Implementation of a voice trigger using low-power Arm processors

Laurent Le Faucheur
Principal Engineer
laurent.lefaucheur@arm.com

Voice Summit (Voice.ai)
Newark, NJ
July 25, 2018

What we will cover today

1. Arm portfolio
2. Power consumption per use-case
3. Signal processing coupled with NN
4. Securing your firmware and data
5. Questions

Arm computing is everywhere

#1

shipping GPU in the world is Mali

8Bn

Arm-based embedded chips shipped in 2017

> 5Bn

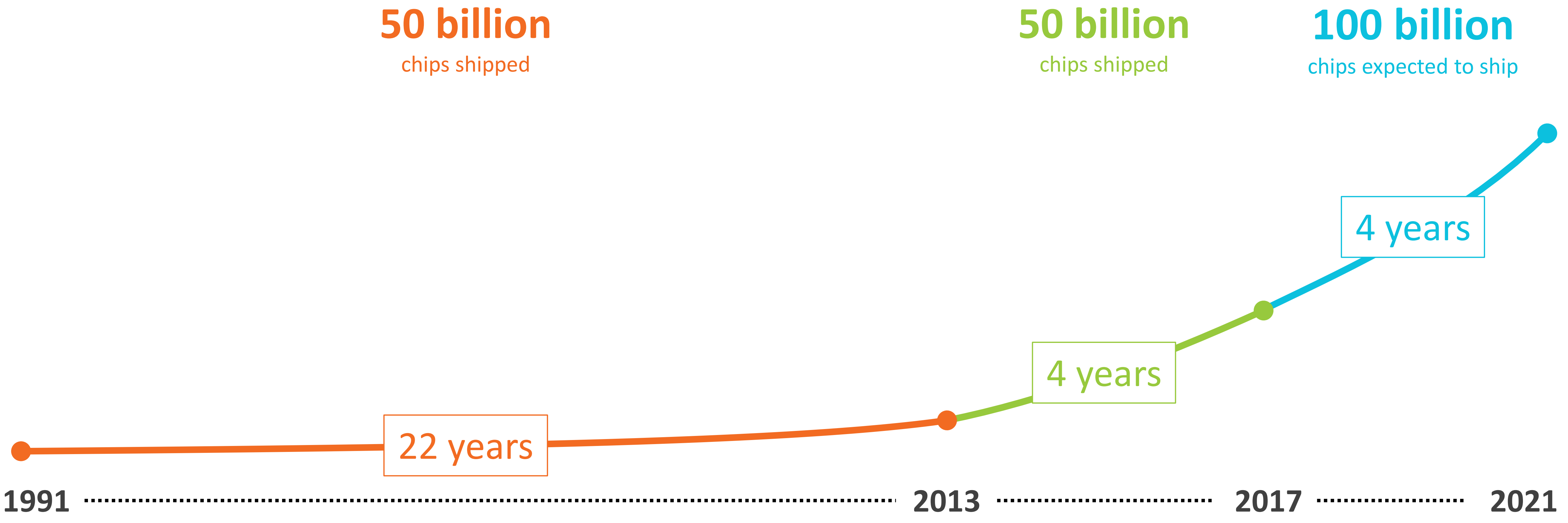
people using Arm-based mobile phones

125Bn

Arm-based chips to date

arm

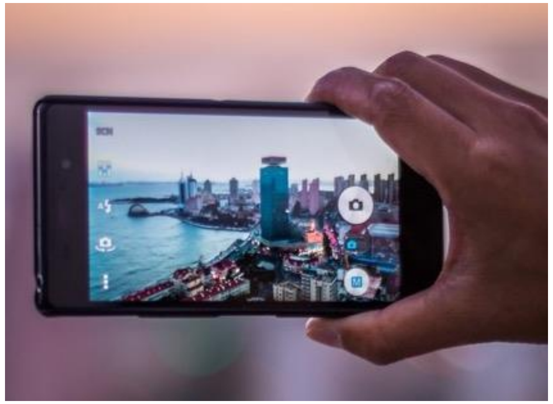
Market trends – acceleration of technology deployment



