

# AUTOSAR MCAL for i.MX 8 Application Processors

Salam Zeidan

Sr. Project Manager

MICR Advanced Technologies

AMF-AUT-T3902



SECURE CONNECTIONS  
FOR A SMARTER WORLD

# Agenda

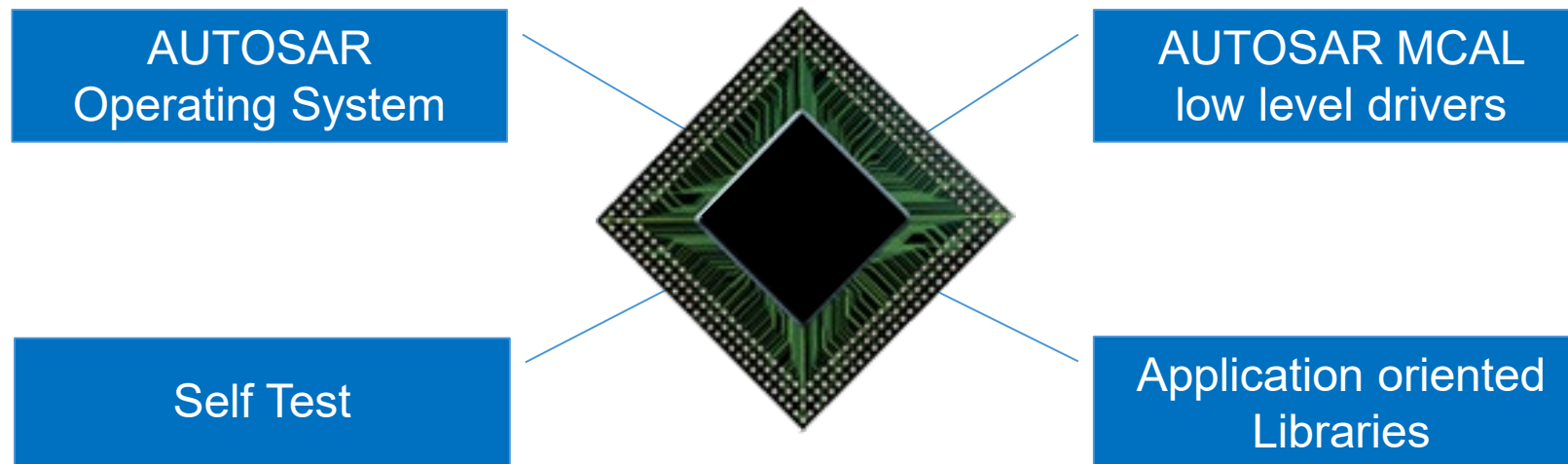
---

- General Information
- i.MX 8 AUTOSAR MCALs:
  - Implementation For All Derivatives In i.MX 8/8X Families
  - Example Startup
  - Configuration Tool View
  - Microcontroller Drivers
  - I/O Drivers
  - Communications Drivers
  - Memory Drivers
  - Crypto Drivers
  - Complex Drivers

# General Information



# i.MX AUTOSAR Solution



# AUTOSAR – NXP Automotive SW

AUTOSAR (AUTomotive Open System ARchitecture) Org. aims to improve complexity management of integrated E/E architectures through increased reuse and exchangeability of SW modules between OEMs and suppliers. The essential means is the standardization of the software architecture of ECUs.



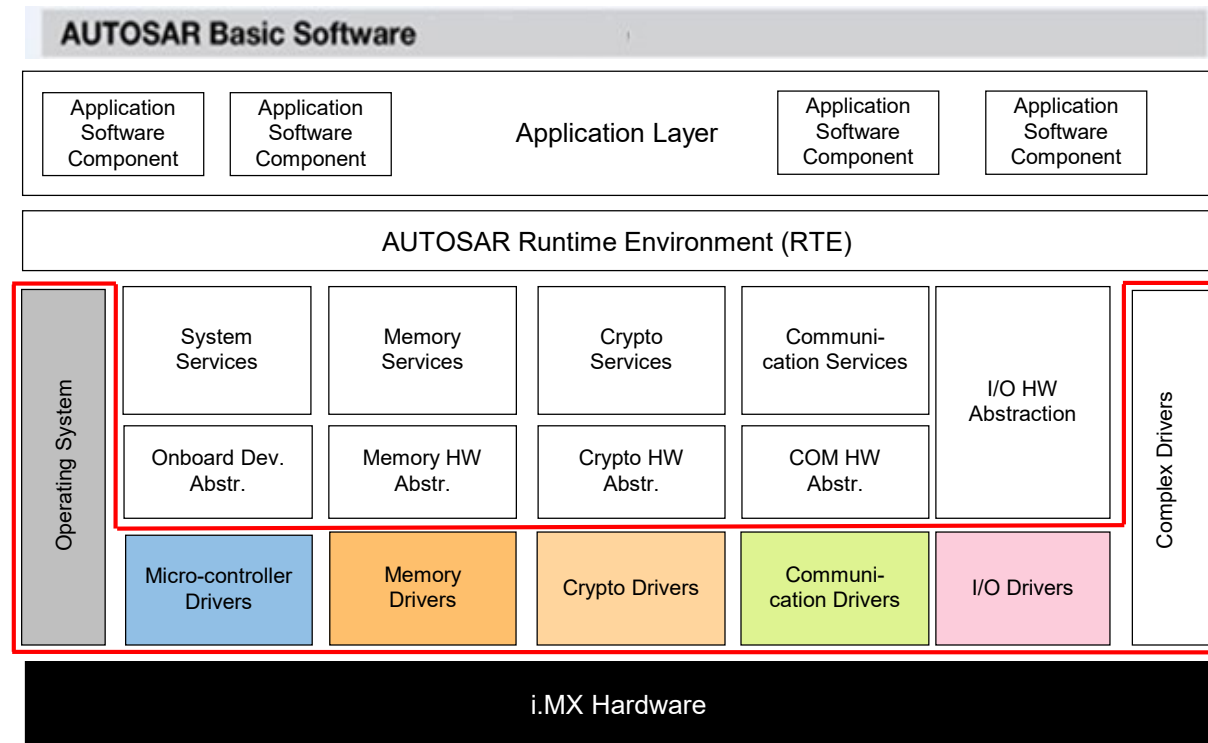
**PREMIUM  
PARTNER**

Design and use the AUTOSAR  
standards

As a premium member of the AUTOSAR partnership NXP continues to develop its AUTOSAR SW offering across its product lines for use in automotive applications

# i.MX 8 AUTOSAR Classic Platform (CP) Product

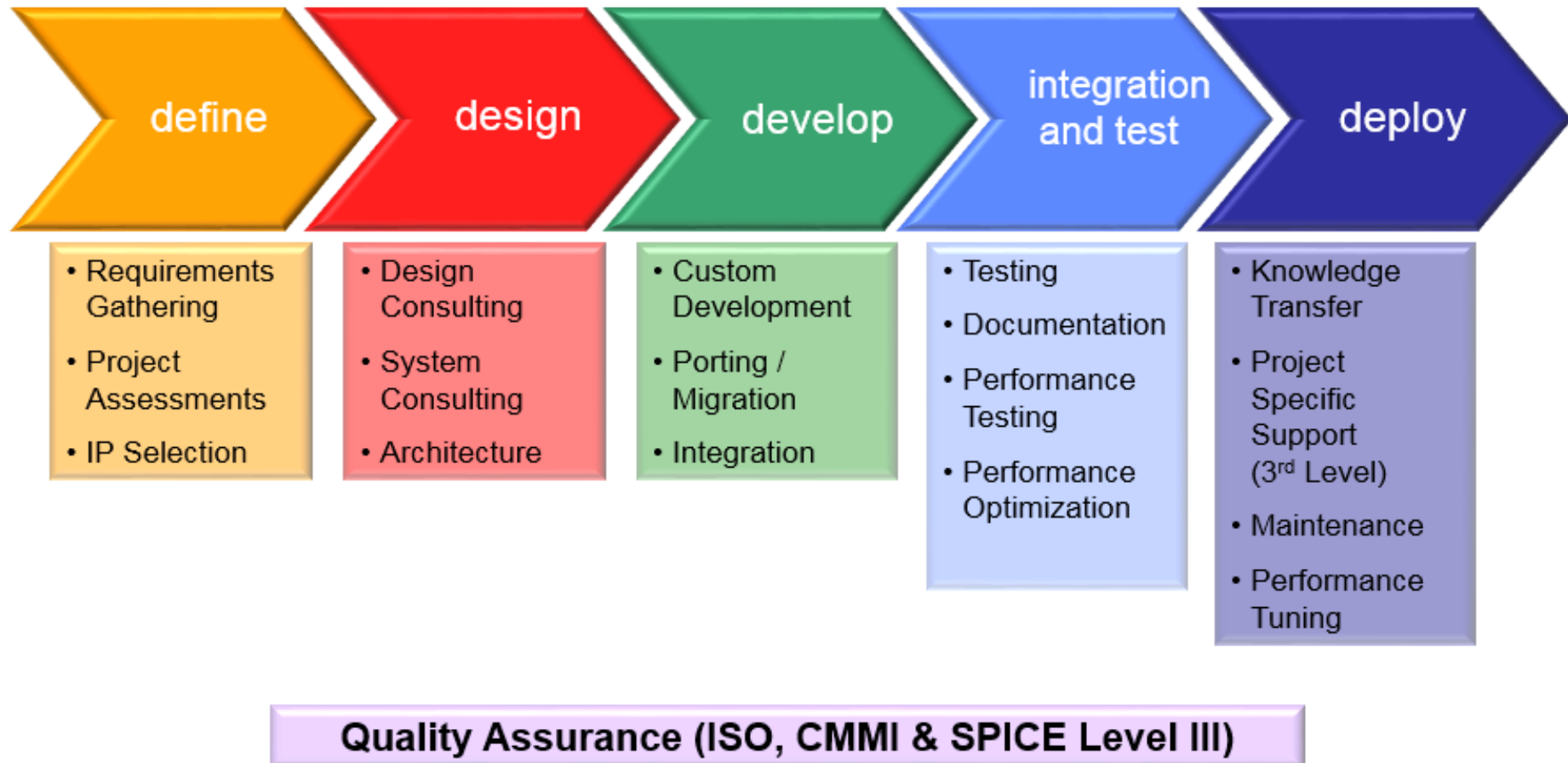
## i.MX 8 AUTOSAR OS, MCALs, and Complex Drivers



# AUTOSAR Software Release Framework Overview

- **BETA Release (Select Customers)/ PRC (Product Release Candidate)**
  - Includes all MCAL Drivers (Feature Complete) fully verified and documented
  - Includes Integration Testing
  - Complete Quality Package: on-request delivery
  - Code coverage: generated
  - No open S1 and S2 defects
- **Release to Market Candidate – RTMC (All Customers)/ RFP (Relapse for Production)**
  - Beta criteria +
  - 100% statement coverage, branch coverage for non-configurable SW
  - 80% statement coverage, branch coverage for configurable SW
  - Complete Quality Package: externally delivery
  - No open S1 and S2 defects
- **Each AUTOSAR Classic Platform MCAL version will have an individual release**

# MCAL Software Life Cycle Methodologies & Quality: Engineering Discipline





# MCAL Quality Assurance

- Quality Assurance Plan
- QA audit, Assessment (TS16949, ASPICE, ISO26262)
- NXP process compliance audit
- Release Readiness review
- Metrics collection and tracking( Quality Metric, KPI...)
- Quality Management System
- Customer satisfaction monitoring

# MCAL Safety Management

## **Safety Management:**

We follow the ISO26262 part that address SW development. In line with implicit safety for MCAL part 10 Safety Element out of Context (SEooC.)

