EIQ FOR I.MX8

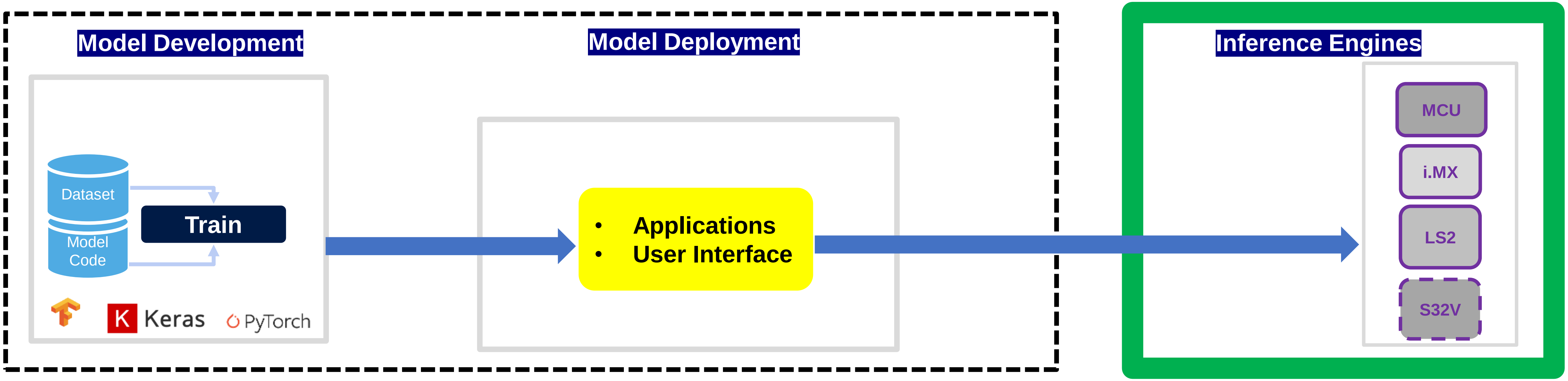
PUBLIC

Raluca Popa
i.MX Systems Engineer

JULY 2020

SECURE CONNECTIONS
FOR A SMARTER WORLD

PUBLIC



Model Development

Dataset

Model Code

Train

Keras

PyTorch

Model Deployment

- Applications
- User Interface

Inference Engines

MCU

i.MX

LS2

S32V

eIQ (2H20-2021)

- Optimize cloud vendor ecosystem (incl. model optimization)
- Model transfer learning and optimizations
- Model zoos - standard network models for ease of training/deployment

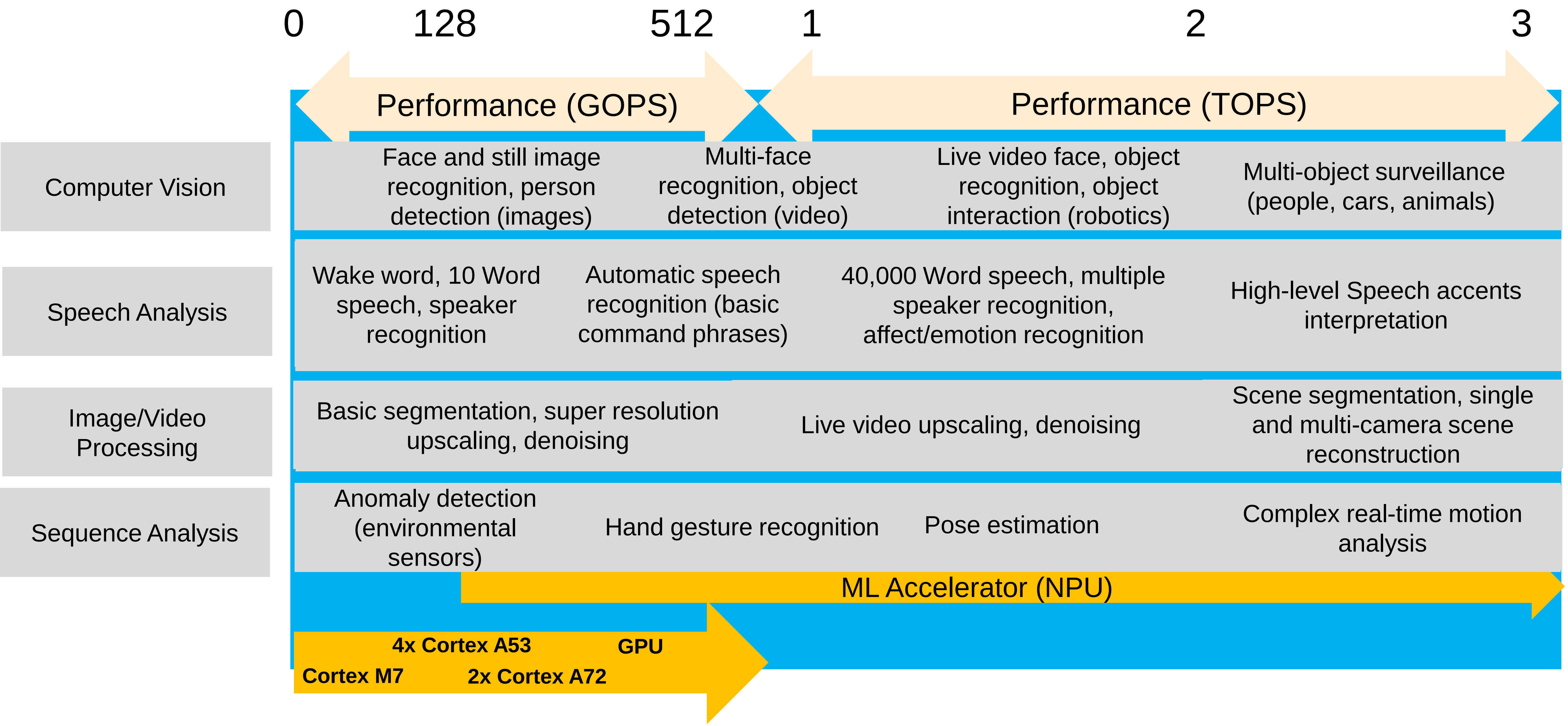
eIQ (2H20-2021)

- PyeIQ: Pyarmnn, Pytflife, Pyonnx (now)
- Sample applications for rapid customer evaluation

eIQ (2020)

- Enhanced Open source inference engines
- MPU: TensorFlow Lite, Arm NN, ONNX runtime, OpenCV)
- MCU: TensorFlow Lite, Glow
- Optimized targeting of CPU, GPU, NPU, DSP

MACHINE LEARNING USE CASES



Computer Vision

- Face and still image recognition, person detection (images)
- Multi-face recognition, object detection (video)
- Live video face, object recognition, object interaction (robotics)
- Multi-object surveillance (people, cars, animals)

Speech Analysis

- Wake word, 10 Word speech, speaker recognition
- Automatic speech recognition (basic command phrases)
- 40,000 Word speech, multiple speaker recognition, affect/emotion recognition
- High-level Speech accents interpretation