Devices with Arm processors

Diverse applications need diverse compute

19Bn
Arm-based processors shipped in 2017



Cortex-A

Highest performance

Designed for high-level operating systems

Cortex-R

Faster responsiveness

Designed for high performance, hard real-time applications

Cortex-M

Smallest/lowest power

Designed for discrete processing and microcontrollers

SecurCore

Tamper resistant

Designed for physical security

Voice interactions use-cases – processor and architecture options

Voice-triggered device and application	Distance	Voice Processing	Voice Processors
MIC Audio Front End (edge capture)	Hands-free	1 MIC. Simple Voice Activity Detection. **No KWS support**. Audio sample are streamed to voice service client AP. Possibly battery operated. (Always ON). Avoid far-field issues. Limited noise immunity	Cortex-M0
Smart Voice Controller	Hands-free	1 MIC. VAD with a **small command-set** recognition for service activation (lights, door, watch). Possibly battery operated. Cloudless	Cortex-M3
Smart Voice Controller Headphone, Wearable	Hands-free	1/2 MIC Smart Voice Controller with **enhanced audio processing** (wind noise removal) and sensor fusion compute. Battery operated. Cloudless	Cortex-M4
Smart Speaker	Far-field	**Multi MIC** smart device with enhanced audio processing capability for noise cancelling, beam forming and barge-in audio player use case. Cloud based voice service. Power plug.	Cortex-M7 Cortex-Ax
Smart Speaker (autonomous)	Far-field	Smart speaker with **autonomous ASR/NLU** services via NN algorithm implementation. Cloudless support. Power plug	Cortex-M7 Cortex-Ax
Smart Home Hub	Far-field	Autonomous **Smart Hub** with sensor fusion (audio/video/radar) to support rich home services. Cloudless support. Power plug	Cortex-Ax NN Copro

arm

Variety of voice triggered devices

There are several system requirements for always-on

- Battery-operated devices have a focus on **power consumption**
- Hubs have a focus on audio capture and rendering **signal-processing performance**

Examples of voice-triggered devices powered by Cortex-A53, Cortex-A9, Cortex-A7, and Cortex-M3 processors

Power optimization for always on

arm

Microphones - overview

Two areas of sound pressure for smart-speakers:

- Below 72dBSPL: automatic gain control, beamforming and noise reduction is recommended
- Above 72dBSPL: speech is sent to the recognizer with minimal signal processing

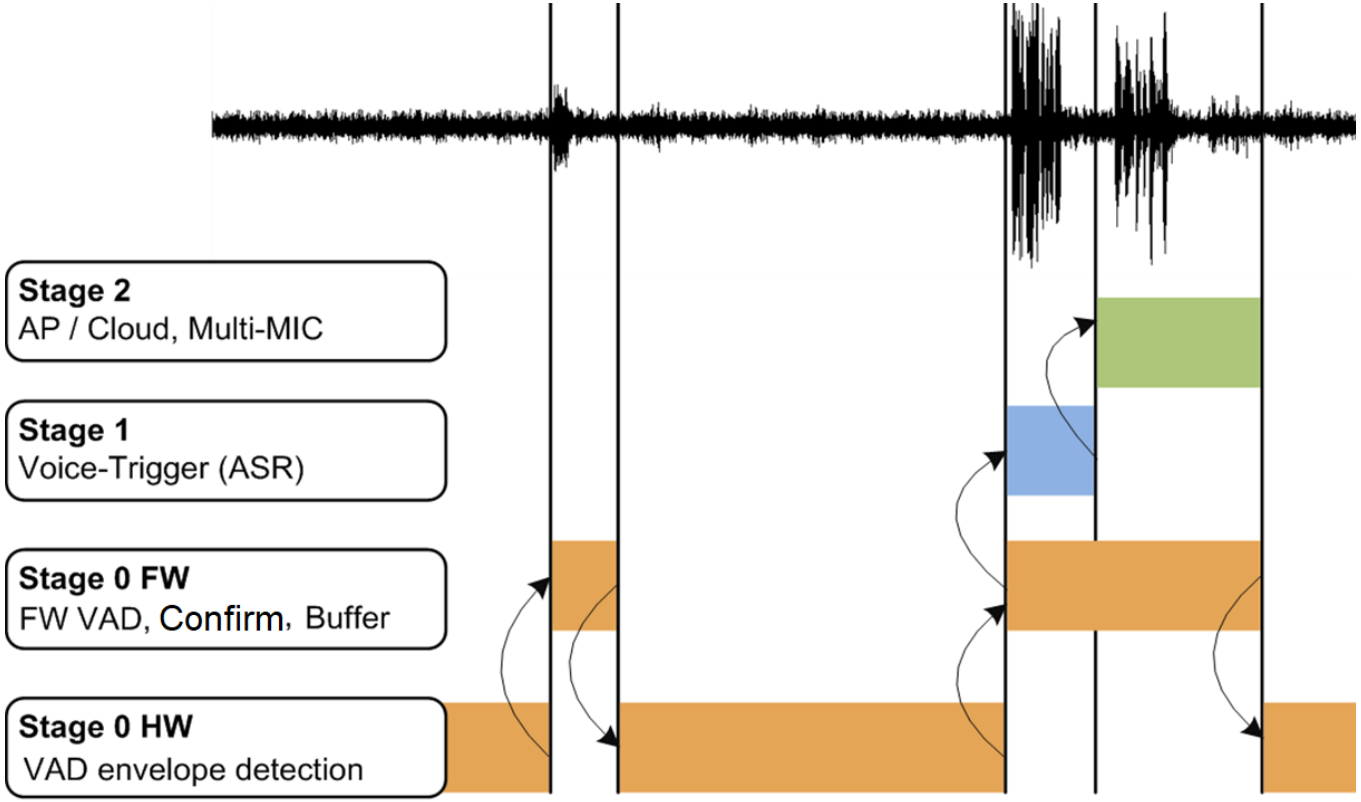
Floor noise	Level at 2m	Level at 50cm	Level at 30cm	MRP (-4dBPa)	Speech crest	Overload
30dB	54dB	66dB	72dB	90dB	106dB	125dB

How many microphones are needed in your application?

