CMSIS Review Meeting

Simplify the distribution and code re-use of software components for microcontrollers

CMSIS Team

22/6/2021

Agenda

- What’s new in CMSIS 5.8.0?
- Open-CMSIS-Pack: new incubation initiative under Linaro's IoT and Embedded Group
- Keil Studio Cloud: browser-based IDE that uses Open-CMSIS-Pack technologies
- CMSIS-DAP v2.x with speed improvements and further enhancements
- Enhanced Pack submission process that leverages Open-CMSIS-Pack
- CMSIS-DSP and CMSIS-NN update
- TinyML development with CMSIS-Pack and cloud-based workflows
- Summary and Questions: how to contribute

CMSIS v5.8.0 - 30 June 2021

Major changes since 5.7.0

- CMSIS-Core(M): 5.5.0
  - Updated GCC LinkerDescription, GCC Assembler startup
  - Added ARMv8-M Stack Sealing
  - C-Startup is now the default
  - Armv8-M Assembler startup uses GAS syntax
- CMSIS-DSP: 1.9.0
  - Purged pre-built libs from Git
  - Enhanced support for f16 datatype
- CMSIS-DAP: 2.1.0
  - UART support
  - Board Identification
- CMSIS-Pack: 1.7.2
  - Support for Microchip XC32 compiler
  - Support for Custom Datapath Extension
- CMSIS-NN: 3.0.0
  - Major interface change for functions compatible with TFLu
  - Added optimization for SVDF kernel
  - Improved MVE performance for fully Connected and max pool operator
  - NULL bias support for fully connected operator in non-MVE case
  - Expanded existing unit test suite along with support for FVP (simulation models)
- CMSIS-RTOS2:
  - RTX 5.5.3
  - CVE-2021-27431 vulnerability mitigation.
  - Enhanced stack overrun checking.

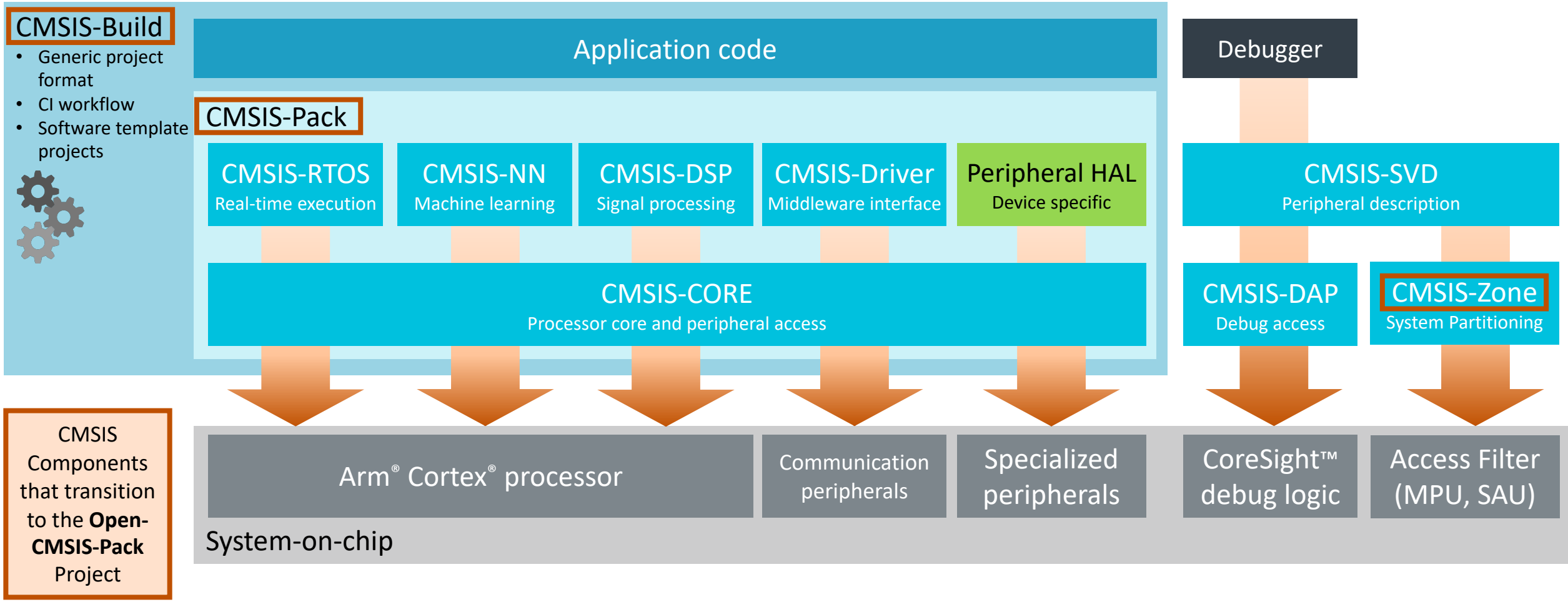
arm

Overview

Reinhard Keil, Senior Director MCU Tools

CMSIS Overview of today's components

Consistent software framework for Arm Cortex-M and Cortex-A5/A7/A9 based systems



More Information: Which CMSIS components should I care about?

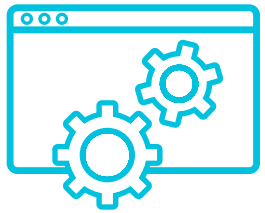
IoT Application Development - Workflows

Version Control

Cloud Storage

Repository hosting service that typically includes access control and a number of collaboration features.

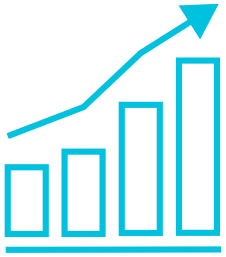
Software Development



Software as a Service (SaaS)

Instead of installing the IDE and software tools on your local device, you access the setup of the cloud provider.

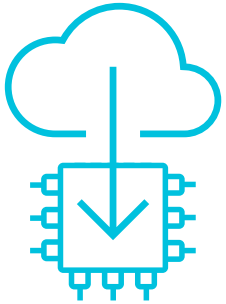
Continuous Testing



Virtual Machine (VM)

A “server” running in the cloud contains a tool environment with simulation models and settings specific to your project.

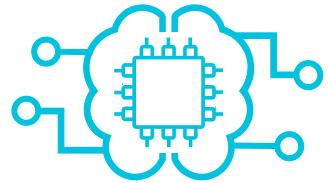
Software Deployment



Geographic Distribution

Over-the-air (OTA) programming offers methods to provision and update software of devices that are already in the field.

Machine Learning (ML)



Data Analytics

Monitor devices to spot anomalies and collect training data for ML algorithms that can be deployed to IoT endpoints.

