

tinyML[®] Summit

Enabling Ultra-low Power Machine Learning at the Edge

Products and applications enabled by tinyML

March 28 – 29, 2023



www.tinyML.org



arm

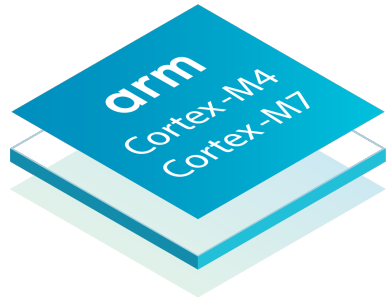


Arm Ethos-U Support in TVM ML Framework

Rahul Venkatram
March 2023

Overview of Cortex-M CPUs

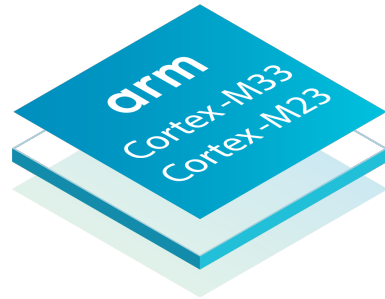
>10 Billion Chips Shipped



Armv7-M

- + Widely adopted and shipped into billions of devices

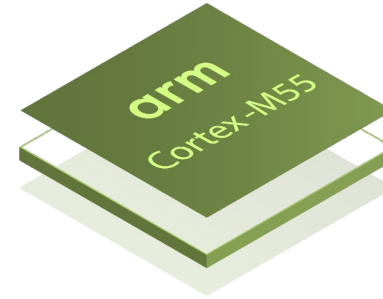
Shipments Ramping Up



Armv8-M
(TrustZone)

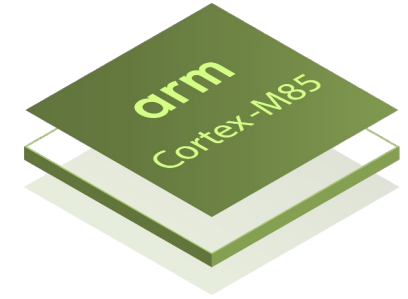
- + First range of CPUs with TrustZone
- + Configurable CPU for mainstream and constrained applications

Released



Armv8.1-M
(TrustZone, Helium)

- + First Helium High Efficiency processor
- + Balanced performance and energy efficiency
- + Advanced ML, RAS and safety features

