Brussels, FOSDEM 2026

Review of kernel and user-space AI ASICs support on Linux

Jakov Petrina Trnski

bytelab

January 31st, 2026.

whoami

Jakov Petrina Trnski

- Almost a decade in embedded Linux
- Tried to build a Yocto/Buildroot/OpenWrt competitor
- Networking focused, recent pivot to lower-level Linux hardware enablement

Byte Lab

- Product design company - “From Idea to Market”
- Focusing on embedded: hardware, firmware, and software
- Development-focused company with manufacturing capabilities

Hardware for AI

VPU, GPU, TPU, or NPU: Just an ASIC?

What does training need?

- Backpropagation is still the fundamental algorithm
- GPU good but not optimal, unused 3D graphics blocks on-die
- Jouppi et al. (2023). TPU v4: … with ***Hardware Support for Embeddings***

What does inference need?

- Matrix multiplications or convolutions, apply activation functions, …

Purpose-specific AI/ML acceleration

- i.e. image, text, sound...

Systolic array vs. meshed compute units

- Fundamental tradeoffs, conditional branching…

TeraOPS vs. TeraFLOPS

- Holy grail >100 TOPS/ Watt

AI acceleration timeline

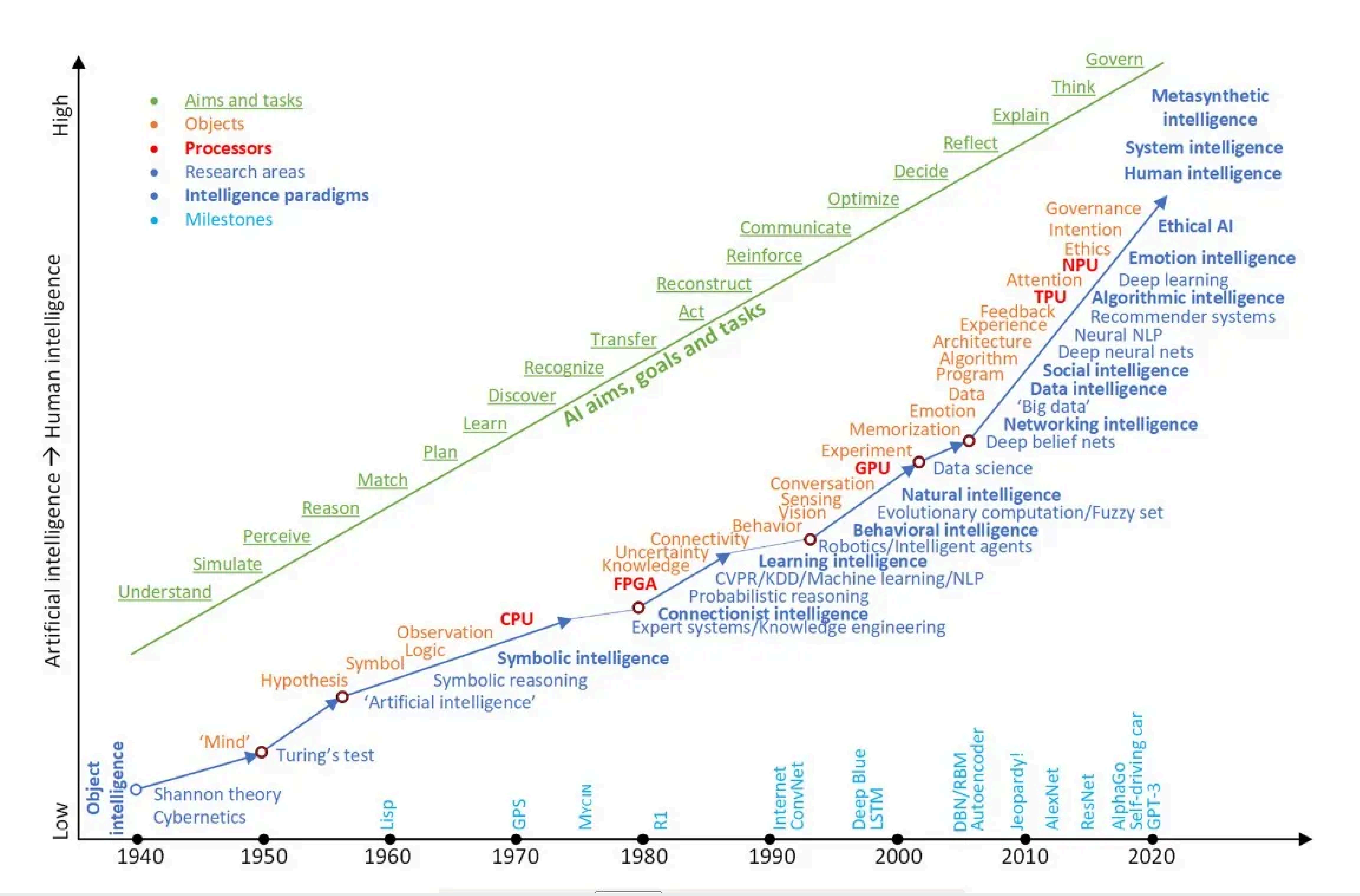
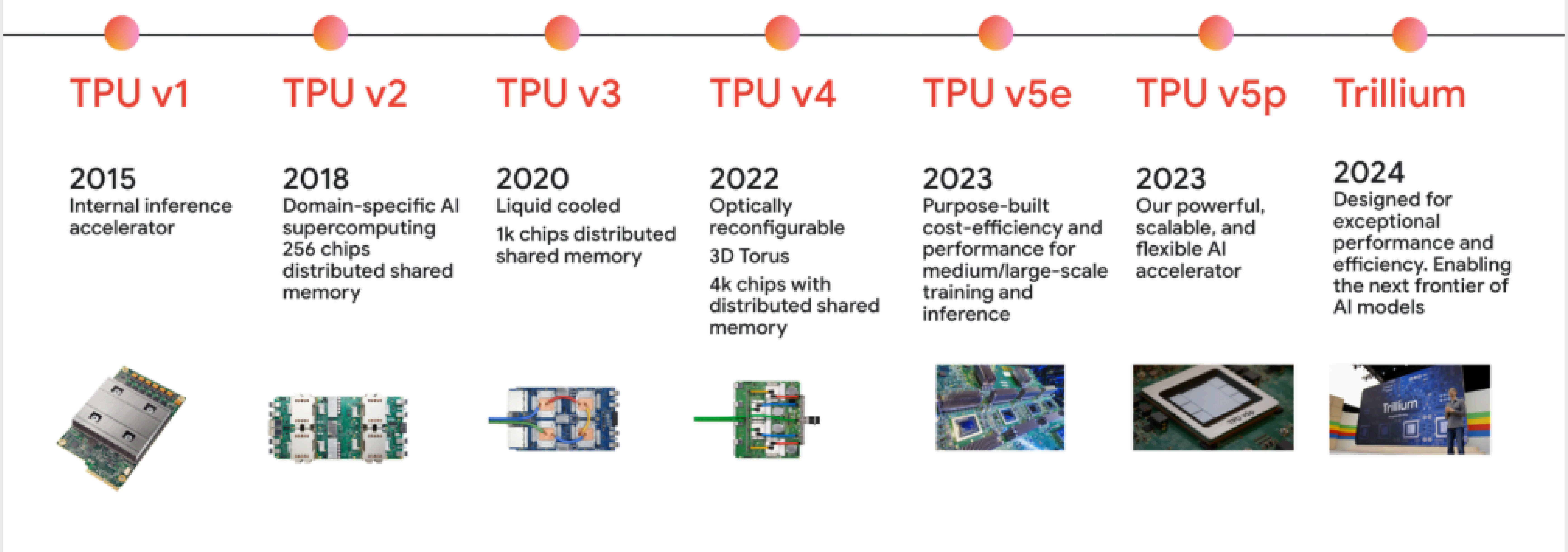


