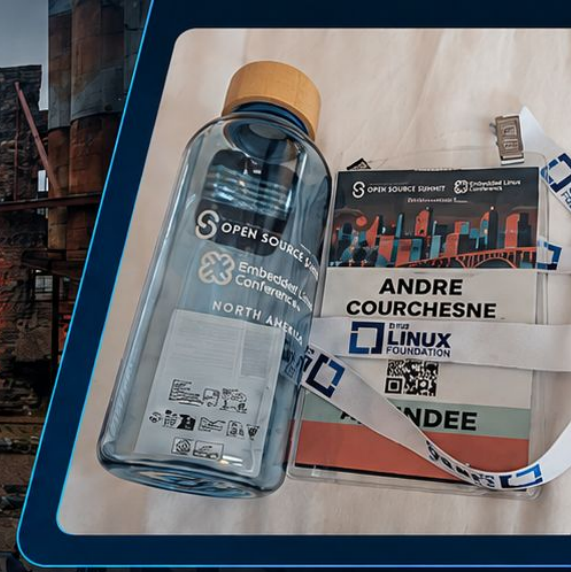
THE LINUX FOUNDATION
OPEN SOURCE SUMMIT
NORTH AMERICA

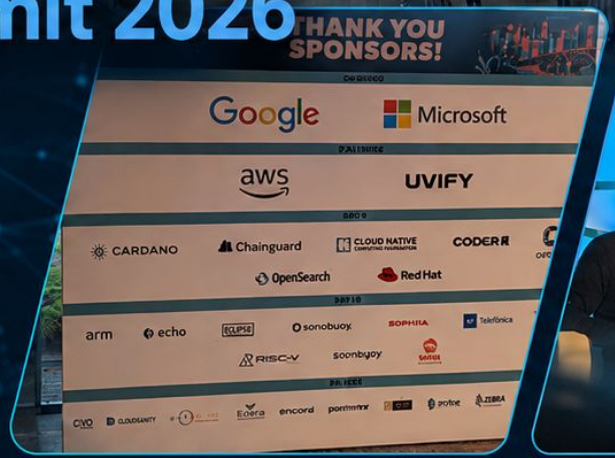
Embedded Linux Conference

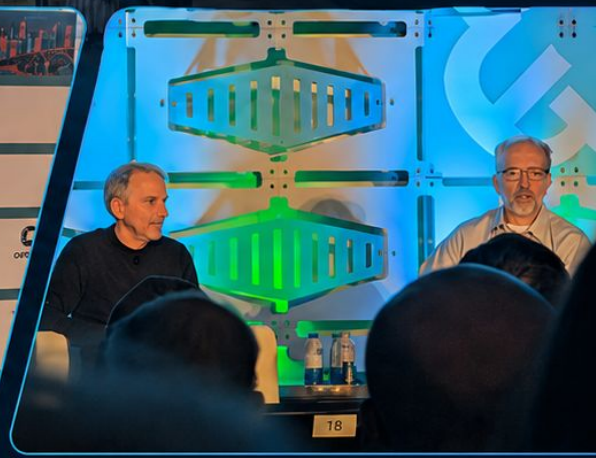
An open source trip to Minneapolis: my discoveries at Open Source Summit 2026

Open Source Summit 2026 • Minneapolis

André Courchesne







THE LINUX FOUNDATION
OPEN SOURCE SUMMIT
NORTH AMERICA

Embedded Linux Conference

Who Am I

Senior Software Developer
at Intello by Telus Business

Founder and inventor
of the VisiblAir air quality monitoring solutions

Inventor and maker —
I bring ideas to real-life things and products

More than 25 years
using Linux and open-source technologies

VisiblAir
https://visiblair.com

LinkedIn
https://www.linkedin.com/in/netforces

OSS 2026: The Whole Source Code

- 12 Keynotes.
- More than 165 major talks.
- 12 simultaneous tracks.
- Estimated more than 1200 attendees on site.

