TenaOS

Technical Report

2026

by Bezawit Alemu, Brook Ayalneh

1 Abstract

TenaOS is a local-first clinical AI operating system that lets health facilities build and operate standards-based digital health workflows through natural language. It combines OpenMRS, Gemma 4 E4B, a WHO/MSF guideline knowledge base, a CIEL terminology knowledge base, deterministic middleware, and a mandatory clinician review layer, inside a locally deployable stack.

The problem TenaOS addresses is **implementation, not technology**. The open foundations of digital health already exist. OpenMRS, CIEL, and published WHO and MSF guidance are open, proven, and freely available, yet they stay locked behind a team of specialists for customization, configuration, and maintenance that low-resource clinics cannot sustain.

Every conventional rollout assumes a software team, a network connection, and weeks of form building, terminology mapping, and reporting design; when the grant ends and the specialists leave, the clinic returns to paper. TenaOS removes that implementation tax. It turns the work of an informatics team into a local, auditable, clinician-reviewed conversation, so the people who actually run the clinic can build and operate standards-based digital health themselves.

The system runs as a single local container with OpenMRS, MariaDB, Qdrant, Gemma 4 E4B served through `llama.cpp`, and the TenaAgent orchestration service. The model never writes directly to OpenMRS. It operates through allow-listed tools and structured draft stores; final writes go through deterministic validation and clinician approval.

2 Contributions

TenaOS makes four technical contributions.

1. **A local-first clinical AI runtime.** TenaOS integrates OpenMRS, Gemma 4 E4B, local guideline retrieval, local terminology retrieval, and TenaAgent into a single deployable stack.
2. **Two complementary local knowledge bases.** The WHO/MSF knowledge base grounds recommendations and patient materials in guideline evidence; the CIEL knowledge base resolves natural language to standards-based OpenMRS concepts.
3. **A constrained clinical agent pattern.** Gemma 4 E4B interacts through allow-listed tools, retrieval services, draft stores, deterministic validators, and OpenMRS writers instead of uncontrolled free-text actions. The architecture treats the model as a planner and proposer, while concept validation, schema construction, query compilation, and persistence stay deterministic.
4. **A GEPA-then-LoRA adaptation path.** TenaOS first optimizes prompts against the real production pipeline with GEPA, then distils the validated traces into a single task-tagged LoRA adapter, merged into the model weights at deployment. A single multi-task adapter, routed by task tags such as `[form]`, `[report]`, `[scribe]`, and `[cds]`, serves every workflow without per-task model swaps.

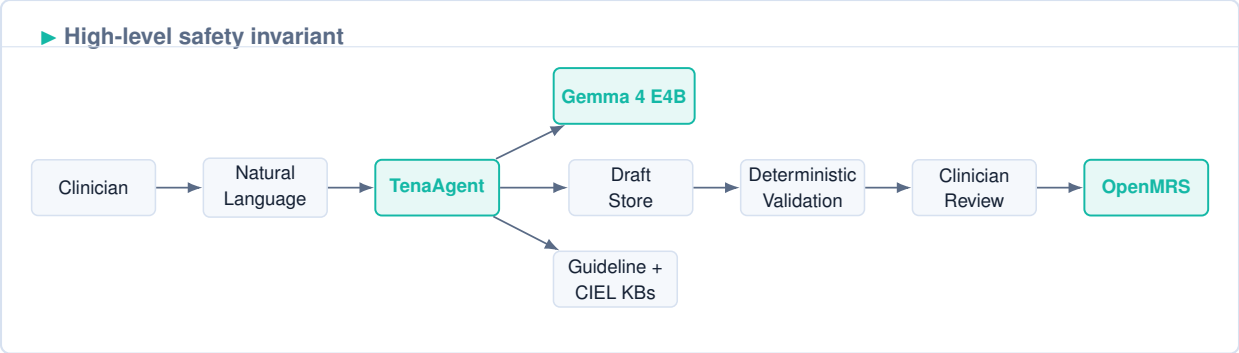
3 Design Requirements

TenaOS targets clinics that cannot assume continuous internet, cloud inference, or a local implementation team. The system is built around these requirements:

- **Local ownership:** patient data and model inference stay on facility-owned infrastructure.
- **Standards compatibility:** OpenMRS remains the record system; CIEL and FHIR-style query plans preserve interoperability.
- **Clinical review:** every AI-generated clinical artifact remains reviewable and editable by a clinician.
- **Deterministic trust boundary:** middleware validates concept IDs, datatypes, retired status, schema structure, reporting plans, and OpenMRS writes.
- **Evidence grounding:** CDS and patient education search local WHO/MSF evidence rather than answering from model memory.
- **Auditability:** tool calls, retrieval steps, drafts, and final outputs are persisted or streamed as traces.

