

AlvoTriX Core

Standalone Smartwatch Safety Monitoring

Version 3.5.0 • April 2026

Technical & Product Documentation

For Technical Managers and Guardians

1. Product Overview

AlvoTriX Core is a native Android (Wear OS) application that turns a standalone smartwatch into a continuous safety-monitoring device. Unlike the companion Gateway APK — which pairs with an external wearable over Bluetooth — Core runs directly on the watch itself and uses the watch's own hardware sensors, SIM, and internet connection.

The wearer (child, elderly person, vulnerable adult) puts on the watch and receives 24/7 passive monitoring. When anomalies are detected (fall, cardiac event, inactivity, curfew violation, etc.), Core fires two independent alert paths: an SMS straight from the watch's eSIM and an email to the guardian through the AlvoTriX server.

Design principles

- Zero interaction by the wearer — no app to open, no button to press in normal use.
- Graceful hardware fallback — any subset of sensors works; missing hardware reduces coverage but never crashes.
- Offline-first alerting — SMS works even without internet; server email is a secondary fail-safe.
- Privacy-first — only biometric signals are transmitted; no audio, no photos, no messages.

2. Device Compatibility

Core is a native Android app targeting Wear OS 2.0 and above (minimum API 27 — Android 8.1). It runs on any standalone smartwatch that ships with a Google-licensed Android derivative and allows APK sideload or Play Store install.

Officially supported (full feature set)

Watch family	LTE variant	SMS-direct capable
Samsung Galaxy Watch 4 / 5 / 6 / Ultra	Yes (LTE model)	YES
Google Pixel Watch / Pixel Watch 2 / 3	Yes (LTE model)	YES
OnePlus Watch 2	LTE model	YES
Mobvoi TicWatch Pro 5 / Pro 3	Bluetooth / Wi-Fi only	Server email path only
Fossil Gen 6	Wi-Fi only	Server email path only
Xiaomi Watch 2 / Watch S1 Pro	Wi-Fi only	Server email path only

Generic Android smartwatches (e.g. AQ7, Unisoc-based)	Depends on model	Depends on SIM presence
---	------------------	-------------------------

Not supported

- **Samsung Galaxy Watch Tizen OS (series 3 and earlier):** Tizen does not run Android APKs. Samsung switched to Wear OS from Watch 4 onward.
- **Huawei Watch GT series:** HarmonyOS is not Android.
- **Amazfit / Zepp OS devices:** Proprietary RTOS, no Android runtime.
- **Garmin / Fitbit:** Closed ecosystems, no third-party APK execution.

Minimum technical requirements

- **Operating system:** Wear OS 2+ / Android 8.1+ (API level 27 or higher)
- **CPU:** Any ARM / ARM64 chipset used in Wear OS watches (Snapdragon W5, Exynos W920, MediaTek MT2601, Unisoc sl8541, etc.)
- **RAM:** At least 512 MB
- **Storage:** ~10 MB free for the APK plus ~20 MB for the local event buffer
- **Mandatory sensor:** Accelerometer (used by every fall detection and activity algorithm)
- **Highly recommended:** Heart rate, GPS, barometer

3. Sensor Capture Pipeline

When the Core foreground service starts, it scans every hardware sensor exposed by the watch and selectively activates only the ones required by the wearer's active plan modules. This balances coverage with battery life.

3.1 Sensor discovery

AdaptiveSensorSampler performs a one-time scan and builds a map of available sensors, identified by Android sensor type IDs:

- **Standard Android:** Accelerometer, gyroscope, heart rate, magnetometer, barometer, ambient light, proximity, step counter, step detector, linear acceleration, gravity, rotation vector, off-body detector.
- **Samsung proprietary:** SpO2 (65572), skin temperature (65544), stress (65571), PPG raw (65573), blood pressure (65574).
- **Huawei proprietary:** SpO2 (70001), skin temperature (70002), stress (70003).
- **Generic manufacturers:** SpO2 (65536 / 65537), skin temperature (65538).
- **Catch-all:** Any unrecognised proprietary sensor is captured under a synthetic key "sensor_<typeld>" and forwarded to the server, which stores it verbatim so future analytics can decode it.