9

Speech and audio detection stages

Power optimization comes from the activation of a progressive number of features



“**VAD**”: Voice Activity Detector

arm

Power numbers for battery operated voice / sound triggers

Sound trigger using Voice Activity Detection (VAD)

One MIC activated at 0.200mA	= 0.2000
Light VAD (0.06MHz) 100% of time @0.1mA/MHz	= 0.0060
Application 6MHz 1% of time (during 1s) @0.1mA/MHz	= 0.0060
Oscillator, Quiescent current, Deep-Power-Down periods...	= 0.0060
Radio communication 0.01% (9 seconds)	= 0.0060

Total = ~0.2 mA *(assuming one voltage domain)* <= > 1 year (9000hours) with a 2500mAh battery

Question : how would you implement a sound detection device having a 3 years life time from a single AA?

Microphone Cortex-M4 Connectivity AA

Power numbers for battery operated voice / sound triggers

Question : how would you implement a **sound detection** device having a **3 years** life time from a single **AA**?

3 years = 26k hours,
AA battery capacity is 2500mAh
Average current consumption is about 0.1mA (the microphone consumes 0.2mA !)

Solution : activate the PDM clock of the microphone by bursts of 10ms and let it idle during 40ms
A digital microphone recovers its sensitivity within ~8ms after power-up and clocking
You will analyze the last 2ms of the burst to check sound-pressure variations from last bursts

Microphone Cortex-M4 Connectivity AA

Arm key differentiation for always-on operations

Small processor with minimal **dynamic power** (*)

Several level of **sleep modes**

Efficient signal processing on Cortex-M0+ thanks to **14 x 32 bits registers** and a **single-cycle 32x32** multiplier

Thumb mode with high code/data density

Firmware R&D investments are preserved

Ease of **dual-sourcing**

(*) Cortex-M0+ : 4µW/MHz @40LP, 2.46CoreMark/MHz
Cortex-M3 : 11µW/MHz @40LP, 3.34CoreMark/MHz
Cortex-M7 : 33µW/MHz @28HPM, 5 CoreMark/MHz
https://developer.arm.com/products/processors/cortex-m