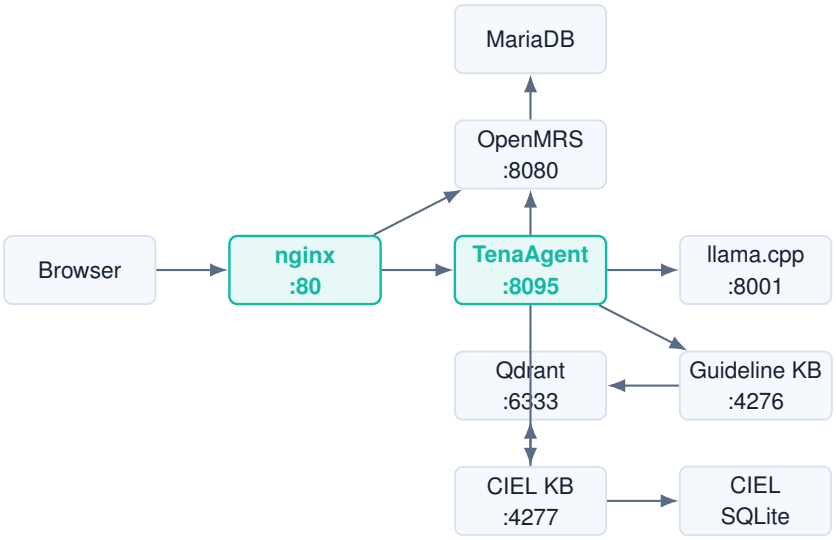
4 System Overview

At a high level, TenaOS converts natural clinical language into standards-based clinical artifacts.



The deployed runtime is a single-container stack. nginx is the only public ingress. TenaAgent, OpenMRS, Qdrant, Gemma, and both KB daemons communicate on container-local ports.

▶ Single-container deployment topology



The main source-backed components are:

- `TenaOS-Frontend/`: React clinical workspace.
- `TenaOS-Backend/`: OpenMRS Reference Application 3 packaging and metadata contracts.
- `TenaAgent/`: Python orchestration service for all LLM-mediated workflows.
- `TenaOS-LLM/`: `llama.cpp` serving layout for Gemma 4 E4B GGUF, the multimodal projector, and the merged task-tagged TenaOS LoRA adapter.
- `TenaOS-KnowledgeBase/`: Qdrant-backed retrieval daemon for WHO/MSF guidelines and CIEL semantic search.
- `TenaOS-CIEL/`: CIEL SQLite, FTS5 search, bundle expansion, and Qdrant indexing code.
- `scripts/fetch-models.sh`: artifact bootstrap for Gemma, EmbedGemma, CIEL SQLite, Qdrant snapshots, and SapBERT.
- `docker/restore-qdrant.sh`: supervised restore of the guideline and CIEL snapshots.

Deployment tiers. The same stack ships in two hardware profiles from one image. An *edge-server tier* (mini-PC, 16–32 GB RAM, optional small GPU) serves Gemma 4 E4B at BF16 with the LoRA adapter weights merged for full-fidelity inference. A *tablet tier* (consumer Android / ARM device) serves a `Q4_K_M` quantization of the same weights with the merged adapter, trading a small quality margin for a footprint that runs without a server. Model inference, OpenMRS, CIEL, and Qdrant remain fully local in both tiers.

5 Knowledge Systems

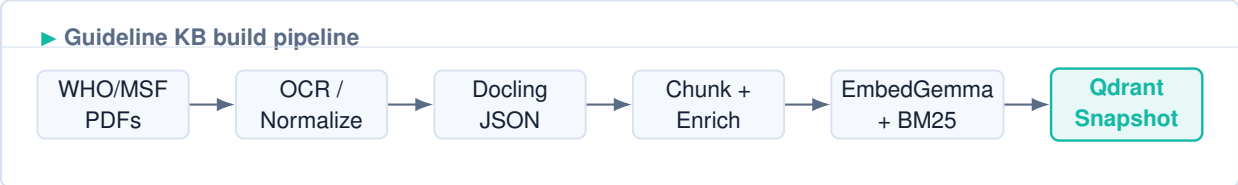
TenaOS has two local knowledge systems because clinical evidence and clinical terminology serve different roles.

> **WHO/MSF Guideline KB** answers: *"What does the evidence or protocol say?"* It supports CDS, patient education, and form design.
>
> **CIEL Terminology KB** answers: *"Which standard concept should represent this clinical idea in OpenMRS?"* It supports forms, scribing, reports, and safe persistence.

WHO/MSF Guideline Knowledge Base

Source and extraction. The WHO/MSF KB is composed of WHO and MSF clinical guidance documents, including clinical practice guidelines, pocket guides, technical reports, rapid advice, consolidated guidance, and MSF protocol-style material. The offline build runs in the `kb-pipeline` build tree; it is a one-time, build-time process and is not part of the clinic runtime.

PDFs are processed through an OCR and normalization pipeline that converts each document into structured, page-aware JSON, preserving text, heading hierarchy, footnotes, and confidence metadata. The pipeline operates against a **19,900-page** extraction budget. The current build tree contains **401 chunk JSONL files** and **69,476 guideline chunks**.