[White Paper: Get More Productivity with Cloud Services]

arm

Open-CMSIS Pack

Incubation initiative under Linaro's IoT and Embedded Group

CMSIS-Pack: A success story

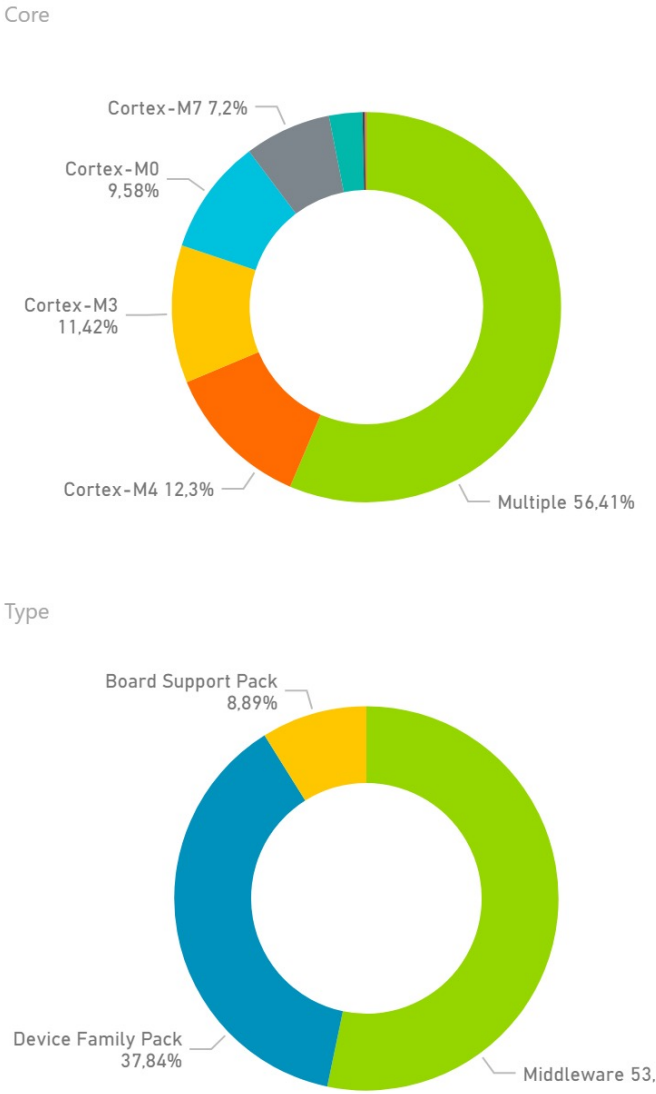
Public index lists packs from more than 60 vendors

More than 730 different packs available publicly

Close to 9,000 devices supported by 40 different silicon vendors

More than 2,000,000 manual pack downloads in MDK during past 12 months

Supported by several major IDEs (including an open-source implementation for Eclipse)

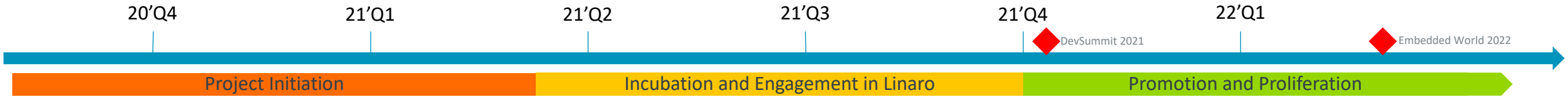


Establishing the Open-CMSIS-Pack Project

Broadening the engagement

- Componentized software distribution was one of the central elements of an earlier initiative
- Several partners engaged strongly with this concept and have actively participated
- The group has completed a requirements review and now has an aligned view of needs
- CMSIS-Pack is agreed as the project starting point, along with Arm open-sourcing some of the currently closed source code

- This led to an independent community project known as **Open-CMSIS-Pack**
- Target is to develop a minimal set of open-source tools to create/manage/compose/build based on CMSIS-Packs
- Linaro's IoT and Embedded Group provides a convenient incubator for the technology to build these tools



- Discussed project in working groups
- Transition to Linaro's IoT and Embedded Group
- Open-Source components and PoC Tools

Open-CMSIS-Pack www.open-cmsis-pack.org

Roadmap

- Create command-line tools for project build based on software packs
- Create workflows and utilities for the verification of software packs
- Extend the pack description format for better usability across the complete workflow
- Define processes that simplify the creation of software packs from other sources, such as CMake based projects
- Develop the concept of a software layer that defines a collection of pre-configured software components
- Organize the taxonomies of standard APIs that are essential for re-useable software stack

Founding Members

Technical Project Meetings

- Tuesdays 16:00 (CET)
- Mailing list: https://op-lists.linaro.org/mailman/listinfo/open-cmsis-pack-dev

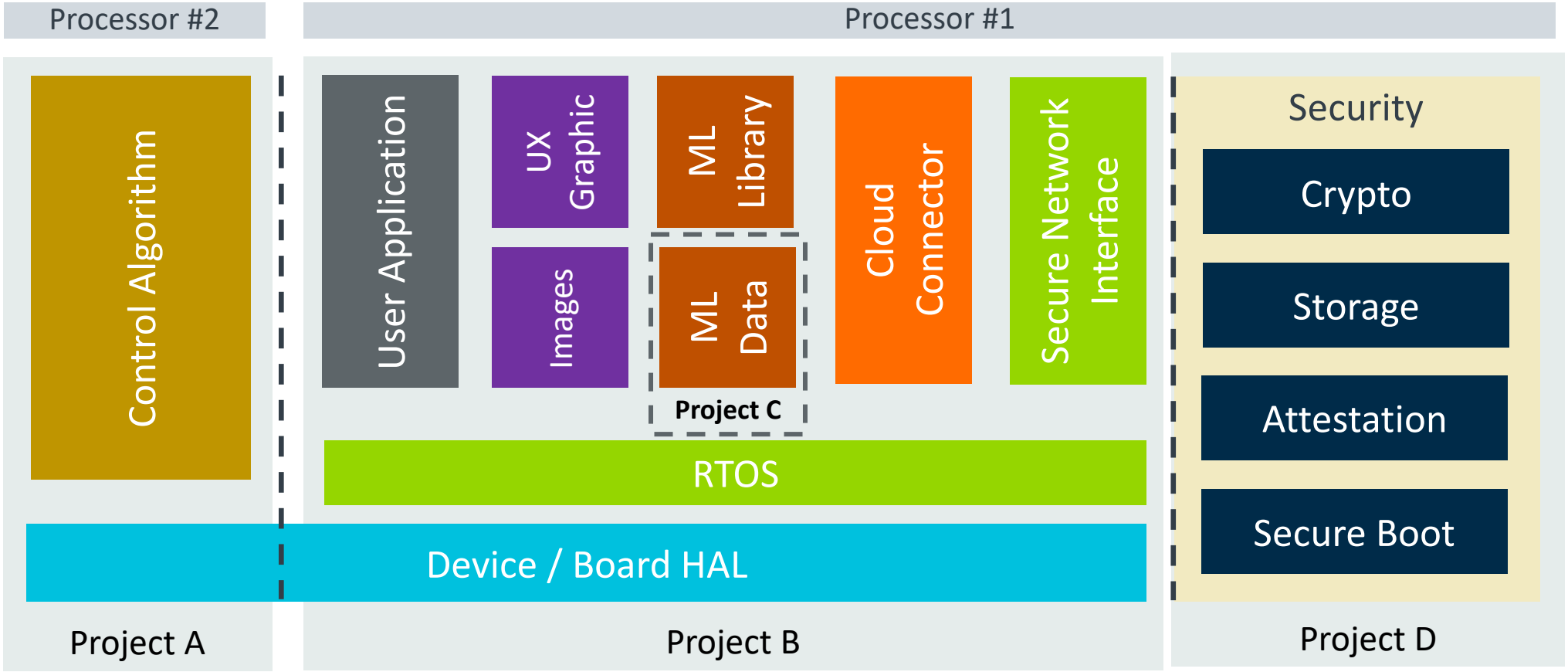
What Requirements should we consider for the future?