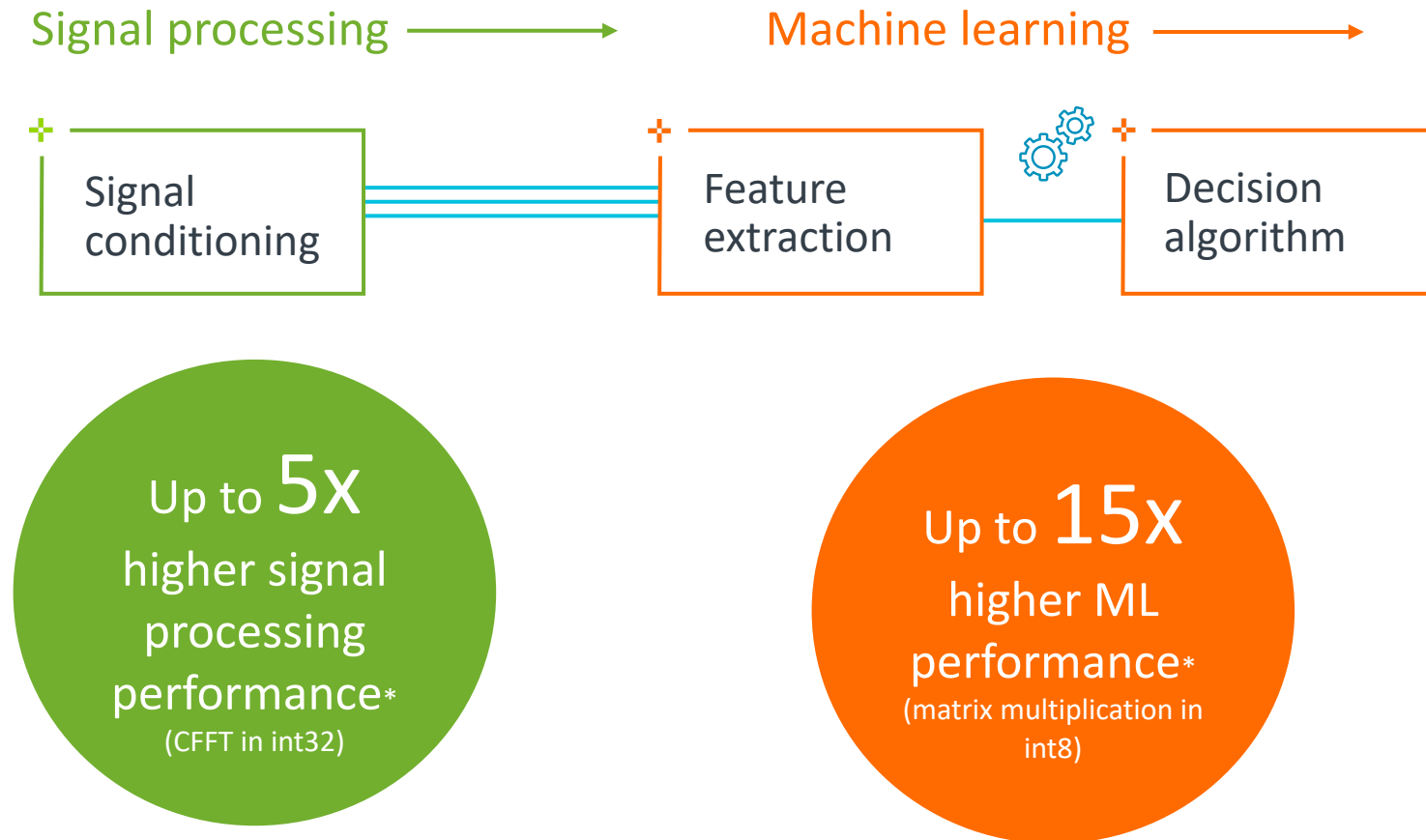


Armv8.1-M
(TrustZone, Helium)

- + Arm's most capable Cortex-M processor
- + Highest scalar and signal processing performance

Helium: the Next Level of Edge Compute

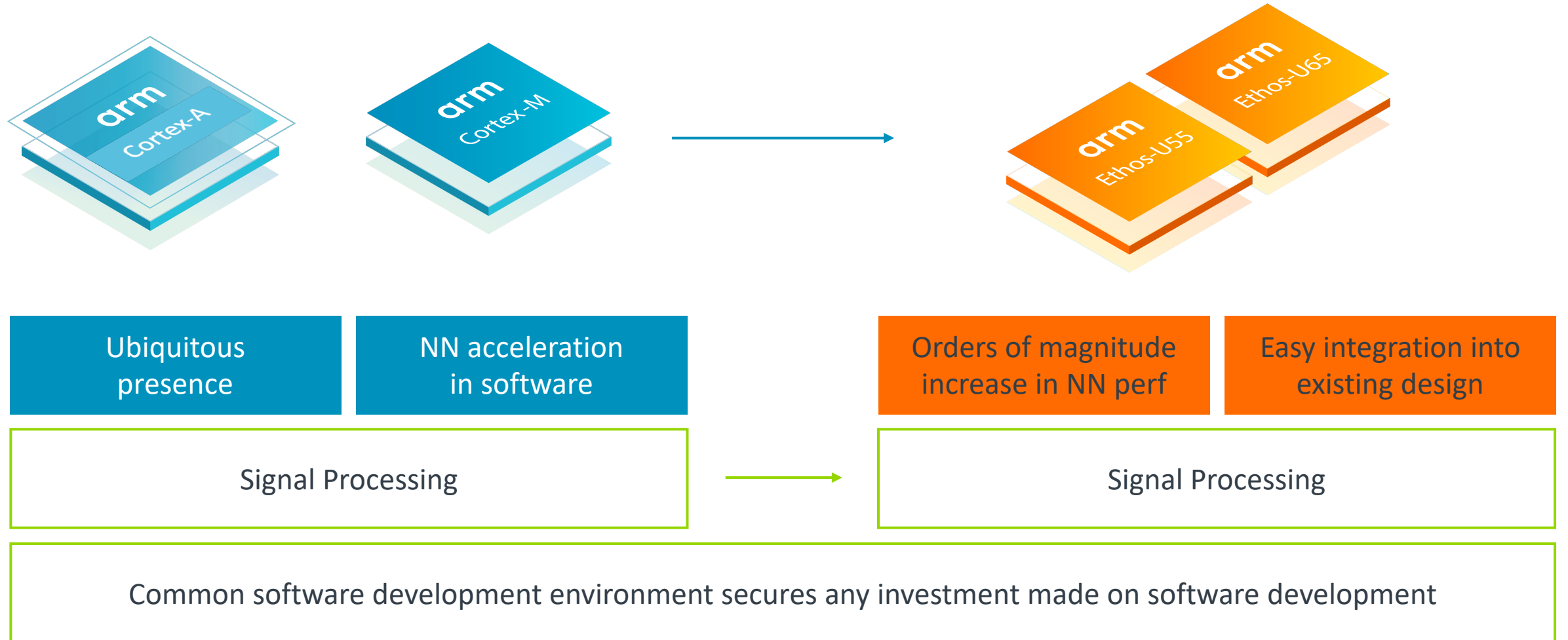
Armv8.1-M architecture introduces Helium - a new vector extension



