arm

Recent Developments at the IoT Edge

Machine Learning

Platform Security Architecture

Tim Hartley, Product Manager, Machine Learning Group

1 March, 2018

History of Arm

Joint venture between Acorn Computers and Apple

1990

Designed into first mobile phones and then smartphones

1993 onwards

Now all electronic devices can use intelligent Arm technology

Today

arm

Arm: the Industry's Architecture of Choice

Extraordinary growth – from sensors to server

Machine Learning on Arm Cortex-M Microcontrollers

Why is ML Moving to the Edge?

Bandwidth

Power

Cost

Latency

Reliability

Security

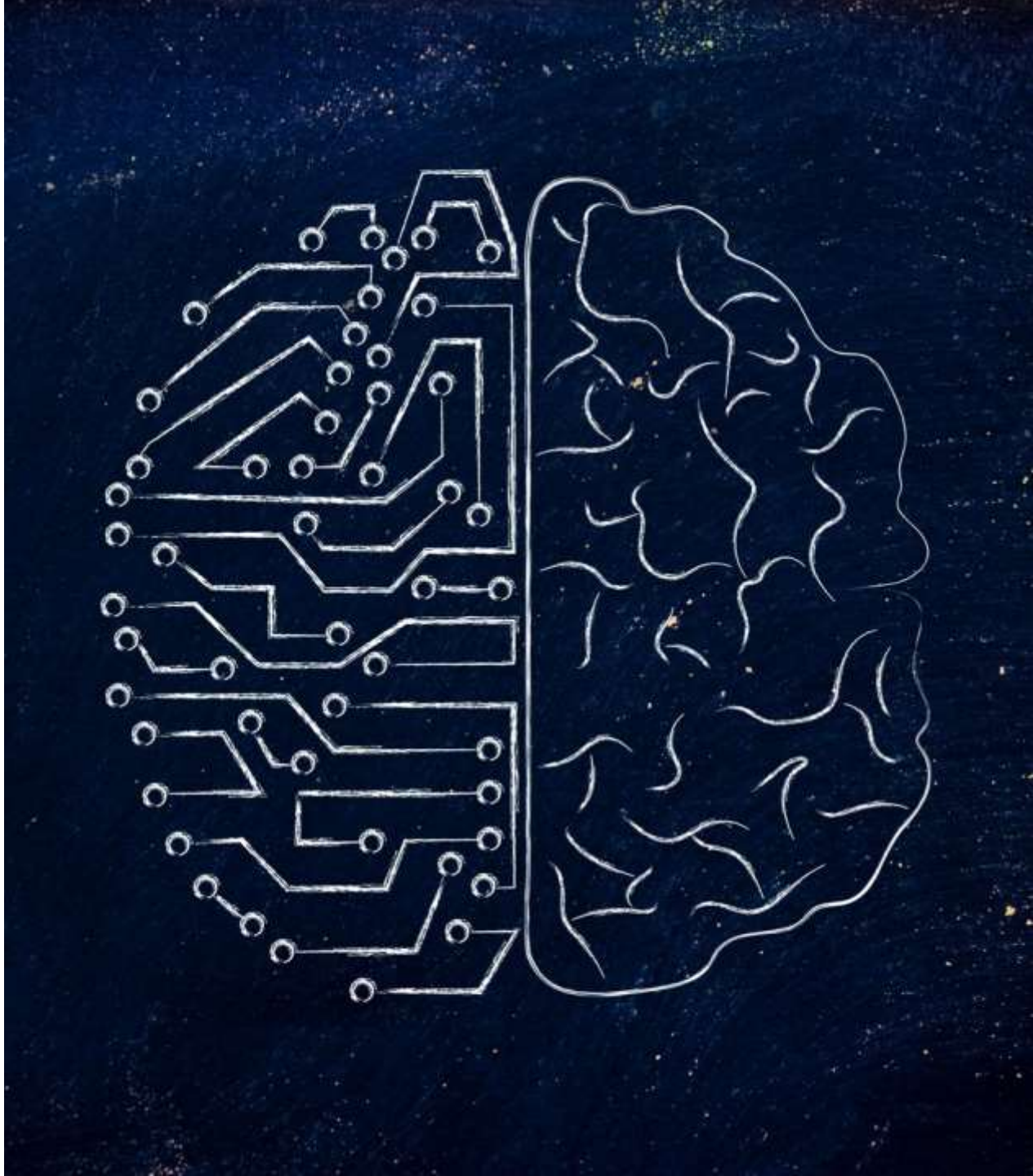
Project Trillium: Arm ML for All Devices

Arm ML suite of IP: designed for unmatched versatility and scalability:

- Machine Learning (ML) processor
- Object Detection (OD) processor
- Neural Network (NN) software libraries

