forgeahead

THE HEALTHCARE LEADER'S GUIDE TO AI-READY AWS ARCHITECTURE

From EHR Interoperability to Compliant GenAI at Scale

Table of Contents

Executive Summary

A patient's medical history doesn't usually live in one place. Part of it's in an EHR from one vendor, part in a lab system from another, and part is still sitting in scanned referral letters nobody ever converted into anything structured. Before a health system can put AI to work on any of that, it has to solve an older problem: getting the data into a form a model can actually read, in a way that survives a HIPAA audit.

This ebook gets into why health systems still treat EHR interoperability and generative AI in healthcare as separate initiatives, when one has to be solved before the other can succeed. It lays out what an AWS healthcare architecture actually looks like when it's built layer by layer, rather than a chatbot bolted onto an EHR export. And it walks through what it technically takes to move from a fragmented data estate to something a clinician can actually trust.

In most cases, a healthcare AI project stalls for reasons that have nothing to do with the model. It's the data underneath that was never built to support one, and a compliance posture never designed with generative AI in mind. Healthcare cloud modernization only works when both get fixed before the model layer goes near production, and Healthcare AI on AWS depends on that same sequence.

Where Healthcare Data Gets Stuck

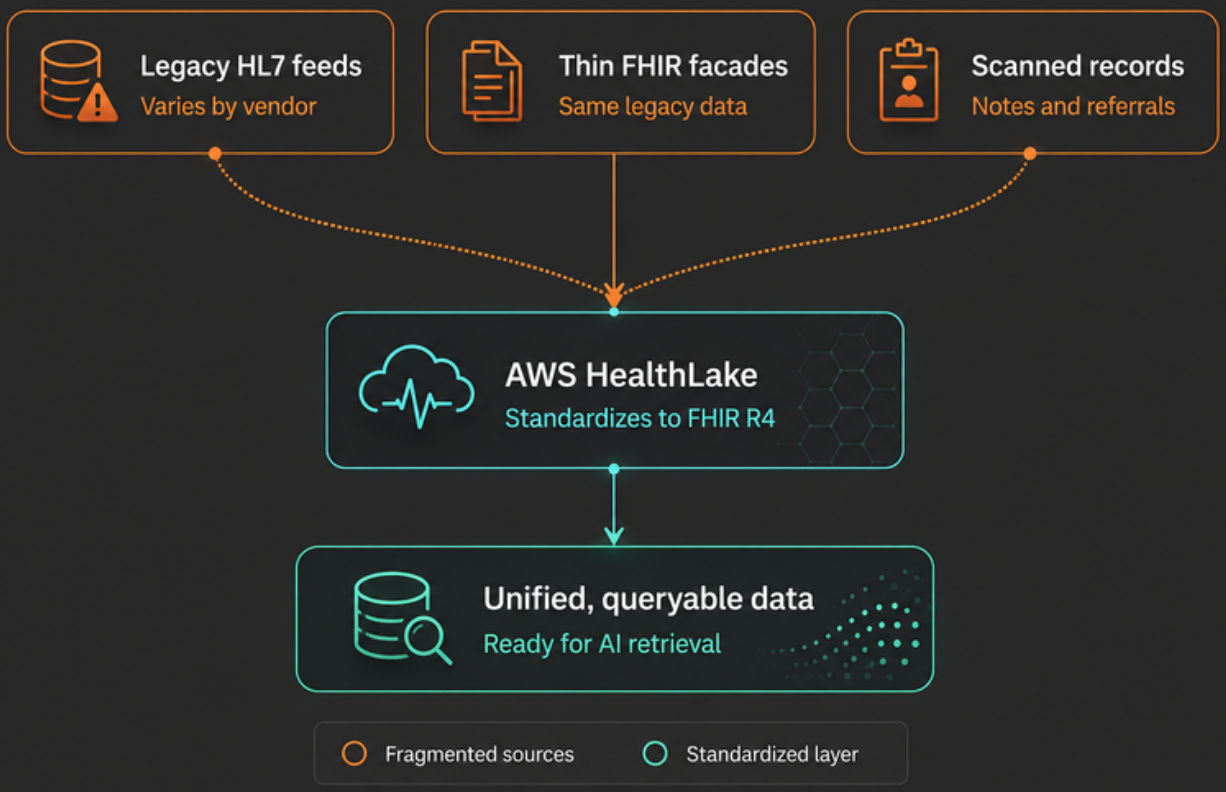
Interoperability sounds like something the industry solved years ago. It hasn't, not below the surface of the EHR's own interface.

