SIIM CAIMI26 Conference on Artificial Intelligence in Medical Imaging
OCT 26-27 | University of Pennsylvania, Philadelphia, PA

Closing the Radiology Reporting Loop with Open Source: A Contract-First Multi-Agent System from Worklist Triage to Critical-Results Communication

Lalitha Pranathi Pulavarthy, Indiana University Indianapolis; Parvati Naliyatthaliyazchayil, MS
Chaitra Sree Sammeta, MS; Viraj Gadeela; Judy Wawira Gichoya, MD, MS, FSIIM;
Saptarshi Purkayastha, PhD

Briefly describe what attendees will see during your demo.
A live run of one imaging study through the full open-source pipeline: a chest X-ray lands in the Orthanc archive and starts a durable workflow; the study appears prioritized on the reading worklist; the radiologist opens it in the LibreHealth RIS with the AI findings and a pre-drafted impression to hand; after sign-off, the verification engine catches an undocumented critical finding, parks the study at a human gate, and pages the referring physician, who releases it with a one-tap acknowledgment from a phone. Every agent call, gate, and audit record is visible in the workflow timeline, running on a 100-study MIMIC-CXR cohort.

Problem & Motivation: What clinical or operational problem does your MVP address?
Imaging AI usually arrives as a point solution: a model that scores a study but stands apart from the workflow where its output must land. The actual reporting loop — prioritizing the worklist, pulling clinical context, interpreting the images, drafting the impression, signing, verifying the signed report, and communicating critical results — runs across four siloed systems: RIS, PACS, viewer, and EHR. Closing that loop is what commercial integration platforms sell, and it is exactly what community hospitals and low-resource deployments running open-source stacks (OpenMRS, Orthanc, OHIF, LibreHealth Radiology) do not have: no vendor layer carries an AI result from the archive into the radiologist's report form, or a critical finding from a signed report to the referring physician's phone. And any system that touches worklists and patient charts raises operational questions a bare model never answers: who gates its actions, where PHI travels, and what is auditable afterward. We built the missing integration layer as open source, with those safety questions answered by construction rather than by policy.

What You Built: Describe your MVP, tool, or workflow in plain language
A multi-agent system that runs the radiology reporting loop end to end on an open-source hospital stack. Five agents — worklist triage, EHR context assembly, image interpretation, impression drafting, and communications — plus a report-verification engine talk over the A2A protocol, coordinated by a durable Temporal workflow, one state machine per imaging study. The input is simply a study arriving in Orthanc; from there the system scores priority onto the reading worklist (surfaced in OHIF), assembles labs, medications, and problem-list context from OpenMRS FHIR, and runs CAD behind a pluggable model registry — any CNN, vision-language, or large language model that emits findings in our inference contract drops in without workflow changes; today a real pneumothorax classifier runs there, and the remaining tools are stubs by design. It stores a preliminary, authorship-stamped draft impression in the EHR, detects sign-off, then verifies the signed report with negation-aware critical-finding scanning, laterality checks, and a documented-communication check. Failures park the study at a human sign-off gate with tiered paging escalation, an

authenticated override, and HMAC-signed one-tap acknowledgment links for the referring physician. Messages carry identifiers only — never PHI. A 100-study MIMIC-CXR cohort has run through it end to end.

Tech Stack & Development Approach: What tools, frameworks, or methods did you use to build it?
Python 3.11 and FastAPI for the agents; the official a2a-sdk for agent-to-agent messaging; Temporal for durable orchestration; pydantic v2 and JSON Schema for contracts, validated in CI on every change. The hospital stack is entirely open source: OpenMRS with its FHIR R4 module as the EHR, Orthanc for DICOM, OHIF as the viewer, and LibreHealth Radiology as the RIS, composed with Docker and fronted by Caddy for TLS in the hosted demo. The pneumothorax classifier is PyTorch. Our approach was contract-first: we froze the JSON Schemas and agent cards before writing any agent, kept every message a lean reference (IDs only, no PHI), and grew from an in-process walking skeleton to the full stack in milestones, with a mock harness so each agent is testable in isolation. The showcase cohort is MIMIC-CXR, ETL'd into Orthanc and OpenMRS.

