ARM KNOWLEDGE LAB: DESIGNING THE FUTURE

DAY 2

Accelerating Embedded ML with CMSIS

Trevor Martin, Hitex UK

Arm Knowledge Lab:
Three short and compact sessions

Day 1

Secure by Design – From Threats to Trusted Devices

Day 2

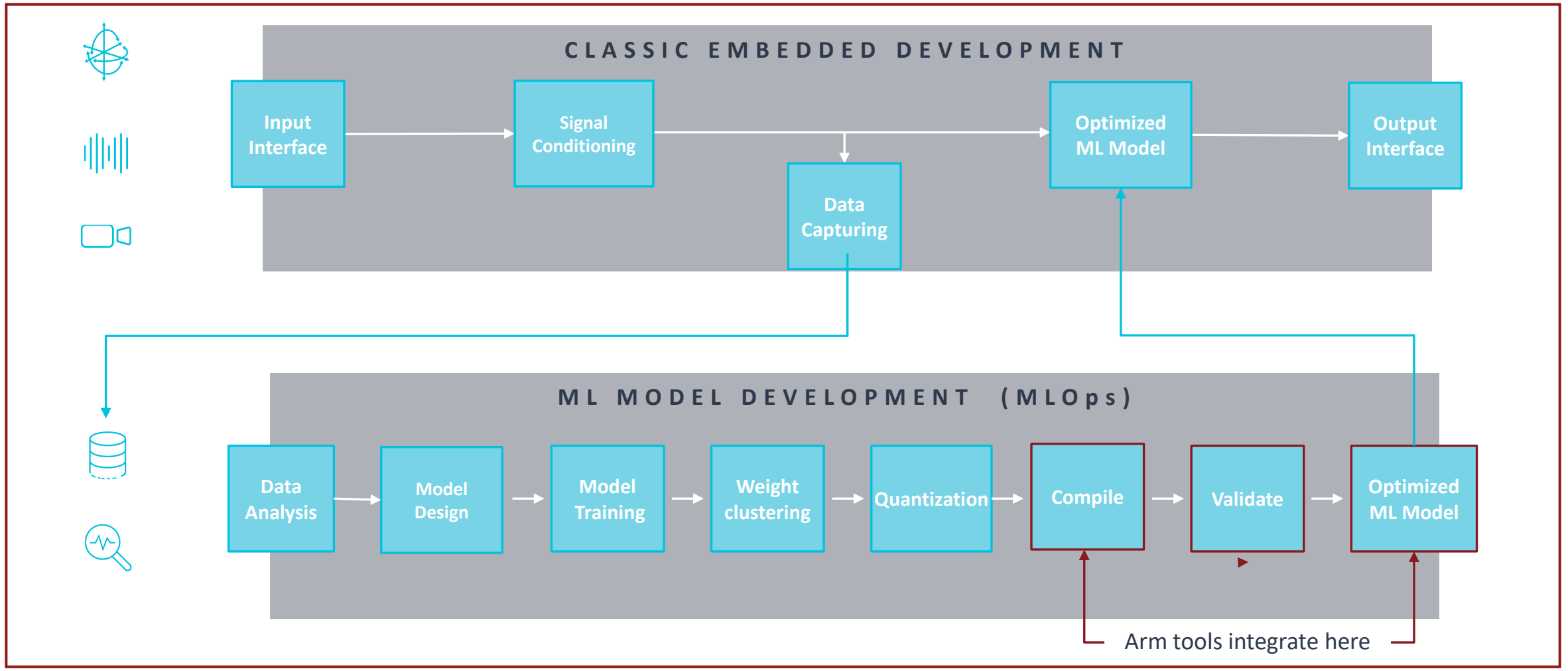
AI at the Edge – Building Smarter Embedded Systems

