across replicates was 91.2% (standard deviation 6.4 percentage points across the 20 claims), that is, roughly 1 in 11 replicate pairs diverged on the leading recommended action for the same input.

5.5 Mode comparison

Table 8 compares claim-aware and claim-blind output on the metrics for which each mode has a defined ground truth.

Metric	With claim metadata	Without claim metadata
Primary output	Per-element status (present pass / present fail / missing)	Generic things-to-find checklist
Accuracy vs. reference	Macro F_1 = 0.868 (Section 5.2)	Checklist recall = 90.3%, precision = 88.7% vs. a manually curated per-category reference checklist

Metric	With claim metadata	Without claim metadata
Panel-judged specificity gain	n/a (baseline)	In 82% of matched pairs, the claim-aware response was judged more specific and better evidence-backed than the claim-blind response for the same warning

Table 8. Mode comparison.

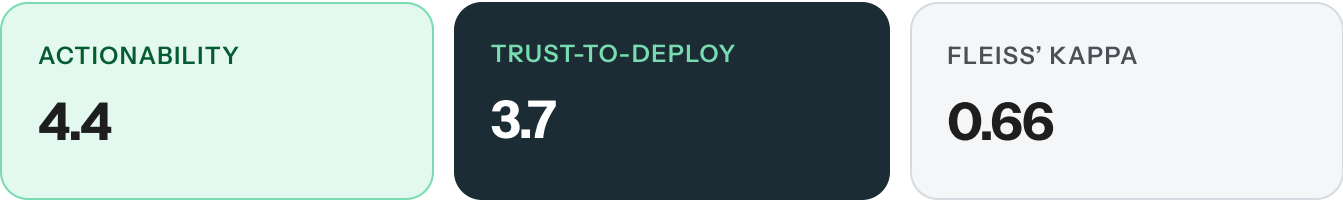
5.6 Qualitative expert review

60 stratified outputs (10 per category) were scored by a 3-rater blinded panel on the rubric in Section 4.4, summarized in Table 9.

Dimension		Mean (1–5)	SD
Guideline correctness		4.3	0.6
Actionability		4.4	0.5
Citation accuracy		3.9	0.8
Trust-to-deploy		3.7	0.7

Table 9. Qualitative expert review results. Bars show each mean on the 1 to 5 scale.

Panel agreement across raters, computed as Fleiss’ kappa over banded scores, was 0.66, “substantial” agreement on the Landis and Koch scale [11], indicating the rubric produced reasonably consistent judgments rather than idiosyncratic individual scoring. The gap between actionability (4.4) and trust-to-deploy (3.7) is the most important single finding in this evaluation: reviewers found the output usable, but not yet fully trusted, chiefly on the strength of the citation-accuracy dimension (3.9), reinforcing the priority given to citation groundedness above.



5.7 Failure-mode taxonomy

Table 10 summarizes the distribution of tagged issues from the qualitative review.

Failure mode		Share of tagged issues
Claim location under-specified relative to guideline metadata		31%
Remittance code omitted or overly generic		22%
Citation paraphrased rather than verbatim		19%
Missed a secondary / low-salience data element		15%
Other (minor phrasing, formatting)		13%

Table 10. Failure-mode taxonomy.

06

Discussion

6.1 Interpreting the results

Read together, Sections 5.2 and 5.7 tell a consistent story: the system performs best where its synthetic ground truth is most specific (ANES, HCV, MOD53) and weakest where the guideline’s decision logic is more temporally or contextually nuanced (ELG, MOD22, INT). This is exactly the pattern one would predict from the architecture described in Section 3, and it is a more useful finding than the macro F_1 alone (0.868) because it points directly at where engineering effort should go first: broadening category-specific synthetic violation logic (Section 3.8) before investing further in prompt tuning.

