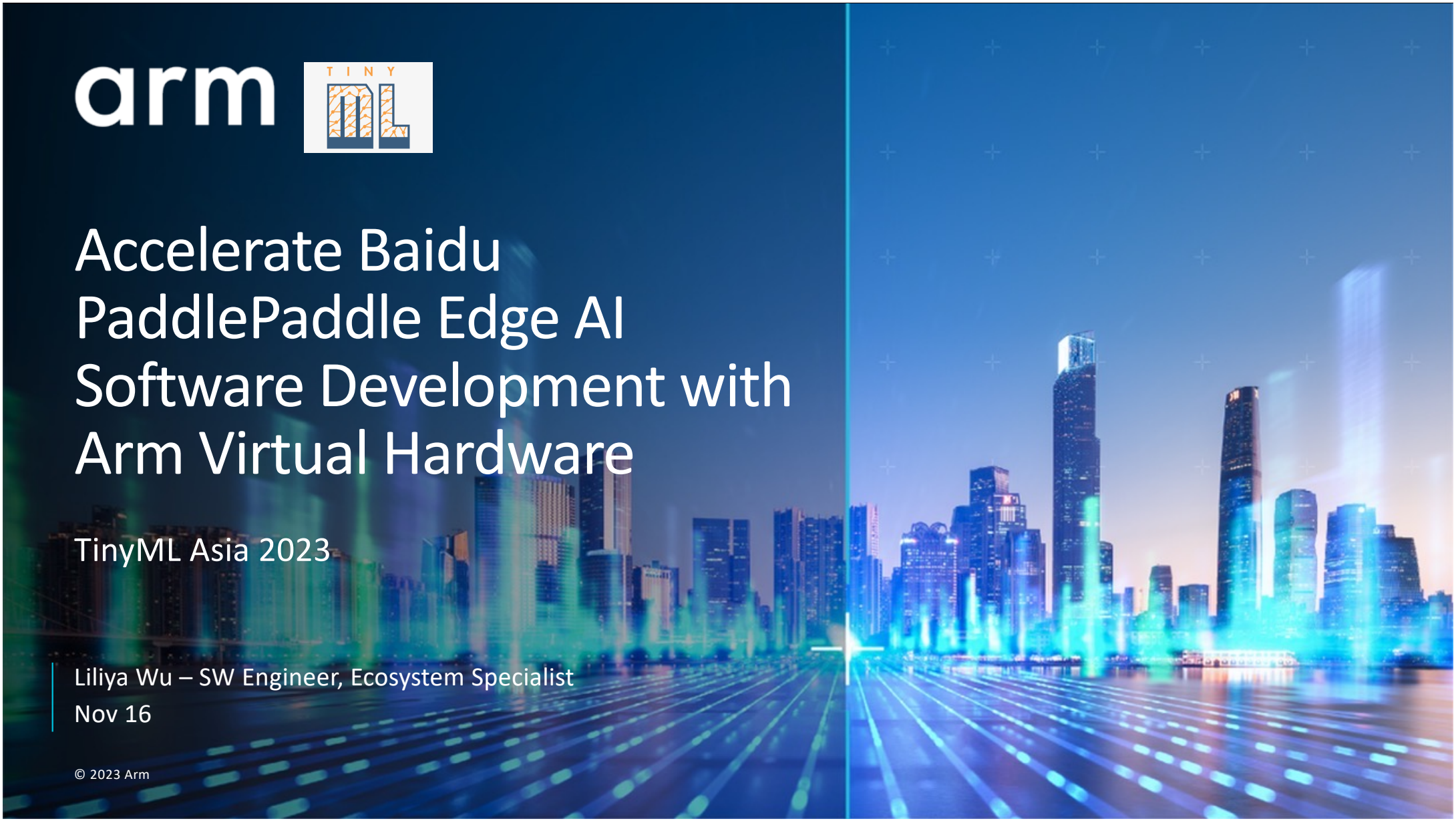


# Accelerate Baidu PaddlePaddle Edge AI Software Development with Arm Virtual Hardware

TinyML Asia 2023

Liliya Wu – SW Engineer, Ecosystem Specialist

Nov 16

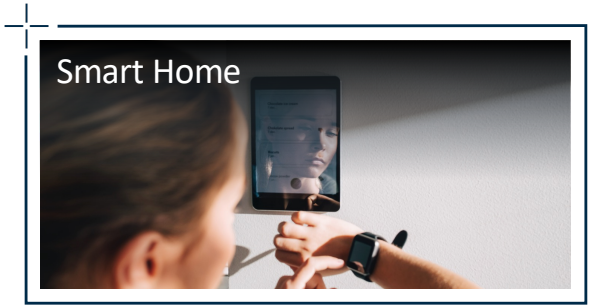
# Agenda

- Background
  - Evolution of compute at the edge
  - Key challenges and solutions
- Technology Overview
  - Arm solutions for Machine Learning
  - Paddle on Cortex-M
- MLOps: Develop & Test with Arm Virtual Hardware
  - Arm Virtual Hardware overview
  - Develop with Arm Virtual Hardware
  - Test with Arm Virtual Hardware
- Summary
  - Developer resources

# Background

The Future of ML Shifts to the Edge

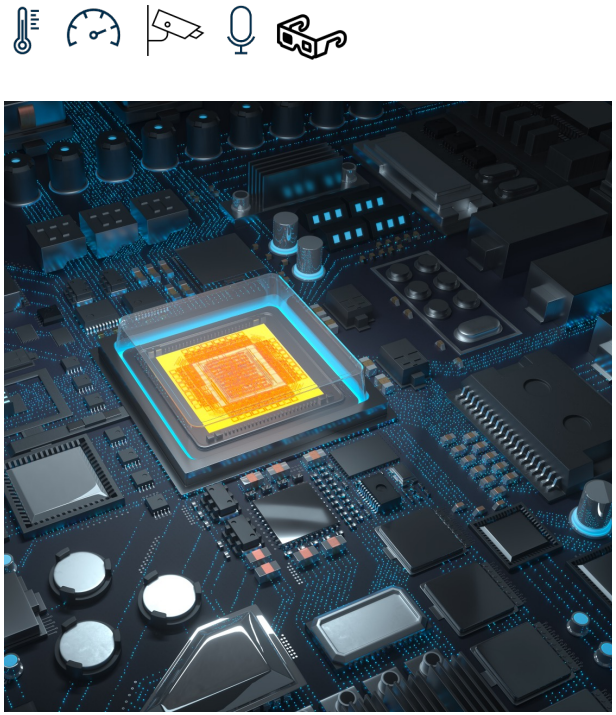
# Machine Learning is Being Deployed Everywhere



