SDNE Platform Issues Report v1.0

Samsung Galaxy XR platform limitations affecting clinical-grade neurologic exam capture

Sevaro Health

2026-05-26

1. Executive Summary

Sevaro built SENSE / SDNE — a Unity-based Standardized Digital Neurologic Exam — on Samsung Galaxy XR (Android XR). After ten rounds of on-device validation between February and May 2026, ten distinct platform-level issues have been identified that materially limit clinical-grade exam capture. Five of these resolve via existing partner-pathway agreements with Samsung Korea R&D and Google Android XR engineering (raw inward-camera access under research NDA, post-hoc Watch ↔ XR time synchronization, pass-through boundary exception, world-facing camera FOV characterization, and eye / face video segment export). The remaining five require new firmware, runtime-service, or sensor-platform work (compositor frame-pacing stall, undocumented eye-tracking precision, face-tracking service unreachable in Google's own first-party app, no real lower-body tracking, SLAM translation drift in passthrough). This document consolidates the on-device evidence behind each issue and maps each to a specific ask, drawn from `docs/SAMSUNG_XR_PLATFORM_ASKS.md` and the Cheri Malo correspondence thread from 2026-05-04 onward. Findings here are reported evidence-first and without advocacy; clinical context is provided per issue. The intended use of this document is to anchor the 2026-05-27 Samsung + Google working session in Plano, TX, and to serve as a reference artifact for downstream partner conversations.

2. Document Purpose & Audience

What this is. A consolidated record of the platform-level issues Sevaro has encountered while developing the SDNE neurologic-exam app on Samsung Galaxy XR. Each issue is grounded in measured on-device evidence (session JSON, logcat, profiling) rather than inference. Each is mapped to a specific ask — either a Samsung Platform Ask (numbered 1 through 7 in `docs/SAMSUNG_XR_PLATFORM_ASKS.md`) or one of the three research-partner pathway asks from the 2026-05-04 Cheri Malo thread (Cheri Asks #1 through #3, with #4 as a narrower fallback to #1).

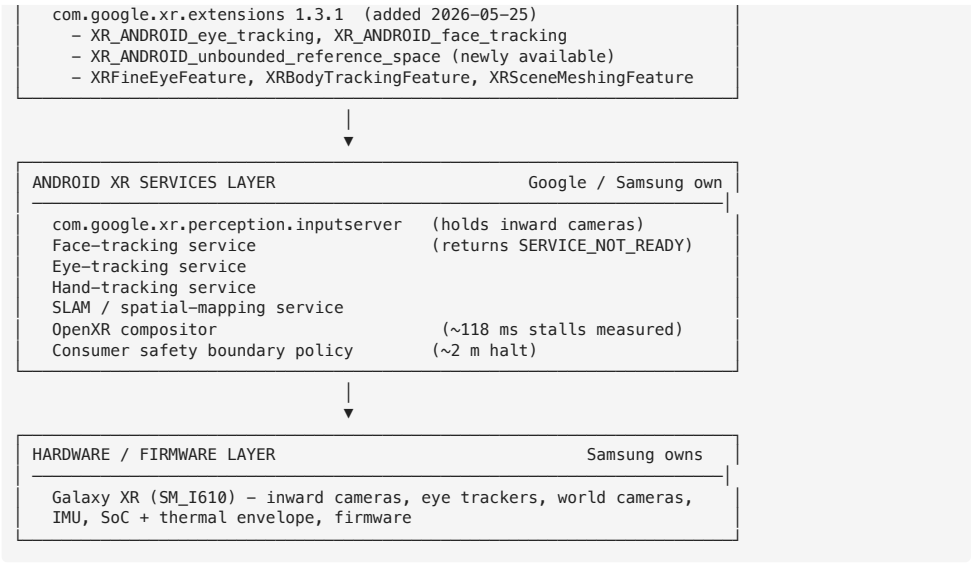
Who it's for. External technical partners. Specifically: Samsung Korea R&D (the May 27 Plano workshop audience), Google Android XR engineering (issue #21 / #22 / #23 maintainers on `android/android-xr-unity-package`), Dr. Jennifer Reneker / UMMC Center of Excellence (research-pathway collaborator), and future investors and clinical partners reviewing platform readiness.

