Project name

Capsule — Edge AI Clinical Documentation with Agentic Intelligence

Team

Mohammed Abed — Solo developer. AI/ML Engineer working in Healthcare Tech as a Machine Learning Engineer. Designed, built, and tested the full system end-to-end: on-device model quantization/deployment, agentic backend with clinical knowledge graph, FHIR R4 integration, and React Native mobile app.

Problem statement

Clinical documentation consumes 4.5 hours of a physician's day — nearly half their working time (Medical Economics, 2025). This burden is the leading driver of physician burnout, now affecting 40.4% of US physicians (AMA, 2024), and costs healthcare systems an estimated $150B annually in lost productivity.

Existing AI scribes (Nuance DAX, Nabla, Abridge) require transmitting patient data to cloud servers — a critical vulnerability given that 276 million patient records were exposed in 2024 alone (HIPAA Journal). These solutions also lack clinical decision support, cost $200-500/month per provider, and are inaccessible in low-resource settings where cloud connectivity is unreliable.

The gap: No solution today runs entirely on a doctor's phone while also providing drug interaction alerts, billing code suggestions, preliminary radiology reads, and EHR-ready structured output — all without patient data ever leaving the clinic. In low-resource and after-hours settings where radiologists are unavailable, doctors are left waiting for formal reads before making time-sensitive decisions.

Impact at scale: Saving 10 minutes per note across 20 notes/day for 1,000 physicians returns **60,000+ hours annually** to direct patient care. Automated ICD-10 coding improves revenue capture by $50-180 per note. Drug interaction detection at the point of documentation prevents adverse drug events — responsible for over 100,000 deaths annually in the US.

Overall solution

Capsule uses a **master-slave architecture**: the phone (slave) runs MedGemma and MedASR on-device for privacy-sensitive inference, while the doctor's laptop (master) handles the knowledge graph, agentic reasoning, and FHIR server over hospital LAN. Tested on a $150 Android phone (Tecno Spark 40, 8GB RAM) paired with a standard laptop (Ryzen 7, 32GB RAM, CPU only). Future work: a dedicated Capsule desktop companion app for one-click workstation setup.

MedASR (on-device, 101MB): 105M-parameter Conformer CTC model, quantized from 402MB to 101MB via INT8 ONNX. Full on-device audio pipeline: 16 kHz mono resampling, mel spectrogram computation (512 FFT, 128 mel bins, sparse filter optimization), CTC greedy decoding over 512-token SentencePiece vocabulary, and medical text formatting. Entirely on the phone's CPU with no cloud dependency.

MedGemma 4B — On-device (2.0GB): Quantized from 7.3GB to 2.0GB (73% reduction) using Q3_K_M GGUF via llama.cpp. Generates structured SOAP notes from transcripts, interprets lab results, and powers clinical chat — all on-device with tuned generation parameters for mobile memory constraints.

MedGemma 4B — Vision (workstation): Runs with the mmproj vision encoder for multi-modal analysis across 8 imaging modalities (chest X-ray, MRI, CT, dermatology, pathology, fundoscopy, ECG, general imaging), each with specialized radiology prompts producing standardized FINDINGS + IMPRESSION reports. This gives doctors preliminary radiology reads when radiologists are unavailable — common in rural clinics, after-hours shifts, and emerging-market hospitals — enabling faster initial clinical decisions while awaiting formal interpretation.

MedGemma 4B — Agentic Reasoner (workstation): Powers a 5-step autonomous SOAP enhancement pipeline: (1) extract medications, (2) check drug-drug interactions against a 222,271-edge Neo4j knowledge graph, (3) suggest ICD-10 codes from 98,186 entries, (4) correlate with patient lab results from FHIR, (5) synthesize a clinical summary with prioritized safety alerts.

MedGemma 4B — EHR Navigator Agent (workstation): A LangGraph state machine (inspired by Google's MedGemma EHR Navigator notebook) that performs progressive narrowing over FHIR resources to answer complex clinical questions — e.g., "What are the trends across this patient's recent labs and active medications?"

Human-in-the-Loop: Three mandatory checkpoints ensure physician oversight — (1) review/edit transcript, (2) approve/edit/regenerate SOAP note, (3) accept or dismiss each agent finding before FHIR export. Critical safety alerts require explicit physician action and cannot be skipped.