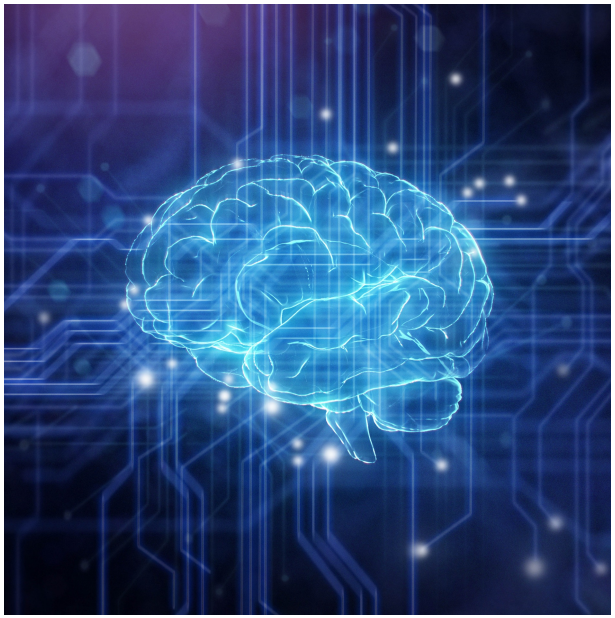
# Evolution of Compute at the Edge

- Embedded

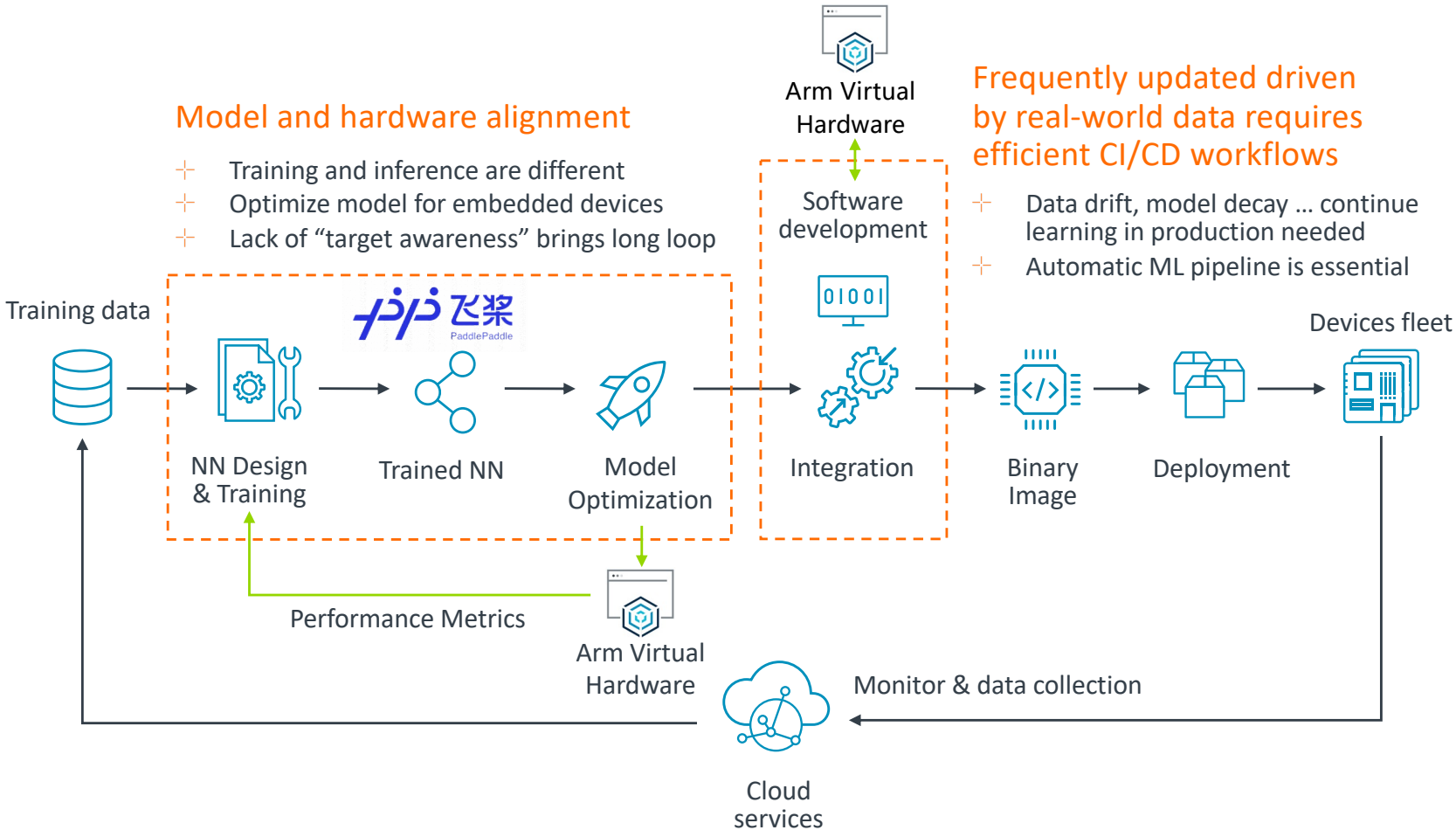


- Connectivity

- Machine Learning



# Key Challenges and Solutions



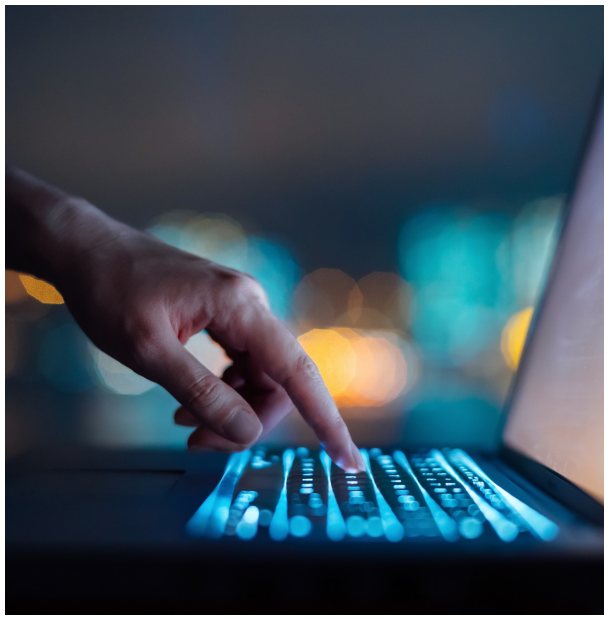
# Technology Overview

Unlocking TinyML Use-cases with
High-performance Cortex-M

# Arm is Enabling Machine Learning within Three Focus Areas

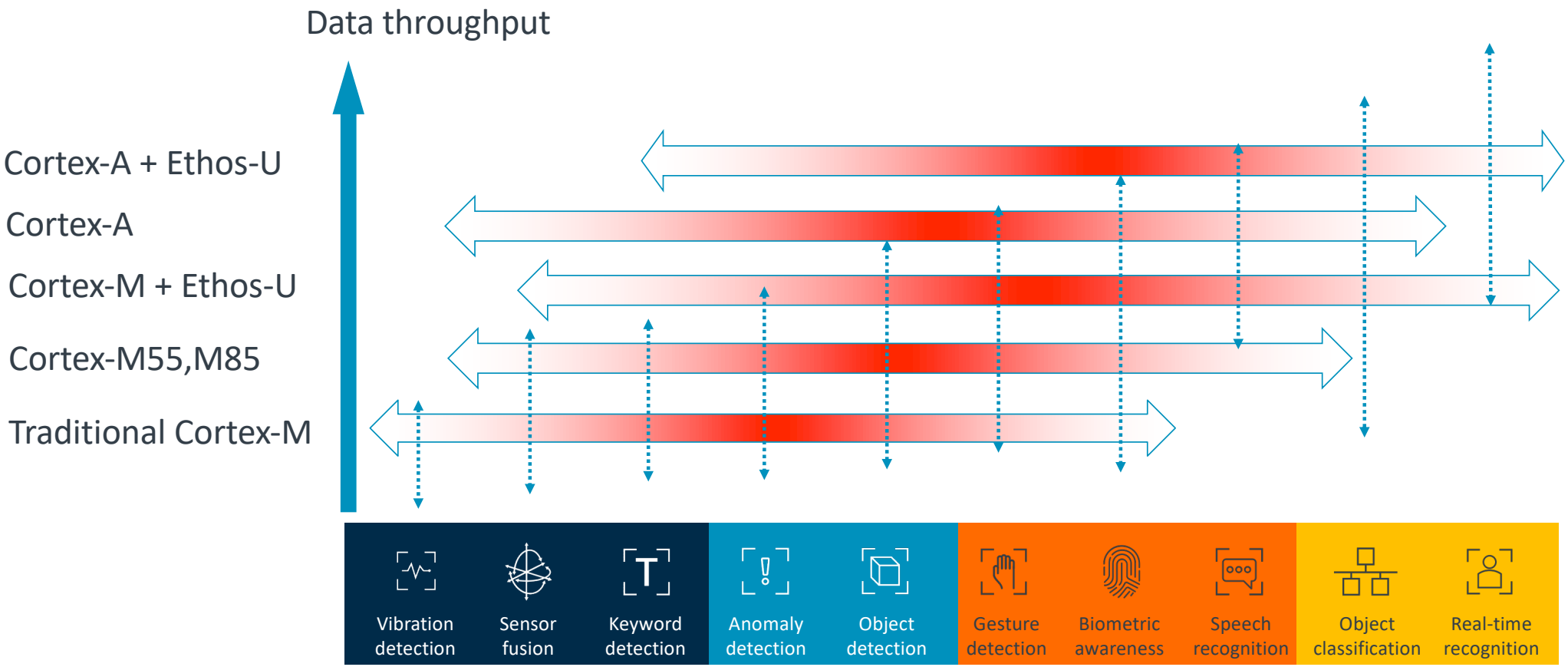
