KUBECON + CLOUDNATIVECON NORTH AMERICA 2026

Legacy-Free, Zero-Maintenance Bare Metal Kubernetes Clusters

With RISC-V and Immutable Linux



Felicitas Pojtinger · LiveKit

THE QUESTION

What do we actually run on, and who controls it?

TODAY

Most of the stack is proprietary

FIRMWARE

a blob

BOOT PATH

signed by someone else

IMAGE

locked to one build

CONTROL PLANE

rented

you can use it, but you can't inspect or replace it

OWNERSHIP

You should be able to own everything

metal | firmware | kernel | image | trust system | cluster

LiveKit

OUTLINE

Six layers, six ideas, seven demos

01
Silicon
the chip

02
Enablement
the hands

03
Firmware
the chain

04
Image
the unit

05
Trust
the proof

06
Cluster
the workload

Decades of legacy

X86

- Only two companies can make the chips
- The license is expensive and private
- The management engine and the PSP run below the OS
- The firmware is a pile of blobs

ARM

- Design your own core, get sued
- No standard boot path
- TrustZone is a black box
- Downstream kernels fork and drift

01 · SILICON

RVA23

LiveKit

The timeline so far

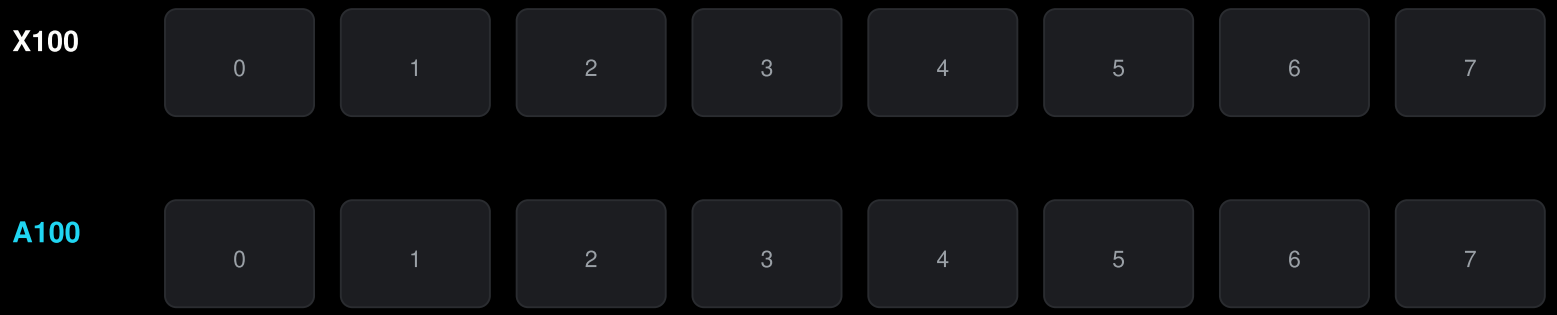


~25%

of global chip production in 2026 is RISC-V already

LiveKit

The K3 processor



8 + 8 cores, one board

60 TOPS at INT4

256 and 1024 bit vectors, one binary for both

What comes next

ALIBABA

C950

SIFIVE

P550

SPACEMIT

K3, then X200

TENSTORRENT

Atlantis

BASELINE

RVA23, many implementations

SOFTWARE

Software support

The software already supports and mandates RVA23, before the first chips even arrived