3.2 Sampling rates

Sensor	Internal rate	Rate sent to server
Accelerometer	50 Hz (fall detection)	2 Hz
Gyroscope	50 Hz	2 Hz
Heart rate	1 Hz	1 Hz
SpO2 / skin temperature	Event-based	Event-based
GPS	Every 10 seconds or 5 m	Every 10 seconds or 5 m
Step counter / step detector	Event-based (hardware trigger)	Event-based

Raw 50 Hz accelerometer is consumed only by on-device fall detection. The server receives a 2 Hz aggregated feed — plenty for biometric trending while keeping bandwidth and battery consumption low.

3.3 Module-based gating

Each alert module declares the sensors it requires. Only sensors needed by at least one active module are registered with the Android sensor framework:

- **Fall detection:** accelerometer, linear acceleration, heart rate, GPS
- **Panic button / SOS:** accelerometer, GPS, microphone
- **HR Guardian:** heart rate, accelerometer (for activity gating)

- **Health Crisis:** heart rate, SpO2, skin temperature, accelerometer
- **Sleep Monitor:** heart rate, SpO2, accelerometer
- **Activity Anomaly (inactivity):** accelerometer, step counter
- **Anti-Bullying:** accelerometer, gyroscope, heart rate, SpO2, GPS
- **Geofence (curfew):** GPS

4. Data Transmission to Server

Core transmits all collected telemetry to a dedicated ingest endpoint hosted at `activare.alvotrix.com`. The protocol is a simple HTTPS POST with an API-key header; no session management or OAuth is required.

4.1 Endpoint & authentication

- **URL:** POST `https://activare.alvotrix.com/ingest`
- **Authentication:** X-API-Key header containing the wearer's license key (format `ALVO-XXXX-XXXX-XXXX`)
- **Content-Type:** `application/json`
- **Payload:** `{ "batch": "<newline-separated JSON lines>", "source": "smartwatch" }`

4.2 Buffering and chunking

Samples are written to an on-disk LocalBuffer first, then flushed to the server every 30 seconds (configurable 5–60 s). Buffer limits prevent runaway growth if the network is unavailable:

- **Upload chunk cap:** 2000 lines per POST (prevents HTTP 413 from nginx/Flask)
- **Retained buffer cap:** 20,000 lines total (oldest dropped on overflow to protect memory)
- **On HTTP 413:** rejected chunk is dropped; leftover lines stay for the next cycle
- **Network-aware:** ConnectivityManager callback triggers an immediate flush when the connection comes back
- **Heartbeat:** if the buffer is empty, a tiny record with the latest HR plus battery level is still sent so the server knows the watch is alive

4.3 Critical readings bypass the 30-second cycle

When a critical value is observed (HR outside 40–150, SpO2 below 90, temperature outside 35–39 °C), Core triggers an immediate flush and runs the local alert evaluation in parallel, so a heart attack or oxygen desaturation reaches the server in under two seconds.

4.4 Device capabilities

On every successful license activation the capability cache is cleared, so the next service start re-scans hardware and re-uploads `device_capabilities` to the server (manufacturer, model, Android version, API level, sensor list). Guardians can therefore swap the watch hardware without the old capabilities sticking around.

5. Nine Alert Modules & Cooldowns

Every module runs locally on the watch, evaluated every five seconds by the service loop. When a condition fires, Core checks two independent cooldowns — one for the outgoing SMS (so guardians are not spammed) and one for the server event upload (so analytics do not see a flood of duplicates).