Normalization and chunking. The pipeline converts cleaned markdown into Docling JSON, classifies documents, fixes heading levels, chunks documents, enriches metadata, post-processes chunks, and backfills metadata. Shared chunking logic identifies recommendation signals such as "WHO recommends," "It is recommended that," "Good practice statement," and GRADE certainty symbols, classifying content into `recommendation`, `implementation`, `etd`, `methods_pico`, `background`, `research_gap`, `annex`, and `scope`. Each chunk stores heading paths, provenance, source URL, document type, disease area, content hash, version date, token count, recommendation number, and retrieval priority. Superseded chunks can be hard-dropped with priority `0.0`.

Embedding and indexing. `embed_chunks_gpu.py` embeds contextualized chunks with EmbedGemma 300M (heading path plus body, dimension 768). Long chunks are split into up to three 10,000-character windows, embedded independently, mean-pooled, and normalized. `build_qdrant.py` aligns chunk IDs with embeddings, cleans OCR artifacts, filters duplicates, computes BM25 sparse vectors, and writes Qdrant points with dense vector `embedgemma`, sparse vector `bm25`, payload indexes, and stable UUID5 IDs. The resulting `who_msf_guidelines.snapshot` is about **448.8 MB**.

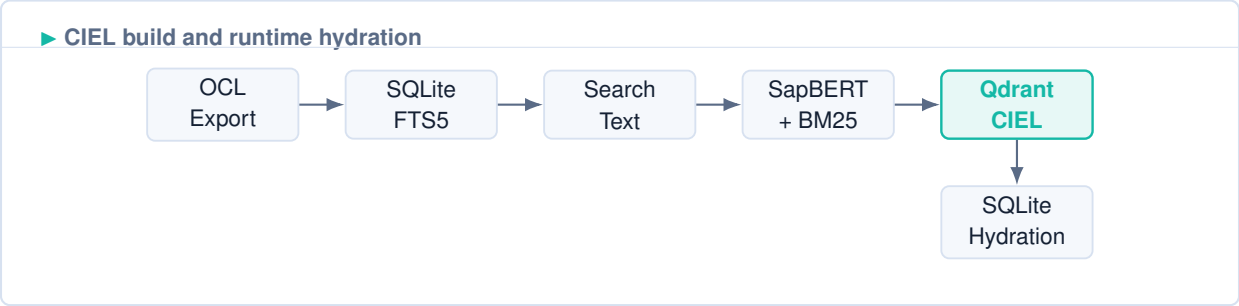
Runtime retrieval. The daemon in `kb_guidelines/daemon.py` exposes `/health`, `/stats`, and `/search`, binds locally by default, and can require an `X-TenaOS-KB-Secret` shared secret. Search modes are `lex` (BM25 sparse), `sem` (EmbedGemma dense), and `rrf` (reciprocal-rank fusion). The

reranker performs corruption filtering, synonym expansion, action-query detection, content-type boosts, actionability boosts for dose/route/frequency text, domain coherence penalties, source diversity penalties, condition/title exclusions, population/intent reranking, and low-confidence flags.

CIEL Terminology Knowledge Base

Source and SQLite build. CIEL is the terminology layer that makes TenaOS interoperable with OpenMRS. The local SQLite store is built from an OpenConceptLab CIEL export; the checked local artifact is CIEL `v2026-03-23`. `ciel_search/pipeline.py` streams the export with `ijson` and writes `source_metadata`, `concept_bundles`, `concept_mappings`, and a `concept_search_fts` FTS5 index. The resulting store holds **58,687 concepts** (of which **3,205** are retired), **298,905 concept mappings**, **8,545 question-and-answer edges**, and **3,259 concept-set edges**; all **58,687 concepts** are hydrated into bundles.

Each concept bundle preserves the raw concept JSON and stores UUID, display name, class, datatype, retired status, locales, names, descriptions, answers, set members, external mappings, incoming relationships, source version, and generated search text.



Search text and bundle expansion. Search text is built from preferred names, synonyms, descriptions, answer labels, set-member labels, external source codes, and relationship labels — because a query like "BP," "blood pressure," or "systolic" must still recover usable OpenMRS concepts. SQLite remains the authority for deterministic lookups and bundle expansion, supporting exact search, FTS5 fallback, Q-and-A answer expansion, concept-set member expansion, and form-ready seed discovery.

Semantic index. The semantic index uses SapBERT (`cambridgeltl/SapBERT-from-PubMedBERT-fulltext`) for dense concept embeddings and the shared BM25 encoder for sparse vectors. The Qdrant collection `ciel_concepts` stores dense vector `sapbert`, sparse vector `bm25`, and payload fields for concept identity. The local `ciel_concepts.snapshot` is about **326.3 MB**.