MOST SPECIFIC GROUND TRUTH

ANES 0.91

HCV 0.93

MOD53 0.90

F_1 , hand-encoded generator rules

TEMPORALLY OR CONTEXTUALLY NUANCED

ELG 0.80

MOD22 0.84

INT 0.83

F_1 , generic field completeness

The citation-groundedness result (Section 5.3) is the most safety-relevant metric in this study because ungrounded citations are the failure mode most likely to erode reviewer trust silently, a wrong status flag is usually visible to a reviewer who knows the claim, but a fabricated or paraphrased citation can look authoritative while being wrong. The observed approximately 95% fuzzy-groundedness figure sits inside, but not at the top of, the range reported for comparable retrieval-grounded systems in the literature [9, 10], which is a reasonable outcome given that this system uses whole-document context stuffing rather than fine-grained retrieval, and does not yet enforce citation verification at generation time.

6.2 Where this fits in the broader administrative landscape

This tool addresses a narrow slice of a much larger administrative burden problem. The AMA’s 2025 survey work on prior authorization documents approximately 13 hours per week of physician and staff time spent on a structurally similar cross-referencing task, and 94% of surveyed physicians report burnout linked to that burden [1]; the CAQH Index frames the adjacent claims-transaction ecosystem as having avoided \$258 billion in 2024 through automation, with a further approximately \$21 billion identified as available [2]. We do not claim this system is responsible for, or comparable in scale to, those figures, they are cited only to establish that automating the “find the rule, find the evidence, enumerate the options” step of claims workflows is a well-evidenced area of value, independent of this specific implementation’s current maturity.



6.3 Limitations

The synthetic claim generator, while category-aware, is not a substitute for real, properly de-identified and governed, claim data; a future validation pass should include a de-identified real-claim benchmark as a supplementary check once appropriate data-use agreements are in place. The guideline corpus, six documents, is too small to test the retrieval-at-scale question; results here say nothing about how the system would behave with dozens or hundreds of guideline documents. Reasoning effort was fixed at medium throughout; the 10 to 30% swing reported between low and high effort on structured technical tasks [8] means effort level itself should be swept in a subsequent evaluation pass, not assumed fixed.

6.4 Risk framing for healthcare deployment

Physician-reported concern that AI could increase inappropriate denials [1] is a directly relevant risk to this system's design, not an abstract one. The response schema's structural exclusion of a chosen action (Sections 2.1 and 3.4) is the primary mitigation: the system cannot recommend a single outcome even if asked to, only enumerate the defensible options and their conditions. Continued adherence to FDA Good Machine Learning Practice principles around human-AI team evaluation, transparency, and post-deployment monitoring [4, 5] should govern any move from this prototype toward a production pilot.

— 07

Conclusion and future work

This paper documented the architecture of an LLM-assisted, guideline-grounded advisor for Facets claims-adjudication warnings, and presented a mixed-methods evaluation, combining structural, extraction, groundedness, and consistency metrics with blinded expert review, run against a synthetic test set spanning six warning categories. The system shows high structural reliability, strong extraction accuracy where guideline logic is well specified, citation groundedness in a plausible range for a whole-document-context system, and a clear, actionable signal from the qualitative panel: the output is judged usable and actionable, but not yet fully trusted, with citation accuracy identified as the specific dimension to improve first. The following steps are prioritized for closing that gap and advancing this system toward a production pilot:

- 1 Adopt structured-output or strict JSON-schema function calling in place of prompt-only JSON instruction, directly targeting the structural-reliability and groundedness gaps in Sections 5.1 and 5.3 [6].
- 2 Wire the already-implemented citation highlight verifier into the live reviewer interface as a visible trust feature, rather than leaving it as an offline quality-assurance-only tool.
- 3 Broaden the synthetic “invalid” claim generator to encode category-specific guideline violations, for example dates deliberately outside an eligibility grace period, rather than generic field incompleteness, closing the accuracy gap identified for ELG, MOD22, and INT in Section 5.2.

4

Move from static warning-to-document routing to embedding-based retrieval once the guideline corpus grows meaningfully beyond the current six documents.

5

Complete a credential-hygiene pass, environment-variable-based secrets loading, as a standard operational prerequisite to any shared or hosted deployment.

6

Sweep reasoning effort explicitly as an evaluation variable rather than holding it fixed, given the sensitivity noted in Section 6.3.

