

Production-ready embedded Linux for *Raspberry Pi*

A custom Yocto-based platform that turns the world's most popular single-board computer into a real industrial product — owned, reproducible, secure, and built to ship for a decade.

Validated on **CM4** · **CM5** · **Pi 4** · **Pi 5** · Companion firmware on **Pico** / **Pico W**

THE PLATFORM

The prototype was Raspberry Pi OS. The product shouldn't be.

Raspberry Pi is fantastic for prototypes — and risky for products when you ship stock Raspberry Pi OS.

*We replace it with a minimal, reproducible Yocto build: **locked-down, updatable, secure, and manufacturable** — a platform you own outright.*

Stock Raspberry Pi OS is Debian-based and general-purpose: great on a bench, a liability in a shipped product. You inherit an image no one can rebuild, patch, or audit six months later. Our platform is built from source with Yocto, so every layer — bootloader, kernel, drivers, middleware, update system — is configured for your hardware and your industry, then handed to you with the full source and build system.

VALIDATED HARDWARE

Compute Module 4 (CM4) · Compute Module 5 (CM5) · Raspberry Pi 4 · Raspberry Pi 5 · Pico / Pico W (RP2040 / RP2350) companion firmware

8 yrs

building production
embedded Linux platforms

6

SoC platform families — ~90%
of the embedded Linux market

1

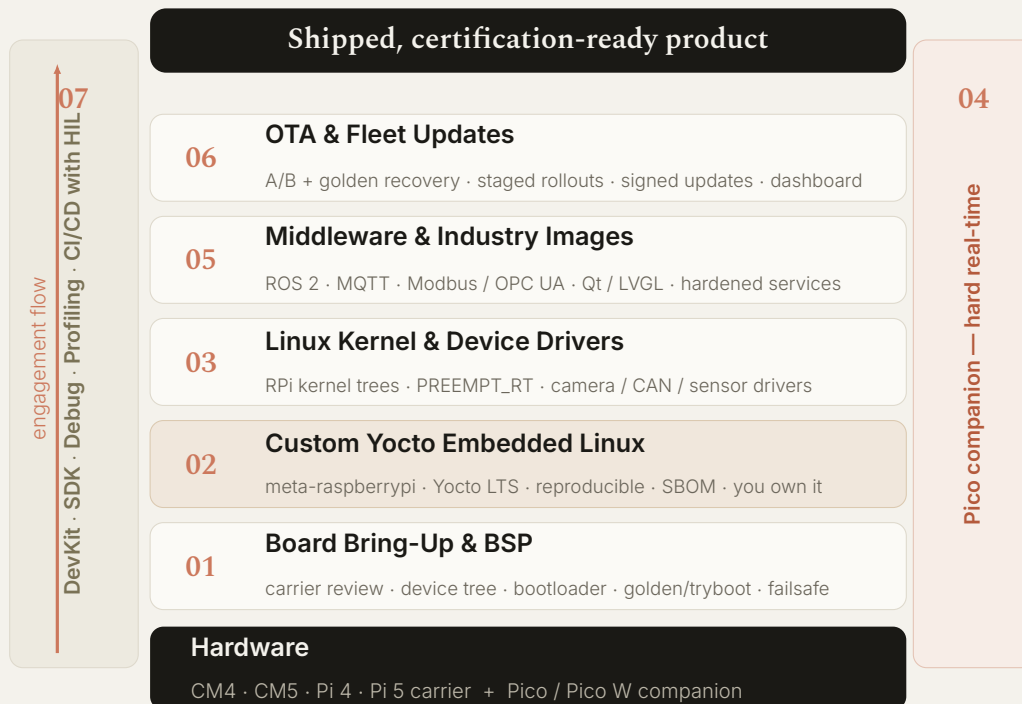
engineering methodology,
applied across every platform

[Add your proof point here] e.g. "Shipped a Yocto platform for a robotics customer — N devices in the field, zero failed OTA rollbacks." One concrete, anonymized outcome here will do more for credibility than any claim on the following pages. Replace this box before sending.

HOW IT FITS TOGETHER

The platform stack

Hardware comes alive at the bottom and ships as a maintainable product at the top. Developer tooling spans the whole engagement; the Pico companion runs the hard-real-time path alongside Linux.



Each layer is a capability detailed on the following pages.

■ Hardware & product ■ Owned OS foundation ■ Real-time companion ■ Spans the engagement

Board Bring-Up & BSP

Your custom carrier boots reliably the first time — and every time after.

The foundation layer: getting custom Raspberry Pi hardware to power on, enumerate every peripheral, and survive the field. This is where a respin gets avoided and where a product earns the right to be unattended.

