Link to doc: https://goo.gl/qLiZLA

AMOS WS23 Project Android Zero Instrumentation

Instructions

The project report will be published on our blog. It should be short and sweet, focussed on what you achieved.

> "I did not have time to write a short letter, so I wrote a long letter instead." Attributed to Mark Twain

Being concise is hard work and takes time. Please write as professional a text as you can. Use formal language and correct grammar.

For illustrations please use a persona rather than "test X" or "testperson12".

Prior examples (not necessarily following our instructions, sadly), can be found here:

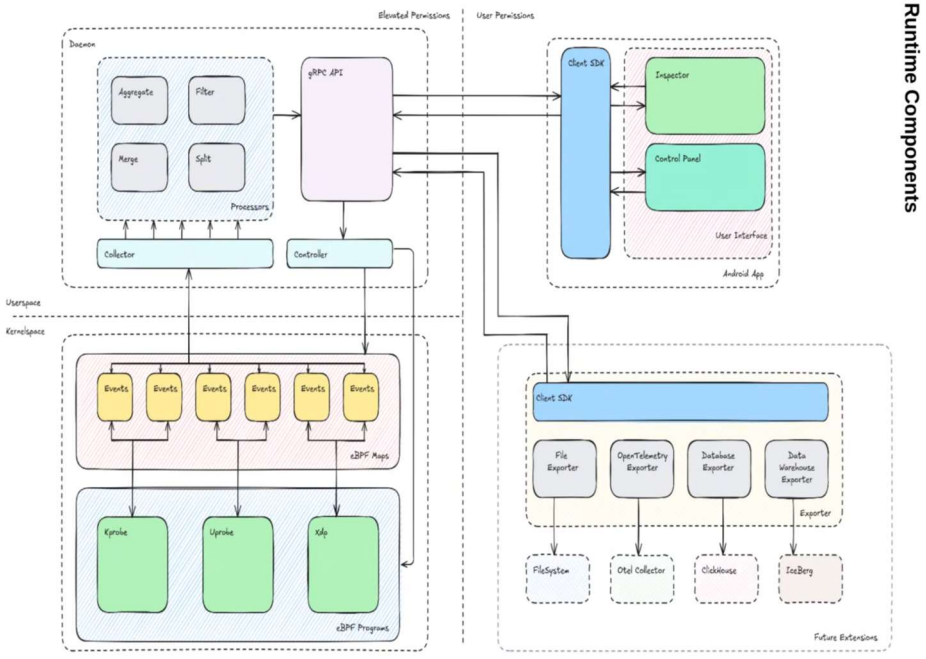
https://dirkriehle.com/2021/03/02/summary-of-the-winter-2020-21-amos-projects/

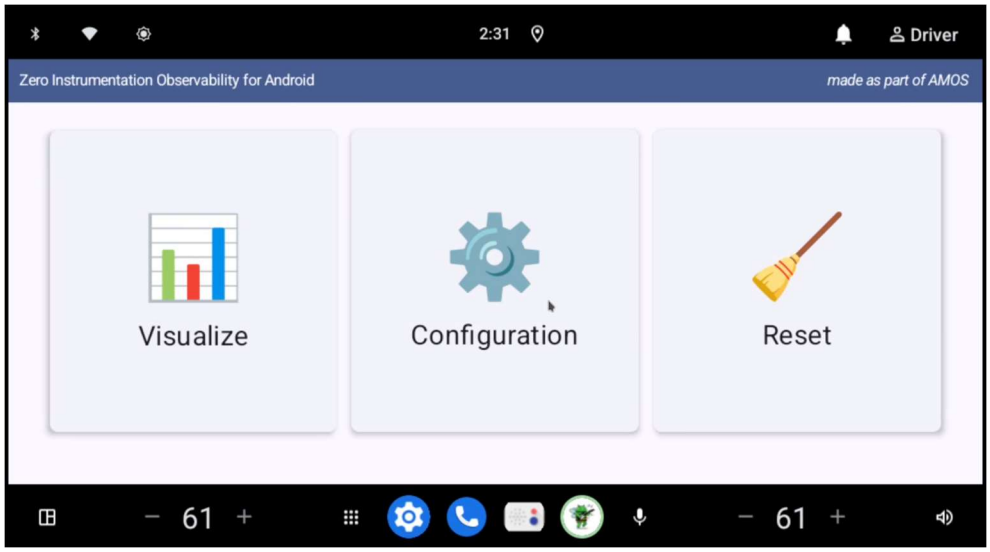
https://oss.cs.fau.de/2018/08/01/show-casing-the-2017-amos-project-simulating-a-cars-ecus-using-a-raspberry-pi-5/

