LearnTrack Analytics Platform

1 Project Vision

LearnTrack is an analytics platform for online learning environments. Its main functions are:

- Streams of student activity logs (page views, quiz attempts)
- Real-time engagement scoring
- Flexible queries on student progress
- Batch analytics for course performance reports

2 Architecture Design Document

2.1 Chosen Technologies

Need	**Technology**	**Type**
Activities & profiles	MongoDB	Document-oriented
Real-time engagement scoring	Redis	Key-Value
Detect students accessing content from multiple regions	Neo4j	Graph
Generate daily course engagement reports	Cassandra	Column-oriented
Batch processing	Apache Spark	Processing framework
Visualization	Grafana	Dashboard / Visualization

2.2 Technologies explanation

MongoDB Document-oriented NoSQL database, well-suited for activity logs and user profiles thanks to flexible schemas, fast reads, and horizontal scalability (sharding, replica sets). From a CAP perspective, MongoDB is primarily AP (Availability + Partition Tolerance) with eventual consistency, but it can be tuned toward more CP-like behavior via read/write concerns.

Redis In-memory key–value store, schema-less and highly performant for direct access patterns (e.g., `studentId` → `score`), with effective horizontal scaling. In a Redis Cluster, partition tolerance is required, and the system favors availability over strict consistency, aligning with an AP profile.

Neo4j Graph database designed for highly connected data (nodes, relationships, properties). It is efficient for graph traversals and relationship-centric queries, making it suitable for fraud detection, social networks, and multi-region access analysis. Neo4j emphasizes strong consistency and availability (CA): replicas are kept in sync, and reads either return the latest write or an error.

Cassandra Distributed, column-oriented database optimized for large-scale, write-intensive workloads, time-series data, and evolving data models. It scales horizontally and supports high throughput. Apache Cassandra is classified as an AP system: it maintains availability and partition tolerance, accepting eventual consistency (some reads may temporarily miss the latest writes).

Apache Spark Distributed computing engine for large-scale data processing, leveraging in-memory computation for performance superior to disk-based systems. It provides a unified stack for batch processing, streaming, machine learning, and graph workloads, and is well-suited for aggregating and analyzing large datasets efficiently.

Grafana Visualization and monitoring platform supporting multiple data sources. It is used to build dashboards and alerts, enabling real-time monitoring of system health and business metrics.

3 Schema and Data Models

3.1 MongoDB Student Document

```
{
  studentId: "S1",
  name: "Hadil",
  email: "hadil@mail.com",
  speciality: "ILSI",
  studentLevel: "M2",
  progress: 10,
  lastUpdated: ISODate()
}
```

3.2 MongoDB Activity Document

```
{
  studentId: "S1",
  courseId: "C1",
  action: "page_view",
  timestamp: ISODate(),
  region: "EU"
}
```

3.3 Neo4j Graph Model

- **Nodes:**
  - `Student` – each student is represented by a unique node
  - `Activity` – each action (page_view, quiz_attempt) for a course
  - `Region` – geographical region of access
- **Relationships:**
  - `PERFORMED` – connects Student to Activity
  - `IN_REGION` – connects Activity to Region

3.4 Cassandra Table

```sql
CREATE TABLE course_engagement (
  courseid text,
  date date,
  count int,
  PRIMARY KEY (courseid, date)
);
```

4 Implementation plan

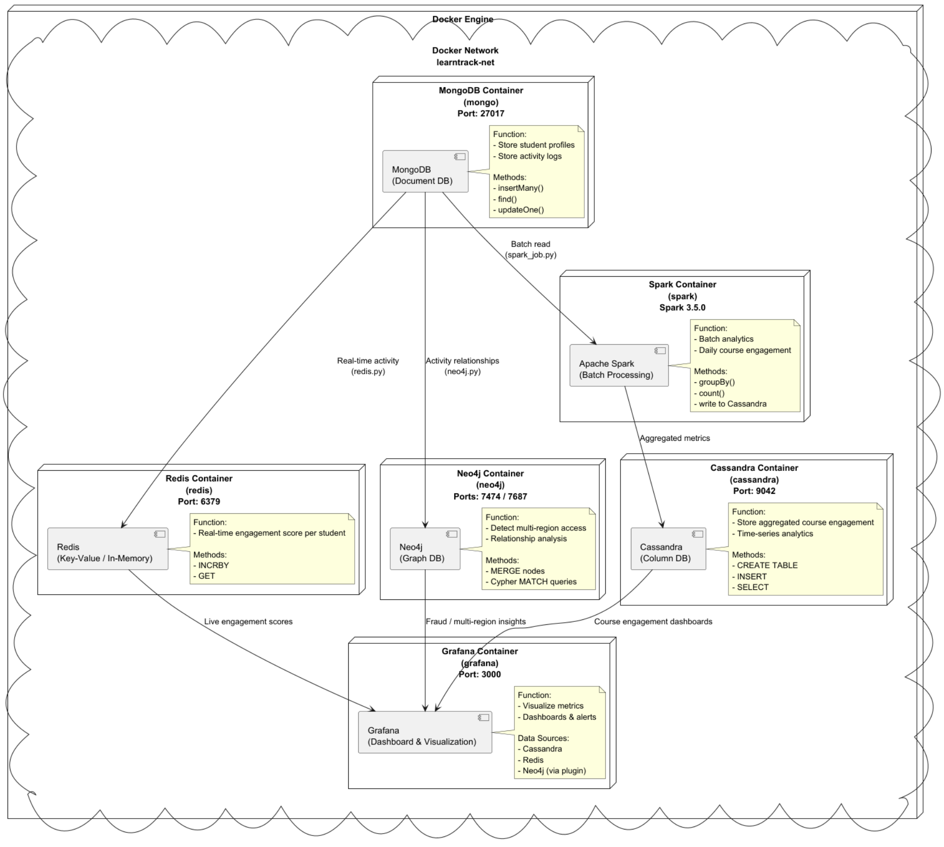

Figure 1: Data Flow in LearnTrack Analytics Platform

5 Conclusion

This polyglot architecture ensures data consistency, scalability, and performance while providing real-time insights, fraud detection, and batch analytics.