Talks I attended — Day 1

OPEN SOURCE SUMMIT 2026 • MINNEAPOLIS

Keynote: Welcome + Opening Remarks —
Jim Zemlin, CEO, The Linux Foundation

Keynote: UCP: The Evolution of an Open Standard for Agentic Commerce —
Anurag Sinha, Senior Staff Software Engineer & Manager, Google

Keynote: The First Decade of Open Quantum —
Sean Dague, Chief Services Architect, IBM Quantum

Keynote: From Open Source to Agentic Systems: Building the AI Native Era —
Brendan Burns, Technical Fellow

The Revolution Hiding in Plain Sight: CI/CD Platform Is About to Change Forever —
Dadisi Sanyika, Sol Duara, Inc.

Debug Everything: Building a Debuginfod Backbone for Embedded Linux at Scale —
Colin Pinnell McAllister & Joshua Pevehouse, Garmin

Lightning Talk: CI/CD Cybersecurity Guide — Open Source Tools to Improve DevOps Security —
Kate Scarcella, Independent

AI in CI/CD Without the Hype: Practical Patterns for Platform Engineers —
Jennifer Mulford, Okta

KernelScript: Unifying EBPF, Userspace, and Kernel Extensions in One Language —
Cong Wang, Multikernel Technologies

Image Composer Tool: Declarative Multi-Distro Linux Image Builds From Packages —
Mats Agerstam & Alpesh Rodage, Intel Corporation

From FreeRTOS To Zephyr: A Practical Migration Guide for Embedded Developers —
Jacob Beningo, Beningo Embedded Group

Talks I attended — Day 2

OPEN SOURCE SUMMIT 2026 • MINNEAPOLIS

1. **Keynote Panel: From Repo to Rocketship: How Open Source Foundations Supercharge AI Ecosystems** — Deepyaman Datta, Open Source Maintainer; Jakub Kuderski, AMD; Libby Clark, AWS; Lin Sun, Solo.io; Stephen Chin, Neo4j

2. **Keynote: Strands - From Framework to Harness** — Belle Guttman, Senior Software Engineering Manager, AWS

3. **Keynote: Where AI Meets the Physical World: The Robot MCP Ecosystem as an Open Bridge Between AI and Robotics** — Rohit John Varghese, Director of Systems Engineering and Product, Contoro Robotics

4. **Corporate Vice President, Azure Cloud Native and Management Platform & Co-Founder, Kubernetes Open Source Project**

5. **State of Embedded Linux** — Walt Miner, The Linux Foundation

6. **Easy Bring-up Your RISC-V SBC Using Yocto Project - RISC-V Architecture Layer** — Khem Raj, Comcast

7. **Securing Software Workflows** — Tabatha DiDomenico, G-Research Open Source; Kadi McKean, ReversingLabs; Stacey Potter, OpenSSF; Katherine Druckman, JetBrains

8. **Architecting for Onboarding: Building a "Docs-as-Code" Pipeline for Open Source Sustainability** — Sai Sravan Cherukuri, Independent Contributor

9. **Secure Boot for Embedded Linux: Explained in Simple Words** — Roy Jamil, Ac6

10. **Optimize Linux Kernel To Fit Microcontrollers With 1 MB RAM** — Jim Huang & Chisheng Chen, National Cheng Kung University

Talks I attended — Day 3

OPEN SOURCE SUMMIT 2026 • MINNEAPOLIS

Keynote: Linus Torvalds,
Creator of Linux & Git, in Conversation with
Dirk Hohndel, Founder, DH Consulting

Keynote: How Maintainers Can Build Their Way Through the AI Flood — Madelyn Olson, Valkey Project Maintainer and Principal Engineer, AWS & Jacob Murphy, Valkey Project Maintainer and Software Engineer, Google

Keynote: Zephyr: By Developers, For Developers — Kate Stewart, VP Dependable Embedded Systems, The Linux Foundation

Keynote: Free to Use, Not Free to Run: Reinventing Package Registries — Robin Bender Ginn, Executive Director, OpenJS Foundation

OpenEmbedded / Yocto BoF — Colin McAllister, Garmin & Chuck Wolber, The Boeing Company

It Works on My Bench (And Nowhere Else): DevOps for Embedded Systems — Colleen Lake, GitLab

Using Embedded Linux for Autonomous Robot Control — Chloe Zhu, The Admissions Authority

Building the Simplest Possible Linux System — Rob Landley, Hobbyist

THE LINUX FOUNDATION
OPEN SOURCE SUMMIT

AI and security in the open source world

Open Source Summit 2026 • Minneapolis

1

Exploitation is now faster than patching

Mean time to exploit:

–7 days

2018: 63 days → 2022: 32 days → 2024: 5 days → 2026: –7 days

2

AI accelerates both builders and attackers

The classic discover → disclose → patch → deploy cycle was designed for a slower adversary.

