**Software Engineering**

**Deliverable-3**

**Team Members**  
Adeel Mahmood Ansari 22i-0979

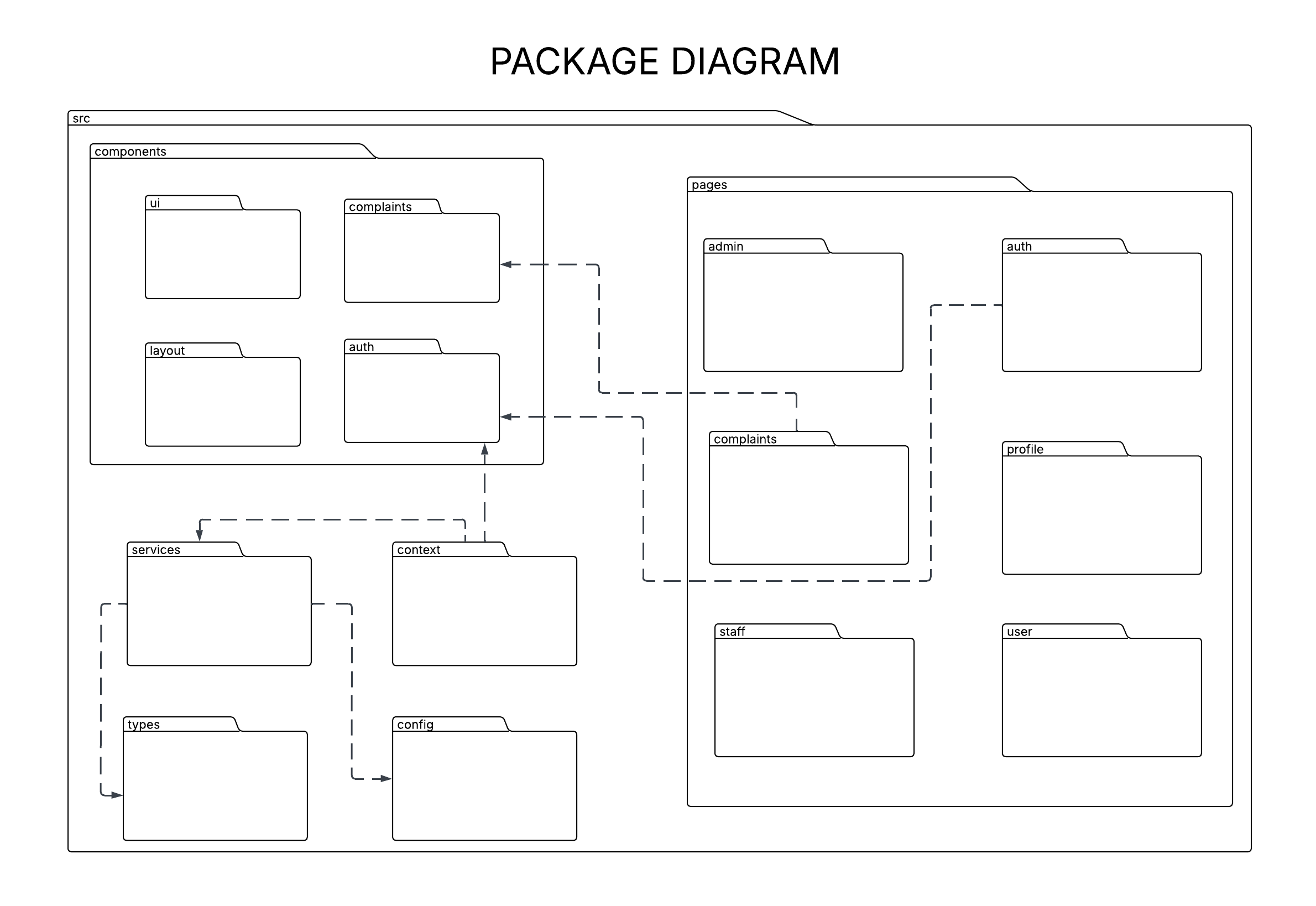
Abdul Wasay 22i-1198

Muhammad Shaif Imran 22i-1024

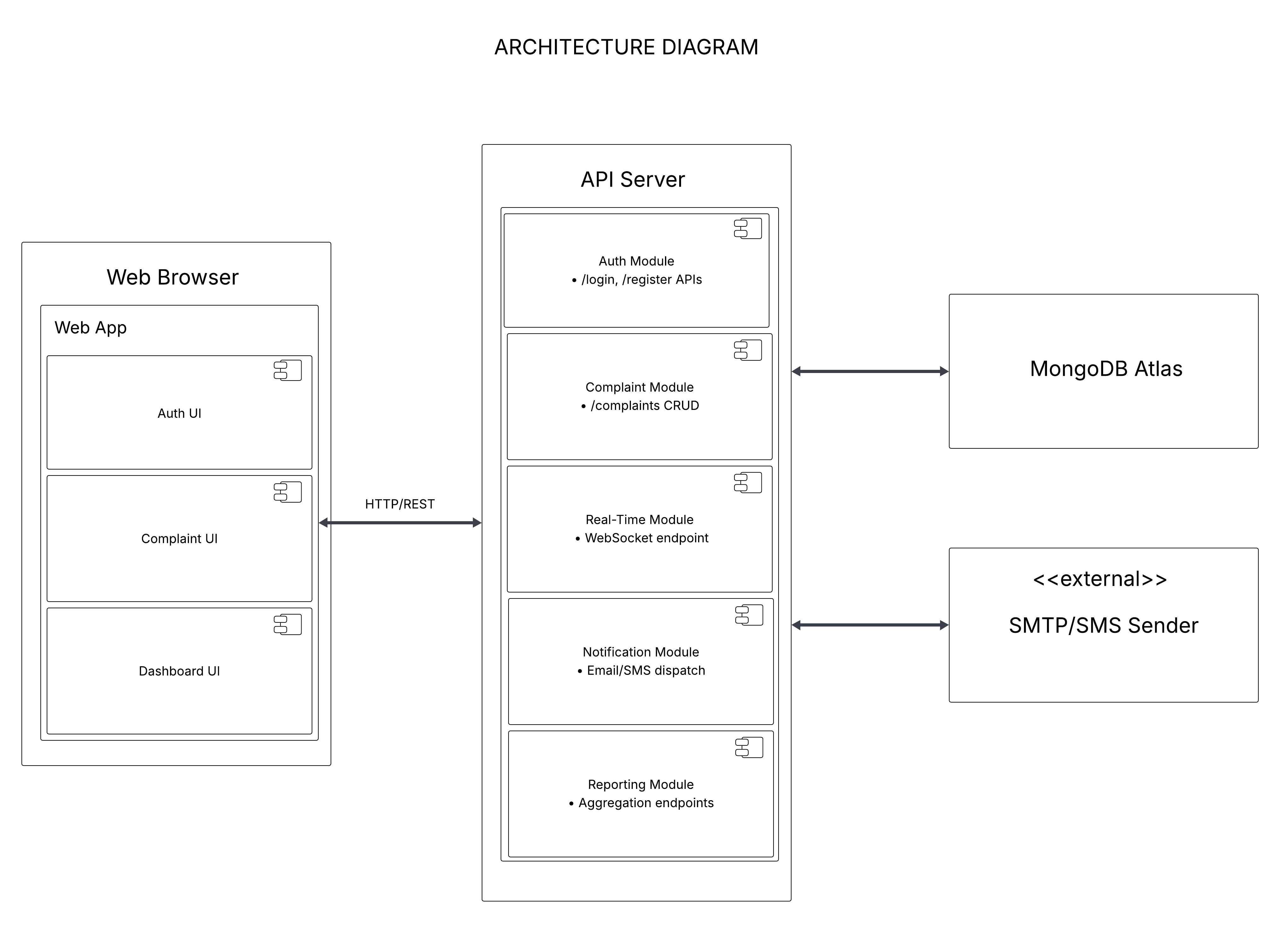
**System Architecture**

**Smart Complaint Management System**

**1. UML Package Diagram**

****   
Each subsystem is represented as a package. Arrows show dependencies, for example, the Reporting and Notification modules depend on Complaint data.

**2. Architecture Styles**



 **Client-Server Architecture**

* The system follows a **client-server model**, where a web browser (client) interacts with an API server (backend) through HTTP/REST.
* The client comprises multiple UI modules: Auth UI, Complaint UI, and Dashboard UI.

 **Modular Monolith / Microservices-Inspired**

* The API Server is structured into **independent functional modules**:
  + Auth Module
  + Complaint Module
  + Real-Time Module
  + Notification Module
  + Reporting Module
* This mimics microservices in modularity while retaining a single deployment unit.