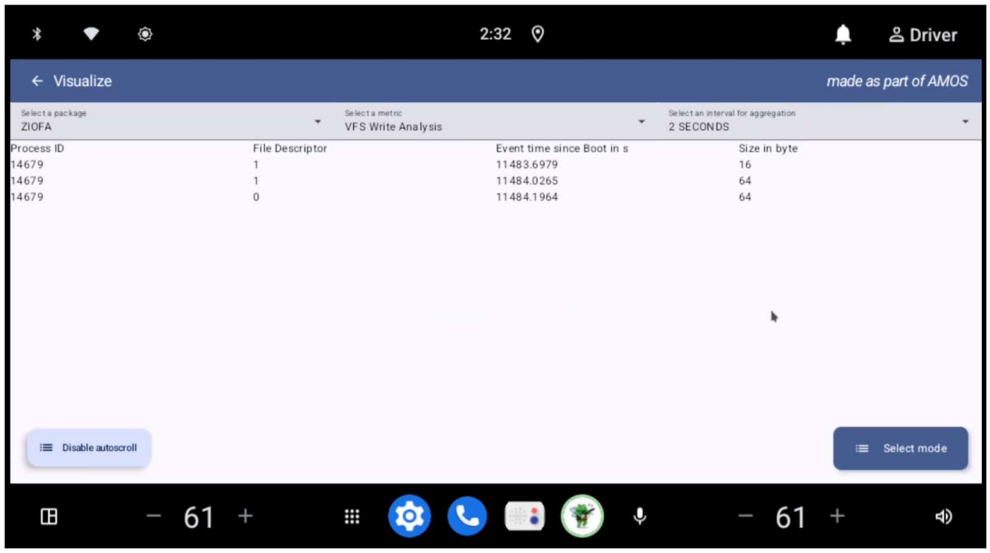
Template

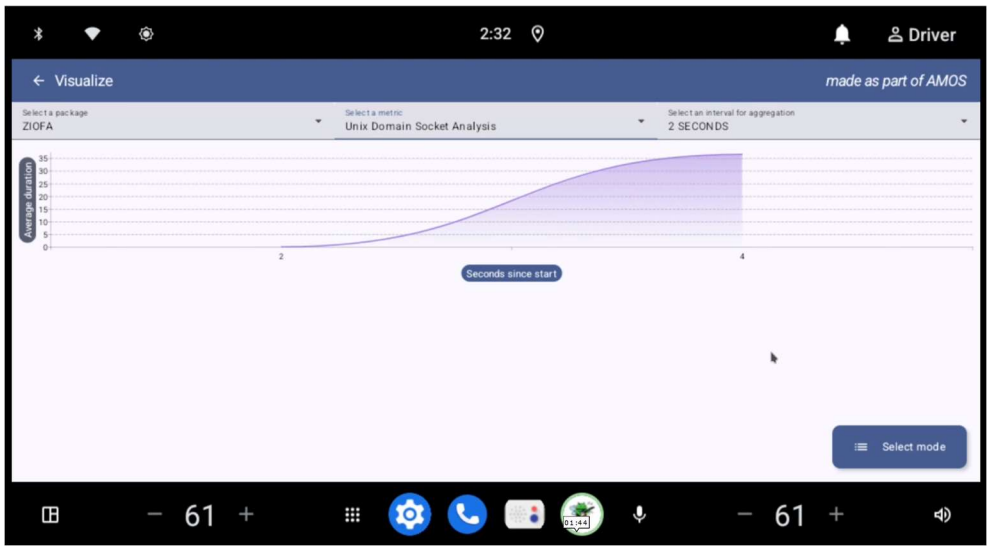
Please use the following template for creating your project report.