Module	Trigger condition	SMS cooldown	Event cooldown
Fall detection	Linear acceleration magnitude > 3.5 g	5 min	30 s
Panic / SOS	Shake, yelling, manual button	0 (instant)	0
HR critical high	HR >= age-adjusted hrMax + 30	10 min	2 min
HR critical low	HR <= hrMin - 10	10 min	2 min
HR high (tachycardia)	HR > hrMax AND wearer not recently active	30 min	10 min
HR low (bradycardia)	HR < hrMin (outside sleep window)	30 min	10 min
SpO2 low	SpO2 below age-adjusted threshold	30 min	10 min
Health Crisis	HR and SpO2 both in dangerous ranges at the same time	5 min	2 min
Sleep Monitor	Abnormal SpO2 / HR between 22:00 and 07:00	6 h	1 h
Inactivity	No significant movement for >= age-adjusted threshold	2 h	30 min
Curfew / Geofence	Wearer active between 22:00 and 06:00	8 h	1 h
Anti-Bullying	Impact >= 5 g + elevated HR between 08:00 and 16:00	10 min	2 min

Smart gating to reduce false positives

- **Activity gate:** tachycardia alerts do not fire if the wearer moved in the last 60 seconds (exercise).
- **Sleep gate:** low or high HR between 22:00 and 07:00 is routed to Sleep Monitor instead of the generic HR Guardian.
- **Plan gate:** every alert first checks hasModule() against the wearer's subscription — if the plan does not include the feature, nothing fires.

- **Off-body gate:** alerts suppress themselves when isOnBody reports that the watch is not being worn.

Age-adjusted thresholds

Age group	Heart rate range (bpm)	SpO2 threshold	Inactivity threshold
Child (0-12)	60 - 140	>= 92 %	4 hours
Teen (13-17)	55 - 130	>= 93 %	5 hours
Adult (18-64)	50 - 120	>= 94 %	6 hours
Senior (65+)	45 - 110	>= 90 %	3 hours

6. Dual-Path Alerting

Every fired alert attempts two independent delivery paths. If one fails, the other still reaches the guardian.

Path 1 — Direct SMS from the watch

On LTE watches with an eSIM or physical SIM, Core sends SMS straight from the device using Android's native SmsManager API. Internet is not required.

- Enumerates every active SIM via SubscriptionManager.activeSubscriptionInfoList.
- Builds a dedicated SmsManager per subscription ID so dual-SIM watches always pick a working channel.
- Splits long messages into multipart SMS (> 160 characters) automatically.
- Message template is identical to the Gateway APK — guardians receive consistent wording across families using both products.
- Example: "EMERGENCY ALERT: Albert Senopol has fallen at 13:45. Location: Strada Muntenia 12. Please check immediately. — AlvoTriX"

Path 2 — Server-mediated email and push

In parallel, Core POSTs to /panic/trigger on the AlvoTriX backend. The server then emails the wearer's emergency contacts and records the event for the guardian dashboard.

- **Endpoint:** POST <https://activare.alvotrix.com/panic/trigger>
- **Retry policy:** three attempts with exponential backoff (1 s, 2 s, 4 s)
- **Trigger taxonomy:** manual_button, fall, shake, yelling, health_crisis, sleep_spo2_low, curfew_violation, inactivity, bullying

SOS countdown

Body-detected triggers (fall, shake, yelling) run a two-second countdown before firing, giving the wearer a chance to cancel a false positive. User-initiated triggers (manual SOS button) use a five-second countdown.

- During the countdown the watch plays an audible alarm tone, vibrates in a distinctive pattern, wakes the screen, and speaks "Help is coming. Fall alert sent at <address>" once the alert goes out.
- Tapping CANCEL anywhere on the overlay aborts both paths.

7. Responsive User Interface

Wear OS watches ship in many form factors, from 1.2-inch round budget watches (240 × 240 px) to 1.9-inch flagship squares (480 × 480 px). Core uses a two-column layout on the home screen that guarantees the SOS button is always visible without scrolling.