How to read it. Section 3 sets up the platform stack diagram. Section 4 is the core inventory, organized by neurologic domain (Eye, Face, Motor, Gait, Audio). Section 5 covers the three research-partner pathway proposals. Section 6 is the issue × ask cross-reference table. Section 7 is methodology. Appendices A, B, and D follow. (Appendix C, Glossary, is deferred to v1.1.)

Version 1.0, 2026-05-26. Status fields in Section 4 will be updated post-workshop.

3. Platform Architecture Overview

```
┌────────────────────────────────────────────────────────────────────────┐
│ APPLICATION LAYER                                          Sevaro owns  │
│ ──────────────────────────────────────────────────────────────────────│
│   SDNE Unity application (com.sevaro.sdne / com.sevaro.sense)           │
│     - 25 exam tasks across 7 neurologic phases                          │
│     - Per-task scoring, threshold logic, session JSON emission          │
│     - AWS Lambda upload pipeline, dashboard at sense.neuroplans.app     │
└────────────────────────────────────────────────────────────────────────┘
                                  │
                                  ▼
┌────────────────────────────────────────────────────────────────────────┐
│ UNITY / OPENXR LAYER                                 Unity / Google own  │
│ ──────────────────────────────────────────────────────────────────────│
│   Unity 6 (6000.3.6f1) + Universal Render Pipeline                      │
│   com.unity.xr.openxr 1.16.1 / 1.17.0-pre.2                             │
│   com.unity.xr.androidxr-openxr 1.3.0-pre.1 / 1.3.1                     │
```



The application layer is fully under Sevaro's control. Issues at this layer are not subjects of this report. The Unity / OpenXR layer is partially under Sevaro's control (we choose package versions and select features) but the underlying extension definitions and runtime are Google's. The Android XR services layer is Google's and Samsung's joint responsibility, and is where the majority of issues in this report surface. The hardware / firmware layer is Samsung's. Each issue in Section 4 is labeled with the layer at which the block originates.

4. Issues by Domain

For each issue: a one-sentence headline with the headline measurement; the evidence trail (file path or session ID); the root cause as measured (not guessed); the ask that resolves it; current status; and the clinical impact in one or two lines.

4.1 Eye / Oculomotor

Issue 1 — Compositor stutter corrupts gaze sample stream

- **Headline:** ~118 ms main-thread stall every ~0.5 s, producing 147 spurious "saccadic intrusions" in a 12-trial healthy saccade run.
- **Evidence:** Session JSON `SDNE_Core15_77e2af69-2524-4178-8626-9f93f58d2385_20260526_024516.json` (2026-05-25 healthy run). Task T04 IntrusiveSaccadeCount = 147 over 12 trials; task T05 SaccadicIntrusionCount = 81 over 30 s pursuit. Per-subsystem profiling: eye / face / hand / UI / render / task subsystems all measured at ≈0.1 ms/frame. The 118 ms stall sits inside the OpenXR runtime frame-submission path, not in app code. The OpenXR runtime reports `Confidence: 0.95` and `IsLikelyHeadPoseFallback: false` throughout the corrupted stream — neither runtime signal exposes the failure.
- **Root cause:** Galaxy XR OpenXR compositor frame-pacing stall, specific to passthrough rendering. During the stall the gaze sample does not update; when it does, the angular delta between stale and fresh sample over the elapsed Δt computes to a velocity > 30 °/s and is counted as a saccade onset.
- **Ask:** Samsung Platform Ask 1 (TOP priority). Secondary ancillary ask: gaze-sample timestamp + validity flag so applications can detect stale-stall samples directly instead of inferring from intrusion count.
- **Status:** Open. Already raised to Google in issue #21 (`android/android-xr-unity-package`) since 2026-04-16, no response.
- **Clinical impact:** Healthy patient's exam reads Oculomotor RED on the live dashboard (false pathology). Saccade peak velocity, pursuit gain, convergence angle, and nystagmus detection are all unreliable until resolved. The current SDNE build mitigates by gating to INVALID on intrusion count > 30 (saccades) or > 50 (pursuit) — a downstream workaround.