Use cases driven by Application Developer

- Holistic view on software projects considering:
  - Structure
    - many dependent/related projects
    - reuse of partial projects
  - Code Generation:
    - build order dependencies
    - multiple build configurations
    - HW resource allocation partitioning and dependencies
    - generated/assisted software configuration
  - Deployment and Download:
    - flash programming setup and configuration
    - Firmware update processes including OTA programming
  - Debugging:
    - debug setup and configuration
- Simplify testing and porting of applications across devices and boards

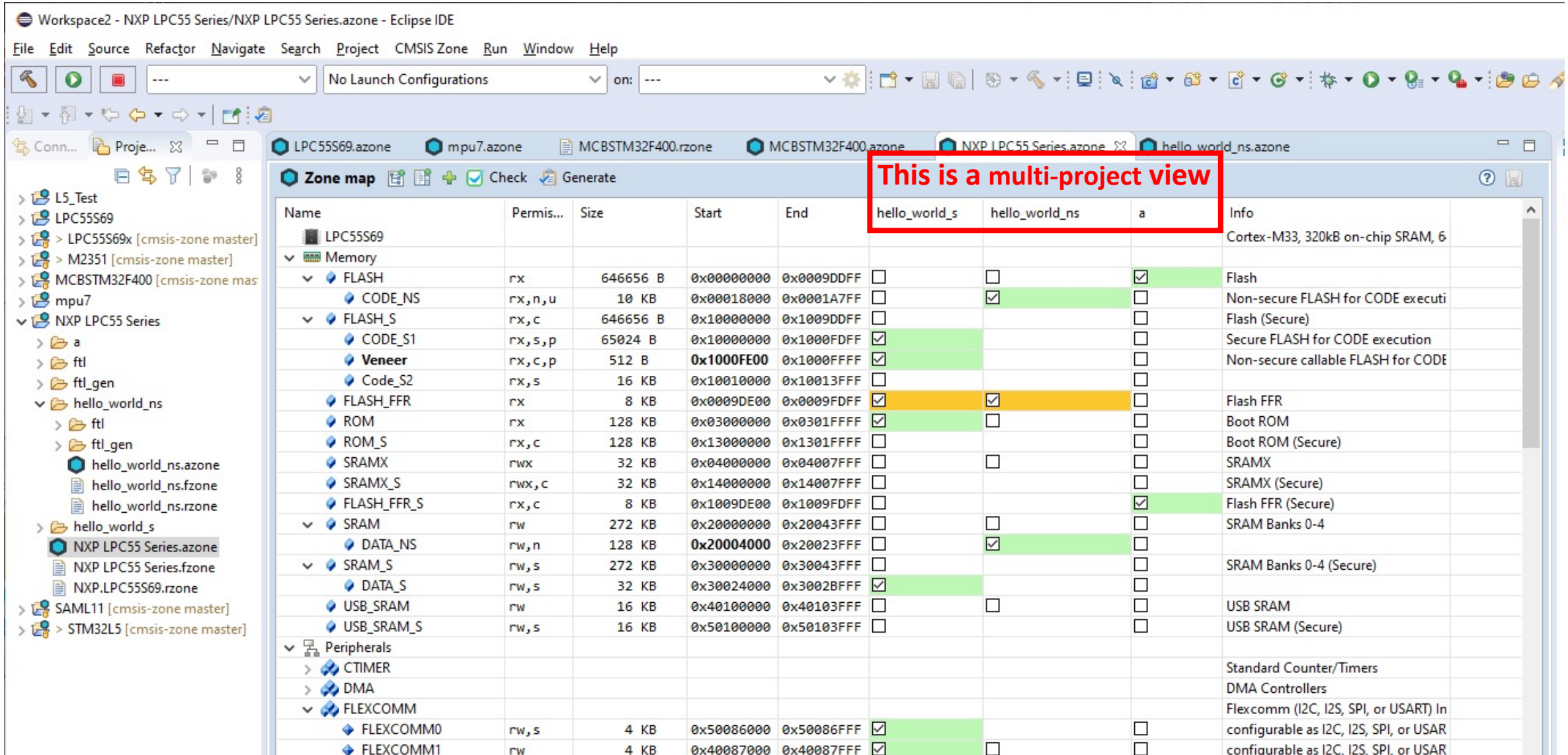
Multi-Project Requirements

Separate projects independently developed; combined in a multi-project workspace



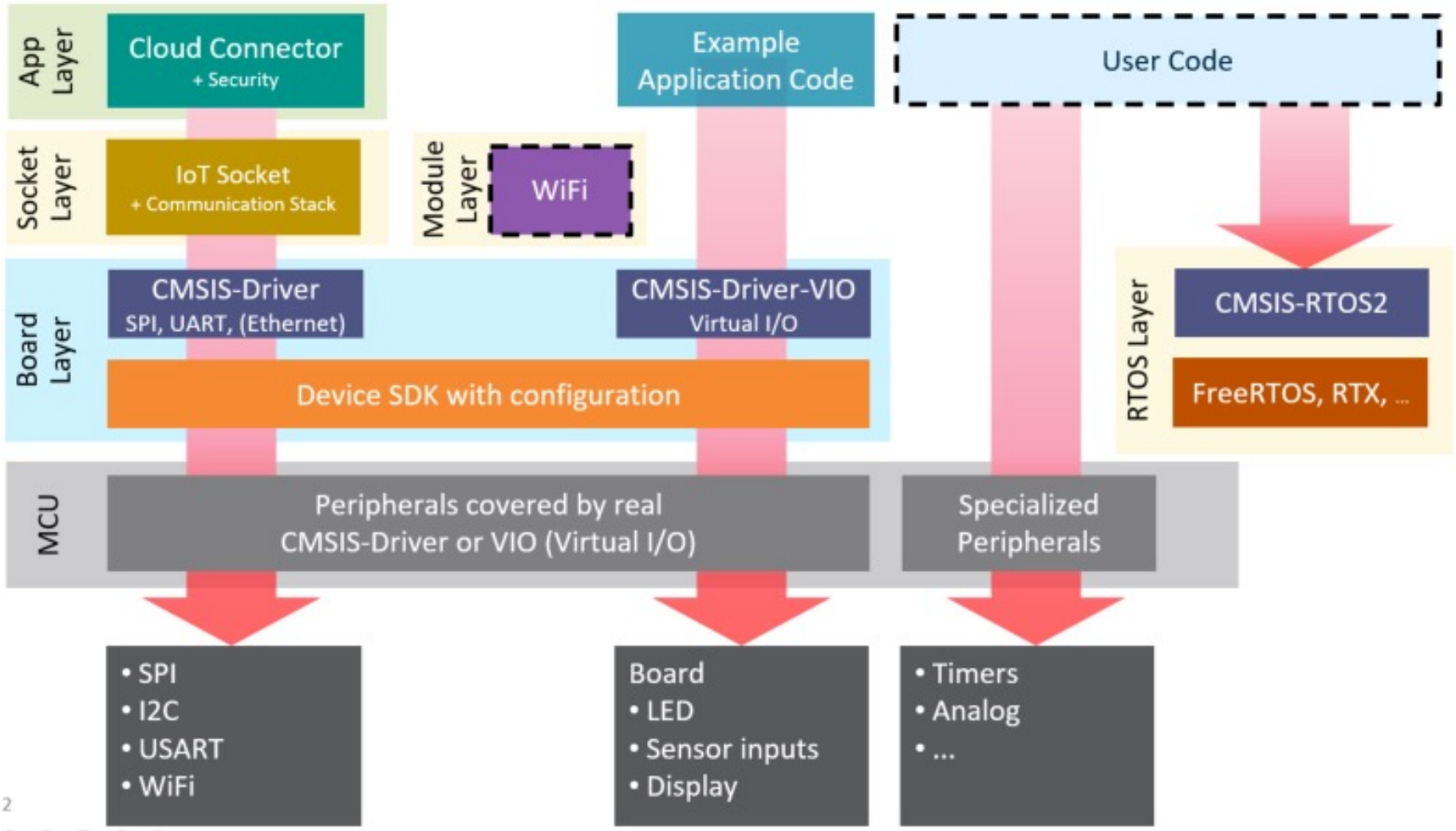
Adopt CMSIS-Zone Concepts for Multi-Project Configuration?

