

ML LABS INTELLIGENCE

Clinic-to-Cloud ECG Pipeline for AI Processing

Three ingestion paths, partial upload recovery, and cross-org data isolation — ML LABS built the device-to-cloud pipeline that moves ECGs from exam rooms to a cloud AI system.

CASE STUDY

Author Omar Trejo

Date 2026-02-01

ML LABS

mllabs.com/intel/clinic-to-cloud-ecg-pipeline

A healthcare AI platform needed to ingest ECG recordings from clinic exam rooms into a cloud processing system. The recordings arrived through three different paths — clinician web uploads, SMB file shares on clinic networks, and EHR integration triggers — each with its own failure modes, timing characteristics, and data isolation requirements. The platform served multiple healthcare organizations, so a configuration error that routed one organization's clinical data to another's storage would be both a compliance violation and a contractual breach.

ML LABS built the device-to-cloud pipeline that unified all three ingestion paths onto a single processing queue, handled partial uploads without losing files, enforced per-organization data isolation at the storage level, and replaced manual operational scripts with event-driven scheduling — all deployed across test, US production, and UK production environments.

The Problem

Getting clinical data from a device in an exam room to an AI processing system in the cloud sounds like a solved problem. Standard data pipelines pull from APIs with predictable schemas and reliable connectivity. Clinical device ingestion operates under fundamentally different constraints: clinic networks use SMB file shares as the only programmatic interface, devices write files without notification mechanisms, network interruptions produce partial uploads, and the same recording frequently arrives through multiple paths when a clinician uploads a study that also syncs automatically through the file share.

“ Clinical data pipelines must handle unreliable source networks, file-based data sources, multi-path ingestion that converges on a single processing queue, and regulatory requirements for complete audit trails – simultaneously and without manual intervention.

The platform was already processing web uploads and EHR-triggered ingestion through separate code paths. The SMB file share path — connecting clinic-side devices directly to the cloud — did not exist yet. Adding it as a third parallel system would have created divergent processing logic, inconsistent audit trails, and bugs that only appeared when the same recording arrived through multiple paths. The engagement required building the SMB infrastructure and unifying all three paths into a single pipeline.

SMB File Share Infrastructure

The SMB ingestion path was the most operationally challenging component. Clinic devices write ECG recordings to a shared folder on the local network. The pipeline must discover those files, retrieve them, and deliver them to the cloud processing system — reliably, without losing files, and without processing files that are still being written.

ML LABS built the file share ingestion system that connects clinic-side file shares to the platform's cloud storage, handling file discovery, completeness validation, upload, and processing queue delivery for each organization.

Cross-Organization Isolation

Each healthcare organization's file share sync operates with per-org credentials, per-org cloud storage destinations, and per-org processing configuration — enforcing isolation at the infrastructure layer rather than relying on application-level routing. When onboarding a new organization, automated provisioning creates the organization's storage, configures credentials, and registers the configuration, eliminating manual setup errors.