*ISO26262\_SW\_SEooC\_SMCAL4.2\_i.MX8\_Safety\_Case*

Define the tailoring of ISO 26262 Work Products for a SW SEooC product development in conjunction with the safety plan, as well as progressively compile the deliverables generated during the safety lifecycle which form the safety case

*ISO26262 (S)MCAL Development Sign-Off*

*Technical safety concept*

*Safety analyses: ASIL- and safety- oriented analysis*

# Complex Drivers/Services

Autosar Complex Drivers / Custom Software  
Development, Migration, Testing, Integration, Consulting

## Complex Driver Development

- Automotive Software Services Engagement Process:
  - Gather Customer Requirements
  - Scope the Engagement Effort
  - Create SOW Proposal for Review and Approval
  - Execute the Project and Test the Deliverables per the Agreed Test Plan
  - Deliver on Time and on Budget

## Autosar Service Offerings

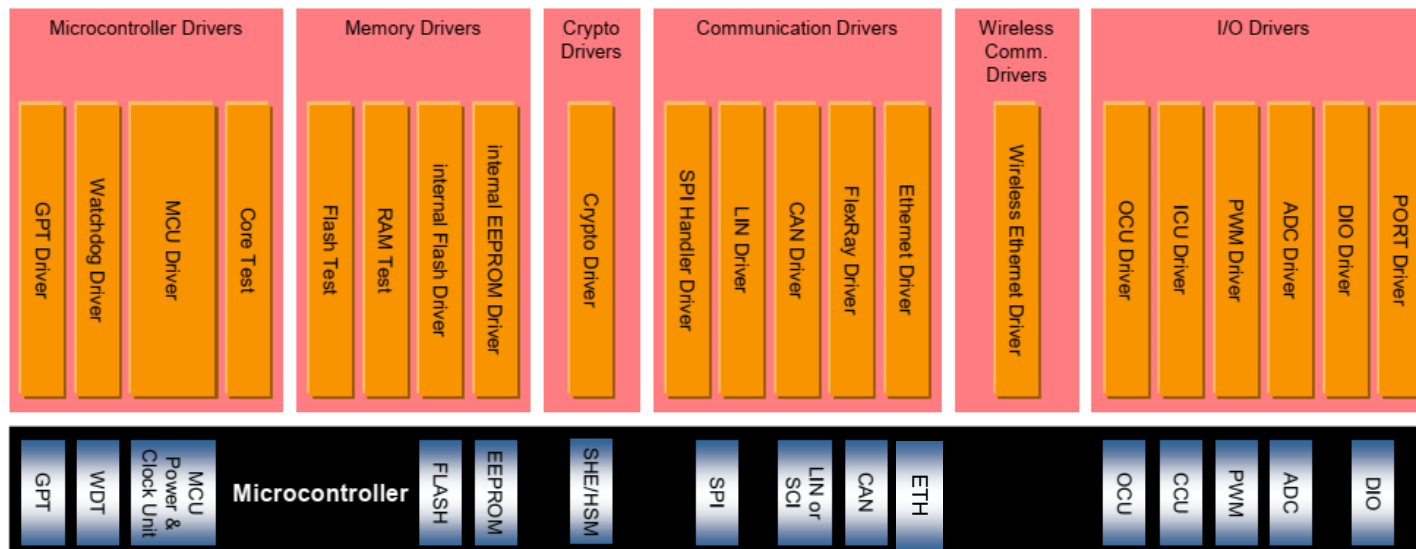
- System Consulting
- Integration
- Configuration
- Custom Development
- Migration
- Modeling and Automatic Code Generation
- Performance Analysis & Optimization

# AUTOSAR MCALs for i.MX 8Series



# i.MX 8 AUTOSAR MCALs Product

- Autosar 4.2/4.3 MCAL: Tested running from RAM using NXP MEK/EVB
- FlexRay, and WDG-External can be provided as a additional drivers Services
- Core Test, Flash Test, or RAM Tests can be provided as a additional drivers Services
- All components configurable in any AUTOSAR-compliant configuration tool
- Configuration Tool EB tresos Studio™ and plugins are part of the product
- MCALs also been deployed using Vector DaVinci Configuration Tool



# i.MX8 AUTOSAR MCAL Roadmap

Compilers	IMX8_ASR4.2_RTM_2.0.0							
	IMX8X (QXP,DX/SXL/DXL,DXP)					IMX8 (QM,QP,DM)		
	QXP	DX	SXL	DXL	DXP	QM	QP	DM
GHS	RTM Available Now	RTM Available Now	Beta 2weeks from Silicon availability	Beta 2weeks from Silicon availability	RTM Available Now	RTM Available Now	RTM Available Now	Beta 2weeks from Silicon availability
GCC	RTM Available Now	RTM Available Now	Beta 2weeks from Silicon availability	Beta 2weeks from Silicon availability	RTM Available Now	RTM Available Now	RTM Available Now	Beta 2weeks from Silicon availability
Diab	RTM Available Now	RTM Available Now	Beta 2weeks from Silicon availability	Beta 2weeks from Silicon availability	RTM Available Now	RTM Available Now	RTM Available Now	Beta 2weeks from Silicon availability
DS5	RTM Available Now	RTM Available Now	Beta 2weeks from Silicon availability	Beta 2weeks from Silicon availability	RTM Available Now	RTM Available Now	RTM Available Now	Beta 2weeks from Silicon availability
Compilers	IMX8_ASR4.3_RTM_2.0.0							
	IMX8X (QXP,DX,DXP)					IMX8 (QM,QP,DM)		
	QXP	DX	SXL	DXL	DXP	QM	QP	DM
GHS	RTM Available Now	RTM Available Now	Beta 2weeks from Silicon availability	Beta 2weeks from Silicon availability	RTM Available Now	RTM Available Now	RTM Available Now	Beta 2weeks from Silicon availability
GCC	RTM Available Now	RTM Available Now	Beta 2weeks from Silicon availability	Beta 2weeks from Silicon availability	RTM Available Now	RTM Available Now	RTM Available Now	Beta 2weeks from Silicon availability
Diab	Per Customer Request	Per Customer Request	Per Customer Request	Per Customer Request	Per Customer Request	Per Customer Request	Per Customer Request	Per Customer Request
DS5	Per Customer Request	Per Customer Request	Per Customer Request	Per Customer Request	Per Customer Request	Per Customer Request	Per Customer Request	Per Customer Request

	RTM Available Now
	Per Customer Request
	Beta 2weeks from Silicon availability

# Licensing Options

License	Description	Support	Upgrade options
Project License	One NXP Target Part only for one Customer Target Project. One OS.	1 year support included, 20% of the license price year 2 and beyond.	To any higher priced production license. Additional OS/SoC at a discount.
Product Line License	One NXP Target Part only for one Customer Product Line (ex. Cluster), One OS	1 year support included, 20% of the license price year 2 and beyond.	To any higher priced production license. Additional OS/SoC at a discount.
Family Multi-Project License	One NXP Target Part Family, (ex. i.MX6 Family,), only for Customer Target Project or Customer Product Line, no restrictions. One OS	1 year support included, 20% of the license price year 2 and beyond.	Additional OS/SoC at a discount.

# AUTOSAR Software i.MX 8 Product Selection Matrix Worksheet

MCAL - i.MX		Selection	
Family	i.MX6/ i.MX8		
Processor/Derivative Part	Q/D/SoloX/DL/S/SL/UL/QM/QXP/DXP/....		
Core	A72/A53/A35/A9/M4		
Standard Modules	I/O Modules	Port	Yes
		Dio	Yes
		Icu	Yes
		Ocu	Yes
		Pwm	Yes
		Adc	Yes
	Comm Modules	Spi	Yes
		Can	Yes
		Lin	Yes
	Mcu Modules	Eth	Yes
		Mcu	Yes
		Gpt	Yes
		Fee	Yes
		Fls FlexSPI	Yes
	Crypto	Wdg Int.	Yes
		Crypto	Yes

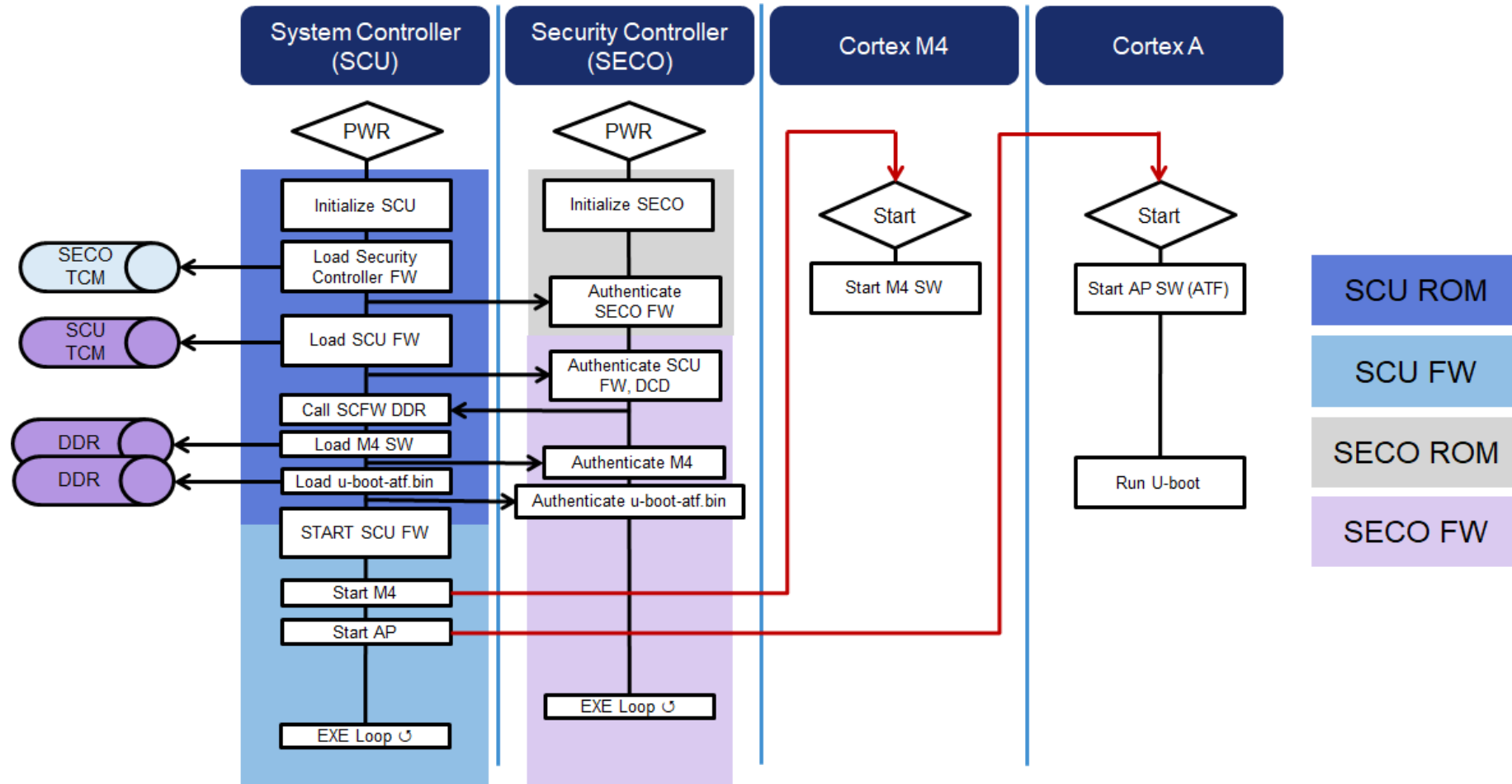
MCAL - i.MX		Selection
Family	i.MX6/ i.MX8	
Processor/Derivative Part	Q/D/SoloX/DL/S/SL/UL/QM/QXP/DXP/....	
Core	A72/A53/A35/A9/M4	
Complex Drivers	I2c	Yes
	eMMC	Yes
	Esai	Yes
	Rpmsg	Yes
	Uart	Yes
	Per customer specific request	
Compilers	GHS	
	GCC	
	DSS	
	Diab	
AutoSAR Revisions	4.0.3	
	4.2.2	
	4.3.0	
Functional Safety	ASIL A	
	ASIL B	