3

Security must move earlier

Open source projects need automated scanning, SBOMs, reproducible builds, secure defaults and faster maintainer workflows.

4

Maintainers are now part of the security perimeter

Package registries, dependencies and CI/CD pipelines are strategic infrastructure.

AI and open source: acceleration and pressure

Open Source Summit 2026 • Minneapolis

1 AI is the next abstraction layer

Machine code → C/Unix → Python/ Ruby/JS → Stack Overflow + GitHub → AI agents conducting OSS

2 Open source becomes the substrate

The new abstraction doesn't replace OSS — it turns it into the standard library AI relies on.

3 Work is being redistributed

31% projected technical hiring increase

8% entry-level IT role growth

upskilling over external hiring

4 Open models are catching up fast

Closed-to-open parity compressed to 3–6 months

~6× cheaper

~90% frontier performance

30% global usage

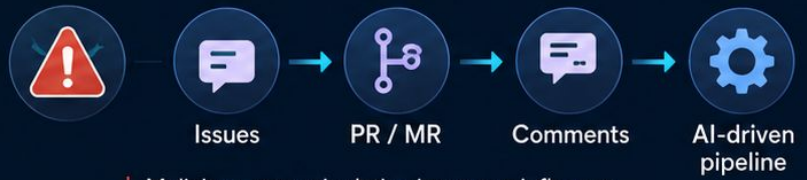
AI in CI/CD Without the Hype

Practical Patterns for Platform Engineers — Jennifer Mulford, Okta

1. **Speed is not a substitute for judgment**
   Faster output does not replace careful human review.
2. **AI is a yes-man; monitor it closely**
   Treat AI suggestions critically and watch its behavior closely.
3. **AI assists, humans decide**
   Human review must remain a required step.
4. **Prompt injection is a new CI/CD threat model**
   Issues, PRs, MRs, and comments can influence AI-driven workflows.
5. **Establish AI accountability**
   Decisions should be visible, reviewable, and auditable.

AI-Enabled CI/CD Decision Flow

Prompt Injection Risk in the Pipeline



Malicious or manipulative input can influence AI-assisted actions.

Use AI to accelerate delivery — not to outsource judgment.

Image Composer Tool

Declarative Multi-Distro Linux Image Builds From Packages — Mats Ågerstam & Alpesh Rodage, Intel Corporation

1. **Declarative image builds**
   Build Linux images from package definitions instead of fragile custom scripts.
2. **Multi-distro support**
   Supports multiple Linux distributions and versions.
3. **Many output formats**
   Can generate RAW, ISO, VHD, VHDX, QCOW2, and VMDK images.
4. **Reproducible builds**
   Package version pinning improves repeatability, traceability, and auditability.
5. **Project status**
   Non-interactive install is in progress; automatic resizing is not available yet.

What stands out

- Package version pinning
- Manifest / SBOM-oriented traceability
- Cross-distro flexibility
- Designed to simplify custom Linux image creation

Open-source project

github.com/open-edge-platform/image-composer-tool

A promising approach for building repeatable Linux images across distros from packages.

Less scripting, more reproducibility.

Open Source Summit North America | Embedded Linux Conference

Secure Boot for Embedded Linux

Explained in Simple Words — Roy Jamil, Ac6

1. **Only run approved software**
   Secure Boot verifies code before it runs.
2. **Trust starts in hardware**
   Boot ROM and eFuses anchor the Root of Trust.
3. **Each stage verifies the next**
   ROM → bootloader → U-Boot → Linux kernel → rootfs.
4. **Signing is simple plumbing**
   Hash + private key signature; public key verifies.
5. **Know the limits**
   It does not fix bugs, protect userspace, or save leaked keys.

