

# CMSIS Version 5

Partner Meeting Embedded World 2016

**ARM**



Nuremberg – Embedded World 2016  
23. February 2016

# Agenda

- Welcome & CMSIS Overview
- ARMv8-M and CMSIS Version 5 Status
- CMSIS-RTOS Status and Validation Suite
- CMSIS-Pack Status and Validation
- CMSIS-Driver and Validation Suite
- Summary and Discussion

# ARM® – Development Solutions Group (DSG)

**DSG Vision:** The ARM architecture has the best software tools [solutions], regardless of whether they come from ARM or the ecosystem

**DSG Mission:** To create or enable tools that improve ARM system performance, productivity or time to market; and to fund this through a growing business.

# Cortex Microcontroller Software Interface Standard (CMSIS)

Vendor-independent Standard for hardware manufacturers and tool vendors

## Software Layers for ARM Cortex<sup>®</sup>-M processor based devices

- CMSIS-CORE API for Cortex-M processor and core peripherals
- CMSIS-DSP DSP Math Library with more than 60 functions
- CMSIS-RTOS API for RTOS integration
- CMSIS-DRIVER API for peripheral driver interfaces

CMSIS Version 5 will address hybrid devices based on Cortex-A / M

## Infra-Structure for Cortex-A / R / M processor based devices

- CMSIS-SVD XML system view description for peripheral debugging
- CMSIS-DAP Firmware for Debug Access Port
- CMSIS-PACK XML description for software components, device parameters, board support

[www.arm.com/cmsis](http://www.arm.com/cmsis)

### Cortex-M Series

- > Cortex-M7 Processor
- > Cortex-M4 Processor
- > Cortex-M3 Processor
- > Cortex-M1 Processor
- > Cortex-M0+ Processor
- > Cortex-M0 Processor
- > CMSDK
- > **CMSIS**

# How popular is CMSIS?

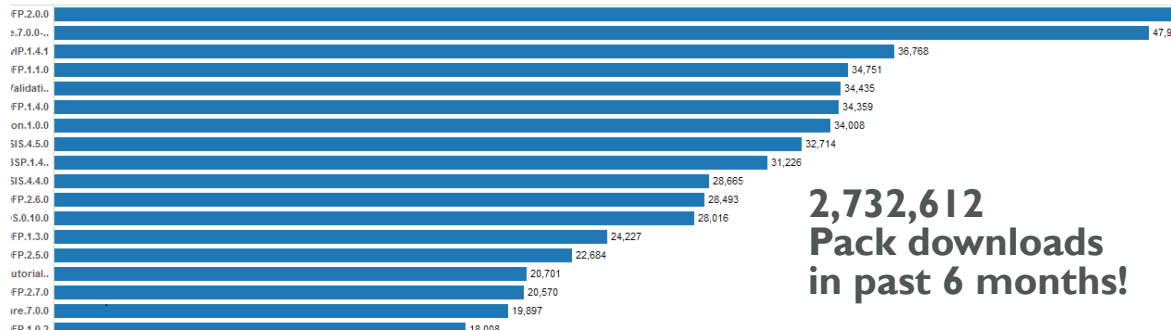
- Supported with CMSIS-Pack

- 3,278 Cortex-M devices

- Keyword “CMSIS” delivers

- Google: about 264,000 results
- Youtube: about 1,920 results

- GitHub Search “CMSIS”



2,732,612  
Pack downloads  
in past 6 months!

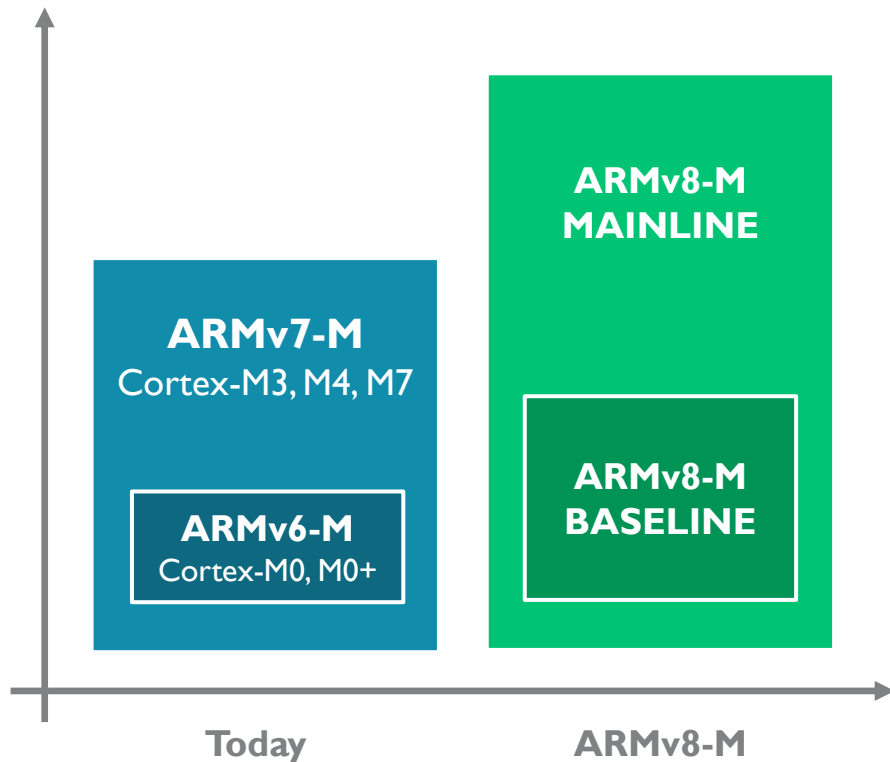
Pack	Action	Description
[-] Device Specific	4 Packs	Infineon selected
[-] Infineon::TLE986x_DFP	Update	Infineon TLE986x Series Device Support
[-] Infineon::TLE987x_DFP	Update	Infineon TLE987x Series Device Support
[-] Infineon::XMC1000_DFP	Update	Infineon XMC1000 Series Device Support, Drivers and Examples
[-] Infineon::XMC1000_DFP	Install	Infineon XMC1000 Series Device Support, Drivers and Examples
[-] Infineon::XMC1000_DFP	Remove	Infineon XMC1000 Series Device Support
[-] Infineon::XMC1000_DFP	Previous	Infineon::XMC1000_DFP - Previous Pack Versions
[-] Infineon::XMC4000_DFP	Up to date	Infineon XMC4000 Series Device Support, Drivers and Examples
[-] Infineon::XMC4000_DFP	Remove	Infineon XMC4000 Series Device Support, Drivers and Examples
[-] Infineon::XMC4000_DFP	Remove	Infineon XMC4000 Series Device Support, Drivers and Examples
[-] Infineon::XMC4000_DFP	Remove	Infineon XMC4000 Series Device Support, Drivers and Examples
[-] Infineon::XMC4000_DFP	Previous	Infineon::XMC4000_DFP - Previous Pack Versions