Discussion and decision for multi-project configuration: CMSIS-Zone + DeviceTree



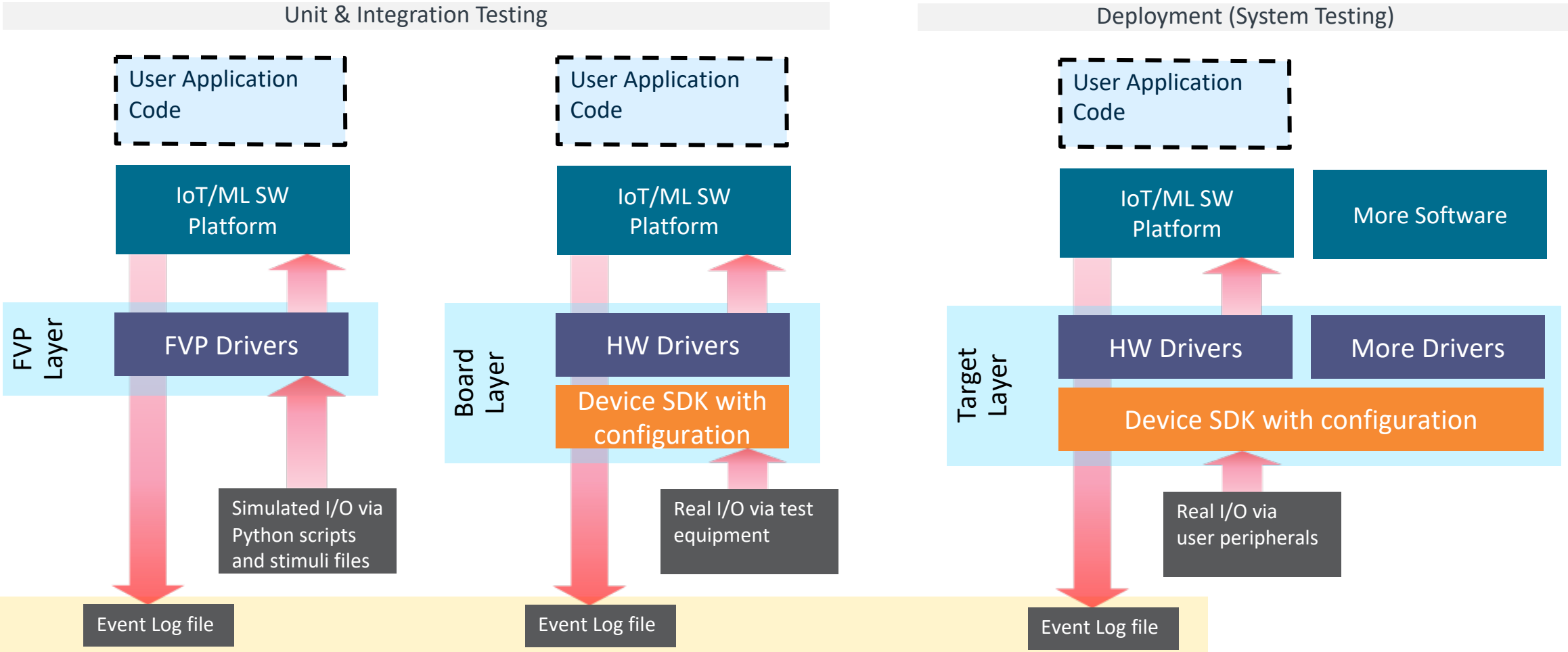
Layers: set of pre-configured software components

[GitHub - MDK-Packs/CB_Lab4Layer: CMSIS-Build Lab with Layers]



Layers: deployment to different targets for test automation

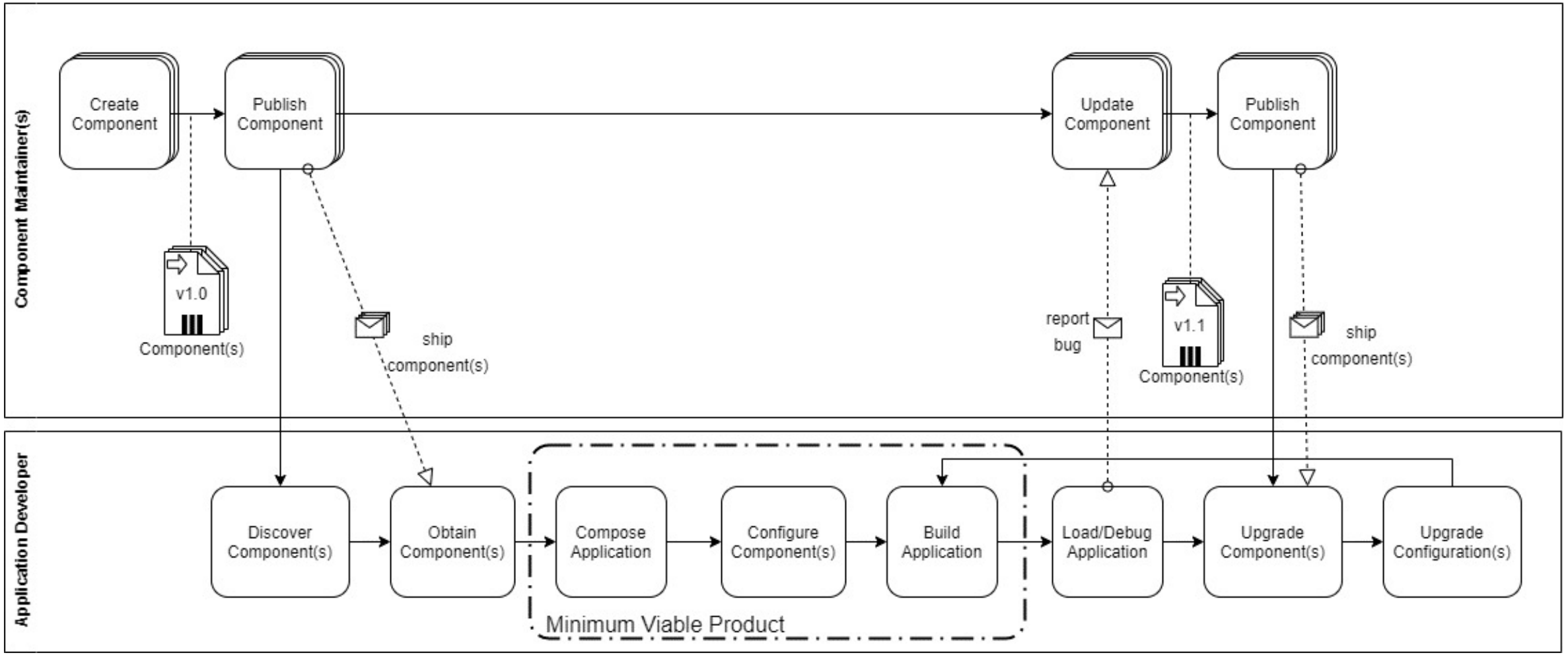
CI/CD environment for test automation – scale from Simulation to Hardware to Deployment



Essential the same event logs are generated across the different deployments. This ensures correctness.