Chain of trust

Common mistakes

- Unsigned DTB
- Open U-Boot console
- Fuses not blown
- Keys on developer laptops
- Only testing the happy path

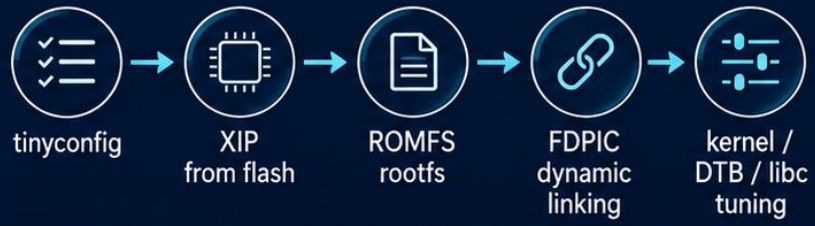
Trust every boot — but still secure everything after boot.

Optimize Linux Kernel To Fit Microcontrollers With 1 MB RAM

Jim Huang & Chisheng Chen, National Cheng Kung University

1. **Linux-noMMU is now practical**
   Running Linux on MCUs is no longer just experimental.
2. **XIP is the key enabler**
   Execute kernel and apps directly from flash to reduce RAM use.
3. **1 MiB RAM demo succeeded**
   On STM32H750, Linux 7.x boots with busybox and about 196 KiB free RAM after boot.
4. **Shared libraries still work**
   FDPIC enables dynamic linking and Linux-style shared libraries on noMMU systems.
5. **A richer system is possible**
   On STM32F429, they showed graphics, touchscreen, and a cellphone-style prototype.

How they made it fit



Goal: keep code in flash, keep only writable data in RAM.

Notable results

- STM32H750: Linux v7.x on a 1 MiB RAM microcontroller
- Unused device-tree cleanup improved free RAM from 116 KiB to 196 KiB
- STM32F429 prototype booted with about 6180 KiB free RAM
- XIP saved about 332 KiB RAM in the feature-rich setup

Linux on tiny MCUs is real: less RAM waste, more reuse, and better maintainability.

Building the Simplest Possible Linux System

Rob Landley, Hobbyist

1. 

Toybox keeps the basics small

Toybox bundles many common Linux command-line utilities into one small BSD-licensed multicall binary.

2. 

The goal is a minimal self-hosting Linux

The idea is to build the simplest Linux system that can rebuild itself from source code.

3. 

Only a few building blocks matter

At the core: command-line utilities, a compiler, a C library, and a kernel.

4. 

It is practical, not just theoretical

Toybox is used in AOSP/Android and helps support a cleaner, self-hosting toolchain.

5. 

Simplicity is the lesson

By stripping Linux down to essentials, the talk shows what is truly required to boot, build, and understand a system.

Why it stood out

A great reminder that Linux can be understood from first principles

Small, fast, and reasonably standards-compliant

Useful for bootstrapping, education, and experimentation

A minimalist mindset can improve maintainability

Open-source references

landley.net/toybox/

codeberg.org/landley/toybox

Toybox also provides a "make root" path to create a simple toybox chroot, and with Linux source it can build a tiny bootable system for QEMU.

Sometimes the best way to understand Linux is to build the smallest one possible.

Embedded Linux & RTOS: Big Themes at the Summit

Open Source Summit North America 2026

1. **A huge topic across the conference**
   Embedded Linux, build systems, RTOS choices, and platform engineering came up repeatedly across the week.

2. **Yocto was everywhere**
   Used across organizations such as Boeing, Garmin, and Honda to build embedded Linux systems.

3. **Yocto in major products**
   Amazon's Vega OS was highlighted as a Yocto-based direction replacing Android-based Fire OS.

4. **Zephyr vs FreeRTOS**
   A recurring comparison point: ecosystem, portability, tooling, and long-term maintainability.

5. **ROS + AI**
   Robot Operating System is increasingly paired with AI for more capable robotics workflows.

Linux beyond general-purpose systems

- **AGL** — Automotive Grade Linux for connected vehicles and in-car platforms
- **SGL** — Space Grade Linux for aerospace and space-focused use cases
- Embedded Linux remains a common foundation once systems outgrow simple firmware
- The conference showed Linux scaling from small devices to specialized industry platforms

Big takeaway

Embedded Linux is not niche — it is a core platform story across automotive, robotics, aerospace, and industrial systems.

