RUNTIME SYSTEMS ENGINEER (M/F/D)

Location: Dresden, Germany
Type: Full-time (Dresden-based preferred; hybrid welcome for EU-based candidates)

We are SMAROBIX. We build software that gives robotics engineers clarity in complex robotic systems. Our tool makes ROS-based robotic systems observable, understandable, and actionable. Like this, developers can build safer, more capable robots with confidence, while using AI where it makes sense.

We are a team of five founders with complementary skill sets and different cultural backgrounds. We are a spin-off of TU Dresden and received €1.5M in non-dilutive funding.

THE ROLE

This role is part of the core team shaping the product's foundations. This is a hands-on, design-and-build role. There is no template to follow. Progress depends on understanding the constraints deeply and designing within them rather than hoping they don't matter.

- You'll work on understanding robotic systems as they run under real conditions. ROS-based systems are dynamic and opaque at runtime: components appear and disappear, communication crosses process boundaries, and timing and reliability guarantees must hold under load
- You'll capture what actually happens, at the right level of detail and with minimal overhead; it is a hard systems problem without an established solution in this domain
- The challenge is not adding instrumentation in the abstract, but deciding what is important, how to understand it, and how to represent it, so the rest of the product has reliable ground truth to reason from.
- The software runs directly on production robots, where correctness and low overhead are not negotiable.

WHAT WE'RE LOOKING FOR

We're looking for an engineer who's comfortable working close to the metal, where OS behavior, memory, concurrency, and timing are explicit design concerns rather than hidden details.

- Reasoning clearly about what happens inside a running system is where you excel, from having programmed and debugged at the lowest level.
- You understand robotic middleware not as a black box but as part of the system to be understood in detail.
- You're familiar with how nodes communicate, how data moves across process boundaries, and what QoS settings mean in practice beneath the surface.
- You've worked with observability or instrumentation where overhead is a real cost, and you reason carefully about the trade-offs between visibility, precision, and impact.
- You can zoom out from low-level implementation details and make architectural decisions that hold up over time.
- You approach tools pragmatically, choosing methods that fit the constraints rather than forcing familiar patterns onto novel problems.

- You think in concepts first; tools are secondary to understanding the underlying principles well enough to apply them in any context, whether you come from academia, industry, or an unexpected path.
- You communicate clearly, take responsibility for reliability in user environments, and work beyond role boundaries when it helps. You know the difference between not knowing yet and not knowing at all, and you're honest about which one it is.

Nice to have

- Prior work with observability tooling, kernel modules, deterministic networking, real-time OS, or embedded/hardware acceleration experience helps

YOU'LL THRIVE IN THIS ROLE IF

You enjoy solving open-ended problems and taking ownership rather than waiting for detailed specifications. You won't maintain an existing runtime tracing system or implement a predefined spec. You want your system-level judgment to shape what counts as trustworthy ground truth. The foundation exists, but the critical decisions about what's trustworthy enough are still open, and what you build will shape the product for years.

WHO YOU'LL WORK WITH

You'll work directly with the founding team and in an interdisciplinary group that values speed and technical excellence over hierarchy.

- Specifically, you'll work closely with one co-founder, with real ownership and room to challenge assumptions.
- As the team grows, this work naturally evolves into technical leadership through influence and clarity rather than hierarchy.

HOW WE WORK TOGETHER

As an early team member, expect priorities to evolve quickly and responsibilities to extend beyond a narrowly defined role.

- We argue hard. We debate ideas openly and don't hold back, but at the end of the day we back each other and go ahead.
- An undesired outcome happens. Validation of assumptions is key. When something goes wrong, we address it directly, focus on the issue, not the person, and move on by adjusting the assumption or the process.
- We have the ability to learn fast. We will listen to feedback but only implement relevant feedback. We make decisions on the strongest argument.

.

HOW TO APPLY

Send your CV and a short cover letter to **jobs@smarobix.com.** In your cover letter, tell us about a hard technical problem you worked on and the moment you realized your first approach was wrong - what did you do?

Questions about the role? Reach out to Ariel, Co-Founder & CTO. We'd be happy to discuss the role before you apply. Write him directly: ariel@smarobix.com