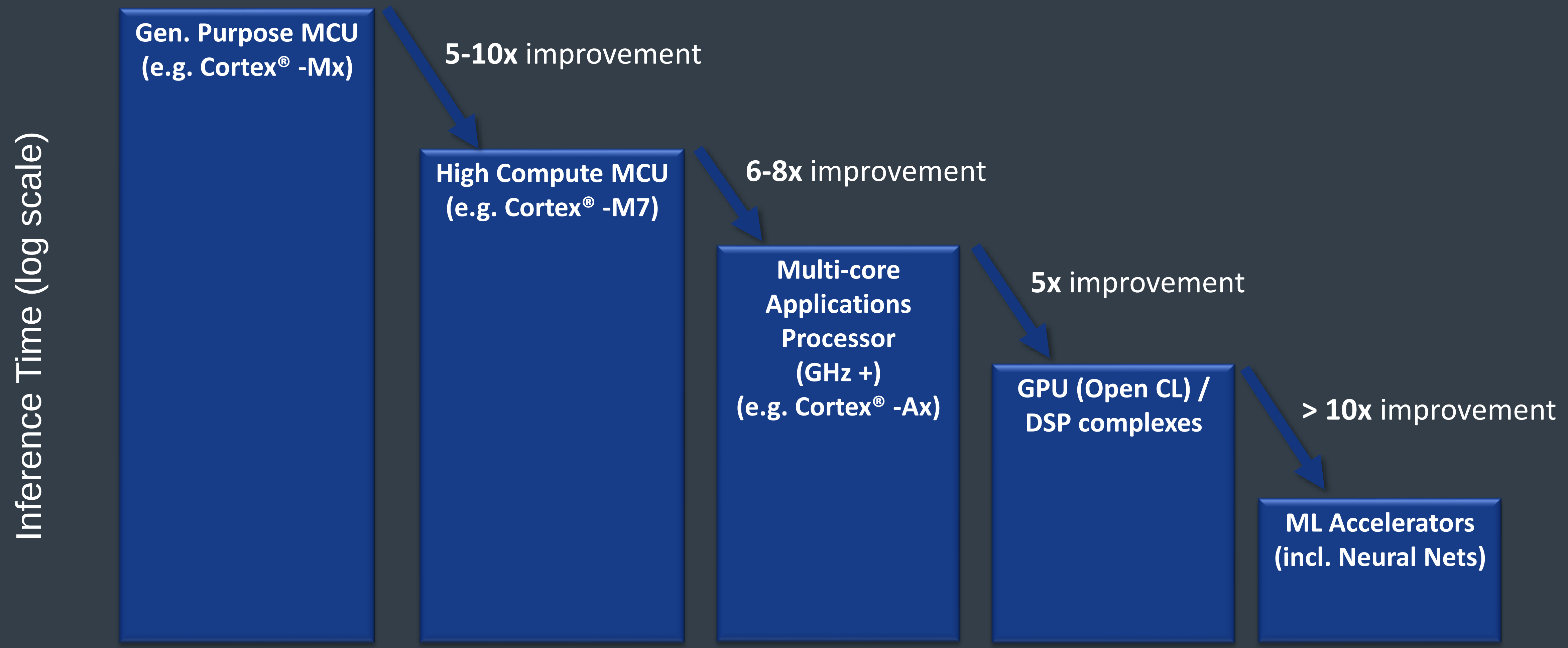
Image/Video Processing

- Basic segmentation, super resolution upscaling, denoising
- Live video upscaling, denoising
- Scene segmentation, single and multi-camera scene reconstruction

Sequence Analysis

- Anomaly detection (environmental sensors)
- Hand gesture recognition
- Pose estimation
- Complex real-time motion analysis

EDGE COMPUTE ENABLER – SCALABLE INFERENCE



Inference Time (log scale)

Gen. Purpose MCU
(e.g. Cortex® -Mx)

5-10x improvement

High Compute MCU
(e.g. Cortex® -M7)

6-8x improvement

Multi-core Applications Processor (GHz +)
(e.g. Cortex® -Ax)

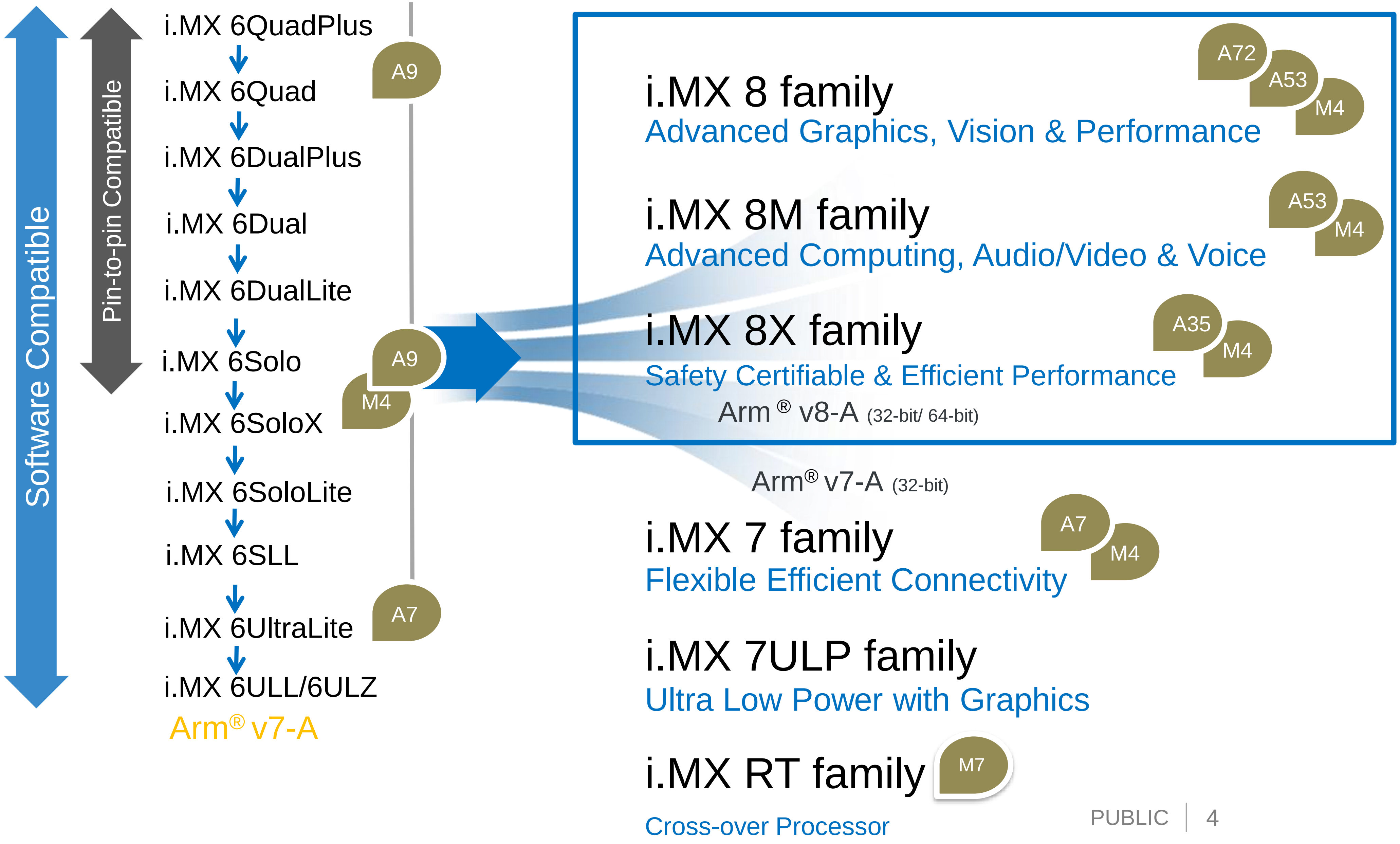
5x improvement

GPU (Open CL) / DSP complexes

> 10x improvement

ML Accelerators
(incl. Neural Nets)