Home screen layout

- **Top:** wearer name, centred and prominent
- **Left column:** TRANSMITTING / NOT TRANSMITTING status banner (green or red), live heart rate, “Last data: 5 s ago” timestamp
- **Right column:** large red SOS button plus a “Hold 2 seconds to send alert” hint
- **Footer:** “AlvoTriX Core” service tag in subdued grey

Screen size adaptation

- **Default dimens (values/):** compact enough to fit on 240 × 240 round watches
- **Round dimens (values-round/):** further tightened because BoxInsetLayout crops about 12 % off each side of the circle
- **Large dimens (values-sw200dp/):** generous spacing and larger text for high-end watches

The activation screen keeps a ScrollView so the Activate button remains reachable on the smallest round watches, but the home screen deliberately has no scroll — SOS must be one tap away at all times.

8. Service Resilience (Watchdog)

Core's foreground service must survive aggressive battery optimisations, Doze mode, and watch reboots. Multiple layers of redundancy keep it running:

- **Foreground service:** ongoing notification "Live monitoring active" that the OS cannot silently kill
- **Wake lock:** four-hour PARTIAL_WAKE_LOCK auto-renewed every three hours on the main handler
- **Watchdog alarm:** AlarmManager.setAndAllowWhileIdle() fires every five minutes and re-triggers the service; this survives Doze and App Standby
- **onTaskRemoved:** if the user swipes the app off the recents list, the service is restarted after five seconds
- **Boot receiver:** auto-starts the service on device boot, time zone change, or after an APK update
- **Network callback:** flushes the buffer as soon as connectivity is restored, so nothing is lost across Wi-Fi drop-outs
- **Session refresh:** every four hours Core re-fetches the wearer profile, picking up guardian-edited emergency contacts without needing a re-activation

9. Summary of Recent Fixes (v3.5.0)

Issue	Fix
Crash on Android 8.1 (API 27) because <code>setForegroundServiceBehavior</code> is API 31+	Call now guarded by <code>Build.VERSION.SDK_INT >= 31</code> check
<code>ClassCastException</code> after home screen redesign	Removed stale <code>ScrollView</code> cast in <code>applyScreenShapeInsets</code>
HTTP 413 — server rejected 16 000-line batches from the watch	Upload chunked to 2000 lines per POST; 413 drops the chunk and keeps the leftover
50 Hz accelerometer flooded the buffer in 30 seconds	Emit rate throttled to 2 Hz (internal fall-detection loop still sees 50 Hz)
Server expected <code>licenseCode</code> in JSON body, Core was sending empty <code>{}</code>	<code>WearerProfileManager</code> now posts <code>{ licenseCode: "ALVO-..." }</code>
Server returns <code>wearer</code> not <code>wearerInfo</code>	Profile parser updated with backwards-compatible fallback
Device capabilities were only sent once per install	<code>ActivationActivity</code> clears the cache on every license entry, so capabilities are re-sent
<code>/panic/trigger</code> only ever received <code>manual_button</code>	Real trigger value (fall / shake / yelling) is forwarded for analytics
Seven of nine alert modules were silently disabled because the server does not return a modules list on <code>/ingest</code>	Fallback list includes all nine modules by default when the plan does not specify otherwise
SOS button was hidden below the fold on small round watches	New two-column layout places SOS on the right, always in view
"Hold 2 seconds" hint invisible on dark background	Upgraded to bold light-grey #E0E0E0 for maximum contrast
Monitoring window and <code>monitoring_paused</code> flags were never saved locally	Fields now parsed and stored in <code>WearerProfileManager</code>

10. Closing Notes

AlvoTriX Core v3.5.0 reaches full feature parity with the companion Gateway APK, adapted for standalone Wear OS smartwatches. The same guardian accounts, alert templates, SMS logic, and biometric thresholds are shared between both products so a family can deploy either — or both in parallel — with a consistent experience.

For questions about device compatibility, deployment, or adding new alert modules, contact the AlvoTriX technical team. Source code for Core is maintained at the project's GitHub repository under the core-v3.5.0 branch.

— *End of document* —