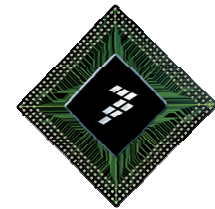


Nov 28–29, 2007

汽车电子软件的趋势 - AUTOSAR 如何改变您 使用飞思卡尔控制器的方式



Andreas Both, 张恩勤

FreescalTM and the Freescale logo are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners. © Freescale Semiconductor, Inc. 2007.



汽车电子软件标准 AUTOSAR 规范了崭新的汽车应用领域软件的体系结构.

沿着 2006年FTF讲座的关于AUTOSAR基本知识介绍, 本次将着重介绍分层架构, 以及 MCU-相关 和 MCU-无关的软件的划分.

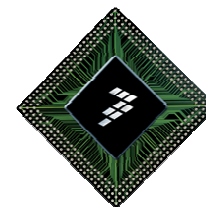
我们将展示在遵守AUTOSAR设计方法下, 使用飞思卡尔产品将有怎样的变化.



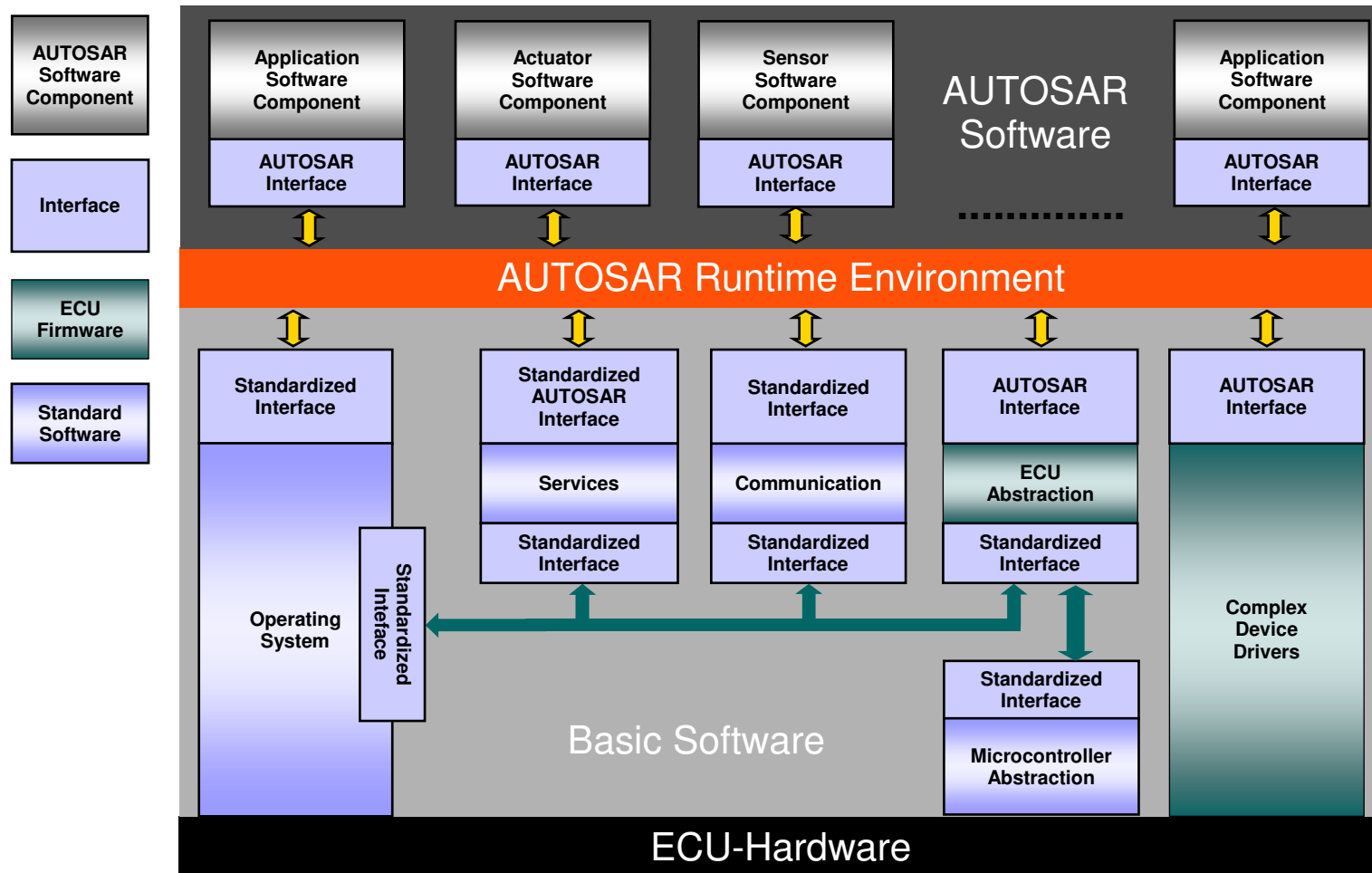
- ▶ AUTOSAR 定义的基础软件
 - 分层结构
 - MCU-相关的基础软件模块
 - MCU-无关的基础软件模块
- ▶ AUTOSAR 操作系统探讨
- ▶ AUTOSAR 配置方法探讨
- ▶ 针对飞思卡尔控制器的AUTOSAR 软件



