

Local Food AI - Architecture Map

Project Deliverable

Course B1AI

Class:

Author: François LANGE

Date: 2026/06/19

Table of Contents

- System Component Architecture Page 2
- Database Design: Grouped Vertical Partitioning Page 2
- Dual-Mode Deployment Topology Page 2
 - Mixed Distributed Topology (Production/Staging Mode) Page 3
 - Resilient Single-Node Local Fallback (Offline Mode) Page 2

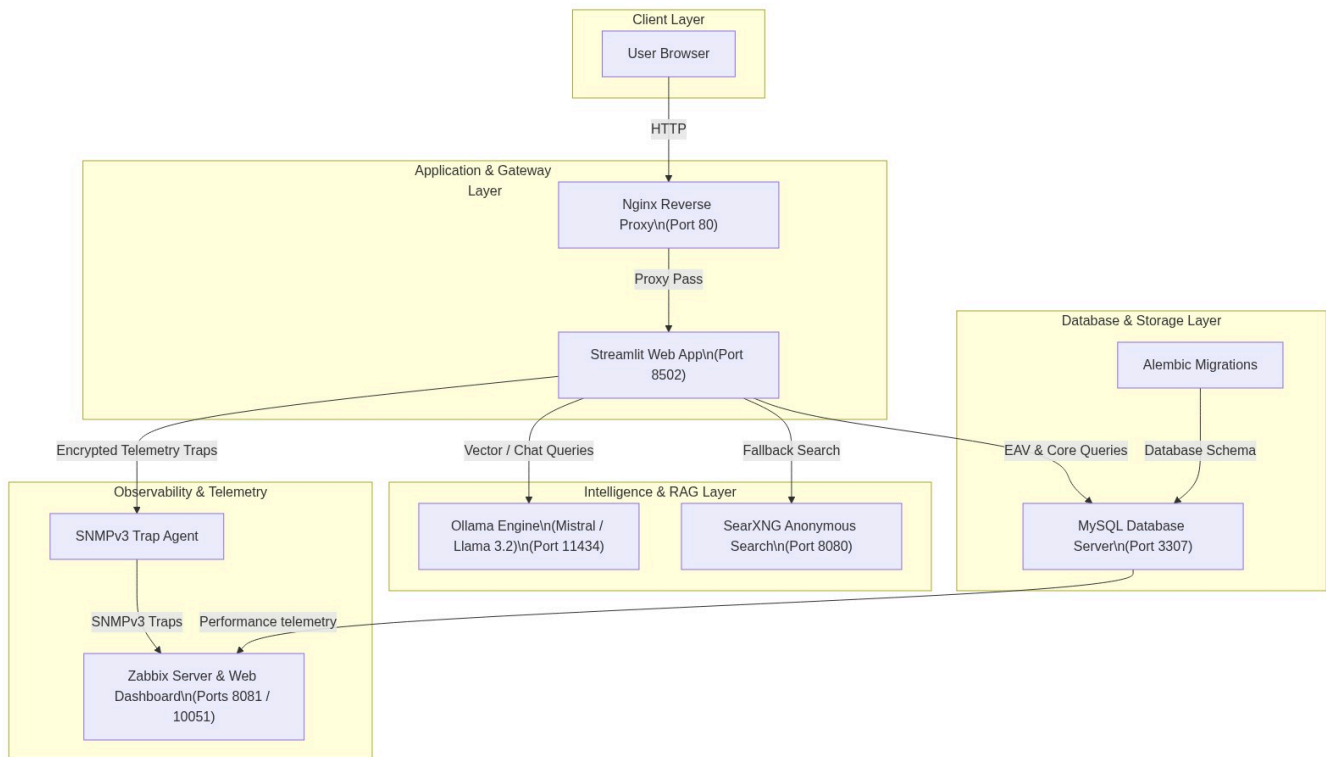
The current version is #ident "@(#)\$Format:LocalFoodAI_lanfr144:architecture.md:Francois Lange:lanfr144@school.lu:2026/06/19 08:08:39:Francois Lange:lanfr144@school.lu:2026/06/19 08:08:39:fe8669521370f31b75ae4710ff7a631eae4ac872:::\$"

Local Food AI - Architecture Map

This document describes the technical architecture, database schema design, AI RAG data flows, and dual-mode deployment topology for the Local Food AI clinical dietitian platform.

