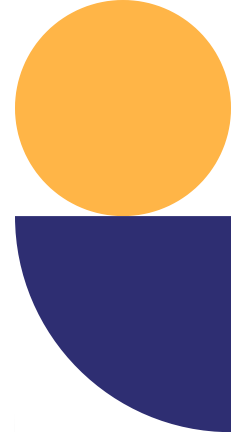


Building a Hybrid Data Pipeline with Neo4j and MongoDB

Data is the new oil. Insight is the new currency

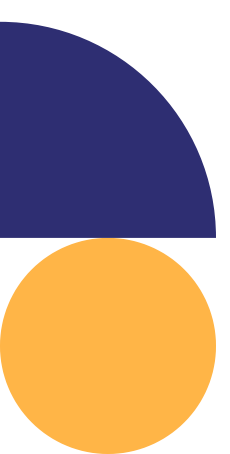




System Architecture and Data Flow (IndexedDB + Neo4j → MongoDB Lake)

This architecture demonstrates the data movement and integration process between three key components — IndexedDB, Neo4j, and MongoDB Lake — forming a hybrid data pipeline for scalable analysis.

Data Flow:

1. IndexedDB (Frontend Layer): Stores temporary or offline user data within the browser.
 2. Neo4j (Graph Layer): Converts user data into graph structures (nodes and relationships).
 3. MongoDB Lake (Storage Layer): Serves as the central data repository for analytics.
- 



Key Insights from Extending the Data Pipeline

Integration Insights:

Cross-Database Compatibility: Managing schema differences between Neo4j and MongoDB.

Data Transformation: Converting JSON data into graph structures improved understanding of relationships.

Performance Observations:

Neo4j excels at relationship queries; MongoDB at aggregation.

Indexing and caching reduced query latency.

Data Consistency:

Timestamp-based synchronization ensures updates remain consistent.

Automated ETL jobs reduce manual intervention.

Team Workflow Improvements:

Established CI/CD process for data updates.

Improved understanding of multi-database orchestration.





What Worked Well and What Did Not

What Worked Well:

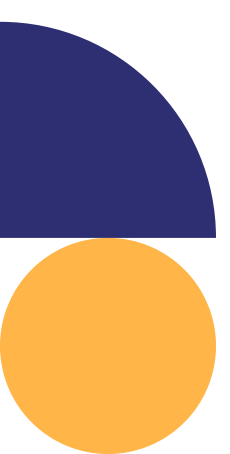
Neo4j visualization revealed hidden relationships.
MongoDB Lake enabled scalable queries.
Modular architecture simplified debugging.

What Did Not Work Well:

Complex ETL transformations increased maintenance cost.
Occasional data synchronization delays.
Schema drift across development environments.

Future Improvements:

Introduce Kafka or Airflow for automated streaming.
Implement schema validation and versioning.
Explore unified query layer across databases.





Thank You For Your Attention

Data is not about numbers, it's about understanding