AUTOSAR 基础软件

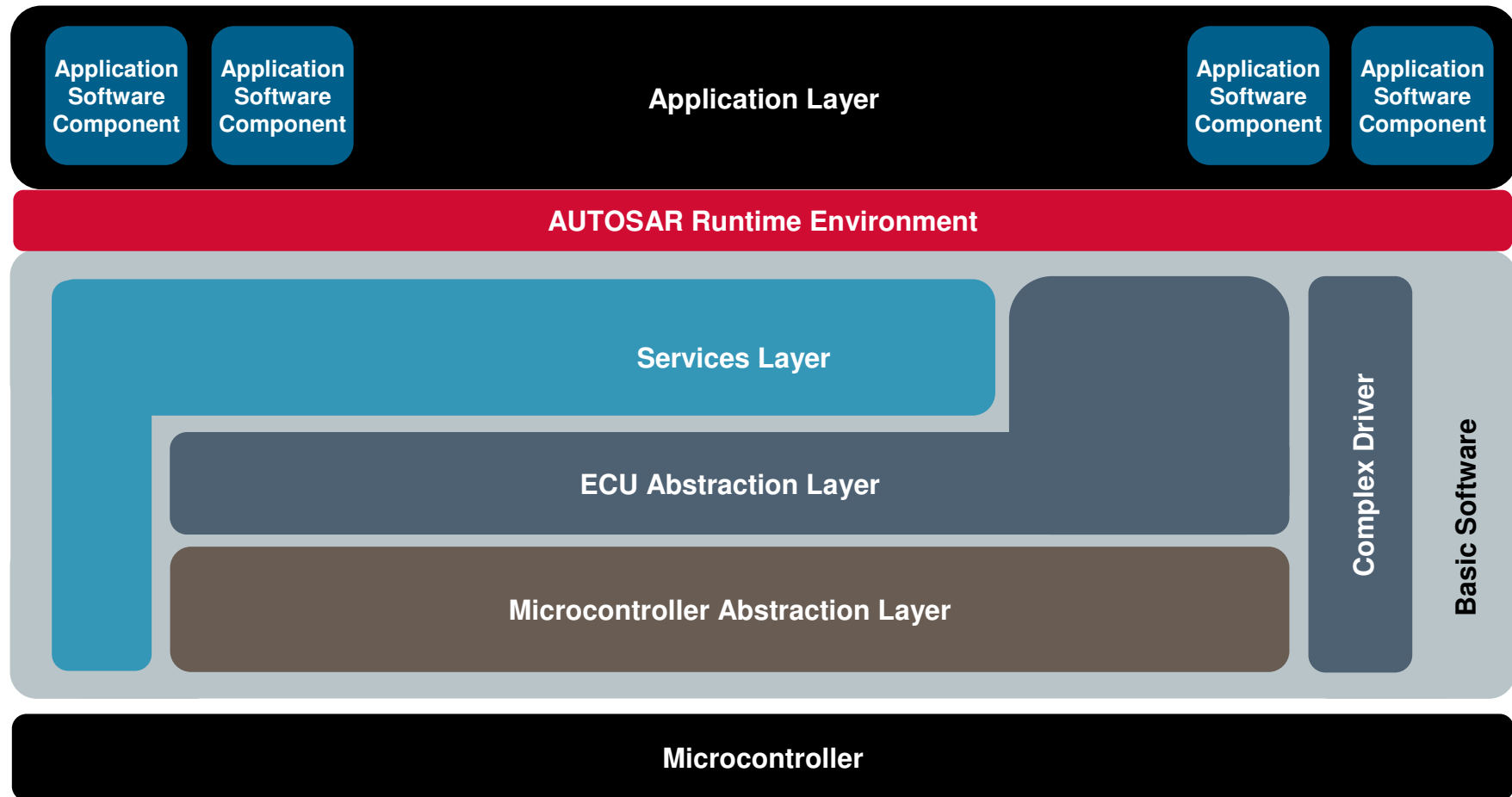


AUTOSAR 体系结构

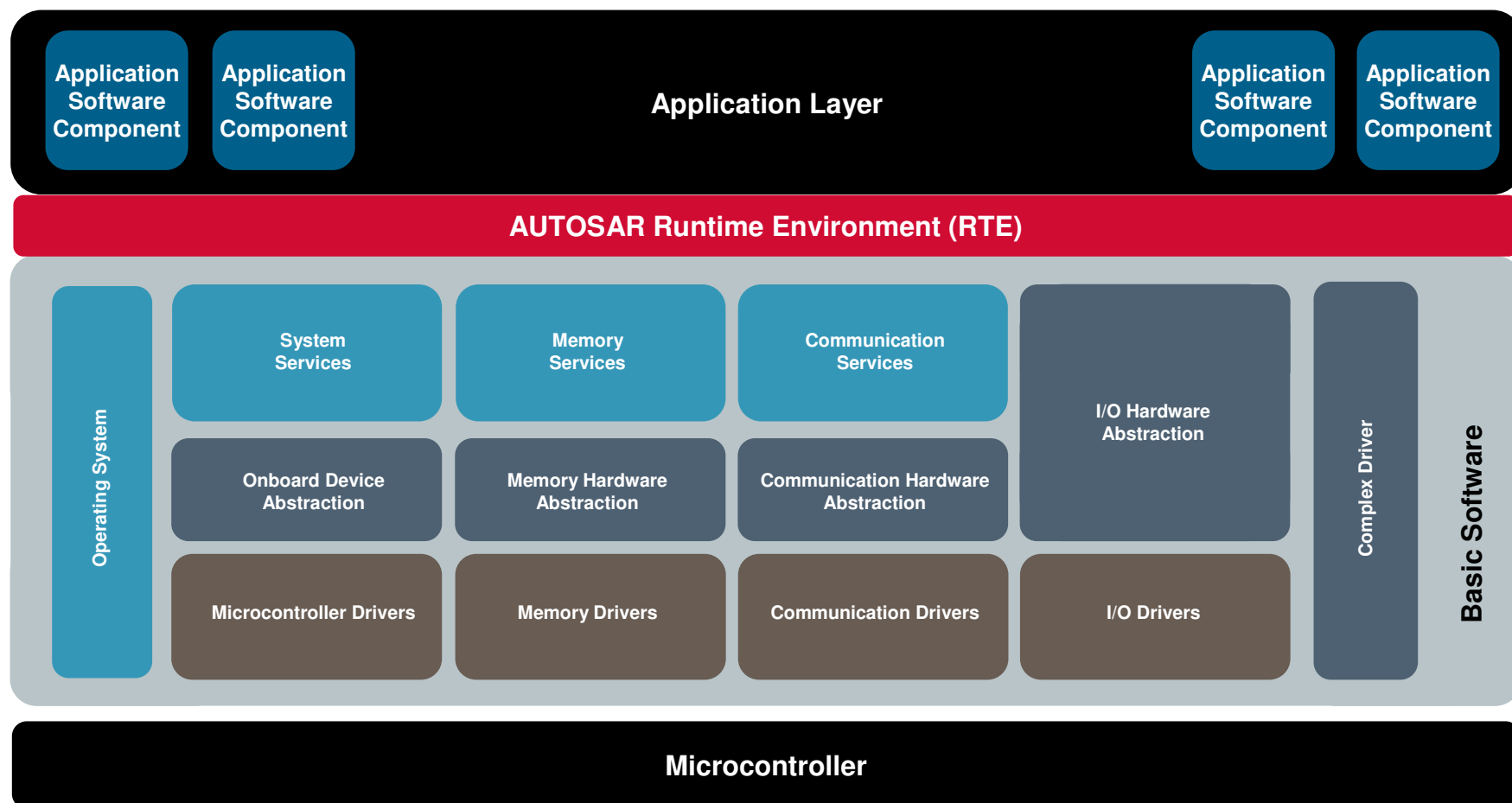


Source: AUTOSAR GbR

AUTOSAR BSW 结构—基础层



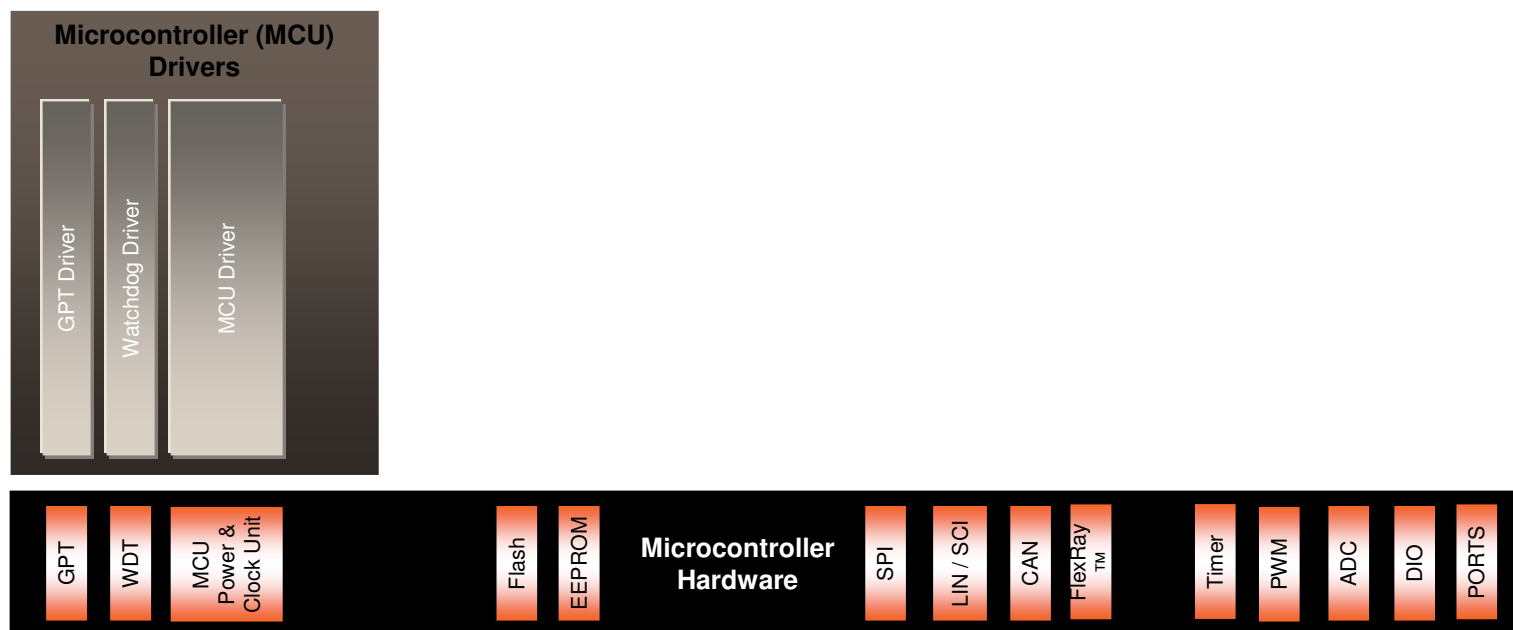
AUTOSAR BSW 体系结构—细化分层



微控制器抽象层

微控制器驱动

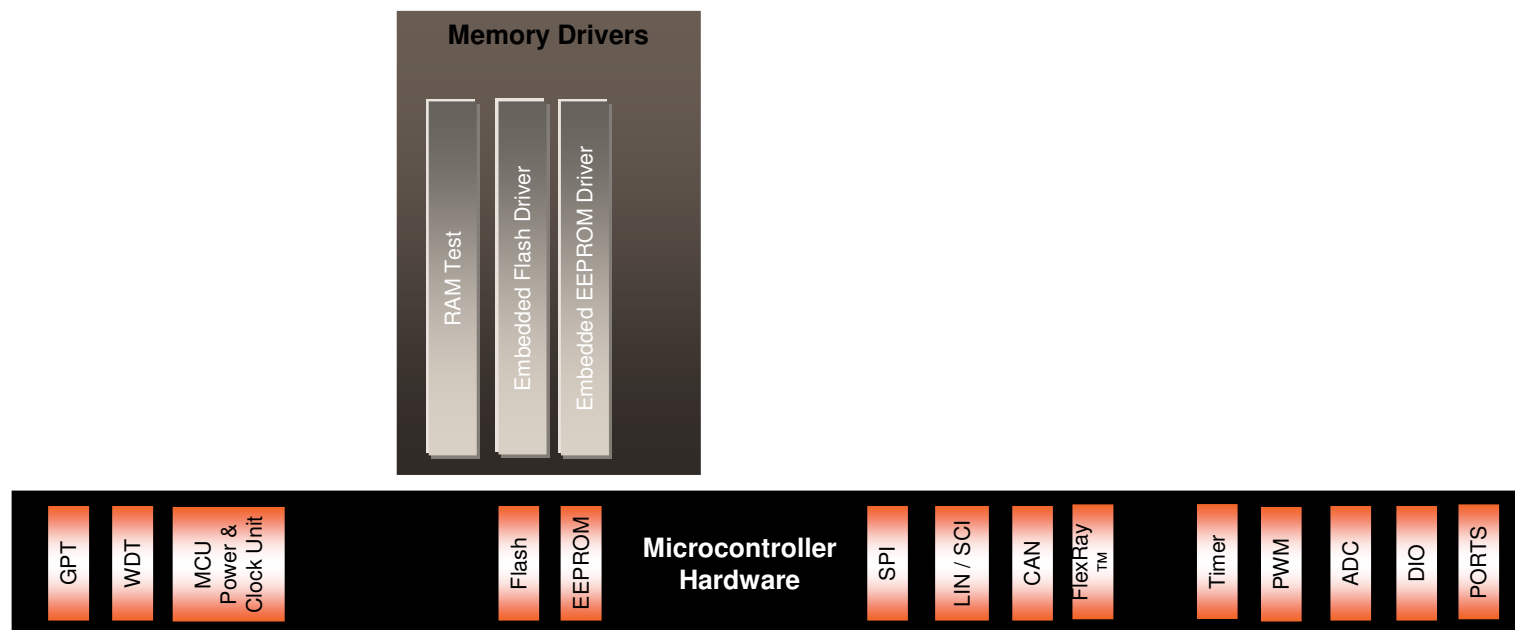
- 内部外设的驱动 (e.g., watchdog, 通用时钟)
- 直接 μ C 可存取的功能 (e.g., Core 测试)



微控制器抽象层

内存控制器

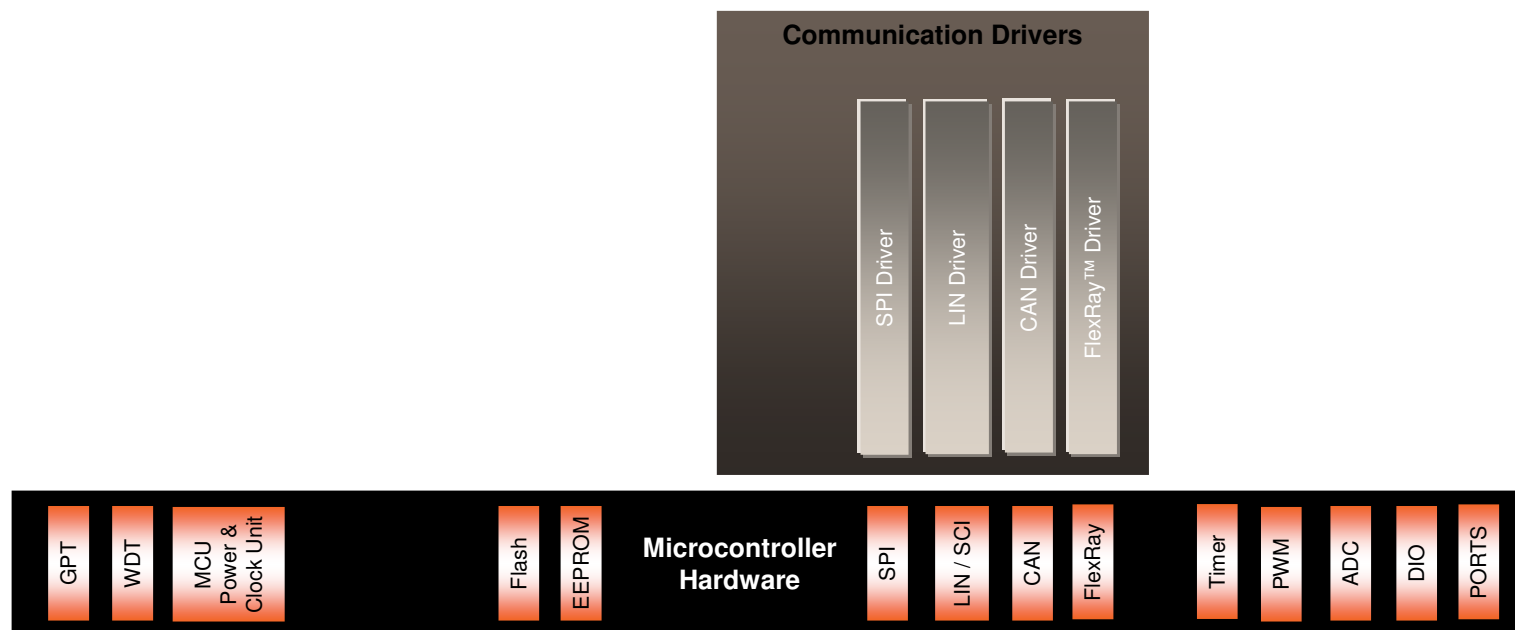
- 片上内存设备的驱动 (e.g., 内部 flash, 内部 EEPROM)