Signal processing

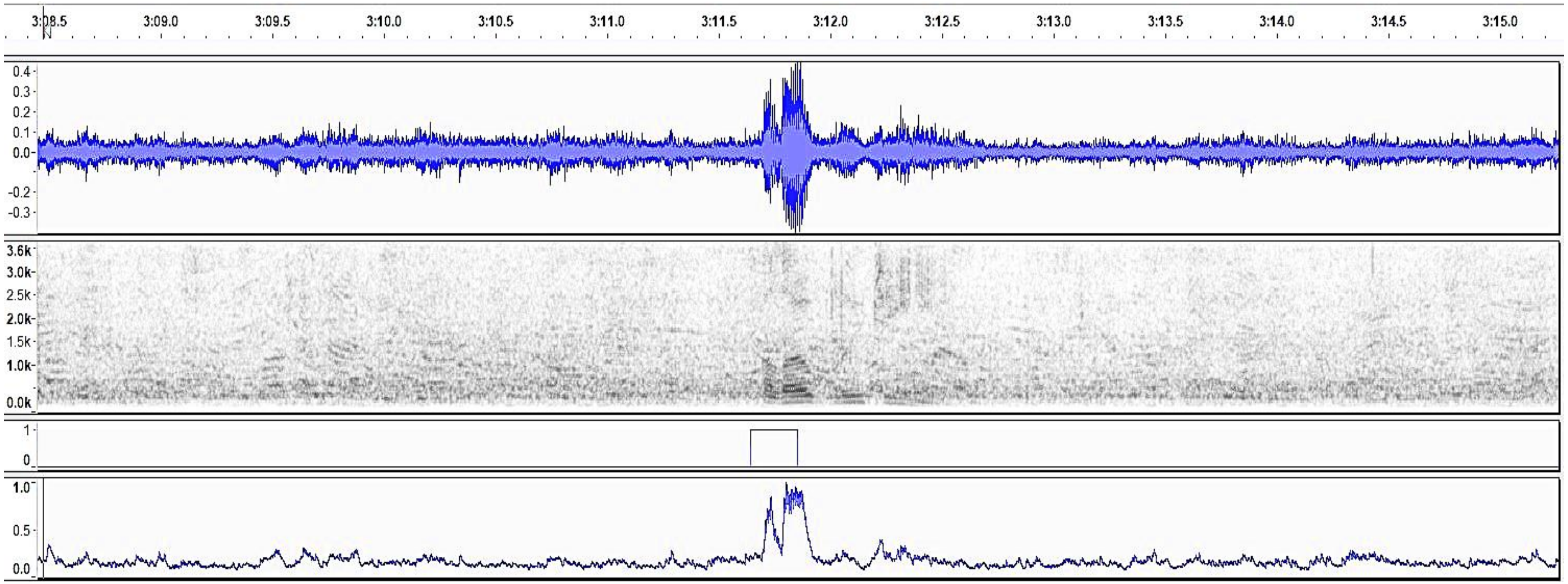
arm

Voice control building blocks

Mic array	Always on hardware
Front end	Beam forming Voice activity detector Echo cancellation Noise suppression
Back end	Voice activity detector Automatic voice recognition
Cloud / Local	Natural language understanding Machine learning

Practical example of a low-power VAD *(voice activity detection)*

Cortex-M0+ with low-power VAD computations



Voice + noise

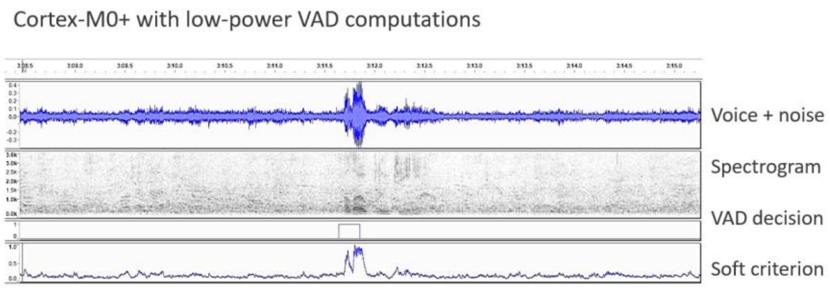
Spectrogram

VAD decision

Soft criterion

Question : how would you implement this VAD at less than 0.1MHz on a Cortex-M0+ with a 16kHz input sampling rate ?

Practical example of a low-power VAD



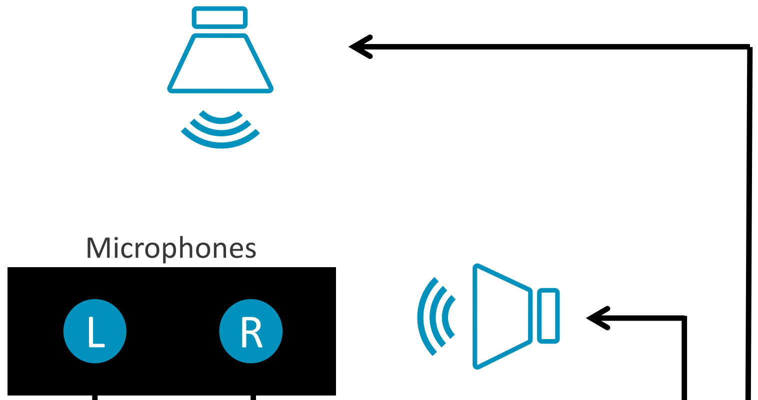
Question : how would you implement this VAD below 0.1MHz on a Cortex-M0+
with a 16kHz input sampling rate ?

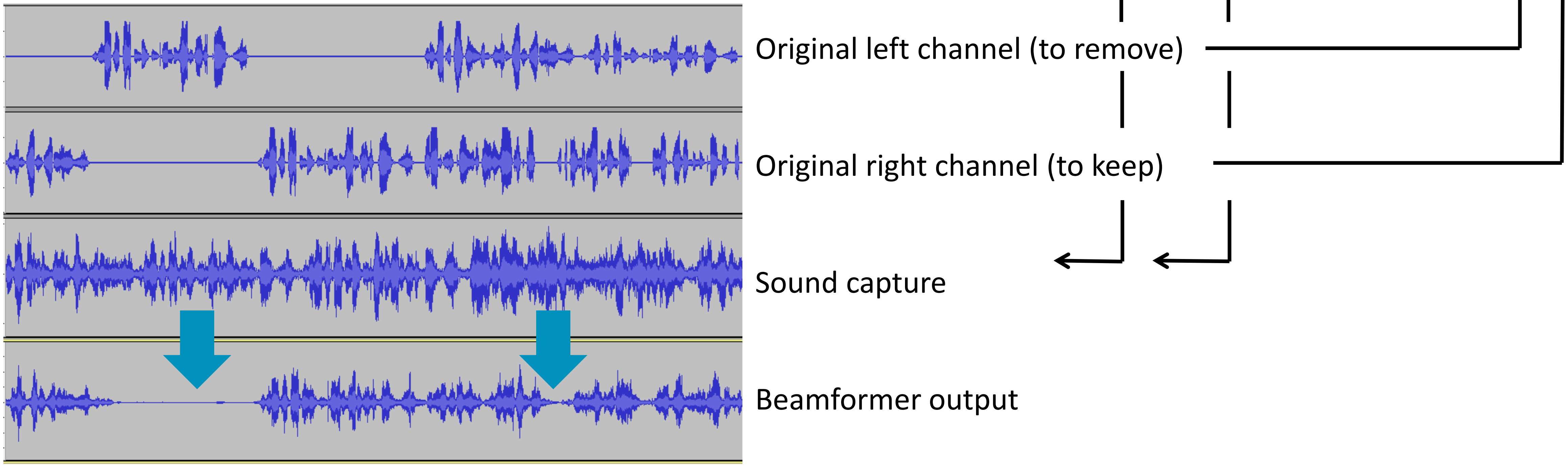
A CPU load of 0.1MHz with an input sampling rate of 16kHz means 6 cycles per sample !