System Component Architecture

The platform is designed around a strictly local, privacy-first microservice topology. The components integrate seamlessly to provide nutritional search, RAG-augmented clinical diet evaluations, and DevSecOps observability.



Database Design: Grouped Vertical Partitioning

To optimize massive dataset ingestion (~24GB OpenFoodFacts dataset) and completely bypass InnoDB row size limits while maintaining sub-second RAG response times, the database utilizes a vertically partitioned structure:



- **products_core**: Contains product base information (barcode, name, brand, primary category) optimized with `FULLTEXT` indexing.
- **products_allergens**: Isolates complex ingredient list arrays and allergen keywords.
- **products_macros**: Implements double-precision floats (`DOUBLE`) for protein, carbs, fats, and energy metrics.
- **products_vitamins**: Stores micronutrient vitamin profiles.
- **products_minerals**: Stores trace mineral concentrations.

[!NOTE] All application search queries, RAG data tools, and ingestion processes interact with a unified database `view` named `products` which uses a series of high-performance `LEFT JOIN` operations across the primary key (barcode), shielding the frontend from database complexity.

Dual-Mode Deployment Topology

To ensure 100% resilience under network restrictions, the Local Food AI system is architected to operate under two distinct networking modes:

1. Mixed Distributed Topology (Production/Staging Mode)

Services can be distributed across native physical Unix/Linux servers (bare-metal), specialized local hypervisors, and Windows subsystems using bridged networking:

- **Database Node (Bare-Metal Server or Hyper-V VM)**: Dedicated bare-metal Ubuntu instance hosting the relational MySQL partitions at `192.168.130.170`.
- **Application Node (WSL 2 or Bare-Metal Unix)**: Runs the Streamlit frontend and local Ollama model engine.
- **Monitoring Node (VirtualBox VM or Bare-Metal Unix)**: Dedicated host running Zabbix Server and receiving SNMPv3 notifications.
- **Agile Scrum Tracker (Taiga)**: Remote agile project server at `192.168.130.161` for syncing deliverables.

2. Resilient Single-Node Local Fallback (Offline Mode)

When the remote VM host network or Taiga server is completely unreachable:

- **Zero-Dependency Containers**: The entire platform runs entirely locally on the notebook host via **Docker Compose** (`docker-compose.yml`).
- **Automatic IP Resolution**: Application configuration, Alembic, and SNMP notifications automatically adjust their endpoints to target local network interfaces (`localhost` / custom Docker networks) rather than unreachable remote IPs, avoiding timeout hangs or crashes.
- **Dynamic Task Tracking**: Agile development logs are dynamically synced into the workspace [task.pdf](#) and [walkthrough.pdf](#) artifacts to track progress until connectivity is restored.

Documented by Antigravity.

References

- **OpenFoodFacts Dataset & API Catalog**: Detailed food ingredients database. (<https://world.openfoodfacts.org/>)
- **Ollama Local LLM Inference Engine**: Lightweight instruction-following llama3.2 runtimes. (<https://ollama.com/>)
- **Zabbix Enterprise Telemetry and Monitoring**: System health and performance logging. (<https://www.zabbix.com/>)

Index

- **AI**: Page 2, Page 3, Page 4, Page 5
- **MySQL**: Page 3, Page 5
- **Zabbix**: Page 3, Page 4, Page 5
- **Docker**: Page 3, Page 5
- **Streamlit**: Page 3, Page 5
- **Nginx**: Page 5
- **RAG**: Page 2, Page 3, Page 5
- **Allergens**: Page 3, Page 5
- **Vitamins**: Page 3, Page 5
- **Minerals**: Page 3, Page 5
- **Clinical**: Page 2, Page 5
- **WSL**: Page 3, Page 5
- **Ollama**: Page 3, Page 4, Page 5
- **LLM**: Page 4, Page 5
- **Database**: Page 1, Page 2, Page 3, Page 4, Page 5
- **Security**: Page 5
- **Telemetry**: Page 4, Page 5
- **Backup**: Page 5
- **Firewall**: Page 5
- **SMTP**: Page 5