Issue 2 — Eye-tracking sample rate, latency, and accuracy undocumented

- **Headline:** `XR_ANDROID_eye_tracking` publishes no sample-rate / latency / accuracy specs; FINE and COARSE modes have no documented numerical differences.
- **Evidence:** Khronos OpenXR registry entry for `XR_ANDROID_eye_tracking`; developer.android.com/develop/xr documentation; verified absent from `com.unity.xr.androidxr-`

`openxr` 1.3.1 source.

- **Root cause:** Platform documentation gap. Clinical nystagmus (~2-5 Hz oscillation) requires Nyquist > 10 Hz with sub-degree precision; saccade peak velocity requires sub-millisecond timing. Neither is specified.
- **Ask:** Samsung Platform Ask 2.
- **Status:** Open.
- **Clinical impact:** Without documented precision, quantitative oculomotor metrics (saccade velocity, pursuit gain, nystagmus frequency) cannot be reported with defensible accuracy.

Issue 3 — Raw eye-camera access not available

Covered in Section 5.1 (Cheri Ask #1) — research-partner pathway.

4.2 Face

Issue 4 — Face-tracking service returns `XR_ERROR_SERVICE_NOT_READY_ANDROID`

- **Headline:** All 68 ARKit-style blendshapes return 0.0; `valid_frame_ratio = 0` across a full T07 task; same error reproduces inside Google's own first-party calibration app.
- **Evidence:** `docs/UNITY_BUG_face_tracking_service_not_ready.md`. Confirmed 2026-05-25 PM. Permissions verified granted via `adb shell cmd appops` (FACE_TRACKING, EYE_TRACKING_COARSE, EYE_TRACKING_FINE all `allow`). System face calibration completed via both `Settings > Avatars > Calibrate` and `com.google.xr.facetracking.calibration/.FaceTrackingCalibrationActivity`. Headset worn (awake) throughout. Logcat shows `E .../FaceProvider.cpp:351 Face state data was unavailable. Could not retrieve region confidence data.` and `XR_ERROR_SERVICE_NOT_READY_ANDROID in xrGetFaceCalibrationStateANDROID` every session. **Critical:** running the first-party Google calibration app itself, *its own* logcat (under its own PID) spams the same `XR_ERROR_SERVICE_NOT_READY_ANDROID` hundreds of times.
- **Root cause:** Face-tracking service unreachable at the device / runtime / service layer. Upgrading to `androidxr-openxr` 1.3.1 cleared the application-side error path (more graceful `TryGetIsFaceTrackingCalibrated`) but blendshape data remained all-zero. The failure reproduces in Google's own first-party app — conclusively a Galaxy XR runtime / service-level bug, not specific to Sevaro's app or to Unity.
- **Ask:** Samsung Platform Ask 7 (added 2026-05-25 after the first-party reproduction was observed).
- **Status:** Open. Already raised to Google in issue #22 (`android/antroid-xr-unity-package`) since at least 2026-04-09, no response.
- **Clinical impact:** Task T07 Facial Activation (CN VII assessment — Bell's palsy, facial-weakness screen, FAST stroke screen) returns INVALID. No facial-asymmetry data available on Galaxy XR today. Bilateral lip-corner pull, brow asymmetry, eye-closure asymmetry (lagophthalmos), and cheek puff asymmetry are all clinically interpretable from the 68-channel blendshape API once the service streams data.