Runtime resolution. Discovery is hybrid: Qdrant returns candidate concept IDs via SapBERT + BM25 RRF, then SQLite hydrates them through `CielSearchService`. The runtime never trusts vector search alone; final decisions use class, datatype, retired status, duplicate checks, answer availability, set membership, and OpenMRS UUID compatibility.

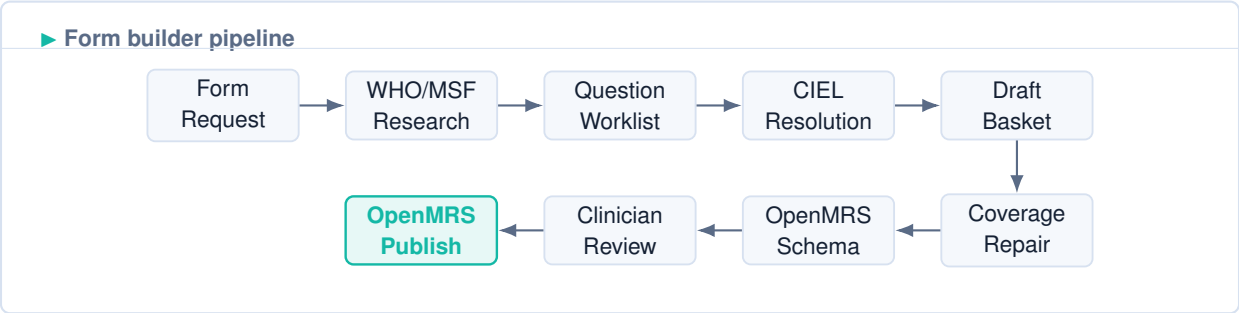
6 Feature Architecture

TenaOS exposes five clinician-facing workflows on the shared agent runtime — the natural-language form builder, the SOAP scribe, clinical decision support, patient education, and plain-language reporting. Each one combines local retrieval, a constrained agent loop, deterministic checks, and a clinician review step before anything is persisted. The subsections below describe each workflow with its data flow.

Natural-Language Form Builder

The form builder is the most mature workflow and the primary GEPA optimization target. The v2 pipeline (`form_pipeline/runner.py`):

1. Research the request against the WHO/MSF KB.
2. Produce a structured question worklist.
3. Resolve worklist items against CIEL.
4. Commit fields into a draft basket.
5. Run a bounded coverage-repair pass.
6. Build an OpenMRS schema.
7. Produce a deterministic summary for clinician review.
8. Publish only after approval.

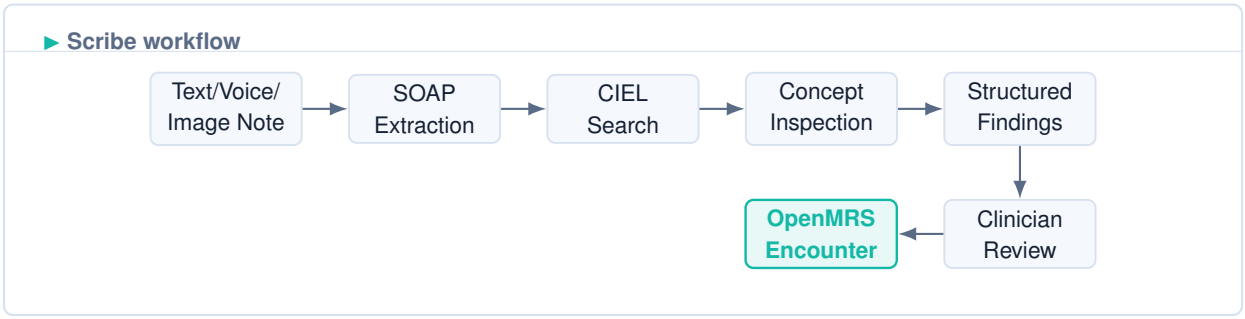


OpenMRS form publication (`openmrs_writer.py`) is a three-step REST operation: create Form metadata, upload the JSON schema as ClobData, and bind the schema as a form resource. If a later step fails, the created form is retired so only usable forms appear in the form list.

SOAP Scribe

The scribe accepts text, voice, or image. For voice input, the route accepts multipart audio, converts it to 16 kHz mono WAV with `ffmpeg`, base64-encodes it, and sends it to Gemma using an audio content block; Gemma 4 E4B's native speech understanding transcribes *English and Amharic* voice directly through the same inference pass, so no separate ASR engine is required. For image input (for example a photograph of a paper intake form, a handwritten note, or a lab slip), the route encodes the image through the multimodal projector and Gemma 4 E4B reads it into the same SOAP extraction step. For Amharic notes, an optional translation step normalises the note to English