## Hardware IP Portfolio

## Standards, Software and Tools



## Developer Experience

# Broadest Range of ML-optimized Processing Solutions



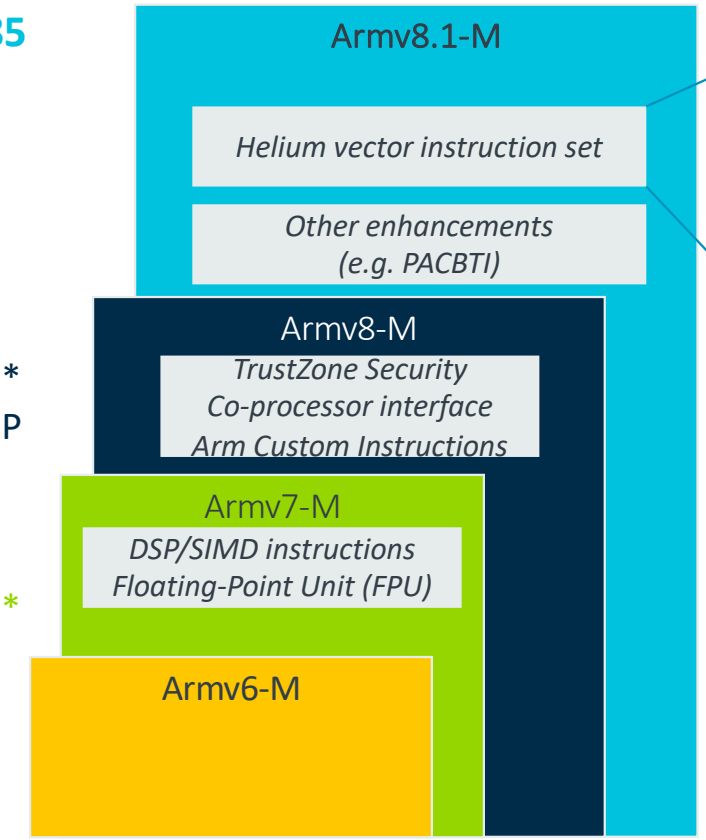
# Cortex-M Processor Portfolio – Instruction Set Evolution

**Cortex-M85**
Cortex-M55

Cortex-M33
Cortex-M23*
Cortex-M35P

Cortex-M7
Cortex-M4
Cortex-M3**

Cortex-M0+
Cortex-M0




>150 new scalar and vector instruction
Low overhead loops
Predication
Arithmetic support for 8-bit fixed and 16-bit float
Gather load, scatter store
Complex math