# i.MX 8 MCAL Startup Example



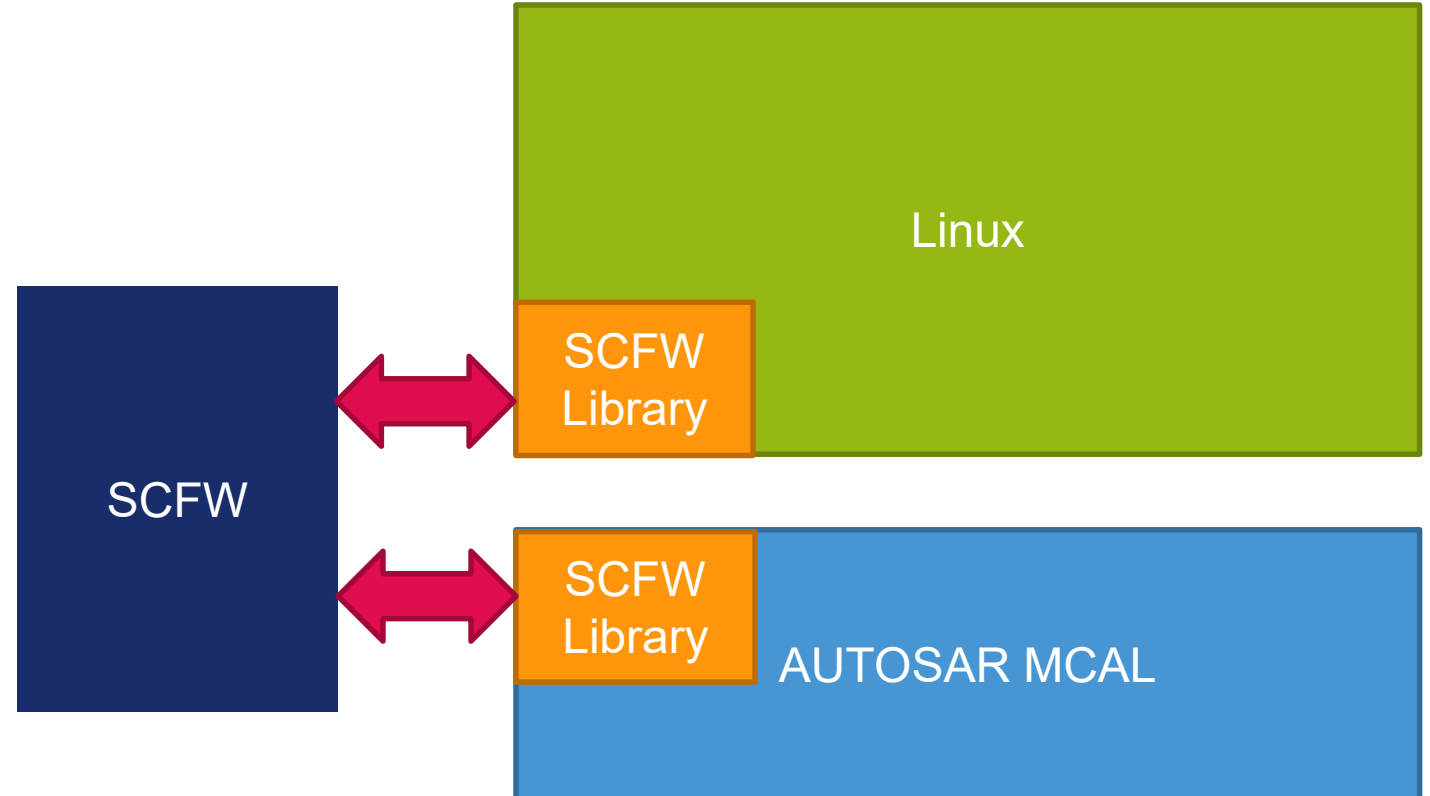
# i.MX 8 Secure Boot Flow



# Background – System Controller Firmware (SCFW)

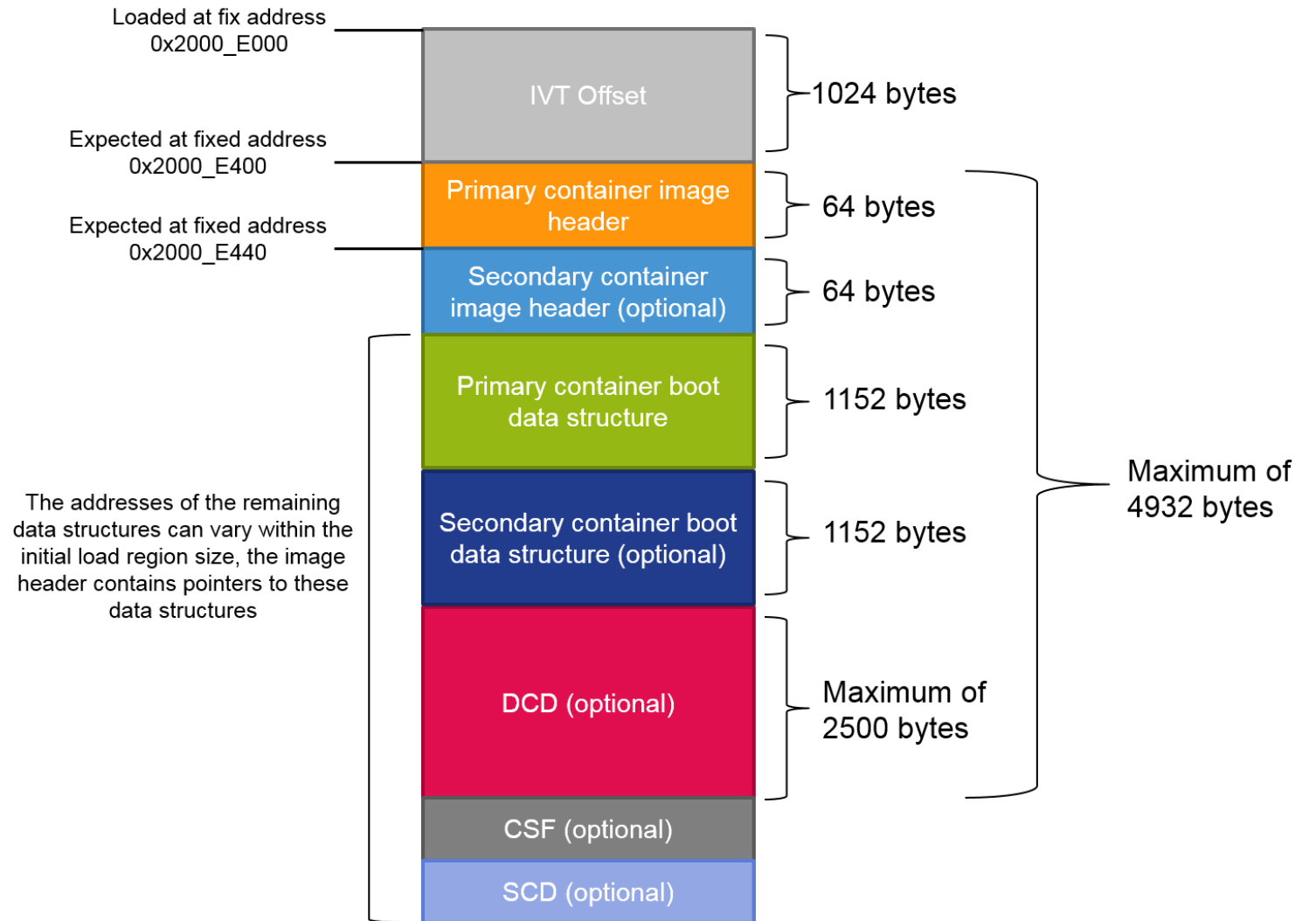
The System Controller Firmware (SCFW) runs on the System Controller Unit and is responsible for managing requests from other cores in the system.

MCAL software components communicate with System Controller Unit (SCU) via an Application Programming Interface (API) library.



# Initial Loading Region Layout

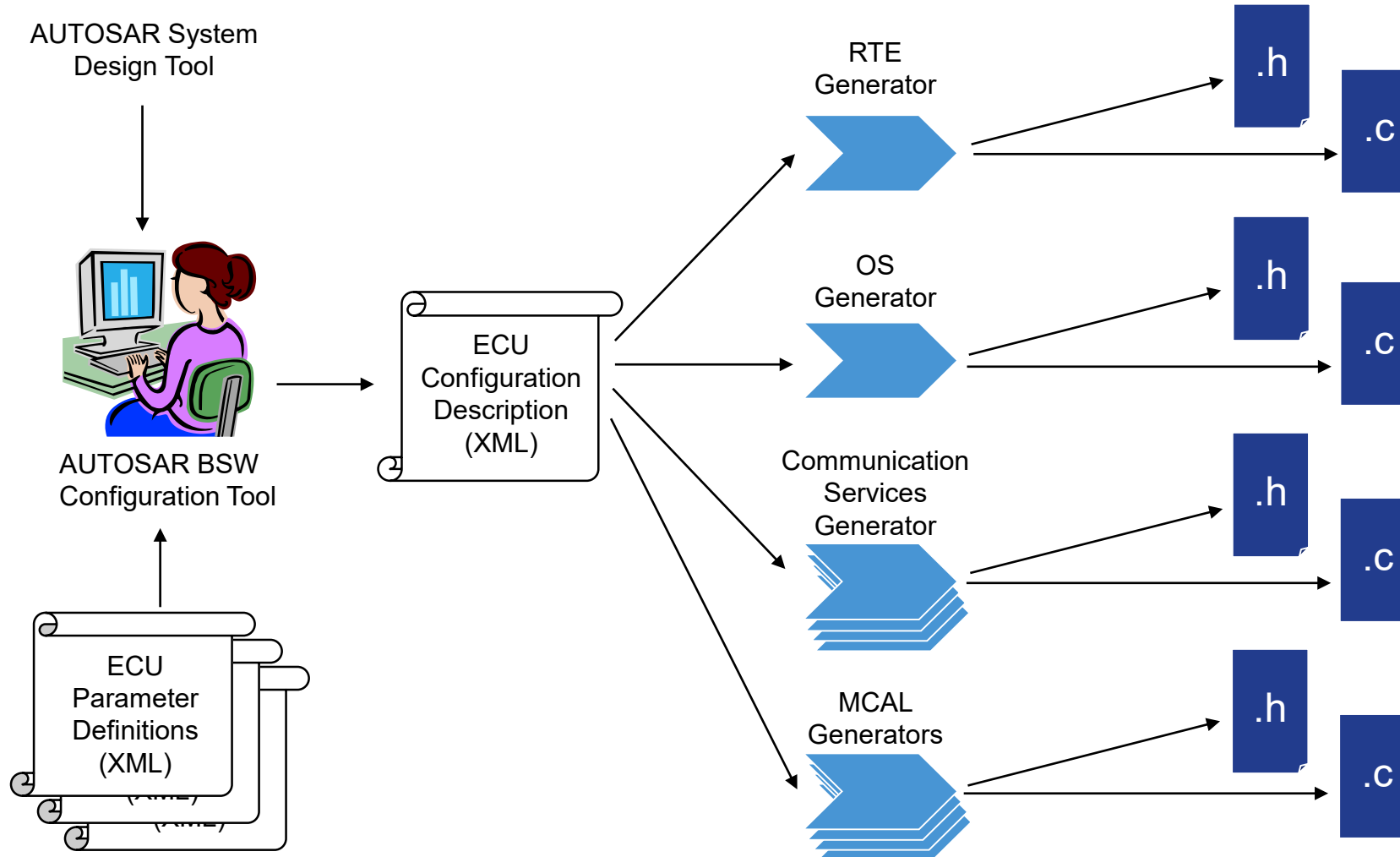
- The SCU Boot ROM loads an initial region from the selected boot media.
- This region contains all the information the device needs to continue the boot process.



# MCAL BSW Configuration Tool



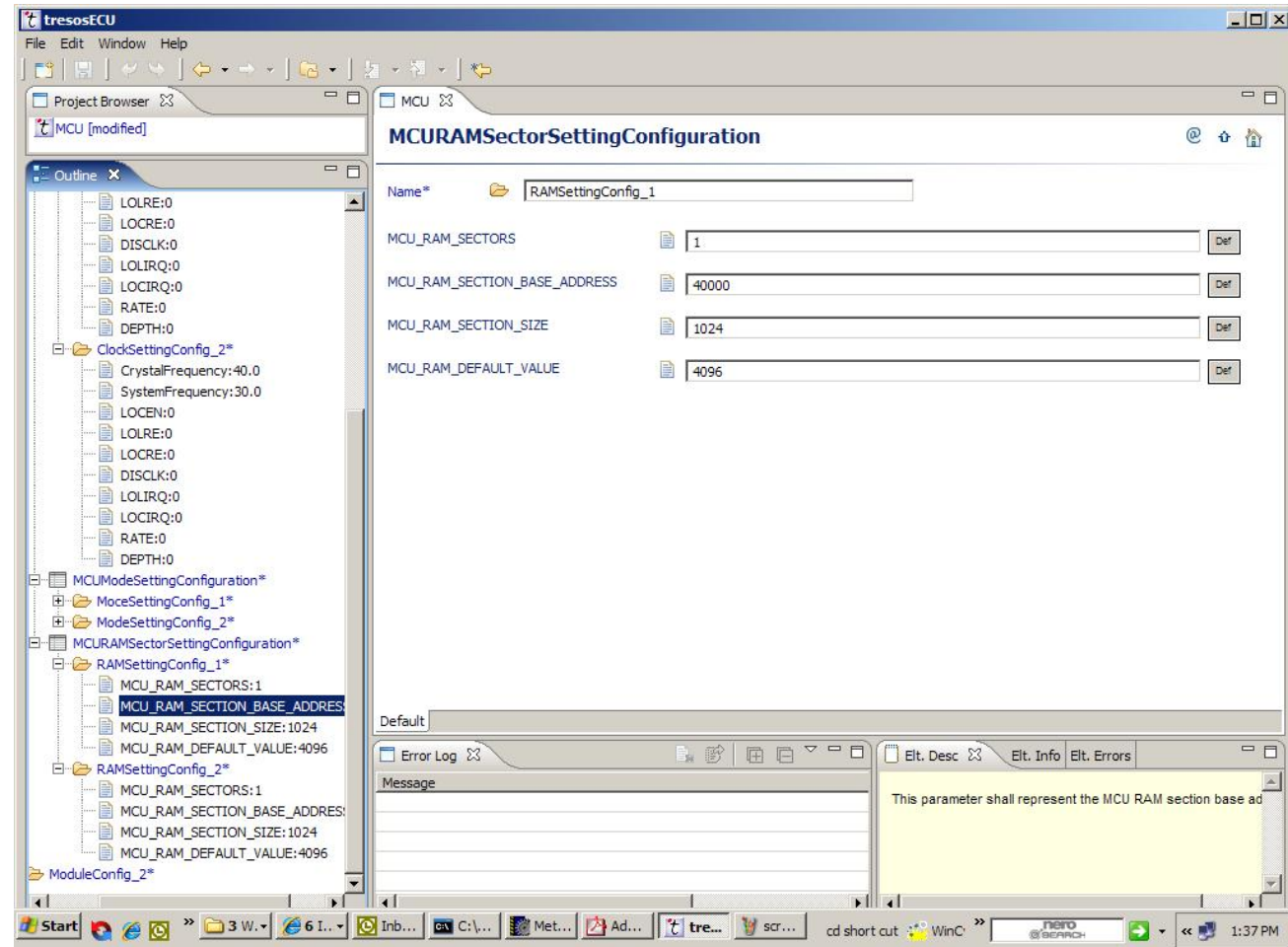
# Basic Software Configuration Process



# AUTOSAR BSW Configuration Tool

## Example: Tresos® ECU

- Graphical representation of ECU configuration description (ECD)
- Import/export of ECD
- Easy configuration of AUTOSAR BSW using pre-compile methodology