Most legacy interfaces still run on HL7 v2, a pipe-delimited messaging standard from the 1980s that varies enough between implementations that two hospitals on the "same" EHR vendor often can't parse each other's feeds without custom mapping. Even where a system claims FHIR support, what's exposed is frequently a thin FHIR facade over the same HL7 v2 data underneath, so the problem doesn't disappear, it just moves a layer deeper.

Then there's the unstructured layer. A C-CDA file can technically hold a patient's medications and diagnoses, but pulling that out reliably, across every EHR vendor a hospital has used over the years, is its own multi-year project. Add scanned referral letters and dictated notes that were never structured, and a large share of clinical history isn't queryable without OCR and clinical NLP doing real work first.

This is partly why FHIR became the industry's real answer. The 21st Century Cures Act requires patient access to their own data, and TEFCA is pushing that access toward standardized, network-level exchange rather than one-off integrations. AWS HealthLake is one concrete implementation: a HIPAA-eligible FHIR R4 store that standardizes clinical terminology on ingestion, mapping to SNOMED CT, ICD-10-CM, RxNorm, and LOINC, so data from different systems actually means the same thing once it lands in one place.

That standardization step is the real starting point for Healthcare cloud modernization. Skip it, and whatever AI gets built afterward just inherits the same fragmentation, now with a chat interface on top.


Legacy HL7 feeds
Varies by vendor
Thin FHIR facades
Same legacy data
Scanned records
Notes and referrals
AWS HealthLake
Standardizes to FHIR R4
Unified, queryable data
Ready for AI retrieval
Fragmented sources
Standardized layer

Why Healthcare AI Projects Stall Anyway

Every health system leadership team has heard the pitch for generative AI in healthcare by now. Most have also watched a pilot lose momentum a few months in, usually for one of the same reasons.

Compliance risk feels heavier than the upside, for good reason. A hallucinated clinical suggestion is a patient safety incident, not a minor bug, and the HIPAA Security Rule's requirement for documented risk analysis doesn't get waived because a workload involves a foundation model instead of a database. So the instinct is to slow down until every control is provable. That's reasonable on its own. Left unchecked, it quietly becomes a reason to never start.

Sometimes the real issue is that the data foundation isn't ready, and nobody wants to say so in a roadmap meeting. It's easier to greenlight a pilot on a curated demo dataset than admit the EHR data needs a structural fix first, even though that fix has to happen eventually.

Budgets add friction too. Interoperability and AI infrastructure both draw from a transformation budget re-justified every cycle, while the operating budget keeping the EHR running never pauses to make room.

And the two efforts often aren't scoped together. Interoperability sits on one roadmap, usually owned by a data team. Generative AI in healthcare sits on another, owned by an innovation group, on a different timeline, even though one genuinely can't be trusted without the other already being solid.


1
Compliance risk
Every control must be provable
2
Data foundation not ready
Rarely admitted in roadmap meetings
3
Budget separation
Re-justified every budget cycle
4
Disconnected ownership
Different teams, different timelines
One Common Outcome
AI initiatives stall.

The Technical Stack Behind an AI-Ready AWS Architecture

A working Healthcare AI architecture is built in layers, and each layer maps to specific AWS services doing a specific job.

Ingestion starts with getting data out of legacy formats. Amazon Textract handles OCR on scanned documents and faxed records. Amazon Comprehend Medical then runs clinical NLP over that extracted text and free-text notes, identifying medications, dosages, conditions, and PHI entities so they can be mapped to structured codes.

The data foundation is AWS HealthLake, which ingests clinical data, including legacy C-CDA documents, and converts it into standardized FHIR R4 resources. AWS launched a data transformation agent in preview in March 2026 that automates CCDA-to-FHIR conversion, cutting what used to be manual mapping down to days instead of months.