Solution : break the Shannon rule
decimate the input rate by taking one sample over 16, without pre-filter
this will create a new signal with spectral components folded in the low-frequency range
and finally compute the energy (first order IIR applied on absolute values)
take the min and max of the observation and create the soft criterion every 16ms
the cycle count comes down to about 30 per sample at 1kHz rate.

Practical example of a beam-former

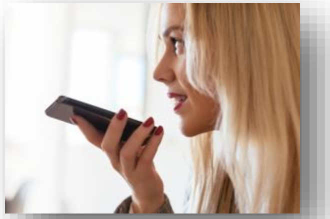
Beamformer complexity in the 6MHz range on Cortex-M7.

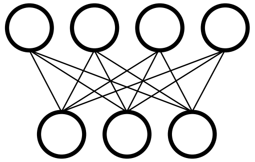




arm

Developing NN Solutions on Cortex-M





```
// conv2 img_buffer2 -> img_buffer1
arm_convolve_HWC_q7_fast(img_buffer2, CONV2_IM_DIM, CONV2_IM_CH, conv2_wt, CONV2_OUT_CH, CONV2_KER_DIM,
                         CONV2_PADDING, CONV2_STRIDE, conv2_bias, CONV2_BIAS_LSHIFT, CONV2_OUT_RSHIFT, img_buffer1,
                         CONV2_OUT_DIM, (q15_t *) col_buffer, NULL);

arm_relu_q7(img_buffer1, CONV2_OUT_DIM * CONV2_OUT_DIM * CONV2_OUT_CH);

// pool2 img_buffer1 -> img_buffer2
arm_avepool_q7_HWC(img_buffer1, CONV2_OUT_DIM, CONV2_OUT_CH, POOL2_KER_DIM,
                   POOL2_PADDING, POOL2_STRIDE, POOL2_OUT_DIM, col_buffer, img_buffer2);
```

Problem

↓ Hardware-constrained NN model search

Trained NN model

↓ NN model translation

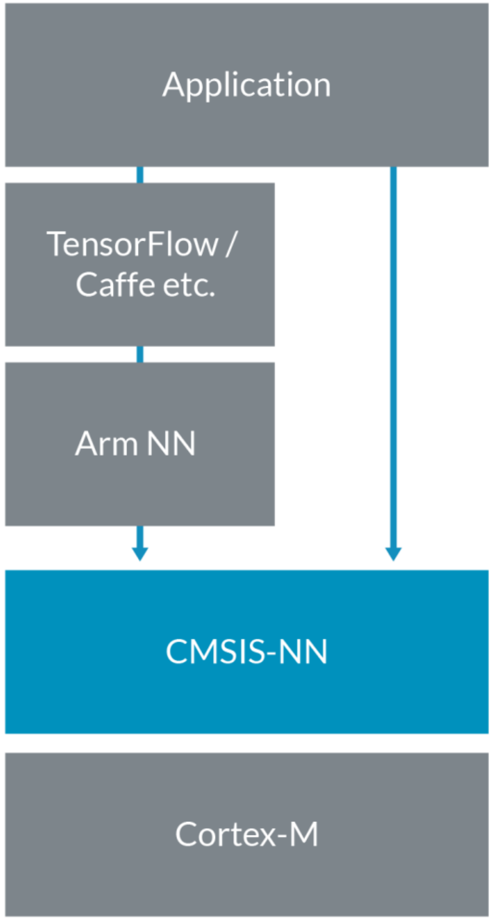
Deployable solution

↓ Optimized NN functions: CMSIS-NN

Optimized code on h/w

Model Deployment on Cortex-M MCUs

- Running ML framework on Cortex-M systems is impractical
- Need to run bare-metal code to efficiently use the limited resources
- Arm NN translates trained model to the code that runs on Cortex-M cores using CMSIS-NN functions
- CMSIS-NN: optimized low-level NN functions for Cortex-M CPUs
- CMSIS-NN APIs may also be directly used in the application code



CMSIS : Cortex Microcontroller Software Interface Standard

A versatile DSP ecosystem for Cortex-M

Fundamental DSP Functions on Cortex-M

-> available for free!

