An Open-Source Mobile and Wearable Sensing Platform for Bridging Passive Sensing and Active Sensing

Junmo Lee[1] · **Rd Pradipta Gitaya Samiadji**[1] · **Uichin Lee**[2]

[1] School of Computing, [2] AI Computing Department, Corresponding Author
Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Republic of Korea
junmo.lee@kaist.ac.kr · rd.pradipta@kaist.ac.kr · uclee@kaist.ac.kr

Paper

GitHub Page

Motivation: Bridging Passive & Active Sensing

Mobile sensing frameworks reduce the engineering burden of in-the-wild data collection, and support passive sensing (device usage, location, physiology) and active sensing (EMA). Both are indispensable for mHealth, yet they remain poorly integrated in long-term deployments.

Gap 1 **Real-time integration**
Context-sensitive EMA (CS-EMA) reduces recall bias, but existing platforms only trigger on low-level sensor values or a narrow set of contexts.

Gap 2 **Integration over time**
EMA compliance declines in long studies, leaving sparse active data. Smartwatch microEMA helps, but platforms that support it lack broad passive sensing or context sensitivity.

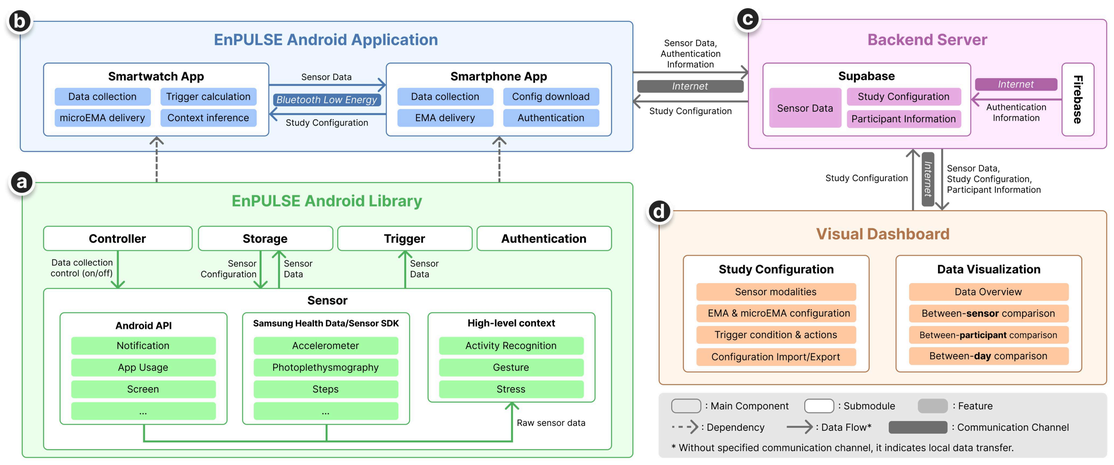
Gap 3 **Engineering burden**
Researchers must re-implement CS-EMA and microEMA on top of existing platforms—negating the reduced burden and reproducibility frameworks promise.

Comparison with Representative Platforms

Feature	AWARE	RADAR-base	Experiencer	EnPULSE
Smartphone passive sensing	✓	✓	✗	✓
Smartwatch passive sensing	Plugin	✓	✓	✓
Smartphone-based EMA	✓	✓	✗	✓
Smartwatch microEMA	✗	✗	✓	✓
Context for CS-EMA	Low-level	Low-level	Activity + low-level	**3 high-level sensors**
High-level context derivation	Server	Server	Not specified	**On-device**
Study configuration	Web	Web	JSON	**Web**

AWARE requires a separate plugin for smartwatch sensing. EnPULSE is the only platform combining smartwatch microEMA with on-device high-level context for CS-EMA.

EnPULSE Architecture



a EnPULSE Android Library
Bottom layer: 29 sensor modules (Android API, Samsung Health SDK, high-level context) plus controller, storage, trigger, and auth modules. EMA / microEMA are built in as sensor modules.

b EnPULSE Android Application
Watch app: wearable sensing, context inference, trigger calculation, microEMA. Phone app: config download, mobile sensing, EMA, CSV export / upload. Synced over BLE.

c Backend Server
Self-hosted Supabase (PostgreSQL) for participants, study configs, and sensor data; row-level security limits each user to authorized data. Firebase handles single-sign-on.

d Visual Dashboard
Web UI to configure sensors, EMA questions & schedules, and triggers; JSON import / export for reproducibility; data visualization view to monitor ongoing studies.

Live Demo: Three-Phase Walkthrough

1 Watch
Video of the visual dashboard: how studies are configured and how data from simulated large-scale studies is visualized.

2 Configure
Hands-on: build a study in the dashboard and compose a custom context-sensitive trigger binding high-level context to a microEMA.

3 Wear & Act
Put on the smartwatch and perform any action — on-device context sensing detects it and, if the trigger fires, a microEMA arrives on the watch.

