SYSTEM ARCHITECTURE FOR TIERED FORECASTING & OFFLINE-FIRST SUPPLY CHAIN

Technical Specification v2.0 - Post-Validation Revision

ARCHITECTURAL PHILOSOPHY & OVERVIEW

This architecture is designed for the constraints identified in the 2025 Baseline Assessment: 89% unreliable connectivity, low to medium digital literacy, and system fragmentation across multiple platforms (DHIS2, e-LMIS, eAFYA, paper registers). Instead of a single complex AI system, we implement a distributed, tiered architecture that places core functionality where it's needed most, on the mobile device at the facility level.

The system comprises three main architectural layers, each designed to function semi-autonomously while supporting higher-level integration when connectivity permits:

- Offline-first mobile application**:** Primary tool for facility staff, containing the Tier 1 rule-based forecasting engine, local data storage with SQLite in WAL mode, and encrypted offline authentication.
- District-level services: Lightweight containerized services for Tier 2 hierarchical time series forecasting, Tier 3 machine learning models, data aggregation from facilities, and district dashboard for supply officers.
- Central data hub: National-level data warehouse for long-term trend analysis, integration with national health information systems (DHIS2, e-LMIS, eAFYA), and policy-level reporting.

CORE ARCHITECTURAL PRINCIPLE: DATA FLOW FOLLOWS CONNECTIVITY

The system does not assume real-time connectivity. Data moves opportunistically, following the Integration Maturity Model (Level 2: Opportunistic Sync). This design decision addresses the baseline finding that 89% of facilities experience unreliable internet and power, making always-online architectures operationally infeasible.

Critical architectural constraint**:** Every core function required for daily facility operations must work without internet connectivity. Synchronization enhances capabilities but is never blocking.

DETAILED DATA FLOW BY LAYER

Facility Level (Always Available - No Internet Required)

- Health workers use the mobile application for daily operational tasks: stock counts, order generation (using Tier 1 rule-based forecast), stockout reporting and consumption tracking.
- All data is stored immediately in the local SQLite database configured in Write-Ahead Logging (WAL) mode to enable concurrent read operations during background sync processes.
- The application generates paper backup forms matching Ministry of Health registers (e.g HMIS 105) to ensure continuity during device failure or prolonged power outages.
- Tier 1 forecasting engine runs entirely locally using preloaded seasonal adjustment rules and the facility's historical consumption data (last 12 months).

Synchronization Layer (Opportunistic - Triggered by Connectivity)

- Automatic trigger: Background sync process initiates when the mobile device detects network connectivity AND battery level is above 20% AND device is not in active use (to preserve user experience and battery health).

- Manual trigger: Users can initiate 'Sync Now' from the application at any time if connectivity is available, bypassing battery threshold checks.
- Upload payload: Batched JSON packets containing new stock counts, human-adjusted orders with override justifications, consumption data, and sync metadata (last sync timestamp, facility ID, user ID).
- Download payload: Tier 2 statistical forecasts (stored separately from primary SQLite database as read-only informational data), updated delivery schedules, budget status notifications, and system messages.
- Conflict resolution principle: Local user edits always win. The system is designed to trust the health worker on the ground. If a user adjusted an order offline and the central system also modified it, the user's version is preserved and the conflict is logged to the override audit trail for review by district officers.

District Level (Periodic Connectivity Required)

- District server aggregates data from multiple facilities within its jurisdiction as mobile applications successfully sync.
- Tier 2 hierarchical time series forecasting service runs on aggregated facility data using Hierarchical Exponential Smoothing (HES), generating district-wide forecasts and facility-specific predictions that account for cross-facility patterns.
- Tier 3 machine learning models (Random Forest or XGBoost) run on district servers to identify facilities at high risk for stockouts based on consumption patterns, delivery delays, storage constraints, and historical override behavior.
- District supply officers access a web-based dashboard to view these insights, review override audit logs, and plan supply redistributions between facilities.

Central Level (National Reporting & Integration)

- Central data hub periodically pulls aggregated data from district servers using scheduled ETL (Extract, Transform, Load) jobs orchestrated by Apache Airflow.
- Integration with national health information systems (DHIS2, e-LMIS, eAFYA) occurs at this level through RESTful APIs and FHIR-compliant data exchange protocols where supported.
- The central hub supports Ministry of Health officials with national-level dashboards showing aggregated trends, regional stockout patterns, forecast accuracy metrics (MAPE), and system performance indicators.
- Critical design note: The central hub is not required for day-to-day facility operations. Districts can operate semi-autonomously if central connectivity is interrupted.