Day 3

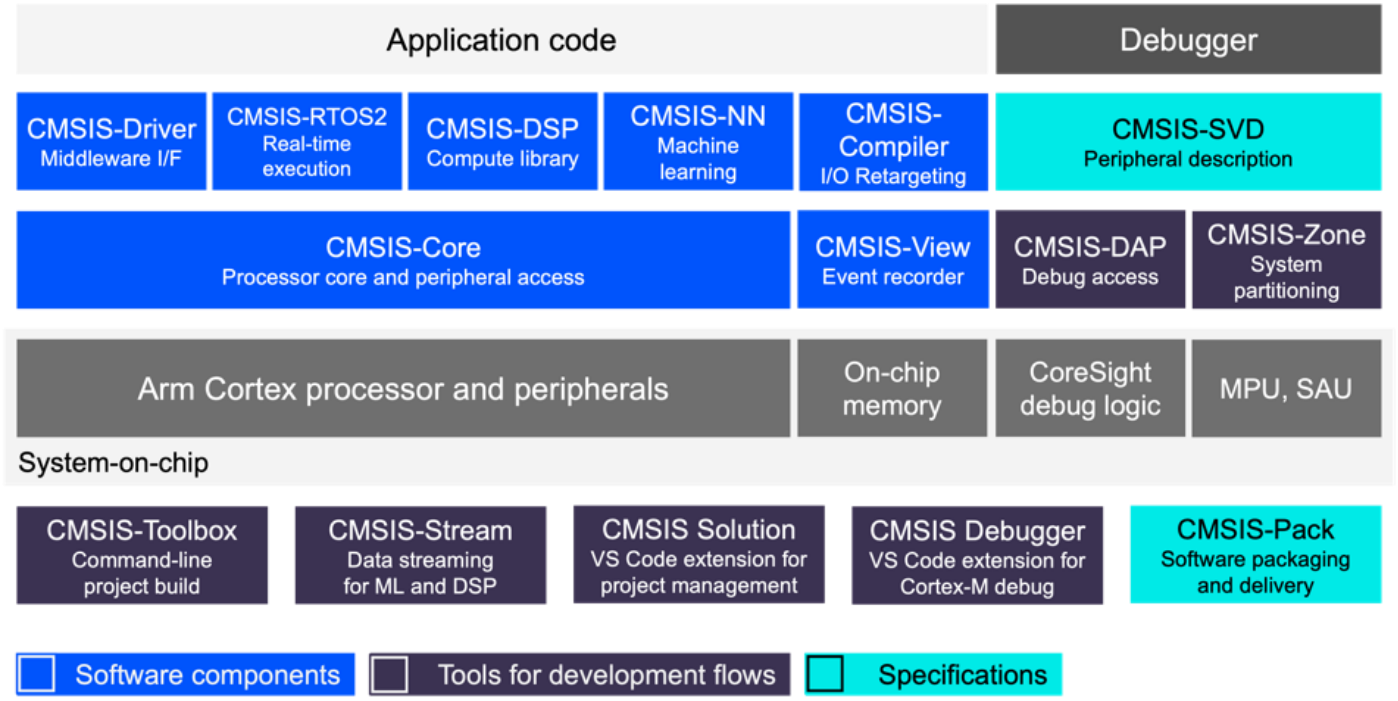
Safety-Critical Systems – Engineering Reliability

Find more presentations from our Arm Knowledge Lab:
https://www2.hitex.com/arm-knowledge-lab-2026-download