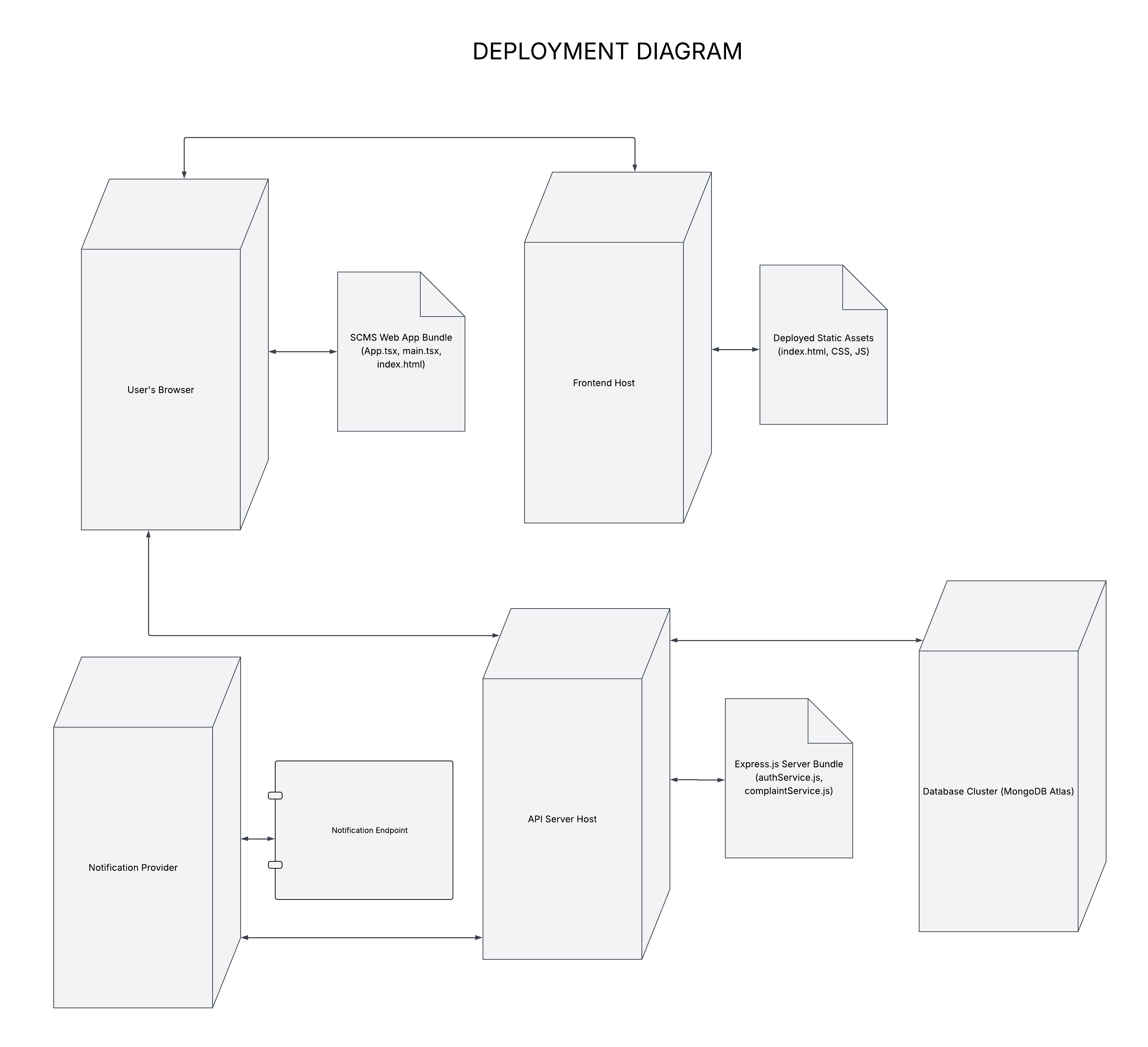
 **Layered Architecture**

* Each module within the API server follows a **layered approach**:
  + **Controller Layer**: Handles incoming requests.
  + **Service Layer**: Contains business logic.
  + **Repository/Data Layer**: Interacts with MongoDB Atlas.

