EMIR Wear OS App — Software Design Document (SDD)

# 1. Purpose

The EMIR Wear OS app runs fully standalone on modern Wear OS smartwatches equipped with Wi-Fi or GSM/LTE. It collects location, heart rate, and activity data, uploads telemetry directly to the EMIR backend, and receives commands via push notifications using Firebase Cloud Messaging (FCM). No phone or tablet pairing is required.

# 2. Platform & Tech Stack

• OS: Wear OS 3+  
• Language: Kotlin  
• Location: FusedLocationProvider  
• Activity: ActivityRecognition  
• Heart Rate: Health Services API  
• Background Tasks: WorkManager, ForegroundService  
• Push Notifications: Firebase Cloud Messaging (FCM)  
• Secure Storage: Android Keystore  
• Minimum SDK: API Level 26+ (Wear OS 3+)  
• Connectivity: Wi-Fi, GSM/LTE  
• IDE: Android Studio + Gradle

# 3. Key Features

• Smart location tracking with motion-based GPS.  
• Continuous or interval heart rate monitoring.  
• Activity detection (walk, run, drive).  
• Battery-saving fused location.  
• Standalone data upload via Wi-Fi or GSM/LTE.  
• FCM push notifications for commands.  
• Secure storage with Keystore.  
• Local offline buffer and retry.  
• Fully independent from phone/tablet pairing.

# 4. Main Components

• LocationService: Uses FusedLocationProvider.  
• ActivityService: Uses ActivityRecognition.  
• HeartRateMonitor: Uses Health Services API.  
• DataUploader: Batches data and posts securely.  
• PushReceiver: Handles FCM wake-ups.  
• Local Storage: Room DB for offline cache.  
• Security: Manages auth and Keystore.

# 5. Connectivity

The app uses Wi-Fi or GSM/LTE for direct server communication. No dependency on Bluetooth pairing with a phone or tablet.

# 6. Push Notifications

• Uses Firebase Cloud Messaging (FCM).  
• Device registers token, uploads to backend.  
• Server sends minimal payload (command ID).  
• Watch wakes, fetches full command over HTTPS.

# 7. Security

• HTTPS (TLS 1.2+) for all server communication.  
• HSTS enabled.  
• Optional certificate pinning.  
• Keystore for API tokens and secrets.  
• Push messages contain no raw telemetry.

# 8. Battery Saving Strategy

• GPS runs only when motion detected.  
• Batching via WorkManager to reduce radio usage.  
• Uses Wear OS best practices for background tasks.

# 9. Data Fields

• Location: timestamp, latitude, longitude, accuracy, activity.  
• Heart Rate: timestamp, bpm, anomaly flag.  
• Activity: type (walk/run/drive).  
• Commands: command ID, status.

# 10. Build & Deployment

• Developed with Android Studio + Gradle.  
• FCM configured via google-services.json.  
• Distributed via Google Play or enterprise MDM.

# 11. Future Extensions

• SOS button to push instant location.  
• Haptic or voice feedback for commands.  
• Local user log of last received commands.