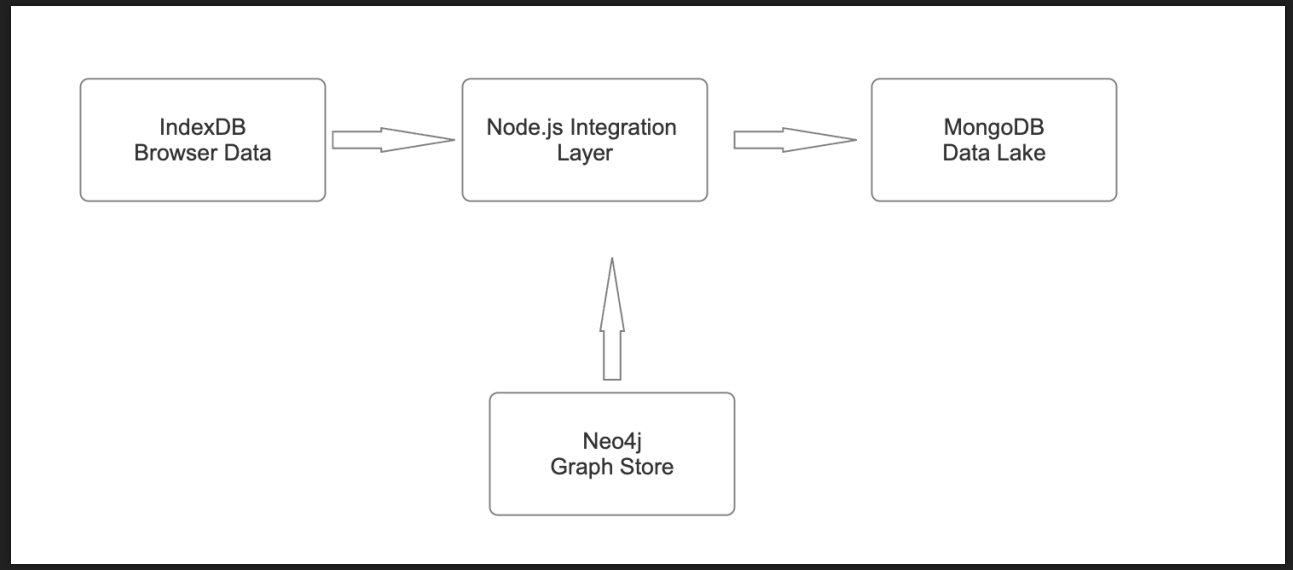
DBMS 2

Lab 3 – Rohan Jagdish Tilwani

Multi-Database Flow:
IndexedDB ↔ Neo4j → MongoDB Lake



- **Metadata Schema:**

{sourceDB, ingestedAt (UTC), tags [array], syncType {partial | full}}

- **Design Notes:**

Clear rectangular nodes and directional arrows.
Vertical and horizontal flow merged to emphasize integration hub (Node.js layer).
Node.js acts as middleware connecting both IndexedDB and Neo4j to MongoDB.

Reflections on Extending the Data Pipeline

Technical Insights:

- **Node.js Layer Control:** Central API streamlined data communication.
- **Metadata Validation:** Timestamp & source tracking prevented duplication.
- **Schema Mapping:** Adapted Neo4j edges to MongoDB documents.
- **ETL Optimization:** Async promises reduced latency and improved consistency.

Process Insights:

- **Debugging Workflow:** Incremental testing minimized failure rates.
- **Version Control:** Git-based iterations simplified pipeline evolution.
- **Documentation:** Enhanced handoff clarity and maintainability.

Successes and Challenges

Strengths	Challenges
Node.js middleware integration was reliable	IndexedDB limited bulk operations
REST endpoints made testing easier	Neo4j–Mongo schema alignment was tedious
MongoDB handled semi-structured data well	Race conditions in event-based sync
Metadata logging improved auditability	Network latency during multi-sync
Modular ETL functions were reusable	Handling API timeouts on local runs