TECHNOLOGY STACK & KEY COMPONENTS

Mobile Client (Android-First Application)

Core function: Primary data entry interface, Tier 1 forecasting, offline-first data storage, paper form generation, and opportunistic synchronization.

Component	Implementation Specification
Development Framework	Native Android (Kotlin/Java) for robust offline performance and access to hardware-level security features. Minimum SDK version: Android 8.0 (API level 26).
Offline Storage	SQLite database configured in Write-Ahead Logging (WAL) mode. Target size: <50MB per facility. WAL mode enables concurrent read operations during background sync, preventing database locks that would block user data entry. WAL configuration: PRAGMA journal_mode=WAL; PRAGMA synchronous=NORMAL; for optimal performance and data integrity.

Component	Implementation Specification
Data Encryption	AES-256 encryption for SQLite database using Android Keystore System. Encryption keys are generated within hardware-backed secure enclaves (where available) using the Android Keystore API. Keys never appear unencrypted in application memory or storage. Key lifecycle: Generated at first app launch, stored exclusively in Android Keystore, no file-based or plaintext key storage permitted.
Forecasting Engine	Embedded Tier 1 rule-based forecasting logic: 3-month simple moving average of facility consumption + preloaded seasonal adjustment multipliers (e.g +30% for malaria season) + storage capacity constraints. Fully deterministic, requires no ML libraries, runs in <100ms on low-end devices.
Synchronization Protocol	RESTful JSON over HTTPS (TLS 1.3). Batched payloads compressed using GZIP to minimize data transfer. Automatic retry with exponential backoff (initial delay 30s, max delay 15 minutes) for failed sync attempts. Battery optimization: Background sync only when device battery >20% AND not in active use. Manual sync bypasses battery checks.
Data Archiving & Purging	Active retention: Last 12 months of consumption data and stock counts stored locally. Archive trigger: Automated monthly job moves data >12 months old to compressed archive files during successful sync, uploaded to district server, then deleted from local SQLite database. Emergency purge**:** If database size exceeds 45MB, oldest archived data is removed first, then oldest active data if necessary. User is notified and archive is uploaded before deletion.
Offline Authentication	User credentials cached locally (hashed using bcrypt) for offline authentication. Credential sync occurs during each successful connection to district server. Credential expiry policy**:** Local credentials valid for 90 days offline. If no sync occurs within 90 days, user must connect to verify credentials. Permission changes sync immediately upon connection.
UI Responsiveness	Responsive design specifications: - Single-column layouts for all screen sizes to ensure usability on entry-level Android devices - Minimum supported screen size: 5-inch display (480x800 pixels) - Adaptive sizing using Android ConstraintLayout with density-independent pixels (dp) - Touch targets minimum 48dp x 48dp per Material Design guidelines

Component	Implementation Specification
	- Tested on: Samsung Galaxy A series, Tecno Spark series, Infinix Hot series (common in Uganda)

District Server Services
Core function: Data aggregation from facilities, Tier 2/3 forecasting, district dashboard for supply officers, and intermediate data hub for national reporting.

Component	Implementation Specification
API Gateway	FastAPI (Python) for RESTful endpoints managing mobile app sync, data validation, and communication with central hub. Includes rate limiting, request logging, and automatic retry handling.
Database	PostgreSQL 14+ for facility data, sync records, forecast results, and override audit logs. Partitioned by date for performance optimization.
Forecasting Engine (Tier 2)	Hierarchical time series models using statsforecast library (Python). Recommended models: Hierarchical Exponential Smoothing (HES) Rationale: Standard ETS requires separate models per facility, creating computational overhead. Hierarchical models generate forecasts for multiple facilities as a single model using top-down or bottom-up reconciliation, significantly reducing resource requirements while capturing cross-facility patterns.
ML Engine (Tier 3)	scikit-learn (Random Forest) or XGBoost for district-level predictive models identifying high-risk facilities for stockouts. Features include storage capacity, delivery lead times, facility type, budget allocation, historical override patterns.
Dashboard	Web-based interface built with Plotly Dash for district supply officers. Features: facility status map, trend graphs, override audit log, stockout risk alerts, and redistribution planning tools.
Deployment	Docker containers deployed on single server per district (or regional cluster for smaller districts). Includes automated backup to external storage, monitoring via Prometheus/Grafana, and GitLab CI/CD for updates.

Central Data Hub
Core function: National-level reporting, long-term trend analysis, integration with national health information systems, and policy-level system monitoring.