14 Silicon vendors adopted CMSIS-Pack



# ARMv8-M Architecture – At a glance

Scalable Architecture for Microcontrollers

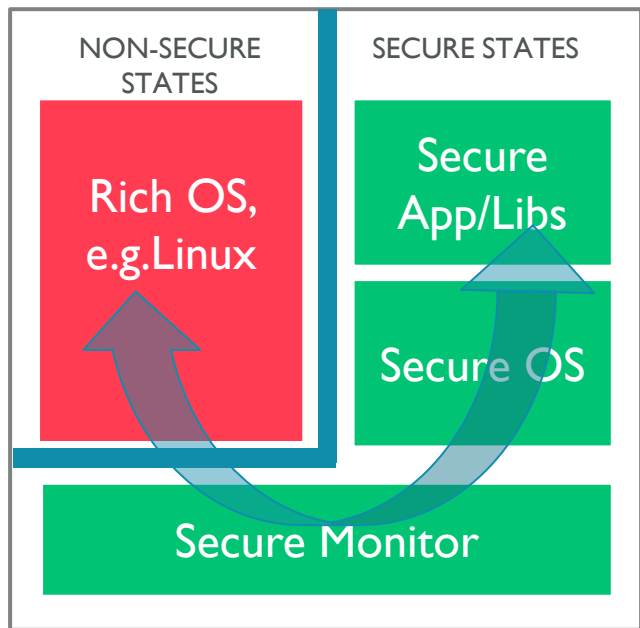


## ARM TRUSTZONE

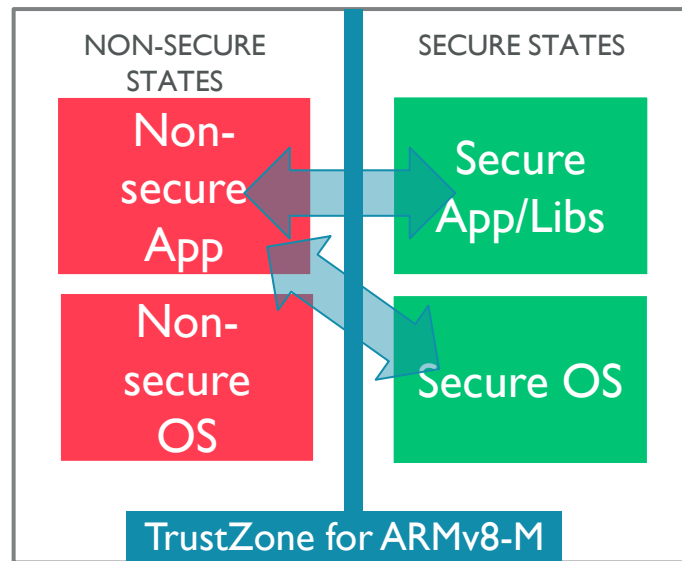
System Security

- ARMv8-M **Baseline**:
  - Lowest cost and smallest implementations
- ARMv8-M **Mainline**:
  - For general purpose microcontroller products
  - Highly scalable
  - Optional DSP and floating-point extensions

# TrustZone for ARMv8-A



# TrustZone for ARMv8-M

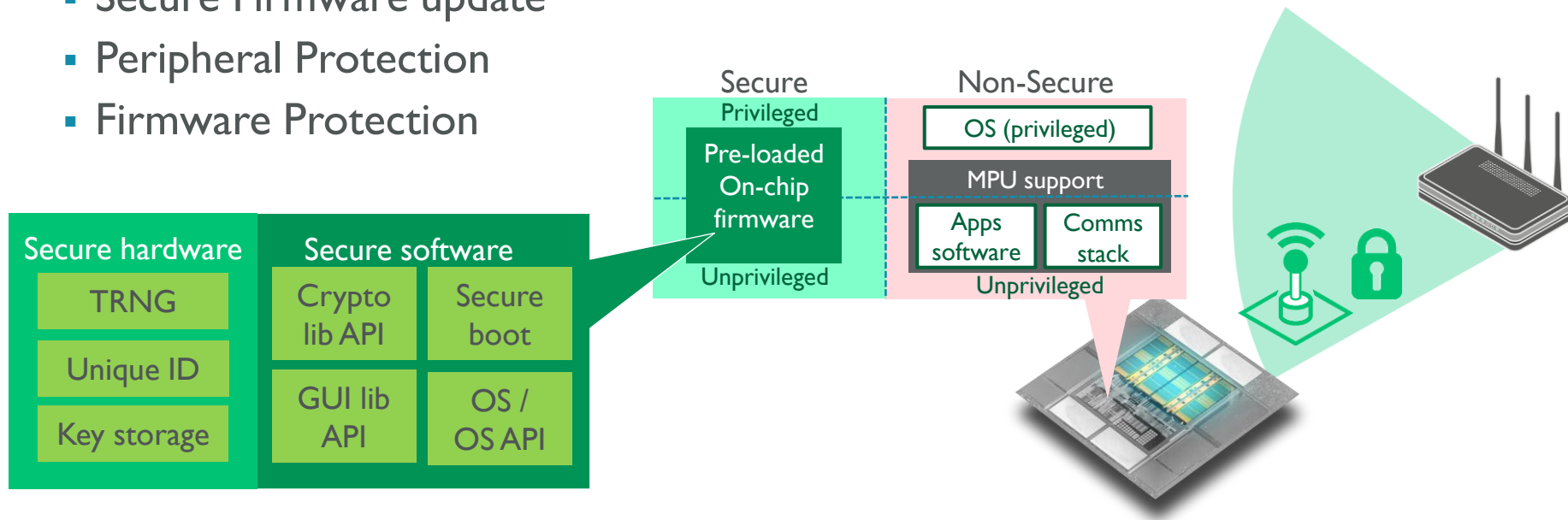


Secure transitions handled by the processor  
to maintain embedded class latency

# ARMv8-M Security Domains and Protection

**Secure State** is used for

- Cryptography and authentication
- Secure Firmware update
- Peripheral Protection
- Firmware Protection





# CMSIS Version 5 - Status

## No new components → Focus on Improvements & Further Industry Adoption!

- License change to **Apache 2.0** to enable contributions from 3<sup>rd</sup> parties
- Public development using GitHub: [https://github.com/ARM-software/CMSIS\\_5](https://github.com/ARM-software/CMSIS_5)
- Add support for ARMv8-M Architecture (Mainline and Baseline)
- Improvements for Cortex-A / M hybrid devices (focus on Cortex-M interaction)

## CMSIS-RTOS API and RTX reference implementation with several enhancements:

- Dynamic object creation, Flag events, C and C++ API, additional thread and timer functions
- Secure and Non-Secure support, multi-processor support

## CMSIS-Pack

- Additions for generic example, project templates, multiple download portals
- Adoption of IAR Flash Loader technology

