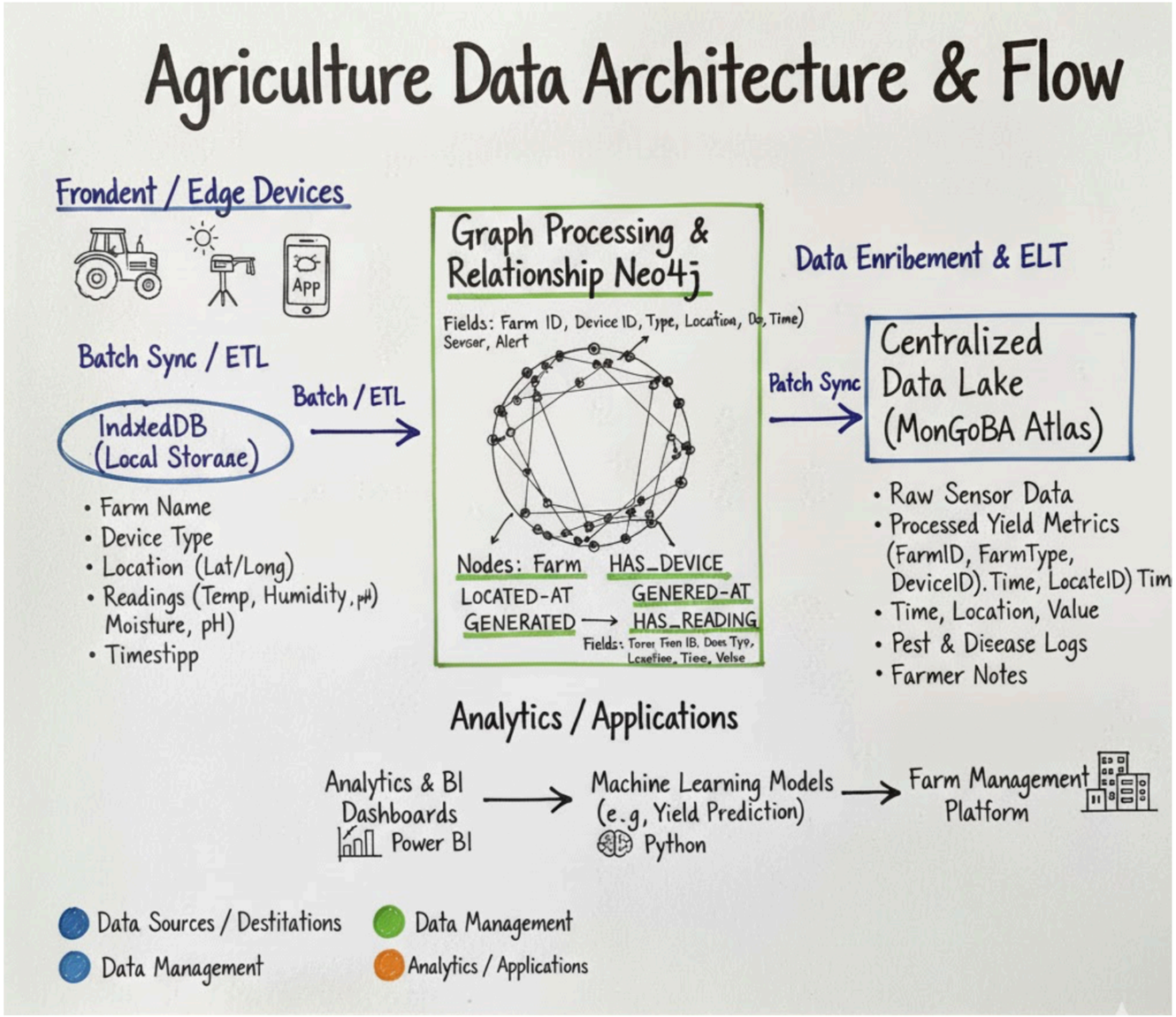
Lab 3

Architecture & Data Flow (IndexedDB + Neo4j → MongoDB lake)

Darshana Patil - 14/10/2025

Data Flow


Agriculture Data Architecture & Flow
Frondent / Edge Devices
App
Batch Sync / ETL
IndxedDB (Local Storaae)
Batch / ETL
• Farm Name
• Device Type
• Location (Lat/Long)
• Readings (Temp, Humidity, pH) Moisture, pH)
• Timestipp
Graph Processing & Relationship Neo4j
Fields: Farm ID, Device ID, Type, Location, Da, Time) Sensor, Alert
Nodes: Farm
HAS_DEVICE
LOCATED-AT
GENERED-AT
GENERATED
HAS_READING
Fields: Toror Tren IB, Does Tyo, Lcxefiee, Tiee, Velse
Data Enribement & ELT
Patch Sync
Centralized Data Lake (MonGoBA Atlas)
• Raw Sensor Data
• Processed Yield Metrics (FarmID, FarmType, DeviceID).Time, LocateID) Tim
• Time, Location, Value
• Pest & Disease Logs
• Farmer Notes
Analytics / Applications
Analytics & BI Dashboards
Power BI
Machine Learning Models (e.g, Yield Prediction)
Python
Farm Management Platform
Data Sources / Destitations
Data Management
Data Management
Analytics / Applications

Lessons Learned from extending the pipeline:

1. Data Transformation:
    - Successfully mapped Neo4j graph relationships to MongoDB document structure.
    - Maintained data integrity across different database.
    - How the connectivity from Neo4j to MongoDB done also how we can connect to Neo4j.
    - How to run it and see the expected output with all these fields.
2. Field Naming:
    - Simplified by using single, consistent field names
    - Clear field mapping improved data clarity

Challenges -

What went well ?

1. Data Pipeline:
    - Successful connection between Neo4j and MongoDB
    - Accurate data transfer with proper type conversion
    - Structured document format with clear hierarchy (farm → device → reading)
2. Error Handling:
    - Proper connection management with try/catch blocks
    - Clean resource cleanup in finally blocks
    - Connection validation before data transfer
3. Data Verification:
    - Immediate verification after insertion
    - Sample document display for quick validation
    - Document count tracking for transfer confirmation

What did not worked well?

1. Initial Field Mapping :
    - Increased complexity in data mapping
2. Timezone Handling:
    - Initial confusion with UTC vs local time.
        - As I observe it was showing date of next day then I finally understand this is happening because the nowUTC() function in your code is using UTC (Coordinated Universal Time). Since we are in a different timezone (likely PDT/PST), when you create a timestamp using new Date().toISOString(), it converts your local time to UTC. The UTC timezone is several hours ahead of your local time, which is why it's showing tomorrow's date when it crosses your local midnight.
    - Need for consistent time representation across databases
3. Data Visibility:
    - Initial difficulty in locating transferred data in MongoDB Compass
    - Need for more and proper understanding of how data and fields gets maps and flow from source to destination. Which I some what got after completing these labs.