 **Event-Driven Integration (for Notifications)**

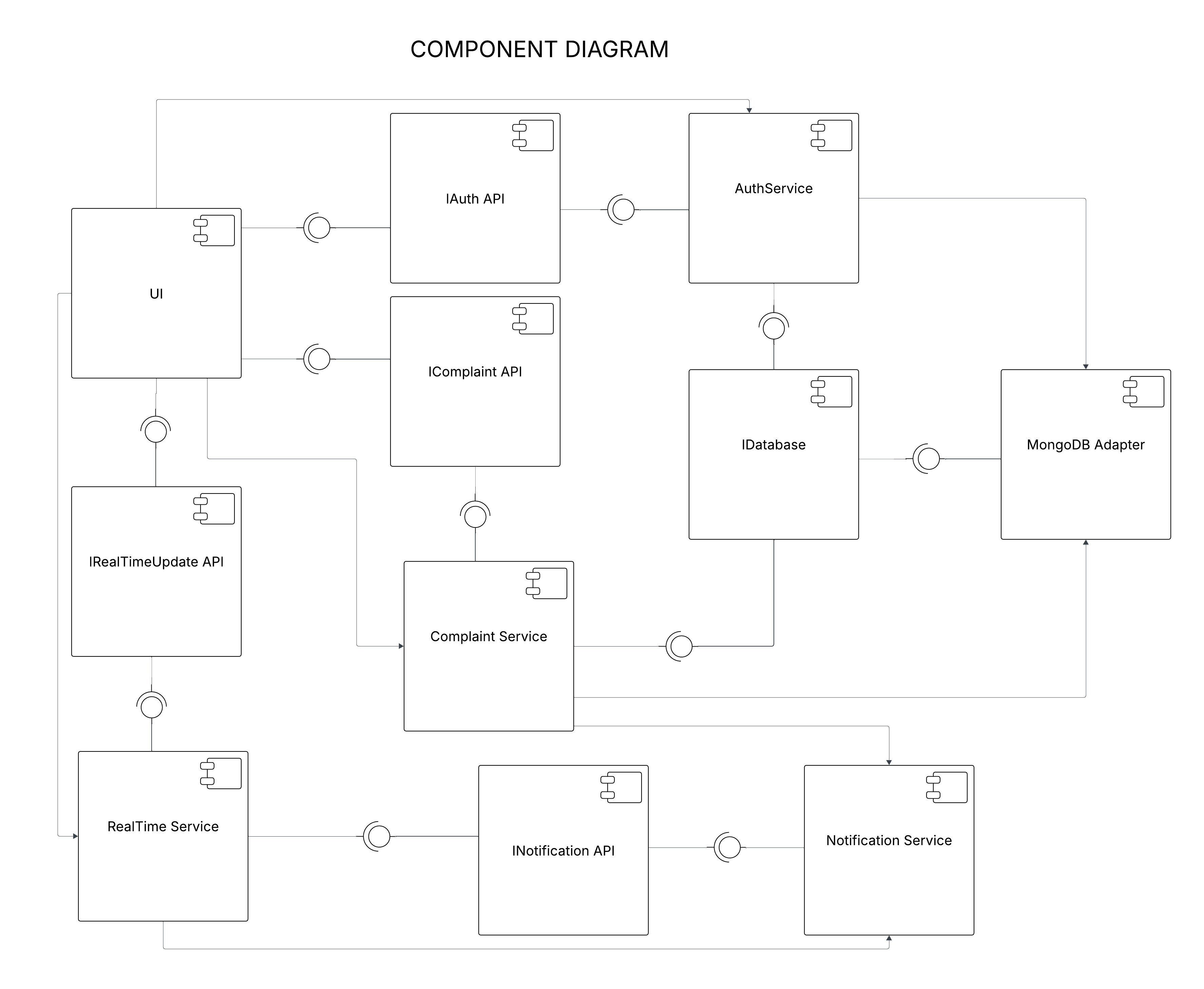
* The Notification Module is **loosely coupled** using an event-driven model (e.g., triggered on complaint status change).
* Integration with external services like SMTP/SMS is also abstracted.

**3. Deployment Diagram**

****

* The **Web Browser** hosts the Web App with UI components.
* The **API Server** hosts the logic:
  + Auth Module
  + Complaint Module
  + Real-Time WebSocket handling
  + Notifications
  + Reporting
* **MongoDB Atlas** stores application data.
* External **SMTP/SMS service** is used to send alerts.

**4. Component Diagrams [4 Marks]**

****