<u>Initially:</u> Keil Studio could focus on Ease-of-Use for this process and Event Log Analysis

Component Lifecycle (Minimal Viable Product)



Compose an Application (MVP)

- User Story: As an Application Developer, I want to create an application project from available software components, so that the project can be compiled without additional user interaction
- CLI tool to create and maintain a CMSIS Project (*.cprj)
- Command set (proposal):
  - **open** *project*, **close** *project*, **exit**
  - **search** *devices, components, compiler*
  - **set** *device*, *compiler*
  - **add**/**remove** *group, file, component*
  - **list** *files, components, device, compiler, dependencies*
  - **resolve** (dependencies)
- Assumption: no manual configuration of components required

Open-CMSIS-Pack Provisional Roadmap Details

	Jul/Aug 2021	Sep/Oct 2021	Nov/Dec 2021	Jan/Feb 2022	Future
Core Library Components	**Base Components** • Open-Source existing Arm implementations	**Project Format** • CPRJ project examples • UX improvements	**Collecting Input** • Evolution of specification • Evolution of common components		
Overall Project Concept	**Base Technology** • Cmake to Pack conversion • Multi-Project Targets	**Target Connection** • Debug and download aspects of Multi-Project Targets			
Resource Management	**Multi-Project** • Review DeviceTree and CMSIS-Zone • Define structure of "Umbrealla" projects for multi-core, etc. • Organize taxonomies of standardized API interfaces		• Implement project management for multi-core and secure/non-secure setups • Refine the layer concept for better code re-use		
PoC Tools	**Project Build** • Recreate Cbuild in public GitHub Infra-structure • Close gaps in Cbuild • Pack download/install	**Pack Content** • CMake to Pack Converter • Pack Validation (PackChk)	**Project Management** • Information from CMSIS-Packs • Many boards with reference designs	**Configuration** • Text based configuration utility (aka Config Wizard)	
Process Improvements	**Optimize Delivery** • Explore potential ways to secure pack content		**Example Contribution** • Partners are enabled to submit own examples	**CI for Example/Pack** • Keep examples up-to-date • Submit process for packs with CI	

Last Update: June 21, 2021

Started | Development | Advanced Planning | Concept

The Open-CMSIS-Pack Project in Linaro

Bill Fletcher - Linaro

Linaro

www.linaro.org

The Open-CMSIS-Pack Project in Linaro

- Open-CMSIS-Pack is an engineering project within Linaro LITE (IoT & Embedded) Group
- Participating companies collaborate within Linaro, benefitting from:
  - Linaro infrastructure (Jira, Confluence, code hosting, CI hardware)
  - Linaro framework (IPR policies, antitrust guidelines)
  - Collaborative engineering (work assignment, project management)
- The Open CMSIS Pack Working Group dictates the scope of work and day to day activities
- The Working Group is a technical meeting accountable to the LITE Group Steering Committee (SC)
- The project visibility/openness is defined by the LITE SC

How To View or Participate

- Open-CMSIS-Pack is an open project in Linaro - it publishes WG technical meeting notes, and the Jira backlog:
  - Project Website https://www.open-cmsis-pack.org/
  - Public Project Pages https://linaro.atlassian.net/wiki/spaces/CMSIS/overview?homepageId=18851201976
  - Technical WG Meeting notes, slides and meeting recordings: https://linaro.atlassian.net/wiki/spaces/CMSIS/pages/28516450540/Meeting+notes
  - https://linaro.atlassian.net/jira/software/c/projects/CMSIS/issues/
- STMicroelectronics, NXP Semiconductors, Linaro and Arm are the founding members of the Open-CMSIS-Pack project
- External contributions are always welcomed
- To learn more about how to collaborate from within as a member of the Open-CMSIS-Pack project (requires minimum of Linaro Project-level Membership), please contact open-cmsis-pack-enquiry@linaro.org

arm

arm KEIL STUDIO

A browser-based IDE using Open-CMSIS-Pack technology

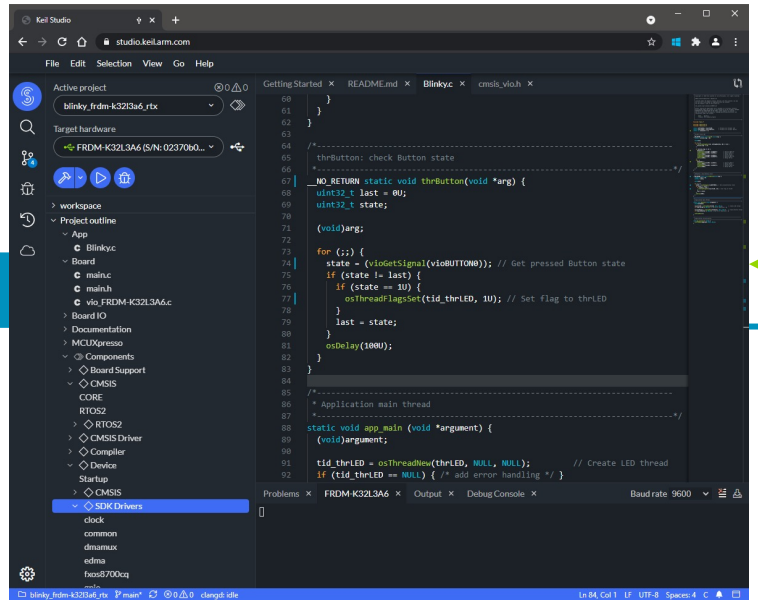
Christopher Seidl, Senior Product Manager
Embedded Tools

Cloud-based tools for embedded/IoT development

Hardware on your Desk

USB

IDE Running in a Browser on your Computer



Tools Running on a Cloud Server

Software Development Tools

Arm C/C++ Compiler
Python, Make, CMake

Embedded Application

CMSIS Software Packs
RTOS, IoT connectors, …

GitHub repository
user application code

CMSIS-DAP Debugger

Connects to many eval boards or ULINK adapters.

“Visual Studio Code” like environment

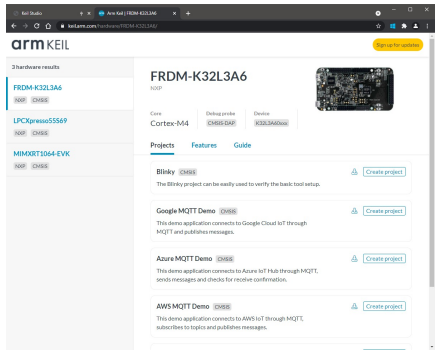
Full-featured IDE with powerful debugger, Git support – designed for IoT developers.