Component	Implementation Specification
Data Warehouse	PostgreSQL data warehouse with schemas for source_data, analytics, predictions. Star schema design for efficient querying of national trends.
ETL Pipeline	Apache Airflow orchestrating scheduled jobs to collect data from district servers (daily) and national systems (weekly). Includes data quality checks, deduplication, and transformation to standardized formats.
National System Integration	RESTful API integration with: - DHIS2: Pull aggregate health indicators, push supply chain metrics - e-LMIS: Pull national commodity consumption data, push facility forecast data - eAFYA: Pull patient visit data for demand modeling - Nutrition Appointment Platform (WHO/WFP/UNICEF/UNHCR): Pull nutrition program data, push supply adequacy indicators
National Dashboard	Web-based reporting interface for Ministry of Health officials showing national stockout trends, regional performance comparisons, forecast accuracy (MAPE), system adoption metrics, and override pattern analysis.

DATA SCHEMA DOCUMENTATION

Mobile SQLite Database Schema

The mobile database schema mirrors the district PostgreSQL schema for seamless synchronization. All timestamps stored in UTC. All tables include created_at and updated_at fields for sync conflict resolution.

- **facilities:** id (INTEGER PRIMARY KEY), name (TEXT), type (TEXT), storage_capacity_rating (TEXT), district_id (INTEGER), latitude (REAL), longitude (REAL), created_at (INTEGER), updated_at (INTEGER)
- **commodities:** id (INTEGER PRIMARY KEY), name (TEXT), code (TEXT UNIQUE), category (TEXT), unit (TEXT), created_at (INTEGER), updated_at (INTEGER)
- **stock_levels:** id (INTEGER PRIMARY KEY), facility_id (INTEGER), commodity_id (INTEGER), quantity (INTEGER), date_counted (INTEGER), user_id (INTEGER), sync_status (TEXT), created_at (INTEGER), updated_at (INTEGER)
- **orders:** id (INTEGER PRIMARY KEY), facility_id (INTEGER), commodity_id (INTEGER), system_quantity (INTEGER), final_quantity (INTEGER), override_reason (TEXT), override_justification (TEXT), date_created (INTEGER), sync_status (TEXT), user_id (INTEGER), created_at (INTEGER), updated_at (INTEGER)
- **consumption_history:** id (INTEGER PRIMARY KEY), facility_id (INTEGER), commodity_id (INTEGER), quantity_consumed (INTEGER), month (INTEGER), year (INTEGER), created_at (INTEGER), updated_at (INTEGER)
- **sync_log:** id (INTEGER PRIMARY KEY), facility_id (INTEGER), sync_type (TEXT), records_pushed (INTEGER), records_pulled (INTEGER), last_sync_time (INTEGER), status (TEXT), error_message (TEXT), created_at (INTEGER)
- **users:** id (INTEGER PRIMARY KEY), username (TEXT UNIQUE), password_hash (TEXT), role (TEXT), facility_id (INTEGER), last_login (INTEGER), credential_expiry (INTEGER), created_at (INTEGER), updated_at (INTEGER)

District PostgreSQL Database Schema

District database extends mobile schema with additional tables for aggregated analytics, forecasting results, and audit trails.

- **facilities**, **commodities**, **users:** Master data tables (same structure as mobile)
- **district_stock_levels:** Aggregated table with same structure as mobile stock_levels, populated from sync uploads
- **tier2_forecasts:** id, facility_id, commodity_id, forecast_quantity, forecast_month, forecast_year, confidence_interval_lower, confidence_interval_upper, model_type (TEXT), created_at, updated_at
- **tier3_predictions:** id, facility_id, stockout_risk_score (REAL), risk_factors (JSONB), prediction_date, model_version (TEXT), created_at
- **override_audit_log:** id, facility_id, user_id, order_id, system_quantity, final_quantity, override_reason, override_justification (TEXT), timestamp, reviewed_by (INTEGER), review_notes (TEXT), created_at

INTEGRATION & INTEROPERABILITY

This architecture explicitly avoids complex real-time integration at the facility level. Integration with national health information systems occurs at the district and central hub levels, not the mobile application.

Integration Architecture

- Mobile to District: Mobile app syncs directly with district server only. No direct connection to national systems.
- District to National Systems: District server acts as intermediary, pulling data from national systems (DHIS2, e-LMIS, eAFYA) and pushing aggregated facility data to central hub.
- Central Hub to National Systems: Central hub performs bulk data exchange with national health information systems using scheduled ETL jobs.
- Data format: JSON over HTTPS for APIs. CSV exports as fallback for Level 1 Integration Maturity facilities. FHIR-compliant resources where supported by national systems.

Security & Compliance

- Data in transit: TLS 1.3 encryption for all API communications
- Data at rest: AES-256 encryption for mobile SQLite database, full disk encryption for district/central servers
- Authentication: Role-based access control (RBAC) with facility staff, store managers, district officers, and national administrators
- Audit logging: All user actions logged with timestamp, user ID, and action details. Override audit log maintained indefinitely for accountability.
- Compliance: System designed to comply with Uganda's Data Protection and Privacy Act (2019) and Ministry of Health Digital Health Guidelines.