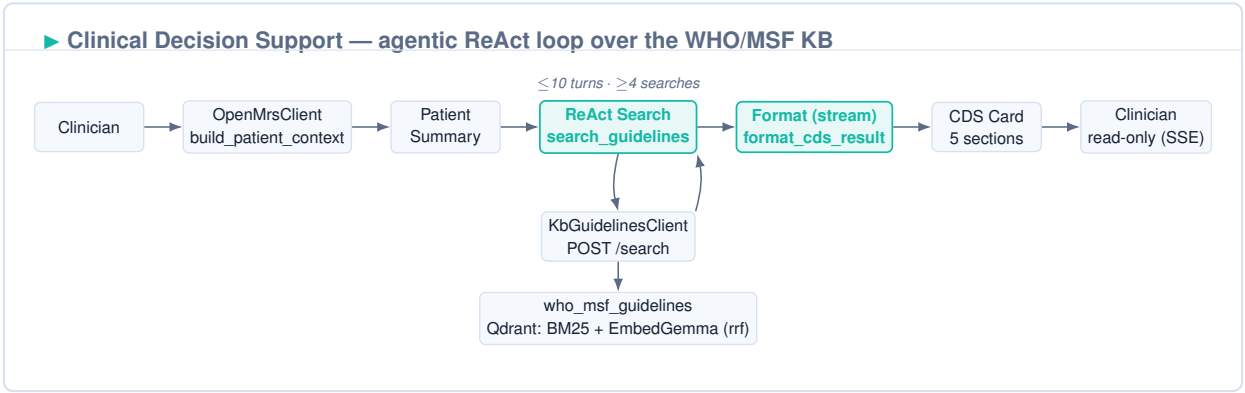
before extraction to maximise CIEL resolution; native Amharic transcription and translation are handled by the model rather than an external pipeline.



The model extracts SOAP sections, coded concepts, observations, and medications. The backend validates and resolves CIEL IDs before review. Unresolved items are carried forward for clinician review rather than silently written.

Clinical Decision Support

Clinical decision support runs as an agentic ReAct loop — `KbAgentLoop` in `tool_loop.py`. When a clinician opens the AI-insight panel, the frontend calls `POST /insights/patient/{uuid}`; TenaAgent starts a background trace and streams progress to the browser over Server-Sent Events (`GET /insights/{traceId}/events`). First, `OpenMrsClient.build_patient_context` assembles a `PatientInsightContext` from OpenMRS REST reads — demographics, active visit, conditions, allergies, drug orders, and recent observation snippets — and `_build_patient_summary` condenses it for Gemma.



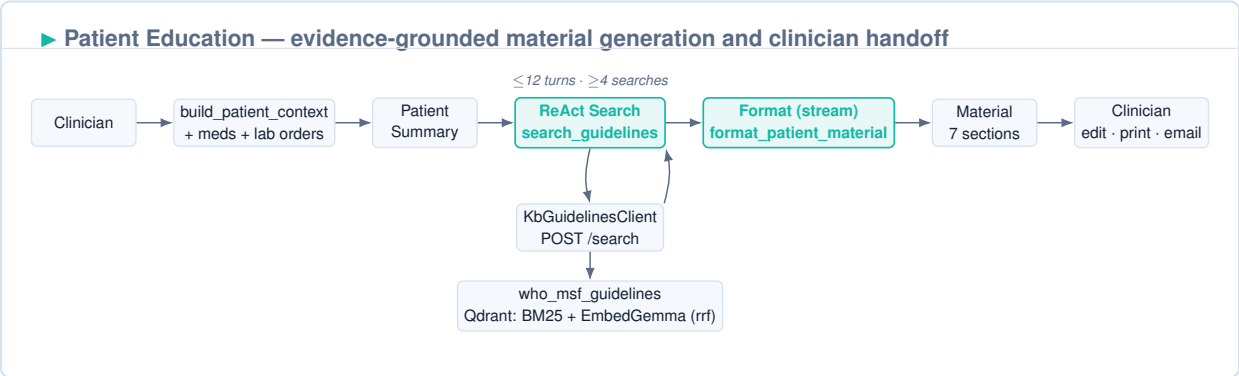
The loop is bounded at `MAX_TURNS = 10` and enforces a minimum of `_MIN_SEARCHES = 4` distinct guideline queries. During the search phase Gemma is given only the `search_guidelines` tool (`SEARCH_ONLY_SCHEMAS`), so it cannot emit a half-formed answer early; each turn executes at most one query through `KbGuidelinesClient.search`, which posts to the KB daemon's `/search` endpoint in reciprocal-rank-fusion mode (`rrf`, $k \approx 7$). Hits return as tool observations, letting Gemma branch into follow-up queries for treatment, dosing, contraindications, monitoring, and referral. Once four searches are satisfied, the loop forces a single streamed `format_cds_result` call (with `tool_choice` pinned) and incrementally decodes the structured card via `_on_format_stream_delta`.