Development Flow for Edge AI Devices



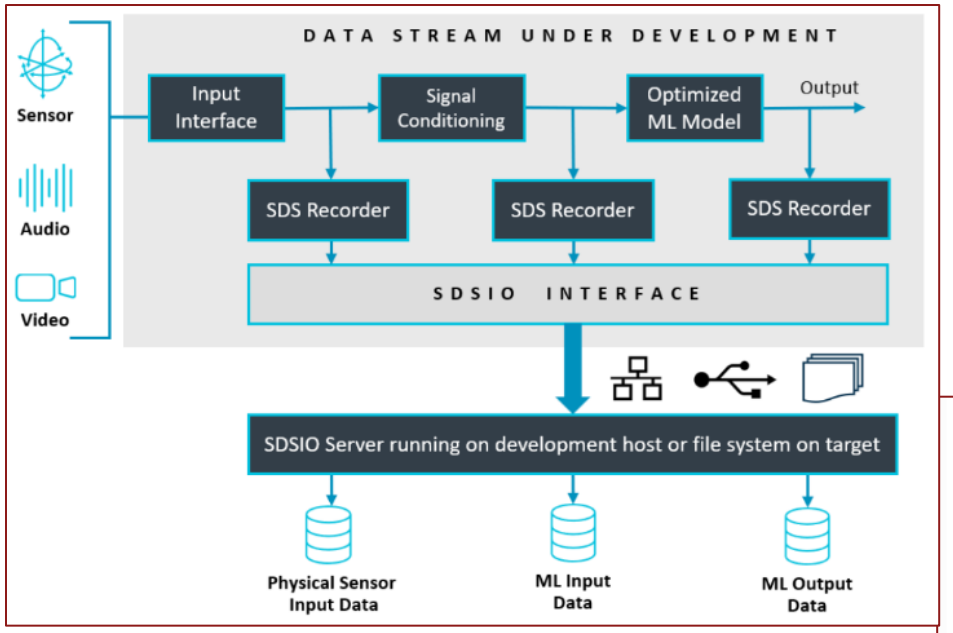
CMSIS

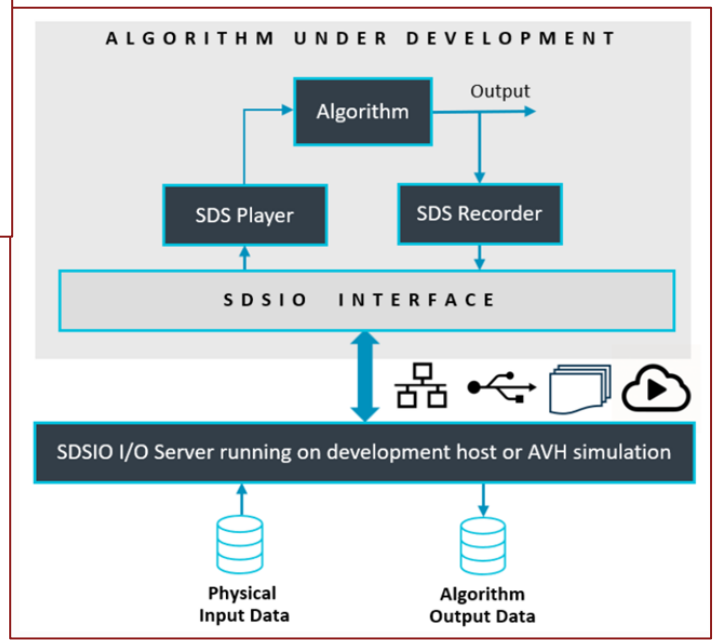


- Common Microcontroller Software Interface Standard
  - Standards, Firmware, Tools and Workflows

Arm SDS Framework

- Data Stream management Framework
- Recording and playback of real-world data streams
  - Audio
  - Video
  - Sensor
- Storage via TCP/IP, USB, Embedded File System