Compiler and Software Pre-Installed

Ready-to-use tools environment with up-to-date software and device support.

Seamless support for all development phases

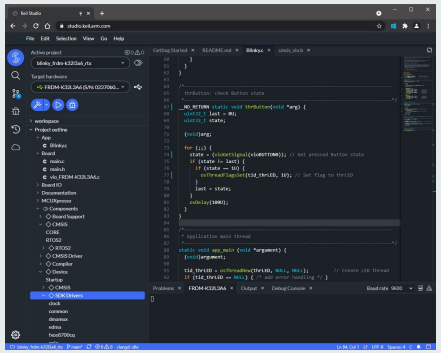
DISCOVER POSSIBILITIES



Enter parameters of your application

- Compare devices
- Evaluation boards
- Reference code examples

EXPLORE REFERENCE DESIGNS



Use online tools for testing

- Explore code
- Zero installation hassle
- Always up to date

DEVELOP APPLICATION

Download reference code and use classic tooling

- Develop and verify custom application functionality
- Extend software framework with additional functionality

DEPLOY TO CUSTOM DESIGN

Optimize application for mass production

- Retarget device pinout
- Verify system behavior
- Analyze power consumption

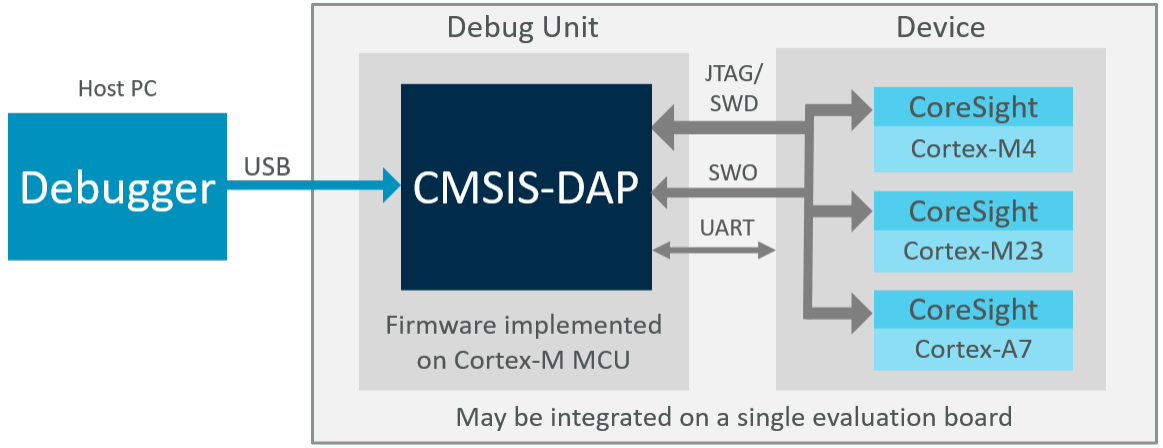
CMSIS-DAP v2.x

Speed improvements and integration of Event Recorder for CI and MLOps workflows

Christopher Seidl

CMSIS-DAP: Standardized firmware for debug adapters

- CMSIS-DAP is a specification and an implementation of a firmware that supports access to the CoreSight DAP.
- The latest version uses WinUSB as interface to the host PC and provides high-speed SWO trace streaming.
- Provides a standardized interface for debuggers.
- Supports multi-core debugging.
- Provides easy and low-cost integration of a debug unit may on an evaluation board.
- Focus is on ease-of-use, with no driver installation required
  - DAP v1 (introduced in 2010) uses HID: limits memory read/write speed to 32 KB/s
    - DAP v1 is now deprecated and not recommended for new designs.
  - DAP v2 (introduced in 2017) uses WinUSB/WebUSB: 1 MB/s memory read/write
    - Limited only by SWD clock and GPIO toggle frequency



arm

CMSIS-Dap v2.x

Roadmap

CMSIS-DAP v2.1 (included in CMSIS 5.8.0)

- Simple process for board identification
  - Extend DAP_Info command and define patch locations for easy adaptation
  - Command line utility for patching DAP_Info on individual development boards
- Debug UART (printf) via USB COM or CMSIS-DAP command interface
  - Standardize on a solution that works for many IDEs
- Board Identification in firmware
  - allows to assign a firmware image to a board pack.
  - Example usage:
    - open web pages once a board is connected
    - configure tool chain for a board
- Example firmware implementations

CMSIS-DAP v2.2 – Sept 2021

- Support for CoreSight ADIv6 debug protocol
- Event Recorder source is open-sourced here
- First-stage capture for Event Recorder
  - Makes Event Recorder available to other tools
  - Improves speed of event streaming
  - Simplifies implementation for host debuggers

More information on Event Recorder:

Webinars:

Identifying timing and power consumption bottlenecks
Optimizing a constrained embedded application

Whitepaper:

Software analysis with event annotations

arm

Board identification for generic CMSIS-DAP firmware

Assigns a Board ID to existing firmware – no need to re-compile

- Create generic DAP FW for the circuit you are using on dev kits
- Info in Board Pack - `NXP.LPCXpresso55S69_BSP.pdsc`

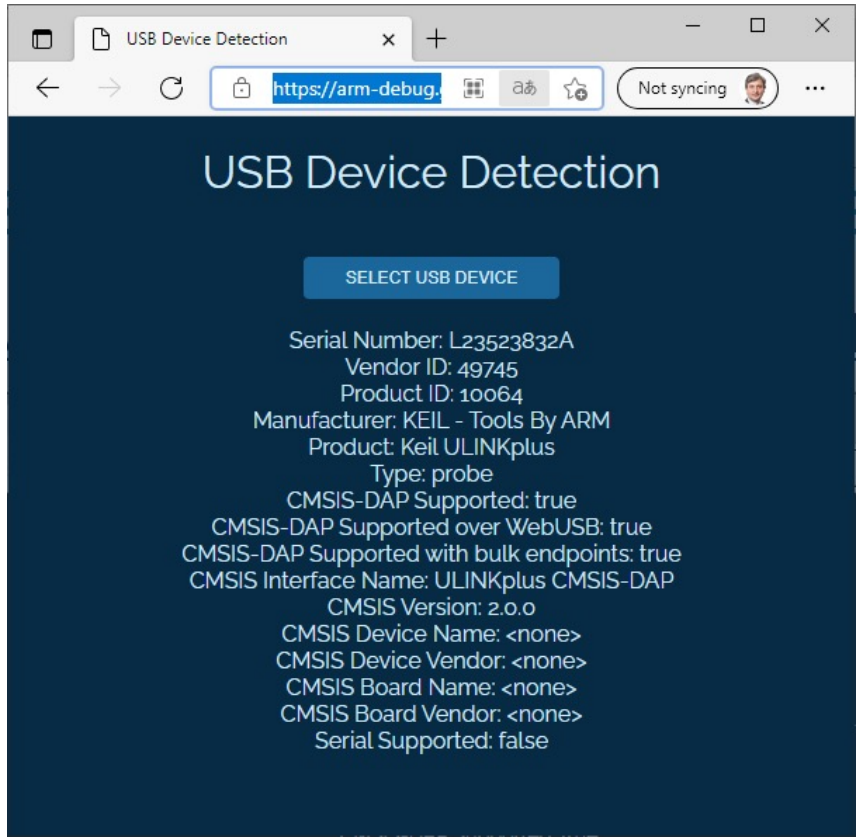
```
<board vendor="NXP" name="LPCXpresso55S69">
 …
    <mountedDevice Dname="LPC55S69" Dvendor="NXP:11"/>
 …
</board>
```