# CMSIS-RTOS

## Status and Validation Suite

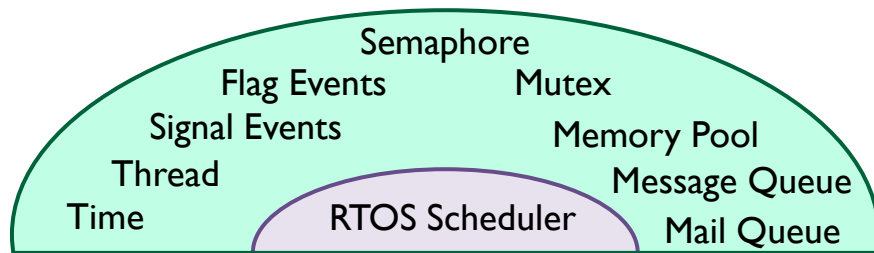


Matthias Hertel  
Product Specialist

CMSIS Version 5 Partner Meeting, Embedded World  
23. February 2016

# CMSIS-RTOS API – Version 2 Enhancements

- Common API for Real-Time Operating Systems, compatible to Version 1



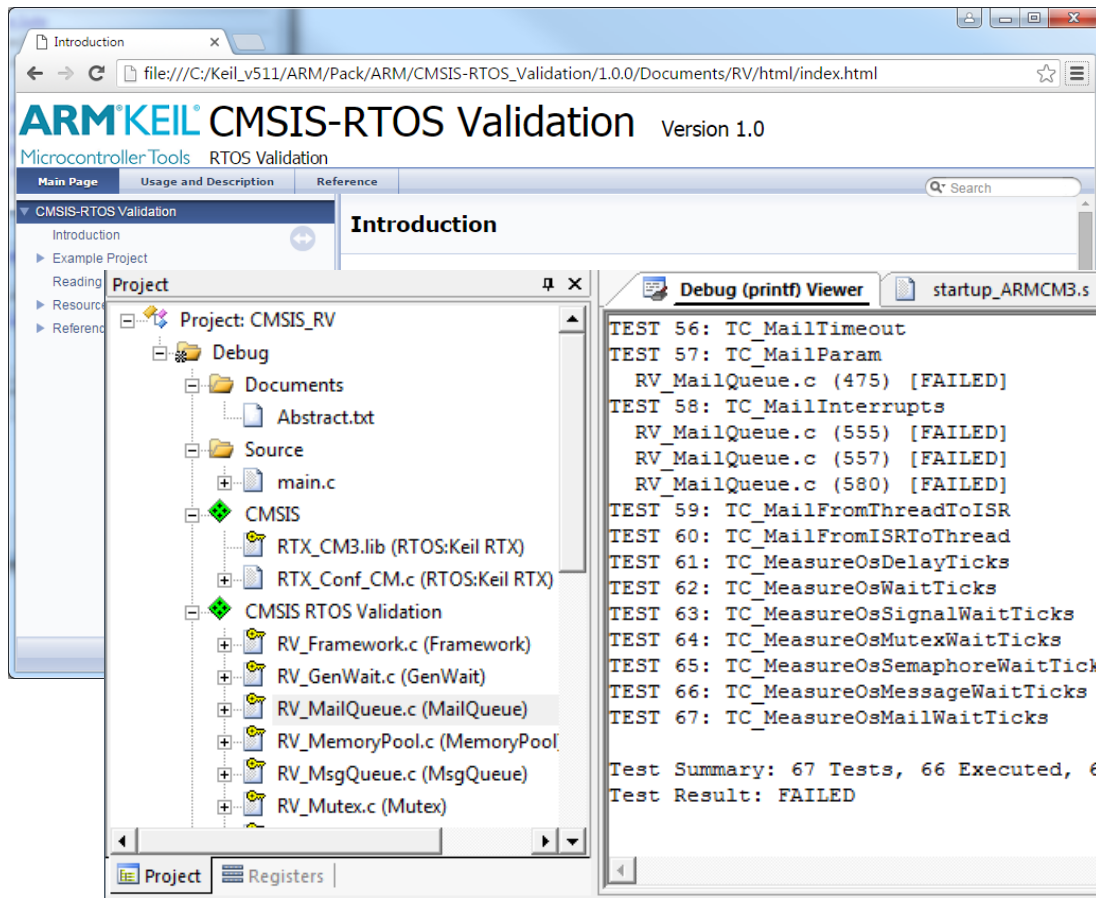
Pre-emptive thread scheduling with priorities

CMSIS-RTOS v2 also addresses:

- Message passing in Cortex-A / M hybrid systems
- C and C++ API
- ARMv8-M Security domain support

Version 2 Enhancements / additions:

- More thread priorities
- Dynamic Object creation
  - Initializing osXxxxDef definitions
  - Multiple instances Mutex & Semaphore
- External reference to object definitions
- osKernelTime, osKernelStop
- osThreadSuspend, osThreadResume
- osPoolDelete
- osMessageCount, ...Reset, ...Delete
- osMailCount, osMailReset, osMailDelete
- osFlagXxx global event flags