i.MX Applications Processor Scalability



Software Compatible

Pin-to-pin Compatible

i.MX 6QuadPlus

i.MX 6Quad

i.MX 6DualPlus

i.MX 6Dual

i.MX 6DualLite

i.MX 6Solo

i.MX 6SoloX

i.MX 6SoloLite

i.MX 6SLL

i.MX 6UltraLite

i.MX 6ULL/6ULZ

Arm® v7-A

i.MX 8 family

Advanced Graphics, Vision & Performance

i.MX 8M family

Advanced Computing, Audio/Video & Voice

i.MX 8X family

Safety Certifiable & Efficient Performance

Arm ® v8-A (32-bit/ 64-bit)

Arm® v7-A (32-bit)

i.MX 7 family

Flexible Efficient Connectivity

i.MX 7ULP family

Ultra Low Power with Graphics

i.MX RT family

Cross-over Processor

i.MX 8 Series: Target Applications

Advanced graphics, video, image processing, vision, audio and voice

i.MX 8M Family

A53 M4 GPU

Advanced Computing, Audio/Video & Voice

i.MX 8X Family

A35 M4 GPU DSP

Safety Certifiable & Efficient Performance

i.MX 8 Family

A72 A53 M4 GPU DSP

Advanced Graphics, Vision & Performance

I.MX 8M PLUS MACHINE LEARNING COMPUTE ENGINES

Machine Learning Accelerator (1GHz)

- Primary Use: Multi-camera classification/detection

Quad Arm® Cortex-A53 (1.8GHz)

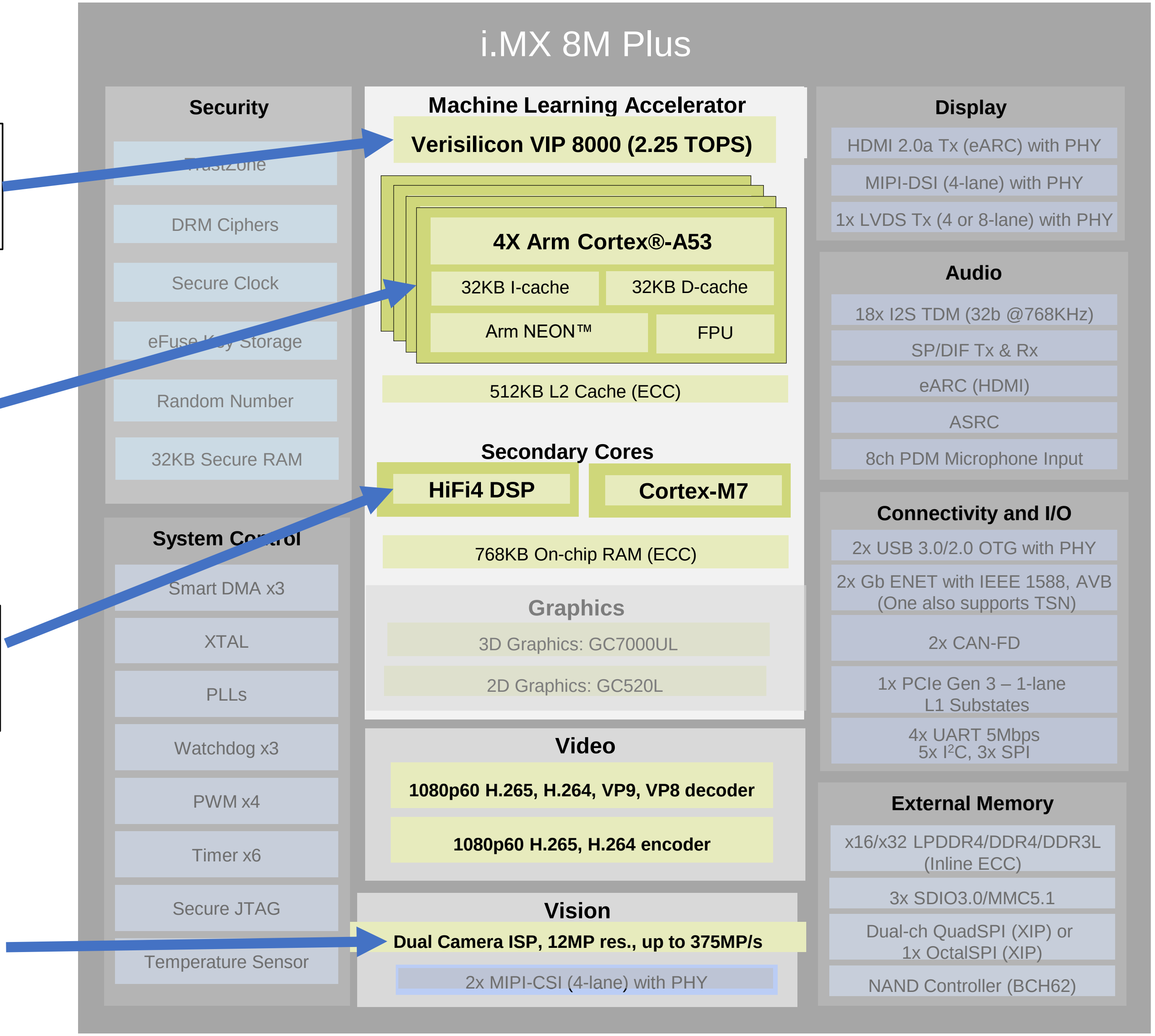
- Primary Use: Speech command recognition, object detect/classification

Cortex-M7+HiFi4 DSP (800MHz)

- Primary Use: Keyword detection, sensor fusion

Bonus: 2 channel Image Signal Processor (ISP)

- Primary Use: Scaling, dewarping, image enhancement



i.MX 8M Plus

Security

TrustZone

DRM Ciphers

Secure Clock

eFuse Key Storage

Random Number

32KB Secure RAM

System Control

Smart DMA x3

XTAL

PLLs

Watchdog x3

PWM x4

Timer x6

Secure JTAG

Temperature Sensor

Machine Learning Accelerator

Verisilicon VIP 8000 (2.25 TOPS)