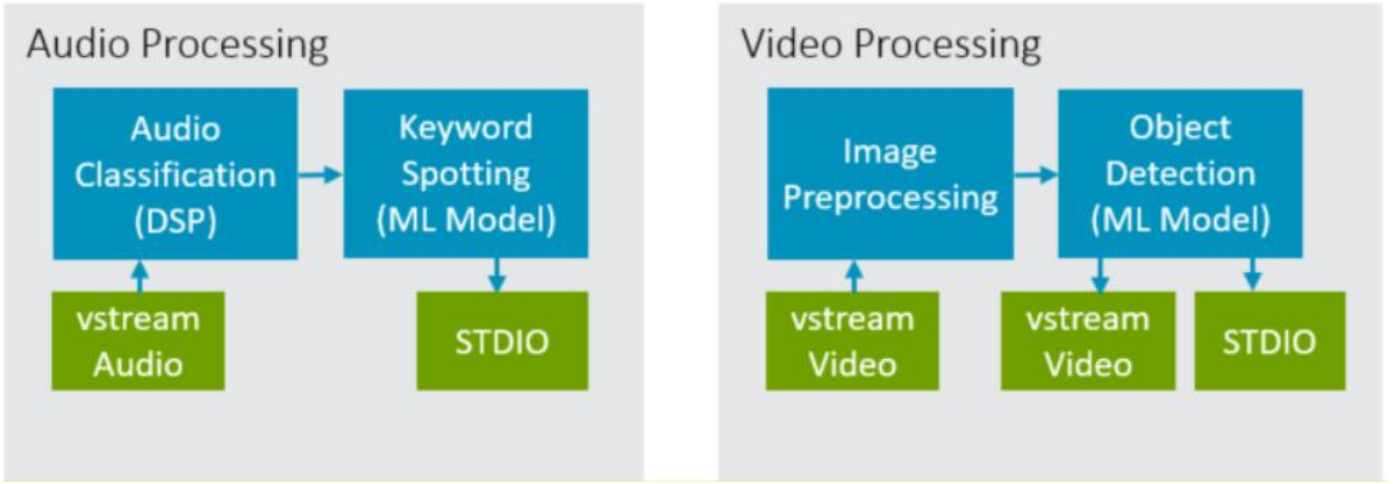




CMSIS-Driver

vStream

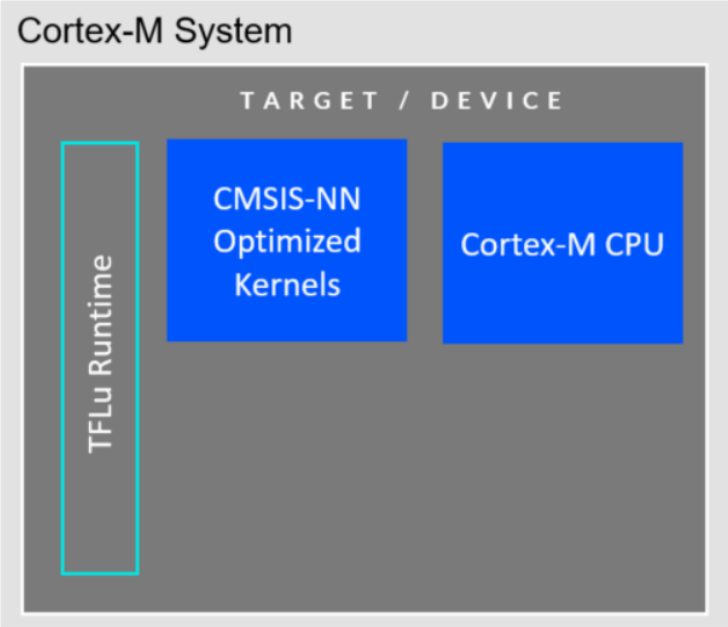
- CMSIS-Driver specification
- A virtual streaming interface
- Abstraction layer for streaming fixed sized data blocks
- Standardised Interface for common sensors
  - Sensors
  - Microphones
  - Camaras
  - Speakers



CMSIS NN

Makes use of the optimised matrix functions in CMSIS DSP

- Optimised for Cortex-M4 and Cortex-M7 SIMD instructions
  - 5 x performance boost over standard C code
- Uses same integer data format as CMSIS DSP