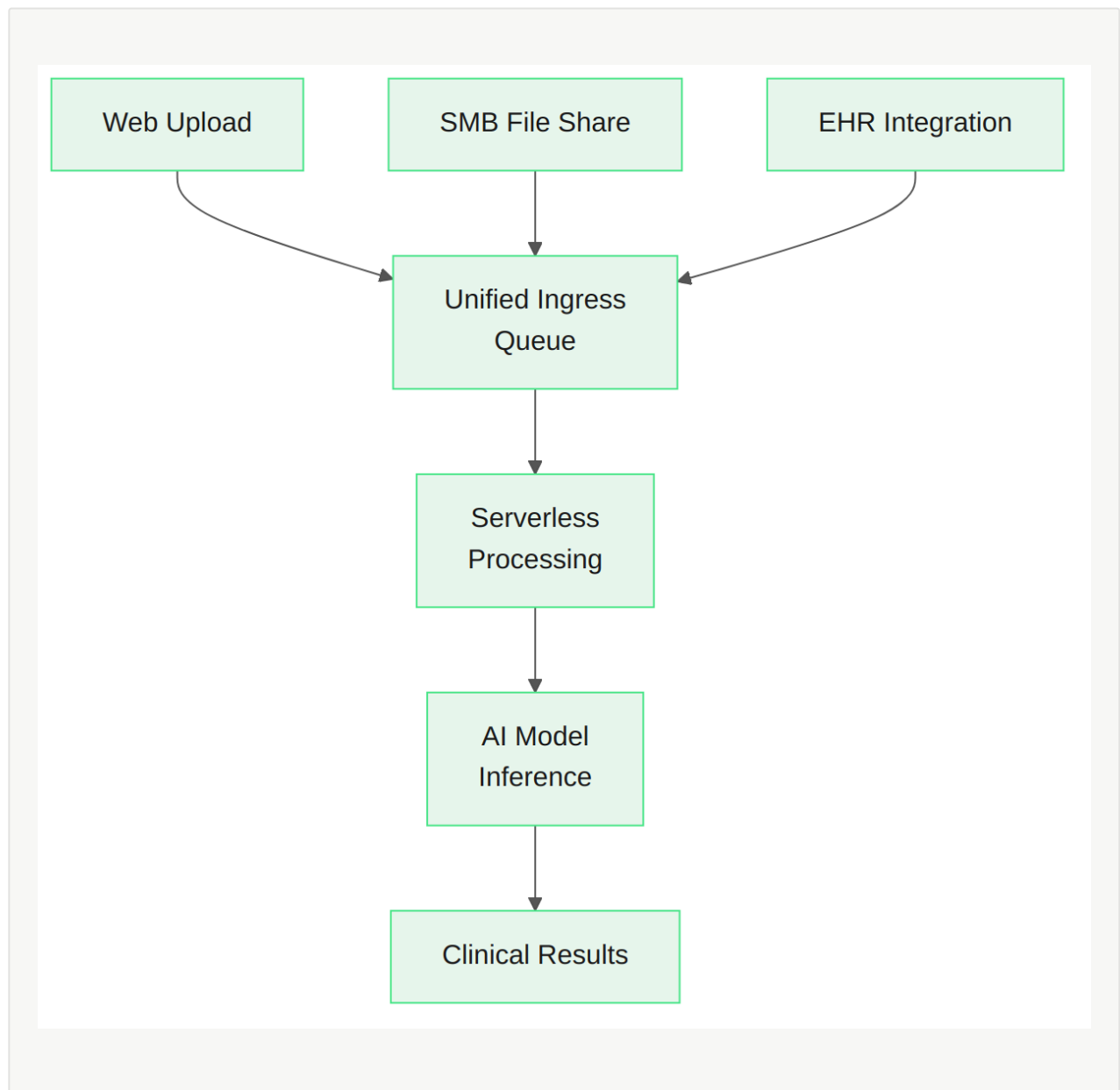
ML LABS resolved partial upload issues where files arriving incomplete from intermittent network connections were being processed before the write completed, producing corrupt records downstream. The pipeline now ensures files are complete before processing them.

File Handling and Archive

The pipeline handles cross-platform file naming inconsistencies that surfaced in production — such as case sensitivity differences between device output and the processing host. Successfully ingested files are archived to prevent re-ingestion while preserving the clinic's local copy for retention requirements. Archive behavior is configurable per organization.

Ingress Queue Unification

Before this engagement, web uploads and EHR integrations used separate processing paths. The SMB ingestion path could have been built as a third parallel system, but that would have tripled the surface area for processing bugs and made cross-path deduplication impossible.



ML LABS unified all ingestion paths onto a single processing queue. Every path — web upload, file share, EHR integration — produces an identical message, and the downstream processor handles all recordings identically regardless of source. This eliminated an entire category of path-specific bugs and made deduplication work across all three paths without special handling.

The unified infrastructure provides full auditability — every file that enters the system through any path gets a log entry with the ingestion source, timestamp, and processing outcome.

Presigned Uploads and Large File Handling

Clinical files vary dramatically in size. Standard ECG recordings are relatively small, but the platform also handles PDF reports, large waveform archives, and video files from certain clinical workflows. Routing multi-megabyte uploads through application servers creates unnecessary latency and timeout failures.