4X Arm Cortex®-A53

32KB I-cache

32KB D-cache

Arm NEON™

FPU

512KB L2 Cache (ECC)

Secondary Cores

HiFi4 DSP

Cortex-M7

768KB On-chip RAM (ECC)

Graphics

3D Graphics: GC7000UL

2D Graphics: GC520L

Video

1080p60 H.265, H.264, VP9, VP8 decoder

1080p60 H.265, H.264 encoder

Vision

Dual Camera ISP, 12MP res., up to 375MP/s

2x MIPI-CSI (4-lane) with PHY

Display

HDMI 2.0a Tx (eARC) with PHY

MIPI-DSI (4-lane) with PHY

1x LVDS Tx (4 or 8-lane) with PHY

Audio

18x I2S TDM (32b @768KHz)

SP/DIF Tx & Rx

eARC (HDMI)

ASRC

8ch PDM Microphone Input

Connectivity and I/O

2x USB 3.0/2.0 OTG with PHY

2x Gb ENET with IEEE 1588, AVB (One also supports TSN)

2x CAN-FD

1x PCIe Gen 3 – 1-lane L1 Substates

4x UART 5Mbps 5x I^{2}C, 3x SPI

External Memory

x16/x32 LPDDR4/DDR4/DDR3L (Inline ECC)

3x SDIO3.0/MMC5.1

Dual-ch QuadSPI (XIP) or 1x OctalSPI (XIP)

NAND Controller (BCH62)

I.MX 8M PLUS NPU

Programmable Engine Unit
(1 instance on i.MX8M Plus)

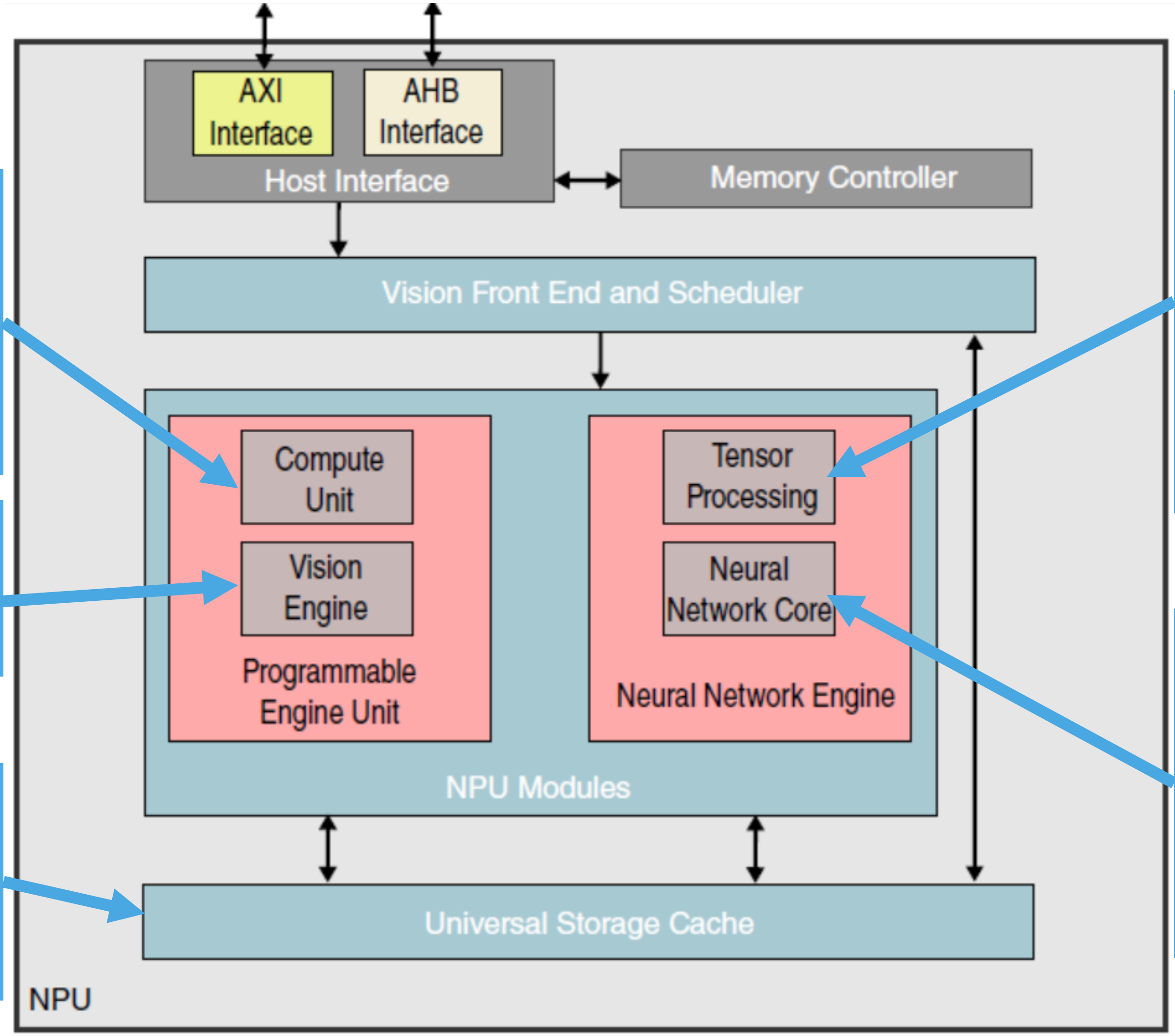
128-bit vector processing.
INT 8/16/32b, FLOAT 16/32b.
Most flexible programming unit to handle everything else

Vision Engine

Provides advanced image processing functions.

Universal Storage Cache
(1 instance on i.MX8M Plus)

Local memory and L1 cache to pass data amongst NPU modules



Tensor Processing Core
(3 instances on i.MX8M Plus)
INT 8/16b, FLOAT 16b

Non-convolution layers.
Multi-lane processing for data shuffling, normalization, pooling/unpooling, LUT, etc.
Network pruning support, zero skipping, compression

Neural Network Core
(6 instances on i.MX8M Plus)
2.3 TOPs INT8

Handle Convolution Layer + RELU + Max Pooling, and Compute Bounded Fully Connected Layers