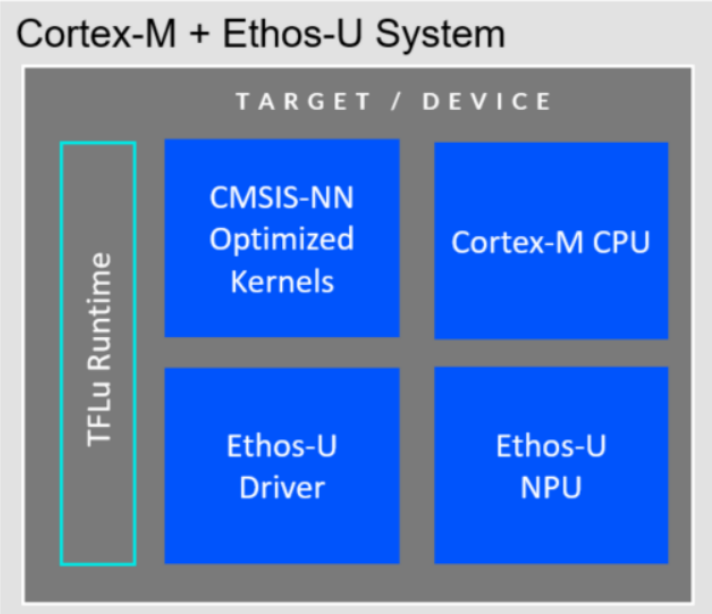


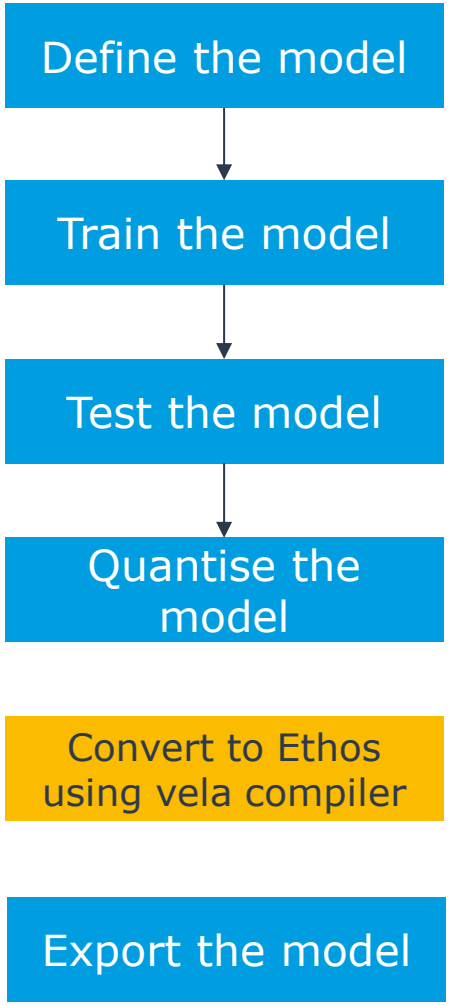
CMSIS-NN
NN Convolution functions
NN Pooling functions
Fully connected layer functions
NN Activation functions
Softmax functions
Support functions

Ethos NPU support

Ethos NPU

- Interface Driver
- Vela Compiler
  - Convert tflite model for Ethos
  - Free to use





Algorithm Design

Define the model. The input shapre is 28 x 28 which is flattened into a 1D vector. This is followed by a fully connected layer and a final softmax layer with ten classes