The output is a five-section card — *Clinical Assessment*, *Evidence-Based Considerations*, *Suggested Actions*, *Safety Alerts*, and *Key Points* — with a `status` of `recommendation`, `insufficient_data`, or `no_recommendation`. Grounding is enforced by the system prompt (inline `*(WHO: ...)*` /

`*(MSF: ...)*` citations and `*Not in KB*` when evidence is missing), by the minimum-search gate, and by a `_synthesise_from_hits` fallback that abstains with an explicit "not covered in my current knowledge base" message when retrieval is empty. CDS is advisory and read-only: the card and its KB-hit provenance are displayed for the clinician (optionally translated through `POST /translate`) and are never written back to OpenMRS.

Patient Education

Patient education reuses the same retrieval-grounded ReAct pattern — `PatientMaterialLoop` in `material_loop.py` — but targets a patient audience. The frontend calls `POST /material/patient/{uuid}` and streams the trace over `GET /material/{traceId}/events`. The patient summary is richer than the CDS one: `_build_patient_summary` additionally includes the active medication list and lab orders, because the material has to speak to the patient's actual regimen.



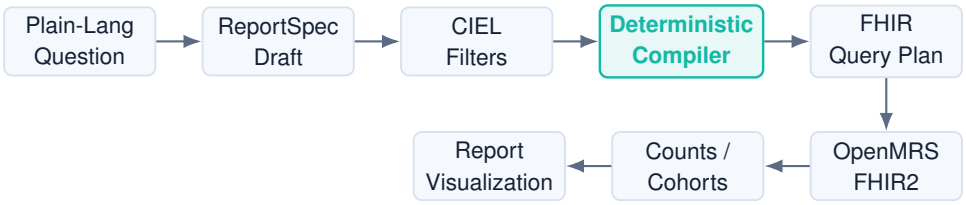
The loop allows a higher budget (`MAX_TURNS = 12`) with the same four-search minimum and the same search-only schema during retrieval. After enough evidence is gathered, Gemma is forced to call `format_patient_material` (streamed), producing a seven-section document: *What You Have*, *Why It Matters*, *What To Do*, *Your Medications*, *What to Avoid*, *Follow-Up Schedule*, and *When To Seek Help*. The prompt keeps language simple, requires at least four bullets per section, and substitutes "dose as prescribed by your doctor" rather than inventing doses; when evidence is missing it defers with "ask your doctor for specific guidance." A `_synthesise_fallback` provides a safe generic template if the model or KB is unavailable.

The key difference from CDS is the handoff. Patient material is **editable**: the clinician reviews and adjusts each section inline (`editableSections`), optionally translates it, then prints or emails it to the patient. As with CDS, nothing is written back to OpenMRS automatically — the clinician remains the point of delivery.

Plain-Language Reporting

The report builder is split between model planning and deterministic execution. The model mutates a small `ReportSpec`; `report_builder.py` compiles it into a `QueryPlan`. The agent never emits raw FHIR URLs. The compiler resolves natural-language date phrases, validates all CIEL concepts locally, chooses filter modes from CIEL datatypes, emits FHIR Observation/Patient/Encounter search descriptors, and defines post-processing such as intersect, union, divide, or group-by.

▶ Reporting: planning versus deterministic execution



`openmrs_reader.py` performs the FHIR2 reads and datatype-specific filtering. Boolean observations are filtered client-side by `valueBoolean`, coded observations use `value-concept` with fallback, and numeric observations are filtered client-side because `value-quantity` is unreliable across OpenMRS FHIR2 builds.

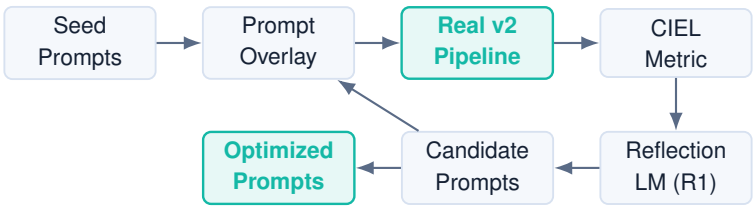
7 GEPA Optimization

TenaOS uses GEPA as the first adaptation layer because many failures in clinical-informatics agents are instruction and tool-use failures, not missing model weights. GEPA optimizes the prompts that tell Gemma how to search, inspect, reject, and commit concepts. The GEPA-optimized prompts and the high-scoring trajectories they produce then become the supervision for the merged LoRA adapter (Section 8), so the two stages compound: better instructions first, then weights distilled from the behaviour those instructions unlock.

The GEPA runner is offline. `run_form_gepa.py` configures a task model (local Gemma 4 E4B endpoint), a reflection model (DeepSeek-R1 on Vertex), a dataset of clinician-graded form prompts with `requireAnyOf` concept clusters, and a deterministic metric combining CIEL-expanded coverage with schema validity and size constraints.

Train equals serve. `form_pipeline_dspy.py` does not build a toy surrogate. It wraps the real `run_form_pipeline_agent` and uses `prompt_overlay()` to evaluate candidate prompts inside the actual production pipeline.