微控制器抽象层

通信驱动

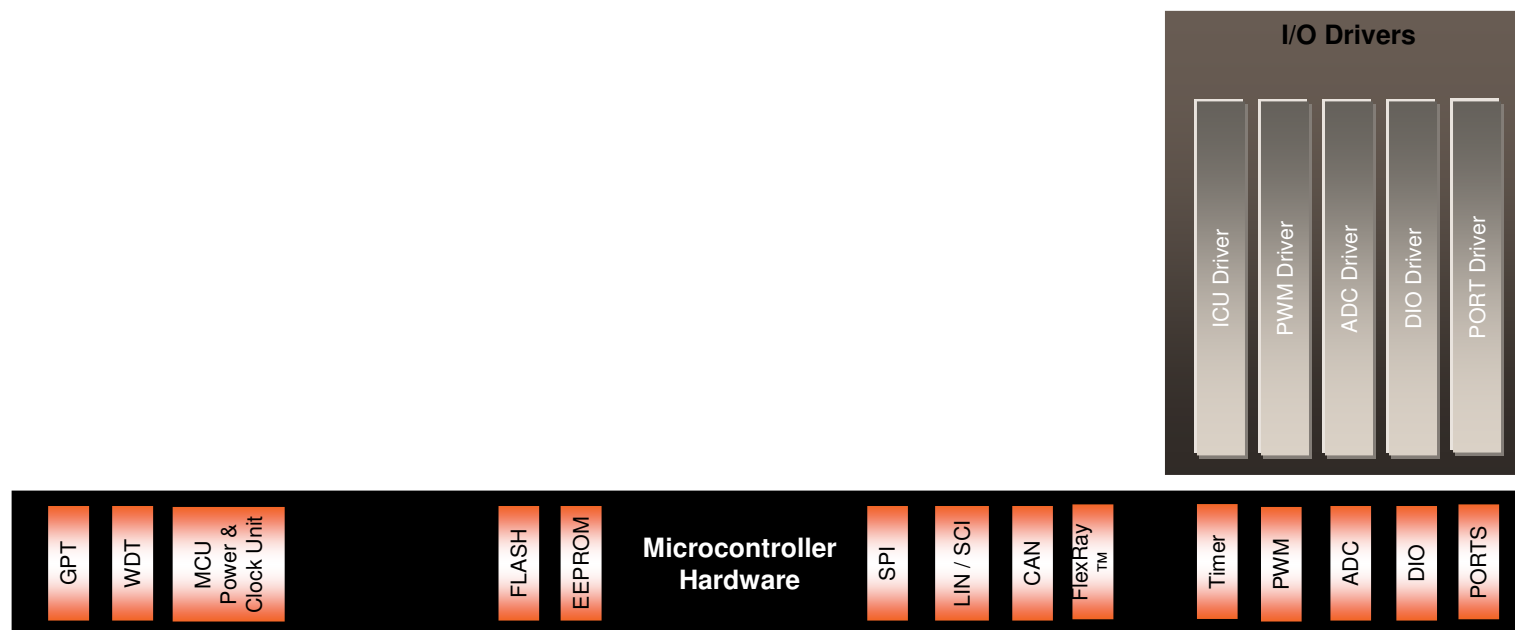
- ECU 板级上的通信驱动: SPI, I2C
- 车载通信驱动: LIN, CAN, FlexRay™ 网络



微控制器抽象层

I/O 驱动

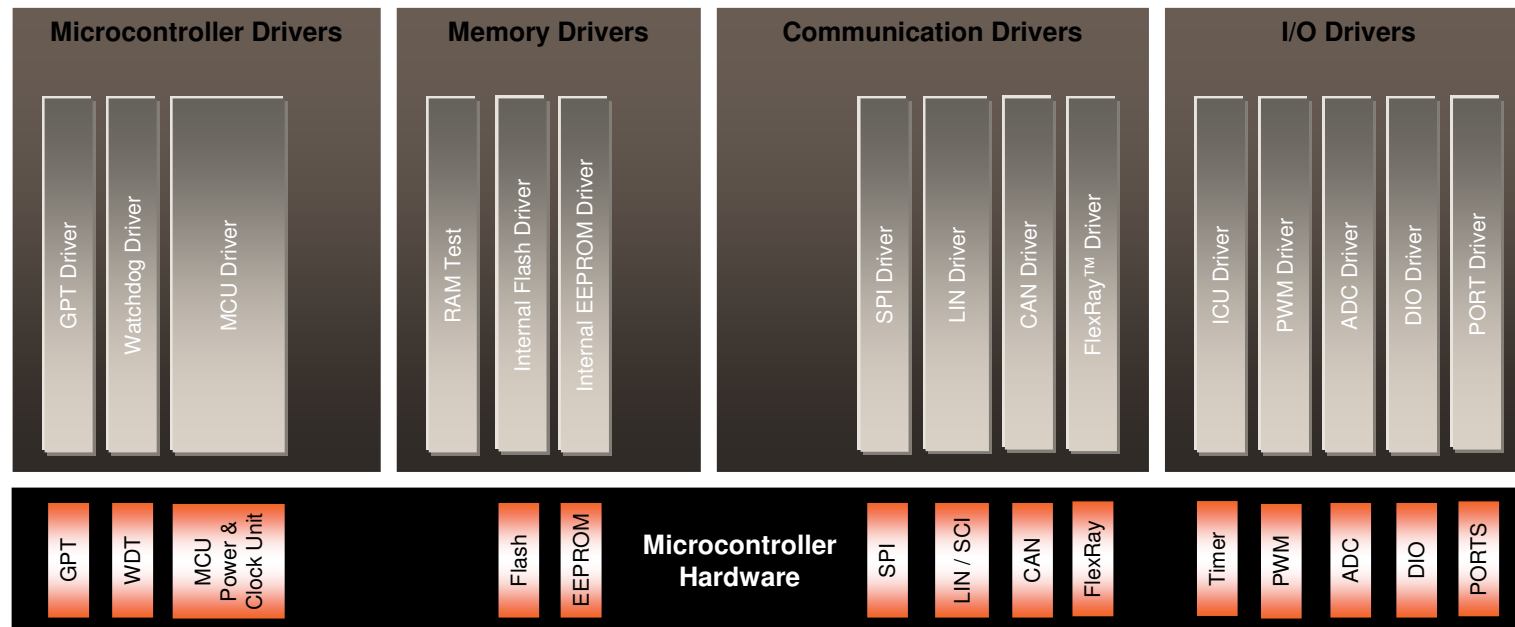
- 模拟和数字 I/O 驱动: e.g., 模数转换器(ADC), 脉宽调制(PWM), 数字输入输出 (DIO)