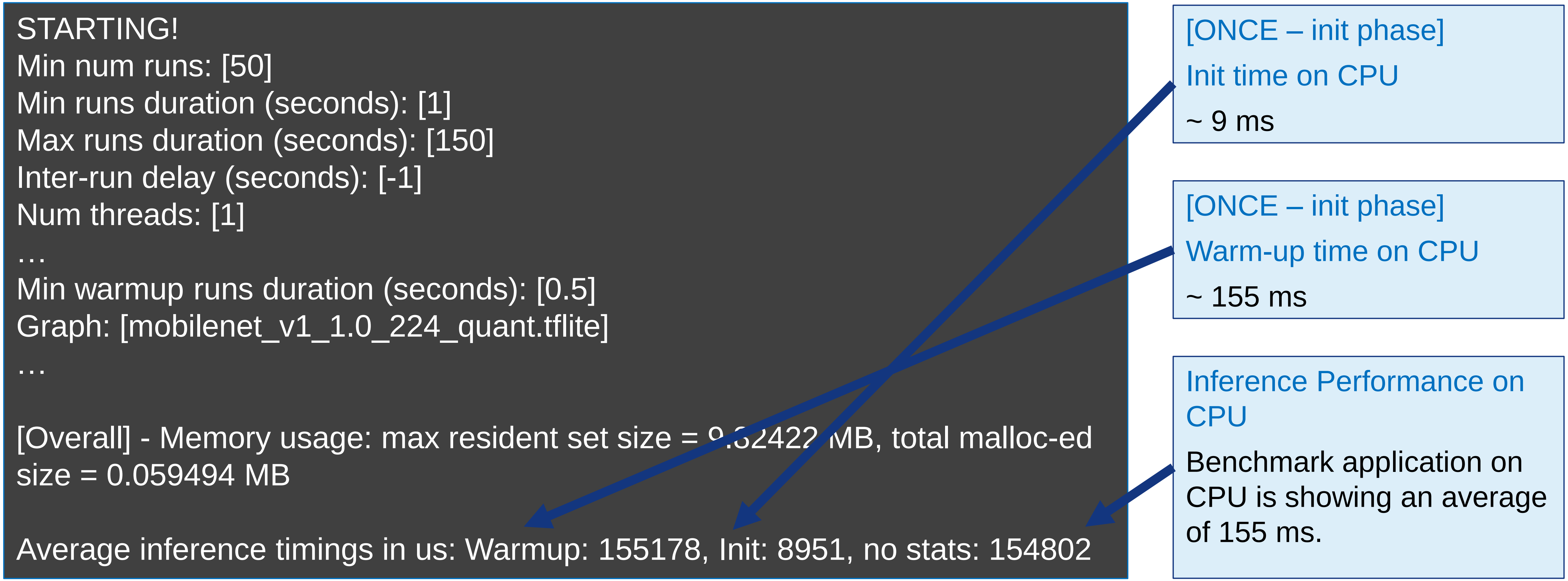
Comparison Between CPU and NPU Performance

SECURE CONNECTIONS
FOR A SMARTER WORLD

PUBLIC

INFERENCE EXAMPLE WITH TFLITE USING CPU

```
$: ./benchmark_model --graph=mobilenet_v1_1.0_224_quant.tflite --max_num_runs=10
```



```
STARTING!
Min num runs: [50]
Min runs duration (seconds): [1]
Max runs duration (seconds): [150]
Inter-run delay (seconds): [-1]
Num threads: [1]
…
Min warmup runs duration (seconds): [0.5]
Graph: [mobilenet_v1_1.0_224_quant.tflite]
…

[Overall] - Memory usage: max resident set size = 9.82422 MB, total malloc-ed
size = 0.059494 MB

Average inference timings in us: Warmup: 155178, Init: 8951, no stats: 154802
```

[ONCE – init phase]

Init time on CPU

~ 9 ms

[ONCE – init phase]

Warm-up time on CPU

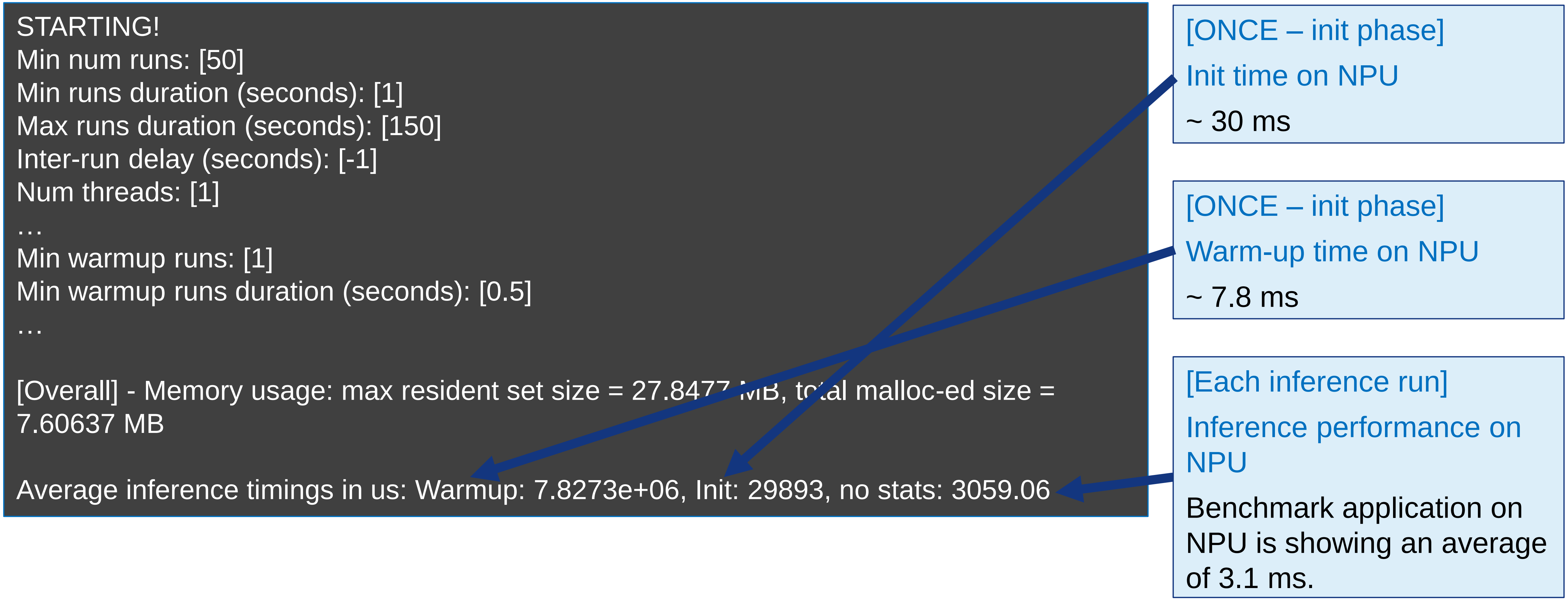
~ 155 ms

Inference Performance on CPU

Benchmark application on CPU is showing an average of 155 ms.