Image credit: altimetrikpoland.medium.com

Google TPU Timeline



TPU v1

2015
Internal inference accelerator

TPU v2

2018
Domain-specific AI supercomputing 256 chips distributed shared memory

TPU v3

2020
Liquid cooled

1k chips distributed shared memory

TPU v4

2022
Optically reconfigurable

3D Torus

4k chips with distributed shared memory

TPU v5e

2023
Purpose-built cost-efficiency and performance for medium/large-scale training and inference

TPU v5p

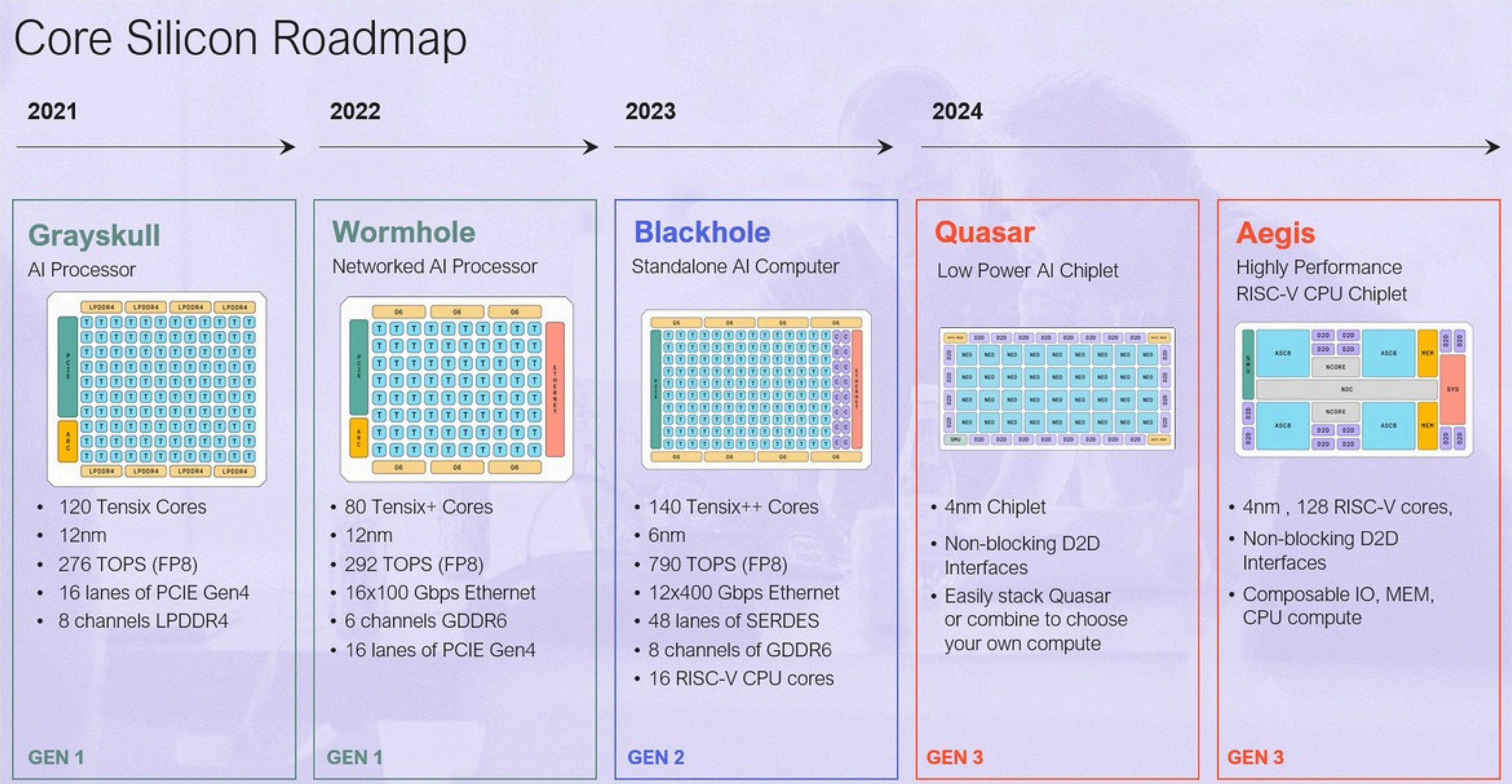
2023
Our powerful, scalable, and flexible AI accelerator

Trillium

2024
Designed for exceptional performance and efficiency. Enabling the next frontier of AI models

Image credit: Google

Tenstorrent Timeline



Core Silicon Roadmap

2021 → 2022 → 2023 → 2024 →

Grayskull
AI Processor

- 120 Tensix Cores
- 12nm
- 276 TOPS (FP8)
- 16 lanes of PCIE Gen4
- 8 channels LPDDR4

GEN 1

Wormhole
Networked AI Processor

- 80 Tensix+ Cores
- 12nm
- 292 TOPS (FP8)
- 16x100 Gbps Ethernet
- 6 channels GDDR6
- 16 lanes of PCIE Gen4