Retrieval runs through Bedrock Knowledge Bases, which handles the retrieval-augmented generation workflow end to end: chunking source documents, generating embeddings, storing them in a vector database, and retrieving relevant clinical context at query time with source attribution attached, so a clinician can trace an answer back to the record it came from. For relationship-heavy queries, a knowledge graph on Amazon Neptune, built from entities Comprehend Medical extracts, can answer questions that depend on how conditions, medications, and encounters connect.
Generation runs on Amazon Bedrock, giving access to foundation models including Anthropic's Claude family through a single API, with Bedrock Agents handling multi-step workflows. Bedrock Guardrails adds a policy layer on top, filtering harmful content and redacting PHI from outputs before anything reaches a user.

No single service here makes a compliant AWS healthcare architecture. What makes the stack work is the sequencing: ingest and normalize, ground in structured FHIR data, retrieve with attribution, generate with guardrails, and log everything in between.


1. Ingestion
Textract for OCR,
Comprehend Medical for clinical NLP
2. Data foundation
HealthLake converts
legacy C-CDA to FHIR R4
3. Retrieval
Bedrock Knowledge Bases for
RAG, Neptune for relationships
4. Generation
Bedrock with Claude,
Bedrock Agents, Guardrails for PHI

Compliance as Infrastructure, Not an Afterthought

Compliance can't be layered on at the end, because by the time it's an afterthought, PHI has usually already touched a system that wasn't designed to handle it.

AWS has published a defense-in-depth reference architecture for HIPAA-ready generative AI that applies AWS Key Management Service encryption consistently across every service touching PHI, including Bedrock, HealthLake, S3, and Bedrock AgentCore, using customer-managed keys rather than defaults. CloudTrail logs every access, supporting the data minimization principle at the center of the HIPAA Security Rule: only the specific data needed for an interaction gets pulled, with a full record of what was accessed, when, and by which service.

IAM roles need tight scoping, with each pipeline component granted only what it needs, inside a VPC with private endpoints wherever possible. A signed Business Associate Agreement with AWS is a baseline requirement before any of this touches real patient data, and it needs to cover every HIPAA-eligible service actually in the pipeline, not just the obviously patient-facing ones.

This layered structure exists so no single control failure exposes patient data. If a guardrail misses something, encryption and access logging are still there behind it. That redundancy is the actual meaning of defense in depth, not a label attached to one security feature.

How to Build an AI-Ready AWS Architecture for Healthcare

Five steps make this practical instead of theoretical, and skipping any of them tends to show up later as a more expensive fix.

Start with an honest map of the data landscape: which EHRs, lab systems, imaging archives, and legacy stores actually exist, including ones from mergers nobody's fully integrated, and which can feed a FHIR store without custom extraction first.

Convert the legacy data before building anything AI-facing. Route scanned and unstructured records through Textract and Comprehend Medical, then into HealthLake's FHIR store, using the transformation agent for CCDA conversion. This is the unglamorous part, and the part that determines whether anything built afterward can be trusted.