- **CM4/CM5 carrier review** — schematic review before fab: power sequencing, PMIC rails, boot-mode straps, and high-speed lanes (PCIe, USB, HDMI, MIPI). Catch layout problems before you commit to silicon.
- **Custom device tree & overlays** — describing exactly what's wired: CSI/DSI cameras and displays, CAN, RS-485, ADCs, industrial I/O, and custom HATs.
- **Bring-up & smoke test** — methodical checkout of power rails, clocks, memory, and every peripheral, one at a time.
- **Bootloader & secure boot** — EEPROM configuration, boot-order policy (NVMe / eMMC / SD / network), and signed-boot enablement on CM4/CM5.
- **Golden boot & tryboot fallback** — a protected, known-good image so a failed update or boot falls back instead of bricking.
- **Storage & failsafe** — eMMC/NVMe partitioning, redundant A/B slots, and watchdog-supervised boot with automatic rollback.

TECHNICAL DETAIL			
Boot chain	EEPROM → firmware → kernel	A/B mechanism	tryboot or U-Boot
Storage	eMMC · NVMe	Recovery	Protected golden image

You receive: custom device tree & overlays, a documented bring-up report, bootloader configuration, golden/tryboot setup, partition map, and production flashing scripts.

Custom Yocto Embedded Linux

An operating system you own outright — reproducible, auditable, and yours to maintain.

Replacing stock Raspberry Pi OS with a real, owned, reproducible platform. This is the core differentiator and the foundation everything else builds on.

- **Reproducible builds** — your entire OS built from source on meta-raspberrypi (Yocto LTS). Same inputs produce the same image, traceable to exact source revisions.
- **Minimal & locked down** — only the packages your product needs. Read-only rootfs, no stray services, no default credentials, smaller attack surface.
- **You own everything** — full source, build system, and recipes handed over. No lock-in, to us or anyone.
- **Auditable by design** — SBOM-ready; point to exactly what’s in your image and where it came from.
- **Maintainable for years** — pull CVE patches, bump kernels, and rebuild reproducibly across the product’s full lifecycle.
- **Manufacturable** — the build system is your manufacturing artifact; every unit flashed identical, no manual setup on the line.

TECHNICAL DETAIL			
Build system	Yocto / BitBake	BSP layer	meta-raspberrypi
Release base	Yocto LTS	Supply chain	SBOM + signed images

You receive: the full source tree and build system, a reproducible signed image, an SBOM, and build documentation your team can run themselves.

Linux Kernel & Device Drivers

Every peripheral on your board behaves like a production-grade component.

Where the OS meets your specific hardware. The kernel is shaped to your board — not the generic desktop config — and every device on the carrier gets a proper, predictable driver.

- **Kernel customization & minimization** — start from the Raspberry Pi kernel trees, then strip unused drivers and subsystems for a smaller, faster-booting, more secure kernel.
- **PREEMPT_RT** — real-time patching for bounded, predictable latency in soft-real-time control applications.
- **Fast boot** — streamlined init for kiosk/HMI time-to-first-frame; measured and optimized against your target.
- **Driver development** — cameras (libcamera / Unicam / CSI), DSI/DPI displays, CAN (MCP2515 / MCP251xFD), industrial sensors, and custom HATs.
- **Deterministic I/O APIs** — clean, predictable userspace interfaces to GPIO, PWM, I2C, and SPI — not hobbyist sysfs poking.

TECHNICAL DETAIL			
Kernel base	RPi downstream trees	Real-time	PREEMPT_RT (soft)
Camera	libcamera / Unicam	CAN	MCP2515 / MCP251xFD

You receive: a tuned kernel configuration, custom drivers for your peripherals, and documented userspace I/O APIs. For hard real-time, see the Pico companion (04).

Real-Time Companion (Pico)

Hard real-time the Pi alone can't promise — handled by a dedicated companion.

The Pi's application processor is excellent for compute, connectivity, and UI — but it can't guarantee hard-real-time deadlines. The Pico (RP2040 / RP2350) closes that gap, turning a platform limitation into a clean architecture.

- **FreeRTOS & Zephyr on RP2040/RP2350** — lightweight or feature-rich real-time firmware; the RP2350 (Cortex-M33, security features) is the stronger target for new designs.
- **Split architecture** — the Pi handles networking, UI, logging, and compute; the Pico handles tight control loops, precise timing, and safety-critical I/O where a missed deadline is a real failure.
- **Programmable I/O (PIO)** — cycle-exact custom protocols and timing, independent of the CPU. This covers much of what teams reach for an FPGA to do — at far lower cost and complexity. *Raspberry Pi has no programmable logic; for designs that truly need an FPGA, we deliver it on our AMD Xilinx Zynq platform.*
- **Structured Pi↔Pico link** — a defined protocol over UART / SPI / USB (framing, commands, acks, error handling) so the two domains coordinate reliably in production.

TECHNICAL DETAIL			
MCU targets	RP2040 · RP2350	RTOS	FreeRTOS · Zephyr
Timing	PIO cycle-exact I/O	Link	UART / SPI / USB

You receive: Pico firmware, the Pi↔Pico protocol implementation on both sides, and a documented partition of responsibilities between the Linux and real-time domains.

Middleware & Industry Images

Ship faster with a hardened, industry-tuned image instead of a blank build.

Pre-integrated platform variants tuned per industry — the right stack, hardened and ready, on day one. Every variant ships on the same production baseline.