INFERENCE EXAMPLE WITH TFLITE USING NPU

```
$: ./benchmark_model --graph=mobilenet_v1_1.0_224_quant.tflite --max_num_runs=10 --use_nnapi=true
```



```
STARTING!
Min num runs: [50]
Min runs duration (seconds): [1]
Max runs duration (seconds): [150]
Inter-run delay (seconds): [-1]
Num threads: [1]
…
Min warmup runs: [1]
Min warmup runs duration (seconds): [0.5]
…

[Overall] - Memory usage: max resident set size = 27.8477 MB, total malloc-ed size =
7.60637 MB

Average inference timings in us: Warmup: 7.8273e+06, Init: 29893, no stats: 3059.06
```

[ONCE – init phase]

Init time on NPU

~ 30 ms

[ONCE – init phase]

Warm-up time on NPU

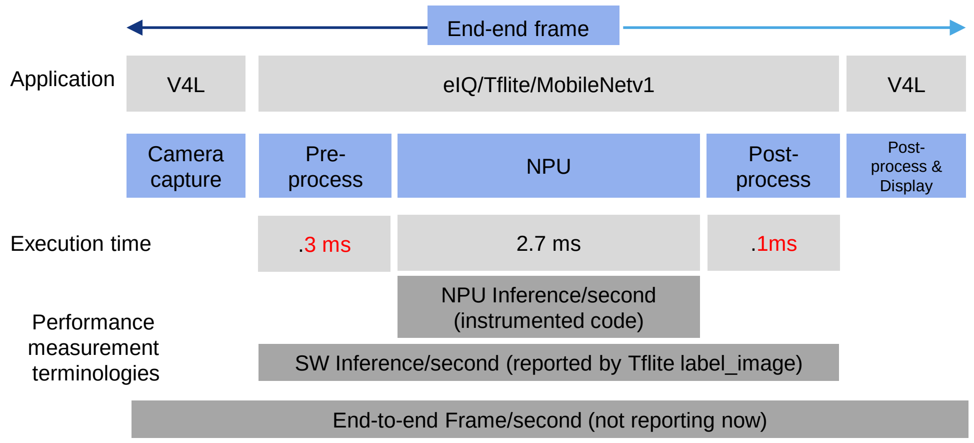
~ 7.8 ms

[Each inference run]

Inference performance on NPU

Benchmark application on NPU is showing an average of 3.1 ms.

NPU PERFORMANCE CLARIFICATION



End-end frame

Application | V4L | eIQ/Tflite/MobileNetv1 | V4L

Camera capture | Pre-process | NPU | Post-process | Post-process & Display

Execution time | .3 ms | 2.7 ms | .1ms

Performance measurement terminologies

NPU Inference/second (instrumented code)

SW Inference/second (reported by Tflite label_image)

End-to-end Frame/second (not reporting now)

TFLite SW IPS = 3.1 ms

eIQ performance is measured as:

1. NPU Inferences per second (Hardware only)
   - Purely NPU execution time
2. eIQ SW Inference per second (Includes SW stack overhead)
   - ML Stack execution time
3. Samples End-to-end FPS (Camera capture to display)
   - A measurement of SoC System performance

NPU PROFILING

Uboot config

Update mmcargs by adding **galcore.showArgs=1 galcore.gpuProfiler=1**

```
u-boot=> editenv mmc
edit: setenv bootargs ${jh_clk} console=${console} root=${mmcroot} galcore.showArgs=1 galcore.gpuProfiler=1
u-boot=> boot
```

Yocto environment variables

```
export VSI_NN_LOG_LEVEL=0
export CNN_PERF=1
export NN_EXT_SHOW_PERF=1
export VIV_VX_DEBUG_LEVEL=1
export VIV_VX_PROFILE=1
```

Output

- Execution time
- Operators list and the NPU compute blocks where they were executed
- DDR bandwidth

eIQ Performance Overview

SECURE CONNECTIONS
FOR A SMARTER WORLD

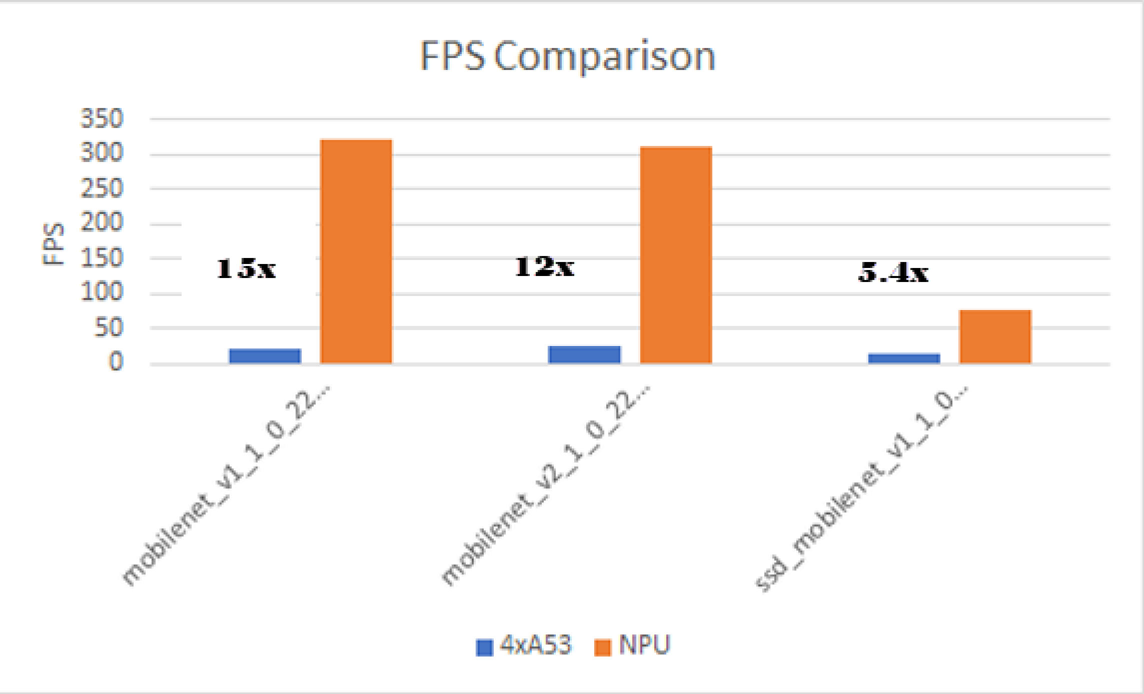
PUBLIC

eIQ Target Device Details

	Cortex-M	DSP	Cortex-A	GPU	NPU
i.MX 8M Plus	M7 (800 MHz)	HiFi4 (800 MHz)	4xA53 (1800 MHz)	GC7000UL (1000 MHz)	VIP9000 (1000 MHz)
i.MX 8QuadMax	2xM4F (266 MHz)	HiFi4 (x MHz)	2xA72 (1600 MHz); 4xA53 (1200 MHz)	2xGC7000X (996 MHz)	---
i.MX 8QuadXPlus	M4 (266 MHz)	HiFi4	4x-A35 (1200 MHz)	GC7000L (850 MHz)	---
i.MX 8M Quad	M4 (266 MHz)	---	4xA53 (1500 MHz); x32DDR	GC7000L (800 MHz)	---
i.MX 8M Mini	M4 (400 MHz)	---	4x-A53 (1800 MHz)x32DDR	---	---
i.MX 8M Nano	M7 (600MHz)	---	4xA53 (1500 MHz); x16DDR	GC7000UL (600 MHz)	---