Issue 5 — Inward camera held exclusively by perception inputserver

- **Headline:** `com.google.xr.perception.inputserver` holds the inward camera exclusively while the OpenXR session is active; `Camera.open()` returns a handle but `MediaRecorder.start()` throws `RuntimeException: start failed`. Resulting MP4 is 0 bytes.
- **Evidence:** `docs/T07_VIDEO_FALLBACK_BLOCKED.md`. Session JSON 2026-05-06: `T07_FacialActivation: QC: T07 video fallback could not start: java.lang.RuntimeException: start failed. Live blendshape pipeline still active.`; `diag.video_fallback_attempted: true`; `diag.video_fallback_failed: true`; output file 0 bytes. Camera2 has the same XR-perception-runtime exclusivity. MediaProjection captures the rendered framebuffer (patient's POV / passthrough world), not the front camera.
- **Root cause:** Platform policy at the runtime layer, not API layer. Confirmed by Cheri Malo (Samsung Korea R&D) 2026-04-30 reply: "raw external camera feed access is 'No' under consumer policy."
- **Ask:** Cheri Ask #1 (raw inward camera under research NDA) — covered in Section 5.1. Narrower fallback: Cheri Ask #4 (eye + face video segment export entitlement) — covered in Section 5.4.
- **Status:** Acknowledged by Samsung Korea R&D 2026-04-30 (consumer policy "No"). Research-pathway exception is the proposed mitigation.
- **Clinical impact:** Off-device CV pipeline for facial asymmetry (MediaPipe Face Mesh, 468 landmarks) is implemented and ready (`tools/post_hoc_face_mesh.py`). Cannot consume Galaxy-XR-captured frames today. Phone-camera capture works but adds a second hardware piece to the protocol.

4.3 Motor

Issue 6 — No real lower-body / leg tracking

- **Headline:** OpenXR “full-body” leg pose is *inferred* from head + hands (and now optionally torso via the experimental Google `XRBodyTrackingFeature`), not measured from leg-mounted sensors or cameras.
- **Evidence:** Per Cheri Malo 2026-04-30 reply and `SAMSUNG_XR_PLATFORM_ASKS.md` Ask 5. Confirmed in the `com.google.xr.extensions` package documentation (Body Tracking marked “Experimental*”).
- **Root cause:** Galaxy XR has no lower-extremity sensor; leg pose in the body model is generated by inverse kinematics from upper-body anchors.
- **Ask:** Samsung Platform Ask 5 (real lower-body sensing roadmap). Partial mitigation via Samsung Platform Ask 4 (world-facing camera + on-device pose ML) where leg geometry falls in the world-camera FOV.
- **Status:** Open.
- **Clinical impact:** Heel-to-shin (Mingazzini) test — sensitive for unilateral pyramidal weakness — not feasible on Galaxy XR today. Foot tapping (Babinski-sensitive UMN screen) blocked. Patellar reflex observation blocked. Each of these is a standard bedside neurologic exam component.

Issue 7 — World-facing camera availability and FOV uncharacterized

- **Headline:** Android XR documents `camera_id=0` world-camera access with Camera2 + permission, but it is labeled “preview”; downward FOV (lap / feet coverage when seated) is not specified.
- **Evidence:** developer.android.com/develop/xr camera documentation; SAMSUNG_XR_PLATFORM_ASKS.md Ask 4.
- **Root cause:** Platform documentation gap + unknown sensor geometry on Galaxy XR.
- **Ask:** Samsung Platform Ask 4.
- **Status:** Open.
- **Clinical impact:** Camera + on-device-pose-ML path for the entire lower-extremity battery (foot tap, heel-shin, patellar reflex observation) depends on world-camera access being live and the FOV covering the patient’s lap / feet when seated.