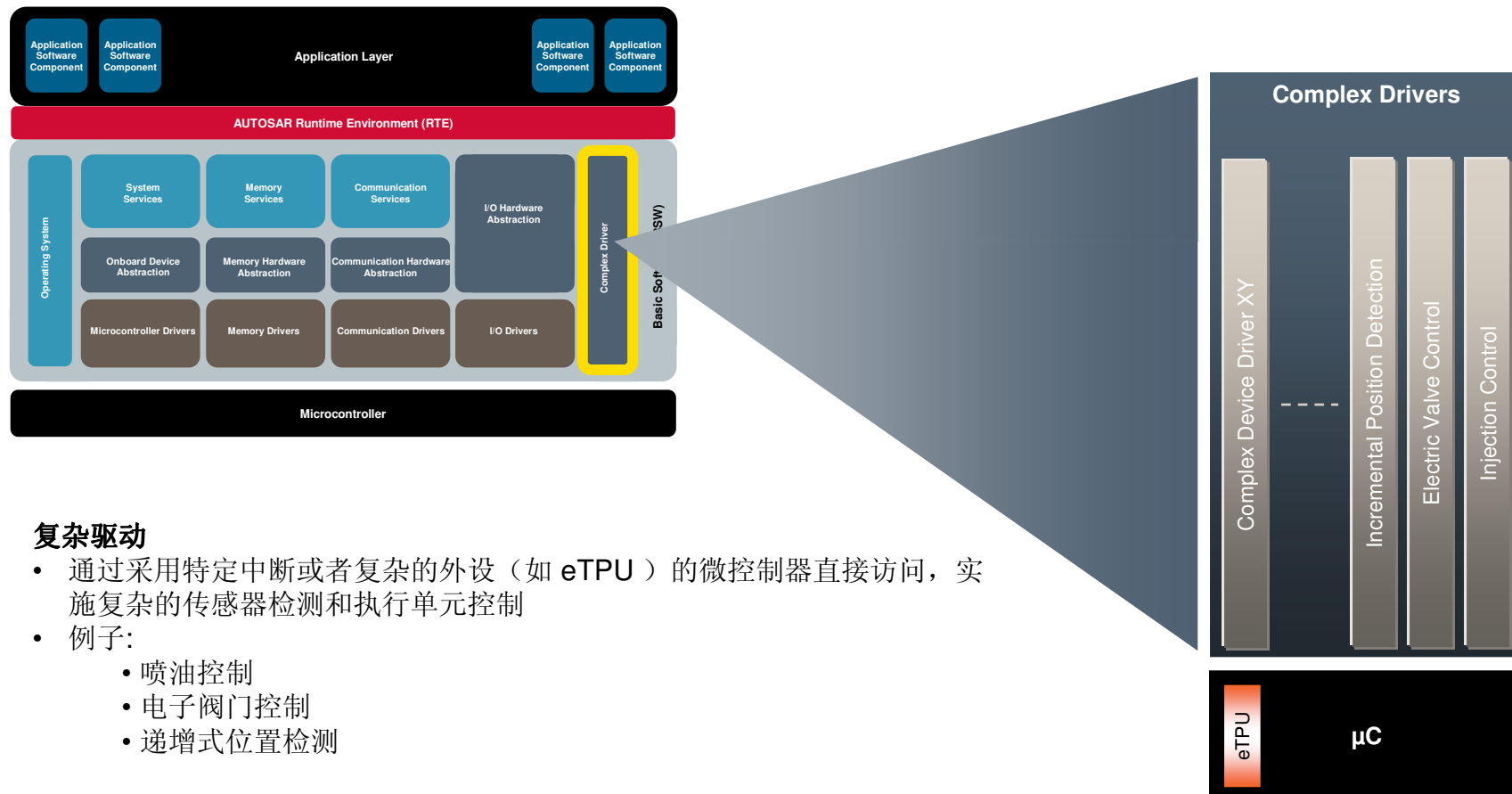
微控制器抽象层

微控制器抽象层的驱动

- 直接与微控制器硬件接口 (核和外设)
- 包括以下软件模块:
 - 微控制器驱动
 - 内存驱动
 - 通信驱动
 - I/O 驱动



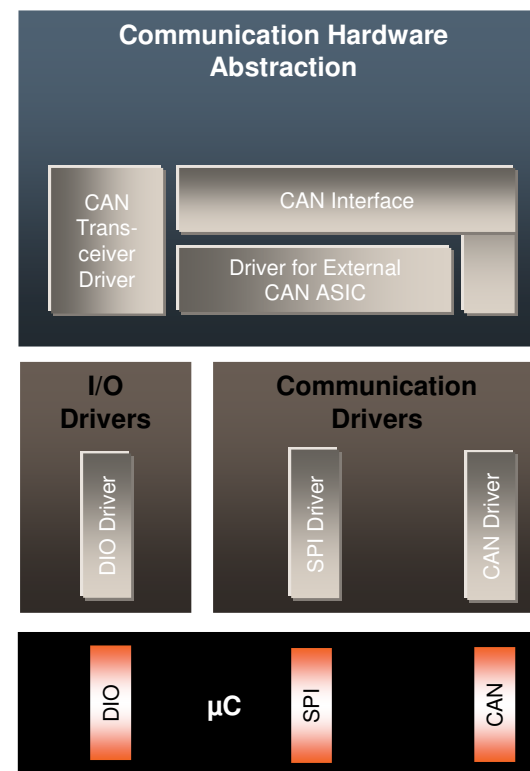
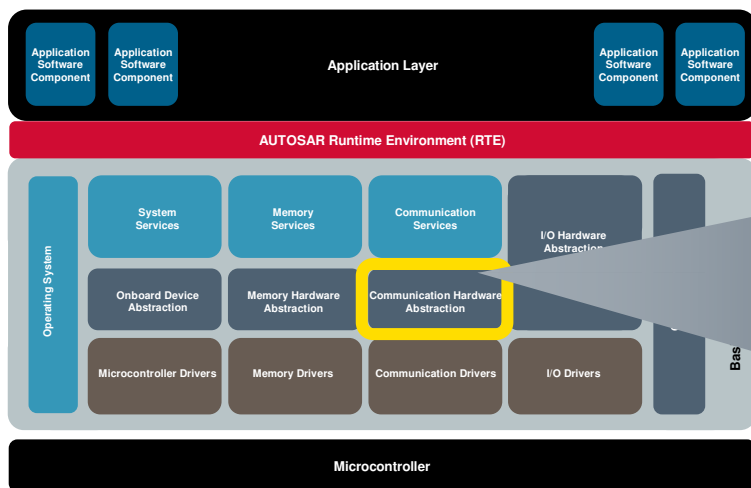
复杂驱动



复杂驱动

- 通过采用特定中断或者复杂的外设（如 eTPU）的微控制器直接访问，实施复杂的传感器检测和执行单元控制
- 例子:
 - 喷油控制
 - 电子阀门控制
 - 递增式位置检测

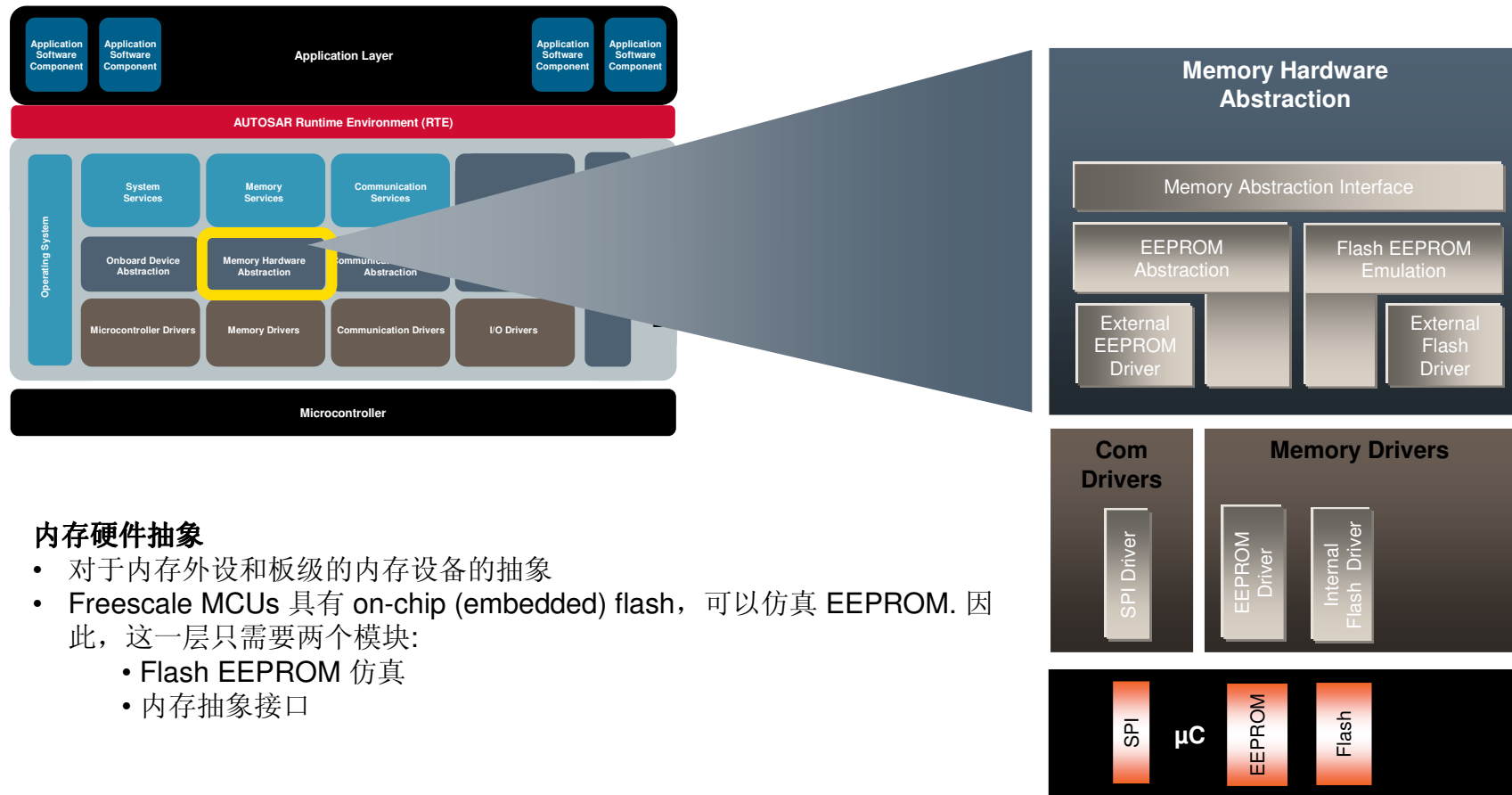
通信硬件抽象层



通信硬件抽象层

- 对于内部和**ECU** 硬件架构上外部通信控制器的抽象
- 对于所有的通信系统 (LIN, CAN, FlexRay), 都需要特定的通信硬件抽象:
 - LIN 接口
 - CAN 接口
 - FlexRay 接口
- Freescale MCUs 具有 on-chip CAN 控制器, 因此外部的 CAN ASIC 一般不需要

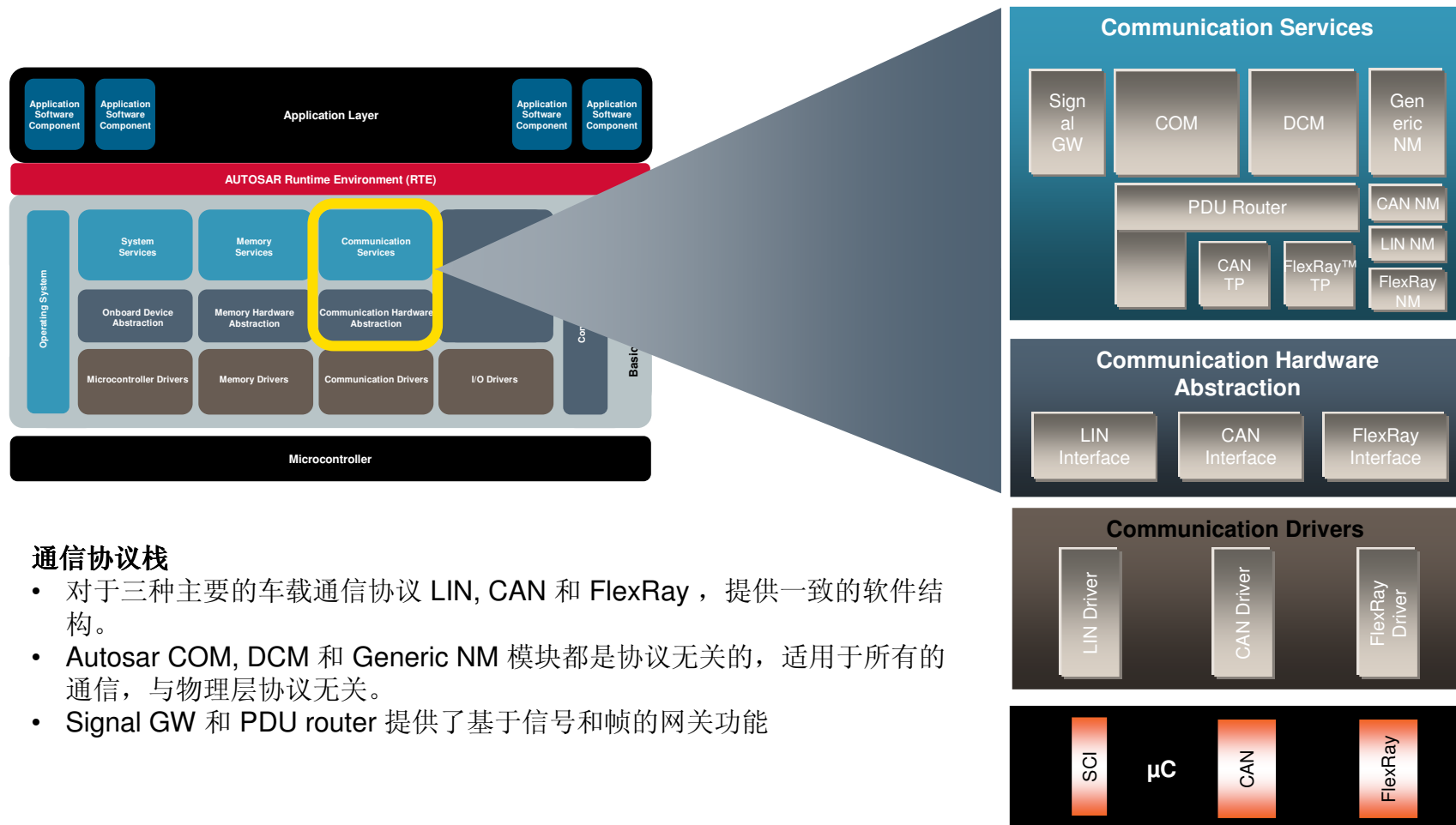
内存硬件抽象层



内存硬件抽象

- 对于内存外设和板级的内存设备的抽象
- Freescale MCUs 具有 on-chip (embedded) flash, 可以仿真 EEPROM. 因此, 这一层只需要两个模块:
 - Flash EEPROM 仿真
 - 内存抽象接口