*Compared to existing Armv8-M implementation

Ethos-U class of NPUs for Embedded Systems

Providing NN Acceleration in Highly Constrained Environments



High level differences between Ethos-U55 and Ethos-U65

Both are instantiations of the same architecture

Ethos-U55

4 configs: 32/64/128/256 MACs/cycle

Designed for SRAM + flash

Host CPU support: Cortex-M55, Cortex-M7,
Cortex-M4 and Cortex-M33

Two 64-bit AXI master interfaces

M0: Full read+write AXI master to SRAM
M1: Read only AXI master to flash

Ethos-U65

2 configs: 256/512 MACs/cycle

Designed for SRAM + DRAM and/or flash

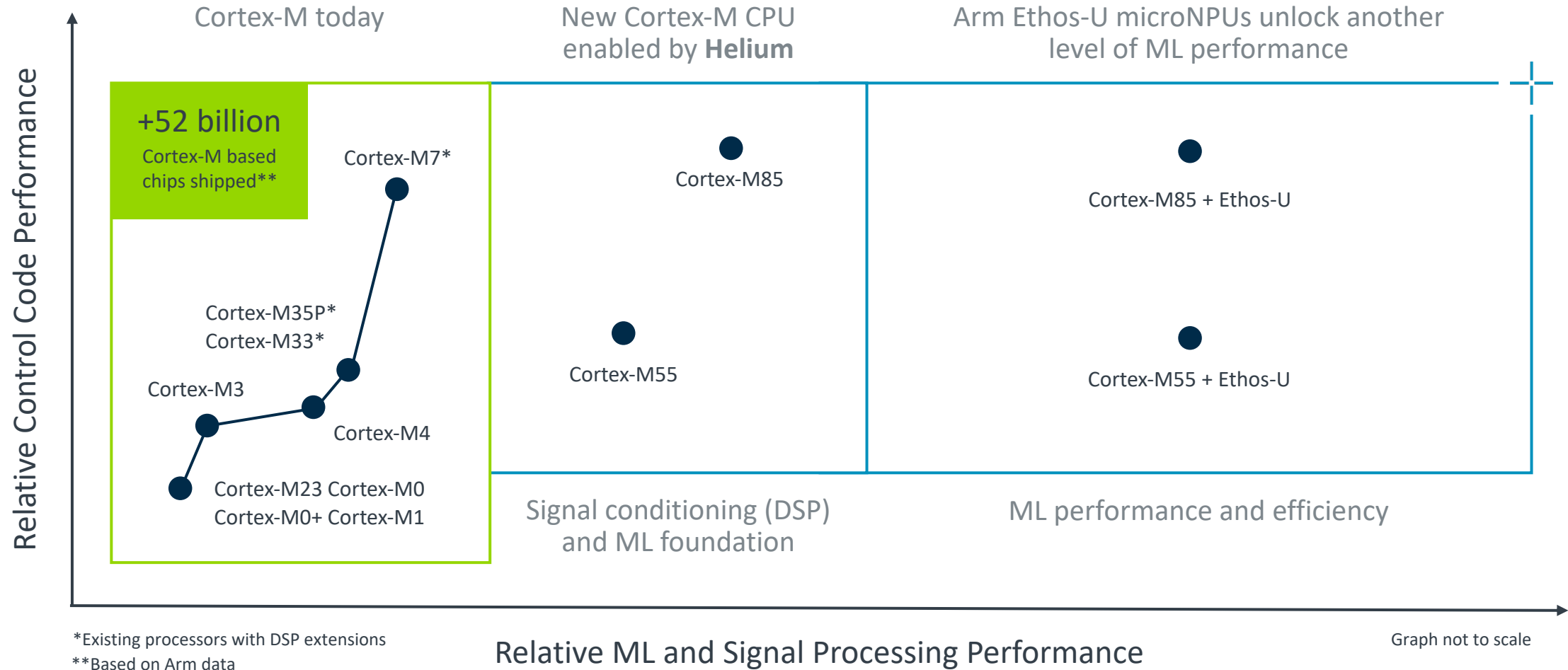
Host CPU support: Cortex-M55 and Cortex-M7