4.4 Gait

Issue 8 — Hard ~2 m consumer safety boundary halts overground walk

- **Headline:** ~2 m consumer safety boundary halts the SDNE app and prompts the user back inside, even with `XR_ANDROID_unbounded_reference_space` extension enabled.
- **Evidence:** March 2026 dev report (single source, awaiting Samsung confirmation); SAMSUNG_XR_PLATFORM_ASKS.md Ask 6 first blocker. The Google `com.google.xr.extensions` package providing `XRUnboundedRefSpaceFeature` was added to SDNE 2026-05-25 and built into SENSE Fine v3; first worn-headset overground test pending.
- **Root cause:** Consumer safety policy enforced at the Android XR services layer, above the OpenXR compositor. The unbounded reference space changes the tracking origin but the boundary policy sits above it.
- **Ask:** Samsung Platform Ask 6 (first of two yes/no questions); also Cheri Ask #3 (research-pathway boundary exception under Samsung Knox) — covered in Section 5.3.
- **Status:** Acknowledged by Samsung Korea R&D 2026-04-30. Research-pathway exception is the proposed mitigation. Open as a fix in consumer firmware.
- **Clinical impact:** Real Timed Up-and-Go (~3 m walk + turn + return) and 10-meter walk blocked. SDNE ships an in-place march surrogate that captures cadence, stride variability, double-support timing, and step regularity, but cannot report gait velocity.

Issue 9 — SLAM translation unreliable in passthrough

- **Headline:** 0.18 m of real walking displacement reported as 3.01 m by SLAM (~17× drift) on the 2026-05-06 loaner.
- **Evidence:** 2026-05-06 loaner test; SAMSUNG_XR_PLATFORM_ASKS.md Ask 6 second blocker. Independent of the boundary issue — even within the boundary, the reported position drifts severely.
- **Root cause:** Galaxy XR SLAM in passthrough is not drift-corrected to clinical-grade accuracy. The Unbounded reference space changes the origin but does nothing about drift.
- **Ask:** Samsung Platform Ask 6 (second of two yes/no questions): is SLAM translation in the unbounded reference space drift-corrected to <5 cm over 3 m?
- **Status:** Open. Issue #23 (`android/android-xr-unity-package`) open since at least 2026-04-16.
- **Clinical impact:** Without sub-5-cm drift over a 3 m path, gait velocity (the most reproducible single clinical gait metric) cannot be measured even if Issue 8 is resolved.

4.5 Audio / Speech

Issue 10 — Thermal envelope for long passthrough sessions

- **Headline:** 43 thermal events in 13.3 min (~3 per minute); 603 MB peak memory.
- **Evidence:** XR Test Companion perf summary, 2026-05-25 session. Recorded in `docs/ONDEVICE_RUN_FINDINGS_2026-05-25.md` §1 row P1.
- **Root cause:** Galaxy XR thermal envelope under sustained sensor-heavy passthrough use.
- **Ask:** No specific ask documented today; recommend Samsung guidance on expected envelope and mitigation for sessions exceeding 15 minutes.
- **Status:** Open question.
- **Clinical impact:** SDNE Core-15 is a 15-minute protocol and runs within the envelope today. Extended bulbar / ALS batteries (T20 DDK, T21 sustained phonation, T22 cough loudness) extend session length to 20+ minutes — characterization needed before clinical pilot.

Note on Audio / Speech specifically. The microphone capture path on Galaxy XR functions correctly. The 2026-05-25 T13 Semantic Fluency "0 words" failure observed in session 77e2af69 was caused by a Sevaro-side audio-upload pipeline gap (the WAV captured locally to `/storage/emulated/0/Android/data/com.sevaro.sense/files/sessions/audio/` but did not reach S3 for downstream Bedrock scoring). This was a client-side bug, now resolved in SENSE Fine v2 via an orphan-WAV scan in `AudioUploader.RetryPendingCoroutine`. No Samsung-side platform ask under Audio at this time.

