EMIR Standalone watchOS App — Software Design Document (SDD)

# 1. Purpose

The EMIR Standalone watchOS app runs directly on Apple Watch models with Wi-Fi or LTE, without requiring an iPhone or iPad. It collects location, heart rate, and activity data, uploads it securely to the EMIR backend, and receives commands via push notifications (APNs). Fully supports offline caching and retry when connectivity is interrupted.

# 2. Platform & Tech Stack

• OS: watchOS 8+  
• Language: Swift  
• Location: CoreLocation  
• Health: HealthKit  
• Activity: CoreMotion  
• Background Tasks: BackgroundTasks  
• Push Notifications: Apple Push Notification Service (APNs)  
• Secure Storage: Keychain  
• Connectivity: Wi-Fi or LTE  
• IDE: Xcode

# 3. Key Features

• Collects location using CoreLocation with significant-change monitoring.  
• Tracks heart rate via HealthKit.  
• Detects activity states (walk, run, drive) with CoreMotion.  
• Uses BackgroundTasks for batching and uploading.  
• Operates fully standalone with Wi-Fi or LTE — no phone needed.  
• Uses APNs push notifications to receive commands.  
• Stores API keys securely with Keychain.  
• Local offline cache for failed uploads.

# 4. Scenarios Supported

• Apple Watch with LTE: Fully standalone, direct upload and push.  
• Apple Watch with Wi-Fi: Same functionality over Wi-Fi.  
• No paired iPhone/iPad required for any core tracking or communication.

# 5. Main Components

• LocationService: Uses CoreLocation for smart GPS sampling.  
• HealthMonitor: Reads heart rate using HealthKit.  
• ActivityService: Uses CoreMotion for physical activity type.  
• DataUploader: HTTPS batch upload handler.  
• PushHandler: Receives APNs payloads and wakes app.  
• LocalStore: Queues telemetry if offline.  
• Security: Handles API tokens via Keychain.

# 6. Push Notifications

• Uses APNs with HTTP/2 for push delivery.  
• Registers device token directly from the watch.  
• Server sends minimal payloads (command ID).  
• Watch wakes, securely pulls full command data as needed.

# 7. Security

• All server communication uses HTTPS (TLS 1.2+).  
• HSTS enforced.  
• Optional certificate pinning.  
• No sensitive raw data in push payloads.  
• Tokens securely stored in the watch Keychain.

# 8. Battery Saving Strategy

• Location runs in significant-change mode for low power.  
• Batches data and defers uploads with BackgroundTasks.  
• Uses push notifications for commands — no persistent socket connections.

# 9. Data Fields

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

# 10. Build & Deployment

• Xcode project with watchOS target.  
• APNs certificates and provisioning profiles configured.  
• Distributed via App Store or managed device enrollment (MDM).

# 11. Future Extensions

• Emergency SOS tap to send instant location.  
• Complication or shortcut for quick status.  
• Local log/history view on the watch.