Two 128-bit AXI master interfaces

M0: Full read+write AXI master to SRAM
M1: Full read+write AXI master to DRAM

Cortex-M Pushes Boundaries for Real-time On-device Processing

Enabling New Workloads and Use-cases in a Unified Development Environment



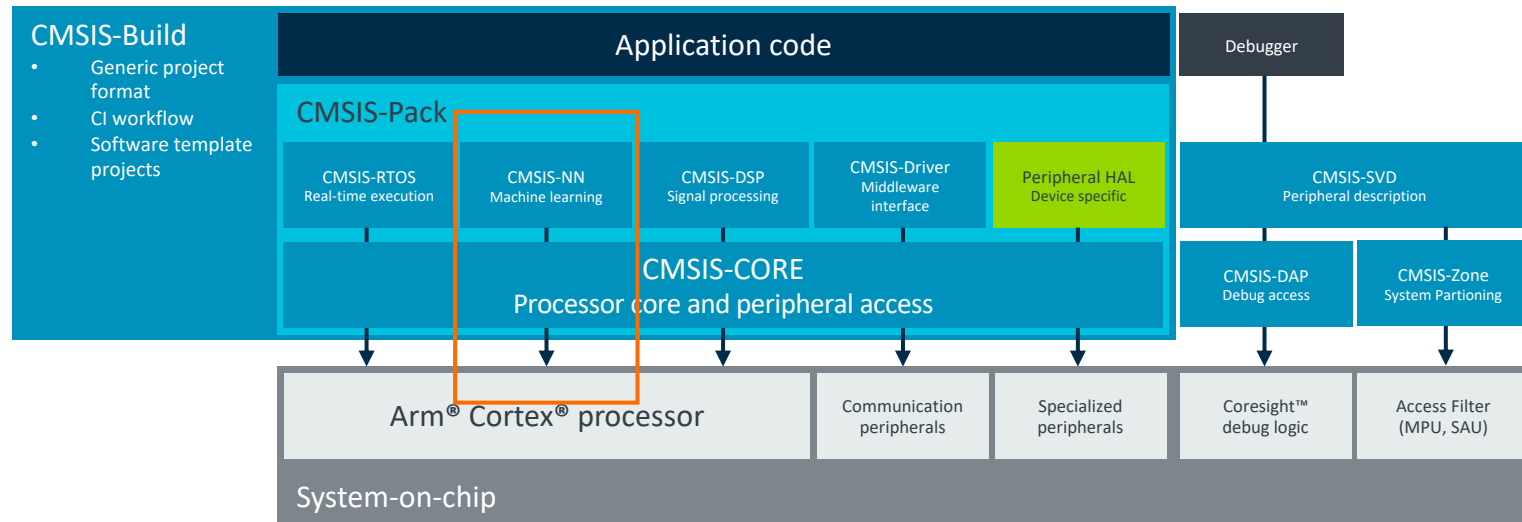
arm

ML software

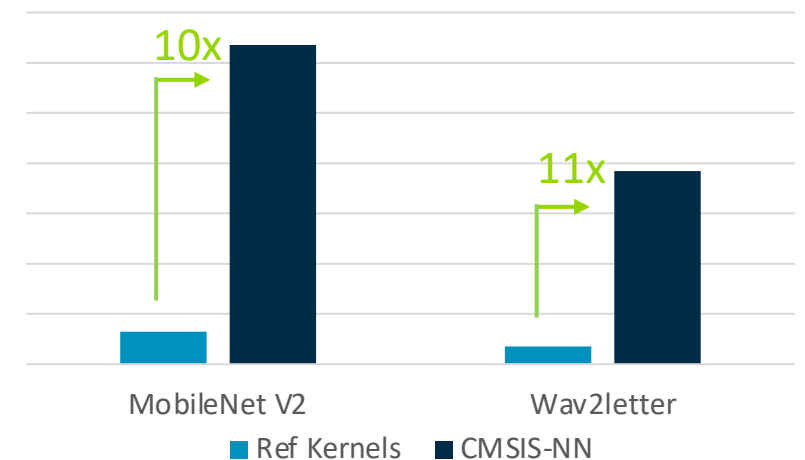
Open Source CMSIS-NN Library

Aiming for Best-in-class Performance for Cortex-M CPUs

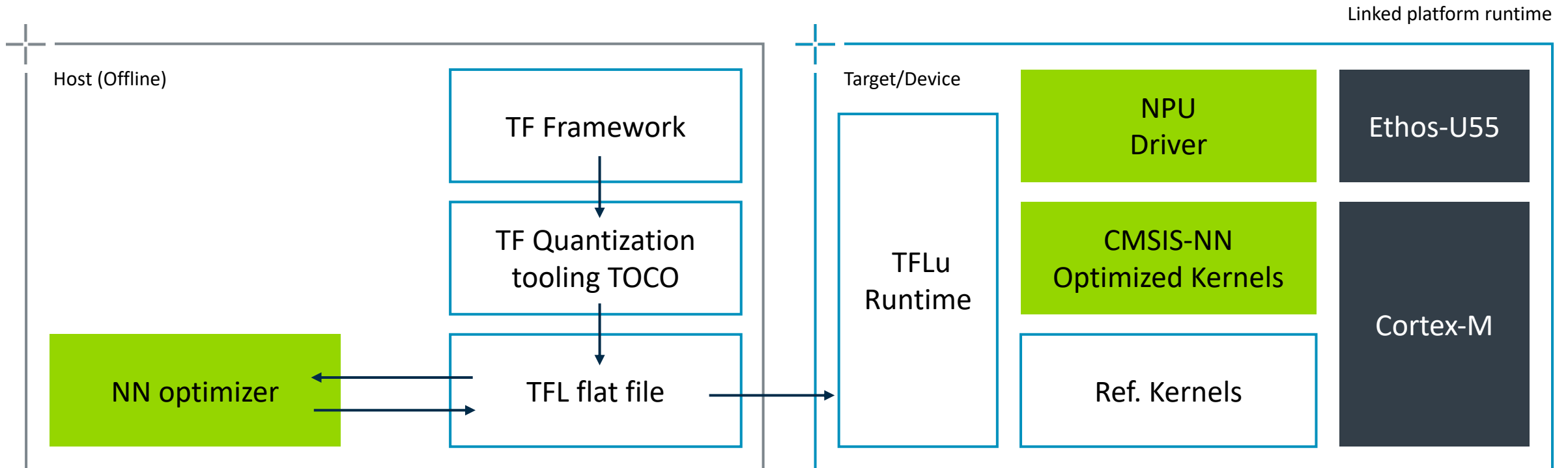
- + CMSIS-NN: Common Microcontroller Software Interface Standard – Neural Networks
- + Optimized software library for key machine learning operators
- + Consistent interface to all Cortex-M CPUs
- + Empower and enable Cortex-M processors for tinyML applications
- + Permissive Apache 2.0 license - available on [GitHub](#)