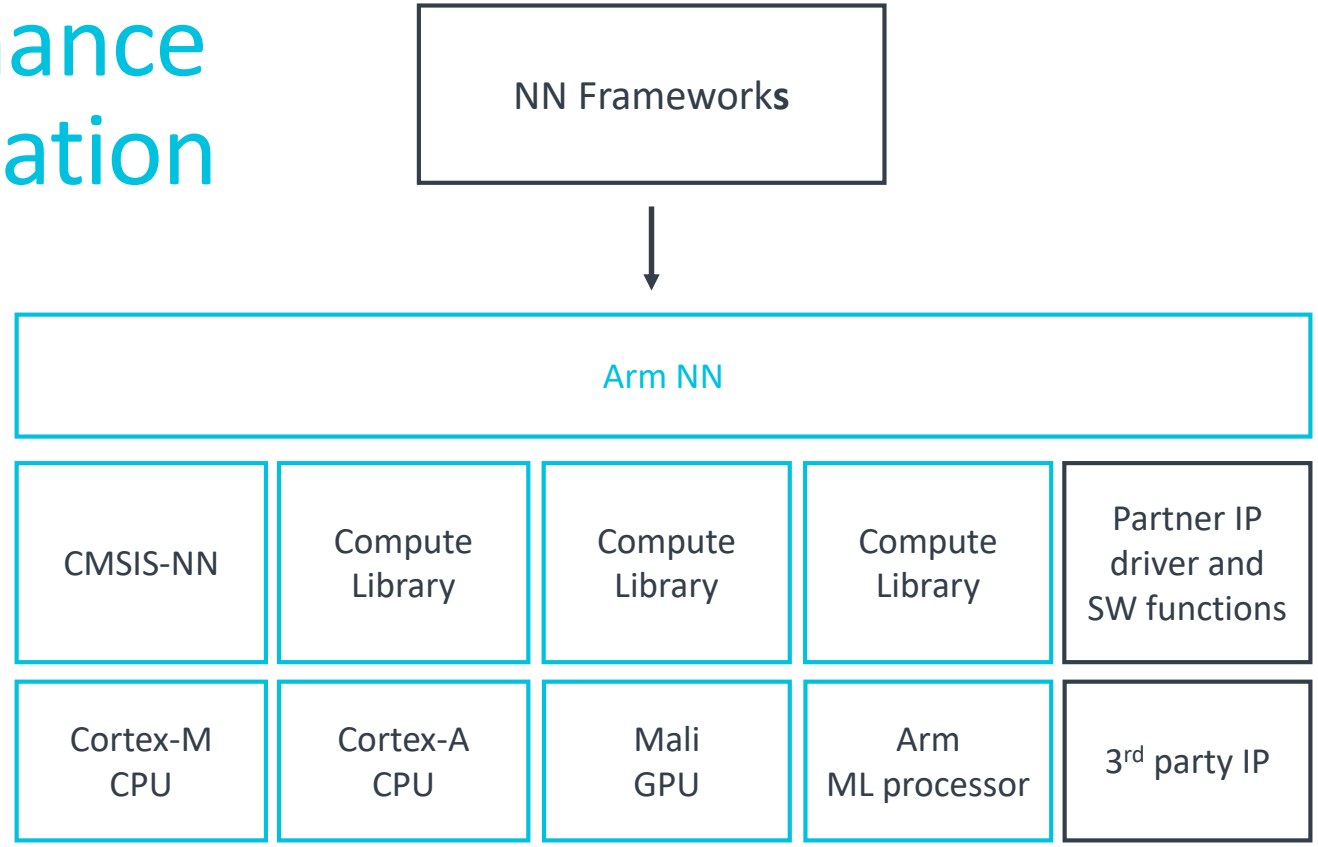
Market growth in units (today to 2028):

- Mobile - 1.7Bn to 2.2Bn
  (source: Strategy Analytics and Arm forecast)
- Smart IP Cameras - 160M to 1.3Bn
  (source: Gartner and Arm forecast)
- AI-enabled devices - 300M to 3.2Bn
  (source: IDC WW Embedded and Intelligent Systems Forecast, 2017-2022 and Arm forecast)



Optimum ML Performance on Arm for Any Application

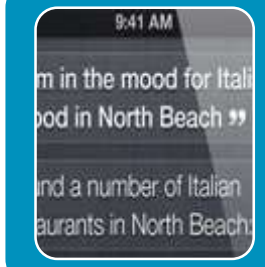
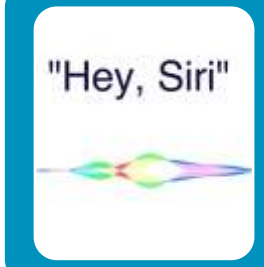
- Arm NN software translates existing NN frameworks:
  - TensorFlow, Caffe, Android NNAPI, MXNet etc.
  - Developers maintain existing workflow and tools
  - Reduces overall development time
  - Abstracts away the complexities of underlying hardware