Yocto stood out as the common build foundation, while RTOS and Linux choices remain central architectural decisions.

From Yocto to robots to vehicles and space systems: embedded Linux was everywhere.

RISC-V and Linux: why it matters

Open Source Summit North America 2026

1 A major architecture theme

RISC-V support in the Linux kernel came up as a strong topic across embedded and platform engineering talks.

2 RISC-V vs x86

RISC-V is an open instruction set architecture; x86 is a mature proprietary architecture dominated by Intel and AMD.

3 Why RISC-V is interesting

Its modular design lets vendors build custom SoCs while still targeting a common ISA and Linux software stack.

4 Linux momentum

Kernel support, toolchains, boot flows, and board support continue to improve, especially for embedded and edge systems.

5 Where it stands today

RISC-V is growing fast, but it is still earlier in its ecosystem maturity than x86 in mainstream computing.

x86 vs RISC-V

x86	RISC-V
Very mature desktop and server ecosystem	Open ISA and easier to customize
Strong performance and compatibility	Strong appeal for embedded, education, and research
Decades of legacy complexity	Growing Linux ecosystem
ISA controlled by a few vendors	Less mature hardware and software support

Advantages and disadvantages

Advantages	Disadvantages
Open standard, no ISA licensing lock-in	Fewer mainstream systems and boards
Custom extensions and design flexibility	Tooling and firmware ecosystem still maturing
Good fit for embedded and specialized devices	Lower software compatibility in some areas
Encourages ecosystem diversity	High-end performance story is less mature than x86

RISC-V will not replace x86 overnight — but for Linux, it is becoming a serious open alternative for embedded, edge, and future platform design.

Linus Torvalds: Linux, AI, and Security

Keynote conversation with Dirk Hohndel • Open Source Summit North America 2026

1 Security disclosures are evolving

The Linux kernel now documents a clearer security disclosure path, with updated guidance for reporting security bugs.

2 AI-assisted bug reports change the rules

Because of the flood of duplicate reports, if AI was used to identify a bug, it should be treated as public information rather than a private disclosure.

3 Sashiko points to agentic review

Torvalds highlighted Sashiko, an agentic Linux kernel code-review system, as an example of how AI may assist engineering workflows.

4 Local AI is the next big thing

A major theme was that useful AI will increasingly run locally, closer to developers and users.

5 AI changes programming, not fundamentals

Programming practices will evolve, but the core engineering fundamentals still matter. As a reminder: the compiler still writes the final executable code.

Standout ideas

- 600k GitHub projects are considered enterprise-critical.
- Speed and automation do not replace good engineering judgment.
- AI can help developers, but it does not change the fundamentals of system design.
- The scale of incoming reports means process discipline matters more than ever.

References / examples

- Linux security disclosure process: docs.kernel.org/process/security-bugs.html
- Sashiko: agentic Linux kernel code review
- GuitarPedal side project: github.com/torvalds/GuitarPedal

The GuitarPedal project added a lighter personal note to the discussion and showed how experimentation still matters.

AI is changing how we program — but not why good engineering, sound process, and human judgment still matter.

Recordings, slides and pictures

Open Source Summit 2026 • Minneapolis

Recordings

Watch the recorded sessions.

YouTube
youtube.com/@LinuxfoundationOrg/videos

Slides

Browse the presentation materials.

Sched
osselcna2026.sched.com/overview/audience/Yes?iframe=no

Pictures

See the official conference photo album.

Flickr
flickr.com/photos/linuxfoundation/albums/72177720333089343/

Scan the QR codes to access the official resources.

THE LINUX FOUNDATION
OPEN SOURCE SUMMIT
NORTH AMERICA

Embedded Linux Conference

Open Source Summit North America 2027

Next stop: Vancouver

May 17–19, 2027

Vancouver

Learn more

Scan the QR code

https://events.linuxfoundation.org/open-source-summit-north-america/

OPEN SOURCE SUMMIT
NORTH AMERICA

EMBEDDED
LINUX
CONFERENCE

Questions?

Thank you!

Open Source Summit 2026 • Minneapolis

André Courchesne