CMSIS-NN performance on Cortex-M55



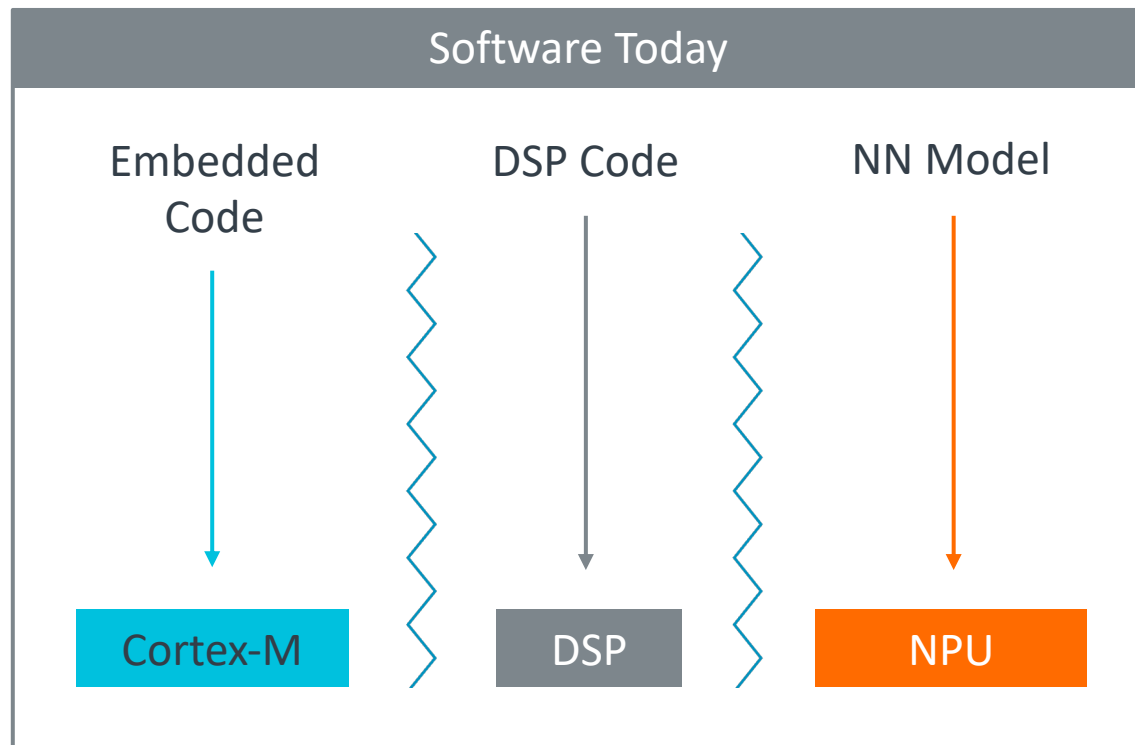
Ethos-U55 Optimized SW Flow






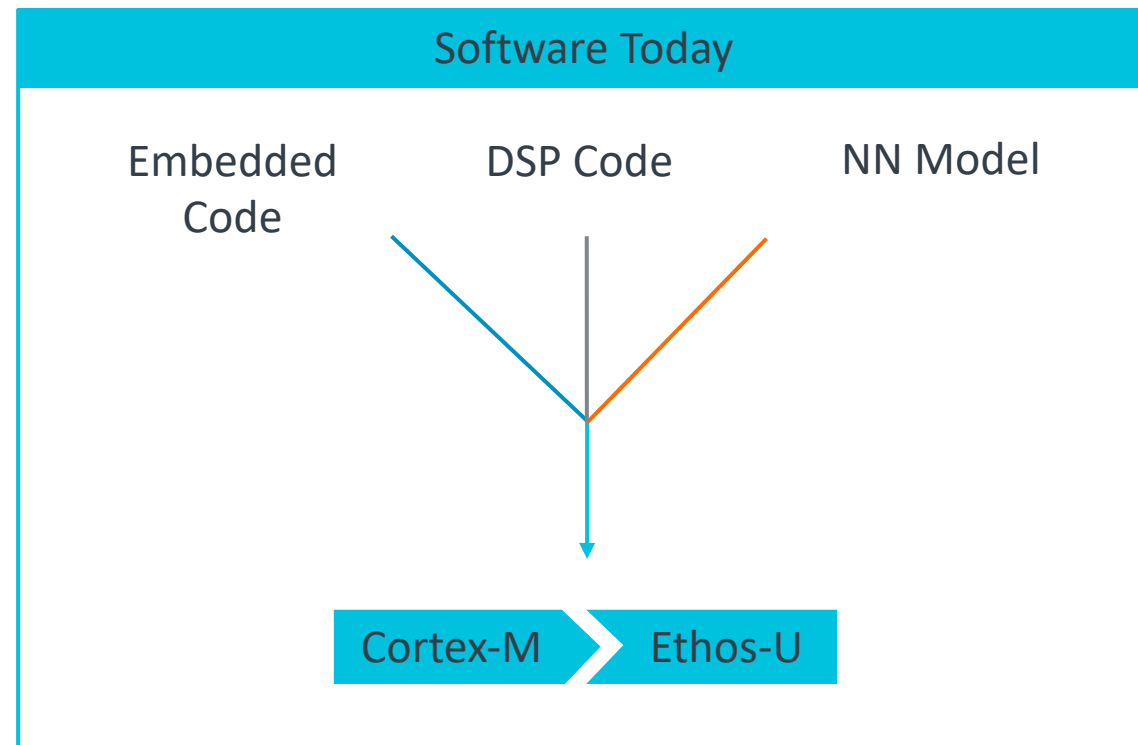
- + Train network in TF
- + Quantize it to Int8 TFL flatbuffer file (.tflite file)
- + NN Optimizer identifies graphs to run on Ethos-U55
 - Optimizes, schedules and allocates these graphs




- + Runtime executable file on device
- + Layers supported on Ethos-U55 are accelerated on it
- + The remaining layers are executed on Cortex-M
 - CMSIS-NN optimized kernels if available
 - Fallback on the TFLu reference kernels