CMSIS-NN 5x
better efficiency and performance for NN functions

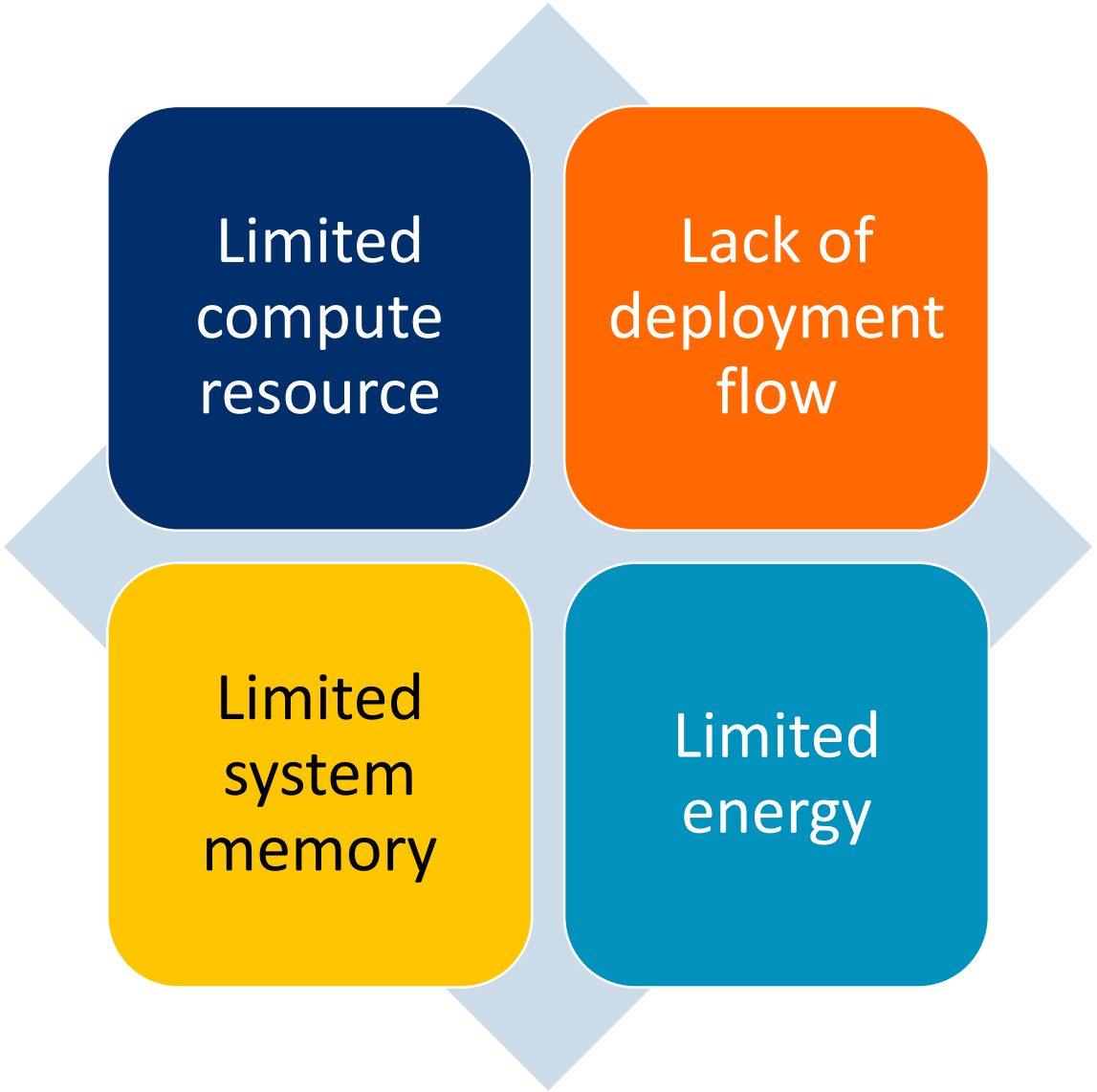
Compute Library 15x
faster than other open-source software (OSS)

ML Use Case Examples

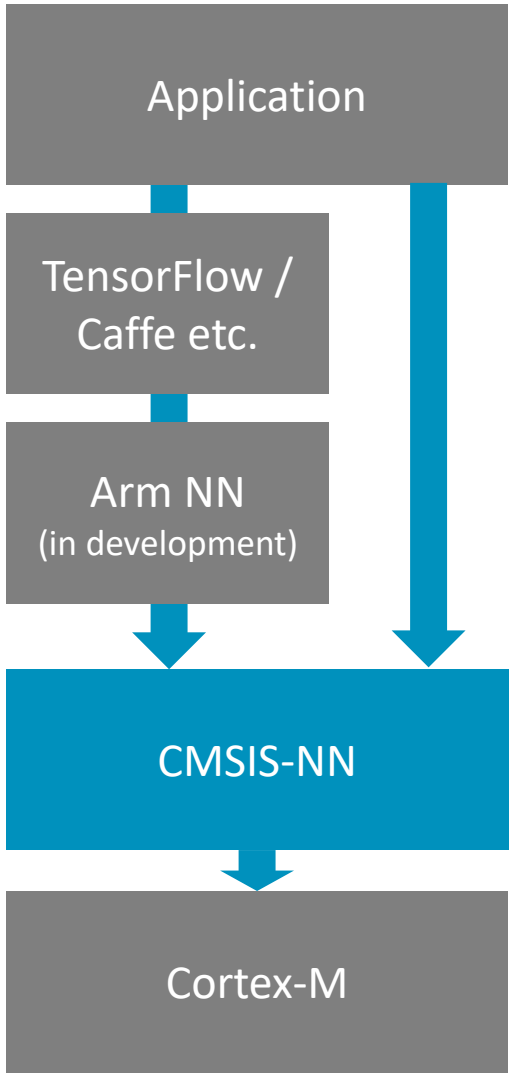
	Big data ML	vs	Small data ML
Vision	 ImageNet • 1000+ classes		 CIFAR-10 • < 10 classes
Audio	 Large scale speech recognition		 Key word spotting, simple commands
Health	 Disease detection		 Human activity monitor