→ **Ease of development**
→ **Energy-efficient compute**
→ **Increased throughput**
→ **Smarter devices**

* Cortex-M23 does not include co-processor interface, ACI, DSP/SIMD and FPU

** Cortex-M3 does not include DSP/SIMD and FPU




arm

# Provides The Best Experience to Develop ML on arm

| | |
|---|---|
| ML Frameworks | TensorFlow Lite, Baidu 百度, 飞桨, PyTorch, ONNX, ... |
| Standards | TOSA |
| Toolchain | tvm, µtvm, Ethos-U Vela NN Optimizer, arm NN |
| Libraries | CMSIS, CMSIS-NN, arm COMPUTE LIBRARY |

Find optimized models for target platform/use cases

Software Toolkit

- mlia
- arm ML-zoo
- ML embedded evaluation kit

Deploy ML application corss arm plaform easily & safely

Hardware & Software Standards

- arm SystemReady IR
- psacertified™
- CMSIS

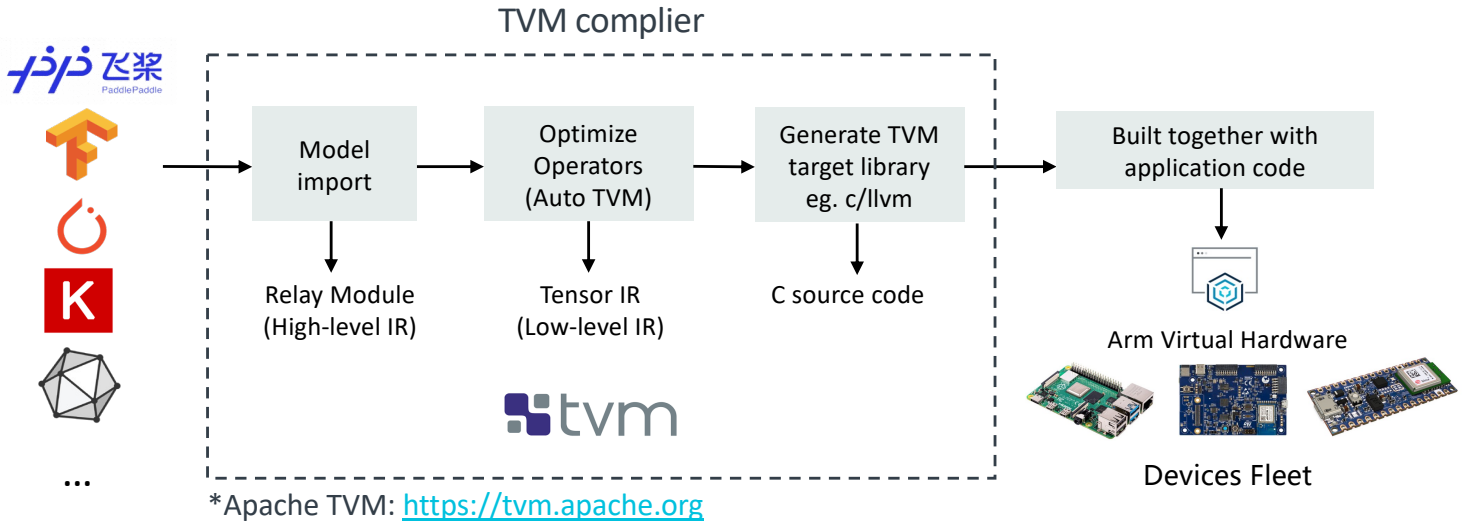
Development Tools

- KEIL™ Tools by ARM
- arm Virtual Hardware
- arm IP Explorer

Enable efficient end-to-end MLOps workflow

# Paddle on Cortex-M

- Challenges for leveraging PaddlePaddle to run NNs on Cortex-M – Today
  - No direct runtime interpreter support on devices (PaddlePaddle, Paddle Inference, Paddle Lite, etc.).
  - Model conversion to TFL is not always efficient.
  - … no obvious software stack “gap” but still complex and requires significant specialist skills.
- TVM Code Generation Technology for the Arm AI Platform
  - TVM is an open deep learning compiler stack that closes the gap between the productivity-focused deep learning frameworks like PaddlePaddle, and the performance-oriented or efficiency-oriented hardware back-ends like CMSIS-NN.
  - MicroTVM runs TVM models on bare-metal (such as IoT) devices




*Apache TVM: https://tvm.apache.org

# Compilation for Each Device and Framework

## TVM Project Provides the Nexus Between Important Frameworks and Back-ends

- PaddlePaddle as the front end – officially supported in TVM v8.0 releasement!
  - Support 120+ operators and 100+ models.
  - Plan to support 200+ operators and models quantized by PaddleSlim in the future.
- CMSIS-NN as the backend – Use CMSIS-NN with TVM (RFC)
  - TVM allows for partitioning and code generation using an external compiler.
  - Partitioned subgraphs containing operators targeted to Cortex-M can then be translated into the CMSIS NN C APIs.
- Example

```
tvmc compile ocr_en/inference.pdmodel\
    --target=cmsis-nn,c \
    --target-cmsis-nn-mcpu=cortex-m55 \
    --target-c-mcpu=cortex-m55 \
    --runtime=crt \
    --executor=aot \
    --executor-aot-interface-api=c \
    --executor-aot-unpacked-api=1 \
    --pass-config tir.usmp.enable=1 \
    --pass-config tir.usmp.algorithm=hill_climb \
    --pass-config tir.disable_storage_rewrite=1 \
    --pass-config tir.disable_vectorize=1 \
    --output-format=mlf \
    --model-format=paddle \
    --module-name=rec \
    --input-shapes x:[1,3,32,320] \
    --output=rec.tar
```

* Compile ppocr-v3 English text recongition model for Cortex-M55 processor with TVMC.

