

The Need for Interoperability Today

Laboratory data is critical for public health, yet most systems are designed for individual facilities rather than for national coordination. Laboratory Information Systems (LIS) typically focus on internal workflows and rely on proprietary codes that cannot be shared or mapped across platforms. The result is fragmented information that slows reporting and limits visibility at scale.

OpenLDRv2 Overview

Learn more about how OpenLDRv2 can support Ministries with better data.

A Platform Built for Exchange

OpenLDRv2 (Open Laboratory Data Repository) is a modern, modular platform built with open source tools and designed for interoperability. It can ingest laboratory data in multiple formats, validate against international standards and process information in real time through a secure microservices architecture. The system is lightweight enough to run in resource-limited environments and robust enough to integrate with national digital health stacks.

System Design and Features

OpenLDRv2 uses a flexible architecture to ensure scalability and adaptability.

- **Data ingestion:** APIs accept HL7, FHIR, XML, CSV, JSON, PDFs and more. A secure web interface allows for manual entry when automated feeds are not possible.
- **Validation and mapping:** Open Concept Lab (OCL) is integrated to translate local codes into international standards such as LOINC, SNOMED and ICD.
- **Microservices and plugins:** Apache Kafka enables real-time data streaming, while customizable plugins support schema validation and recipient-specific workflows.
- **Storage:** PostgreSQL manages core laboratory data, MinIO handles objects and OpenSearch provides fast indexing and analytics.

All source code is maintained in the [Association of Public Health Laboratories' \(APHL\) GitHub repositories](#) and published under the Apache 2.0 license, ensuring that country IT teams can access, adapt and extend the platform for local requirements.

Security and Reliability

Security is built into the platform at every level. Keycloak manages user identity and access, Apache APISIX secures API routing and sandboxed plugin execution protects against malicious code. End-to-end data lineage ensures transparency, while audit logging and error handling provide confidence in data quality.

Deployment Considerations

OpenLDRv2 is designed for in-country deployment, aligning with national data sovereignty requirements and resource constraints. It can be hosted on ministry servers, sovereign private clouds or public cloud environments for pilots where appropriate. Offline data capture ensures continuity of operations even where connectivity is limited.

Get Started Today

**Complete our Inquiry Form
to connect with APHL**

Deployment Playbook

| | |
|--------------------------------|---|
| Hosting Models | <p>Deployable on national data centers or sovereign private clouds, with a footprint suited for resource-limited sites.</p> <p>Optional public cloud hosting can be used for regional pilots if requested, but production deployments are intended to align with data sovereignty policies and ministry operations</p> |
| Core Components | <ul style="list-style-type: none">• API and Security: APISIX API Gateway for traffic routing; Keycloak for identity and access management.• Streaming and Processing: Kafka for event streaming; Flink for real-time metrics.• Storage: PostgreSQL for relational data; OpenSearch for indexing and analytics; MinIO for object storage; Redis for caching.• Interfaces: Web dashboards for data feeds, plugin management, users and manual result entry. |
| Compute Tiers | <ul style="list-style-type: none">• Pilot (single site/few feeds): 2 nodes with 4–8 vCPUs, 16–32 GB RAM, 1 TB SSD storage.• Regional (multi-lab pilot): 3–5 nodes with 8–12 vCPUs, 32–48 GB RAM, 2–4 TB SSD storage, and a basic Kafka cluster.• National (full scale): 6–10 nodes with 12–16 vCPUs, 48–64 GB RAM, 6–12 TB SSD storage, and clustered deployments of PostgreSQL, OpenSearch, Kafka and MinIO. |
| People and Skills | <ul style="list-style-type: none">• Leadership: Project lead or product owner for governance and success criteria; solution architect for environment design, security and data flows.• Operations: DevOps or platform engineer skilled in Linux, containers and CI/CD.• Data and Standards: Integration engineer for Kafka, HL7 and FHIR; terminology specialist for OCL, LOINC, SNOMED and ICD mappings.• Data Management: DBAs for PostgreSQL and OpenSearch.• Security and Support: Security officer for access reviews and audit policies; trainer/support lead for user enablement and ongoing operations. |
| Implementation Sequence | <ol style="list-style-type: none">1. Environment readiness (infrastructure, TLS, backups)2. Core platform installation (APISIX, Keycloak, Kafka, PostgreSQL, OpenSearch, MinIO)3. Data model and code mappings4. Feed and plugin setup5. Dashboards and reporting6. Security and monitoring7. User acceptance testing and training8. Go-live and operate |

What Support Can Unlock

With resources for deployment, countries will be able to:

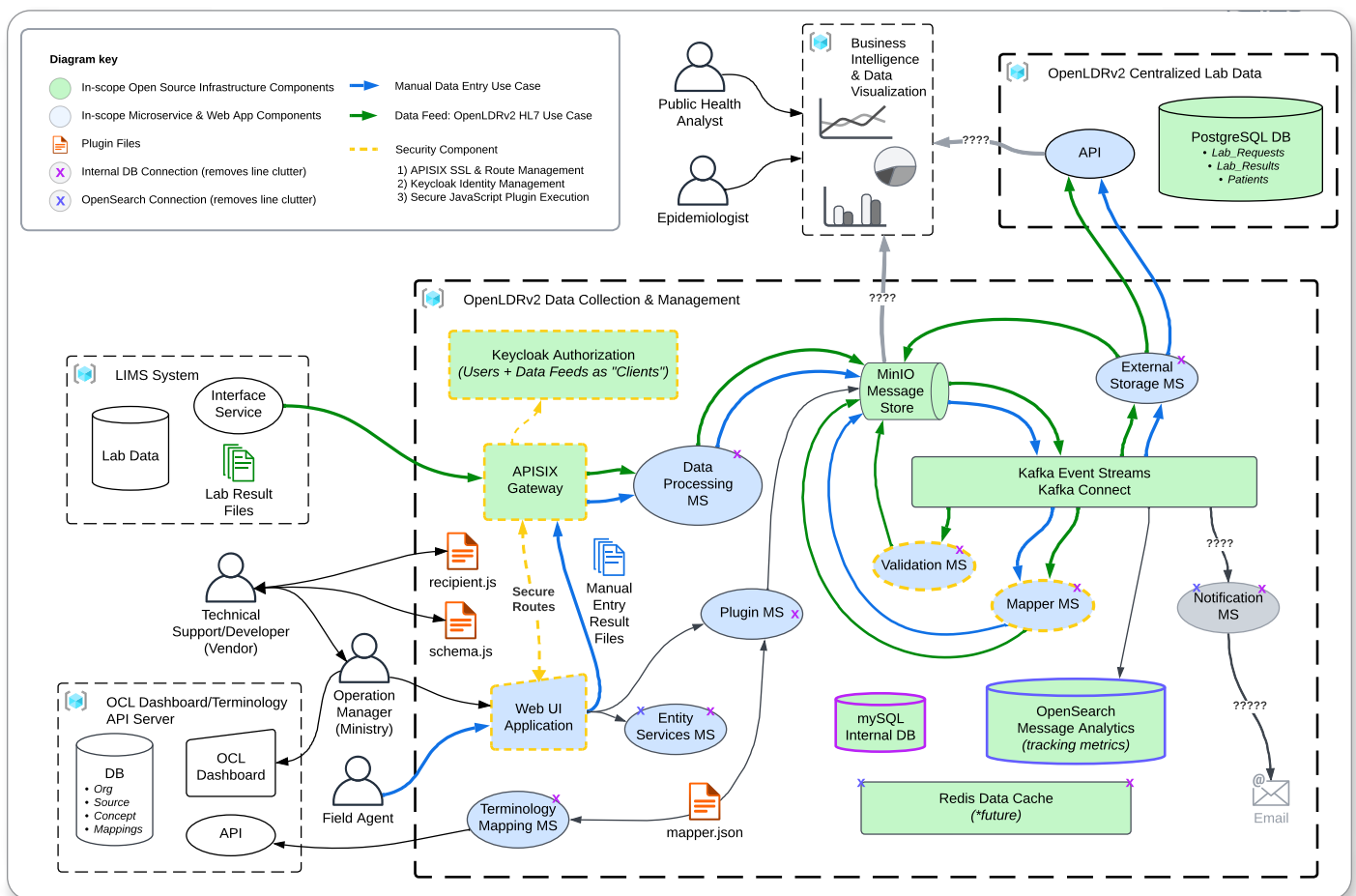
- Pilot the platform across multiple laboratories and LIS environments
- Customize plugins for country-specific workflows and terminology
- Enable offline functionality to support facilities with limited connectivity
- Expand use beyond laboratories to program-specific and regional applications
- Build dashboards that give IT teams and ministries shared visibility into performance

A Technical Foundation for Scale

OpenLDRv2 is designed to integrate with national digital health systems through FHIR-enabled APIs and to align with global donor reporting requirements. By supporting interoperability and standardization from the start, the platform provides a foundation that can scale from a single pilot to national adoption. It is open source, standards-driven, and ready for country IT teams to adapt and extend.

See the figure below for a detailed view of how OpenLDRv2 handles ingestion, validation, storage and interoperability through its modular microservices.

Figure: OpenLDRv2 System



This diagram illustrates the core architecture of OpenLDRv2, including API and security services (APISIX, Keycloak), streaming and processing (Kafka, Flink), storage (PostgreSQL, OpenSearch, MinIO, Redis) and terminology mapping through OCL.

Complete our Inquiry Form to connect with APHL and learn more!