arm

Cortex-M Challenges for ML



CMSIS-NN – Efficient NN Kernels for Cortex-M CPUs



- Open Source: launched 23 Jan'18
- CMSIS-NN has the equivalent role for Cortex-M CPUs as Compute Library has for Cortex-A CPUs and Arm Mali GPUs (and ML processor in mid 2018)
- But flow is entirely offline, creating a binary targeting Cortex-M class platform
- SIMD instructions in Cortex-M7/M4 targeted
- Will run on Cortex-M0

arm

CMSIS-NN – Efficient NN Kernels for Cortex-M CPUs

Convolution

- Boost compute density with GEMM based implementation
- Reduce data movement overhead with depth-first data layout
- Interleave data movement and compute to minimize memory footprint

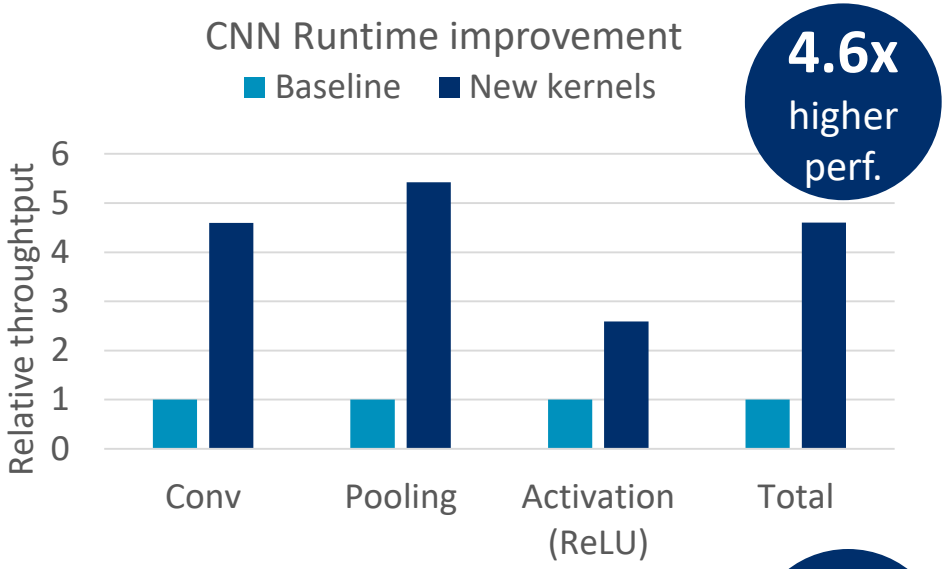
Pooling

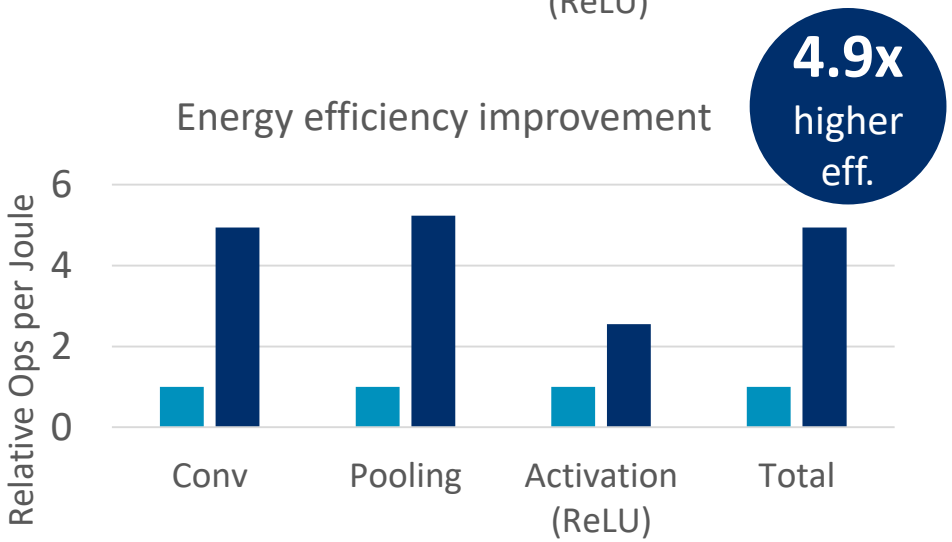
- Improve performance by splitting pooling into x-y directions
- Improve memory access and footprint with in-situ updates

Activation

- ReLU: Improve parallelism by branch-free implementation
- Sigmoid/Tanh: fast table-lookup instead of exponent computation

