21 October 2025

Architecture for Real-Time Air Quality Monitoring and Personalized Recommendations

Authors: Johan Esteban Castaño, Stivel Pinilla, Jose Alejandro Cortázar

Introduction & Motivation

- Air pollution causes millions of premature deaths; Bogotá is highly impacted.
- Data is fragmented across multiple sources; users lack timely, actionable insights

Mission & Vision

Mission
Deliver innovative, usable air-quality software that provides clear, reliable, and accessible information for citizens, environmental agencies, and researchers in Bogotá.

Vision (2030)
Be Colombia’s leader in air-quality software, recognized for impact on decisions, research, and citizen empowerment.

Objectives

Delivering Innovative Air-Quality Software

General

Build a platform that integrates official data sources, offers intuitive visualizations, and enables informed decision-making.

Specific

Design accessible, usable interfaces for real-time air-quality information.

Implement analytics and indicators that support policy, research, and education.

Existing Platforms

Overview of current air quality solutions

AQICN

AQICN offers a global air quality index with extensive data coverage, but lacks personalized features for individual users seeking tailored health recommendations.

Google Air Quality

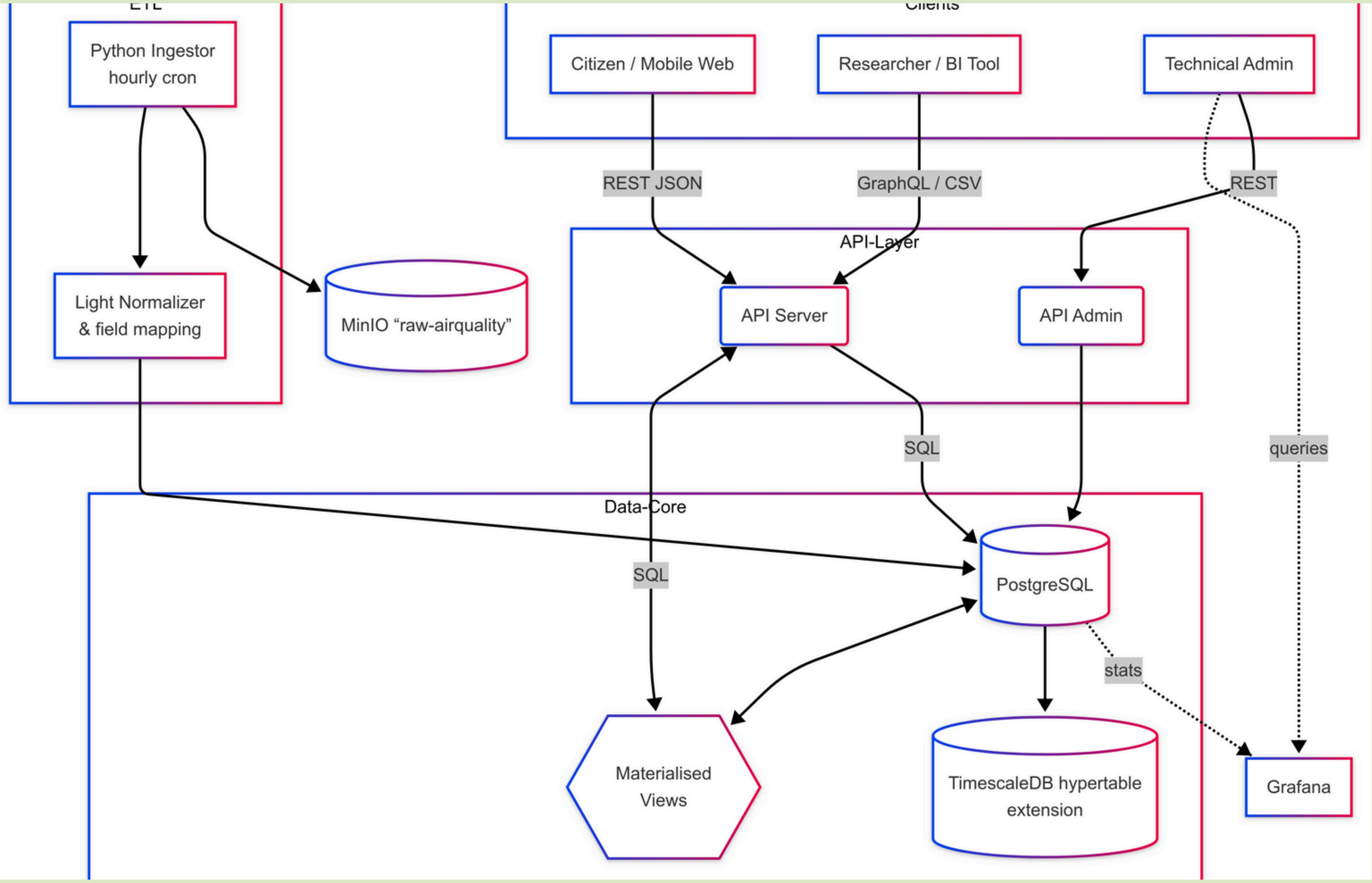
Google Air Quality provides real-time air quality readings with easy access, but is limited in data granularity and doesn't support extensive historical insights.

IQAir AirVisual

IQAir AirVisual delivers localized data and forecasts, yet operates under strict API quotas, making it less accessible for developers and researchers needing comprehensive data.

Proposed Architecture

- Python ingestor every 10 minutes (AQICN, Google, IQAir) → raw JSON in MinIO.
- Normalizer → TimescaleDB hypertables (monthly, by city).
- Concurrent materialized views → fast queries.
- REST/GraphQL API → dashboards and monitoring.



Data Model & Database

- 3 layers: Geospatial, Customer, Recommendation Engine.
- Main table: monthly-partitioned TimescaleDB hypertable; ≈2.5M rows/month (Bogotá).
- Partitioning improves pruning and concurrency; FKs ensure integrity.

Expected Results

Query Performance and Update Efficiency

Query Latency

Our goal is to achieve **sub-2 second** p95 query latency, even with over **1 million rows** and **1,000 concurrent users**, ensuring seamless access to vital information.

Update Frequency

We aim for a **10-minute update** frequency for personalized recommendations that align with the latest **health guidelines**, enhancing users' ability to respond to air quality changes.

Conclusions & Future Work

Conclusions

- Integrated, PostgreSQL/TimescaleDB-based stack can meet performance and usability goals for Bogotá.

Future Work

- Predictive models (ARIMAX, LSTM) for 6–24h forecasts.
- Read replicas, caching, and message queues for scale.
- Multi-city deployment in Latin America.

Thanks!!