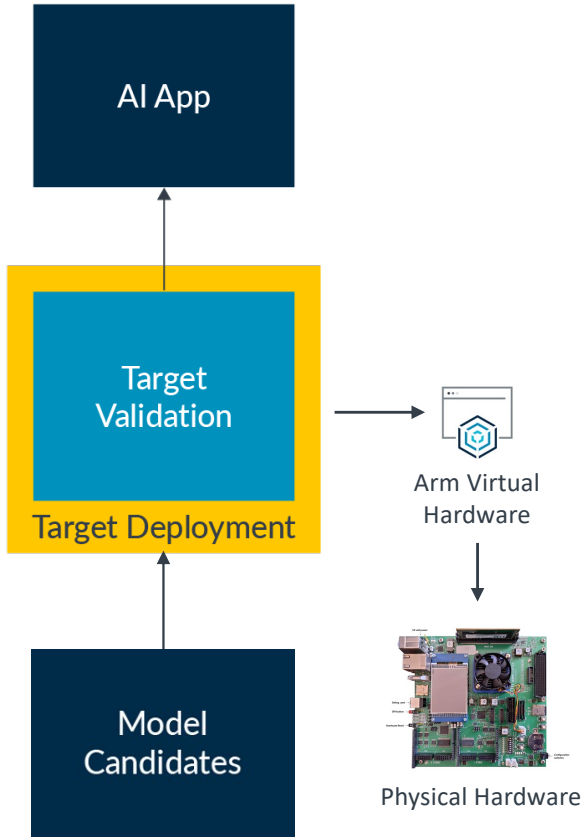


arm

# MLOps:
# Develop & Test with
# Arm Virtual Hardware

# Benefits with Virtual Hardware for MLOps

- Speed
  - Friendly to traditional ML engineers/ data scientists/ software engineers(eg. Paddle developers) to have “hardware awareness” without much extra efforts to master embedded skills.
  - Virtual hardware have no overhead for flashing the application on physical hardware. Verify on-device inference results efficiently.
- Scale
  - Test algorithm across multiple target devices and operating systems without purchasing and debugging additional hardware Enable building various models (eg. Paddle) on various Arm processors easily.
  - Virtual hardware can scale to run many tests in parallel - this makes virtual platforms more cost-effective than a farm of physical hardware.
- Maintenance
  - Unlike physical hardware, virtual hardware do not overheat, wear out from overuse, break from misuse, or use physical space and resources. Repeated ML cycles cause no loss to virtual boards.



# Arm Virtual Hardware (AVH)

Consists of two different virtual hardware products to meet different application demands

Arm Virtual Hardware Corstone and CPUs

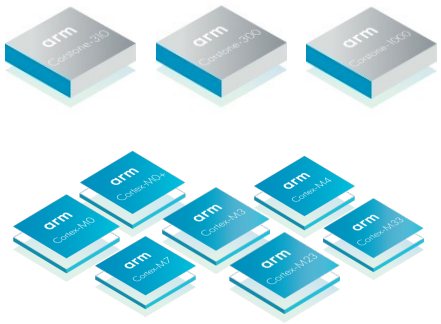
Arm Virtual Hardware Third-Party Hardware

In the cloud

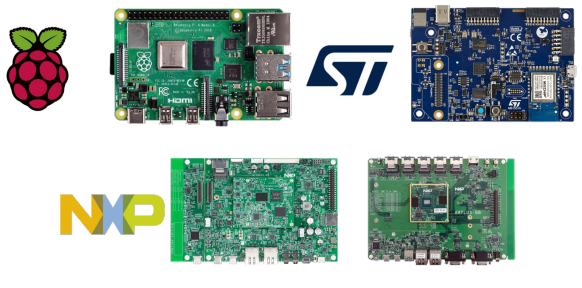
Virtual Hardware Models

Corstone

CPU



Development Boards



Core Modeling Technology

Fixed Virtual Platforms (FVPs) based on Arm Fast Model technology

Hypervisor Technology

Infrastructure & Host OS

Arm & x86 (ubuntu)

Arm (bare-metal)

Arm based processors and systems

Popular IoT development kits

Develop Software with Confidence

Achieve Maximum Scalability

Accelerate Development

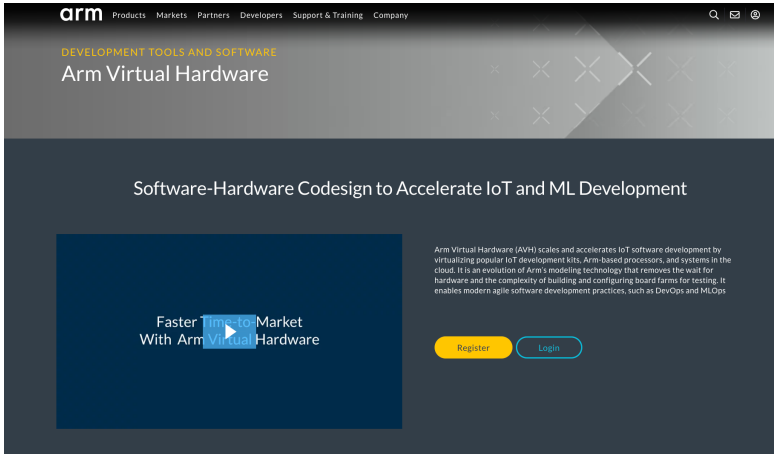
Features and Benefits

# Arm Virtual Hardware Corstone and CPUs

Visit https://avh.arm.com for more details or search “Arm Virtual Hardware” at arm.com

Arm Virtual Hardware Corstone and CPUs

| | | |
|---|---|---|
| Virtual Hardware Models | Corstone |  |
| | CPU |  |
| Core Modeling Technology | Fixed Virtual Platforms (FVPs) based on Arm Fast Model technology | |
| Infrastructure & Host OS | Arm & x86 (ubuntu) | |
| | Arm based processors and systems | |