GEN 1

Blackhole
Standalone AI Computer

- 140 Tensix++ Cores
- 6nm
- 790 TOPS (FP8)
- 12x400 Gbps Ethernet
- 48 lanes of SERDES
- 8 channels of GDDR6
- 16 RISC-V CPU cores

GEN 2

Quasar
Low Power AI Chiplet

- 4nm Chiplet
- Non-blocking D2D Interfaces
- Easily stack Quasar or combine to choose your own compute

GEN 3

Aegis
Highly Performance RISC-V CPU Chiplet

- 4nm , 128 RISC-V cores,
- Non-blocking D2D Interfaces
- Composable IO, MEM, CPU compute

GEN 3

Image credit: Tenstorrent

Software stacks for AI

Software

Training vs inference

- Beefy training, but maybe re-training is not
- Beefy inference, usually cloud
- Local / edge AI is lightweight inference <10 TOPS

Stack

- Kernel hardware support
- User-space APIs, runtimes, and frameworks

Frameworks and routines

- TensorFlow, TFLite / LiteRT, PyTorch, Keras, JAX, LangChain, ONNX… *ad infinitum!*

Hardware chooses software for you?

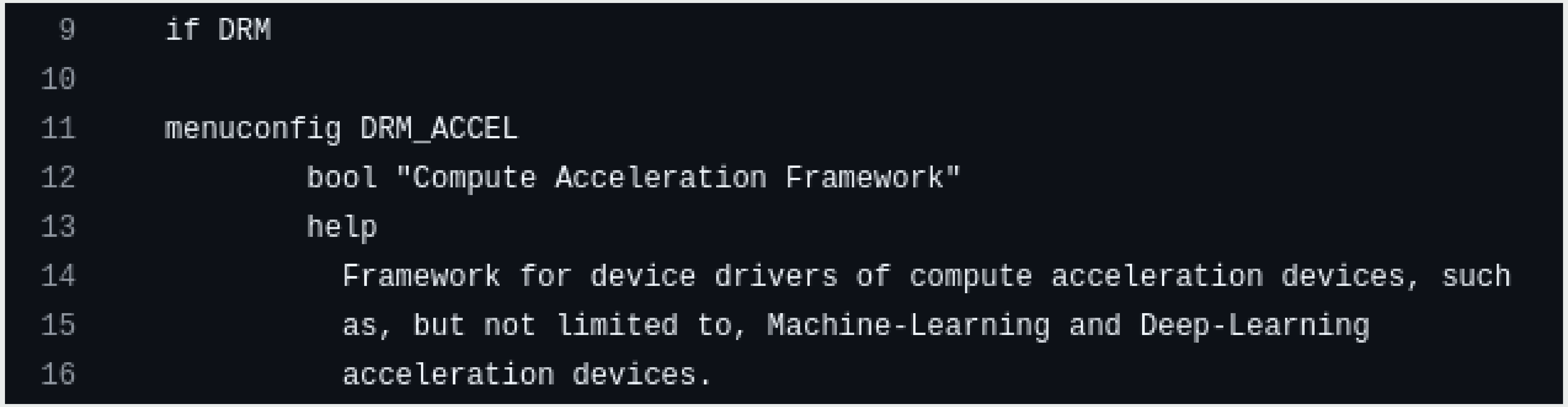
- e.g. airockchip/rknn-toolkit2 and “RKLLM format model”

Linux kernel for AI

Linux kernel for AI

- Is there generic infrastructure for AI like “switchdev” for switching ASICs?

  → There is drivers/accel (CONFIG_DRM_ACCEL):



```
if DRM

menuconfig DRM_ACCEL
	bool "Compute Acceleration Framework"
	help
	  Framework for device drivers of compute acceleration devices, such
	  as, but not limited to, Machine-Learning and Deep-Learning
	  acceleration devices.
```

- Usually DRM render-node style interfaces (GPUs/compute)
- Sometimes skipped, direct hardware access (!) e.g TT-Metalium

Kernel Case Study: Google TPUs

- USB accelerators, development boards, PCIe cards
  → Cloud (train) and Edge (infer) TPUs

- GASKET + APEX kernel driver for Coral Edge TPU v1 (deprecated)
  → https://github.com/google/gasket-driver

- LiteRT new kid on the block, uses “delegate” mechanism
  → Surprisingly supports Qualcomm, MediaTek, Google, and Apple hardware
  → Vendors probably have to add support

