

Avocado OS: Sim2Real Reference Design

Production-grade Jetson™, without the re-platform



Robot One is the beginning. Robot Two is the challenge.

NVIDIA built the best on-ramp in physical AI. Policies train fast in Isaac Sim, Isaac ROS gives you perception and control, and JetPack stands up your first robot in days. Then you build ten more. The image that worked on the bench drifts unit to unit, a field update has no way back, and every JetPack bump means rebuilding your kernel modules. That's the layer every team ends up rebuilding from scratch.

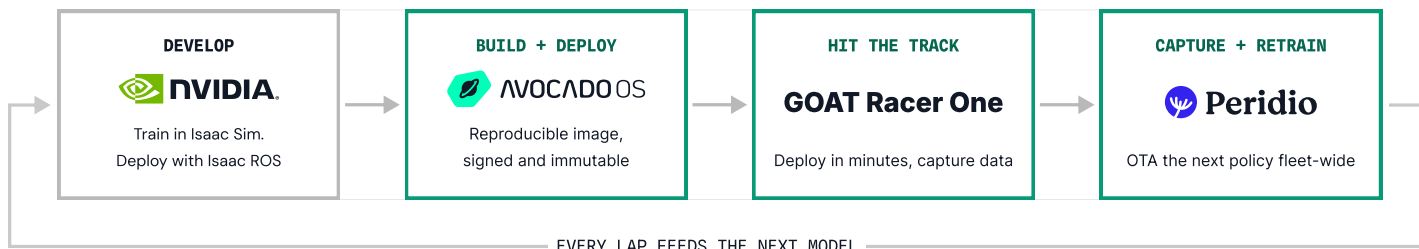
GOAT Racer One is that reference. Cryptide Research built it on exactly this stack and open-sourced it. Not a demo: they ran 90 OTAs in a couple weeks, iterating policies on the live car. The blank Yocto tree is already a running robot.

You Bring the Robot. We Bring Production.

NVIDIA's toolchain trains and runs the policy. Avocado OS gives it a production home: a reproducible, open image, atomic OTA with rollback, and fleet management. The sim-to-real result ships secure and updatable, identical from the twin to the track.

Start from the reference and fork what you need, not from zero.

The Goat Racer One Software Stack



Give your systems engineers a break.

The first robot was hard enough. Shipping a fleet of them shouldn't mean rebuilding your whole stack. Peridio brings the production layer so your product ships secure and maintainable on any hardware you need.

Four problems. One runtime.

SHIP WHAT YOU TESTED

Exact image build. Signed, bit for bit. Ship exactly what you tested.

A WAY BACK

Native OTA. Atomic with rollbacks. A way back when you need it.

RAPID PROVISIONING

Flash from any machine. Rapid provisioning in minutes, no specialized hardware required.

NO FORK, NO LOCK-IN

Your hardware kernel. Kernel and drivers, unchanged. No fork, no lock-in.

Built for physical AI teams.

MECHANISM	WHAT IT DOES
ROS Support	Run ROS 2 natively or in a container. Your architecture, your call.
Silicon support	One config builds for Jetson, i.MX, Qualcomm, Raspberry Pi, and x86. Curated BSPs across 20+ targets, so one chip's EOL or price spike can't stall your roadmap.
Model updates	Push a new model or policy without shipping a whole new image. Sub-second, no reboot, small enough for a constrained link.
Host access	Run containers where they help, with full access to the host OS underneath. No sandbox, no walled garden.
GPU workload isolation	MIG reserves deterministic GPU capacity for real-time control while ROS perception runs in parallel. No contention between the two.
Real-time kernel	Upstream and PREEMPT_RT kernel patches. Real-time is a flash away, not a project.

CONTROL PLANE



Peridio Core

Device management across the full lifecycle

Fleet | OTA + LTS | Observability | Remote Ops

Operates the deployed fleet

BUILD INFRASTRUCTURE

Package feeds

Public and private, versioned and pinned

Extensions

Your applications, layered on the base

Avocado OS base

Immutable, composable Yocto runtime

DEVELOPERS

Compose feeds, extensions, and base into signed releases.

WORKSTATIONS - CI

SIGNED IMAGE

Immutable artifact, reproducible bit for bit.

SIGNATURE - PROVENANCE

FIELD DEVICES

Production fleet running signed Avocado images.

EDGE - PHYSICAL AI

BUILT ON Yocto Project, curated open-source components, and silicon-vendor BSPs across 16+ targets.

Deploy *faster* on NVIDIA Jetson with **production-ready infrastructure**

Peridio replaces custom OS pipelines and fragmented tooling giving teams faster time to production, higher reliability, and scalable lifecycle management across every robot they ship.