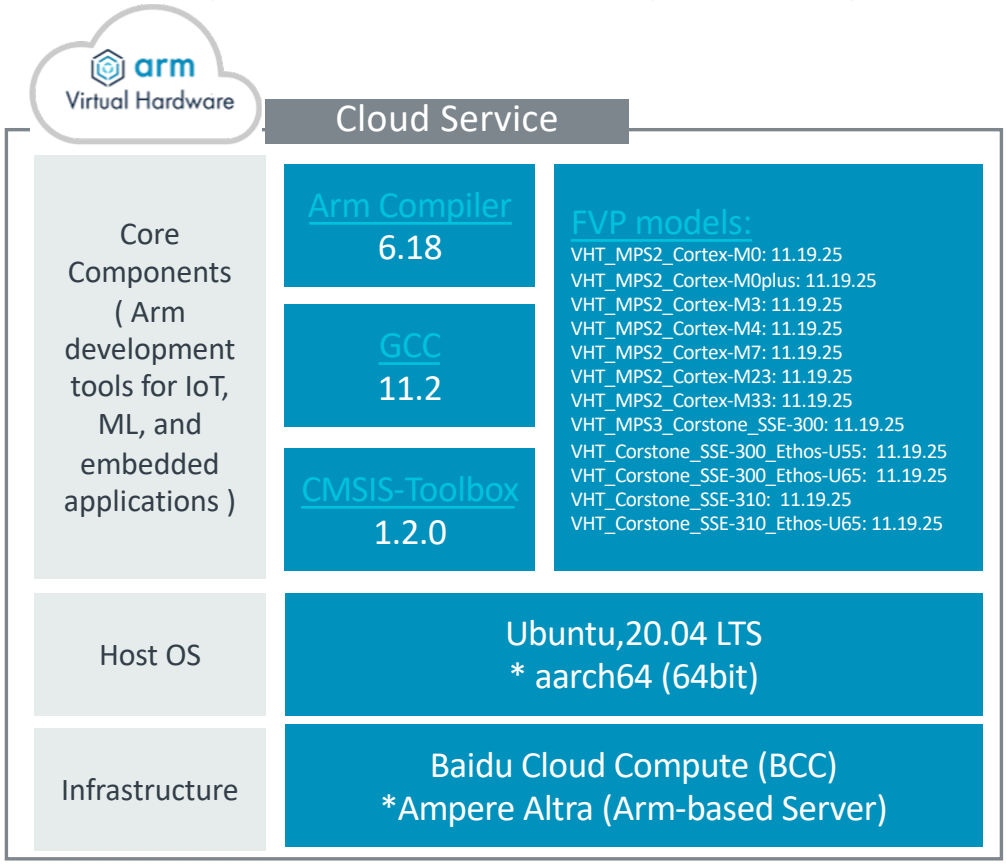

## Designed for:

- Software validation and MLOps workflows for IoT, ML, and embedded applications.
- CI workflows for unit and integration testing of software modules with abstracted I/O interfaces.
- Exploration of reference IoT software and stacks.
- Training and education.

# Arm Virtual Hardware Corstone and CPUs

Avaiable through multiple ways to better match developers' various development style

- Devliered as "Image Service Product" on Baidu Cloud Market (Beta)
- Including Arm development tools for IoT, Machine learning, and embedded applications.
  - Compiler: Arm Compiler for Embedded, GNU C/C++ Compiler (Arm GNU Toolchain).
  - Command-line tools for Open-CMSIS-Pack project: CMSIS-Toolbox.
  - Virtual Hardware target: Arm Cortex-M based reference platforms (FVP models)
- Other available methods:
  - Amazon Machine Image(AMI) at AWS Marketplace
  - GitHub Runner
  - Keil-MDK Professional
  - Arm's SaaS platform as a public beta - Register
  - MLOps system container – tutorial
  - Arm Partner solutions – Arm Partner Ecosystem Catalog

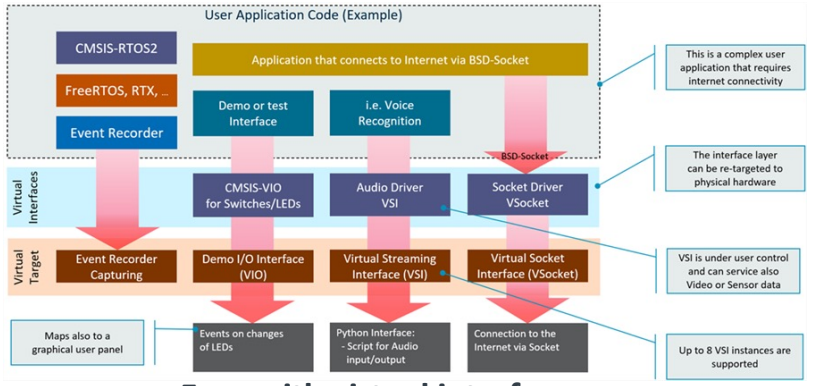



* Product document: https://arm-software.github.io/AVH/main/overview/html/index.html

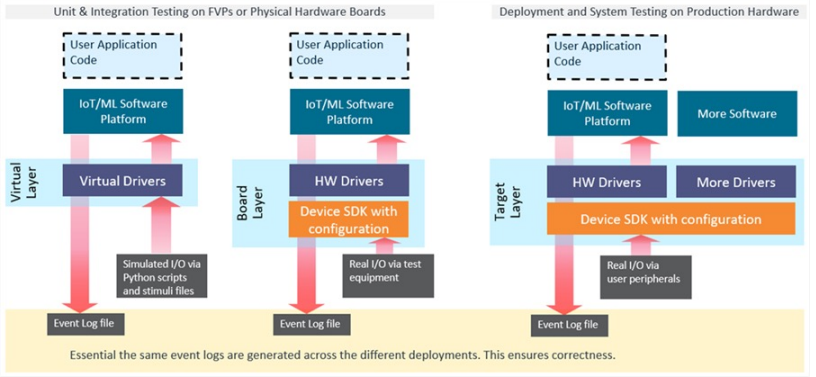
# Arm Virtual Hardware Fixed Virtual Platforms (FVPs)

## Use the Arm Fixed Virtual Platforms (FVPs) in Arm Virtual Hardware context

- Fixed Virtual Platforms (FVPs)
  - Based on Arm Fast Models technology and developed alongside Arm's processor IP.
  - Complete simulations of an Arm system, including processor, memory and peripherals.
  - The FVP models in AVH:
    - Subset of Arm FVPs providing precise simulation models of Cortex-M based sub-systems (such as Corstone-310, Corstone-300 ).
    - Delivered as pre-built executables in the cloud. The composition is fixed but can configure their behaviors using command line(CLI) parameters.
    - With extensions for Virtual Interfaces including:
      - Virtual Input/Output (VIO)
      - Virtual Streaming Interface (VSI)
      - Virtual Socket Interface (VSocket)