Privacy by design: Patient data never leaves the phone. The workstation communicates over hospital LAN only. Only de-identified medical terms (drug names, symptom terms) reach terminology services — never patient identifiers or narratives.

Technical details

Three-tier architecture:

Tier	Components	Data boundary
Phone (8GB RAM, ARM)	MedASR (ONNX 101MB) + MedGemma (GGUF Q3_K_M 2.0GB) + React Native	PHI stays on-device
Workstation (Ryzen 7 8845HS, 32GB RAM — CPU only)	MedGemma Q4_K_M + vision (llama-server, CPU), HAPI FHIR R4, FastAPI MCP Server (25+ endpoints), Neo4j, LangGraph	PHI on hospital LAN only
Cloud (no PHI)	Neo4j knowledge graph, NLM RxNorm, UMLS/ SNOMED CT	De-identified entities only

Edge model optimization:

Model	Original	On-device	Reduction	Method
MedGemma 4B	7.3GB	2.0GB	73%	GGUF Q3_K_M (3-bit k-means)
MedASR 105M	402MB	101MB	75%	ONNX INT8 dynamic quantization

Total on-device footprint: **2.1GB**. Sequential model loading (MedASR → unload → MedGemma) fits within 8GB RAM. MedGemma pre-loads silently during physician transcript review to overlap load time with user interaction.

Clinical knowledge graph (Neo4j): 1,868 drugs with 222,271 drug-drug interaction edges (sourced from Mendeley DDI dataset / DrugBank IDs), and 98,186 ICD-10-CM codes with full hierarchy (sourced from CMS icd10cm_order_2026). Fulltext indexes for sub-200ms fuzzy search. 45-entry alias table maps brand names to canonical forms.

FHIR R4 compliance: Single-tap export creates 7 interoperable resource types — Encounter, DocumentReference, MedicationRequest (RxNorm-coded), Condition (ICD-10 + SNOMED CT dual-coded), DetectedIssue (DDI alerts), Observation (LOINC-coded labs), DiagnosticReport (radiology). Stored in local HAPI FHIR R4 server, ready for Epic/Cerner integration.

Real-time UX: SOAP generation and agent reasoning stream token-by-token to the UI, with collapsible reasoning traces so physicians can inspect the agent's logic. Clinical chat supports voice input via MedASR — the same on-device pipeline used for dictation — enabling hands-free queries. Patient names are masked throughout the interface as an additional PHI safeguard.

Performance (measured on Tecno Spark 40, 8GB RAM, MediaTek Helio G100):

Operation	Time	Where
MedASR transcription (30s audio)	<10s	Phone (on-device)
Mel spectrogram computation	3–5s	Phone (on-device)
SOAP note generation	~60s	Phone (on-device)
Agentic enhancement (5-step)	<5s	Workstation CPU
DDI graph lookup	<200ms	Workstation (Neo4j)
ICD-10 fuzzy search	<200ms	Workstation (Neo4j)
Radiology vision analysis	<10s	Workstation CPU
FHIR bundle export (7 resources)	<1s	Workstation

Peak phone memory: ~3.2GB during MedGemma inference. Battery impact: <3% per hour of active use. All timings on CPU — no GPU anywhere in the pipeline.

Four complete clinical workflows: (1) Voice dictation → SOAP note → safety enhancement → FHIR export, (2) Lab intelligence with 3 analysis paths (table view, on-device AI summary, EHR Navigator agent), (3) Radiology AI across 8 imaging modalities, (4) Clinical chat with voice input. All demonstrated end-to-end on a $150 phone — no GPU required anywhere in the system.

Deployment: The app includes a built-in settings screen for configuring the workstation URL, with connection testing and a demo mode for evaluation without local infrastructure. Dictation history persists across sessions, showing SOAP generation and FHIR export status for each encounter.

Tech stack: React Native 0.83 + TypeScript (mobile), llama.rn v0.11 (on-device LLM), onnxruntime-react-native (MedASR), FastAPI + LangGraph + langchain-openai (backend), Neo4j 5 Community (knowledge graph), HAPI FHIR R4 (clinical data), Docker Compose (infrastructure). All open-source dependencies.

Reproducibility: Single `docker-compose up` for backend services, mobile app connects over LAN. Setup under 30 minutes. No proprietary APIs, no cloud subscriptions, no GPU.