deploymentDiagram  
 title Pro Mood Tracker - Deployment Architecture  
  
 node "User Devices" {  
 node "Mobile Device" {  
 component [React Native App] {  
 component [UI Components]  
 component [Business Logic]  
 component [State Management]  
 component [Data Sync]  
 }  
 component [Local Database] {  
 component [SQLite/Realm]  
 component [AsyncStorage]  
 component [Secure Storage]  
 }  
 [React Native App] --> [Local Database] : stores data  
 }  
 }  
  
 node "Firebase Cloud Platform" {  
 node "Firebase Authentication" {  
 component [Auth Services] {  
 component [Email/Password Auth]  
 component [OAuth Providers]  
 component [Phone Authentication]  
 }  
 }  
   
 node "Firebase Database" {  
 component [Firestore] {  
 component [User Collections]  
 component [Mood Records]  
 component [Settings]  
 component [Metadata]  
 }  
 }  
   
 node "Firebase Storage" {  
 component [Cloud Storage] {  
 component [Media Files]  
 component [Backup Data]  
 }  
 }  
   
 node "Firebase Functions" {  
 component [Cloud Functions] {  
 component [Data Processing]  
 component [Notifications]  
 component [API Integrations]  
 component [Analytics Processing]  
 }  
 }  
   
 node "Firebase Analytics" {  
 component [Analytics Engine] {  
 component [User Metrics]  
 component [Usage Patterns]  
 component [Performance Monitoring]  
 }  
 }  
   
 node "Firebase Hosting" {  
 component [Web Assets] {  
 component [Privacy Policy]  
 component [Terms of Service]  
 component [Help Documentation]  
 }  
 }  
 }  
  
 node "Third-Party Services" {  
 node "Push Notification Service" {  
 component [FCM/APNS]  
 }  
   
 node "Weather API Provider" {  
 component [Weather Data Service]  
 }  
   
 node "Location Services" {  
 component [Geolocation API]  
 }  
   
 node "App Stores" {  
 component [Google Play Store]  
 component [Apple App Store]  
 }  
 }  
  
 ' Connections between nodes  
 [React Native App] --> [Auth Services] : authenticates  
 [React Native App] --> [Firestore] : reads/writes data  
 [React Native App] --> [Cloud Storage] : uploads/downloads files  
 [React Native App] --> [Analytics Engine] : sends analytics  
 [React Native App] --> [FCM/APNS] : receives notifications  
 [React Native App] --> [Weather Data Service] : fetches weather  
 [React Native App] --> [Geolocation API] : gets location  
   
 [Cloud Functions] --> [Firestore] : processes data  
 [Cloud Functions] --> [Cloud Storage] : manages files  
 [Cloud Functions] --> [FCM/APNS] : triggers notifications  
 [Cloud Functions] --> [Weather Data Service] : enriches data  
   
 [App Stores] --> [React Native App] : distribute app

## Figure 4.16: Deployment Diagram - Pro Mood Tracker System

This deployment diagram illustrates how different components of the Pro Mood Tracker application are distributed across various physical and virtual environments.

### Key Deployment Environments:

#### 1. User Devices

* **Mobile Device**: The primary platform where users interact with the application.
  + **React Native App**: The cross-platform mobile application providing the main user interface and client-side functionality.
  + **Local Database**: Stores data locally for offline use and performance optimization.

#### 2. Firebase Cloud Platform

* **Firebase Authentication**: Manages all user authentication processes securely in the cloud.
* **Firebase Database (Firestore)**: Stores and synchronizes user data, mood records, settings, and metadata.
* **Firebase Storage**: Stores larger binary data such as images and backup files.
* **Firebase Functions**: Executes serverless functions for backend processing, notifications, and integrations.
* **Firebase Analytics**: Collects and processes anonymous usage data for improving the application.
* **Firebase Hosting**: Hosts static web content like documentation and legal pages.

#### 3. Third-Party Services

* **Push Notification Service**: Delivers timely updates to users’ devices.
* **Weather API Provider**: Supplies contextual weather information.
* **Location Services**: Provides geographical data to enhance mood context.
* **App Stores**: Distribution channels for the application.

### Key Deployment Characteristics:

1. **Cloud-Based Backend**: The system leverages Firebase’s managed cloud infrastructure, eliminating the need for custom server management.
2. **Client-Heavy Architecture**: Significant processing occurs on the user’s device, reducing cloud resource usage and enhancing privacy.
3. **Serverless Computing**: Backend logic is implemented as cloud functions that scale automatically with demand.
4. **Cross-Platform Deployment**: The mobile application is deployed to both iOS and Android platforms from a single codebase.
5. **Offline-First Functionality**: The local database ensures the application remains functional without internet connectivity.
6. **Seamless Synchronization**: Data synchronizes between local storage and cloud services when connectivity is available.
7. **Secure Authentication**: User identity management is delegated to Firebase’s battle-tested authentication services.

This deployment architecture provides high availability, scalability, and reliability while minimizing operational complexity. The system benefits from Firebase’s global infrastructure, automatic scaling, and built-in security features while providing a seamless user experience across different devices and network conditions.