CMSIS-NN is now open-sourced



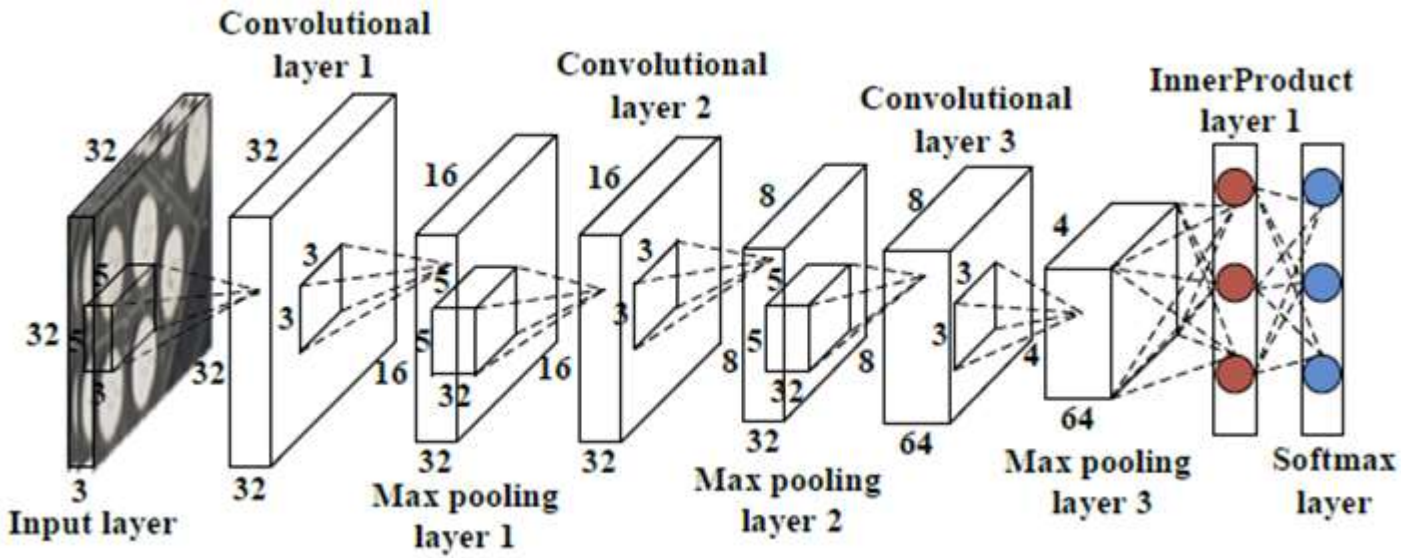


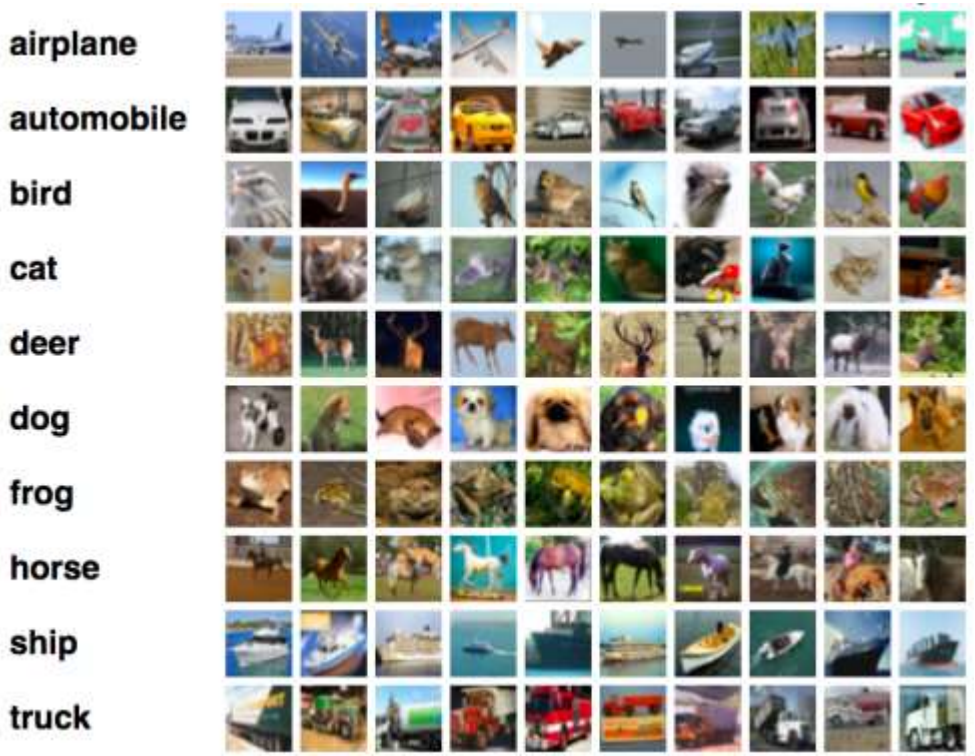
*Baseline uses CMSIS 1D Conv and Caffe-like Pooling/ReLU

arm

Image Classification - Convolutional Neural Network

- CIFAR-10 classification – classify images into 10 different object classes
- 3 convolution layer, 3 pooling layer and 1 fully-connected layer (~80% accuracy)





arm

CNN on Cortex-M7

- CNN with 8-bit weights and 8-bit activations
- Total memory footprint: 87 kB weights + 40 kB activations + 10 kB buffers (I/O etc.)