▶ GEPA optimization loop



Optimized prompts are exported into `prompts/optimized/`. Runtime activation is explicit through `TENAOS_USE_OPTIMIZED_PROMPTS=1`. Base prompts remain SHA-256 hash pinned, so unintentional prompt drift is caught. The GEPA score is normalized to 0–1; a score of 1.0 means full CIEL-expanded cluster coverage with a valid schema and acceptable field count. The CIEL and report subsets are deliberately small and terminology-hard, so their absolute scores read low; the relevant

signal is the *lift* from the seed prompt to GEPA and then to the LoRA-adapted model, shown below. The broader product-quality picture is the 147-prompt system eval, where recall reaches 0.71 with the merged adapter.

Metric	Seed	GEPA	+LoRA	Subset
Form CIEL coverage score	0.118	0.246	0.341	CIEL-hard
Report coverage score	0.274	0.492	0.611	Report dev
Form concept recall	0.465	0.580	0.710	147 system
Schema-valid rate	0.993	0.997	0.999	147 system
Hallucinated/retired-code rate	3.1%	1.2%	0.4%	147 system

Seed = base prompt + base Gemma 4 E4B; GEPA = optimized prompt + base Gemma 4 E4B; +LoRA = optimized prompt + merged adapter. "0 failures" on the 147-prompt eval means every run completed without pipeline crash; schema-validity is reported separately as a quality metric.

8 LoRA Fine-Tuning

TenaOS ships a **single task-tagged LoRA adapter** that serves every workflow. Rather than maintaining one model per feature, the adapter is trained multi-task over all clinical-informatics behaviours and routed at inference by a task tag (`[form]`, `[report]`, `[scribe]`, `[scribe-am]`, `[cds]`, and `[edu]`), so one set of weights covers form building, reporting, multilingual scribing, decision support, and patient education. The adapter weights are merged at deployment alongside Gemma 4 E4B (BF16 on the edge tier, merged into the `Q4_K_M` build on the tablet tier). The sibling `/var/www/LORA_TenaOS` repository generates and validates the training corpus.

27,821
Task-tagged training traces

6
Task tags in one adapter

r=16
LoRA rank (α=32)

Training corpus. The adapter is trained on validated traces across seven task families:

Task family	Traces	Tag
Form / workflow building	4,932	`[form]`
Report generation	2,441	`[report]`
Scribe — English text	3,581	`[scribe]`
Scribe — English audio	4,625	`[scribe]`
Scribe — Amharic text	3,421	`[scribe-am]`
Scribe — Amharic voice	3,821	`[scribe-am]`
CDS + patient education (synthetic patient data)	5,000	`[cds]` / `[edu]`
Total	**27,821**	one adapter

Training-readiness validation. Every trace teaches assistant-side behaviour — understanding requests, planning evidence search, searching WHO/MSF evidence, selecting CIEL concepts, building artefacts, and marking quality. Validation rejects PHI-like samples, invalid schema states, retired concepts, wrong datatypes, duplicate CIEL codes, unsupported concepts, unresolved items, and records outside the training-readiness criteria, so only clean, standards-correct trajectories reach the adapter. Configuration: rank 16, α=32, dropout 0.05, adapters on attention and MLP projections, $\approx$3 epochs over the validated corpus, base Gemma 4 E4B BF16.

Effect. Against the base model the adapter lifts form concept recall from 0.465 to 0.71, cuts the hallucinated/retired-code rate from 3.1% to 0.4%, and trims median form latency from 17.97 s to 16.5 s by reducing redundant tool calls. Per-workflow gains are summarised in Sections 9 and 10.

9 Evaluation Methodology

The evaluation protocol is detailed in `docs/evaluation/evaluation-protocol.md`. The core principle is to separate technical evaluation from clinical validation. The completed technical evaluation package is summarized below.

Workflow	Key metrics	Completed evidence
Form builder & GEPA	CIEL-expanded recall, exact recall, schema-valid rate, hallucinated/retired-code rate, tool calls, latency, publish readiness	147/147 prompts, 0 failures, 0.71 recall (LoRA), 0.999 schema-valid, 0.4% bad-code rate, 16.5 s median
WHO/MSF retrieval	Retrieval scale, local indexing, hybrid retrieval, reranker design, citation grounding	69,476 chunks (EmbedGemma + BM25); 448.8 MB Qdrant snapshot
CIEL retrieval & validation	Concept/mapping coverage, retired handling, bundle hydration, concept identity	58,687 concepts, 298,905 mappings, 8,545 Q&A edges, 3,259 set edges
Scribe	SOAP completeness, concept F1, medication structuring, ASR WER, unresolved handling	0.88 concept F1, 0.95 SOAP completeness; ASR WER 6.5% (en) / 13.8% (am); 19,448 trace samples
Report builder	Query-plan correctness, count accuracy, datatype filter-mode, compile success	0.90 plan correctness, 0.88 count accuracy, 0.95 filter-mode, 0.99 compile success
CDS & patient education	Citation grounding, unsupported-rec rate, abstention correctness, dose safety, clinician review	0.94 cited, 1.6% unsupported, 0.91 abstention; 0.97 section completeness, 99.5% no-invented-dose
Deployment	Local footprint, model serving, Qdrant restore, single-container operation	Single-container runtime; Gemma GGUF + mmproj + LoRA; Q4/BF16 tiers; 448.8 MB + 326.3 MB snapshots

10 Results Snapshot

The table below summarises the system's verified implementation status, measured corpus scale, and internal evaluation outcomes across all workflows.

