**SmartFleet – Fleet Management Microservices Platform**

**Overview:**

SmartFleet is a cloud-native fleet management platform designed to help logistics companies monitor, manage, and optimize their vehicle operations. The system must be built using **.NET 8.0**, leveraging **microservices architecture** with **Web API** for external communication and **gRPC** for internal service-to-service communication.

**Requirements:**

**1. Microservices Breakdown:**

* **Vehicle Service**: Manages vehicle metadata (registration, type, capacity, etc.).
* **Driver Service**: Handles driver profiles, licenses, and assignments.
* **Trip Service**: Manages trip schedules, routes, and status updates.
* **Telemetry Service**: Ingests real-time vehicle data (speed, location, fuel level) via gRPC.
* **Notification Service**: Sends alerts (e.g., maintenance due, route deviation) via email/SMS.

**2. Communication:**

* **External Clients** (e.g., mobile apps, dashboards) interact via **RESTful Web APIs**.
* **Internal Services** communicate using **gRPC** for performance and type safety.

**3. Data Storage:**

* Each service should use its own **database** (e.g., PostgreSQL or SQL Server) to ensure loose coupling.
* Implement **CQRS** in the Trip Service for read/write separation.

**4. Authentication & Authorization:**

* Use **JWT-based authentication**.
* Implement **role-based access control** (Admin, Dispatcher, Driver).

**5. Observability:**

* Integrate **OpenTelemetry** for distributed tracing and metrics.
* Use **Serilog** for structured logging.

**6. Deployment:**

* Containerize services using **Docker**.
* Orchestrate using **Kubernetes** or **Azure Container Apps**.

**🔧 Technical Constraints:**

* Use **.NET 8.0** with **Minimal APIs** where applicable.
* gRPC services must support **streaming telemetry data**.
* Ensure **idempotency** in Trip creation and updates.