**Fvps with virtual interfaces**

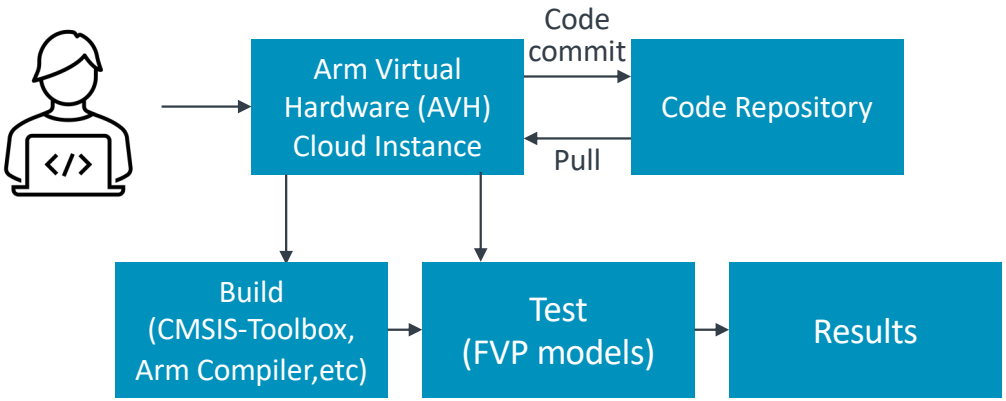



**Re-target from Simulation to Hardware**




* Available models and their compositions: https://arm-software.github.io/AVH/main/simulation/html/Using.html

# Develop in the Cloud

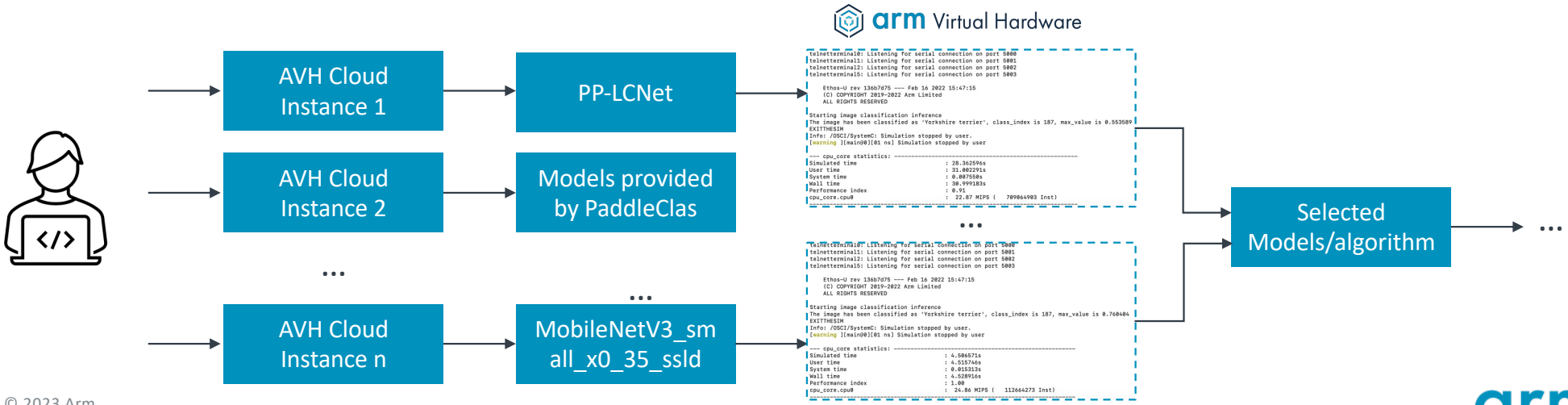



* Launch an FVP from the command line and configure its behaviors

```
ubuntu@avh-bd-1:~/Paddle-usecase-test/OCR-example$ VHT_Corstone_SSE-300_Ethos-U55 -C cpu0.CFGDTCMSZ=15 \
> -C cpu0.CFGITCMSZ=15 -C mps3_board.uart0.out_file=\"-\" -C mps3_board.uart0.shutdown_tag=\"EXITTHESIM\" \
> -C mps3_board.visualisation.disable-visualisation=1 -C mps3_board.telnetterminal0.start_telnet=0 \
> -C mps3_board.telnetterminal1.start_telnet=0 -C mps3_board.telnetterminal2.start_telnet=0 -C mps3_board.telnetterminal5.start_telnet=0 \
> ./build/demo --stat
telnetterminal0: Listening for serial connection on port 5000
telnetterminal1: Listening for serial connection on port 5001
telnetterminal2: Listening for serial connection on port 5002
telnetterminal5: Listening for serial connection on port 5003

    Ethos-U rev 136b7d75 --- Feb 16 2022 15:49:32
    (C) COPYRIGHT 2019-2022 Arm Limited
    ALL RIGHTS RESERVED

Starting ocr rec inference
text: QBHOUSE, score: 0.986746
EXITTHESIM
Info: /OSCI/SystemC: Simulation stopped by user.
[warning ][main@0][01 ns] Simulation stopped by user

--- cpu_core statistics: -------------------------------------------------------
Simulated time                          : 76.450841s
User time                               : 126.140433s
System time                             : 0.031346s
Wall time                               : 126.273550s
Performance index                       : 0.61
cpu_core.cpu0                           :  15.15 MIPS (  1911271056 Inst)
--------------------------------------------------------------------------------
```


*Image source: ICDAR 2015

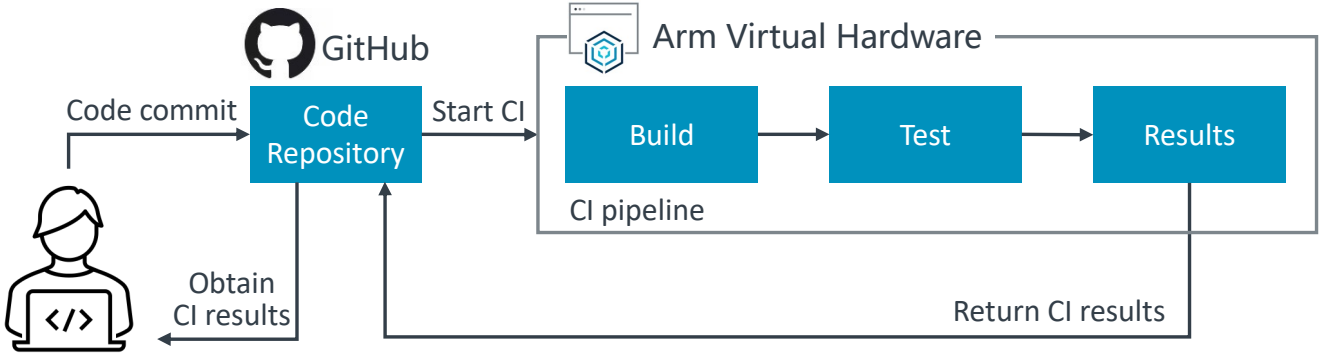
*https://github.com/ArmDeveloperEcosystem/Paddle-examples-for-AVH