+ Code + Text

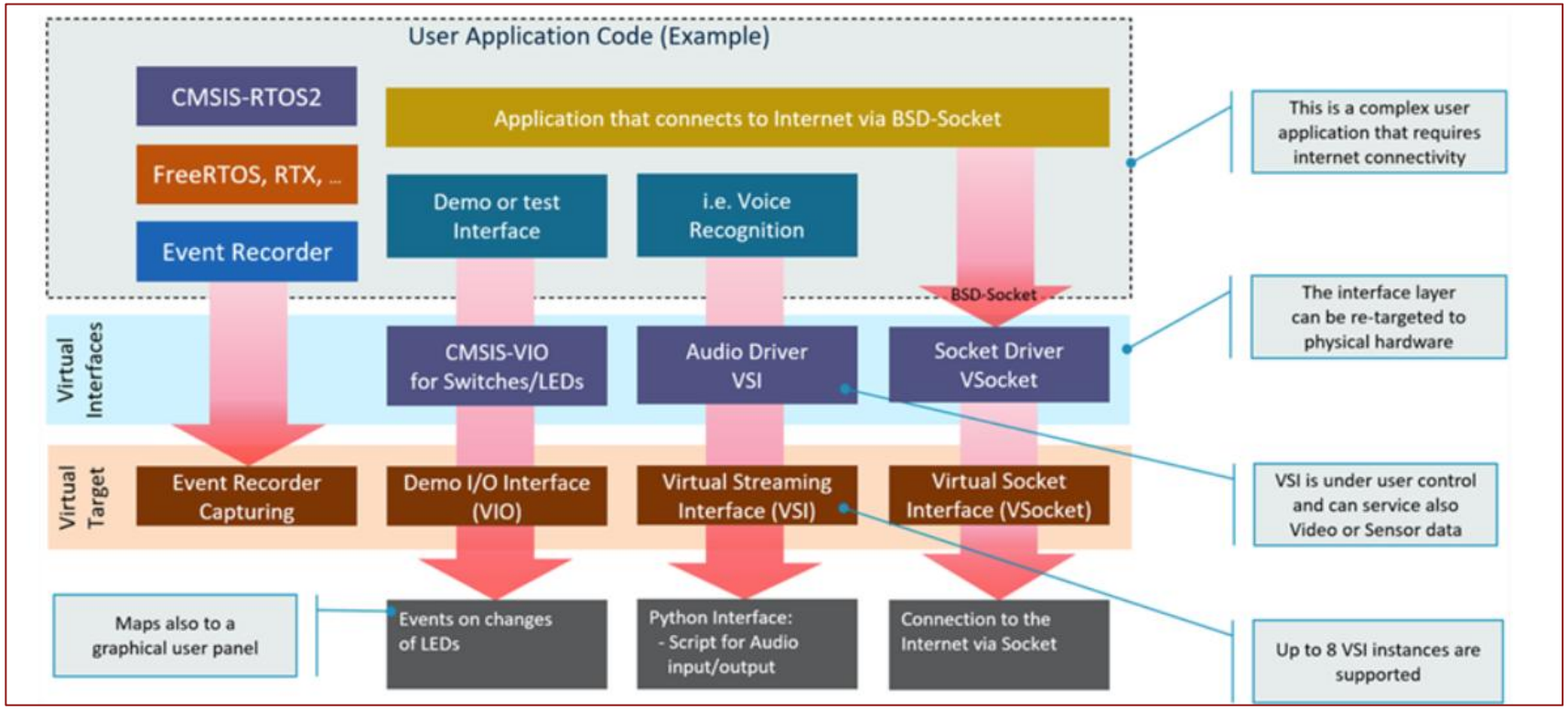
Add text cell

```
[ ]
model = keras.Sequential(
    [
        layers.Flatten(input_shape=[28,28]),
    #    layers.Dropout(0.2),
        layers.Dense(200, activation='relu'),
    #    layers.Dropout(0.2),
    #    layers.BatchNormalization(),
        layers.Dense(100, activation='relu'),
    #   layers.Dropout(0.2),
    #   layers.BatchNormalization(),
        layers.Dense(10, activation='softmax')
    ]
)
```

- Jupyter Books
  - Interactive Notebook
  - Combines Text, Code and visualisations
  - Typically runs in a Cloud based system
  - Can use VS-Code IDE in place of a browser
- Python Libraries
  - Keras
  - NumPy
  - Pandas
  - Scikit-Learn

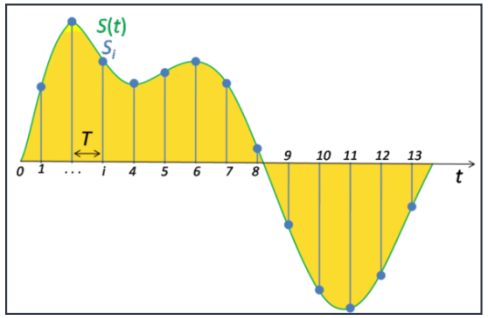
Fixed Virtual Platform

- Cortex-M CPU simulator
- Streaming abstraction layer
  - Playback and recording controlled by Python scripts