# i.MX 8 MCAL Microcontroller Drivers

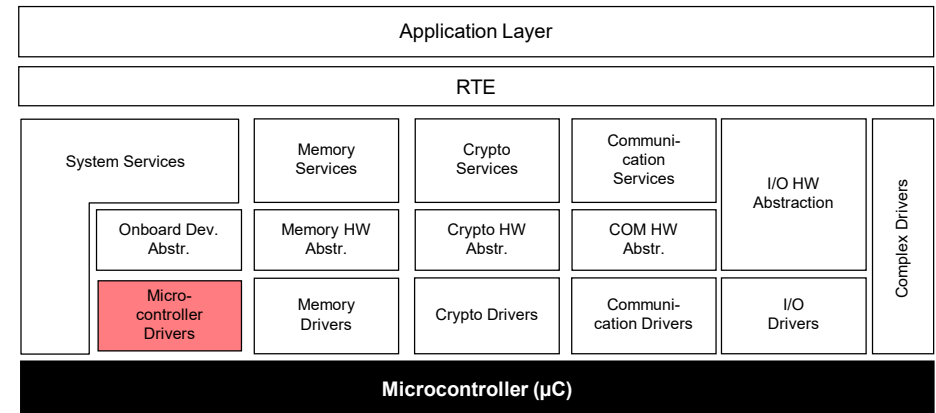
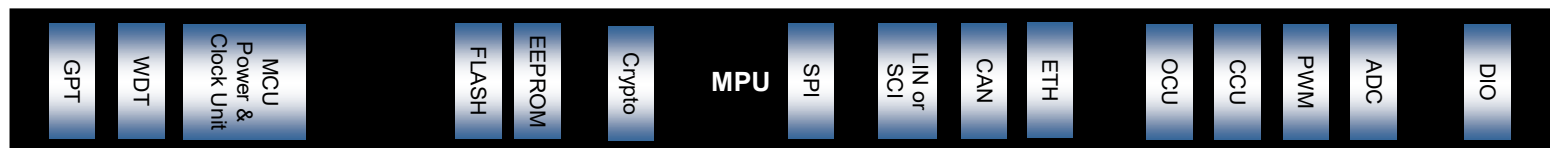
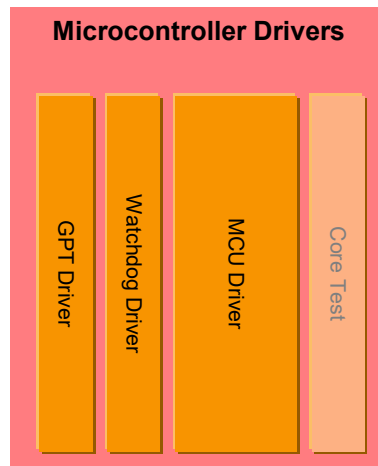




# i.MX MCALs— Microcontroller Abstraction Layer

- **Microcontroller Drivers**

- Drivers for internal peripherals (e.g. Watchdog, General Purpose Timer)
- Functions with direct  $\mu$ C access



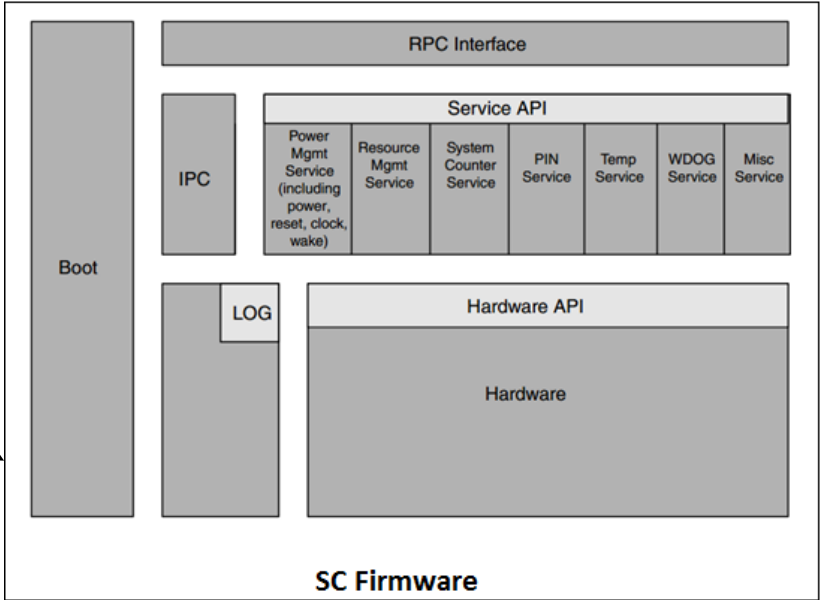
Source: AUTOSAR



# MCU Module – NXP Implementation

**MCU implementation:**  
Mcu\_Init (...)  
Mcu\_InitRamSection (...)  
Mcu\_InitClock (...)  
Mcu\_GetResetReason (...)  
Mcu\_DistributePIIClock (...)  
Mcu\_GetPIIStatus (...)  
Mcu\_GetResetRawValue (...)  
Mcu\_PerformReset (...)  
Mcu\_SetMode (...)  
Mcu\_GetVersionInfo (...)  
Mcu\_GetRamState (...)

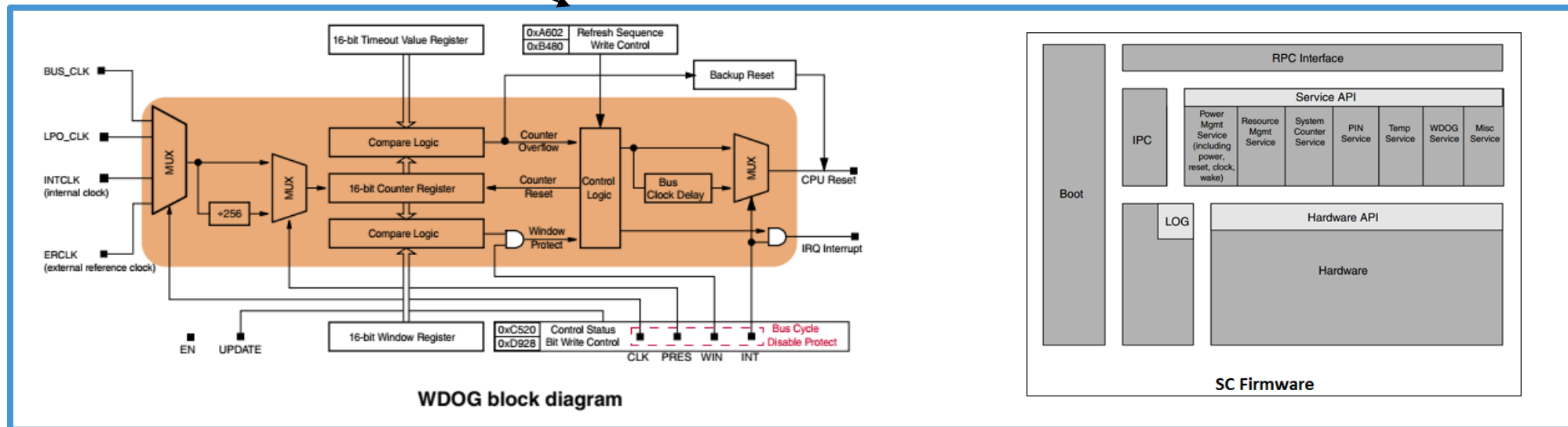
IMX8



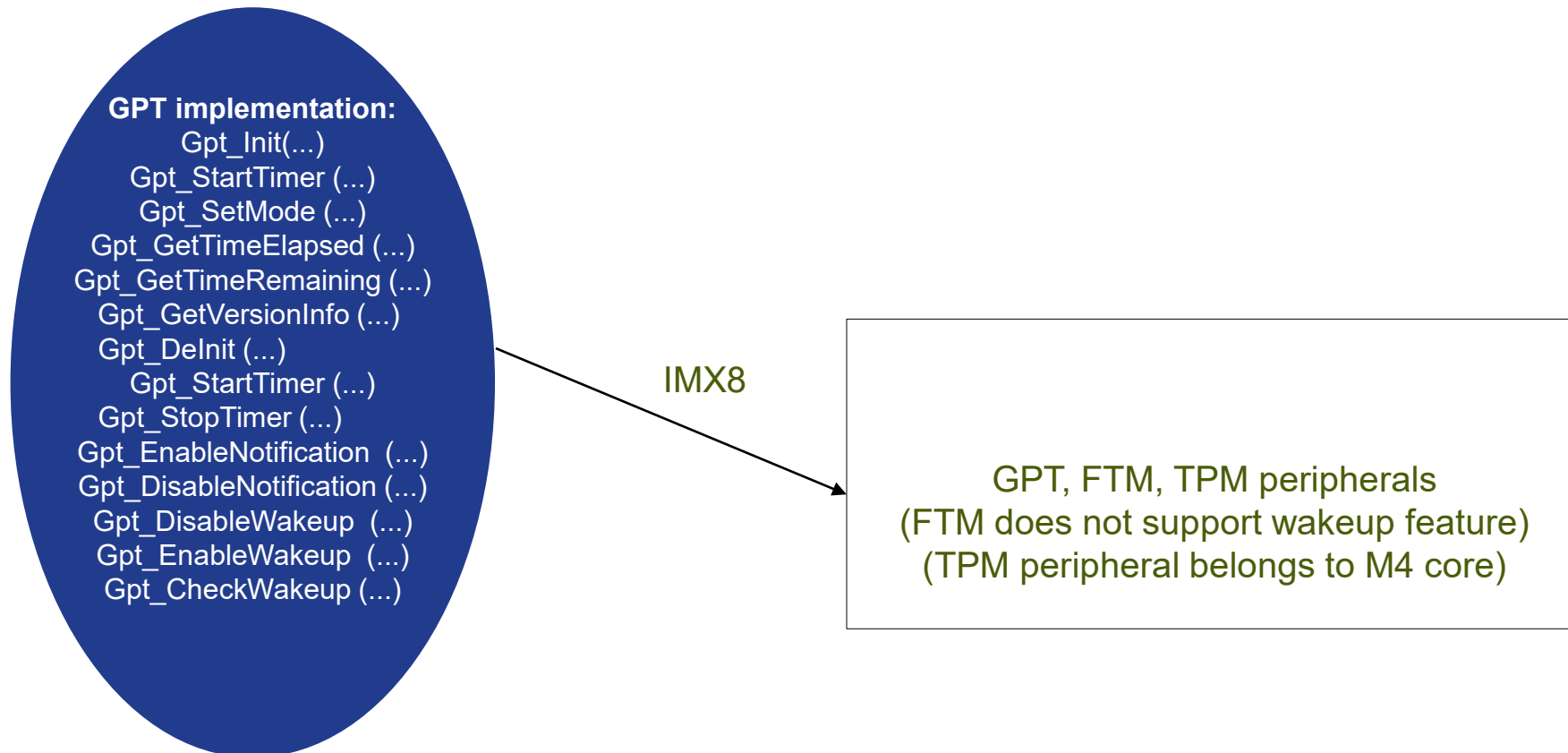
# Watchdog Module – NXP Implementation

Watchdog implementation:  
Wdg\_Init (...)  
Wdg\_SetMode (...)  
Wdg\_SetTriggerCondition(...)  
Wdg\_GetVersionInfo (...)

IMX8



# GPT Module – NXP Implementation



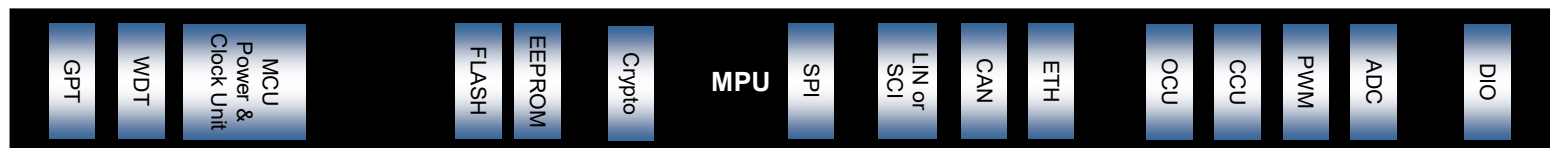
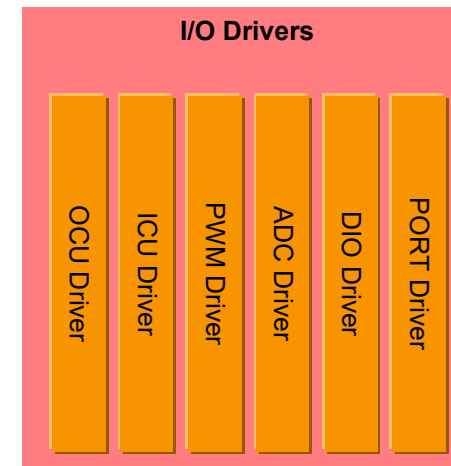
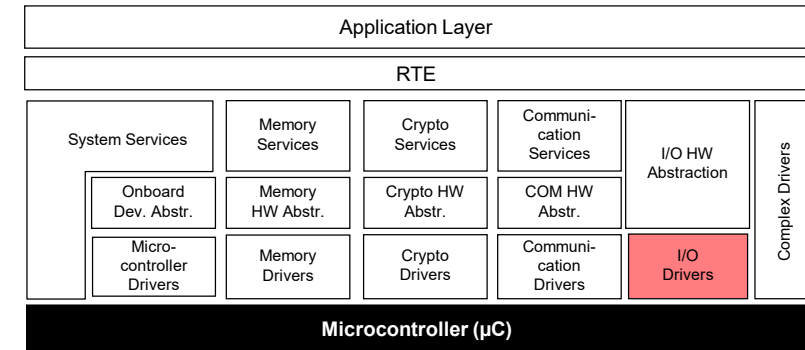
# i.MX 8 MCAL I/O Drivers