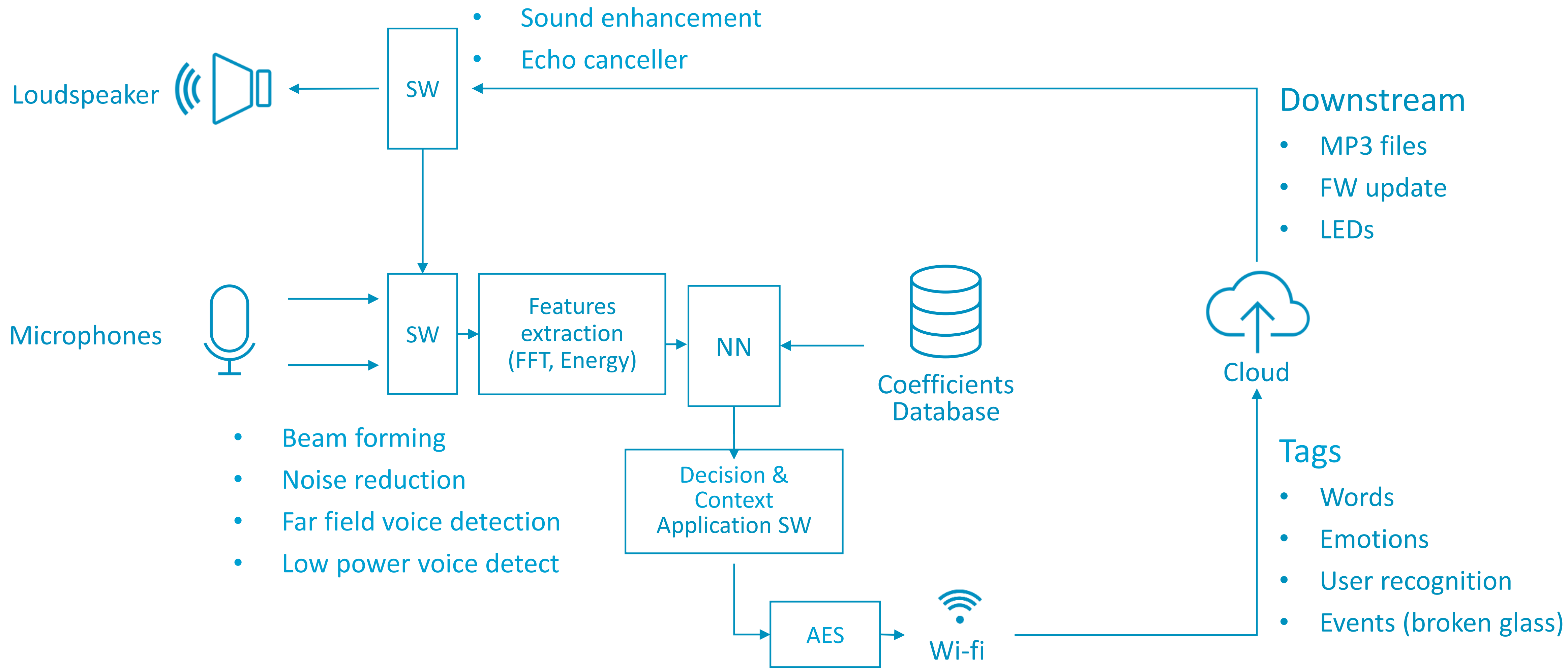
CMSIS-DSP library	
Filters	Controller functions
Basic math functions	Interpolator functions
Statistical functions	Matrix functions
Support functions	Complex math functions
Fast math functions	Transforms