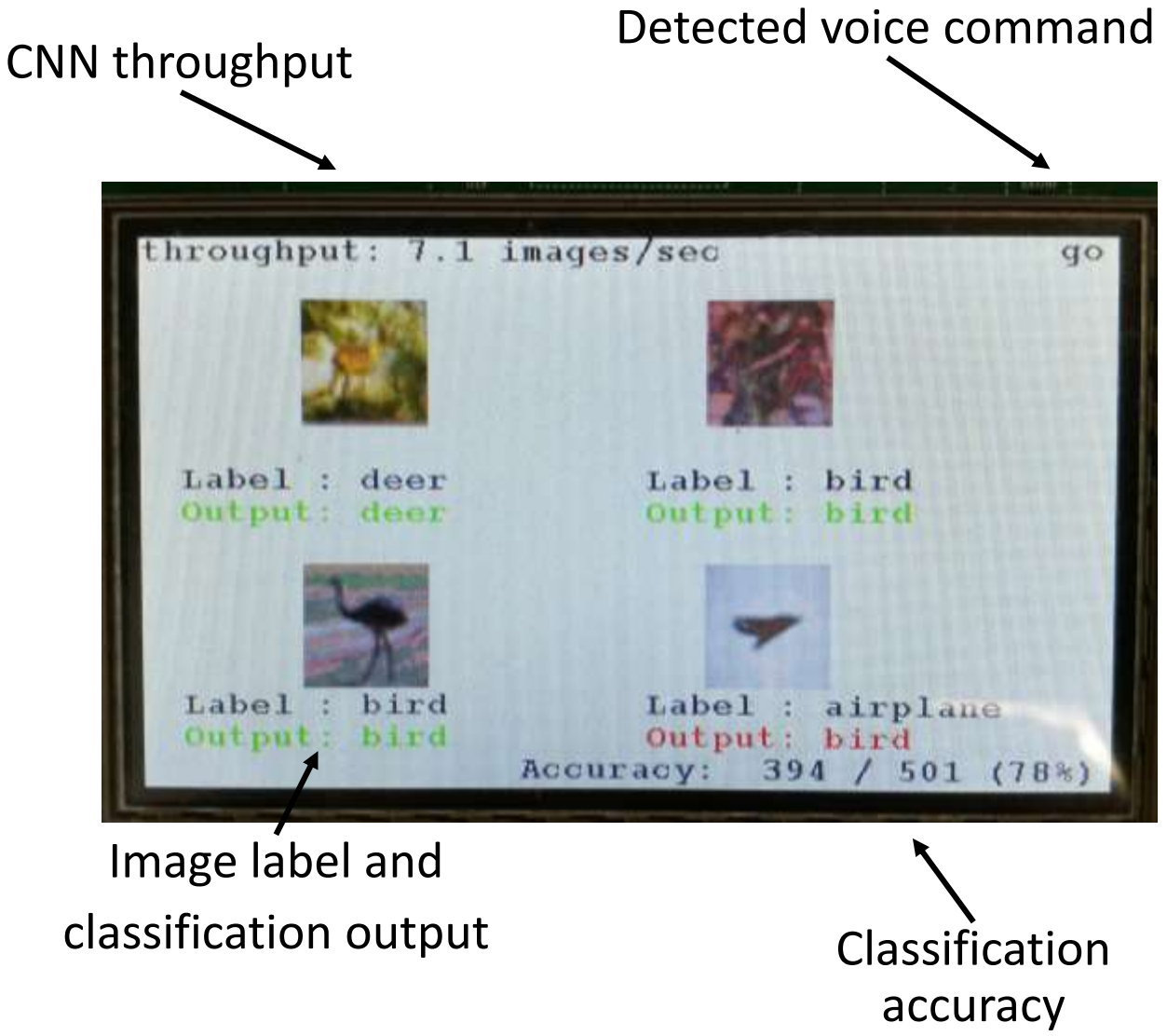
NUCLEO-F746ZG - 216 MHz, 320 KB SRAM

Layer	Network Parameter	Output activation	Operation count	Runtime on M7
Conv1	5x5x3x32 (2.3 KB)	32x32x32 (32 KB)	4.9 M	31.4 ms
Pool1	3x3, stride of 2	16x16x32 (8 KB)	73.7 K	1.6 ms
Conv2	5x5x32x32 (25 KB)	16x16x32 (8 KB)	13.1 M	42.8 ms
Pool2	3x3, stride of 2	8x8x32 (2 KB)	18.4 K	0.4 ms
Conv3	5x5x32x64 (50 KB)	8x8x64 (4 KB)	6.6 M	22.6 ms
Pool3	3x3, stride of 2	4x4x64 (1 KB)	9.2 K	0.2 ms
ip1	4x4x64x10 (10 KB)	10	20 K	0.1 ms
Total	**87 KB weights**	**Total: 55 KB Max. footprint: 40 KB**	**24.7 M Ops**	**99.1 ms**

arm

Demo with Multiple NNs

- Both image classification and keyword spotting are running at the same time
- Voice command controls the start/stop of the image classification
- Total memory footprint:
  - CNN: 87 KB weights + 40 KB activations + 10 KB buffers
  - DNN: 66 KB weights + 1 KB activations + 2 KB buffers