5. Research-Partner Pathway Asks

Three of the issues in Section 4 surface through consumer-policy boundaries that Samsung has confirmed will not move in the consumer build. For these, the proposed resolution is a research-pathway agreement: a partner NDA framework that grants Sevaro (and clinical collaborators including UMMC CoE) access to data streams that consumer apps cannot reach. These three asks were originated in the 2026-05-04 Cheri Malo correspondence thread and refined through the 2026-05-08 must-haves draft.

5.1 Cheri Ask #1 — Raw eye-camera frames under research NDA

- **What:** Sevaro requests access to the raw inward-camera frame stream (per-eye), gated by a research-NDA and IRB-approved clinical protocol.
- **Why:** Unlocks the full quantitative oculomotor battery: saccade peak velocity, nystagmus slow-phase velocity, convergence (true 3D vergence point), internuclear ophthalmoplegia (INO) detection, per-eye fixation gating, lid fatigability for myasthenia gravis screen.
- **Resolves issues:** Issue 1 (compositor stutter — direct camera stream bypasses the compositor-pacing artifact entirely if Sevaro can timestamp the frames), Issue 2 (eye-tracking precision — direct frames allow Sevaro to characterize precision independently), Issue 3 (no raw eye-camera access — by definition), Issue 5 (inward-camera exclusivity — the inputserver continues to hold the camera for the OpenXR pipeline; this ask creates a parallel research stream).
- **Proposed framework:**
  - IRB-approved clinical protocol (Sevaro + UMMC CoE).
  - BAA / DUA between Sevaro and Samsung + Google.
  - Patient consent specifically calling out eye-video collection and research use.
  - Data handling: encrypted on-device, encrypted in transit, encrypted at rest. Sevaro HIPAA-compliant pipeline (`sense.neuroplans.app` infrastructure).
  - Strictly research; not deployed to consumer clinics until the policy moves at the platform level.
- **Status:** Acknowledged by Samsung Korea R&D 2026-04-30 ("No" under consumer policy). Research-pathway proposal pending response.

5.2 Cheri Ask #2 — Documented post-hoc Watch ↔ XR NTP synchronization

- **What:** Sevaro requests a documented method for time-synchronizing a Samsung Galaxy Watch capture session with a concurrent Galaxy XR session, sufficient to align IMU samples post-hoc with sub-100 ms accuracy. This is *not* a request for live streaming (Samsung confirmed live streaming "Not supported" 2026-04-30) — it is a request for the time-base + offset documentation needed to align two independent recordings.
- **Why:** Unlocks Watch-based limb capture as a complement to XR sensors: arm swing during the 10-m walk, wrist-IMU tremor cross-check against XR hand tracking, ankle-IMU as a partial mitigation for Issue 6 (no lower-body tracking).
- **Resolves issues:** Adjacent to Issue 6 (no real lower-body tracking) — gives Sevaro an ankle-mounted alternative without a phone-on-tripod.
- **Proposed framework:** Documentation only. No new APIs required. Just published guidance: which Android time source is canonical on the Watch, which on the headset, and the expected offset / jitter envelope.

- **Status:** Live BLE streaming acknowledged by Samsung as "Not supported." Documentation-only ask pending response.

5.3 Cheri Ask #3 — Pass-through boundary exception under Samsung Knox

- **What:** Sevaro requests an enterprise / Knox-managed exception that disables the ~2 m consumer safety boundary for a Sevaro-managed clinical app.
- **Why:** Pass-through is mandatory for SDNE's brain-injury / vestibular patient population — turning passthrough off is not an option (vestibular patients in immersive VR risk falls). The ~2 m boundary fires *with* passthrough on, *with* the unbounded reference space enabled.
- **Resolves issues:** Issue 8 (consumer safety boundary). Does not address Issue 9 (SLAM drift) — that requires a runtime-side change separately.
- **Proposed framework:**
  - Knox-managed device profile signed by Samsung.
  - Sevaro-controlled medical-app entitlement.
  - Medical-assistant supervision required (already a Sevaro protocol requirement for gait tasks).
  - Clinical-environment-only — not deployable to consumer headsets.
- **Status:** Acknowledged by Samsung Korea R&D 2026-04-30. Research-pathway exception is the proposed mitigation. Open as a fix in consumer firmware.