Kernel Case Study: Rockchip SoC

A FOSS

- 2024-06-12 [PATCH 0/9] New DRM accel driver for Rockchip's RKNN NPU
- 2025-07-21 [PATCH **v9** 00/10] New DRM accel driver…
- API for userspace in uapi/drm/rocket_accel.h
- Used by the Rocket userspace driver in Mesa3D

B OEM open API + closed firmware blob

- Code running on the NPU ASIC is basically “firmware blob” and uploaded from user-space, similar to Marvell’s Prestera packet switching ASICs
- Kernel portion is an open interface to the closed application running on NPU
- rk-6.1-rkr5.1/drivers/rknpu

Kernel Case Study: AMD, NVIDIA, Intel…

- Does it even work non-proprietarily, CUDA?

- ROCm part in kernel?

- Intel NPU kernel driver
  → Proprietary closed source firmware blob is uploaded to the NPU
  → https://github.com/intel/linux-npu-driver
  → drivers/accel/ivpu & uapi/drm/ivpu_accel.h → /dev/accel/accel0

Kernel Case Study: *et al*

- **Tenstorrent “Tensix Processors”**

→ https://github.com/tenstorrent/tt-kmd → /dev/tenstorrent/%d

- **Qualcomm**

→ Hexagon NPU inherits from Hexagon DSP
→ drivers/media/platform/msm/npu → /dev/msm_npu (old hardware)
→ drivers/char/adsprpc.c → FastRPC (some kind of complex marshalling)

- **MediaTek**

→ Don’t confuse with Airoha NPU on which is a Network Processor Unit for… packet acceleration

- **Apple “Neural Engine”**

→ Haha no… reverse engineered by community?

User-space APIs and tooling for AI

Software Case Study: Rockchip SoC

A FOSS

- Mesa3D
  → rocket: Initial commit of a driver for Rockchip's NPU
  → TFLite Delegate “Teflon”