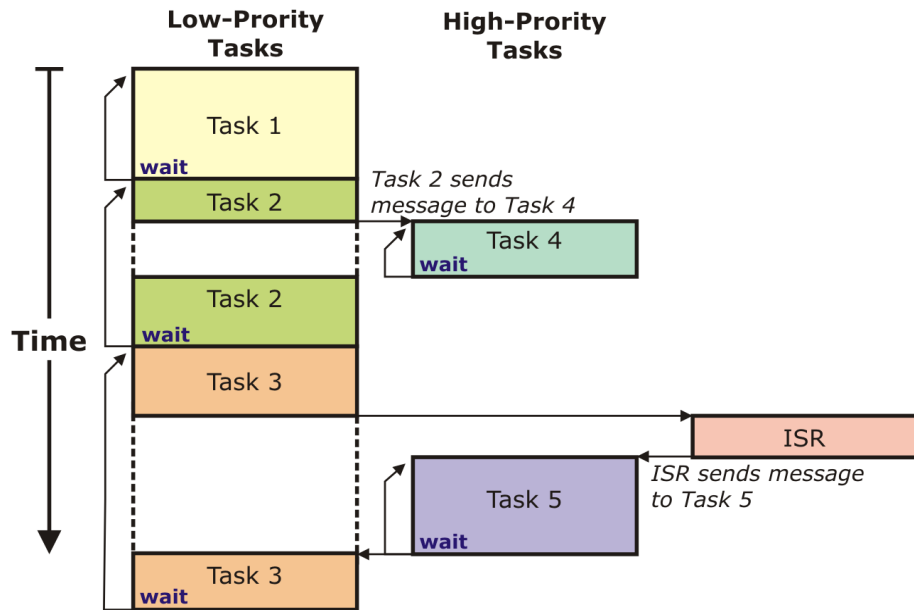


## Validation Suite

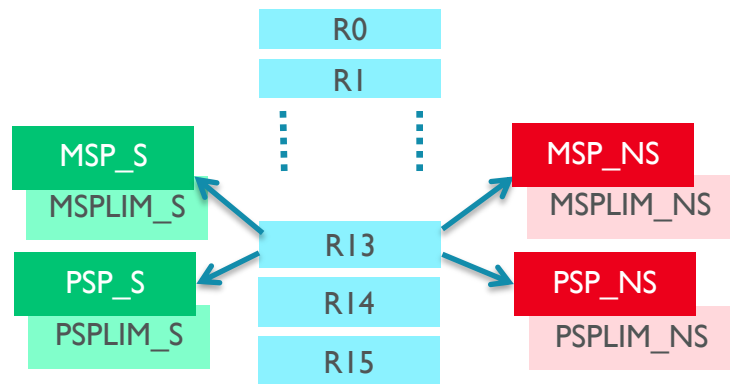
- Available as Software Pack

# RTOS Scheduling on v8-M

- Thread or task context requires to change stack



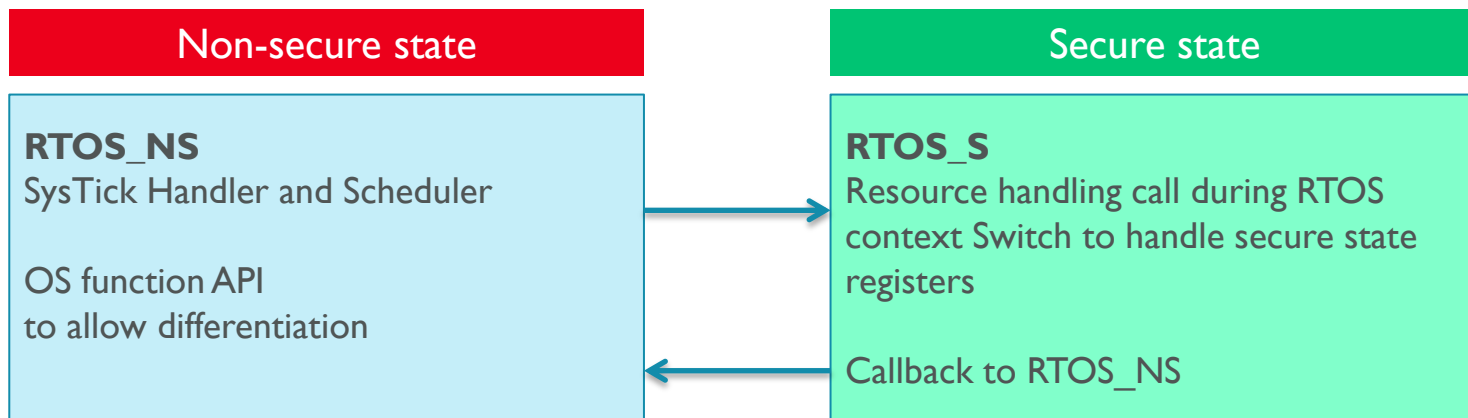
- ARMv8-M provides additional stack pointers for Secure State



# RTOS for ARMv8-M – Potential Implementations

- Variants for potential implementations are:
  - A. RTOS running in Secure state: RTOS functionality available to Secure and Non-Secure software
  - B. RTOS running in Non-Secure state: RTOS functionality only available to Non-Secure software
  - C. RTOS running in Non-Secure state: RTOS functionality available to Non-Secure and Secure software
- Variant B is the most common case and our initial focus
- Variant C is the most flexible implementation, but maybe not required

# ARMv8-M CMSIS-RTOS Scheduler – Concept



- Split RTOS into Secure/Non-Secure part
- Allow calls to software that runs in Secure State
- Allow callbacks from RTOS\_S to RTOS\_NS
- Extend Task Context Block to handle Secure Stack
- Use new Stack Limit Registers

# ARMv8-M RTOS Secure Process Stack Management - API

## Initialize Secure Process Stack Management

```
int32_t TZ_Init Stack S (void);           returns 0: success, != 0 error code
```

## Allocate Memory for Secure Process Stack Management (called on osThreadCreate)

```
module id indicates box with software modules
```

```
int32 t TZ Alloc Stack S (uint32 t module id);           returns >=0 context id, < 0: no memory
```

## Free Memory for Secure Process Stack Management (called on osThreadTerminate)

```
int32 t TZ Free Stack S (int32 t context id);           returns 0: success, != 0 error code
```

### Load Secure PSP (and set limit PSP\_LIM\_S) (call on ContextRestore)

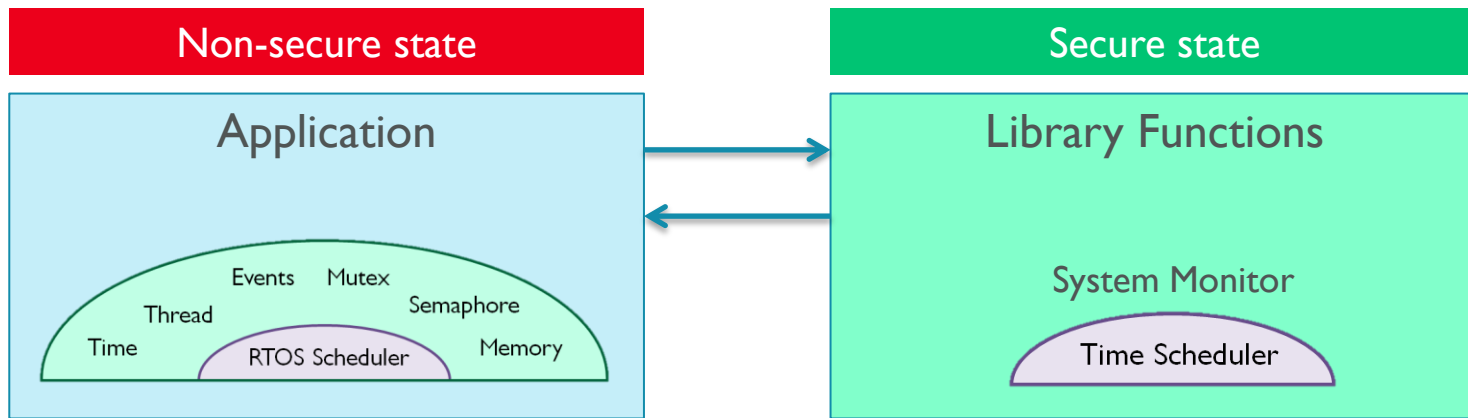
```
int32 t TZ Load Context S (int32 t context id);      returns 0: success, != 0 error code
```

## Store Secure PSP

```
int32_t TZ_Store_Context_S (uint32_t context id);      returns 0: success, != 0 error code
```



# ARMv8-M CMSIS-RTOS RTX Implementation



- Full-featured RTOS for Non-Secure Application
  - Supports function calls to Secure state
  - Callback events from Secure state
- Uses Process Stack Pointer (PSP\_NS, PSP\_S)
  - Interrupts use Main Stack Pointer (MSP)

- Secure state provides data and firmware protection
- System Monitor for operation protection
  - Additional Scheduler using Secure SysTick timer
  - Runs on Main Stack Pointer (MSP\_S)
  - Runs on highest interrupt priority

# CMSIS-Pack

## Status and Validation



Christopher Seidl  
Technical Marketing Manager

CMSIS Version 5 - Partner Meeting, Embedded World  
23. February 2016

## ▼ CMSIS-Pack

Revision History of CMSIS-Pack

- ▶ Create Software Packs
- ▶ Pack with Software Components
- ▶ Pack with Device Support
- ▶ Pack with Board Support
- ▶ Pack Example **Tutorials**
- ▶ Utilities for Creating Packs
- ▶ Publish a Pack
- ▶ Pack Description (\*.PDSC) Format
- ▶ Configuration Wizard Annotations
- ▶ Flash Programming Algorithms

## CMSIS-Pack Documentation

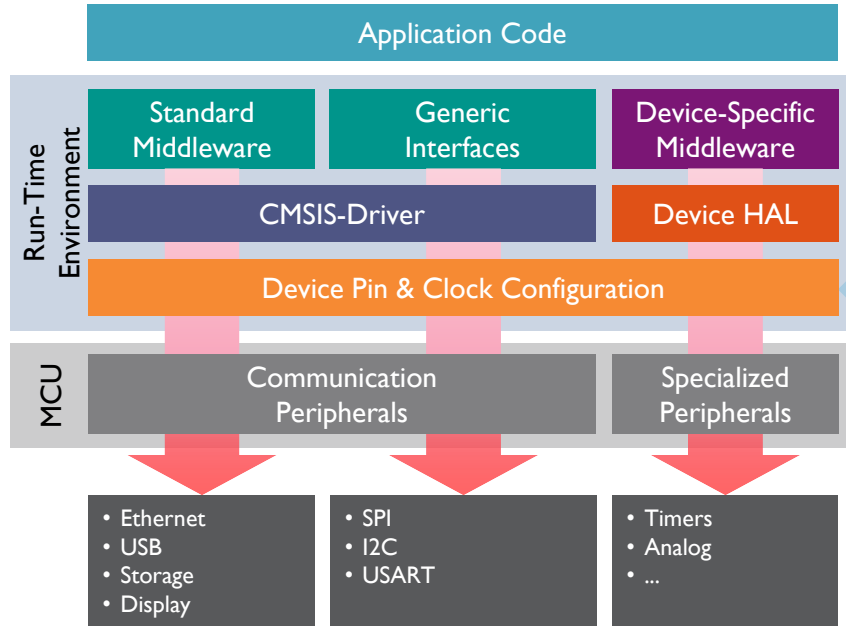
**CMSIS-Pack** describes a delivery mechanism for software components, device parameters, and evaluation board support. The XML-based package description (PDSC) file describes the content of a **Software Pack** (file collection) that includes:

- Source code, header files, and software libraries
- Documentation and source code templates
- Device parameters along with startup code and programming algorithms
- Example projects

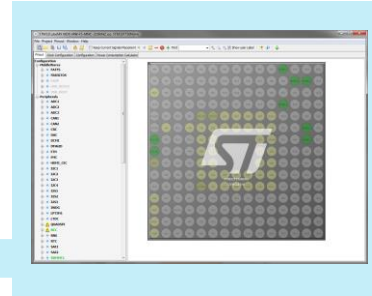
## Software Pack Use Cases

Variant	Device Family Pack	CMSIS Pack	Middleware Pack	Board Support Pack	In-house Software Pack
Source	Silicon Vendor, Tool Vendor	ARM	Silicon Vendor, Tool Vendor, 3rd Party	Board Vendor	Tool User

# CMSIS-Pack - Focus for CMSIS Version 5



- Simplify Device Configuration



Generated Software Pack  
\*.GPDSC

- IDE Independent Project Examples & Templates
  - \*.cpdsc file with <create> element
- <requirements> element to list Software Packs
- C/C++ Language requirements
- Eclipse integration

# Standardization for Pack Download Portals

- Allow multiple download portals (all are equal), with references to other portals

www.vendorA.com



vendorA.pidx

www.vendorN.com



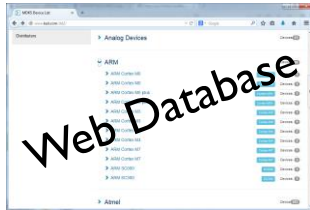
vendorN.pidx

remote.list:

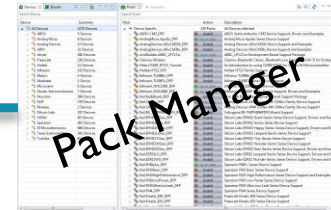
- `<url> vendorA.pidx`
- ...
- `<url> vendorN.pidx`



index.idx

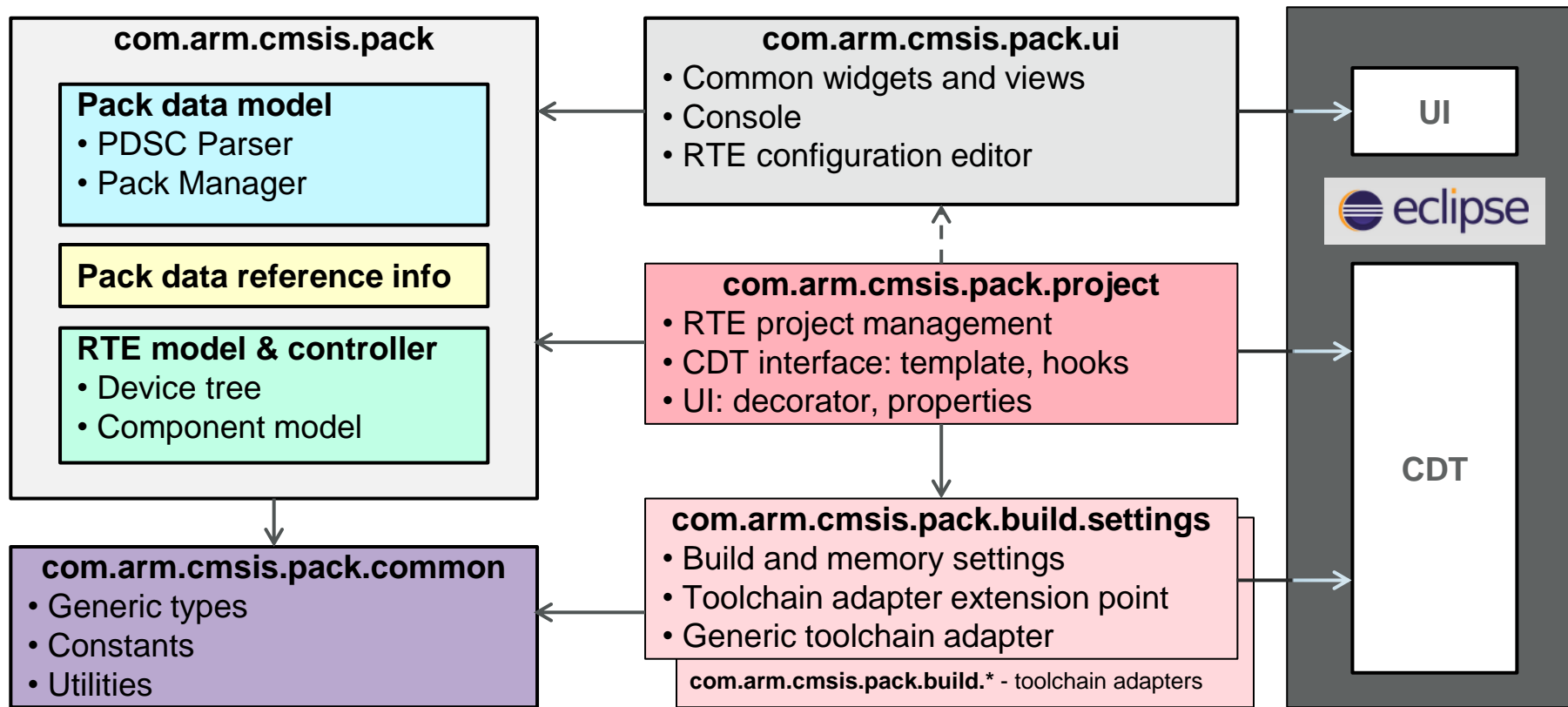


Web Database



Pack Manager

# CMSIS-Pack for Eclipse – Architecture & Modules



# Demo

The screenshot displays the Eclipse IDE interface for configuring an STM32 project. The main window is titled "C/C++ - STM32/STM32.rteconfig - Eclipse Platform". The left sidebar shows the project structure, including "Includes", "RTE", "CMSIS", "Device", and "main.c". The central pane shows the "Components\*" table, which lists various software components and their configurations.

Software Components	Sel.	Variant	Vendor	Version	Description
STM32F407IEHx	<input checked="" type="checkbox"/>		STMicroelectr		
Board Support	<input checked="" type="checkbox"/>	MCBSTM32F400	Keil	2.0.0	
CMSIS	<input checked="" type="checkbox"/>				
CORE	<input checked="" type="checkbox"/>		ARM	4.1.0	
DSP	<input checked="" type="checkbox"/>		ARM	1.4.5	
RTOS (API)	<input checked="" type="checkbox"/>			1.0	
Keil RTX	<input checked="" type="checkbox"/>		ARM	4.78.0	
CMSIS Driver	<input checked="" type="checkbox"/>				
Compiler	<input checked="" type="checkbox"/>				
Device	<input checked="" type="checkbox"/>				
Startup	<input type="checkbox"/>		Keil	2.3.1	System

Below the table, the "Validation Output" section shows a warning: "ARM::CMSIS.RTOS.Keil RTX require Cclass='Device', Cgroup='Startup'".