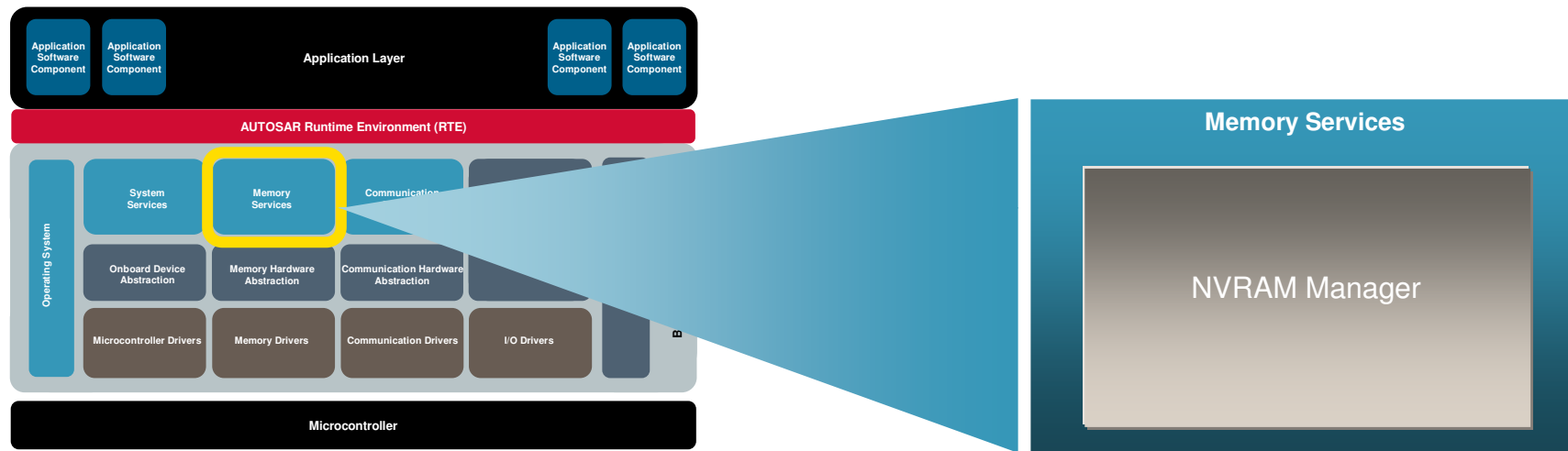
通信软件包



通信协议栈

- 对于三种主要的车载通信协议 LIN, CAN 和 FlexRay，提供一致的软件结构。
- Autosar COM, DCM 和 Generic NM 模块都是协议无关的，适用于所有的通信，与物理层协议无关。
- Signal GW 和 PDU router 提供了基于信号和帧的网关功能

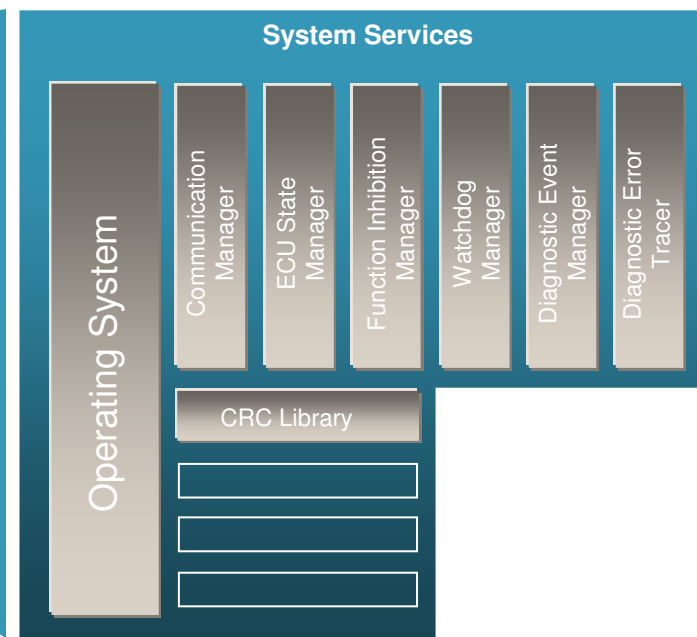
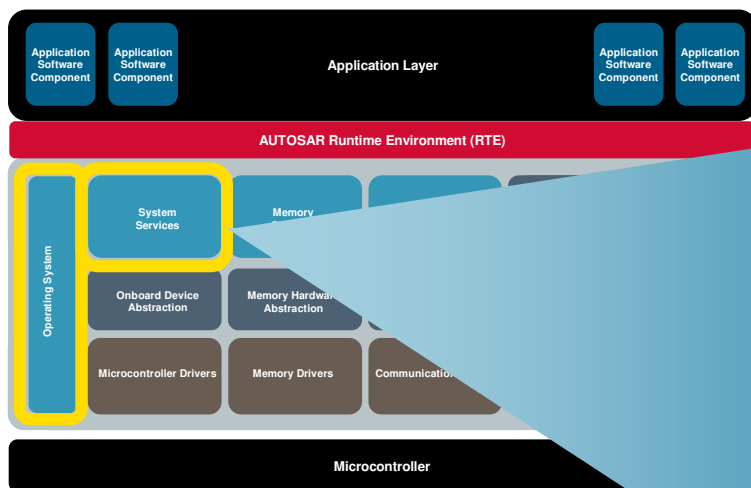
内存服务



内存服务

- 提供非易失数据的管理功能，例如数据存取，冗余校验保护，和可靠存储
- 采用不同的内存驱动
- 采取 RAM 映射作为应用程序快速读取数据的接口
- NVRAM Manager 模块是 MCU 无关的

系统服务



系统服务

- 提供一些功能以供别的模块调用，如诊断事件管理 (DEM), 诊断错误跟踪 (DET), 操作系统 (OS)
- OS 是与 MCU 相关的系统服务，需要针对于每个MCU优化
- 其余模块可以是应用或硬件相关的(e.g., ECU state manager)

AUTOSAR Release 2.1

▶ 基础软件模块:	46 modules
▶ 基础软件文档:	3609 pages
▶ 其他文档:	1957 pages
▶ 辅助文档	

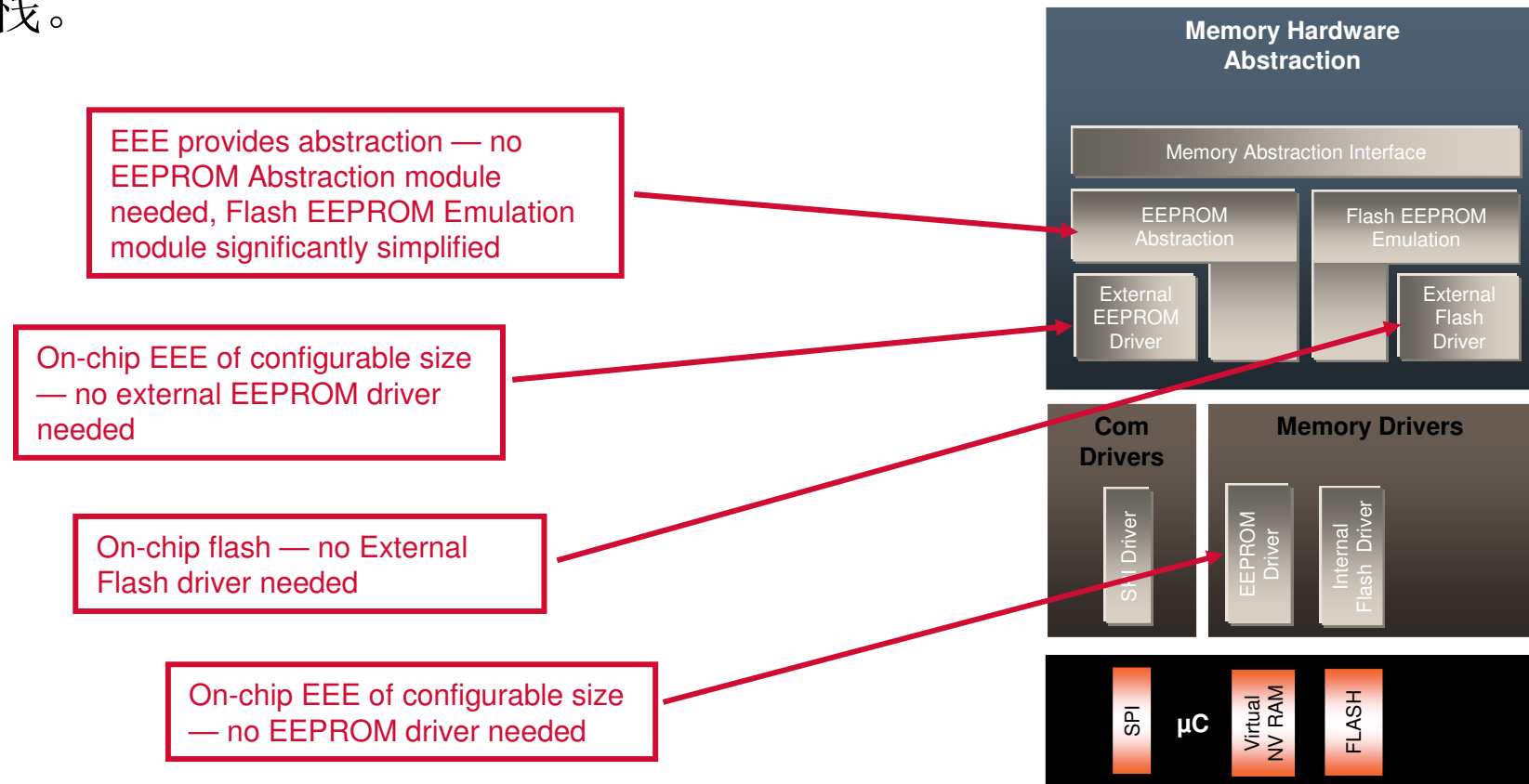
以上虽然不是一个很好的度量方法，但足以看出 AUTOSAR 基础软件的复杂性。

Q: 全部的AUTOSAR 基础软件模块是否是每个 MCU 必需的？

A: 不是的。请看下面的例子。

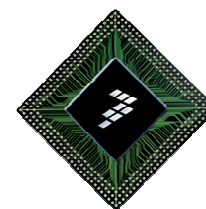
例子: S12XEP100 AUTOSAR 内存软件栈

- 智能的 MCU 外设将大大简化 BSW 内存软件栈。





AUTOSAR 操作系统



▶ AUTOSAR OS 是 OSEK/VDX™ OS 加上:

- 新的核心功能
 - 软件和硬件计数器
 - 基于时间同步的调度表
 - 堆栈监视
- 保护功能
 - 定时保护, 内存保护和服务保护
 - OS 应用程序, 信任的和不信任的代码
 - Protection hook



AUTOSAR OS Scalability Classes 1–4

	Scalability Class 1	Scalability Class 2	Scalability Class 3	Scalability Class 4
OSEK OS (all conformance classes)	✓	✓	✓	✓
Counter Interface	✓	✓	✓	✓
Schedule Tables	✓	✓	✓	✓
Stack Monitoring	✓	✓	✓	✓
Protection Hook		✓	✓	✓
Timing Protection		✓		✓
Global Time/Synchronization Support		✓		✓
Memory Protection			✓	✓
OS Applications			✓	✓
Service Protection			✓	✓
CallTrustedFunction			✓	✓