Unified Software Development: Fastest Path to Endpoint AI



-  Multiple software development flows
-  Harder to program and debug
-  More complex, longer time-to-market



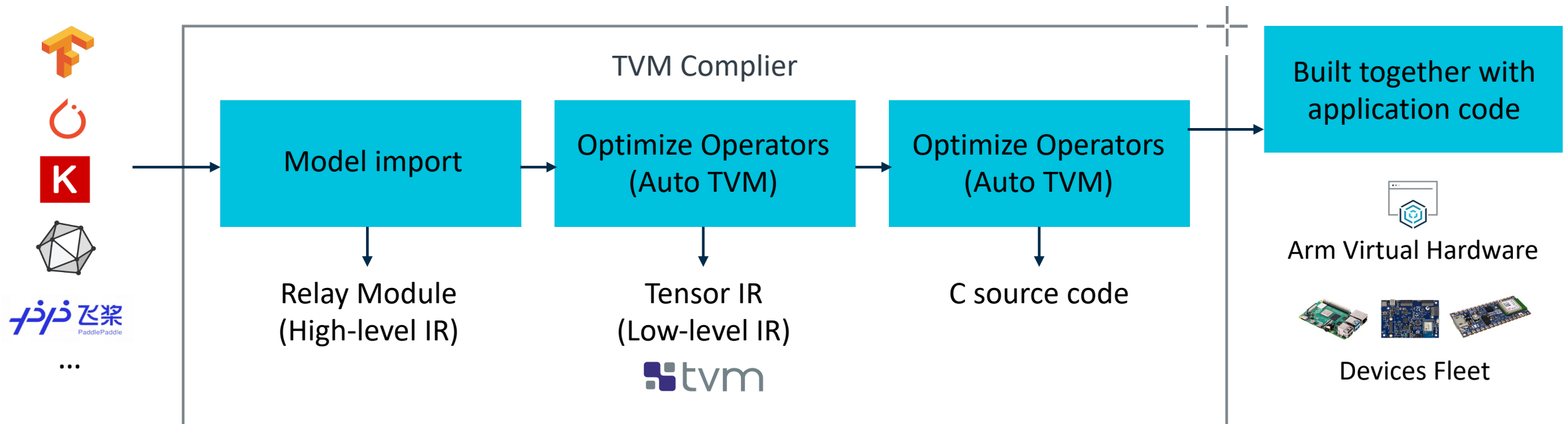
-  Unified software development flow
-  Works with common ML frameworks and existing tools
-  More productivity, faster time-to-market

arm

Addressing the ML Fragmentation Issue

Addressing ML Fragmentation Challenges

- + Fragmentation of ML frameworks leads to divergent evolution of network architectures and supported operators
- + Lack of “target platform awareness” during the training process leads to long model development loops
- + TVM Code Generation Technology for the Arm AI Platform
- + Any developer, targeting Arm hardware can potentially:
 - Take any network trained in any ML frameworks and compile it down to a binary that will run on the full range of Arm processors



Current Status of TVM Support

- + Cortex-M is supported natively by TVM with the “C” codegen
 - CMSIS-NN can also be used to further accelerate operators
 - microTVM also supports Zephyr RTOS for Cortex-M
- + Added support to over 26 popular operators for Ethos-U

Conv2D	Depthwise Conv2D	Transposed Convolution	Fully Connected	Maxpool	Average Pool
Pad	Add	Sub	Mul	Min	Clip

...and many more

There is a demonstration app to showcase Cortex-M/CMSIS-NN/Ethos-U at:
<https://github.com/apache/tvm/tree/main/apps/microtvm/ethosu>

Support for more popular networks coming soon...

Anomaly Detection
(Deep Auto Encoder)

Image Classification
(ResNet-8)

Keyword Spotting
(DS-CNN)

Visual Wake Words
(MobileNetV10.25x)

Noise Suppression
(RNNoise)

Object Detection
(SSDMobileNetV1)

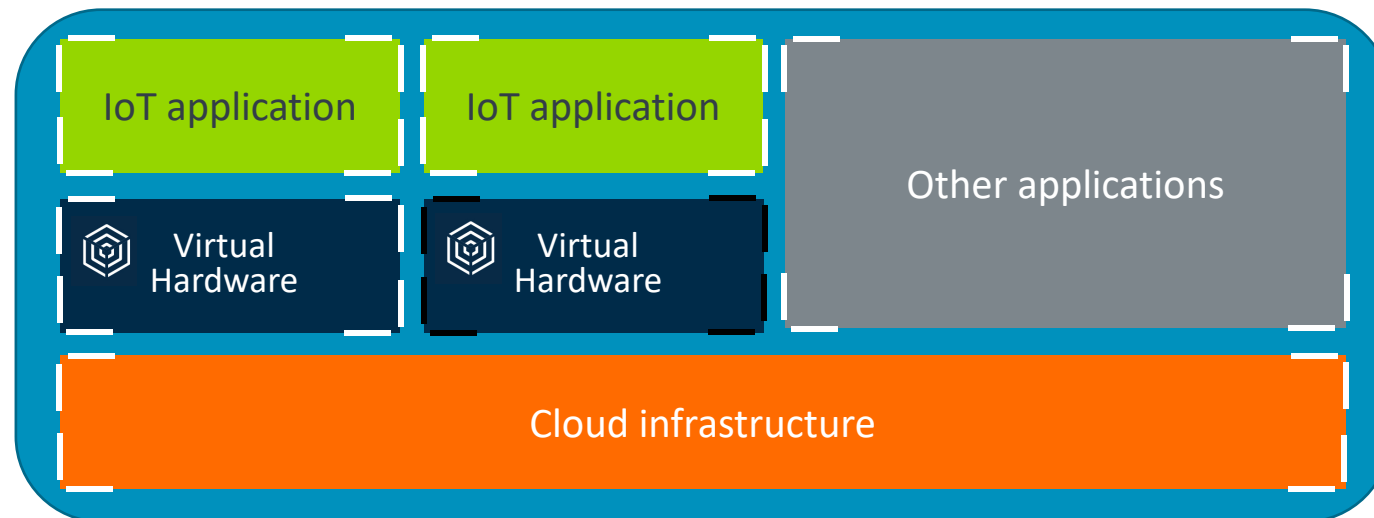
Speech Recognition
(Wav2Letter)