IoT & Connected
MQTT, cloud agents (Azure / AWS), fleet telemetry.

Industrial Automation
Modbus, OPC UA, RS-485, real-time I/O.

Robotics
ROS 2 tuned for Pi 5, time-synced I/O.

Kiosk & HMI
Wayland kiosk, Qt / LVGL / Chromium.

Edge AI & Vision
TensorFlow Lite / ONNX, camera pipelines, AI accelerators.

Medical
IEC 62304-aligned traceable builds, SBOM, audit logs.

Also available: digital signage, point-of-sale, energy & utilities (Modbus / DNP3), and building automation (BACnet / KNX) — or a variant built to your vertical.

PRODUCTION BASELINE — EVERY VARIANT			
Filesystem	Read-only + overlayfs	Resilience	Watchdog recovery
Services	Hardened daemons	Tamper / power loss	Survives both

You receive: an industry image pre-integrated with the relevant stack, hardened to the production baseline, ready to evaluate on real hardware.

OTA & Fleet Updates

| *Update a whole fleet in the field, safely, without sending a truck.*

The capability that separates a product from a project. A prototype gets flashed by hand; a product gets updated over the air for a decade — safely, with a way back from any bad image.

- **A/B updates** — via native tryboot or RAUC / Mender; run one slot while the other updates, then switch.
- **Golden recovery** — a protected image that updates can never overwrite. Worst case, the device still boots and recovers.
- **Auto rollback** — a failed health check or watchdog trip reverts to the last working image automatically, with no field visit.
- **Fleet management** — cloud or on-prem server, staged rollouts (1% → 10% → all), and delta updates for slow or metered links.
- **Dashboard** — upload an image, pick a fleet, watch the rollout, one-click rollback.
- **Signed & secure** — cryptographically signed updates chained to signed boot on CM4/CM5.

TECHNICAL DETAIL			
Engine	tryboot · RAUC · Mender	Scheme	A/B + golden
Rollout	Staged + delta	Trust	Signed, chained to boot

You receive: a configured OTA pipeline, the update server (hosted or on-prem), the rollout dashboard, and signed-update tooling integrated with secure boot.

DevKit, SDK, Debug & Profiling

Your team owns and operates the platform — no black box, no dependency.

Spans the whole engagement. This is the tooling that lets your own engineers develop on, debug, profile, and rebuild the platform themselves — and it de-risks the entire purchase.

- **Evaluation images** — a working platform on a standard Pi 4/5 or CM4/CM5 IO board, running on day one, before your custom carrier even exists.
- **Application SDK & Yocto eSDK** — a cross-toolchain for your app developers, plus the full extensible build-system workflow so your platform team can modify recipes and rebuild themselves.
- **Debug & profiling** — perf, ftrace, remote gdb, and boot analysis.
- **CI/CD with hardware-in-the-loop** — automated builds and smoke tests on real hardware catch regressions before they ship.

TECHNICAL DETAIL			
Eval target	Pi 4/5 · CM4/CM5	SDKs	App SDK + eSDK
Profiling	perf · ftrace · gdb	Pipeline	CI/CD + HIL

You receive: an evaluation image, both SDKs, debugging/profiling setup, and a CI/CD pipeline with hardware-in-the-loop tests — everything needed to operate the platform without us.

WHY US

Why SoC Centric

One engineering methodology, applied with the rigor of a regulated-industry supplier.

- Replaces a prototype OS with a reproducible, owned, manufacturable platform built to ship.
- You own everything — full source, build system, and documentation. No lock-in, to us or anyone.
- Reproducible Yocto builds with SBOMs — **certification-ready** for ISO 26262, IEC 62304, IEC 61508, and DO-178C environments.
- The hard-real-time gap closed by a Pico companion — an architecture strength, not a workaround.
- Part of a platform family covering ~90% of the embedded Linux market, with one consistent methodology.
- A natural growth path: platform → independent verification & validation → hardware-in-the-loop testing → field data logging.

GETTING STARTED

How an engagement begins

01

Eval image

We send a working platform image for your board to evaluate — no commitment.

02

Architecture call

A 30-minute call to map your hardware, industry, and target to a plan.

03

Pilot

A scoped bring-up + Yocto build on your actual carrier, with deliverables.

04

Platform delivery

Full source, build system, OTA, and tooling — handed over, owned by you.

GET STARTED

Let's talk

Ready to turn your Pi project into a product?

Raspberry Pi gets you to a prototype fast. We turn it into a product you can ship, validate, and support for a decade.

→ Request an evaluation image for your board | → Schedule a 30-minute platform architecture call

GET IN TOUCH

[*name@soccentric.com*]

[*+1 (000) 000-0000*]

[*soccentric.com*]

SOC CENTRIC

Production embedded Linux platforms across
NVIDIA Jetson · AMD Xilinx Zynq · NXP i.MX ·
TI Sitara · Intel/AMD x86 · Raspberry Pi.

[Before sending] Replace the bracketed contact details with your real email, phone, and website, and add your logo to the cover. These are the only placeholders left in the document.