OS 应用以及可信任和不信任代码

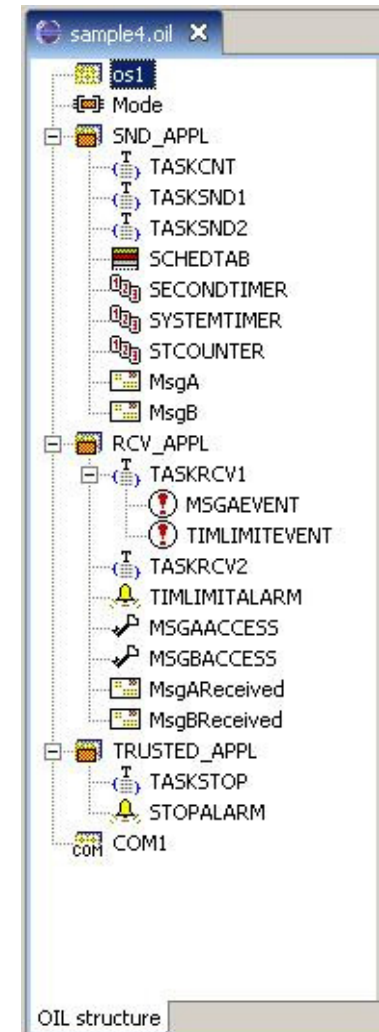
► 完整性: 可信任和不信任代码

► OS 应用

- 一组软件包括 tasks, ISRs, hooks 和 trusted 函数
- **Trusted:** 具有无限制的存取权限
- **Non-trusted:** 受限制的存取权限

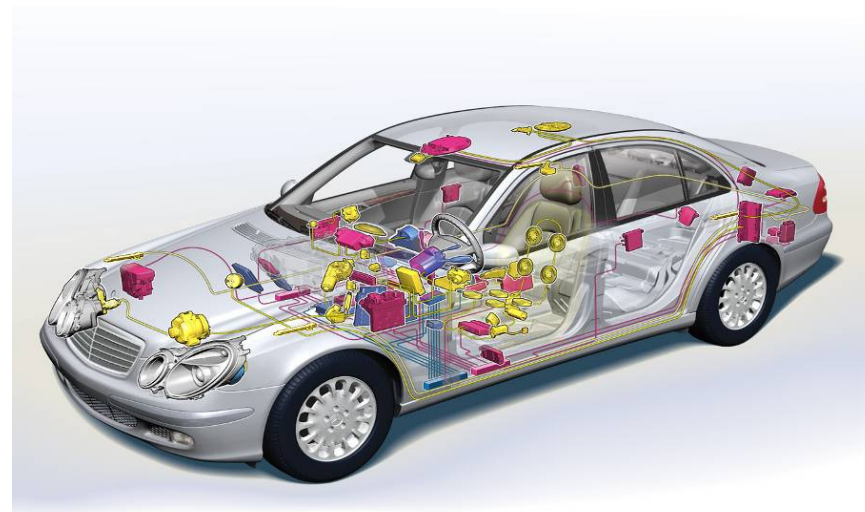
► 可信任函数

- 具有无限制存取权限的一种服务功能
- 由一个 trusted OS 应用提供



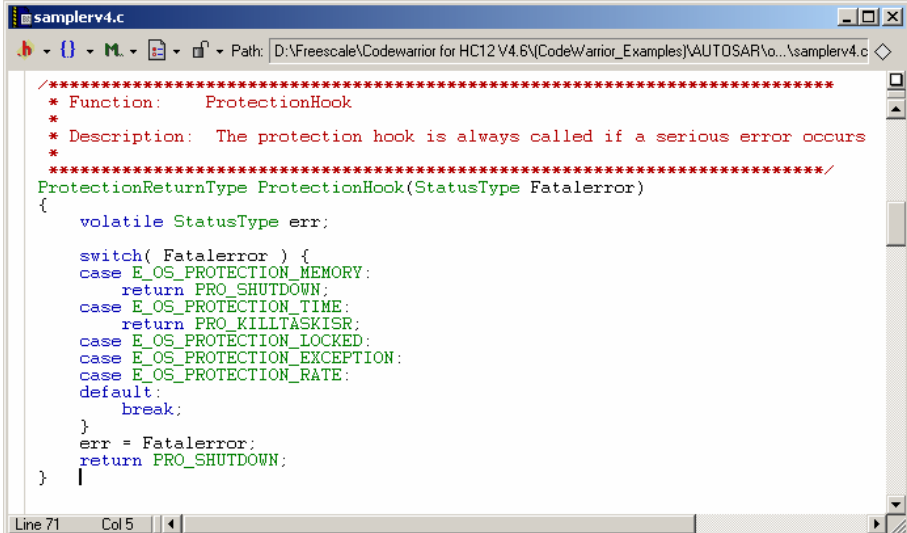
AUTOSAR OS 内存保护

- ▶ 保护的边界
 - Task/category 2 ISR
 - OS application (all its Task/Category 2 ISRs)
- ▶ 保护的内存
 - Stack RAM
 - Private data RAM
 - Code flash
- ▶ 受保护的存取方式
 - 主要是写保护，读和执行保护是可选的



► The Protection Hook

- 在所有的保护出错被调用 (memory / service / timing)
- 根据返回值确定采取的操作:
 - Kill the faulty Task/Category 2 ISR, or
 - Kill the faulty OS application, or
 - Kill the faulty OS application and restart the OS application, or
 - Call ShutdownOS()



```

sampler4.c
D:\Freescall\CodeWarrior for HC12 V4.6\CodeWarrior_Examples\AUTOSAR\o...sampler4.c
/*****
 * Function:    ProtectionHook
 * Description: The protection hook is always called if a serious error occurs
 *****/
ProtectionReturnType ProtectionHook(StatusType Fatalerror)
{
    volatile StatusType err;

    switch( Fatalerror ) {
        case E_OS_PROTECTION_MEMORY:
            return PRO_SHUTDOWN;
        case E_OS_PROTECTION_TIME:
            return PRO_KILLTASKISR;
        case E_OS_PROTECTION_LOCKED:
        case E_OS_PROTECTION_EXCEPTION:
        case E_OS_PROTECTION_RATE:
        default:
            break;
    }
    err = Fatalerror;
    return PRO_SHUTDOWN;
}
Line 71 Col 5

```

- ▶ 不信任 OS 应用任务
 - 只能存取它所需要的内存
 - 因此，不是指这一应用任务受保护，而是其他任务与它隔离

- ▶ 内存保护可被用于以下
 - 分离运行于同一 MCU 的不同应用程序
 - 分离来自于不同供应商的控制功能模块
 - 实现安全性限制
 - 作为一种调试功能 (防止错误的内存访问，防止堆栈溢出，protection hook 将被调用)

- ▶ 内存保护必须有片上 MPU 的支持

▶ 服务保护

- 保护一个 OS 应用调用错误或被破坏的 OS 服务
- 例子
 - OS Application 调用 ShutDownOS()
 - OS Application 试图通过执行 ActivateTask() 激活别的 OS Application 的任务
- 在出现服务保护错误时，Protection Hook 将被调用

定时保护的使用以及全局时间

▶ 定时保护

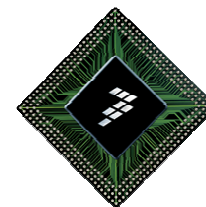
- 执行时间增强
 - 限制中断，资源锁定和中断屏蔽时间在一个静态可配置的时间 („time budget“)
- 调用频率增强
 - 限制中断在指定时间内的调用次数在一个静态可配置的界限
- 在出现时间保护错误时，**Protection Hook** 将被调用

▶ 全局时间 / 同步支持

- 需要一个全局的时间源，例如 **FlexRay** 网络时间
- 这一功能通过 **OS** 服务调用使得调度表与全局时间同步

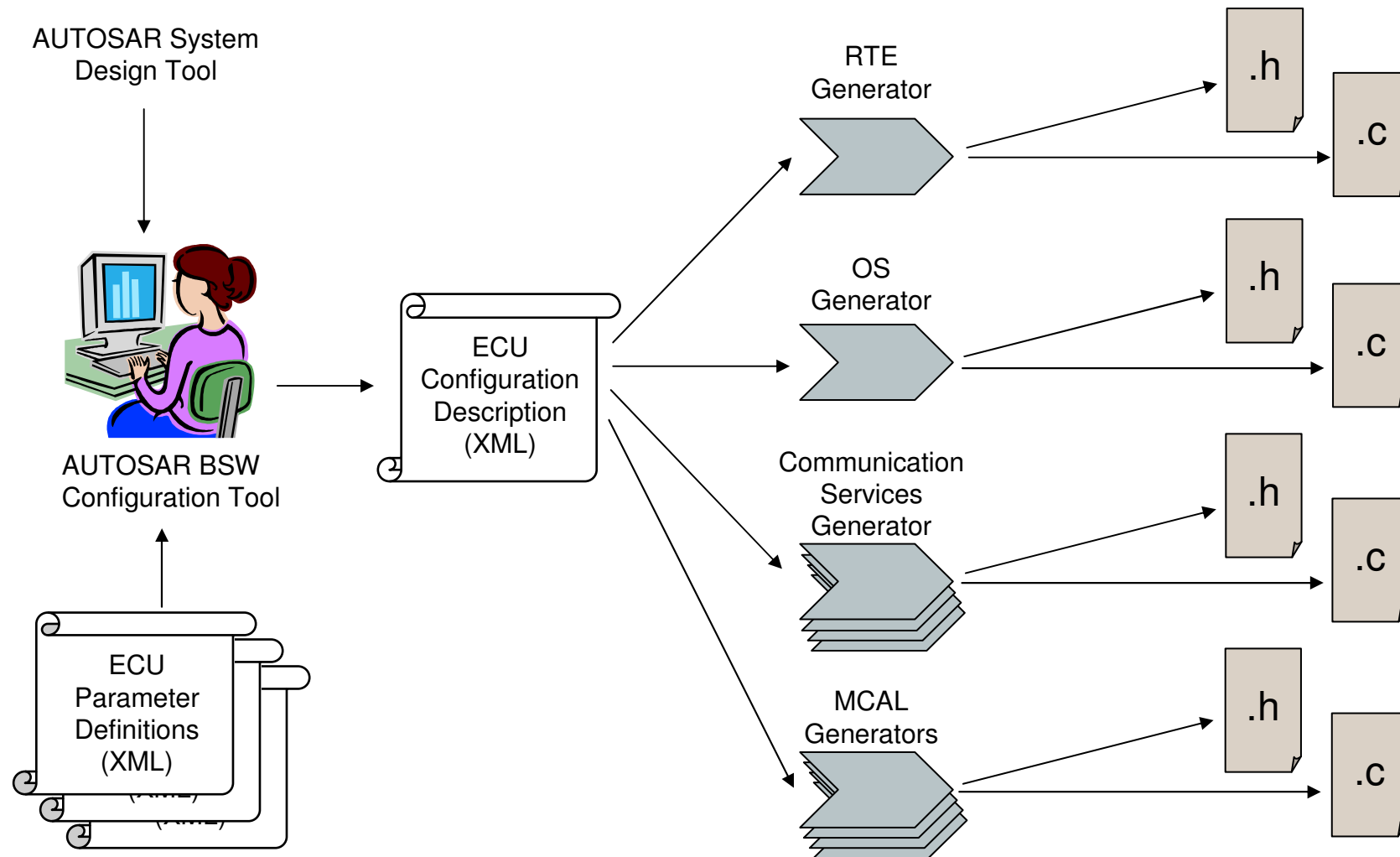


AUTOSAR 配置方法



- ▶ 预编译阶段配置
 - 在编译后，配置参数将不能被改变
 - 需要提供源程序
- ▶ 连接阶段配置
 - 配置决定于 **linker scripts**
 - 在 **link** 后，配置参数将不能被改变
 - 适于提供 **object code**
- ▶ 生成后配置
 - 生成后可载入的
 - 配置参数在生成结束后可更改，不需要重新烧写 ECU
 - 生成后可选择的
 - 在启动阶段配置参数可从多组选项中选择
 - 所有可能的配置选项都包含在编译阶段
 - 配置参数存储在已知的内存空间地址
 - 可提供 **source** 和 **object code**
 - 生成后配置类的 **BSW** 模块可提供预编译或连接时的参数设置（并非所有的参数都可生成后配置）