The "Select Device" dialog is open, showing the following information:

- Device: LPC4357:Cortex-M4
- Vendor: NXP
- Pack: Keil.LPC4300\_DFP.2.5.0
- URL: <http://www.keil.com/dd2/nxp/lpc4357>
- CPU: ARM Cortex-M4
- Clock: 204 MHz
- Memory: 96 kB RAM, 1 MB ROM
- FPU: single precision
- Endian: Little-endian

The dialog also displays a tree view of the device's components, including "LPC432x", "LPC433x", "LPC435x", "LPC4350", "LPC4353", "LPC4357", "LPC4357:Cortex-M0", "LPC4357:Cortex-M4", "LPC437x", "Spansion", and "STMMicroelectronic".

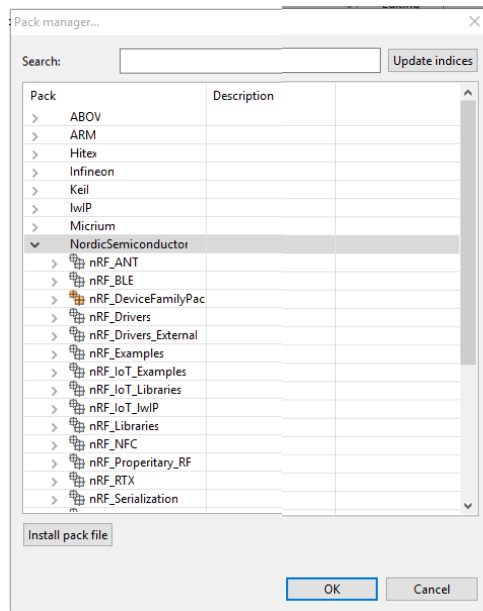
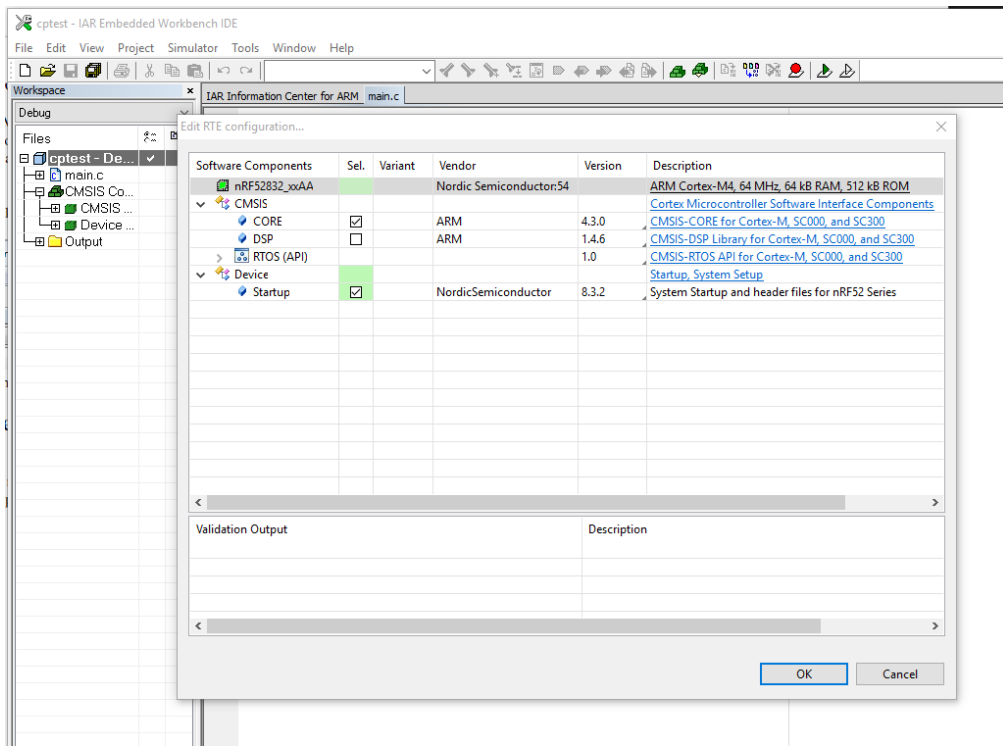
# IAR CMSIS-PACK roadmap

Anders Lundgren, IAR Systems Sweden



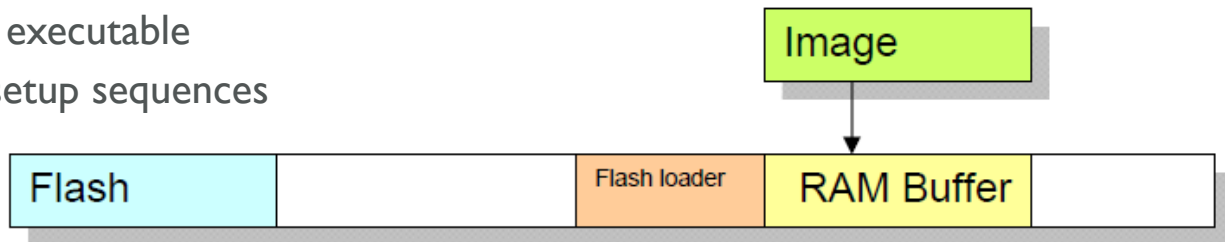
# IAR EW-ARM CMSIS-Pack Support - Schedule

- March 2016 – preview release to partners
- June 2016 - EWARM 7.70 public release



# CMSIS-Pack – Adoption of IAR Flash Loader

- CMSIS 5.0 is planned to use IAR flash loader framework; based on 4 files:
  - .board (xml) - describes all flash memory devices in a system
  - .flash (xml) - describes the flash memory layout & properties
  - .out (ELF) - the flash loader executable
  - .mac (script) - startup/exit setup sequences



- Features:
  - Gap handling/protection
  - Argument handling (for example --clock 14746)
  - Offset handling (absolute/relative to ELF image)
  - Runtime override of all flash memory parameters (good for NAND)
  - Runtime redirect to another flash loader

# CMSIS-Driver

## Status and Validation Suite



Johannes Bauer  
Product Manager

CMSIS Version 5 Partner Meeting, Embedded World  
23. February 2016

## Overview

The CMSIS-Driver specification is a software API that describes peripheral driver interfaces for middleware stacks and user applications. The CMSIS-Driver API is designed to be generic and independent of a specific RTOS making it reusable across a wide range of supported microcontroller devices. The CMSIS-Driver API covers a wide range of use cases for the supported peripheral types, but can not take every potential use-case into account. Over time, it is indented to extend the CMSIS-Driver API with further groups to cover new use-cases.