Validation & Evidence: Have you done any informal testing or validation? If so, what did you find?
Informal but extensive: the full pipeline has run a 100-study MIMIC-CXR cohort end to end (report finalization driven by our seeding harness), and every workflow emits a structured result payload we roll into metrics (verification hit rates, sign-off gate releases, triage tier spread). Honest findings so far: our pneumothorax classifier's positive scores cluster just above its 0.5 threshold (it flagged 47 of 100 studies), so the operating point needs calibration before the flag rate is clinically plausible — tracked as an open issue. Live integration testing kept surfacing gaps that unit tests missed: the EHR's FHIR build rejects unqualified LOINC search tokens and serializes medications by reference (both initially made our context packets silently empty), and the RIS module's sign event never reached FHIR at all, which we bridged with a sidecar that watches the reporting table. The verification engine's documented-communication rule behaved correctly on seeded reports lacking notification language, parking exactly those studies at the human gate.

Feedback You're Seeking: What specific feedback, help, or collaboration are you hoping to get from the CAIMI26 community?
From clinicians: which post-sign verification rules would actually earn trust in practice, and where the line sits for pre-sign AI drafts appearing in a reporting workflow. From informaticists: sensible defaults for escalation ladders and acknowledgment windows, and lessons learned from FHIR write-backs into production EHRs. We are also actively looking for OpenMRS/LibreHealth deployments interested in piloting the stack, and for collaborators on the next cohort (EMBED mammography).

Known Limitations & Honest Failures: What isn't working yet, or what have you already tried that failed?
Several pieces aren't fully wired yet. Impression-to-RIS write-back: the impression agent drafts correctly, but the last-mile write into the OpenMRS report form is still in progress — in today's demo we type the impression in manually to close the loop. Sign-off-to-verification trigger: the FHIR path from a signed report into the verification agent works in tests but is simulated with manual API calls in the demo rather than firing automatically on real sign-off events. No prospective clinical evaluation: everything measured so far is retrospective public data; we don't yet know how radiologists would actually interact with the AI findings panel or a pre-drafted impression under time pressure. And two vendor-side gaps we worked around: OHIF v3.6's extension points required a thin custom mode for right-panel mounting, and the FHIR write path into OpenMRS relies on conventions the module doesn't formally publish — both work, and both reveal gaps an upstream contribution could close.

Potential for Impact: If this MVP were fully developed and deployed, who would benefit and how?
If deployed at a small community hospital or safety-net facility, this system directly addresses three problems those settings feel most acutely. For radiologists, particularly overnight or single-coverage shifts, the workflow reduces the cognitive load of prioritizing a busy worklist, drafting impressions from scratch, and remembering to communicate critical findings. The pre-drafted impression alone could shave minutes per study. For ordering physicians, especially in ED and inpatient settings, the auditable acknowledgment closes a documented safety gap. Critical findings reach the right person on the right device with a signed timestamp — not a missed page or a note buried in the chart. For patients, faster time-to-notification on critical findings translates directly to shorter time-to-intervention for time-sensitive conditions like tension pneumothorax, aortic dissection, and acute stroke. For trainees, the impression drafts are a teaching artifact — residents can compare their read against the agent's draft and the attending's final, seeing where AI reasoning diverges from clinical judgment. For hospital systems, particularly those without existing enterprise critical-results platforms, this offers an open-source, standards-based (DICOM, FHIR, A2A) alternative to vendor-locked point solutions. The Temporal audit trail supports compliance reporting. For the open-source clinical AI community, this MVP is a concrete demonstration that multi-agent orchestration can be built entirely on open standards and open platforms — no proprietary integrations required.

GitHub Repo:
https://gitlab.com/librehealth/radiology/lh-radiology-agents