B OEM open API + firmware blob

- RKNN-Toolkit2

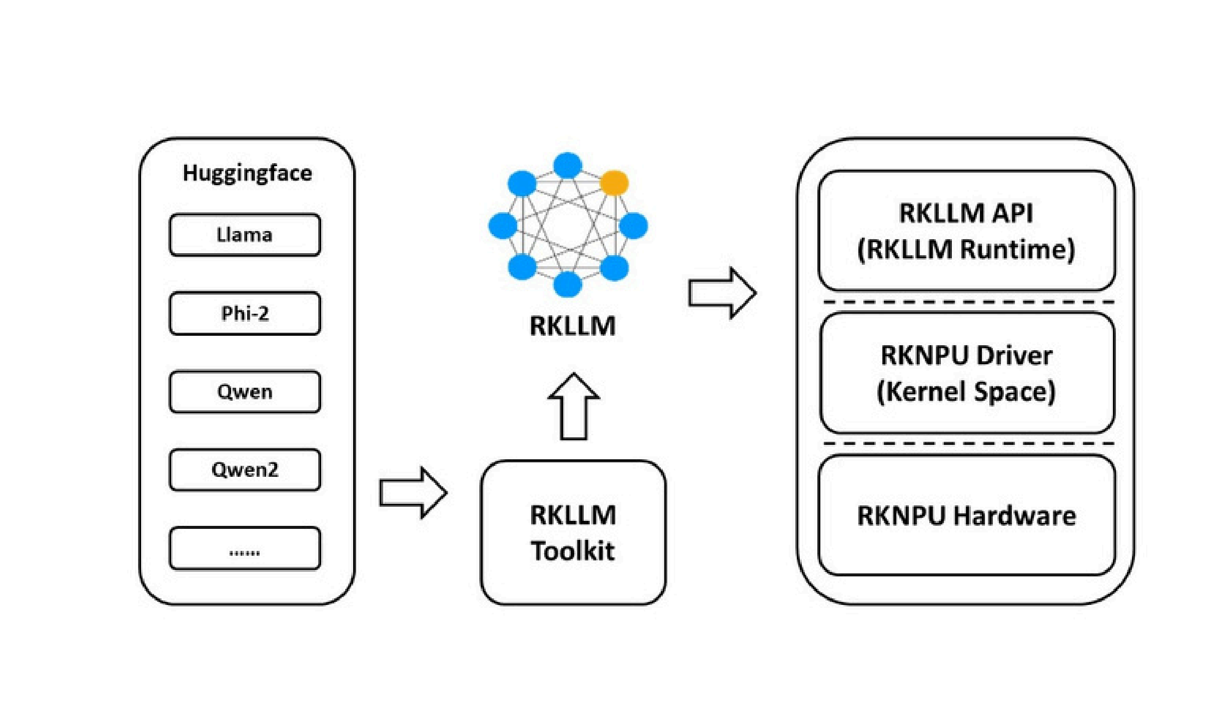
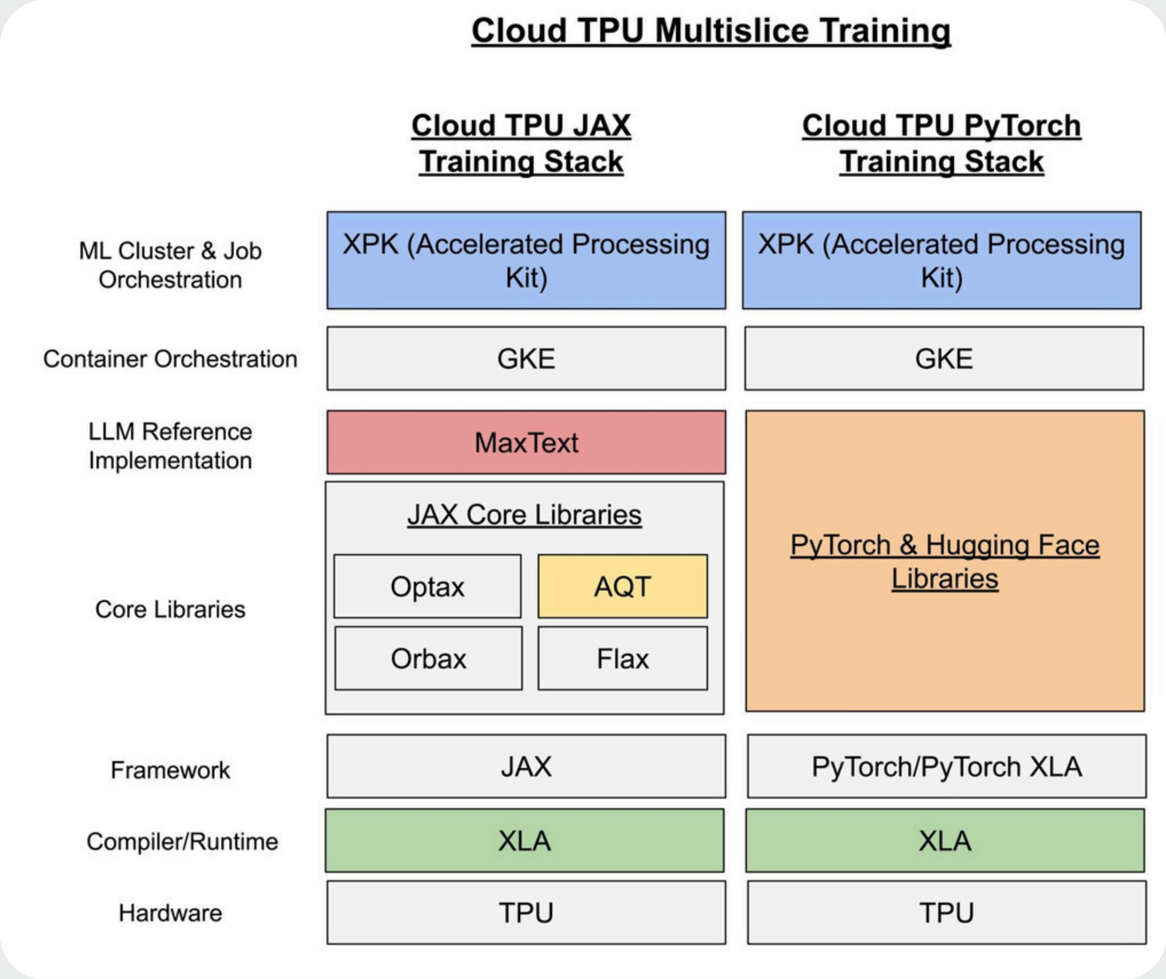