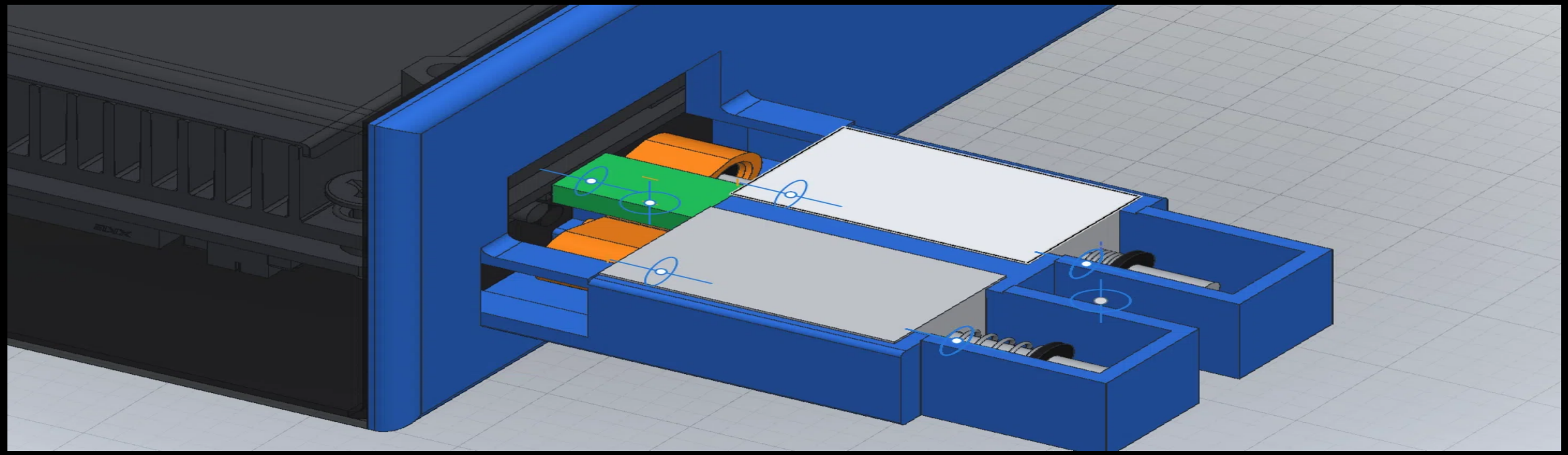


02 · ENABLEMENT

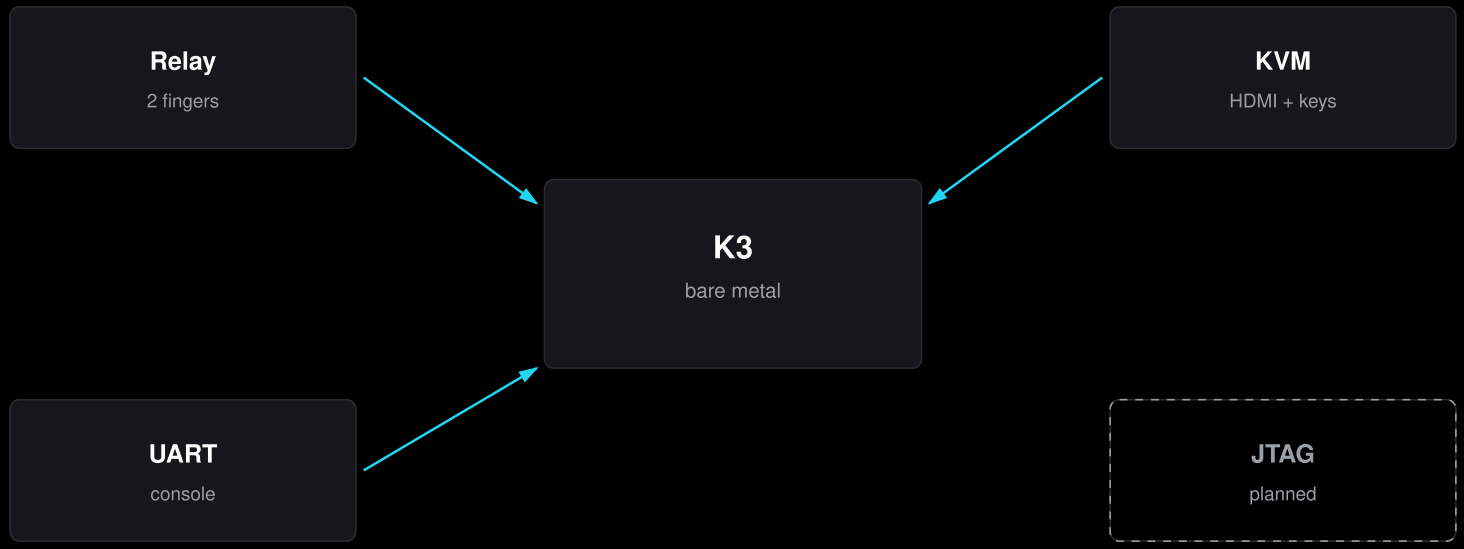
Pushrod

Autonomous, hands-free hardware enablement

The button presser



The harness



fastboot over USB → Titanflasher protocol, reimplemented in Go

MCP server

TOOLS
typed, not shell strings

STATE
stateless

FILES
over the wire

Tool naming

RELAY.*
relay board control

TARGET.*
button sequences

SERIAL.*
console and shell

FILES.*
uploads

FLASH.*
images and partitions

KVM.*
screen and input

Flash order



Images are the distributable asset

One signature, one switch, one rollback

Buildroot tree

TREE

- make and Kconfig, one external tree
- One toolchain, one download cache
- Two defconfigs, K3 and QEMU

TOOLING

- ukify and sbsign for the UKIs
- mkfs.erofs and veritysetup for /usr
- systemd-repart for the extensions
- genimage for the disk and the GPT

Disk layout

K3 and QEMU



sysupdate fills the second usr slot with the first update

DEMO

DEMO 01

Flash a blank board

01

03 · FIRMWARE

From a blank chip to a boot menu

K3	BROM	FSBL	ESOS	OpenSBI	EDK2	systemd-boot	UKI
QEMU	SPL			OpenSBI	EDK2	systemd-boot	UKI

QEMU also has a TPM

What we could prove

K3

◐ Secure Boot, if we blow the eFuses

We leave them unblown, so the board stays debuggable

QEMU

✓ Secure Boot and a software TPM

The full chain, measured end to end

◐ disabled because the eFuses stay unblown

DEMO

DEMO 02

Fix a black screen

02

A package manager rewrites /usr in place

The image model takes the image as the unit, so a switch is one boot entry

Two generations before this talk

COREOS

A/B image

One image into slot A or B, the bootloader flips

RPM-OSTREE

layering

The old commit stays on disk, so the boot menu still offers it

bootc

THE UNIT
an OCI image

THE SWITCH
one boot entry

THE CHANNEL
a registry

THE KERNEL
in /usr/lib/modules

composefs

EROFS → fs-verity → read-only root → writable overlay

Secure Boot checks the UKI, the UKI carries the digest, every read checks the file