# AUTOSAR – Microcontroller Abstraction Layer

- I/O Drivers

- Drivers for analog and digital I/O (e.g. ADC, PWM, DIO)

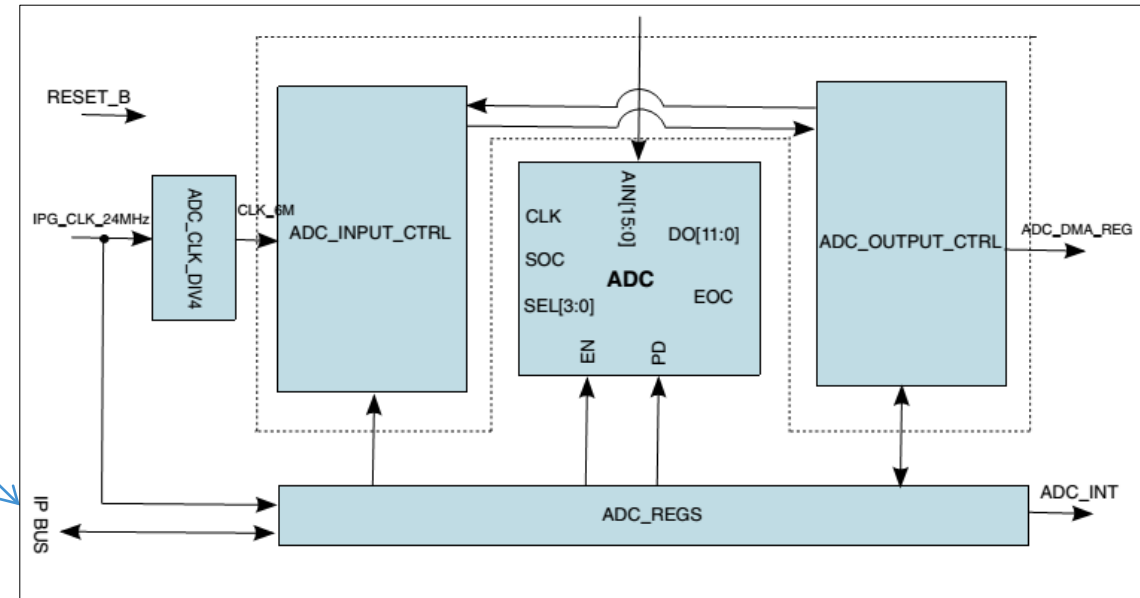


Source: AUTOSAR

# ADC Modules – NXP Implementation

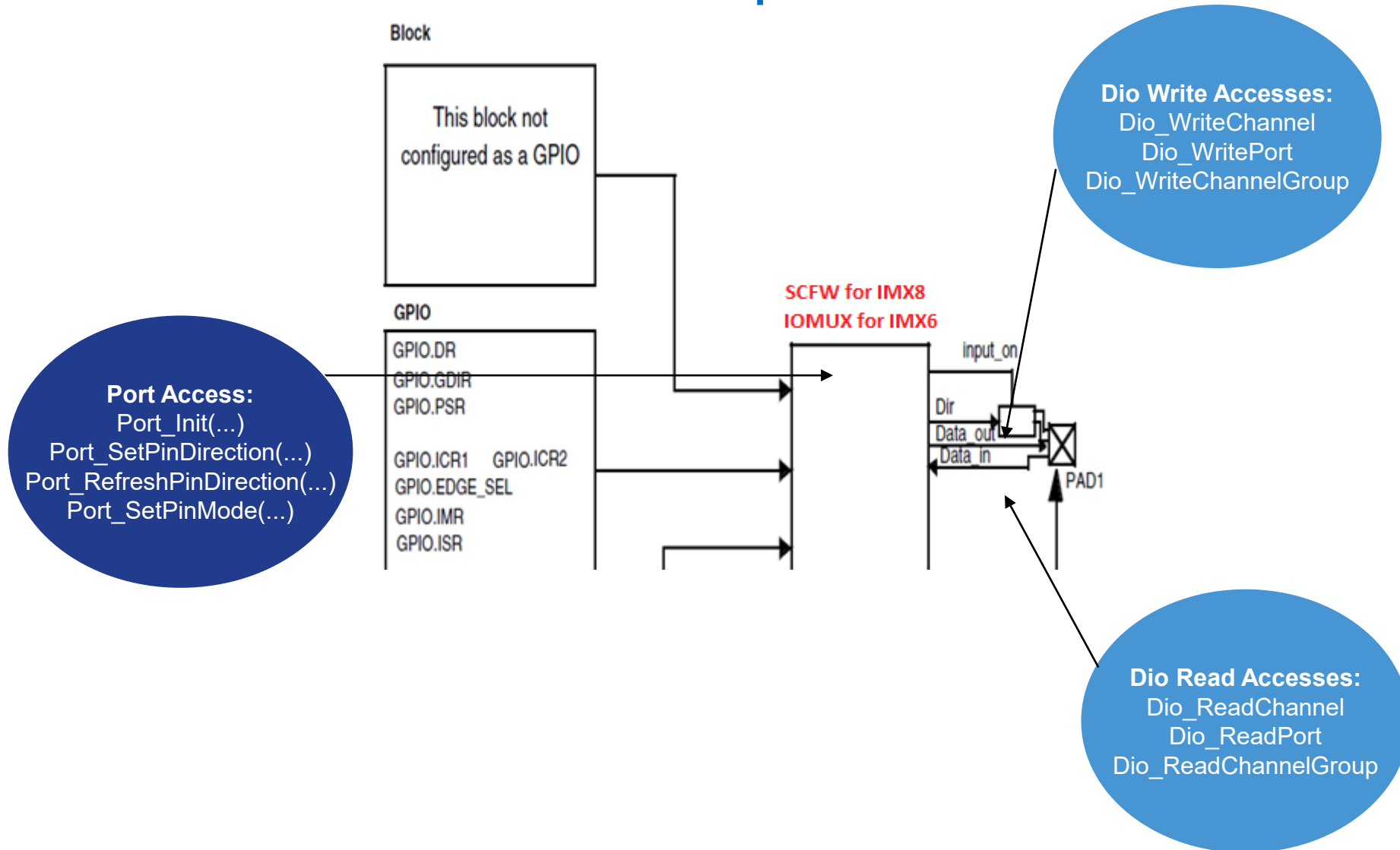
- Adc Access:**
- Adc\_Init(...)
- Adc\_DeInit(...)
- Adc\_StartGroupConversion(...)
- Adc\_StopGroupConversion(...)
- Adc\_ReadGroup (...)
- Adc\_DisableHardwareTrigger (...)
- Adc\_EnableHardwareTrigger (...)
- Adc\_EnableGroupNotification(...)
- Adc\_DisableGroupNotification(...)
- Adc\_StopGroupConversion(...)
- Adc\_GetGroupStatus(...)
- Adc\_GetStreamLastPointer (...)
- Adc\_GetStreamLastPointer (...)
- Adc\_GetVersionInfo(...)
- Adc\_SetPowerState (...)
- Adc\_GetCurrentPowerState (...)
- Adc\_GetTargetPowerState (...)
- Adc\_PreparePowerState (...)

IMX8



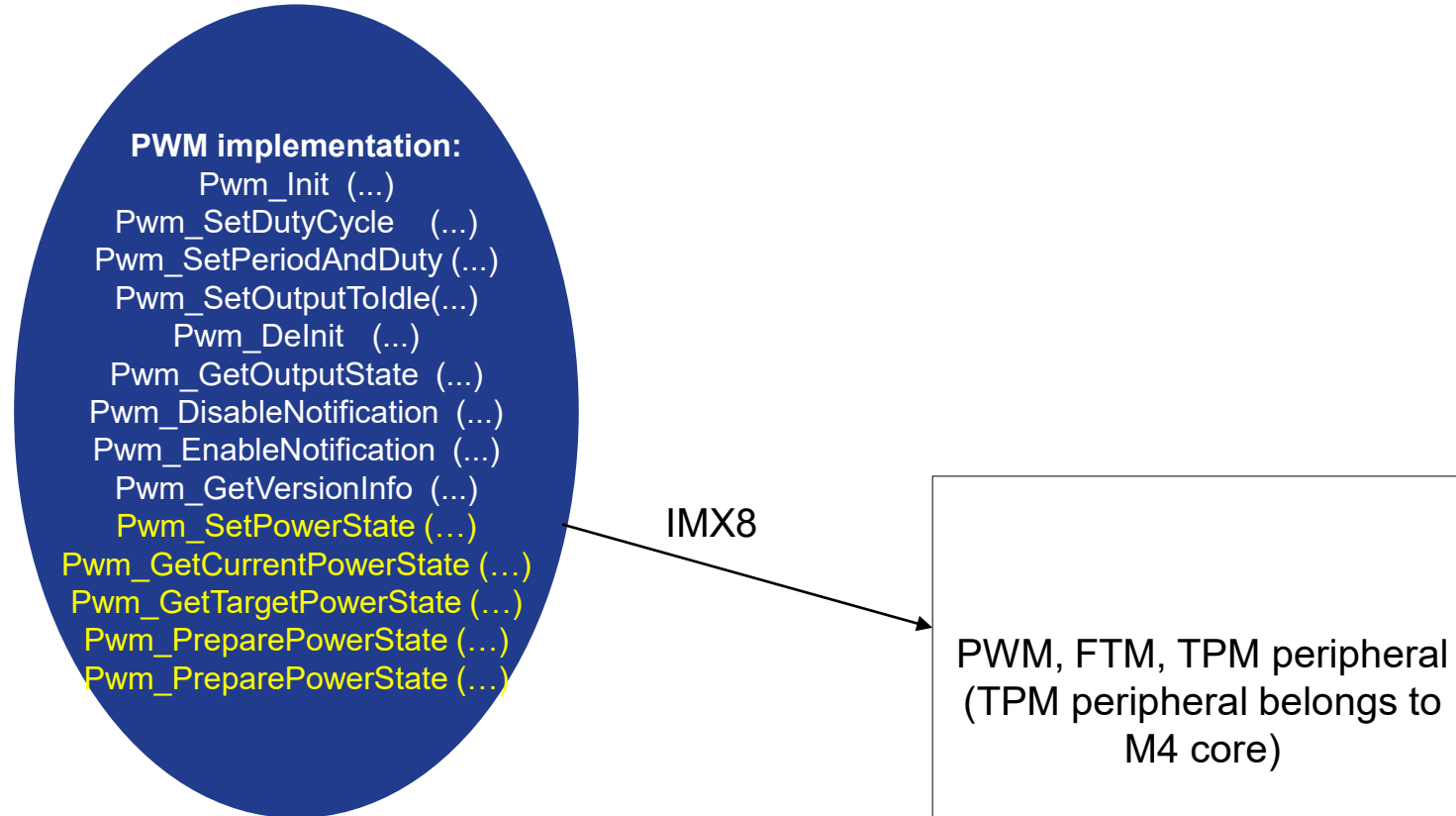
APIs in white are for ASR4.0, ASR4.2 and ASR4.3  
APIs in yellow are new APIs in ASR4.2 and ASR4.3

# PORT/DIO Modules – NXP Implementation





# PWM Module – NXP Implementation



APIs in white are for ASR4.0, ASR4.2 and ASR4.3  
APIs in yellow are new APIs in ASR4.2 and ASR4.3