Examples of ecosystem solutions and partners

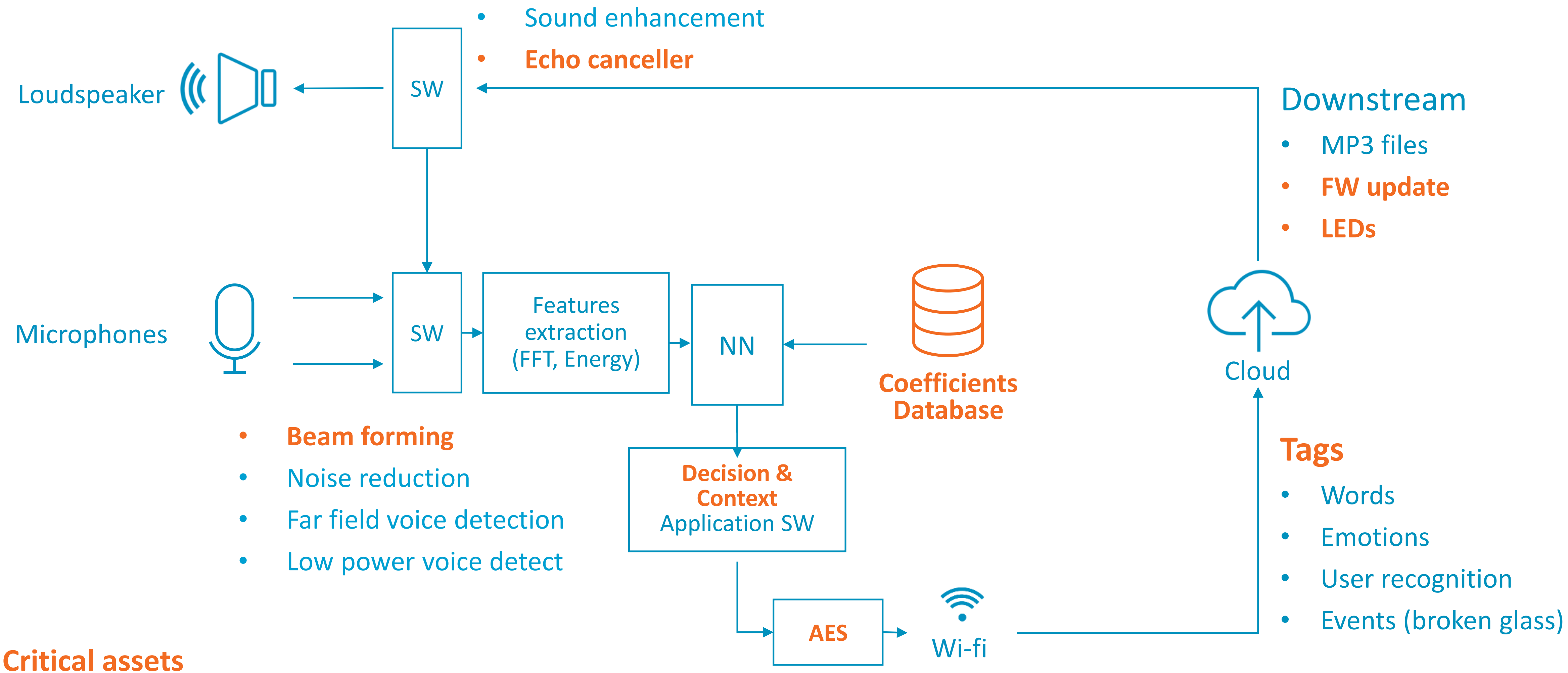
- Audio codecs
- Voice codecs
- Image processing
- Keyword spotting
- Audio enhancement
- Sensor fusion
- Motor control
- Connectivity
- Simulation tools