ML LABS implemented presigned URL endpoints for direct-to-cloud-storage upload. The application server generates a time-limited presigned URL scoped to the organization's storage, and the client uploads directly without the file transiting through the backend. This pattern was extended to presigned upload for large videos, where the file sizes made server-proxied upload impractical. Cross-storage utilities handle the mechanics of moving data between organization-specific storage when ingestion and processing are separated.

Operational Automation

The platform's operational maintenance had accumulated manual processes. Weekly backups ran as GitHub Actions workflows that required manual triggering and monitoring. Environment-specific configuration was scattered across deployment scripts.

ML LABS replaced the manual backup scripts with event-driven scheduling and serverless automation. Scheduled rules trigger serverless functions that perform backups without human initiation or monitoring. Failures route to automated alarms rather than going unnoticed until someone checks a GitHub Actions run.

CDN deployment for message delivery was configured across all three environments — test, US production, and UK production — with environment-specific configuration managing the differences between regions. The multi-

environment deployment ensures that configuration changes are validated in test before reaching production clinical workflows.

When Pipeline Complexity Exceeds Internal Capacity

This pipeline architecture is tractable for a bounded number of clinic sites with consistent device types. It becomes categorically harder when the platform must support dozens of sites with heterogeneous device populations, when new organizations are onboarded continuously, or when regulatory requirements span multiple jurisdictions with different data residency rules.

The SMB sync configuration, per-org storage setup, queue infrastructure, and cross-storage utilities form a tightly coupled operational surface where a misconfiguration in one layer propagates silently until clinical data appears in the wrong place or fails to appear at all. The gap between "we have a working pipeline" and "the pipeline runs unattended across all sites" is where most clinical AI initiatives stall. Speed to production is the differentiator — competitors who ship reliable infrastructure first accumulate compounding data and operational advantages that widen with every clinic onboarded.

First Steps

1. **Unify your ingress queue.** If your platform processes web uploads, file share data, and EHR integrations through separate code paths, converge them on a single queue with a common message format — this eliminates path-specific processing bugs that compound as you add sites.
2. **Automate per-organization isolation at provisioning time.** Storage resources, sync credentials, and queue configuration for a new organization should be

created by a single setup script, not a manual checklist — one missed step means a data isolation gap that may not surface until audit.

3. **Replace manual operational scripts with event-driven automation.** Any recurring task that depends on someone remembering to run it — backups, sync monitoring, environment deployments — should run on a schedule with automated failure alerting.