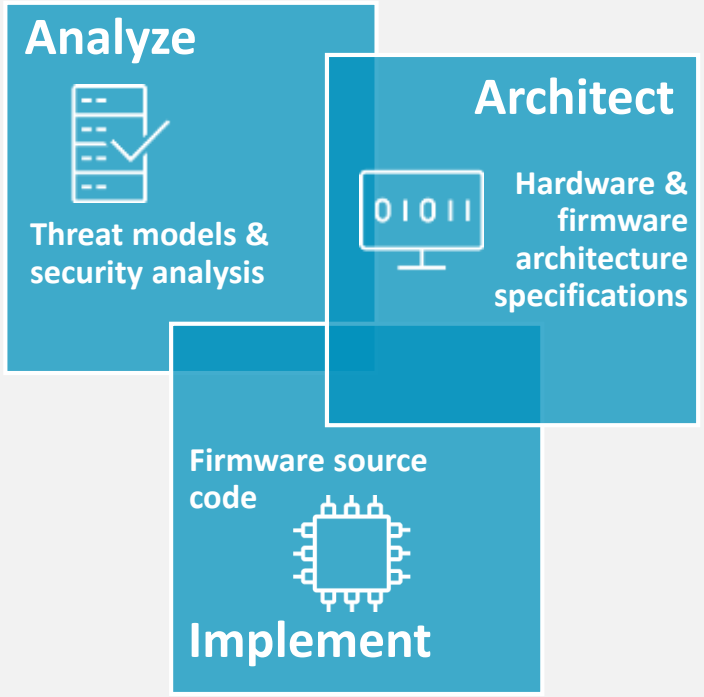
arm

arm

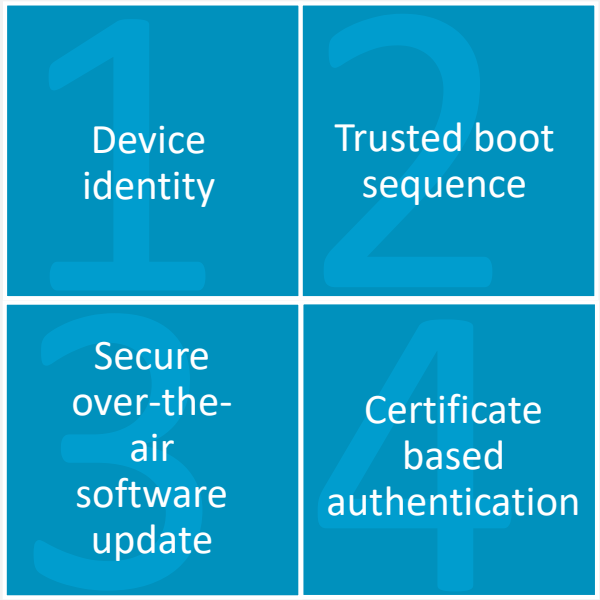
Platform Security Architecture

Platform Security Architecture

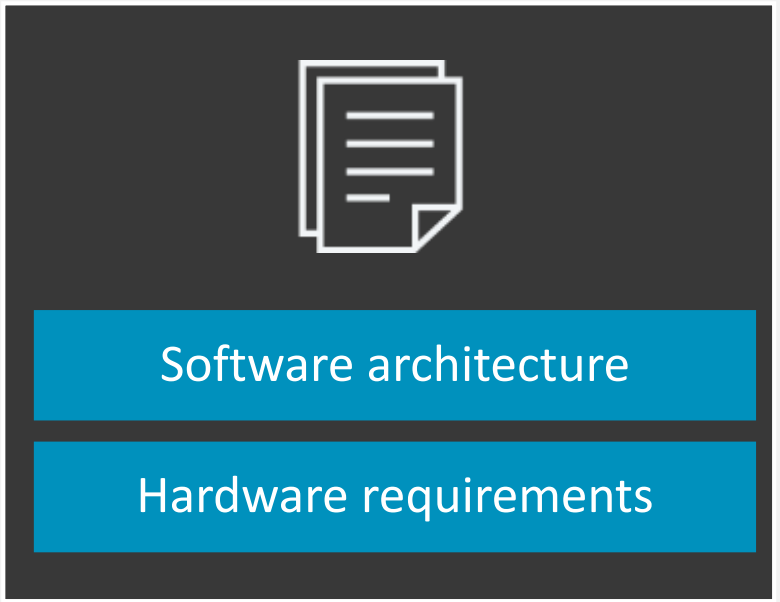
A recipe for building a secure system & a reference implementation