5.4 Cheri Ask #4 (fallback) — Eye + face video segment export entitlement

A narrower fallback to Cheri Ask #1: rather than streaming raw eye-camera frames in real time, allow Sevaro to receive *short post-hoc video segments* (e.g., per-task 30-second clips) for clinician review and off-device CV processing. Used as evidence for Issue 4 (face-tracking SERVICE_NOT_READY) by enabling MediaPipe Face Mesh post-hoc analysis on captured video. Less ambitious than #1 but easier to scope for an initial research agreement.

- **Status:** Internal fallback to #1, not yet surfaced to Samsung Korea — narrower scope option if #1 stalls.

6. Issue → Ask Mapping Table

Issue	Domain	Samsung Platform Ask	Cheri Research Pathway Ask	Status
1 Compositor stutter	Eye	Ask 1 (TOP priority)	Partial via #1	Open
2 Eye-tracking specs	Eye	Ask 2	Partial via #1	Open
3 Raw eye-camera	Eye	Ask 3	**#1 (primary), #4 (fallback)**	Acknowledged consumer "No"; research-path pending
4 Face-tracking SERVICE_NOT_READY	Face	**Ask 7**	Partial via #4	Open; reproduces in Google's first-party app
5 Inward camera held by inputserver	Face	Ask 3	**#1 (primary), #4 (fallback)**	Acknowledged consumer "No"; research-path pending
6 No real lower-body tracking	Motor	Ask 5	Partial via #2 (Watch-on-ankle)	Open
7 World-camera availability + FOV	Motor	**Ask 4**	—	Open
8 ~2 m safety boundary	Gait	Ask 6 (first)	**#3 (Knox exception)**	Acknowledged consumer "No"; research-path pending
9 SLAM passthrough drift	Gait	Ask 6 (second)	—	Open
10 Thermal envelope	Audio / system	(open question, no formal ask)	—	Open question

Each row connects a measured platform issue to (a) the Samsung platform-side ask from `docs/SAMSUNG_XR_PLATFORM_ASKS.md` and (b) where applicable the research-partner pathway from the Cheri thread. Bold entries mark the *primary* ask for that issue.

7. Methodology

The findings in this report are drawn from ten rounds of on-device validation on Samsung Galaxy XR loaner unit `R32Y8003FKF` (SM_I610, Android XR), between 2026-02 and 2026-05. Each round captured a full Core-15 protocol session, the resulting session JSON, and synchronized logcat. The companion XR Test Companion tool captured per-frame perf telemetry, including thermal events, frame-time distribution, GC pressure, and memory peak. Where the underlying root cause sits inside the OpenXR runtime or below, profiling was extended via Unity Profiler (per-subsystem) and `adb logcat` filters scoped to `OpenXR`, `FaceProvider`, `EyeTrackingProvider`, and `com.sevaro.sdne`.

This work is positioned as V3+ analytical validation, in collaboration with Dr. Jennifer Reneker at the University of Mississippi Medical Center Center of Excellence. Platform-level rigor is established here before clinical validation begins. Healthy-baseline runs (Steve Arbogast, MD, clinically healthy) anchor the false-positive analysis: any RED / INVALID flag on a healthy patient is reported either as a platform issue (this document) or as a Sevaro-side scoring or capture bug (tracked separately in repository commit history). The 2026-05-25 healthy session (`sessionId 77e2af69-2524-4178-8626-9f93f58d2385`) is the most recent and most thoroughly characterized data point and is referenced throughout Section 4.

No clinical claims are made in this document. The asymmetry indices, drift measurements, and intrusion counts are reported as platform characterization data — they describe what the platform delivers (and fails to deliver) on a healthy subject, not what a clinical population would produce.