# ICU Module – NXP Implementation

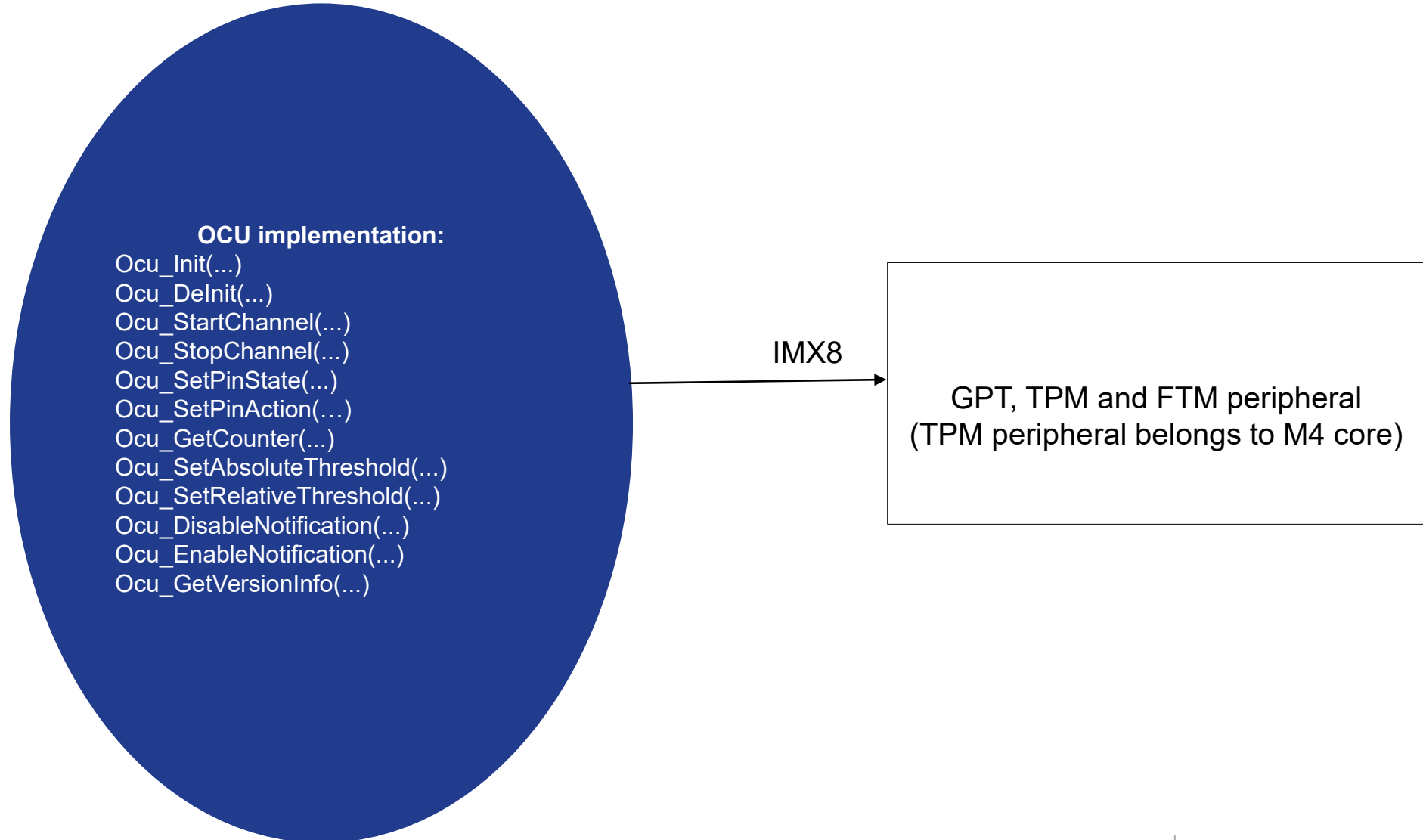
## ICU implementation:

Icu\_Init(...)  
Icu\_DeInit(...)  
Icu\_SetMode(...)  
Icu\_DisableWakeup (...)  
Icu\_EnableWakeup (...)  
Icu\_SetActivationCondition  
Icu\_DisableNotification (...)  
Icu\_EnableNotification (...)  
Icu\_GetInputState (...)  
Icu\_StartTimestamp (...)  
Icu\_StopTimestamp (...)  
Icu\_GetTimestampIndex (...)  
Icu\_ResetEdgeCount (...)  
Icu\_EnableEdgeCount (...)  
Icu\_DisableEdgeCount (...)  
Icu\_GetEdgeNumbers(...)  
Icu\_StartSignalMeasurement (...)  
Icu\_StopSignalMeasurement (...)  
Icu\_GetTimeElapsed (...)  
Icu\_GetDutyCycleValues (...)  
Icu\_GetVersionInfo (...)

IMX8

GPT, TPM and FTM peripheral  
(TPM peripheral belongs to M4 core)

# OCU Module – NXP Implementation



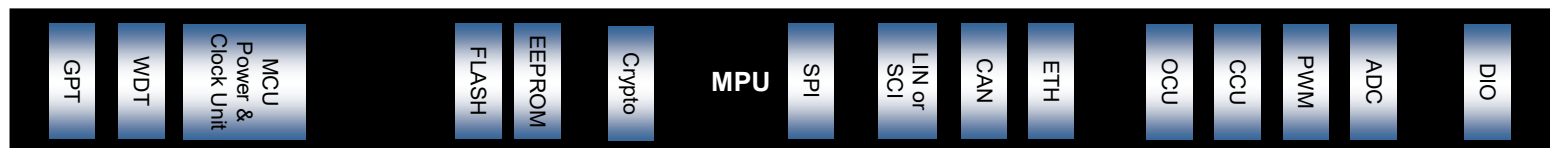
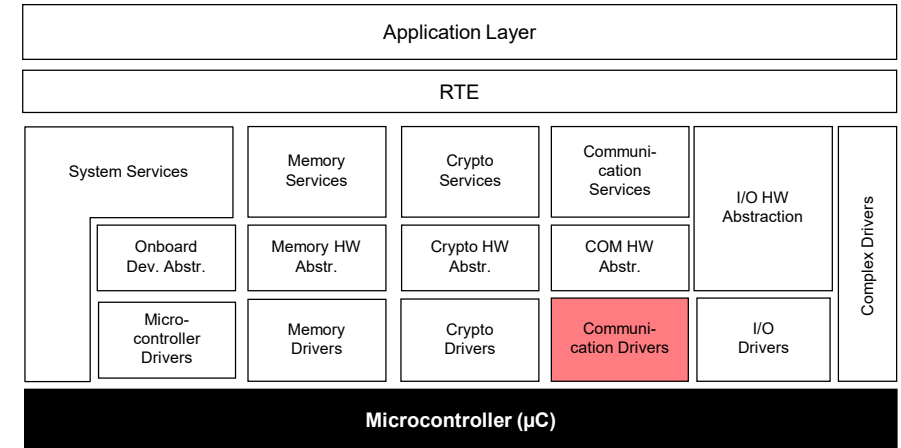
# i.MX 8 MCAL Communication Drivers



# AUTOSAR – Microcontroller Abstraction Layer

## Communication Drivers

- Drivers for ECU onboard (e.g. SPI) and vehicle communication (e.g. CAN)
- OSI Layer: Part of Data Link Layer

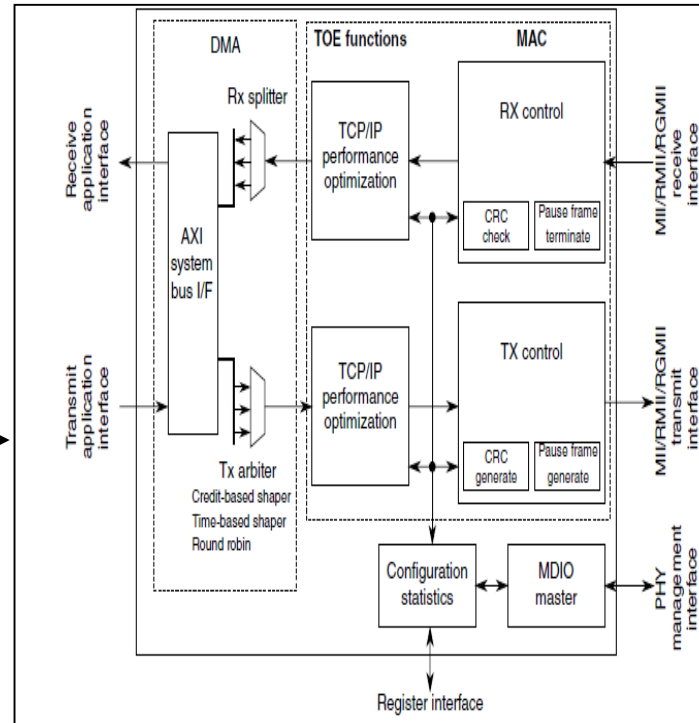


Source: AUTOSAR

# Ethernet Module – NXP Implementation

## Ethernet implementation:

Eth\_Init (...)  
 Eth\_ControllerInit (...)  
 Eth\_SetControllerMode (...)  
 Eth\_GetControllerMode (...)  
 Eth\_GetPhysAddr (...)  
 Eth\_WriteMii (...)  
 Eth\_ReadMii (...)  
 Eth\_GetCounterState (...)  
 Eth\_ProvideTxBuffer (...)  
 Eth\_Transmit (...)  
 Eth\_Receive (...)  
 Eth\_TxConfirmation (...)  
 Eth\_GetVersionInfo (...)  
 Eth\_SetPhysAddr (...)  
 Eth\_UpdatePhysAddrFilter (...)  
 Eth\_GetDropCount (...)  
 Eth\_GetEtherStats (...)  
 Eth\_GetCurrentTime (...)  
 Eth\_EnableEgressTimeStamp (...)  
 Eth\_GetEgressTimeStamp (...)  
 Eth\_GetIngressTimeStamp (...)  
 Eth\_SetCorrectionTime (...)  
 Eth\_SetGlobalTime (...)  
 Eth\_GetDropCount (...)  
 Eth\_GetEtherStats (...)  
 Eth\_GetCounterValues (...)  
 Eth\_GetRxStats (...)  
 Eth\_GetTxStats (...)  
 Eth\_GetTxErrorCounterValues (...)



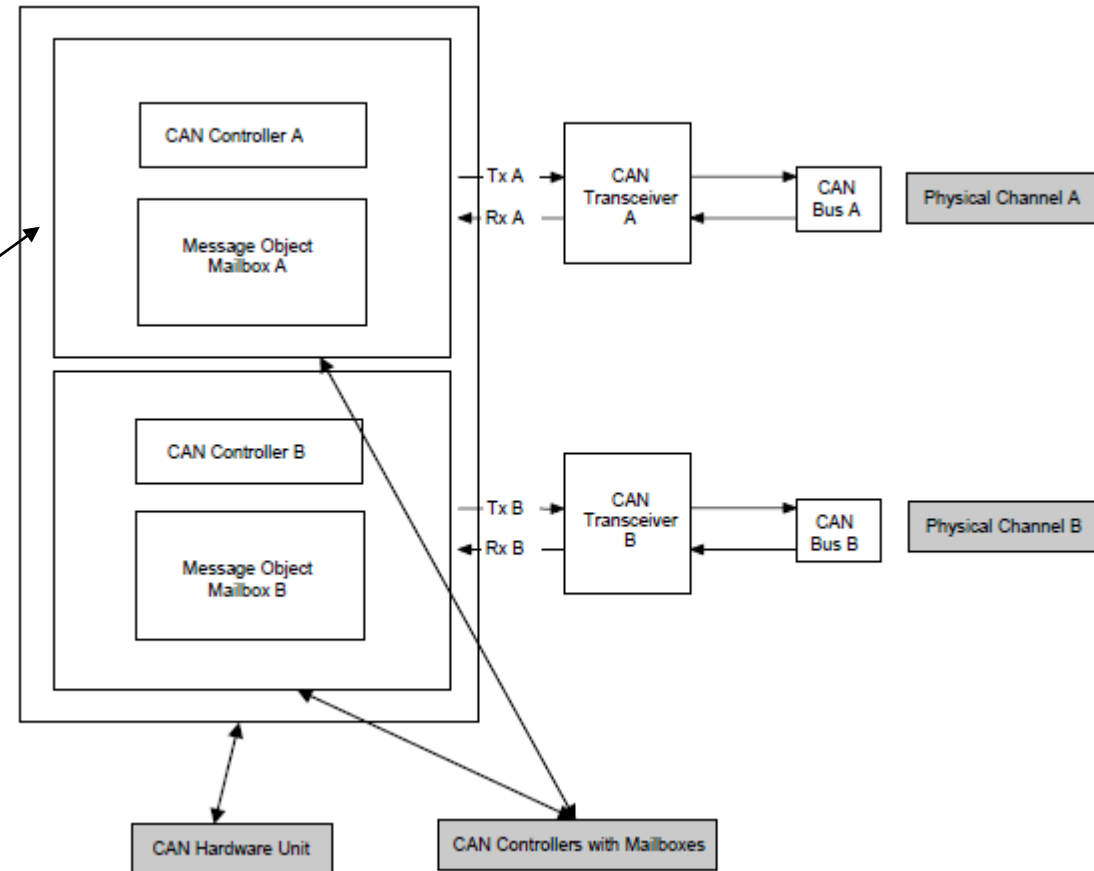
Eth\_GetDropCount (...)  
 Eth\_GetEtherStats (...)  
 Eth\_SetCorrectionTime (...)  
 Eth\_SetGlobalTime (...)  
 Eth\_GetCounterValues (...)  
 Eth\_GetRxStats (...)  
 Eth\_GetTxStats (...)  
 Eth\_GetTxErrorCounterValues (...)

APIs in white are for ASR4.0, ASR4.2, and ASR4.3  
 APIs in yellow are new APIs in ASR4.2 and ASR4.3  
 APIs in red are removed from ASR4.2 and ASR4.3

As comparing with ASR4.2  
 APIs in red are removed from ASR4.2  
 APIs in yellow are new APIs in ASR4.3

# CAN Module – NXP Implementation

CAN implementation:  
Can\_Init (...)  
Can\_ChangeBaudrate (...)  
Can\_SetControllerMode (...)  
Can\_EnableControllerInterrupts (...)  
Can\_CheckWakeup (...)  
Can\_GetVersionInfo (...)  
Can\_CheckBaudrate (...)  
Can\_DisableControllerInterrupts (...)  
Can\_Write (...)  
Can\_MainFunction\_Write (...)  
Can\_MainFunction\_Read (...)  
Can\_MainFunction\_BusOff (...)  
Can\_MainFunction\_Wakeup (...)  
Can\_MainFunction\_Mode (...)  
Can\_SetBaudRate (...)  
Can\_GetControllerErrorState (...)  
Can\_GetControllerMode (...)  
Can\_DeInit (...)

