Robotics Software Engineer — Teleoperation

LatentWorlds builds the **deployment backend for physical AI**. We turn fleet hours into compounding autonomy by making robotics data reliable to collect, usable to retrieve, and practical to turn into training and debugging artifacts.

Teleoperation is part of our **ops plane**: the operator-visible layer that keeps deployments safe and productive when autonomy is imperfect—and makes interventions traceable and replayable.

Type: Full-time
Location: Flexible (remote-friendly)
Apply: Send your CV + a short note + links to relevant work to cristianmeo@latentworlds.ai

The role

You'll help build **teleoperation and operator tooling** that holds up outside the lab: unstable networks, real operators, and safety constraints.

The scope is intentionally flexible. We're a small team. As we learn from customers, priorities shift—and we want someone who enjoys that kind of work.

What you'll work on

Projects vary, but you'll likely touch a mix of:

- **Operator console UX** for monitoring, takeover, and handoff (clear states under failure, calm interaction design).
- **Teleop sessions + guardrails** (scoped access, TTLs, revocation, audit trail, safety defaults).
- **Low-latency feedback loops** (video + telemetry that stays usable on real networks).
- **Robot-stack integration** (ROS/ROS2 or custom stacks; small interfaces, minimal assumptions).
- **Testing against reality** (a bad-network harness; latency and reliability metrics we can regress).

Why this exists

Most teams can "remote control a robot." The hard part is making teleop **operable**: safe-by-default, reliable under bad connectivity, and connected to the data workflows that make incidents reproducible and useful.

What success looks like

In the first weeks, success is shipping an end-to-end loop (operator ↔ robot) with:

- clear UI states and safety behavior,
- measurable latency and failure modes,
- a test harness that makes regressions obvious.

Over the next months, success is hardening it into primitives that teams can run in production:

- sessions + permissions that operators trust,
- usability under degraded networks,
- interventions that show up as searchable, shareable slices in DataCore.

What we're looking for

We care most about **taste, ownership, and learning speed**. Frameworks change; good instincts compound.

Strong candidates usually have evidence of:

- Shipping an interactive system that real users relied on (robotics, streaming, AR/VR, simulation, safety-critical tooling, etc.).
- Comfort with both **product/UI details** and **systems constraints** (latency, reliability, failure modes).
- Clear communication: you write down invariants, make tradeoffs explicit, and don't hand-wave safety.
- Strong programming fundamentals (language doesn't matter as much as craft; we use Rust heavily).
- Willingness to work close to customers and operators.

Bonus points (not required)

- ROS/ROS2 integration experience.
- WebRTC / media pipelines / low-latency A/V.
- Permissioned control paths, audit logs, observability.
- Operator tools in high-stakes environments.

About LatentWorlds

LatentWorlds builds DataCore: edge-to-cloud ingestion, robotics-native indexing, training/debugging retrieval, reproducible pipelines, and ops primitives (teleop + monitoring) that make deployments workable.

How to apply

Email **cristianmeo@latentworlds.ai** with:

- your CV / LinkedIn,
- a short note on what you've built that's most relevant (and why),
- links to code, demos, writing, or a portfolio (if available).

If you're excited about the problem but don't match every bullet, you should still reach out.

Equal opportunity

LatentWorlds AI is an equal opportunity employer. We welcome applicants of any background and identity, and we care most about the quality of your work and your ability to collaborate.