High-Level Context Sensors (On-Device)

Inference runs on the device, not the server, so semantic context is available in near real time for CS-EMA triggering. Three sensors widely used in mHealth are implemented:

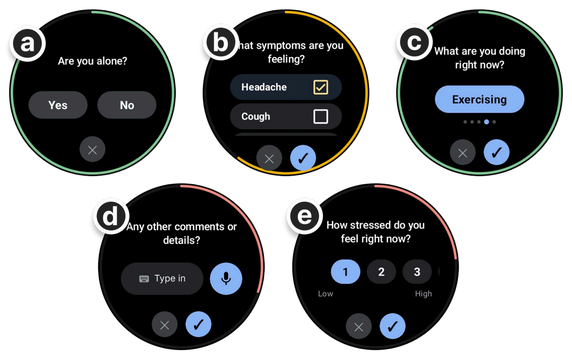
Activity Recognition
Google Activity Recognition API — 7 types: still, walking, running, on bicycle, in vehicle, tilting, unknown. Minimal resource use.

Gesture
WatchHAR model on IMU + voice. Voice is captured only after IMU alone detects an event of interest — privacy- and battery-aware.

Stress
RMSSD of IBI from PPG. ibiStatus filtering → feasibility filter (30 < BPM < 220) → median ± 3 MAD → 60 s window, 75 % overlap → personalized 20th-percentile threshold.

Smartwatch microEMA

Designed so that each question is answered with as few interactions as possible — interactions per EMA are a dominant factor in compliance.



4 scheduling modes
- Fixed time of day
- Randomized intervals (min / max bounds)
- Event-based via trigger system
- Manual triggering by participant

5 Question types
- Binary, Multiple choice, Single choice, Short-text, Number scale
- Supports multiple interaction methods

Trigger: Sense → Analyze → Act

SENSE Inputs from any sensor, incl. high-level context (gesture, stress, activity)

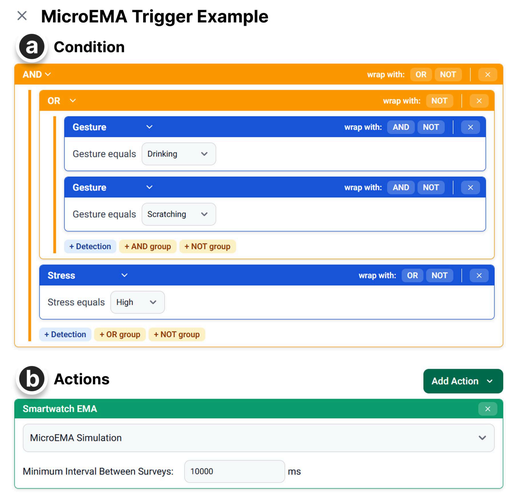
↓

ANALYZE Researcher-defined condition: atomic conditions combined with AND / OR / NOT

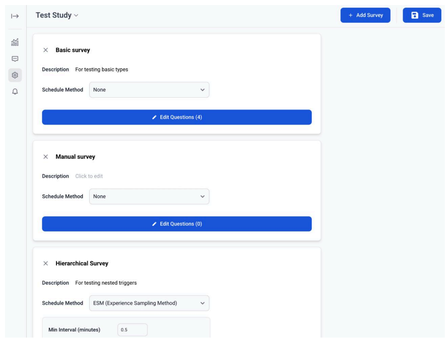
↓

ACT Deliver EMA / microEMA or broadcast an Android Intent for third-party apps

Example: (Gesture = Drinking OR Gesture = Scratching) AND Stress = High → deliver microEMA. A minimum interval between consecutive actions prevents over-triggering. Everything is configured in the visual dashboard — no code.

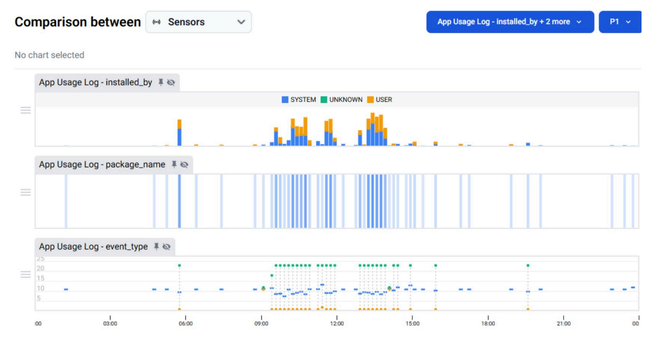


Web-Based Visual Dashboard



Study configuration: sensors, EMA & microEMA, triggers

- Configure sensor modalities, EMAs and trigger conditions without code
- Export / import config as JSON



Data visualization: between-sensor comparison

- Monitor collected data per participant & sensor in 2-hour slots
- Compare across sensors, participants, and days to diagnose missing data

Acknowledgments This research was supported by the Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korean government (MSIT) (RS-2025-02305801). We thank Sanghyeon Lee and Arslan Kenbayev for porting and training the WatchHAR model on Android.