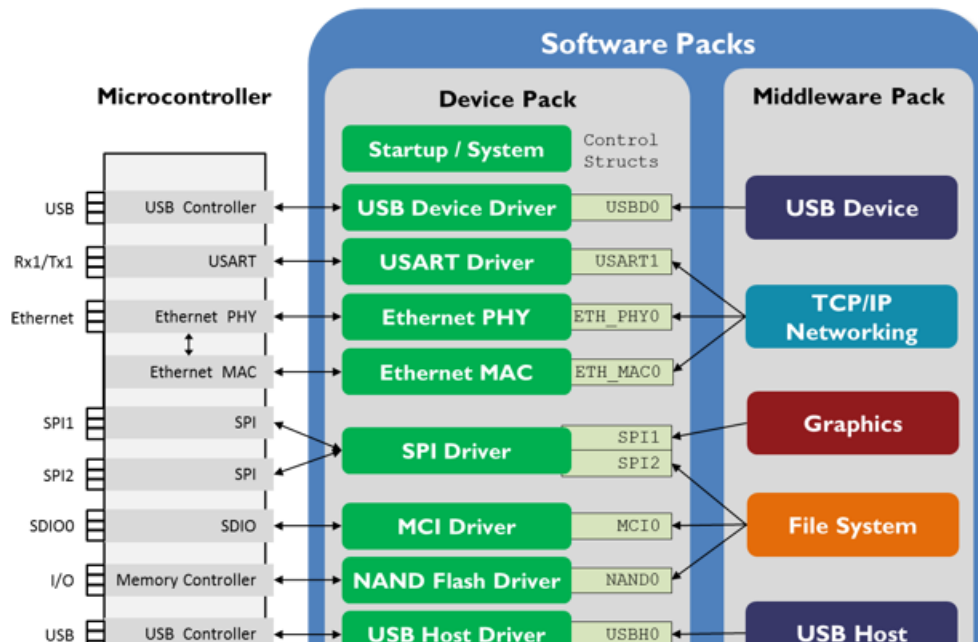
- Overview
- Revision History of CMSIS-Driver
- Theory of Operation
- Reference Implementation
- Driver Validation
  - Sample Test Output
  - Setup for Loop Back Communication

- Reference
  - Common Driver Definitions

- CAN Interface
- Ethernet Interface
- I2C Interface
- MCI Interface
- NAND Interface
- Flash Interface
- SAI Interface
- SPI Interface
- USART Interface
- USB Interface

**API Documentation**

- Data Structures
- Data Structure Index
- Data Fields



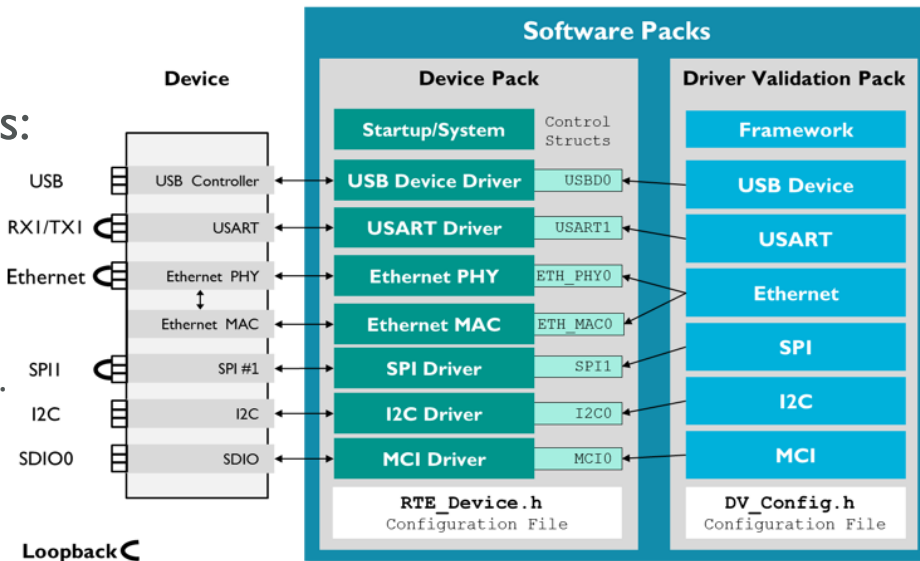
# CMSIS-Driver – Validation Suite

## Validation of CMSIS-Driver using loop-back communication

- Verify CMSIS-Driver with configuration in user hardware
  - Very popular Software Pack (Rank #5 in download list)
  - CAN Verification will come soon

## CMSIS-Driver validation tests and verifies:

- API interacting using driver capabilities with valid and invalid parameters
- Data Communication with various transfer sizes and communication parameters (i.e. baudrate).
- Loopback communication test the underlying hardware.
- Transfer speed of the data communication with time measurement of data transfers.

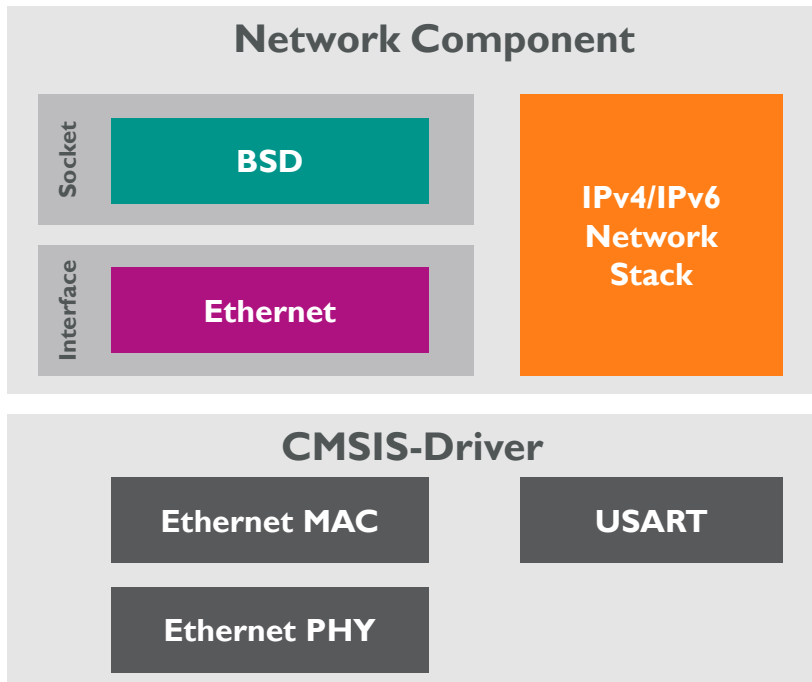


# CMSIS-Driver – What's next

- Team works on new CMSIS-Driver Interfaces for
  - Hardware Random Number Generator (for TLS/SSL)
  - Hardware AES acceleration (for TLS/SSL)
  - Time Functions (for TLS/SSL, File System, etc.)
  - Storage Functions (for Flash Programming and Firmware Update)
- Implementations will be available on GitHub for collaboration
  - NXP: Kinetis K64F, LPC1700, LPC1800, LPC4300
  - Silicon Labs EFM32 Giant Gecko
  - ST Microelectronics STM32F1, STM32F2, STM32F4, STM32F7
  - Other implementations are vendor driven (Infineon, Atmel)

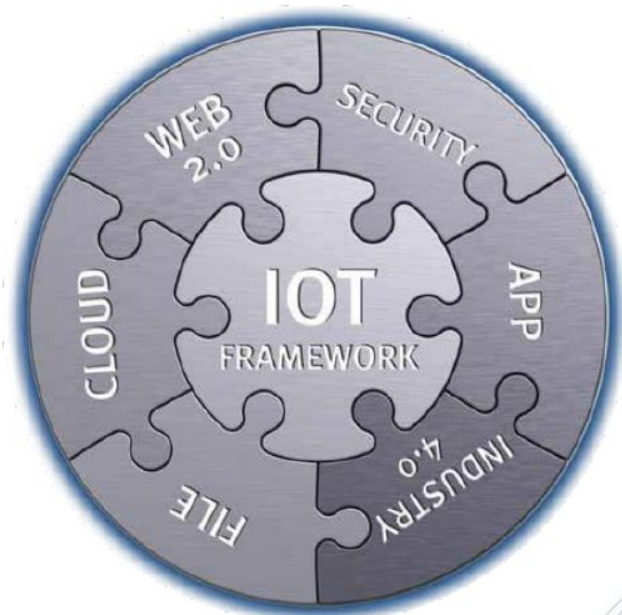
# CMSIS-Driver – Use Case Example

- CMSIS-Driver enable re-use of complex software components



- Example: Network Component
- Provides BSD socket
- Requires:
  - Ethernet (for communication)
  - USART (for debug output)
  - CMSIS-RTOS (for scheduling)
- CMSIS-Driver make software portable
  - to devices that provide CMSIS-Driver
  - It is easy to change Ethernet PHY