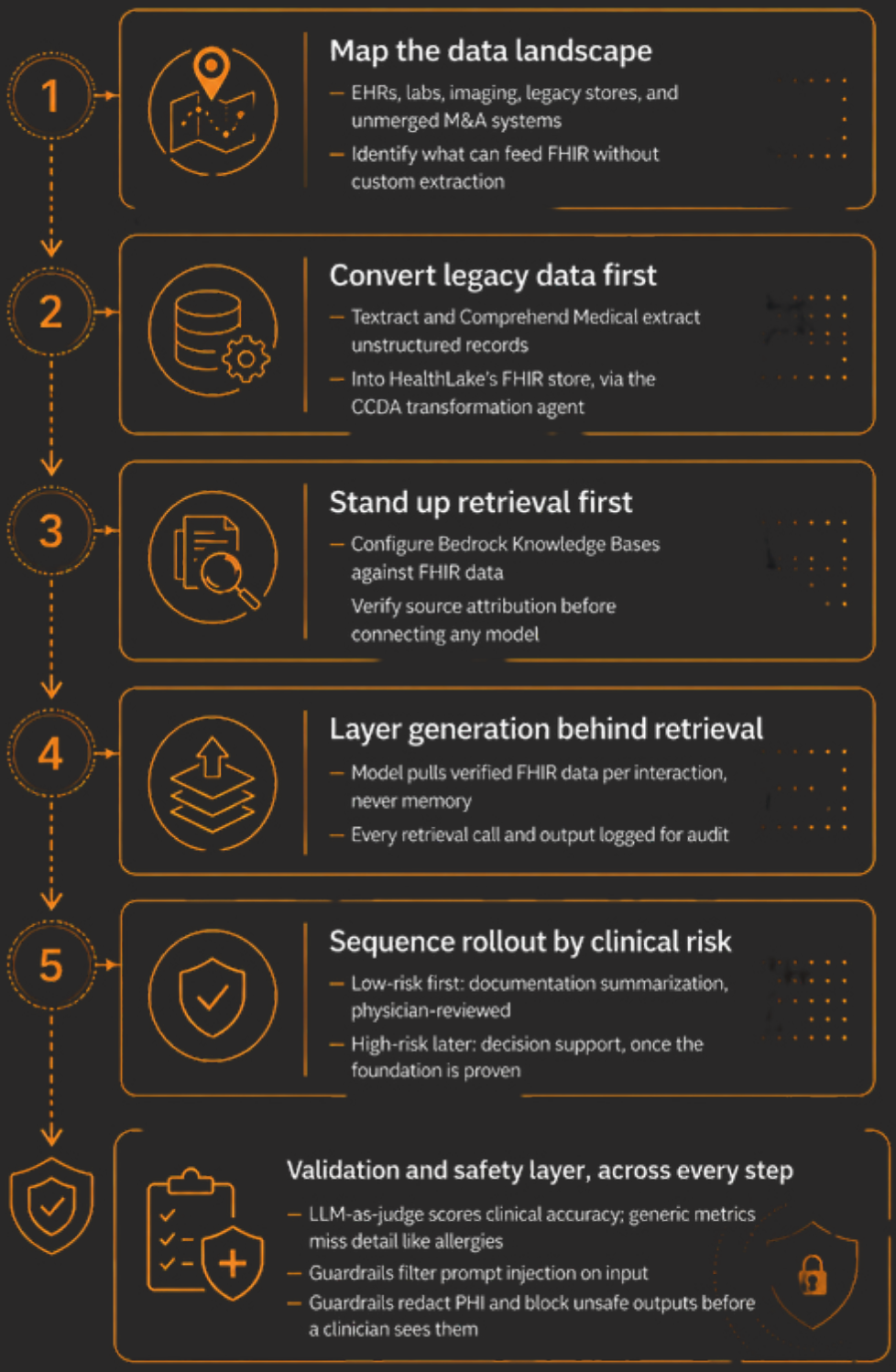
Stand up retrieval before generation. Configure Bedrock Knowledge Bases against the FHIR data, verify source attribution, and test retrieval accuracy against known clinical questions before connecting a model at all. Layer generative AI is behind that retrieval step, never in front of it. The model should pull verified FHIR data for a specific patient interaction, not answer from memory, with every retrieval call logged alongside the model's output.

Sequence the rollout by clinical risk. Lower-risk use cases, like summarizing documentation for a physician who'll review it, are reasonable starting points. Higher-risk use cases, like AI-assisted clinical decision support, belong later, once the data foundation and audit trail have proven themselves in production.

Data first, compliance built in throughout, generative AI added last: that sequence is, in practice, how to build an AI-ready AWS architecture for healthcare a compliance team will actually sign off on, not one that just looks good in a slide deck.

Grounding a model in real FHIR data cuts hallucination risk, but doesn't eliminate it, and healthcare is one of the few domains where a plausible-sounding wrong answer causes real harm. Traditional metrics like ROUGE, built for general summarization, don't measure whether a clinical summary preserved a critical detail like a drug allergy. An LLM-as-a-judge approach, using a second model to score outputs against clinical accuracy and completeness, has proven more effective for tasks like generating radiology impressions, and it belongs in the pipeline before production, not after.

Bedrock Guardrails should sit at both the input and output boundary: filtering prompt injection attempts on the way in, and redacting PHI or blocking out-of-scope clinical advice on the way out. None of this replaces human review for anything clinically significant. It catches a meaningful share of failures before a clinician sees a bad output, which is the difference between a system that earns trust and one abandoned after its first bad mistake.