I.MX 8M PLUS NPU COMPARED TO CPU PERFORMANCE

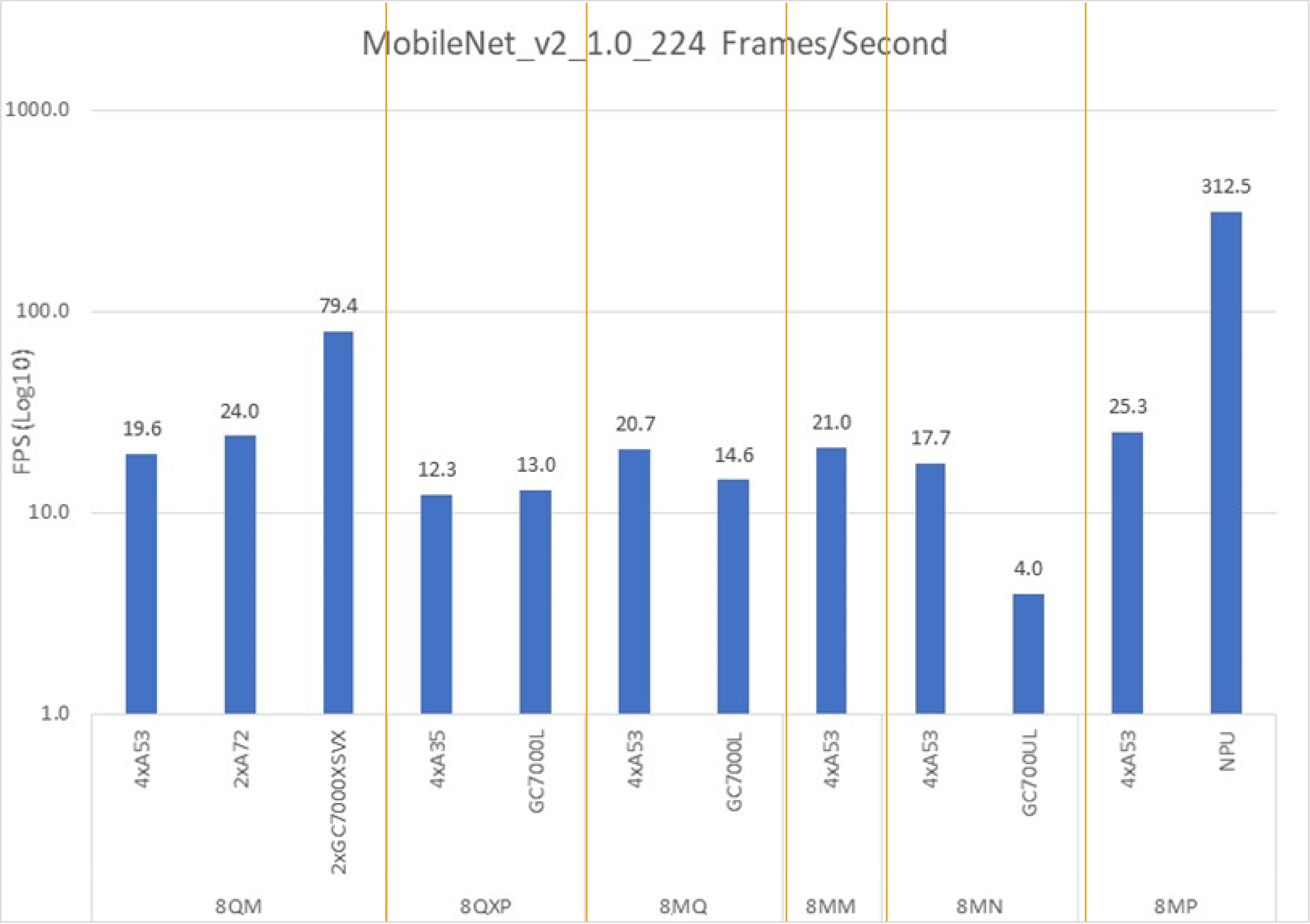
- Quantized results – NPU is 5-15x faster than CPUs



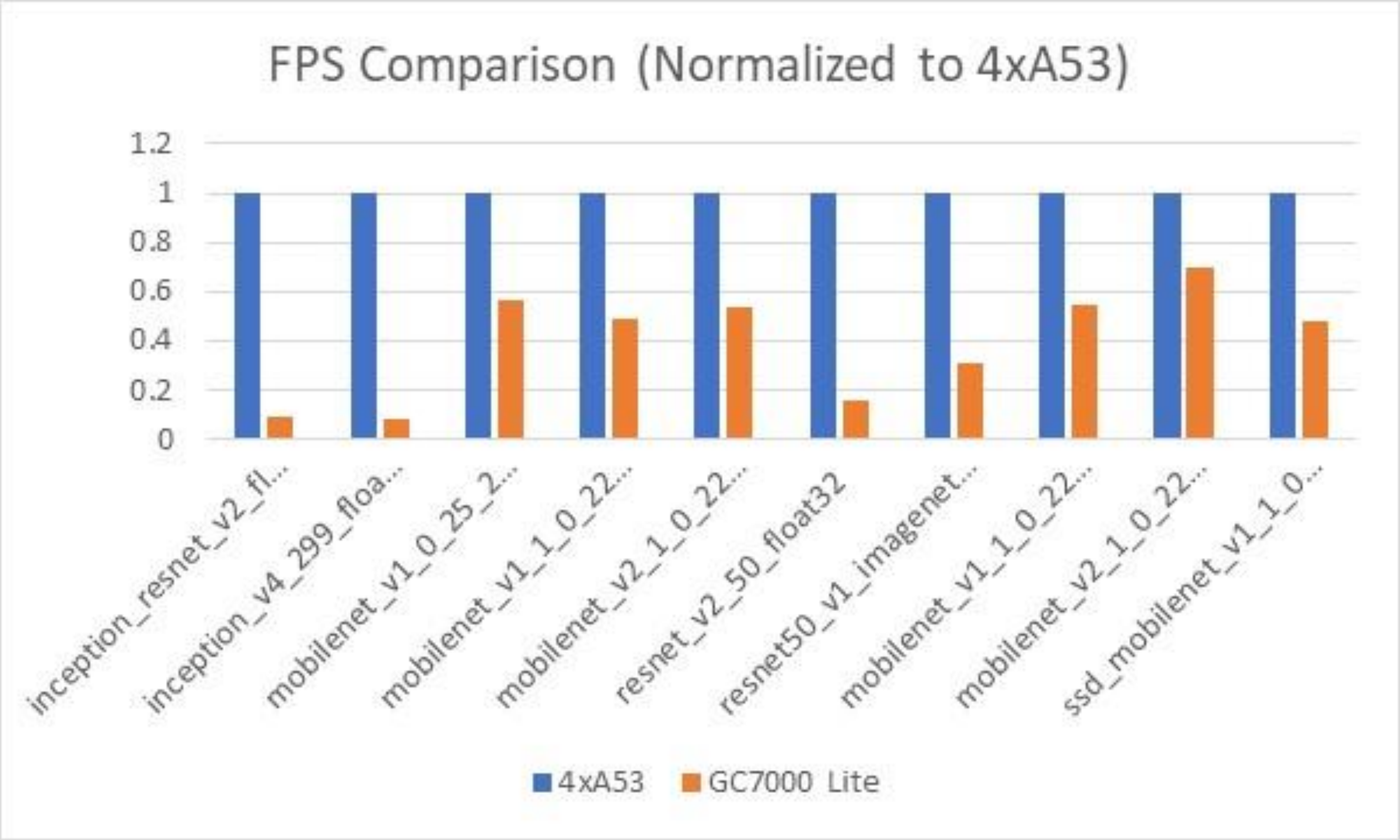
NOTES:

1. FP results (not shown) – NPU is 5-12x slower than CPUs
2. SSD has post processing overhead (not tail end of the model). After objects are detected, all the bounding box information has to be processed and identified. SSD would identify many boxes for the same object and hence post processing consumes CPU time.