## Your IoT Solutions Provider



**SOFTWARE made in Germany  
for more than 15 years**

### ■ Embedded IoT Software Stacks

- TCP, IPv4 & IPv6, UDP
- Webservices, AJAX/JSON, REST, SOAP
- TLS 1.2
- Apple HomeKit, NEST, IFTTT, EEBUS
- ModbusTCP, EthernetIP, PTP, etc.
- Autosar software modules

### ■ Cloud & App & Security

- Smartphone Application Framework
- Webapplication Frameworks
- Secure Cloud Connectivity
- PKI Solutions
- Security Consulting



# SEVENSTAX CMSIS Evaluation Packs



## Tiny Pack

minimum  
RAM/ROM consumption



## Cloud Pack

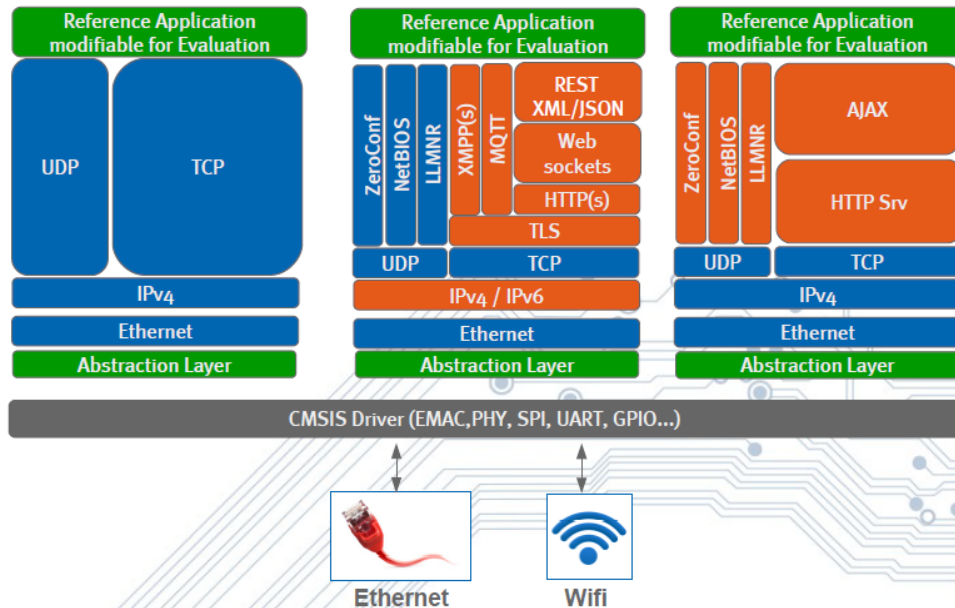
Secure Cloud access



## APP Pack

Access per Browser  
and Smartphone App

- Source Code module may be modified for Evaluation purpose
- Precompiled library



iEmbedded Control  
App Development  
Framework already available  
in App Store and Play Store



Customer specific  
evaluation packs  
on request!

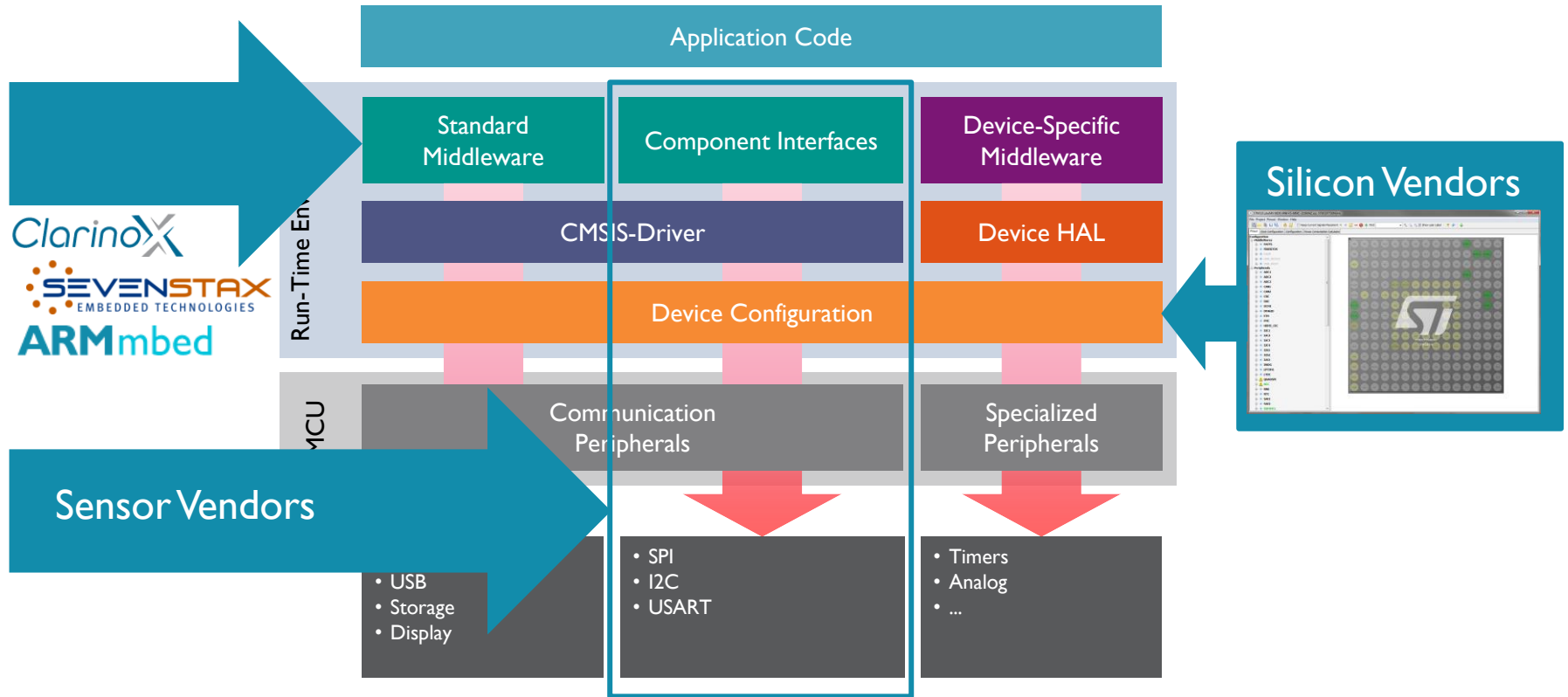
# Eco-System Collaboration



Reinhard Keil  
Director Microcontroller Tools

CMSIS Version 5 Partner Meeting, Embedded World  
23. February 2016

# ARM Eco-System Collaboration



# ARM

# World's No. 1 Embedded Ecosystem

Thank you.....now it's time for drinks and side discussions...

The trademarks featured in this presentation are registered and/or unregistered trademarks of ARM Limited (or its subsidiaries) in the EU and/or elsewhere. All rights reserved. All other marks featured may be trademarks of their respective owners.

Copyright © 2016 ARM Limited