Face Detection
(Yolov4)

...and many more

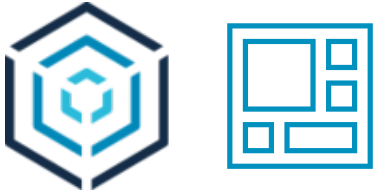
What is arm Virtual Hardware?

- + **Virtual, functional** representation of a physical hardware
- + **Cloud-native** - runs and scales easily in the cloud
- + **Suitable for all IoT workloads** from MCUs through to Intelligent Edges
- + **No dependency on RTL** or silicon availability

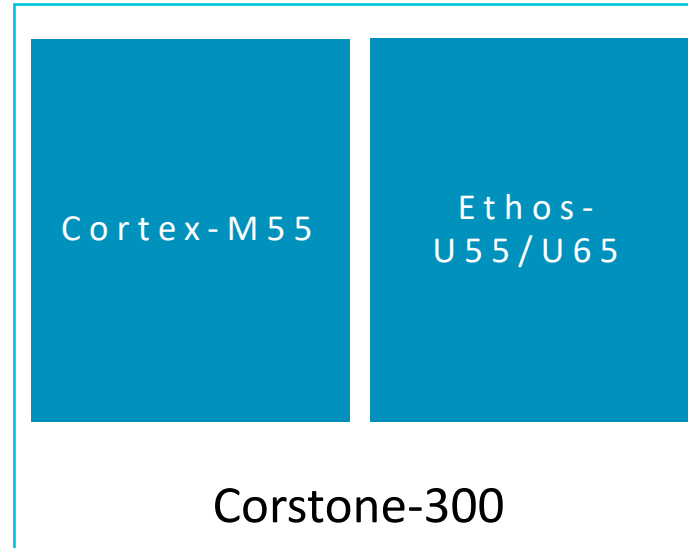


Target platform

In addition to FPGAs and Native execution



AVH for **arm** Total Solutions for IoT



arm

TOSA

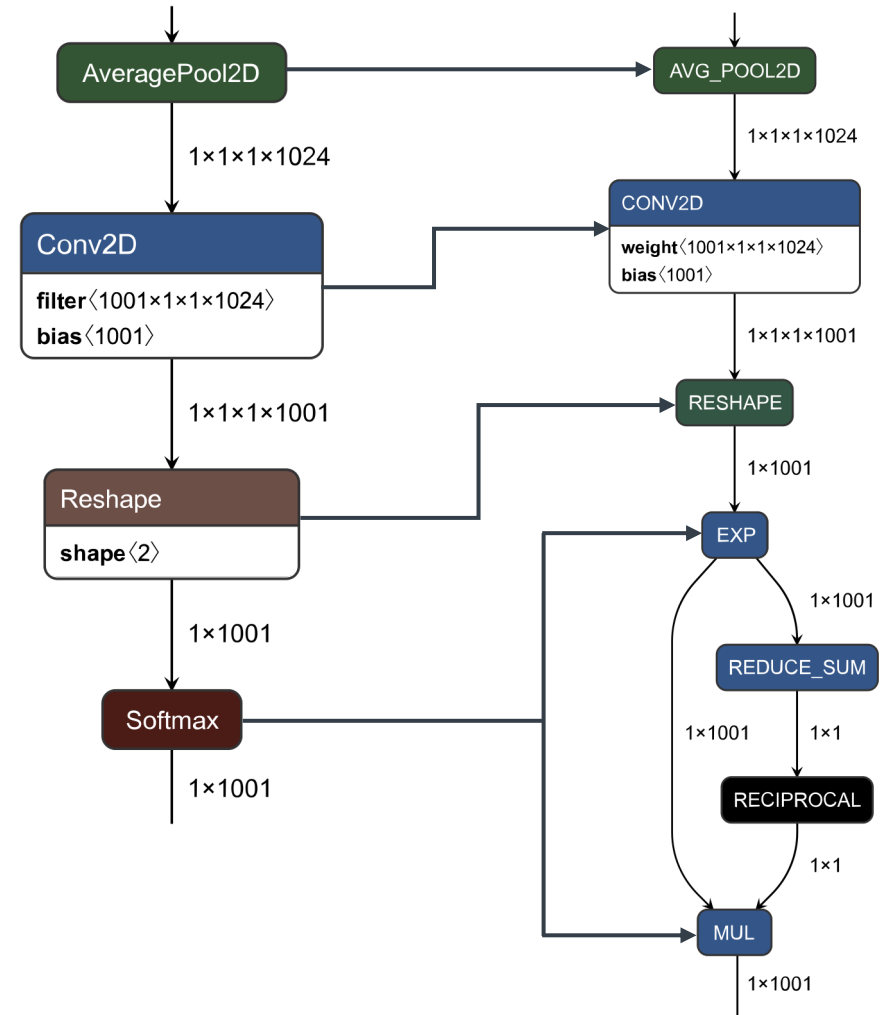
TOSA – What is it?