MONITORING, DEVOPS & SYSTEM GOVERNANCE

Monitoring & Performance Metrics

- Operational metrics:
  - Sync success rate (target: >95%)
  - Average time since last sync by facility
  - Mobile app crash rate (target: <1%)
  - District server uptime (target: 99%+)
- Forecasting metrics:
  - Mean Absolute Percentage Error (MAPE) for Tier 2/3 forecasts
  - Override frequency by facility and commodity
  - Stockout frequency and duration
- User engagement metrics:
  - Daily active users by facility
  - Paper form generation frequency (indicates trust in digital system)

Deployment & Update Strategy

- Mobile app: Distributed via Google Play Store for automatic updates. APK sideloading supported for facilities without Play Store access.
- District servers: Docker containerized deployment with GitLab CI/CD pipeline for automated testing and staged rollouts.
- Central hub: Blue-green deployment strategy for zero-downtime updates.

Stakeholder Governance Structure

Based on stakeholder feedback from the November 2025 validation event, the following governance structure will be established:

- District-level ownership: District Health Officers designated as primary system owners, involved as core partners throughout implementation and system refinement.

- Ministry of Health technical working groups: Framework to be presented to MoH thematic working groups for policy-level review and alignment with national digital health strategy.
- Quarterly review committee: Cross-stakeholder committee (MoH, district officers, facility representatives, implementing partners) to review model performance, override patterns, stockout trends, and recommend framework refinements.
- Frontline worker representation: Facility staff participation in design reviews and pilot validation to ensure system reduces rather than increases workload burden.

CONCLUSION & IMPLEMENTATION READINESS

This revised architecture incorporates all technical validation findings from Kyambogo University, stakeholder feedback from the November 2025 validation event, and recommendations from the validation matrix. The architecture has been validated across eight critical dimensions, with seven components fully validated and one (Tier 2 forecasting) validated with modification.

Architecture Validation Summary

- Offline-first functionality: Fully validated (Feasibility 5/5). SQLite with WAL mode addresses concurrent access concerns.
- Data security: Fully validated (Feasibility 5/5). Android Keystore System meets Uganda's Data Protection and Privacy Act (2019) requirements.
- Database performance: Fully validated (Feasibility 4/5). WAL mode enables safe concurrent operations with periodic optimization.
- Data archiving: Fully validated (Feasibility 5/5). Scheduled archiving maintains optimal performance.
- System interoperability: Fully validated (Feasibility 4/5). Standards alignment confirmed with MoH digital health strategy.
- UI responsiveness: Fully validated (Feasibility 5/5). Single-column adaptive layouts validated for common devices.
- Data governance**:** Fully validated (Feasibility 5/5). Continuous MoH oversight and compliance audits built into governance structure.

Design Principles Maintained

This architecture remains true to the core design principles identified in the baseline assessment:

- Resilient: Functions fully offline at facility level
- Practical: Uses simple, explainable forecasting where needed most
- Scalable: Districts can operate semi-autonomously
- Human-centered**:** Formally integrates human override into core data model and workflow

Proposed Post-Grant Implementation Roadmap (2026+)

Following technical validation, the recommended next step is a phased pilot implementation:

- Phase 1: Mobile application development with Tier 1 forecasting (2-3 months)
- Phase 2: Pilot deployment in 3-5 facilities per district across 2 districts (Karamoja region + southwestern region) for 3 months
- Phase 3: District server deployment with Tier 2/3 forecasting and dashboard (2 months)
- Phase 4: Pilot evaluation and framework refinement based on operational data (1 month)
- Phase 5: Scaled rollout with district-level ownership and MoH technical working group oversight

---
Document Version: Final (post-technical validation)
Date: November 2025
Authors: Gideon Abako, Timothy Kavuma, International Foundation for Recovery & Development
Validator: Ojok Ivan, Kyambogo University Department of Data Science, Networks & Artificial Intelligence

DOCUMENT REVISION HISTORY

Version	Date	Changes
1.0	Oct 2025	Initial architecture specification based on baseline assessment
2.0	Nov 2025	Post-validation revision incorporating: - SQLite WAL mode specification for concurrent access - Android Keystore System implementation details - Data archiving and purging policy - Tier 2 forecasting model change (HES/Panel ARIMA) - UI responsiveness specifications - Sync battery optimization conditions - Offline credential management details - Integration with Nutrition Appointment Platform - Stakeholder governance structure