CMSIS DSP Library

- Collection of 61 DSP algorithms
- Uses Block Processing for efficient execution
- Python Wrapper for testing



- Basic Math Functions
  - Vector Multiplication
  - Vector Subtraction
  - Vector Addition
  - Vector Scale
  - Vector Shift
  - Vector Offset
  - Vector Negate
  - Vector Absolute
  - Vector Dot Product
- Fast Math Functions
  - Cosine
  - Sine
  - Square root of number
- Complex Math Functions
  - Complex conjugate
  - Complex dot product
  - Complex magnitude
  - Complex magnitude squared
  - Complex by complex multiplication
  - Complex by real multiplication
- Filters
  - Biquad Cascade IIR Filters Using Direct form I Structure
  - Finite Impulse Response (FIR) Filters
  - Convolution
  - Partial Convolution
  - Correlation
  - Finite Impulse Response (FIR) Decimation
  - Finite Impulse Response (FIR) Lattice Filters
  - Infinite Impulse Response (IIR) Lattice Filters
  - Biquad Cascade IIR 32x64 filter using Direct form I structure
  - Biquad Cascade IIR Filters Using a Direct form II Transposed Structure
  - Finite Impulse Response (FIR) Sparse Filters
  - Finite Impulse Response (FIR) Interpolation
  - Least Mean Square FIR Filter
  - Least Mean Square Normalized FIR Filter

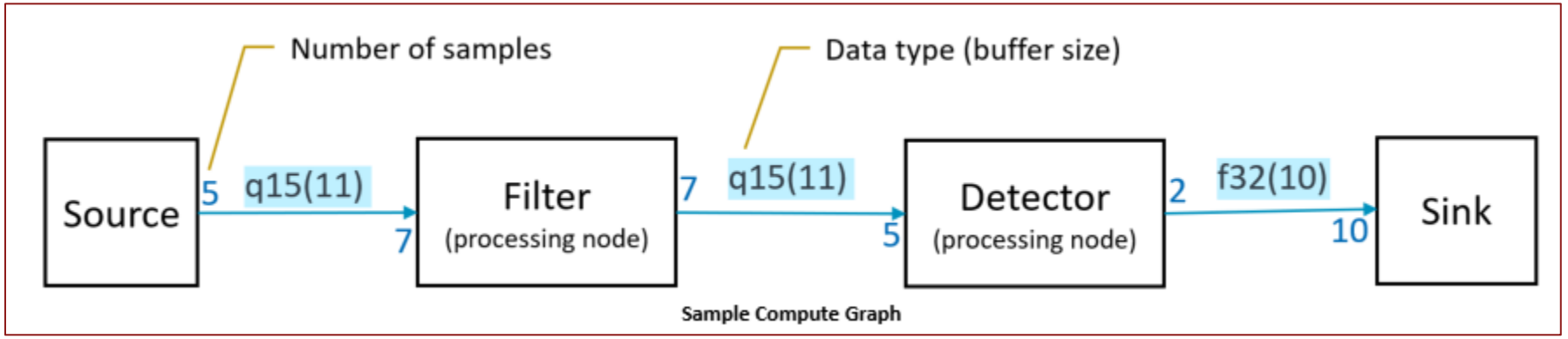
- Matrix Functions
  - Matrix Addition
  - Matrix Initialization
  - Matrix Scale
  - Matrix Subtraction
  - Matrix Multiplication
  - Matrix Inverse
  - Matrix Transpose
- Transforms
  - Complex FFT Functions
  - Real FFT Functions
  - DCT Type IV Function
- Controller Functions
  - SineCosine
  - PID Motor Control
  - Vector park transform
  - Vector Inversepark transform
  - Vector Clarke transform
  - Vector Inverse Clarke transform
- Statistical Functions
  - Power
  - Root mean square (RMS)
  - Standard deviation
  - Variance
  - Maximum
  - Minimum
  - Mean
- Support Functions
  - Vector Copy
  - Vector Fill
  - Convert 8-bit Interger value
  - Convert 16-bit Interger value
  - Convert 32-bit Interger value
  - Convert 32-bit floating point value
- Interpolator Functions
  - Linear Interpolate Function
  - Bilinear Interpolate Function