Tensor Operator Set Architecture (TOSA)

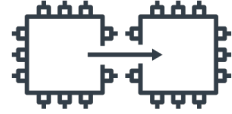
- + A minimal and stable set of tensor-level operators to which most machine learning framework operators can be reduced
- + Agnostic to any single high-level framework, compiler backend stack or particular target
- + TOSA specification contains detailed functional and numerical descriptions which enables precise code construction for a diverse range of hardware – CPU, GPU & NPU

Neural Network Operators

Tensor Level Operators

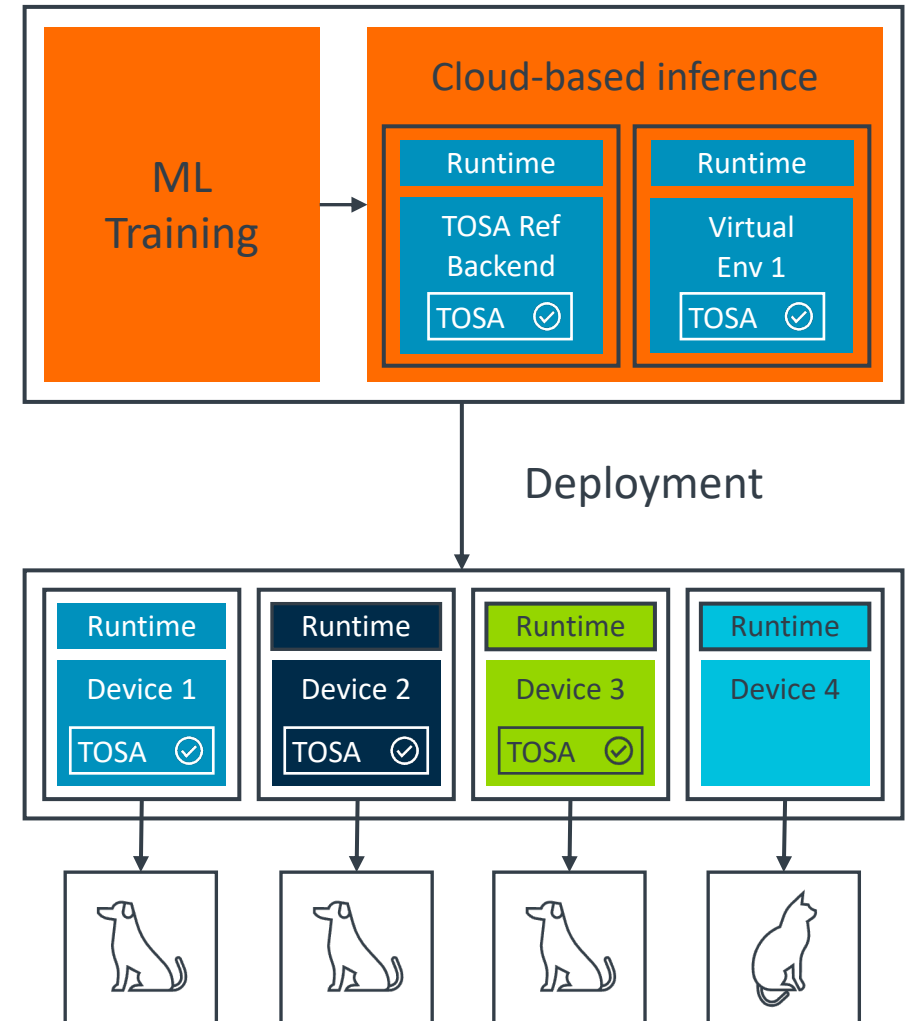


Benefits of TOSA



Portability

- + TOSA compliant Neural Networks can run on any TOSA compliant HW whilst guaranteeing numerically consistent behaviour
- + The specification defines precision for each operator
- + Enables a dramatic reduction in development, test and certification time when deploying to multiple devices



arm

Thank You

Danke

Gracias

Grazie

谢谢

ありがとう

Asante

Merci

감사합니다

धन्यवाद

Kiitos

شكرًا

ধন্যবাদ

תודה



The Arm trademarks featured in this presentation are registered trademarks or trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere. All rights reserved. All other marks featured may be trademarks of their respective owners.

www.arm.com/company/policies/trademarks

Copyright Notice

This presentation in this publication was presented at the tinyML[®] Summit (March 28 - 29, 2023). The content reflects the opinion of the author(s) and their respective companies. The inclusion of presentations in this publication does not constitute an endorsement by tinyML Foundation or the sponsors.

There is no copyright protection claimed by this publication. However, each presentation is the work of the authors and their respective companies and may contain copyrighted material. As such, it is strongly encouraged that any use reflect proper acknowledgement to the appropriate source. Any questions regarding the use of any materials presented should be directed to the author(s) or their companies.

tinyML is a registered trademark of the tinyML Foundation.

www.tinyml.org