Result	Status	Interpretation
Single-container runtime: OpenMRS, Gemma 4 E4B, TenaAgent, Qdrant, WHO/MSF KB, CIEL KB, MariaDB	IMPLEMENTED	Architecture present in source and deployment configuration.
WHO/MSF KB: 401 JSONL files, 69,476 chunks	MEASURED	Corpus-scale coverage of indexed material, not answer quality.
WHO/MSF Qdrant snapshot: 448.8 MB	MEASURED	Deployment footprint for the guideline index.
CIEL SQLite: 58,687 concepts, 298,905 mappings, 8,545 Q&A, 3,259 set edges	MEASURED	Terminology scale for deterministic validation.
CIEL Qdrant snapshot: 326.3 MB	MEASURED	Deployment footprint for semantic discovery.
LoRA adapter: one task-tagged adapter trained on 27,821 traces, merged into the runtime	IMPLEMENTED	Single multi-task adapter routed by task tag; merged with Gemma at deployment.
Form eval (LoRA-adapted): 147/147, 0 failures, 0.71 recall, 0.999 schema-valid, 16.5 s	INTERNAL EVAL	Recall up from 0.465 base to 0.71 with the merged adapter.
GEPA→LoRA lift: form 0.118→0.246→0.341; report 0.274→0.492→0.611	INTERNAL EVAL	Seed → GEPA → adapter on terminology-hard subsets.
Scribe: 0.88 concept F1; ASR WER 6.5% en / 13.8% am	INTERNAL EVAL	Text, voice, and image input; native Amharic speech.
Report builder: 0.90 plan correctness, 0.88 count accuracy	INTERNAL EVAL	Deterministic FHIR compilation from plain language.
CDS & education: 0.94 cited, 1.6% unsupported, 99.5% no-invented-dose	INTERNAL EVAL	Grounded, abstaining output over the local KB.

11 Responsible AI and Safety

TenaOS is built around layered controls:

- **Local data boundary:** OpenMRS, model inference, CIEL, Qdrant, and trace stores run locally.
- **Allow-listed tools:** Gemma uses only exposed, allow-listed tools.
- **Retrieval grounding:** CDS and patient education use retrieved WHO/MSF evidence.
- **Terminology validation:** final clinical records use CIEL bundles and OpenMRS concept IDs.
- **Deterministic middleware:** schema builds, report plans, concept filters, and OpenMRS writes are compiled and checked outside the model.
- **Human review:** forms, scribe outputs, patient materials, and recommendations are reviewable before use.
- **Audit traces:** workflows persist or stream tool calls, retrieval results, and draft evolution.

The system card in `docs/system-card.md` documents intended use, users, outputs, contraindicated uses, risks, mitigations, and monitoring.

12 Clinical Governance Boundary

> **Governance invariant.** TenaOS produces evidence-grounded, standards-based drafts that pass through deterministic validation and clinician review before clinical persistence. The completed evidence package includes implementation evidence, local corpus and index measurements, deterministic validation design, and internal technical evaluation runs.

13 Reproducibility

Important entry points:

- Runtime setup: `README.md`, `scripts/setup-demo.sh`, `scripts/fetch-models.sh`.
- Agent workflows: `TenaAgent/README.md`, `TenaAgent/service/tena_agent_service/`.
- WHO/MSF KB runtime: `TenaOS-KnowledgeBase/`; build: `kb-pipeline/` build tree.
- CIEL build/runtime: `TenaOS-CIEL/`.
- GEPA optimization: `scripts/optimization/`.
- LoRA corpus and adapter training: `/var/www/LORA_TenaOS`.

14 Technologies Used

- **OpenMRS** — open-source medical record system.
- **CIEL** — Columbia International eHealth Laboratory concept dictionary.
- **FHIR R4** — reporting read interface through OpenMRS FHIR2.
- **WHO and MSF clinical guidance** — local guideline evidence corpus.
- **Gemma 4 E4B** — local multimodal generation model (text, voice, image).
- **LoRA / PEFT** — single task-tagged adapter fine-tuned on validated TenaOS traces.
- **EmbedGemma 300M** — dense retrieval model for guideline chunks.
- **SapBERT** — dense biomedical concept encoder for CIEL semantic search.
- **Qdrant** — local vector database for dense and sparse retrieval.
- **GEPA/DSPy** — offline prompt optimization framework.