1
Map the data landscape
– EHRs, labs, imaging, legacy stores, and unmerged M&A systems
– Identify what can feed FHIR without custom extraction
2
Convert legacy data first
– Textract and Comprehend Medical extract unstructured records
– Into HealthLake's FHIR store, via the CCDA transformation agent
3
Stand up retrieval first
– Configure Bedrock Knowledge Bases against FHIR data
Verify source attribution before connecting any model
4
Layer generation behind retrieval
– Model pulls verified FHIR data per interaction, never memory
– Every retrieval call and output logged for audit
5
Sequence rollout by clinical risk
– Low-risk first: documentation summarization, physician-reviewed
– High-risk later: decision support, once the foundation is proven
Validation and safety layer, across every step
– LLM-as-judge scores clinical accuracy; generic metrics miss detail like allergies
– Guardrails filter prompt injection on input
– Guardrails redact PHI and block unsafe outputs before a clinician sees them

Proof in Practice

Forgeahead's healthcare cloud modernization work sits alongside engagements across regulated, data-sensitive industries, life sciences among them. One DevOps-led cloud modernization for a life sciences data platform reached 99.99% data durability, the same reliability bar healthcare workloads treat as table stakes, not something to aspire to.

That pattern isn't unique to one client. AWS reports that HealthLake processes billions of FHIR transactions annually with sub-second latency, proven across hundreds of healthcare customers before generative AI even enters the picture. AWS's own published work on RAG-based medical intelligence applications, combining Bedrock, Comprehend Medical, and Neptune knowledge graphs, shows the same layered pattern holding up for entity extraction and relationship queries, not just document summarization. That track record is the real argument for starting with the data foundation instead of the AI layer.

A Quick Framework for Assessing Your Readiness

A few honest answers show how far a health system actually is from a working Healthcare AI architecture.

- How much of your clinical data is already in FHIR format, versus HL7 v2 feeds, C-CDA exports, or scanned documents that have never touched OCR?
- If a generative AI feature needed a patient's verified medication history right now, could your systems retrieve it in one call with source attribution, or would someone go looking across three platforms?
- Who owns the interoperability roadmap, and is it the same team building your generative AI in healthcare pilots?
- What encryption, IAM scoping, and audit logging exist today for every system touching PHI, not just the newest one?
- Do you have a signed BAA covering every HIPAA-eligible AWS service actually in your pipeline?
- If an auditor asked exactly what data a generative AI feature accessed for one patient interaction, could that trail be produced in an afternoon?

More than one uncomfortable answer means the real project isn't the AI model. It's the data foundation and compliance infrastructure underneath it.


Data format
Access control
Retrieval speed
Readiness framework
BAA coverage
Shared ownership
Audit trail
Data
Governance
Security

Where to Start

Healthcare AI on AWS doesn't start with picking a model. It starts with an honest assessment of where the clinical data actually lives, how much of it is FHIR-ready, what compliance controls are genuinely in place, and which use case carries low enough clinical risk to be a reasonable first deployment.

Forgeahead's AWS Well-Architected review for healthcare organizations maps that gap directly, covering data architecture, security posture, and interoperability readiness together, and comes out the other end with a sequenced plan that treats EHR interoperability as the first milestone and compliant generative AI as the outcome it was building toward all along.

REFERENCES

- https://aws.amazon.com/healthlake
- https://aws.amazon.com/about-aws/whats-new/2026/03/aws-healthlake-data-transformation-agent
- https://aws.amazon.com/blogs/industries/building-a-hipaa-ready-generative-ai-architecture-for-healthcare-on-aws/
- https://aws.amazon.com/blogs/aws/knowledge-bases-now-delivers-fully-managed-rag-experience-in-amazon-bedrock
- https://docs.aws.amazon.com/prescriptive-guidance/latest/rag-healthcare-use-cases/case-1.html
- https://aws.amazon.com/blogs/machine-learning/evaluate-healthcare-generative-ai-applications-using-llm-as-a-judge-on-aws/
- https://forgeahead.io/blog/reducing-costs-in-healthcare-with-aws-cloud-automation/
- https://forgeahead.io/case-studies/beckman-coulter-case-study/

forgeahead

www.forgeahead.io

+1-650-948-1787