What this talk uses instead

BOOTC

Extend the image

A new full OS artifact, because /usr is one image

SYSUPDATE

Extend the root

A sysext merges a small subtree, and one transfer carries one artifact

Who else ships this

GNOME OS

BuildStream

In preview, updates with sysupdate, apps from Flatpak

PARTICLEOS

mkosi and OBS

Fedora packages, signed UKI and disk image, IPE enforces integrity

AOS

Amutable OS

What this tree adds

SYSEXT

a subtree of /usr

Merges only when its release file matches the host

CONFEXT

a subtree of /etc

The same model for configuration

DEMO

DEMO 03

Push an update

03

A TPM is a small crypto processor

EK

identifies the chip

AK

signs the quotes

PCR

a hash register

A PCR only extends

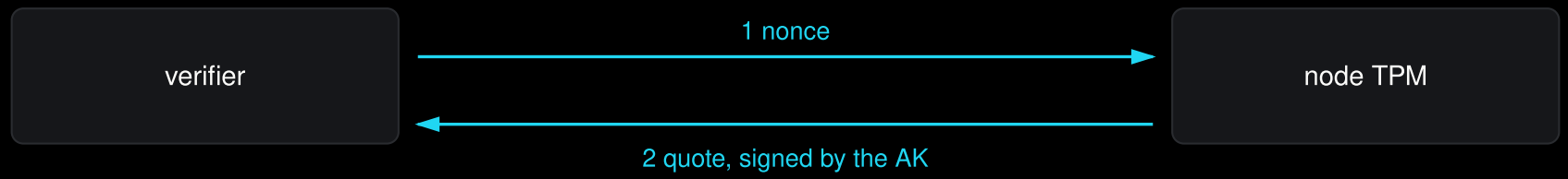
one register, chained



PCR 11 holds the UKI

nothing can set it back

Remote Attestation Challenges



3 EK certifies the AK, the vendor certifies the EK

4 the verifier checks the signature, the nonce, and the PCRs

nonce → quote → verify → consume

LiveKit

pcrlock and systemd-report

PCRLOCK

local policy

Predicts the PCRs, then the TPM unlocks secrets without a remote verifier

SYSTEMD-REPORT

export

The report, then one signature per signer, so the receipt is self-contained

DEMO

DEMO 04

Secure boot and attestation

04

05 · TRUST

Attestable User Consent

key residency

provable consent

OS, or a key

LiveKit

Assurance Level Tiers

	Host	FIDO2	Card
Tier 0	✓	—	—
Tier 1	✓	✓	✓
Tier 2	✓	✓	✓

DEMO

DEMO 05

Tier and wallet check

05

Secure Fleet Management Primitives

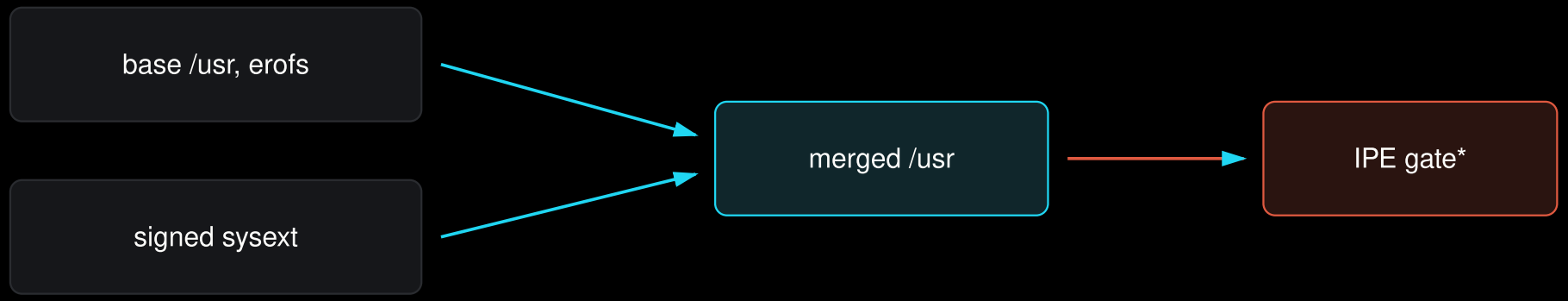
a signed image

a signed update

an attested boot

provable consent

Signed system extensions



an unsigned image never reaches exec

IPE gate*

*IPE gates execution

LiveKit

Demo Scope

ON STAGE
K3, single node

READY
QEMU + K3

READY
RVA22 bare metal

READY
RVA23 beta

Talk to me afterwards or at the booth for more info

DEMO

DEMO 06

Signed Kubernetes

06

07 · MIGRATION

Checkpoint/Restore and Migration

What we attest, and when

BEFORE, ON THE SOURCE

TPM quote, nonce, PCR 11, and counter

quote nonce PCR 11 counter

The server knows which attested machine produced the checkpoint

AFTER, ON THE DESTINATION

Same proof, from the other guest

quote nonce PCR 11 counter

The server knows which attested machine now runs the process

DEMO

DEMO 07

Move a live process

07

CLOSE

You own the stack, and you hold the keys

01
Silicon
the chip

02
Enablement
the hands

03
Firmware
the chain

04
Image
the unit

05
Trust
the proof

06
Cluster
the workload

LiveKit

The Future

Hardware that is just as open as software today

THANK YOU

Thank you



Felicitas Pojtinger

LiveKit

WEB	felicitas.pojtinger.com
MASTODON	@pojntfx@mastodon.social
BLUESKY	@felicitas.pojtinger.com
LINKEDIN	linkedin.com/in/pojntfx
CODEBERG	codeberg.org/pojntfx
GITHUB	github.com/pojntfx

CODE

Pushrod

codeberg.org/pojntfx/pushrod

LIVEKIT BOOTH 254

Careers

livekit.io/careers

SESSIONIZE

Feedback

sfeedback.com/e35dxm

LiveKit