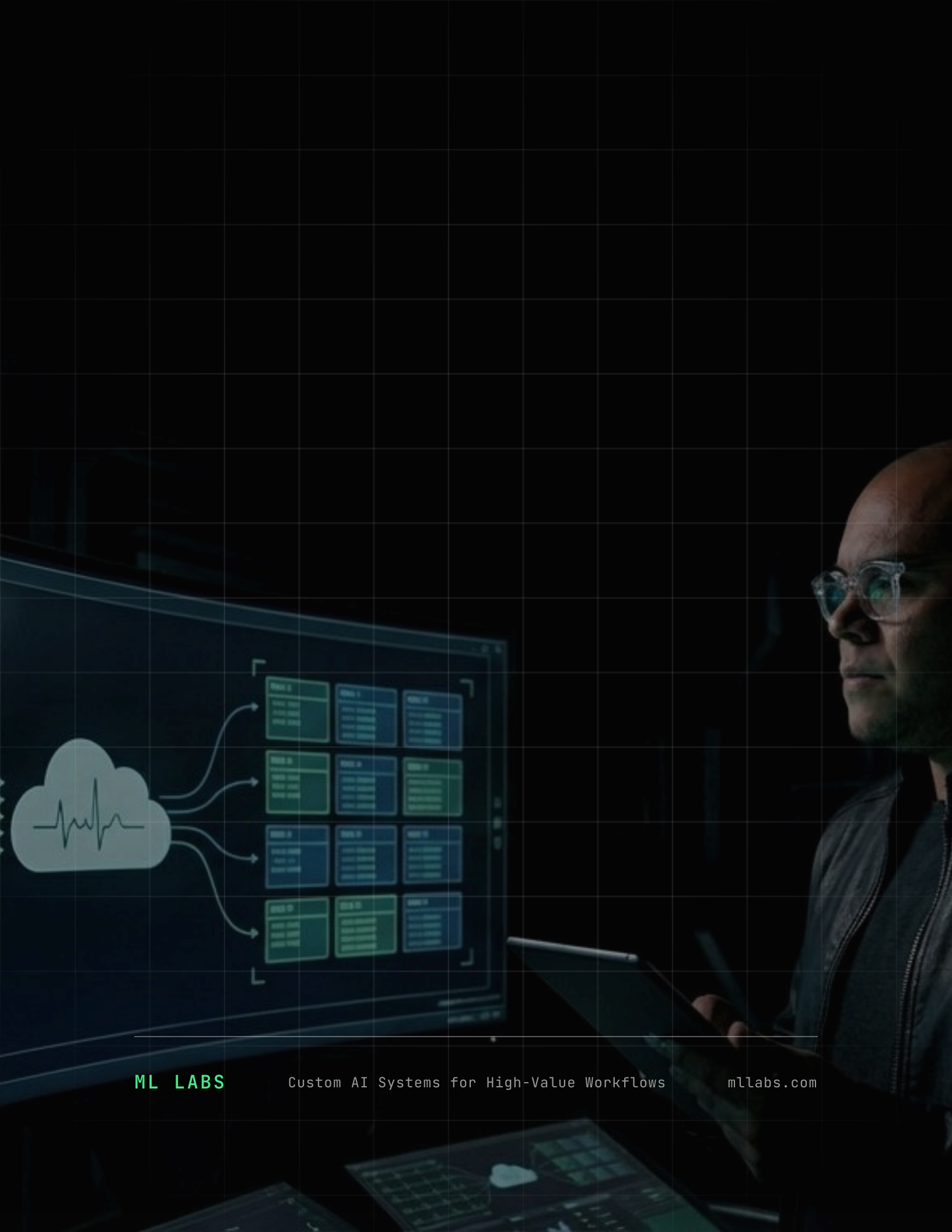
Practical Solution Pattern

Build a unified device-to-cloud pipeline where every ingestion path — web upload, SMB file share, EHR integration — produces an identical message on a single processing queue, with per-organization storage isolation enforced at provisioning time, partial upload recovery at the ingestion layer, and event-driven automation replacing manual operational scripts. Start with the most operationally challenging path (typically SMB), solve its failure modes (partial uploads, case-sensitive file matching, cross-org isolation), then unify the remaining paths onto the same queue and deduplication infrastructure.

This works because ingress unification eliminates the path-specific processing branches that diverge and fail in untested combinations, while automated provisioning removes the human configuration errors that produce data isolation gaps. Organizations that build this infrastructure fastest concentrate the full operational picture — networking, storage, queuing, and deployment — in a single experienced operator rather than distributing it across teams that must coordinate on shared infrastructure they each understand partially. A working pipeline processing real clinical data beats a comprehensive infrastructure plan on every metric that matters. If your organization needs to scope a device-to-cloud pipeline, a **Strategic Scoping Session** can map ingestion paths, isolation requirements, and automation gaps before the build begins.

References

1. Ranchal, R., et al. **Disrupting Healthcare Silos: Addressing Data Volume, Velocity and Variety With a Cloud-Native Healthcare Data Ingestion Service.** *IEEE Journal of Biomedical and Health Informatics*, 2020.
2. U.S. Food and Drug Administration. **Artificial Intelligence-Enabled Medical Devices.** *Regulatory Reference*, 2025.
3. U.S. Food and Drug Administration. **Cybersecurity in Medical Devices.** *Regulatory Reference*, 2025.
4. NIST. **Healthcare Security Rule Guidance.** *National Institute of Standards and Technology*, 2024.
5. An, D., Lim, M., and Lee, S. **Challenges for Data Quality in the Clinical Data Life Cycle: Systematic Review.** *Journal of Medical Internet Research*, 2025.
6. AWS. **Building a Real-Time ICU Patient Analytics Pipeline with Serverless Event Source Mapping.** *AWS Big Data Blog*, 2025.
7. Persons, K. R., et al. **Interoperability and Considerations for Standards-Based Exchange of Medical Images: HIMSS-SIIM Collaborative White Paper.** *Journal of Digital Imaging*, 2020.



ML LABS

Custom AI Systems for High-Value Workflows

mllabs.com