

Architecture & ETL Data Flow

Cheng Han Chung

11/04/2025

Architecture & ETL Data Flow

01

Data Source

MongoDB stores agricultural readings
(e.g., temperature, humidity, etc.)

02

Cache Layer

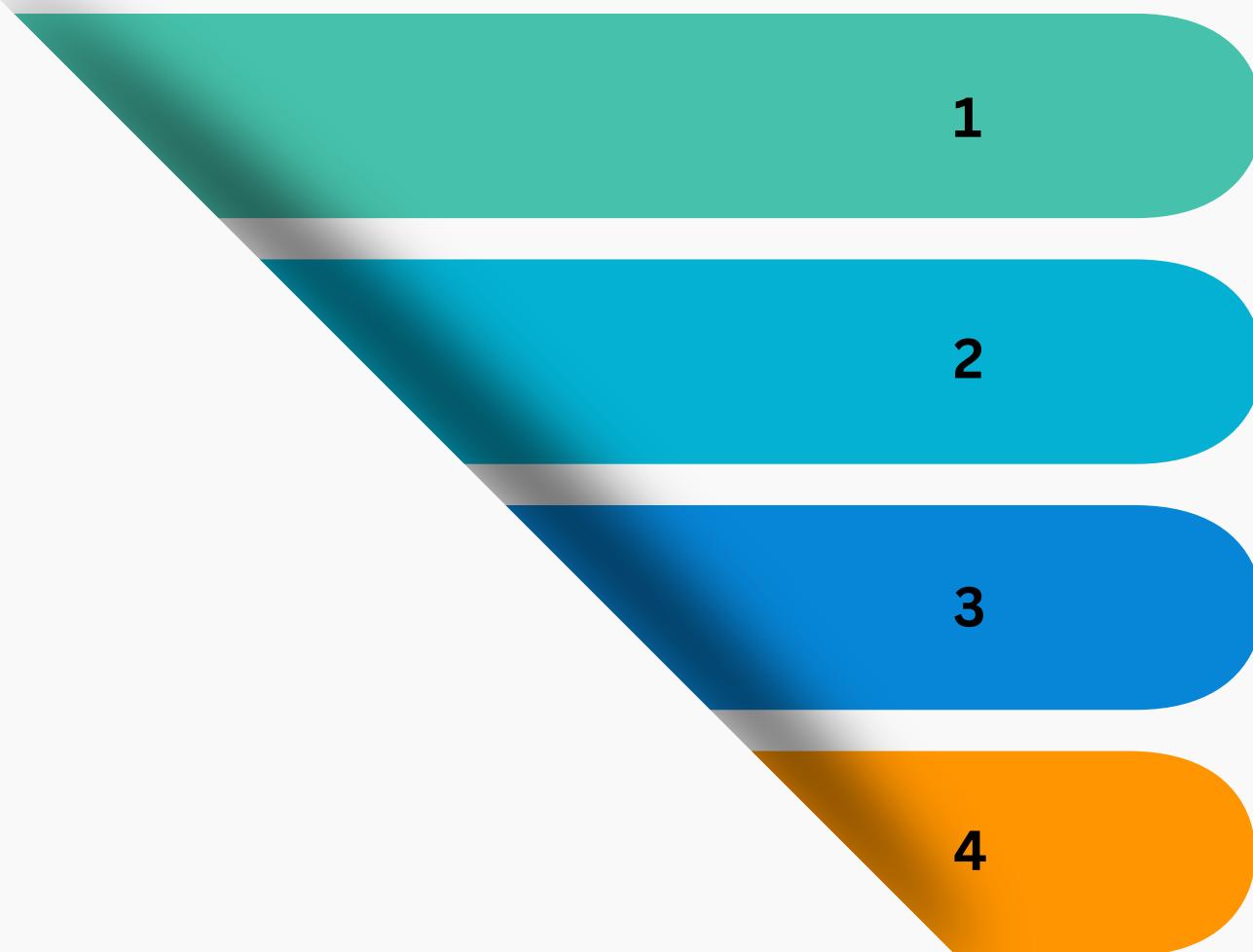
Redis is used to temporarily store
frequently accessed readings.

03

Flow Overview

- a. Client → API request (GET /cache-aside/:sensorId)
- b. App checks Redis cache
 - If hit → return cached data
 - If miss → query MongoDB, then store result in Redis
- c. TTL (Time-to-Live) ensures cache entries auto-expire and refresh from MongoDB.

Key Lessons from Implementation



Implementing caching significantly reduces response time (from ~80ms to <1ms after caching).

Cache-Aside pattern gives more control over when data is refreshed.

TTL (Expiration) helps maintain data freshness automatically.

Learned how Redis and MongoDB complement each other
– Redis for speed, MongoDB for persistence.

Challenges and Observations



Good things:

- Cache layer integration improved performance dramatically.
- TTL ensured automatic invalidation without manual cleanup.



Bad things:

- Initial connection setup, especially Redis + MongoDB credentials, took time to debug.
- Needed to handle missing cache carefully to avoid “stale data.”
- Performance varies slightly depending on network and dataset size.