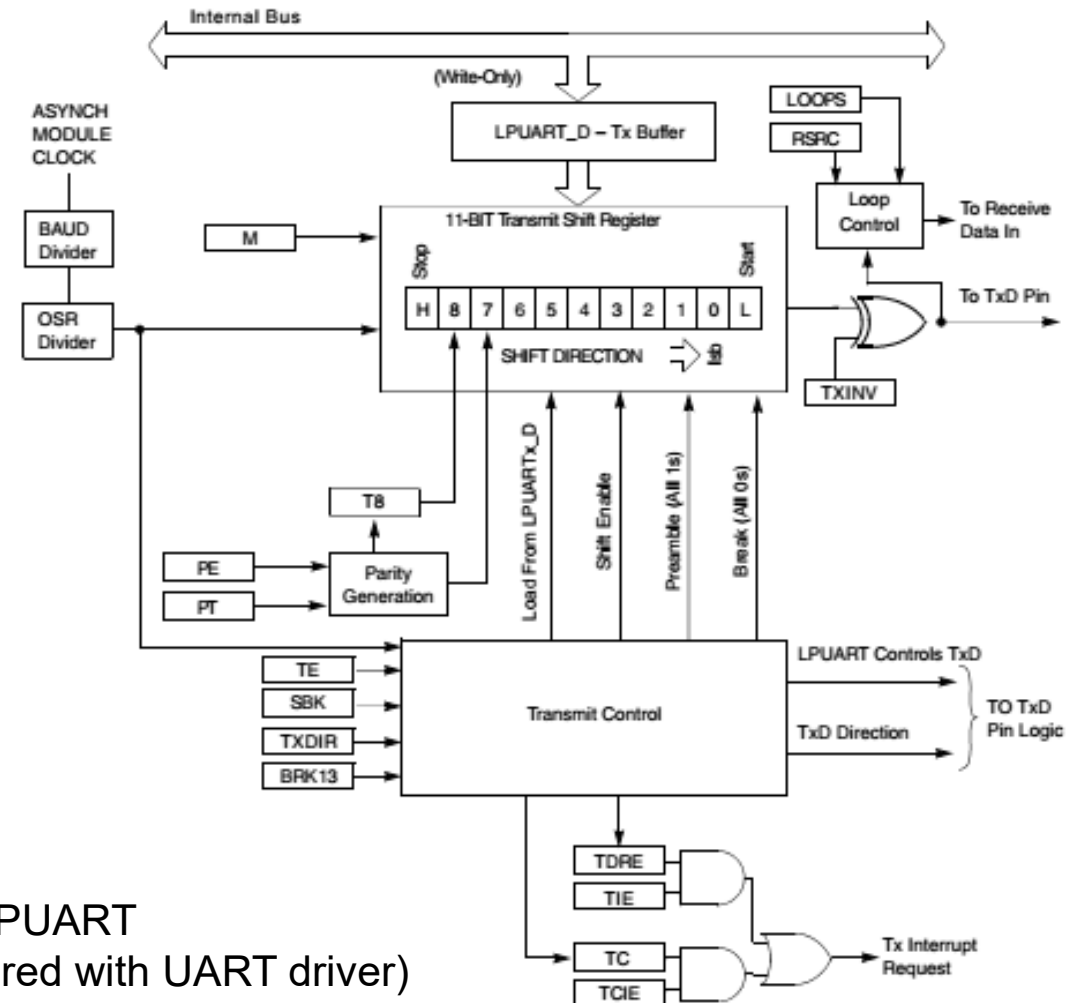


APIs in white are for ASR4.0, ASR4.2, and ASR4.3  
APIs in yellow are new APIs in ASR4.3  
APIs are in red are removed from ASR4.3

# LIN Module – NXP Implementation

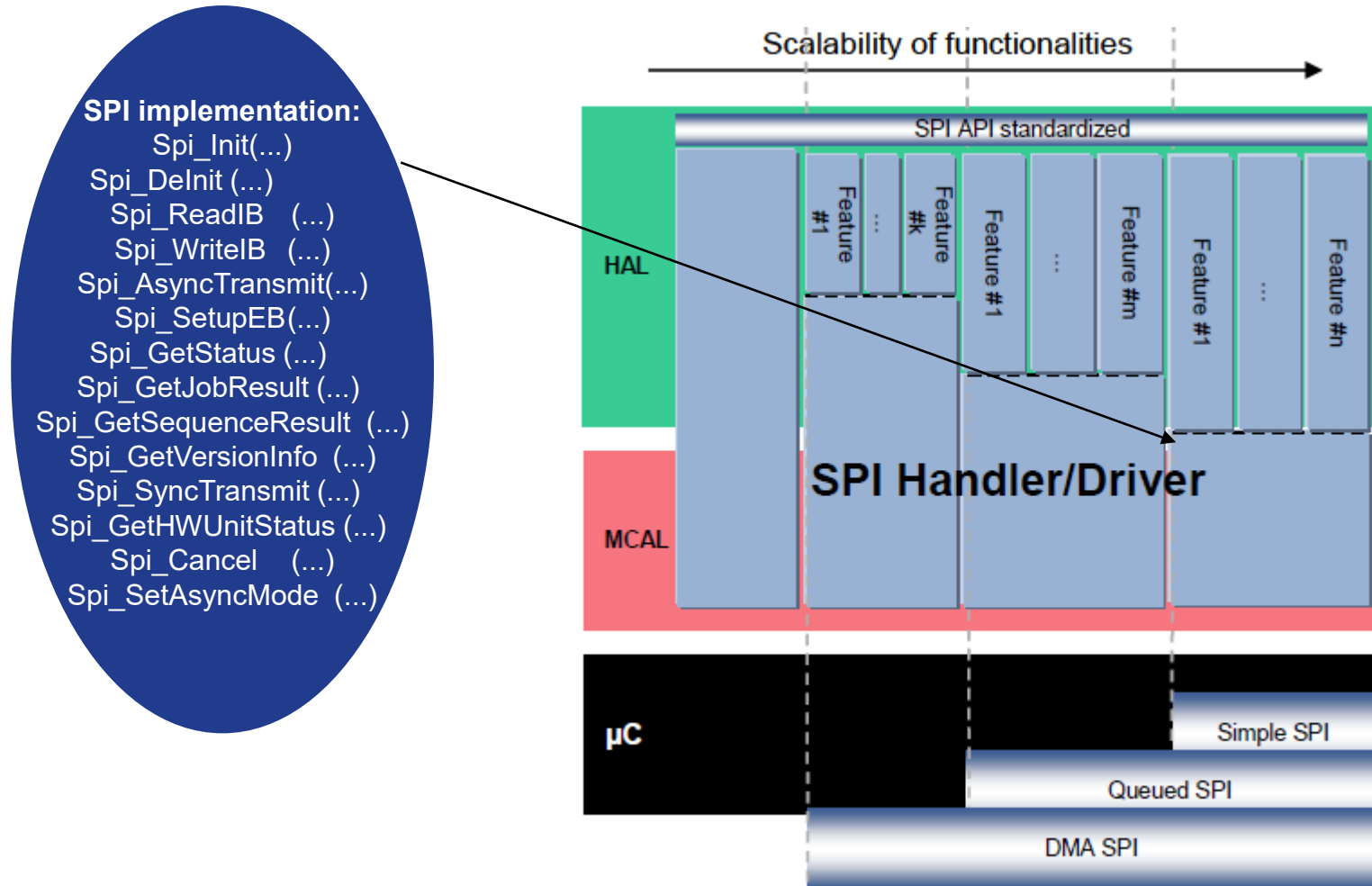
LIN implementation:  
 Lin\_Init (...)  
 Lin\_CheckWakeup (...)  
 Lin\_GetVersionInfo (...)  
 Lin\_SendFrame (...)  
 Lin\_GoToSleep (...)  
 Lin\_GoToSleepInternal (...)  
 Lin\_Wakeup (...)  
 Lin\_WakeupInternal (...)  
 Lin\_GetStatus (...)

LPUART  
 (LPUART is shared with UART driver)





# SPI Module – NXP Implementation



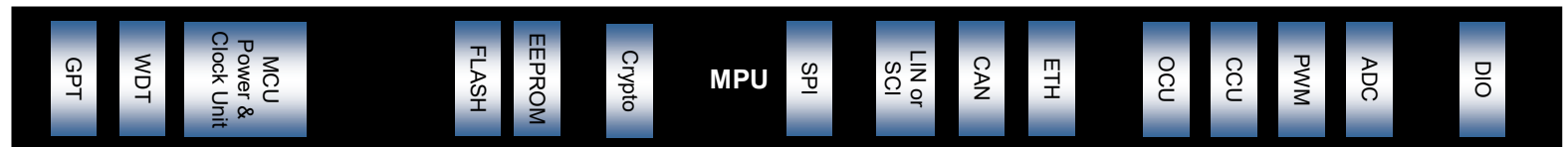
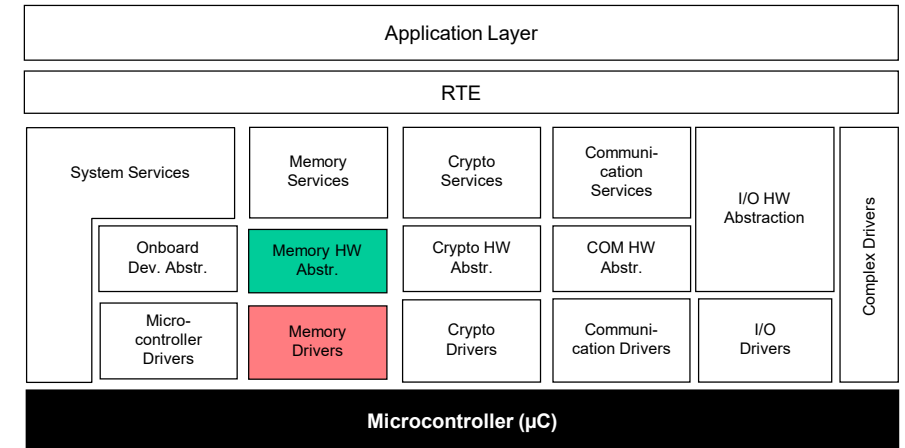
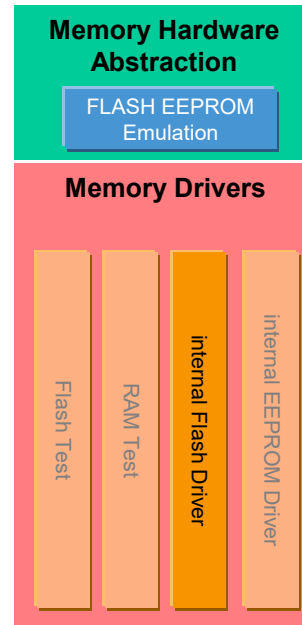
# i.MX 8 MCAL Memory Drivers



# AUTOSAR — Microcontroller Abstraction Layer

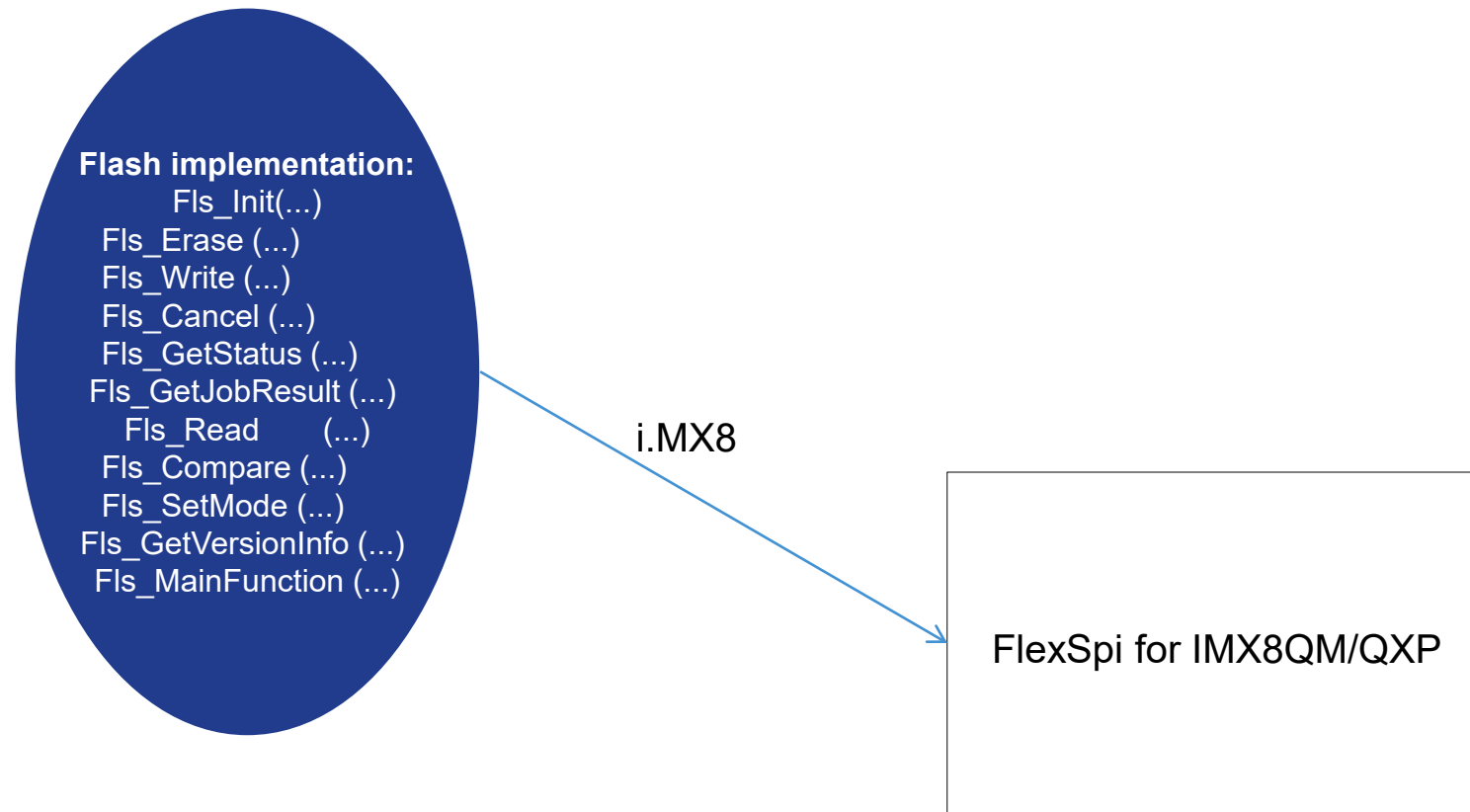
- **Memory Drivers**

- The Memory Hardware Abstraction is a group of modules which abstracts from the location of peripheral memory devices (on-chip or on-board) and the ECU hardware layout
- The Memory Drivers are accessed via memory specific abstraction/emulation modules (e.g. EEPROM Abstraction)