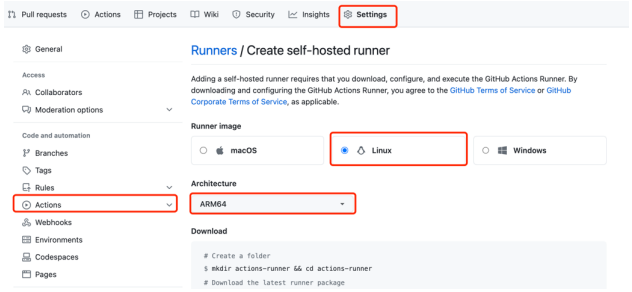

*Execute on Arm Corstone-300 FVP with Cortex-M55

# CI/CD and DevOps

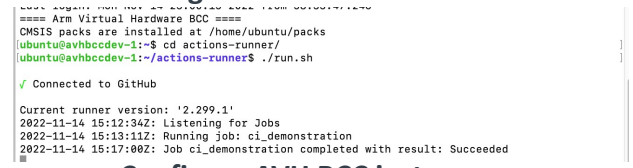
- With GitHub Actions as an example,
  - Configure Arm Virtual Hardware as your self-hosted runner.
  - Use GitHub hosted runner and connect with Arm Virtual Hardware using Arm Virtual Hardware Client (avhclient). avhclient enables uniform implementation of CI operations in various environments.
  - Use GitHub Arm Virtual Hardware runner – Register private beta!
- Examples
  - Runner: Self-hosted (BCC) & GitHub-hosted via avhclient（AMI）
  - Details of individual steps to be executed on AVH is defined in avh.yml



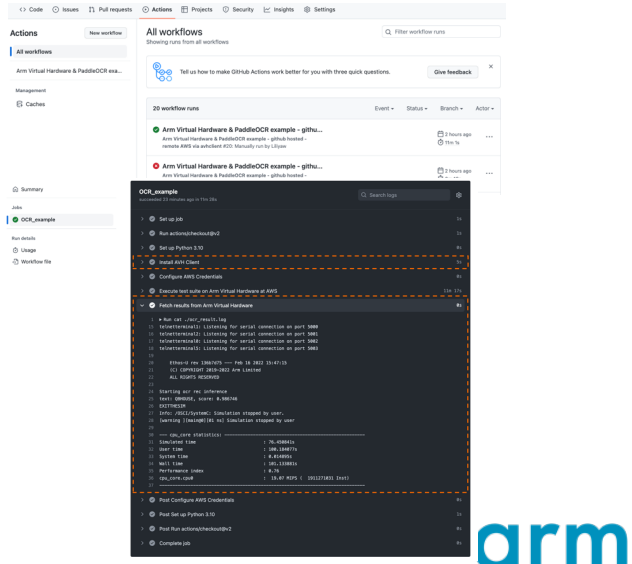

**Continuous Integration (CI) workflow**




**Configure GitHub Actions**




**Configure AVH BCC instance**




**GitHub Action Dashboard**


# Summary

# User Benefits and Ecosystem

## Find More in your Own User Experience — Try it Today!

### Test without Hardware

**Access to the latest Arm processing IP**

- Early software development for faster time-to-market
- Select optimal target device once the software workload is analysed
- Re-target applications to production hardware with driver abstractions

### Verify Correctness

**Integration with CI/CD tools**

- Perform algorithm testing with identical logical behavior of the target device
- Precisely repeat complex input patterns in CI/CD test environments
- Analyse software behavior with event annotations

### Evaluate Performance

**Integration in NAS & AutoML platform**

- Compare speed of different implementations of an algorithm
- Identify timing issues during system integration
- Optimize resources (i.e. data buffers) towards application requirements

New Arm Virtual Hardware Integrations

Arm is dedicated to making the advantages of Arm Virtual Hardware available wherever developers do their work, which is why we continue to support integrations of Arm Virtual Hardware into the offerings of leading cloud-based development workflow providers.

Read about the latest native integrations with Arm Virtual Hardware with GitHub Actions, Qeexo and Nota AI.

Arm Virtual Hardware Service and Partner Integrations

The ecosystem for AVH is growing rapidly. Partners are integrating AVH into service offerings. Explore the leading software companies who are leveraging AVH to change IoT development.

Qeexo GitHub GitLab circleci

Jenkins Nota AI ORACLE remote.it

[Contact us](#) to become part of the Arm Virtual Hardware ecosystem!

# Developer Resources

- Open-source software stacks for Cortex-M based ML applications
  - CMSIS
    - CMSIS NN – Neural network kernels optimized for Arm
    - CMSIS DSP – Compute library for embedded systems; includes compute graph for efficient data streaming between different algorithms
  - Arm ML Embedded Evaluation Kit – Build and deploy ML applications targeted for Corstone-300/310
  - Synchronous Data Streaming (SDS) framework - Simplify Development of Embedded Applications that utilize DSP or ML algorithms with Sensor/Audio Input
- Example projects
  - Paddle Examples for AVH – Co-launch model zoo with more use cases based on PaddlePaddle models
  - Arm Virtual Hardware developer resources on GitHub – broader usage examples from ML to IoT, DevOps
- Websites, blogs and courses
  - Courses: Arm Tech Talks
  - IoT Solutions at Arm - Arm is the Company, Technology and Unifying Force Behind the IoT Revolution
  - Blogs: The future of ML shifts to the edge, New Arm Virtual Hardware Integrations
  - Tutorial:
    - How to Deploy PaddlePaddle on Arm Cortex-M with Arm Virtual Hardware
    - ML Developers Guide for Arm Cortex-M Processors and Ethos-U NPUs

arm

Thank You
Danke
Gracias
Grazie
谢谢
ありがとう
Asante
Merci
감사합니다
धन्यवाद
Kiitos
شكرًا
ধন্যবাদ
תודה

TINY ML ASIA
Seoul, South Korea
November 16, 2023

# Copyright Notice



**www.tinyml.org**