DSP and ML pipeline optimisation

CMSIS-Stream

- Uses Compute graph
- Scheduler for streaming data through a ‘graph’ of DSP algorithms

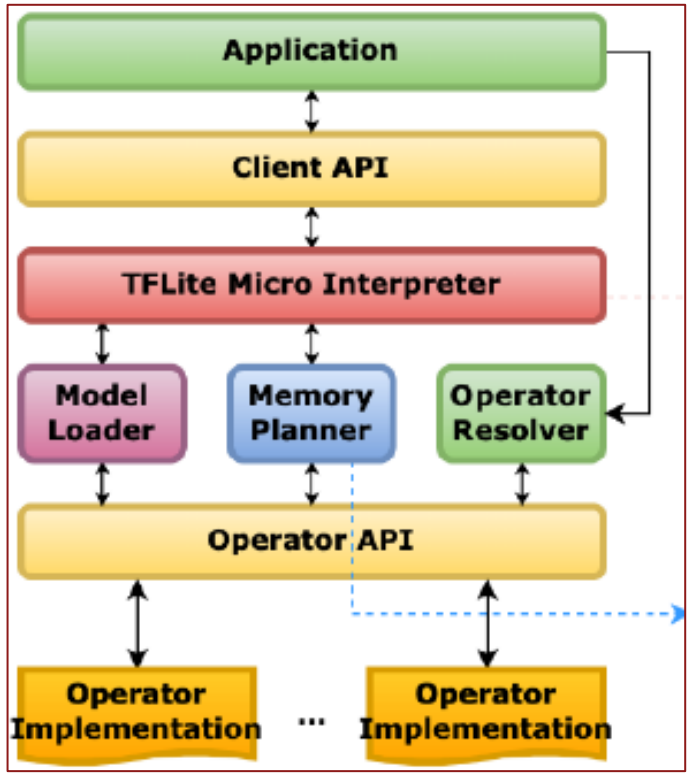
https://github.com/ARM-software/CMSIS-Stream



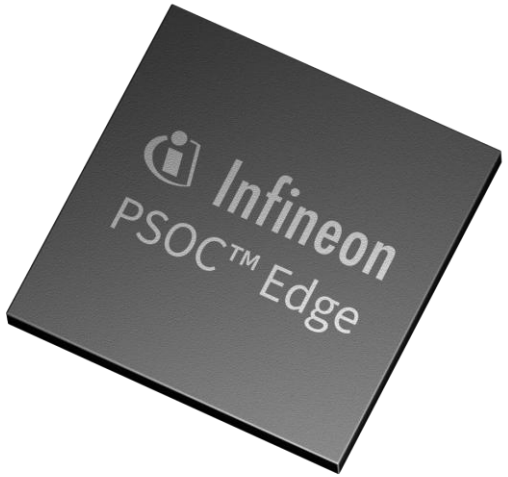
Sample Compute Graph

LiteRT Interpreter

- LiteRT (Tensor Flow)interpreter is installed as a C++ library on the microcontroller
- Loads the LiteRT model and runs an inference



Conclusion



- The Cortex-M ecosystem now provides:
  - Powerful processors with vector maths acceleration
  - Dedicated neural-network accelerators
  - Optimised DSP and Neural Net libraries
  - Cloud-enabled development environments
  - Simulation tools for rapid testing

Hitex offerings …

Ask alexander.hess@hitex.de – he looks forward to receiving your inquiries regarding Hitex offerings, including:

- Arm encryption libraries
- Development support and services
- Training courses for Arm, functional safety and security
- SO-DIMM compute modules for PSOC™, TRAVEO™ and more for an easy entry into the Infineon MCs.

Stay informed: webinars, training sessions, knowledge labs
Find all the information at www.hitex.de