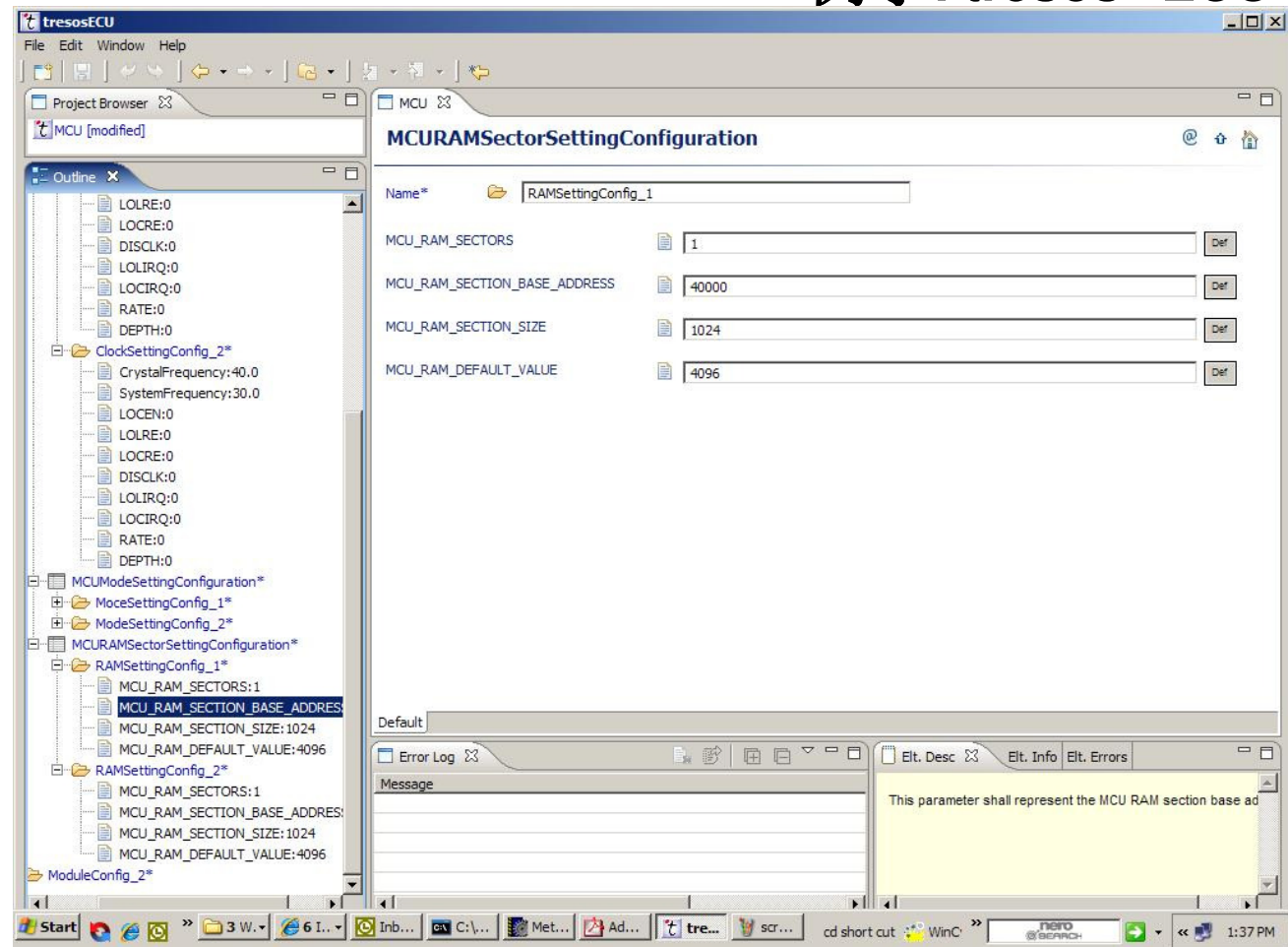
基础软件配置流程



AUTOSAR BSW 配置工具

例子: tresos® ECU

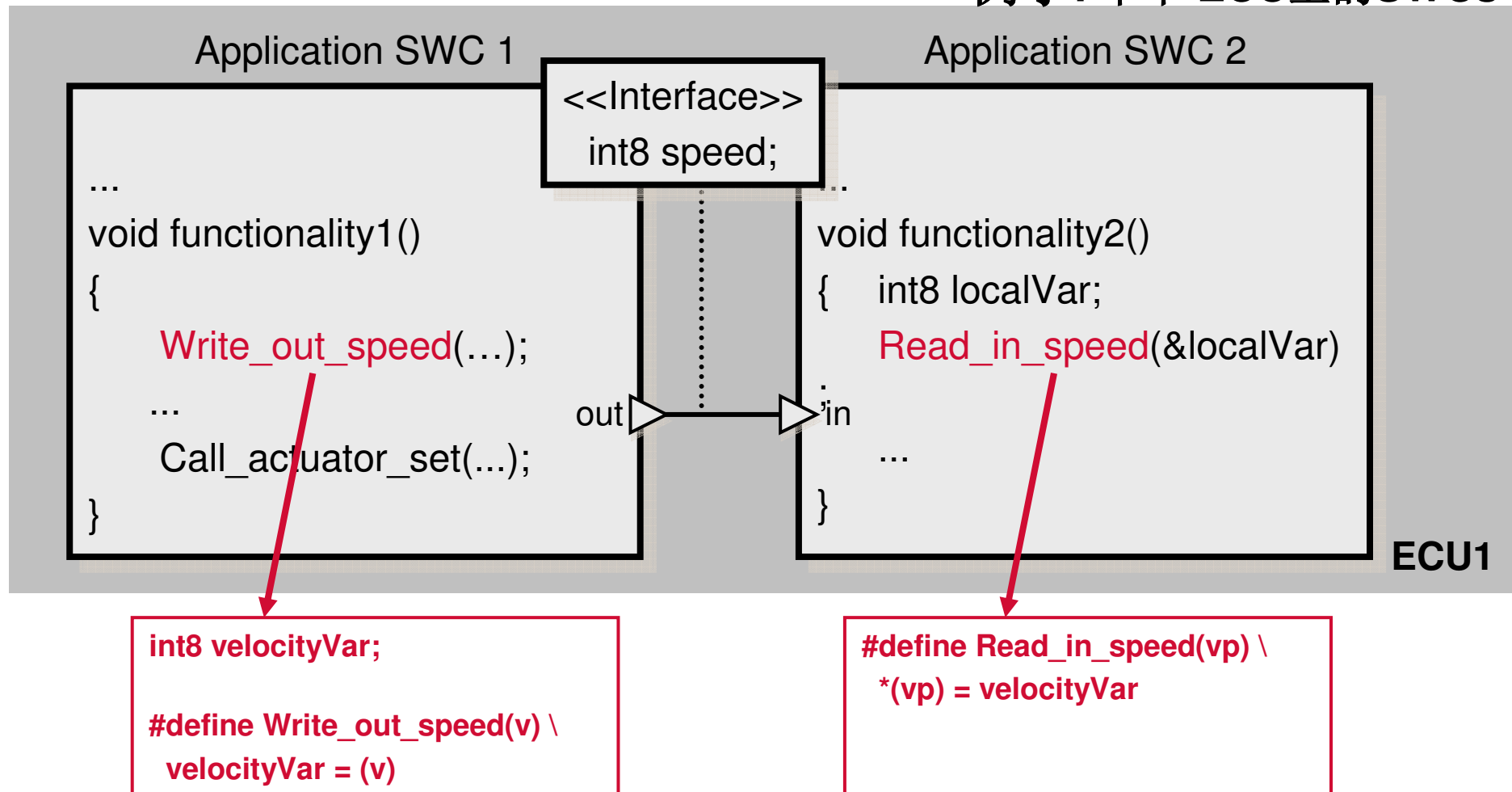
- ▶ 图形化的 ECU 配置描述 (ECD)
- ▶ ECD 的输入 / 输出
- ▶ 使用预编译方式，可方便配置 AUTOSAR BSW



