CASE STUDY

AI Support Triage:

Reducing Patient Support TAT with an AI Triage Agent for a Clinical Genomics Lab

Executive Summary

Manual triage was causing inconsistencies, delays, and downstream clinical risk. NonStop built a self-hosted AI agent that reads every inbound request, classifies intent and urgency, and drafts a Jira ticket routed to the right team — human reviews before anything is created, PHI never leaves the environment.

Problems faced by the Business

1. Every inbound request is manually read, triaged, and routed — staff as human routers.
2. Jira tickets created by hand — inconsistent categorization and priority across staff.
3. Lag between patient reporting and the right team seeing it — volume made it worse.

Impacts of the problem

1. Same issue categorized differently depending on who picked it up
2. Skilled support time burned on routing, not resolution
3. SLA misses with downstream clinical consequences

Solution: Self-hosted AI agent classifies intent, extracts fields, generates a draft Jia ticket routed to the right team — human reviews before creation, PHI never leaves the VPC.

Impact

- Triage: manual read-route → automated draft in seconds
- Ticket quality: person-dependent → standardized classification and priority
- Staff role: routing and data entry → review and exceptions
- HIPAA posture: strengthened — self-hosted, stateless, no third-party PHI exposure

The Problem

Clinical support teams today are manual routers. Every inbound request, whether it arrives as an email or a portal form submission, follows the same expensive path: a support person reads the message, interprets the customer's intent, assesses urgency, and then manually creates a Jira ticket with the right description, labels, priority, team assignment, and assignee. This works at low volume. It breaks down fast.

> **“The real cost isn't just the time spent triaging; it's the cognitive load of context-switching across dozens of requests, the inconsistency in how tickets get categorized and prioritized, and the lag between a customer reporting an issue and the right team seeing it.”**

For a healthcare or life sciences organization handling patient-adjacent workflows, delays and inconsistencies have downstream consequences that go beyond SLA misses.

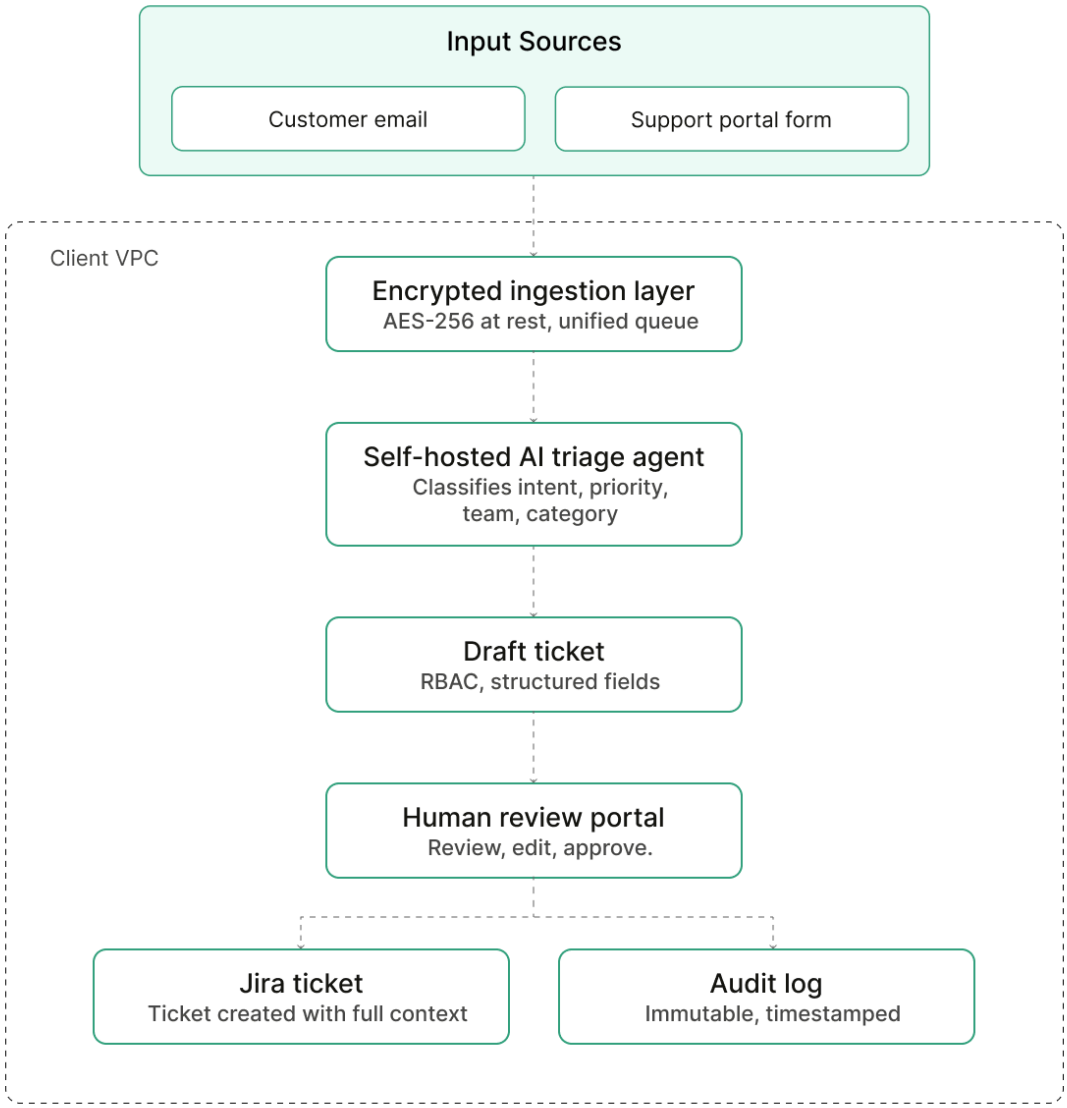
The Approach

The core design principle is simple: patient data never leaves the client's controlled environment.

We're not routing emails through a third-party AI service and hoping the terms of service cover HIPAA. Instead, the architecture is built around three constraints that every component must satisfy: data isolation, auditability, and minimum necessary access.

Model Selection & Hosting

We use a self-hosted open-source LLM (Llama 3 or Mistral) deployed within your own cloud infrastructure. The model runs in an isolated VPC with no public internet egress; it reads what it needs, generates structured output, and that's it. This eliminates the single biggest risk vector: data leaking to a model provider's servers.



AI support agent architecture

Data Flow

Inbound emails and form submissions land in an encrypted ingestion layer (AES-256 at rest, TLS in transit). The AI agent processes each request in a stateless, ephemeral container; no conversation history is persisted in the model, and no patient context accumulates across requests. Each run is isolated: input in, structured ticket out, memory wiped.

The output (draft Jira ticket with extracted fields) is stored in an access-controlled database with role-based permissions. Only authorized support personnel see the review portal. Every action, AI classification, human edit, and ticket creation is written to an immutable audit log.

HIPAA Alignment

No PHI exposure to third parties, self-hosted model, no external API calls with patient data. Minimum necessary access, the agent only extracts what's needed for ticket creation. Audit trail, every AI decision and human override is logged with timestamp, user, and action.