MOBILENET PERFORMANCE ACROSS I.MX8 COMPUTE UNITS



I.MX 8MQ 4XA53 COMPARED TO GC7000L



- CPUs are 1.4-6.3x faster than GPU (8M Nano CPUs are 4.4-9.3x faster than GPU; graph not shown)
- Use GPU as offload engine – not performance accelerator
- TF Lite faster on quantized workloads – Arm NN faster on floating-point

eIQ Demos - pyEIQ

SECURE CONNECTIONS
FOR A SMARTER WORLD

PYEIQ OVERVIEW

PyeIQ - A Python Framework for eIQ on i.MX Processors

- Easy to install

```
$: pip3 install eiq-<version>.tar.gz
```

- Easy to run

```
root@imx8:~# cd /opt/eiq/demos
root@imx8:~/opt/eiq/demos# python3 <demo_name>.py
root@imx8:~/opt/eiq/demos# python3 <demo_name>.py --help
```

- Support demos based on TensorFlow Lite (2.1.0) for image classification and object detection.
- Support inference running on GPU/NPU and CPU.
- Currently support file and camera as input data.
- Allows easy benchmarking
- Sources available on the Code Aurora Forum
  https://source.codeaurora.org/external/imxsupport/pyeiq/

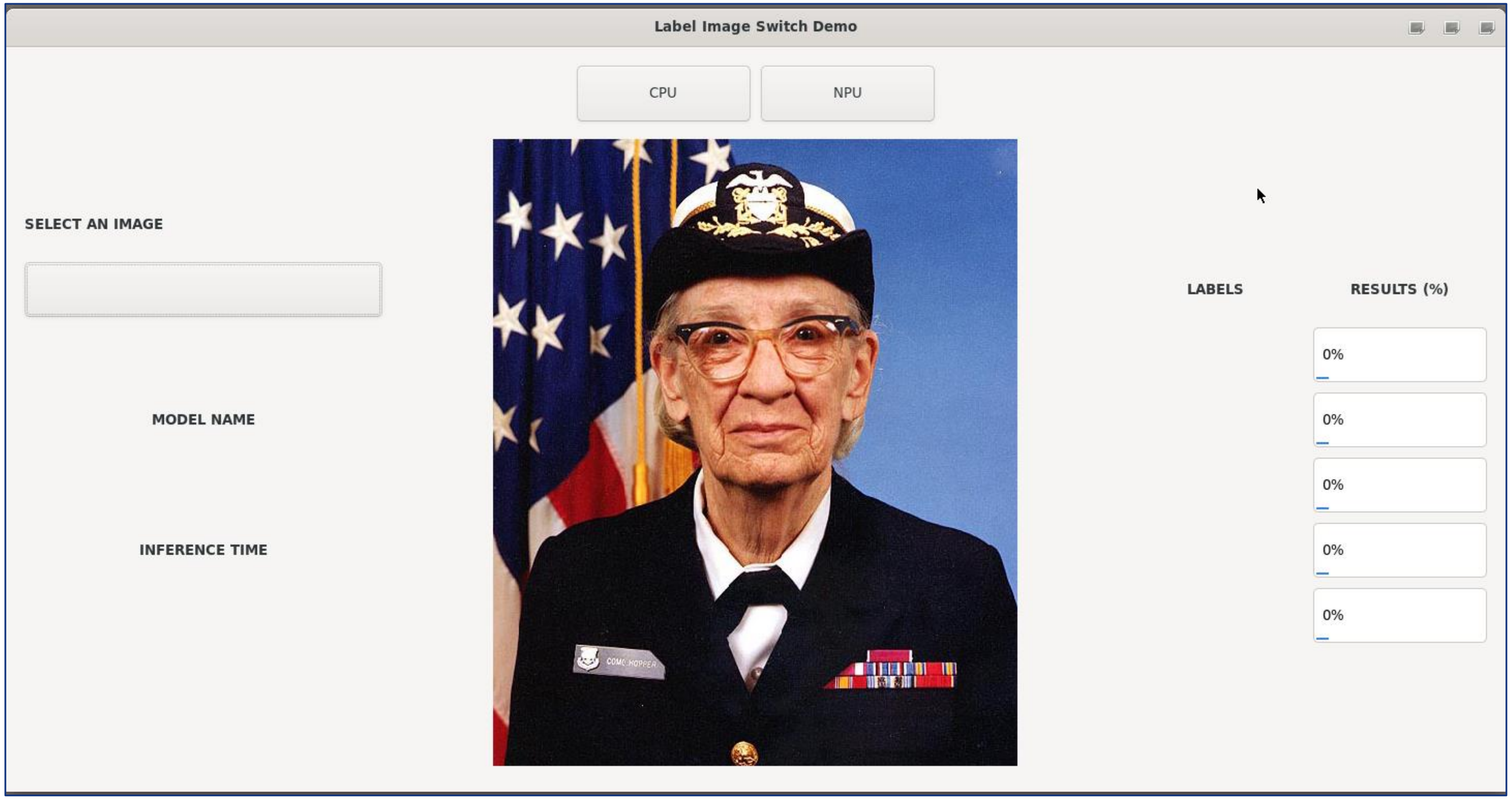
SAMPLE EXAMPLE – COMPARE PERFORMANCE BETWEEN CPU AND NPU/GPU

To run the demo app on Target.

```
# cd /opt/eiq/apps
# python3 switch-demo.py
```

Display and camera connected to the board.

- [Input] Select compute unit: CPU/NPU.
- [Input] Select the image for inference.
- [Output] Model Name.
- [Output] Inference time.
- [Output] Top 5 Accuracy.



Resources

SECURE CONNECTIONS
FOR A SMARTER WORLD

RESOURCES

- Product page i.MX 8M Plus applications processor
- 4K MIPI Camera for i.MX 8M Plus applications processor
- i.MX 8M Plus applications processor Fact Sheet
- Technology Blog: Why Add an ISP and Machine Learning to the i.MX 8M Family

- eIQ™ Machine Learning Software Development Environment
- Community: eIQ Software Community
- eIQ Security Toolkit
- Demos: pyeIQ

SECURE CONNECTIONS
FOR A SMARTER WORLD