Source: **AUTOSAR**

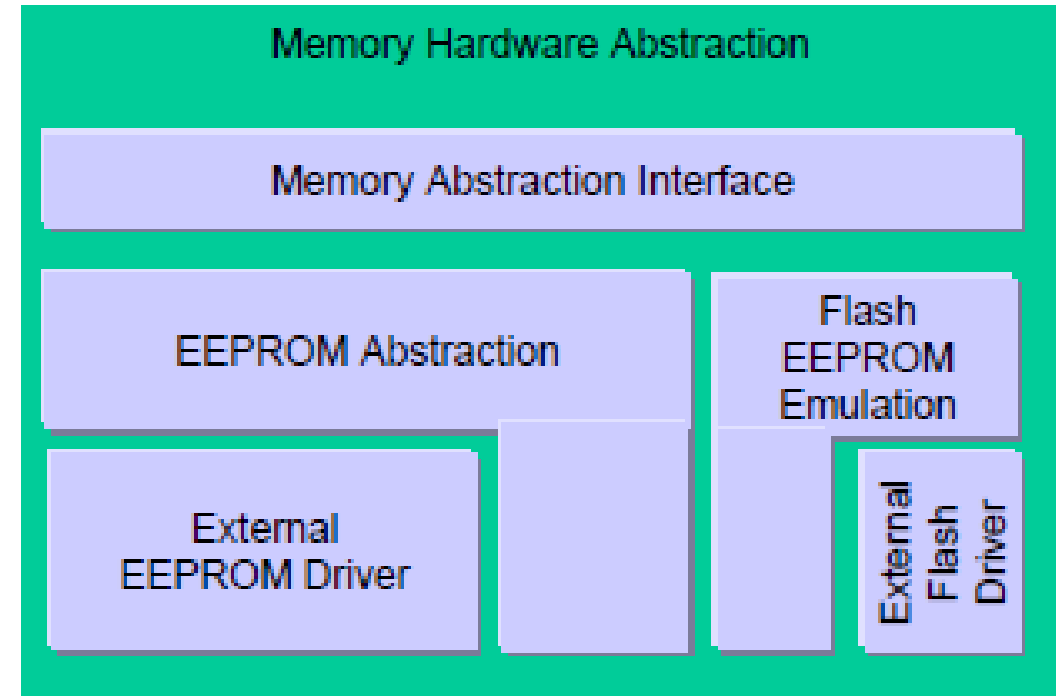
# Flash Module – NXP Implementation Per External Memory



# FEE Module – NXP Implementation Per Flash Memory

FEE is located in the “Memory Hardware Abstraction” layer which falls under the ECU Abstraction Layer

**FEE implementation:**  
Fee\_Init (...)  
Fee\_SetMode (...)  
Fee\_Read (...)  
Fee\_Write (...)  
Fee\_Cancel (...)  
Fee\_GetStatus (...)  
Fee\_GetJobResult (...)  
Fee\_InvalidateBlock (...)  
Fee\_GetVersionInfo (...)  
Fee\_EraseImmediateBlock (...)



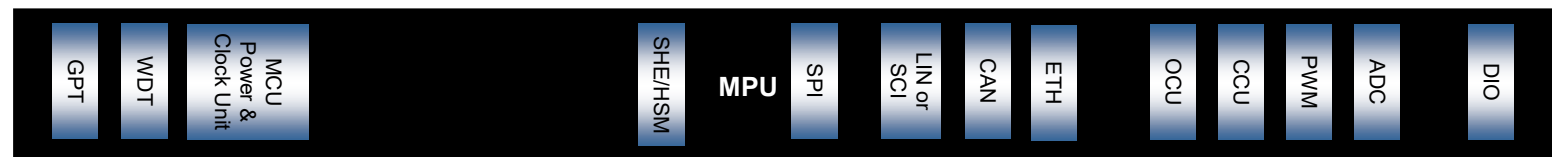
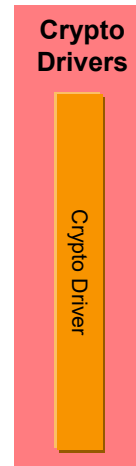
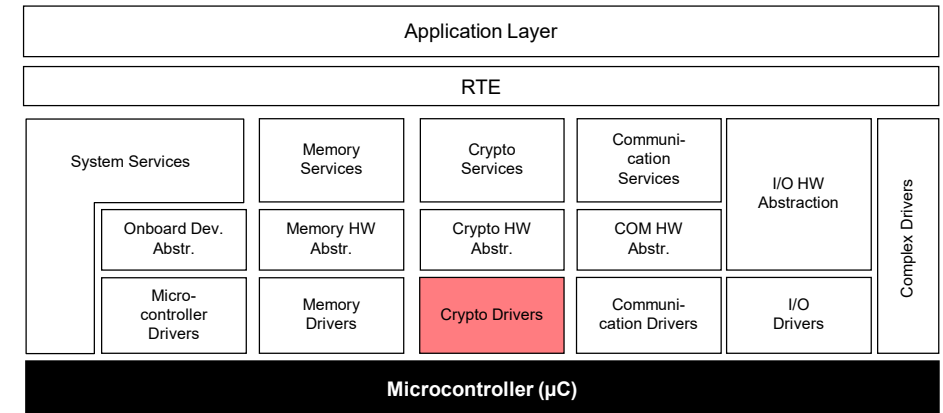
# i.MX 8 MCAL Crypto Drivers



# AUTOSAR — Microcontroller Abstraction Layer

## CRYDRV

- Crypto driver communicates SHE/HSM through SECO firmware by using message unit



Source: AUTOSAR

# i.MX 8 MCAL Complex Drivers





# AUTOSAR — Complex Device Drivers

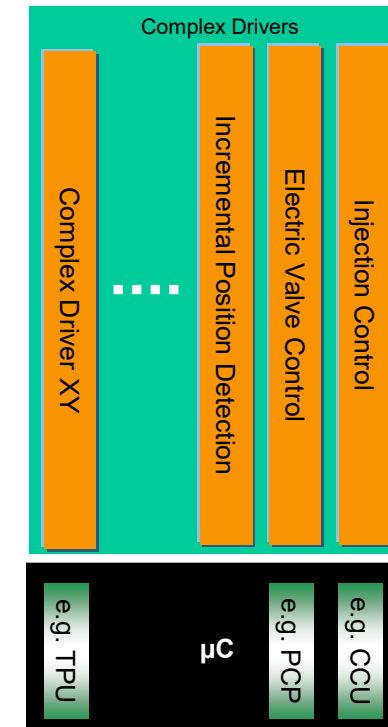
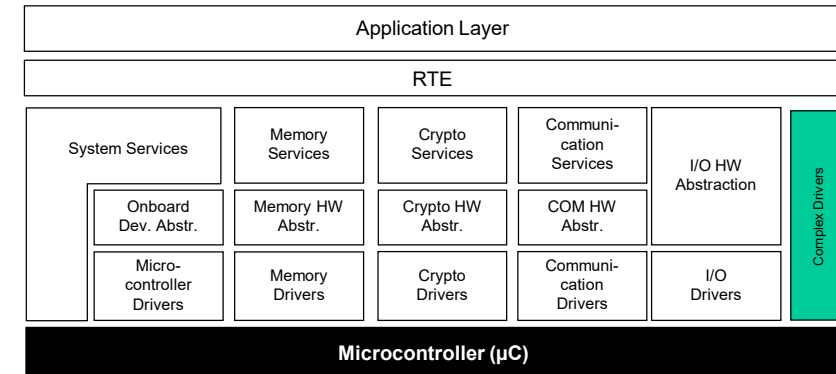
A **Complex Driver** is a module which implements non-standardized functionality within the basic software stack.

## Examples:

- **Core and RAM/Flash Self Tests Library**
- **CRC Driver**
- **EMMC**
- **ESAI (I2S)**
- **Fault Monitoring Drivers**
- **I2C**
- **MicroController Library (MCL) – eDMA management/ Cache management**
- **RPMSG**
- **SDIO**
- **UART**
- **USB**

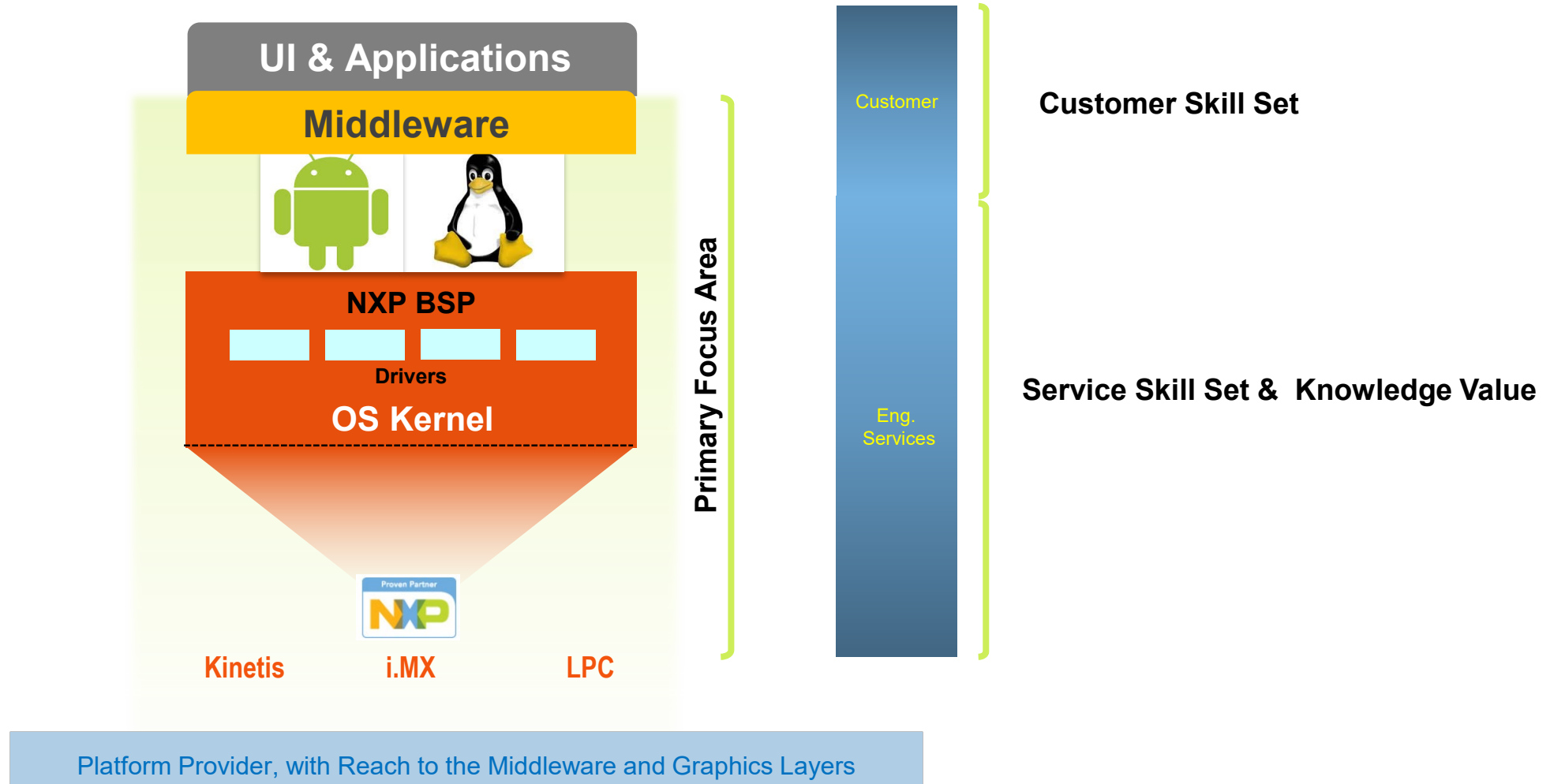
## Properties:

- **Implementation:** highly  $\mu\text{C}$ , ECU and application dependent
- **Upper Interface to SW-Cs:** specified and implemented according to AUTOSAR (AUTOSAR interface)
- **Lower interface:** restricted access to Standardized Interfaces



Source: **AUTOSAR**

# Professional Engineering Services Technical Competency





**SECURE CONNECTIONS  
FOR A SMARTER WORLD**