- Create patch file – `LPCXpresso55S69.patch`:

```
# symbol            : string/values
TargetBoardVendor   : "NXP"
TargetBoardName     : "LPCXpresso55S69"
TargetDeviceVendor  : "NXP"
TargetDeviceName    : "LPC55S69"
```

- Patch firmware:
  `patchELF CMSIS_DAP.axf LPCXpresso55S69.patch`
- Program patched CMSIS_DAP firmware on specific development board

- Validate deployment to board
  https://arm-debug.github.io/device-detection/

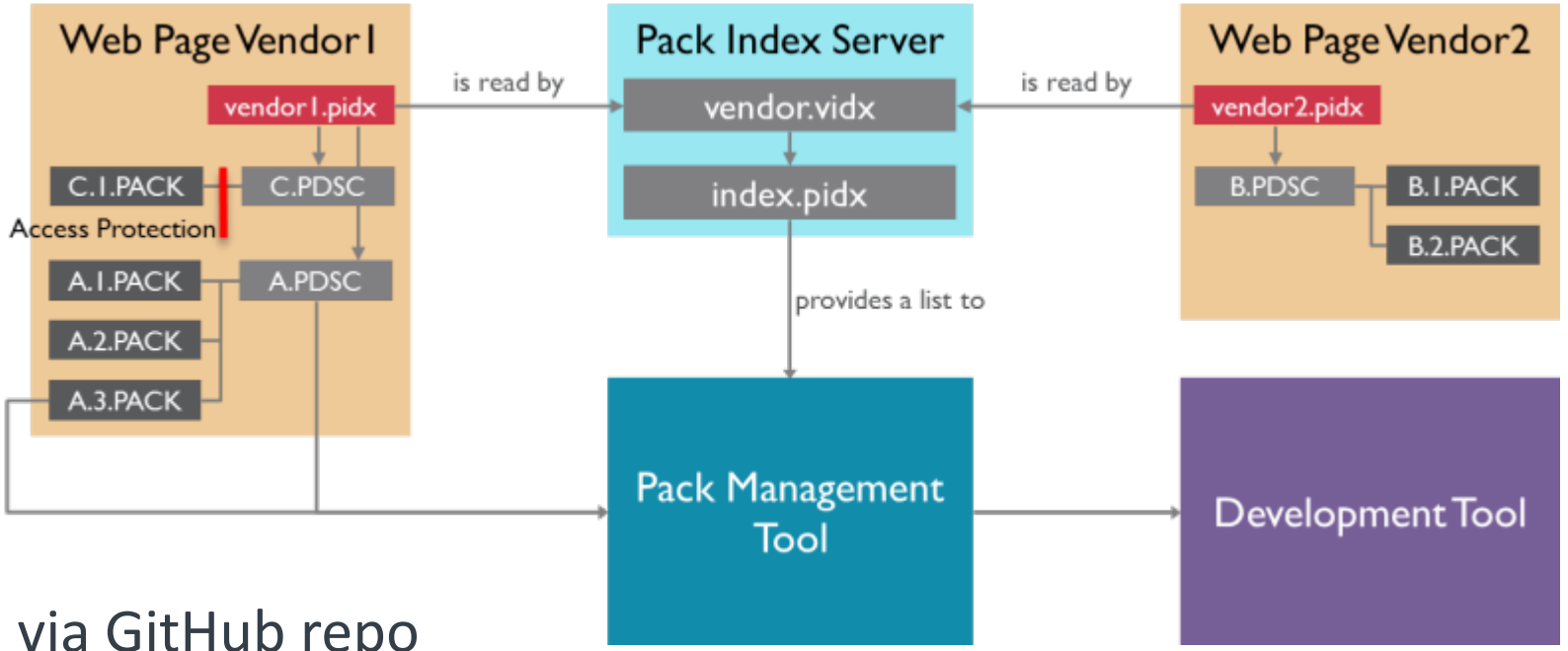


Enhanced pack submission process

Will Lord, Technology Manager Embedded Tools

Current pack submission process

- Arm hosts an index server
- For packs to appear on that index, submit to cmsis@arm.com
- Pack generation and checking tools are available via GitHub repo
- Updates are checked automatically every night when using an index file



Performant, consistent pack delivery

Pack source
Index server
…

CMSIS Core Tools and Services

Pack status

Pack index + cache

Pack Parsing

Software DB

Hardware DB

Build status

Build service

(Public) APIs

Hardware API

Software API

Clients

Keil.arm.com

Keil Studio

Partners

Process and store data from packs

Expose hardware and software metadata, status and pack relationships to clients

Consume APIs and expose data to end users

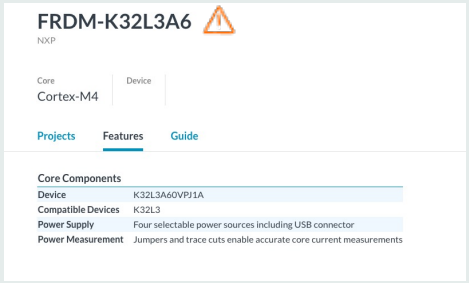
Streamlined, iterative pack submission and management

Guided pack submission process

LOGIN OR AUTH | VALIDATE | PUBLISH | MANAGE

Access vendor account

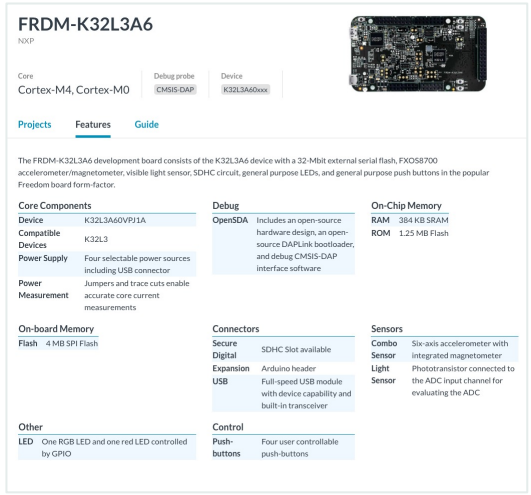
- Web UI, API, CLI
- Manage software and hardware packs




```
> cmsis validate my-pack.pack
```

Upload new pack

- Validate metadata, dependencies, software build state
- Highlight errors



Preview web pages before publication

- View and share
- Iterate and publish

Manage packs and web presence

- Data enhancement
- Updates
- Publication lifecycle
- Alerting
  - Pack unavailable
  - Build failing
- Analytics
  - Downloads
  - Project usage

Best display of development boards

What does awesome look like?

