```mermaid deploymentDiagram title Deployment Diagram - Pro Mood Tracker

node "Client Device" {  
 component[React Native Application] as RNApp  
 artifact[Local Storage] as LocalDB  
 artifact[Device Sensors] as Sensors  
   
 RNApp -- LocalDB  
 RNApp -- Sensors  
}  
  
node "Firebase Cloud" {  
 component[Firebase Authentication] as Auth  
 component[Firebase Firestore] as Firestore  
 component[Firebase Storage] as Storage  
 component[Firebase Cloud Functions] as CloudFunctions  
 component[Firebase Analytics] as Analytics  
   
 Auth -- Firestore  
 Firestore -- Storage  
 Firestore -- CloudFunctions  
 CloudFunctions -- Analytics  
}  
  
node "External Services" {  
 component[Weather API] as WeatherAPI  
 component[Notification Service] as NotificationService  
 component[Maps/Location API] as MapsAPI  
}  
  
RNApp -- Auth : "HTTPS"  
RNApp -- Firestore : "HTTPS"  
RNApp -- Storage : "HTTPS"  
RNApp -- Analytics : "HTTPS"  
RNApp -- WeatherAPI : "HTTPS"  
RNApp -- NotificationService : "HTTPS"  
RNApp -- MapsAPI : "HTTPS"  
CloudFunctions -- NotificationService : "HTTPS"

}

## Figure 4.5: Deployment Diagram - Pro Mood Tracker

This deployment diagram illustrates the physical architecture and infrastructure of the Pro Mood Tracker application, showing how different components are distributed across the system.

### Components

#### Client Device

* **React Native Application**: The cross-platform mobile application that runs on users’ devices
* **Local Storage**: Persistent storage on the device for offline functionality and caching
* **Device Sensors**: Hardware components that provide data for contextual mood tracking (GPS, camera, etc.)

#### Firebase Cloud

* **Firebase Authentication**: Manages user authentication and session handling
* **Firebase Firestore**: NoSQL cloud database that stores user profiles, mood entries, and application data
* **Firebase Storage**: Cloud storage for media files like profile pictures and mood entry attachments
* **Firebase Cloud Functions**: Serverless functions that handle backend logic, data processing, and integrations
* **Firebase Analytics**: Monitors application usage patterns and performance metrics

#### External Services

* **Weather API**: Third-party service that provides weather data based on user location
* **Notification Service**: Manages push notifications for reminders and achievements
* **Maps/Location API**: Provides geographic data and location services

### Communication

* All communication between the client application and cloud services occurs via secure HTTPS connections
* The React Native application interfaces directly with Firebase services for data persistence and user authentication
* Cloud Functions act as intermediaries for complex operations and external service integration
* Local device storage provides offline capabilities when network connectivity is unavailable

This architecture provides a scalable, maintainable, and secure foundation for the Pro Mood Tracker application. The serverless approach using Firebase reduces operational complexity while providing robust cloud capabilities for data synchronization, authentication, and analytics.