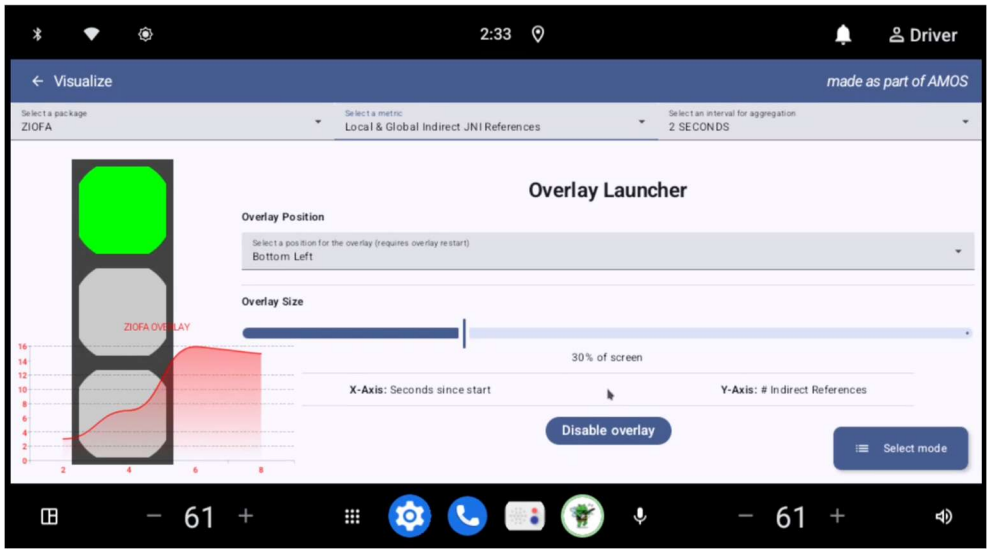
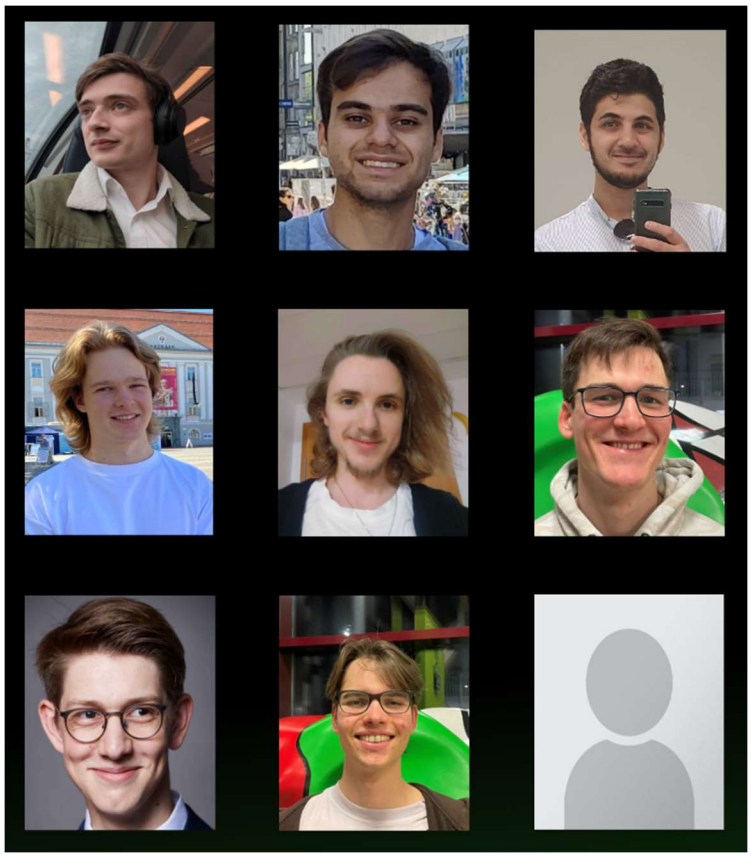
Project name	Zero Instrumentation Observability for Android
Project mission	Creating a longterm solution to test and collect debugging data from the Android-Kernel for Car Based software, utilizing eBPF.
Industry partner	e.Solutions gmbh
Team logo	
Project summary	**Persona: Alex – Android Automotive OS Developer** Alex is a senior software engineer working on in-car infotainment systems. His team frequently faces app performance issues but struggles to debug them because they lack access to source code or real-time system insights. **User Story 1 – Eliminating UI Freezes with eBPF**

	💬 *"As an Android Automotive OS developer, I want to track blocking system calls in real-time so that I can identify and eliminate UI lags that degrade the user experience."* 🚀 **Why it matters:** eBPF-powered observability allows Alex to monitor system calls that slow down the UI thread, visualizing delays as charts. This helps the team fix performance bottlenecks without modifying application code. **User Story 2 – Protecting Flash Storage in Cars** 💬 *"As a system engineer, I want to track excessive write operations in real time so that I can extend the lifetime of in-car flash storage."* 🚀 **Why it matters:** Continuous, high-frequency writes wear out flash storage, especially in vehicles designed to last a decade or more. The eBPF-based solution tracks all writing syscalls and reports their frequency and size, helping engineers optimize storage usage before failures occur. **User Story 3 – Debugging Memory Leaks Without External Logs** 💬 *"As a tester, I want to see real-time garbage collection and heap size data on-screen while running an app so that I can identify memory leaks faster."* 🚀 **Why it matters:** Current debugging methods require pulling logs from external devices, slowing down development. The eBPF-based approach enables real-time overlays for JVM garbage collection and heap tracking, making memory debugging faster and more intuitive. 🔹 **The Pitch:** Our project brings **zero-instrumentation observability** to Android Automotive OS. By leveraging **eBPF tracing**, developers gain **real-time performance insights** without modifying app code. This is a **game-changer** for debugging, optimizing, and ensuring long-term reliability in automotive software. We differ from traditional approaches as we do not need to modify the Kernel which limits the changes that need to be made and doesn't compromise the security. 🚗 ✨
Project illustration	


2:31
Driver
Zero Instrumentation Observability for Android
made as part of AMOS
Visualize
Configuration
Reset
61
61


2:32
Driver
Visualize
made as part of AMOS
Select a package
ZIOFA
Select a metric
VFS Write Analysis
Select an interval for aggregation
2 SECONDS
Process ID
File Descriptor
Event time since Boot in s
Size in byte
14679
1
11483.6979
16
14679
1
11484.0265
64
14679
0
11484.1964
64
Disable autoscroll
Select mode
61
61


2:32
Driver
Visualize
made as part of AMOS
Select a package
ZIOFA
Select a metric
Unix Domain Socket Analysis
Select an interval for aggregation
2 SECONDS
Average duration
Seconds since start
Select mode
61
61

	
Team photo	
Project repository	https://github.com/amosproj/amos2024ws03-android-zero-instrumentation/tree/main
Additional information	