Image credit: Rockchip

Software Case Study: Google TPUs

- https://github.com/google-coral/libedgetpu

- LiteRT (formerly TensorFlow Lite) for Edge TPUs
  → Actually supports various NPU vendors like Qualcomm, MediaTek, Google, Apple



Cloud TPU Multislice Training

	Cloud TPU JAX Training Stack	Cloud TPU PyTorch Training Stack
ML Cluster & Job Orchestration	XPK (Accelerated Processing Kit)	XPK (Accelerated Processing Kit)
Container Orchestration	GKE	GKE
LLM Reference Implementation	MaxText	PyTorch & Hugging Face Libraries
Core Libraries	JAX Core Libraries Optax AQT Orbax Flax	
Framework	JAX	PyTorch/PyTorch XLA
Compiler/Runtime	XLA	XLA
Hardware	TPU	TPU

Image credit: Google

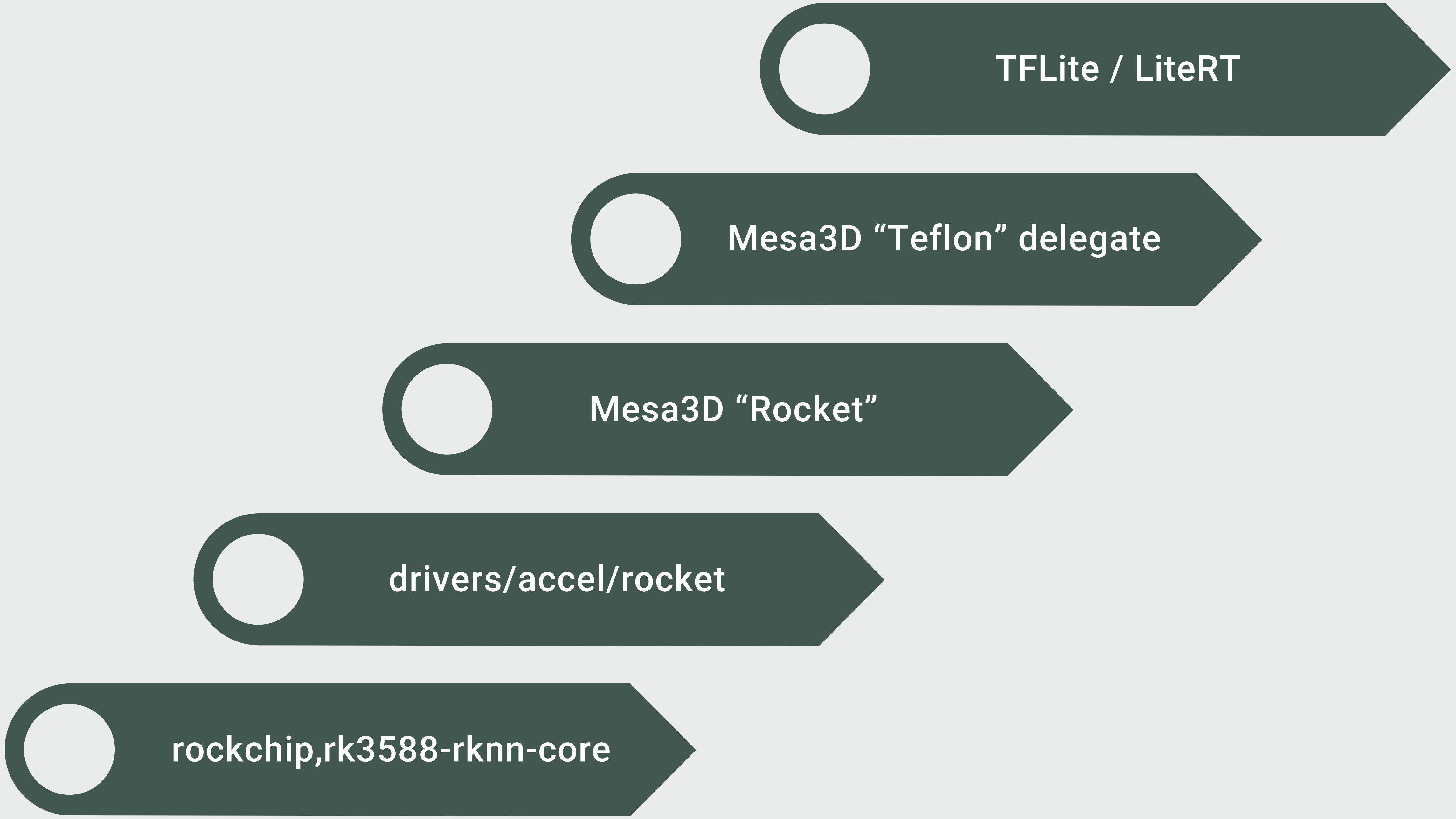
Software Case Study: NVIDIA, AMD, Intel…

NVIDIA CUDA

AMD ROCm

Intel OpenVINO

Fully FOSS AI Stack?