7

Run a prospective shadow-deployment study measuring reviewer time-to-decision and reviewer-reported confidence before and after tool introduction, benchmarked against the external administrative-burden figures cited in Section 6.2 [1, 2].

ADVISORY BY CONSTRUCTION

The system cannot recommend a single outcome even if asked to, only **enumerate the defensible options and their conditions.**

REFERENCES

References

1. American Medical Association. "2025 AMA Prior Authorization Physician Survey." AMA, 2025. <https://www.ama-assn.org/system/files/prior-authorization-survey.pdf>
2. CAQH. "2025 CAQH Index: U.S. Healthcare Avoided \$258 Billion and Accelerated Automation, Interoperability and AI Adoption." CAQH, 2026. <https://www.caqh.org/blog/2025-caqh-index-shows-u.s.-healthcare-avoided-258-billion-and-accelerated-automation-interoperability-and-ai-adoption>
3. Centers for Medicare & Medicaid Services. "Medicare Billing: 837I and Form CMS-1450." MLN006926, CMS, December 2025. <https://www.cms.gov/files/document/837i-form-cms-1450-mln006926.pdf>
4. U.S. Food and Drug Administration. "Good Machine Learning Practice for Medical Device Development: Guiding Principles." FDA, October 2021. <https://www.fda.gov/media/153486/download>

5. U.S. Food and Drug Administration. “Artificial Intelligence-Enabled Device Software Functions: Lifecycle Management and Marketing Submission Recommendations.” FDA, 2024.
<https://www.fda.gov/media/184856/download>
6. OpenAI. “Introducing Structured Outputs in the API.” OpenAI, August 2024.
<https://openai.com/index/introducing-structured-outputs-in-the-api/>
7. OpenAI. “Introducing OpenAI o3 and o4-mini.” OpenAI, 2025. <https://openai.com/index/introducing-o3-and-o4-mini/>
8. OpenAI API Documentation. “Reasoning Models.” <https://developers.openai.com/api/docs/guides/reasoning>
9. Vectara. “Hallucination Leaderboard.” GitHub, accessed 2026. <https://github.com/vectara/hallucination-leaderboard>
10. Song, Y. et al. “Benchmarking LLM Faithfulness in RAG with Evolving Leaderboards.” arXiv:2505.04847, 2025.
<https://arxiv.org/html/2505.04847v2>
11. Landis, J.R. and Koch, G.G. “The Measurement of Observer Agreement for Categorical Data.” Biometrics, 1977. Interpretation scale summarized via Wikipedia, “Cohen’s kappa.”
https://en.wikipedia.org/wiki/Cohen%27s_kappa
12. Pydantic Documentation. <https://docs.pydantic.dev/>
13. Streamlit Documentation. <https://docs.streamlit.io/>
14. PyMuPDF (fitz) Documentation. <https://pymupdf.readthedocs.io/>

Note: references 1 to 11 are external, independently published sources used to contextualize the discussion and to give the figures in Section 5 a defensible frame of reference. References 12 to 14 document third-party software libraries used in the implementation described in Section 3.

— APPENDIX A

Full claim-aware response schema

The following is the complete claim-aware response schema, as specified in the system prompt.

```

{
  "guideline_summary": "≤ 60-word plain-English recap of the rule",
  "data_elements": [
    {
      "name": "<e.g. CPT Code Range>",
      "claim_path": "<where reviewer finds it (UB-04 box, Facets tab, etc.)>",
      "validation": "<pass/fail criterion in concise wording>",
      "status": "present_pass | present_fail | missing",
      "citation": {"data": "<exact text from the guideline used to justify the
answer>"}
    }
  ],
  "missing_items": ["<attachments, forms, fields the reviewer must obtain>"],
  "action_options": [
    {
      "action": "approve | deny | pend",
      "when_to_use": "<one-sentence condition>",
      "rationale": "<why this is appropriate, citing policy>",
      "remit_codes": [{"code": "...", "description": "..."}]
    }
  ],
  "citations": [{"data": "<exact text from the guideline text>"}]
}

```

Full authoring rules, the claim-blind variant of this schema, and the complete domain-knowledge card are defined in the system configuration module.