Security

Keyword spotting - security



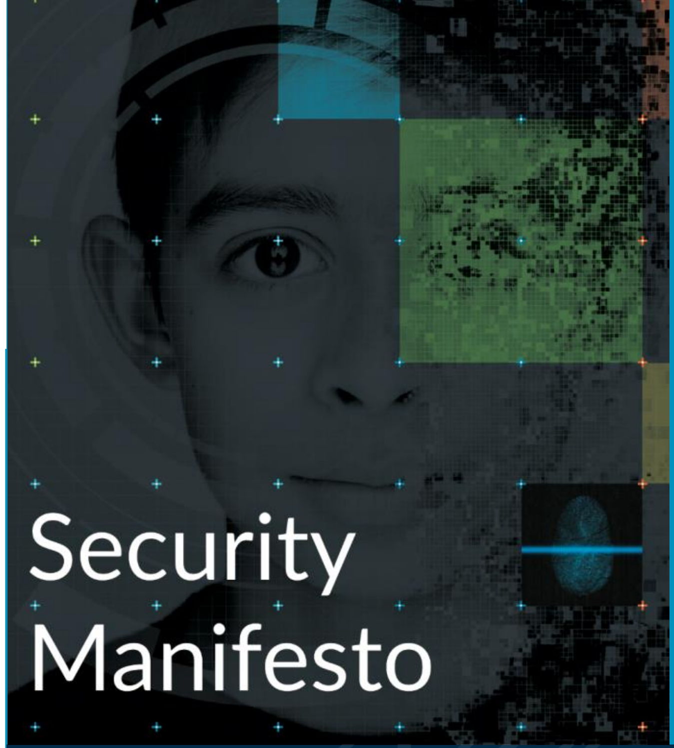
Keyword spotting - security

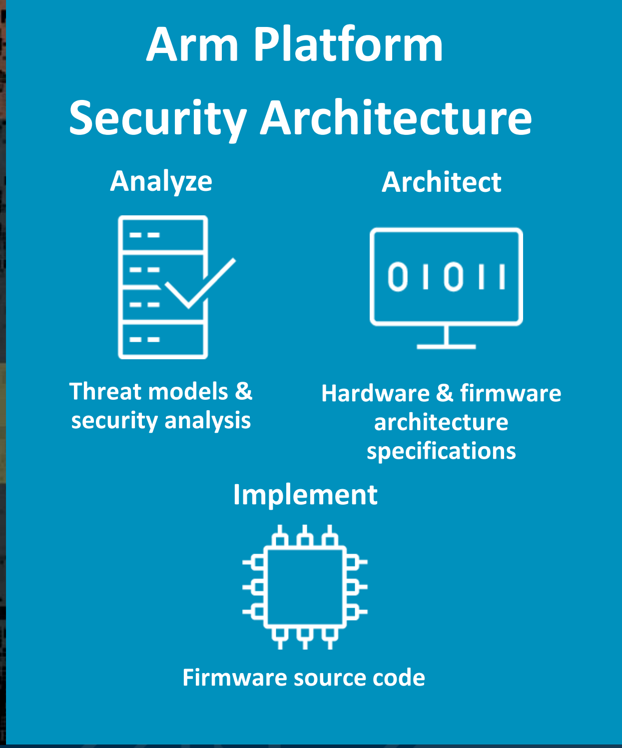


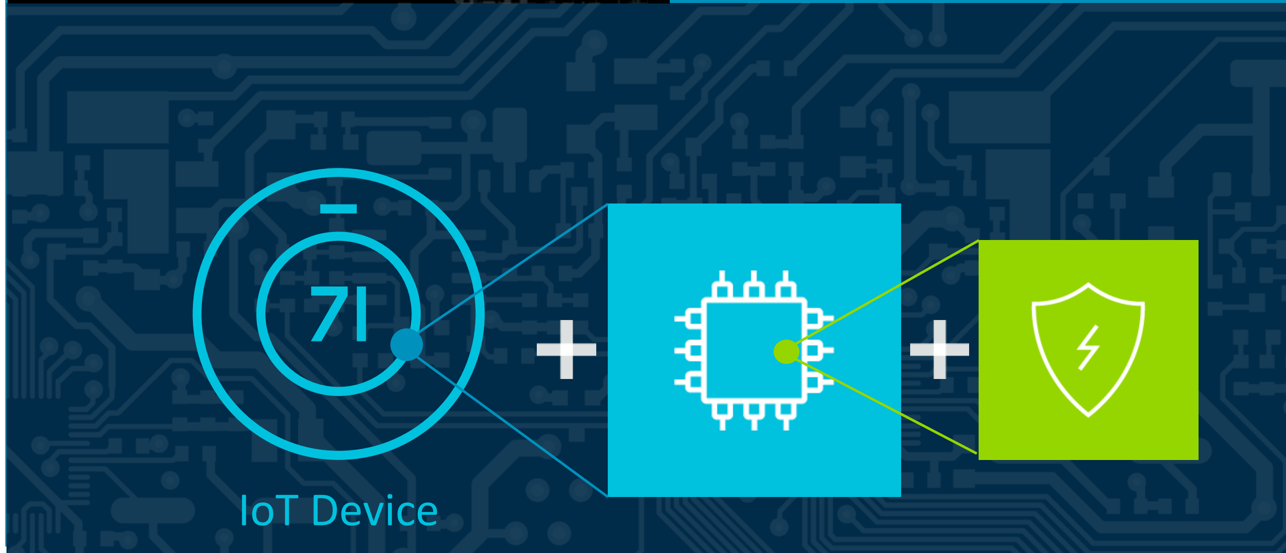
Critical assets

Arm's security vision

- Security needs to be built from the ground up… and at the core of every device
- No single point of ownership, whole IoT value chain needs to share the responsibility
- Simple and seamless integration of security from foundational architecture to cloud service is key







Summary

Arm portfolio lets you secure your R&D investment and go to market faster.

Resources

- CMSIS-NN paper: https://arxiv.org/abs/1801.06601
- CMSIS-NN blog: https://community.arm.com/processors/b/blog/posts/new-neural-network-kernels-boost-efficiency-in-microcontrollers-by-5x
- CMSIS-NN Github link: https://github.com/ARM-software/CMSIS_5/

- KWS (Keyword Spotting) paper: https://arxiv.org/abs/1711.07128
- KWS blog: https://community.arm.com/processors/b/blog/posts/high-accuracy-keyword-spotting-on-cortex-m-processors
- KWS Github link: https://github.com/ARM-software/ML-KWS-for-MCU/

- ArmNN: https://developer.arm.com/products/processors/machine-learning/arm-nn
- ArmNN SDK blog: https://community.arm.com/tools/b/blog/posts/arm-nn-sdk
- https://developer.arm.com/technologies/machine-learning-on-arm/developer-material/white-papers/the-power-of-speech

arm

Thank You!
Danke!
Merci!
谢谢!
ありがとう!
Gracias!
Kiitos!