8. Appendices

Appendix A — Raw evidence index

Artifact	Location
2026-05-25 healthy session JSON	`/storage/emulated/0/Android/data/com.sevaro.sense/files/sessions/SDNE_Core15_` `2524-4178-8626-9f93f58d2385_20260526_024516.json`
Local copy of same session JSON	`/tmp/sense_fine_session.json`
T07 video-fallback failure JSON	Session JSON 2026-05-06; `videos/T07_<sessionId>_<ts>.mp4` (0 bytes)
SAMSUNG_XR_PLATFORM_ASKS.md	`docs/SAMSUNG_XR_PLATFORM_ASKS.md`
UNITY_BUG_face_tracking_service_not_ready.md	`docs/UNITY_BUG_face_tracking_service_not_ready.md`
T07_VIDEO_FALLBACK_BLOCKED.md	`docs/T07_VIDEO_FALLBACK_BLOCKED.md`
ONDEVICE_RUN_FINDINGS_2026-05-25.md	`docs/ONDEVICE_RUN_FINDINGS_2026-05-25.md`
PLAN_2026-05-11_may27_demo.md	`docs/PLAN_2026-05-11_may27_demo.md`

ADB commands to reproduce key evidence on a fresh loaner:

```
# Confirm device + serial
adb devices -l
```

```
# Confirm Galaxy XR permissions granted
adb shell cmd appops query-op --user 0 FACE_TRACKING
adb shell cmd appops query-op --user 0 EYE_TRACKING_FINE

# Reproduce Issue 4 (face SERVICE_NOT_READY) in Google's first-party app
adb shell am start com.google.xr.facetracking.calibration/.FaceTrackingCalibrationActivity
adb logcat -d | grep "xrGetFaceCalibrationStateANDROID"

# Pull the most recent SDNE session JSON
adb pull /storage/emulated/0/Android/data/com.sevaro.sense/files/sessions/ ./sessions_pulled/

# Companion perf telemetry (XR Test Companion)
# Capture via the iPhone companion app while a session runs;
# export the JSON sidecar from the companion's session view.
```

Appendix B — OpenXR call traces for SERVICE_NOT_READY (Issue 4)

Logcat lines observed on every session, both on the Sevaro app (`com.sevaro.sdne` / `com.sevaro.sense`) and inside Google's own first-party `com.google.xr.facetracking.calibration` app:

```
E .../FaceProvider.cpp:351 [FaceProvider] Face state data was unavailable.
  Could not retrieve region confidence data.

E .../FaceProvider.cpp:330 [FaceProvider] Face state data was unavailable.
  Could not retrieve blend shape data.

E com.sevaro.sense  XR_ERROR_SERVICE_NOT_READY_ANDROID in
  xrGetFaceCalibrationStateANDROID: Failed to get face calibration state.

E .../XrFuncTable.h:119 xrGetFaceCalibrationStateANDROID failed with result Unknown.

W .../FaceProvider.cpp:225 [FaceProvider] Could not determine face calibration
  state. Face state data may not yet be available.
```

The first-party reproduction (running Google's `com.google.xr.facetracking.calibration` app, observing *its* PID in logcat) shows the same `XR_ERROR_SERVICE_NOT_READY_ANDROID` repeated hundreds of times. This conclusively places the failure inside the Galaxy XR runtime / service layer rather than inside Sevaro's app or Unity's package layer.

Appendix D — Version history

Version	Date	Notes
1.0	2026-05-26	Initial release. Drafted ahead of the 2026-05-27 Plano workshop. Status fields reflect pre-workshop state.

Planned updates post-workshop:

- Status field per issue updated to reflect Samsung / Google response (Acknowledged / In Discussion / Resolved).
- New section if any new platform issues surface during the workshop or in follow-up testing.
- Appendix C (Glossary) added in v1.1.
- Sevaro-side mitigation table summarizing what we've shipped to work around each issue (SENSE Fine v3 commit `6df9fd8` and prior).