- PDSC board description section contains:
  - Full featured `<board>` element
  - **Picture of the board** in the `<image>` element
  - **Short** `<description>` not to overload the page
  - **All available** collateral described in `<book>` elements (including an MD based user's guide)
  - `<mountedDevice>` and `<compatibleDevice>` elements
  - A **complete list** of `<feature>` elements
  - A `<debugInterface>` and `<debugProbe>` element

- Examples
  - BSP for STM32L562E-DK (github.com/MDK-Packs/STM32L562E-DK_BSP/blob/develop/Keil.STM32L562E-DK_BSP.pdsc)
  - BSP for Flex iENBL-DK (github.com/MDK-Packs/iENBL-DK_BSP/blob/master/Keil.iENBL-DK_BSP.pdsc)

CMSIS-DSP/NN update

Laurent Le Faucheur, Senior Principal Engineer

CMSIS DSP 2021 : Version 1.9.0

- **New float16 datatype** for most of the float32 functions
  - Process in 4 MAC/cycle on the Cortex-M55
- Big re-organization of arm_math.h:
  - No need anymore to include all functions. Smaller headers available in Include/dsp
- **Interpolation functions** are no more header only:
  - There is a new Interpolation folder
- **Quaternion** folder added: Required for robots and moving objects
- **New algorithms** available (or new datatypes)
- Preliminary support for **Helium** with gcc
- Upgrade to the **Python wrapper**: More functions supported
- Source code only release

CMSIS-NN

- Cortex-M55 alignment (int16 tensors x 8bits weights)
- Corstone-300 support
- SVDF operator (for speech recognition)
- Endpoint-AI repository with:
  - Stereo vision
  - Pixel Layer
  - Voice activity detection (VAD)

arm

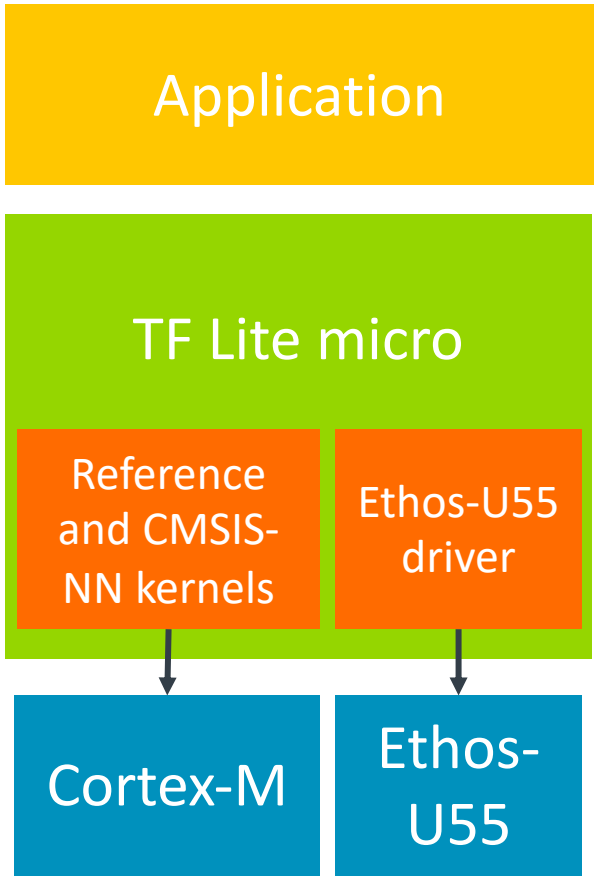
TinyML with CMSIS-Pack

For cloud-based workflows

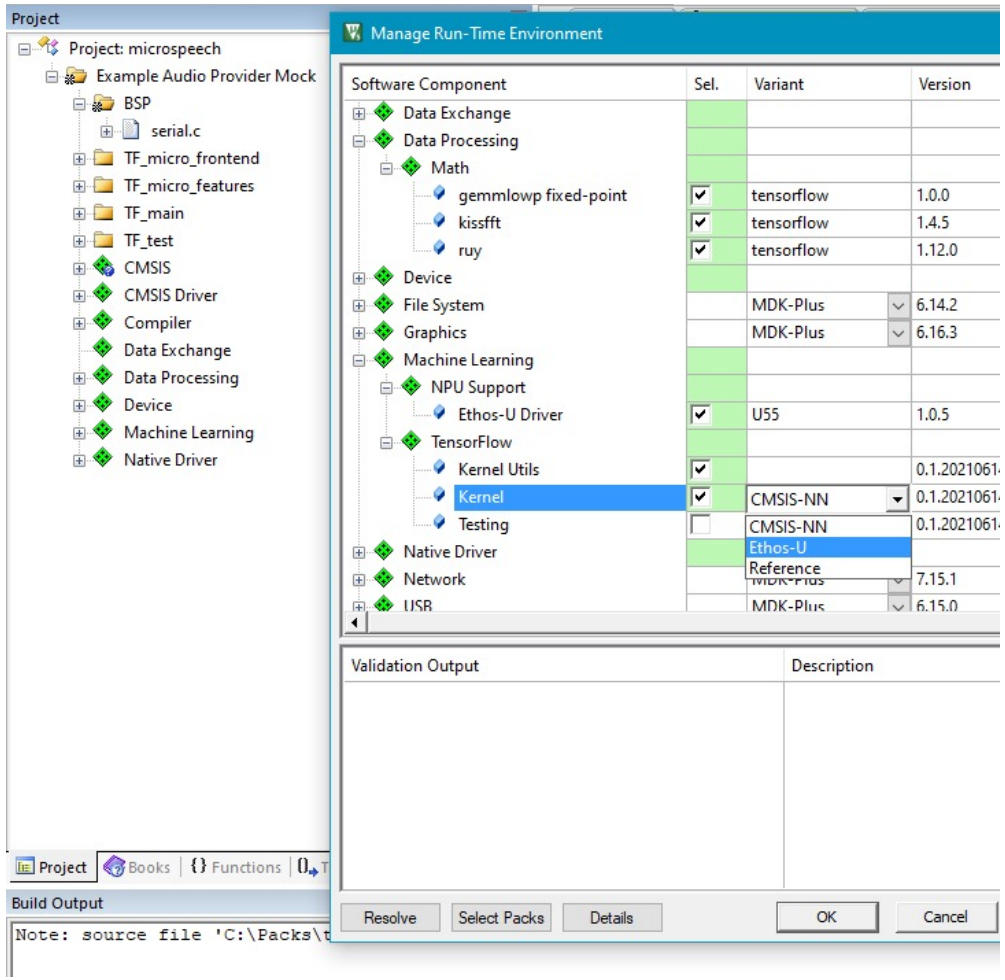
Reinhard Keil

Software Stack Integration with TensorFlow

CMSIS-NN and Ethos-U55 under the same stack



- TFLu is provided as a pack
  - TFLu is a source component for better configurability and enablement of Link-Time Optimization
- Variants enables optimized kernels in the build system
- Software is open source:
  - Vela compiler
  - Ethos-U55 driver
  - TFLu
  - CMSIS-NN
- Uses Event Recorder to provide timing information



Summary and Questions

Reinhard Keil

Timeline	Description	How you can contribute
June 2021	CMSIS V5.8.0	https://github.com/arm-software/cmsis_5 Use Issues to provide feedback
Starting now	**Open-CMSIS-Pack project** under Linaro www.open-cmsis-pack.org	Sign-up to mailing list: https://op-lists.linaro.org/mailman/listinfo/open-cmsis-pack-dev To become a member contact: open-cmsis-pack-enquiry@linaro.org
Ongoing	Device Support and Board Support via CMSIS-Pack	Submit new device support or board support Contact: cmsis@arm.com
Nov 2021	Enhanced process for pack submission and example project contribution	Arm will invite to an update meeting where we introduce the new process. To get involved early contact: cmsis@arm.com
March 2022	CMSIS Review Meeting @ Embedded World	We hope that we can meet physically again for good discussions

arm

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