3 Parts to PSA



Common principles across multiple use cases

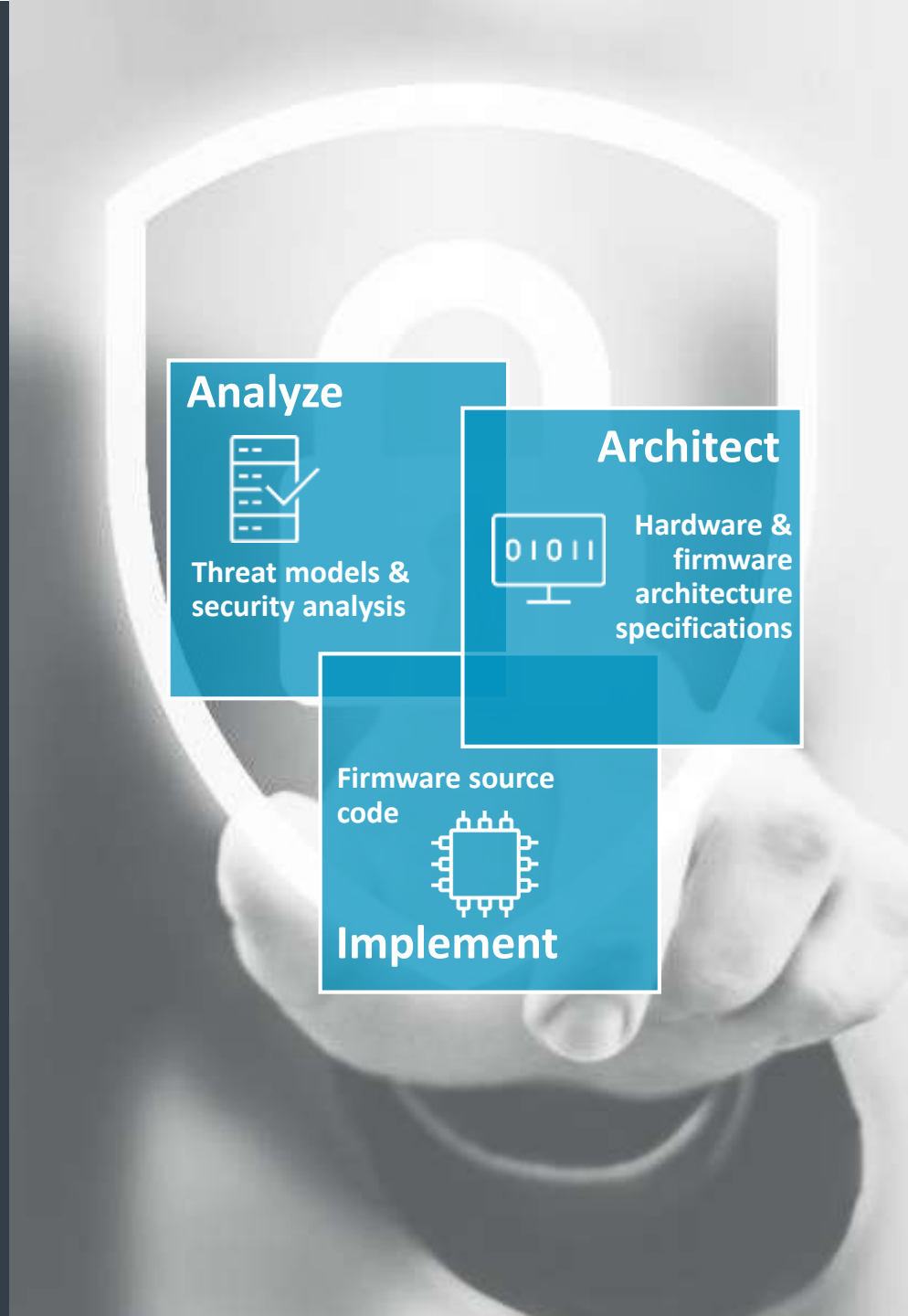


Architecture & Specifications

arm

Arm Platform Security Architecture (PSA)

- A common framework for scaling connected device security
- Enables consistent level of security
- Broad ecosystem support from industry leaders
- Trusted Firmware-M – Open source reference firmware



First PSA deliverables available

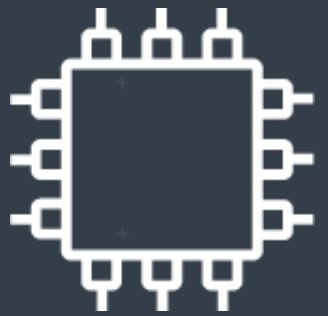
www.arm.com/psa-resources

Threat Models and Security Analyses (TMSA) documentation

- **Step 1 of PSA**: gather information about threats to a particular device and develop the right security specifications
- Three example TMSAs **freely available now**

Arm Trusted Firmware-M

- The **first open source reference implementation** firmware, which conforms to the PSA specification
- Available as a GitHub project in March



arm

Summary

Major initiatives from Arm supporting Cortex-M microcontrollers

- Machine Learning on IoT-class devices
  - Enabling existing ML frameworks on Cortex-M through Arm NN
  - ML enabled everywhere: Cortex-M0 and upwards
  - CMSIS-NN library open source and available now (https://developer.arm.com/embedded/cmsis)

- Platform Security Architecture
  - Security from the ground up, at the core of every device
  - Trusted Firmware-M coming Q1’18
  - First deliverables available now (www.arm.com/psa-resources)

arm