**Architecture Requirements Document (ARD)**

**1. Component Overview**

Our system architecture comprises several key components:

* **Frontend (ReactJS)**: The user interface component of the application, providing a graphical interface for pilots and control center operators to interact with the system.
* **Backend (Node.js/Express)**: Handles API requests, executes route planning computations, and manages interactions with the database and external services.
* **Database (MongoDB)**: Stores airport information, weather data, flight routes, user feedback, and other relevant data for the application.
* **External Services (Weather APIs)**: External weather APIs are utilized to obtain real-time weather data to supplement our route planning algorithm.

**2. API Design Principles**

Our APIs follow RESTful principles, providing clear and predictable endpoints for accessing weather data, airport information, flight routes, and route planning functionalities. Endpoint URLs are structured logically, using nouns to represent resources (e.g., /weather and /airport). We employ versioning in our APIs to ensure backward compatibility and facilitate future enhancements without breaking existing client applications.

**3. Scalability Requirements**

* **Horizontal Scalability**: Node.js clustering enables our backend services to scale horizontally by distributing incoming requests across multiple worker processes.
* **Database Scalability**: MongoDB sharding allows us to distribute data across multiple shards, ensuring efficient storage and retrieval of large datasets.
* **Frontend Scalability**: ReactJS enables us to build scalable and maintainable user interfaces by composing reusable components and managing state effectively

**4. Fault Tolerance Mechanisms**

In the event of service degradation or failure, our system gracefully handles requests by providing fallback responses or error messages to users. Retry mechanisms are implemented for handling transient errors, such as network timeouts or database connection issues, to improve overall system reliability. We employ monitoring tools to track system health metrics, detect anomalies, and trigger alerts for proactive intervention by the support team.

**5. Monitoring and Logging Practices**

Comprehensive logging is implemented throughout the system to capture key events, errors, and performance metrics for troubleshooting and auditing purposes. Monitoring tools are used to collect and analyze system metrics, including API usage, response times, error rates, and resource utilization, to ensure optimal performance and availability.

**6. Security Considerations**

Access to sensitive endpoints and data is restricted based on user roles and permissions, implemented using JSON Web Tokens (JWT) or OAuth2. Confidential data, such as user credentials and sensitive configuration parameters, are encrypted both in transit and at rest to protect against unauthorized access. Input validation mechanisms are employed to prevent injection attacks and ensure the integrity of user-supplied data.

**7. Continuous Integration and Deployment (CI/CD)**

Automated CI/CD pipelines are established to facilitate rapid and reliable deployment of code changes to production environments, ensuring minimal downtime and faster time-to-market. Unit tests, integration tests, and end-to-end tests are integrated into the CI/CD pipeline to validate code changes and prevent regressions before deployment.

* **Future Development Plans:**
* **Advanced Route Planning Algorithms**: We intend to explore and implement more advanced route planning algorithms to further optimize flight routes and minimize risks.
* **Enhanced Monitoring and Alerting**: In the future, we plan to enhance our monitoring and alerting capabilities by integrating more sophisticated tools and algorithms to detect and respond to anomalies in real-time.
* **Integration with Additional External Services**: We aim to integrate with additional external services, such as traffic management systems and airspace regulations databases, to provide more comprehensive and accurate flight planning information.
* **Enhanced User Authentication and Authorization**: Future development efforts will focus on enhancing user authentication and authorization mechanisms to provide finer-grained access control and improve overall system security.
* **Improved User Experience**: We plan to continually improve the user experience by incorporating user feedback, conducting usability studies, and implementing user interface enhancements based on industry best practices and emerging technologies.

**System Architecture Diagram**