TFLite / LiteRT

Mesa3D “Teflon” delegate

Mesa3D “Rocket”

drivers/accel/rocket

rockchip,rk3588-rknn-core

Takeaways

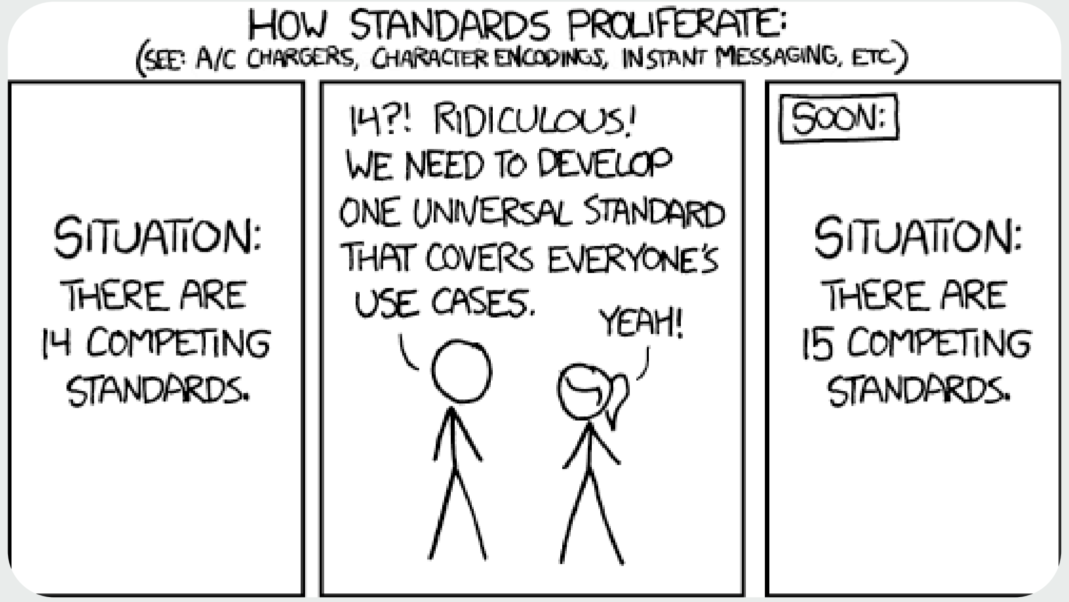

HOW STANDARDS PROLIFERATE:
(SEE: A/C CHARGERS, CHARACTER ENCODINGS, INSTANT MESSAGING, ETC.)
SITUATION: THERE ARE 14 COMPETING STANDARDS.
14?! RIDICULOUS! WE NEED TO DEVELOP ONE UNIVERSAL STANDARD THAT COVERS EVERYONE'S USE CASES.
YEAH!
SOON:
SITUATION: THERE ARE 15 COMPETING STANDARDS.

Image credit: XKCD

1

- **Extreme fragmentation through the whole stack**

  → Can you even “genericize” ASICs?

2

- **Vendor-specificness and rare mainlineness**

  → Every vendor has their own kernel fork / SDK, even model format

3

- **Big difference inference vs. training**

  → Local/edge on commodity hardware

  → Closed cloud frameworks for high-performance

What’s next?

Per-domain ASICs

- “Language” PUs, Transformer-optimized, power, latency, memory, network interconnects

Per-ASIC community Linux hardware enablement

- Tomeu Vizoso’s work on Rockchip (rocket), VeriSilicon Vivante NPU (etnaviv)
- 22 Jul 2025 [PATCH **RFC** 0/2] accel: Add Arm Ethos-U NPU

AIFoundry.org

- AINekko's ET platform
  → Open-source manycore ASIC platform for parallel computing acceleration

- ET backends

... and so much more!

bytelab

Thank you!

Questions?

jakov.petrina@byte-lab.com