Source: Elektrobit Automotive

AUTOSAR RTE 生成

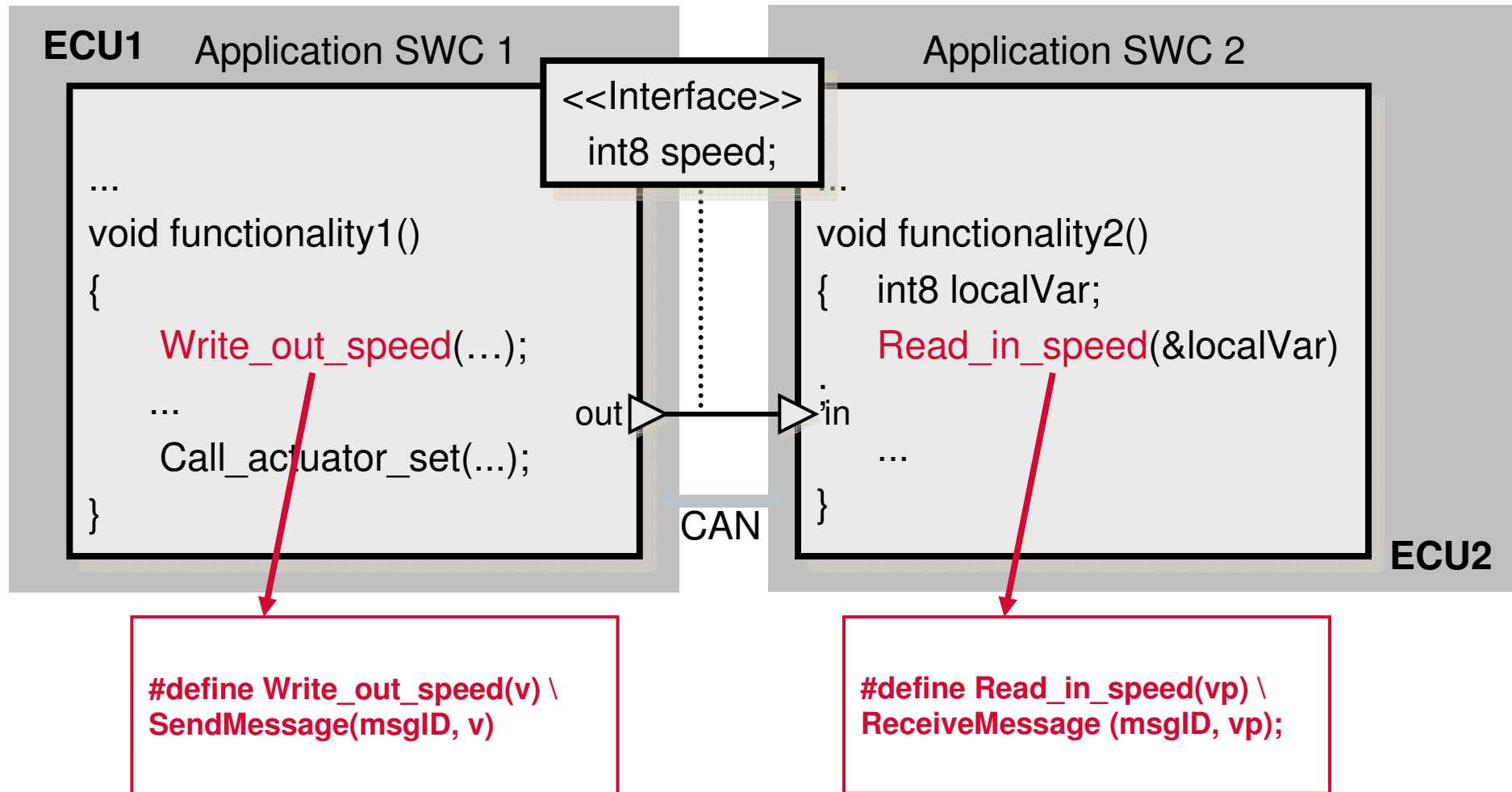
例子: 单个 ECU 上的 SWCs



Source: dSpace

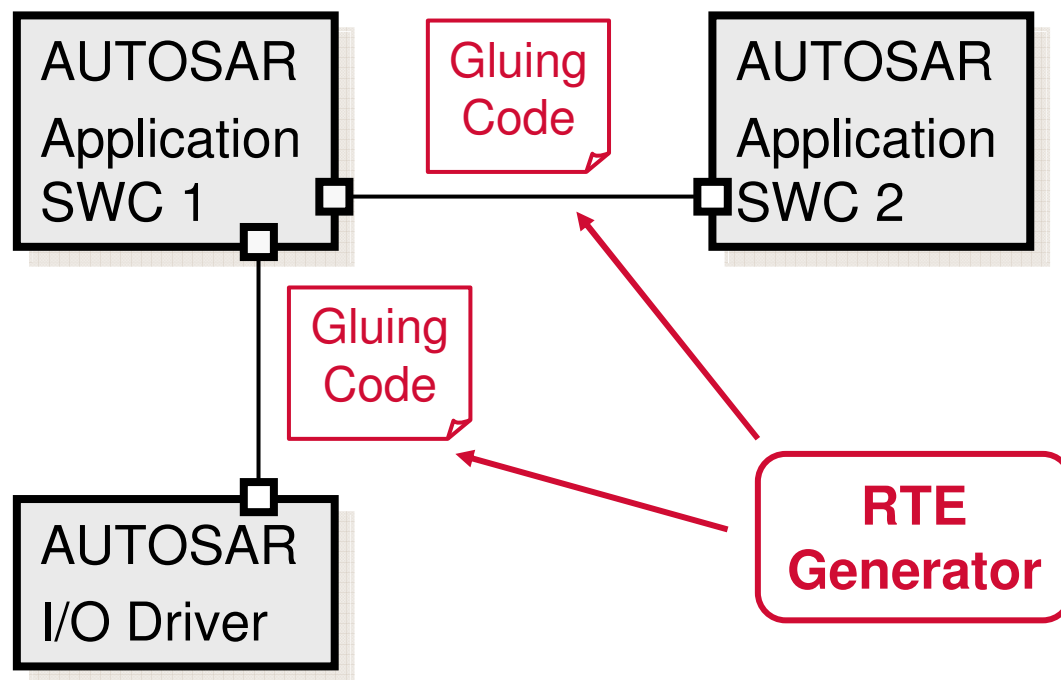
AUTOSAR RTE 生成

例子: 两个ECUs上的SWCs



Source: dSpace

AUTOSAR RTE 生成



This environment and ECU dependent “gluing” code is generated by the so-called AUTOSAR RTE Generator.

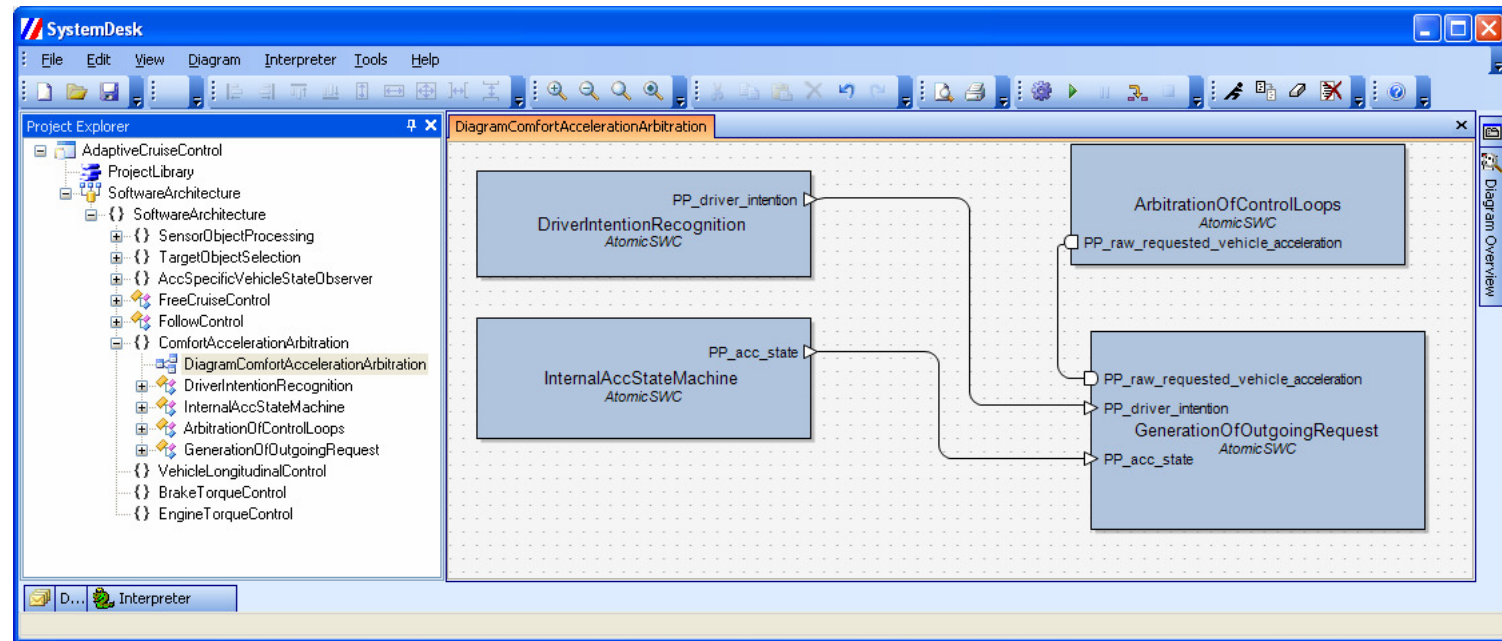
Source: dSpace

AUTOSAR 应用程序生成

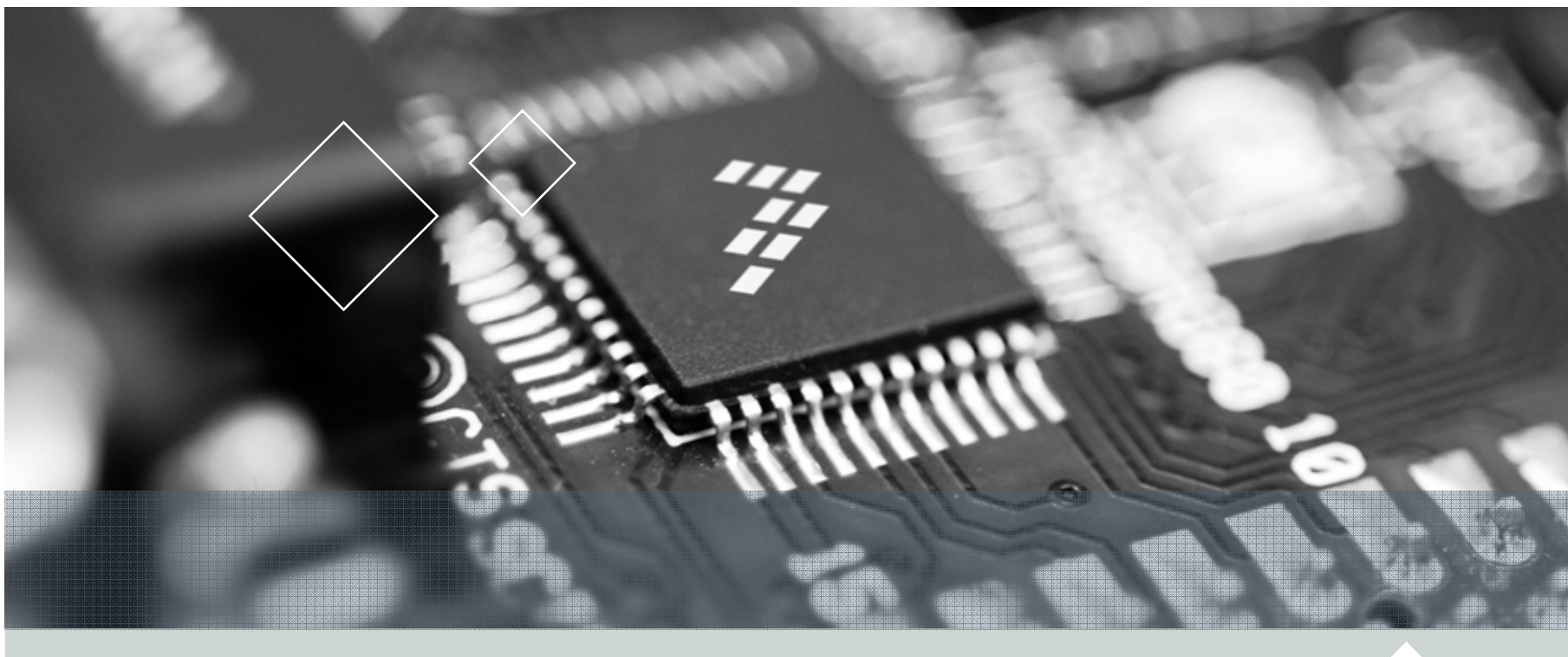
例子: SystemDesk™

► SystemDesk 以以下方式支持 ECU 网络应用程序的开发:

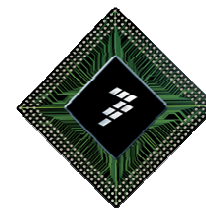
- 设计分布式的系统和软件结构
- 将软件模块集成在一个软件架构和ECUs上
- 符合 AUTOSAR 模型
- 集成与功能开发工具上, 如 TargetLink



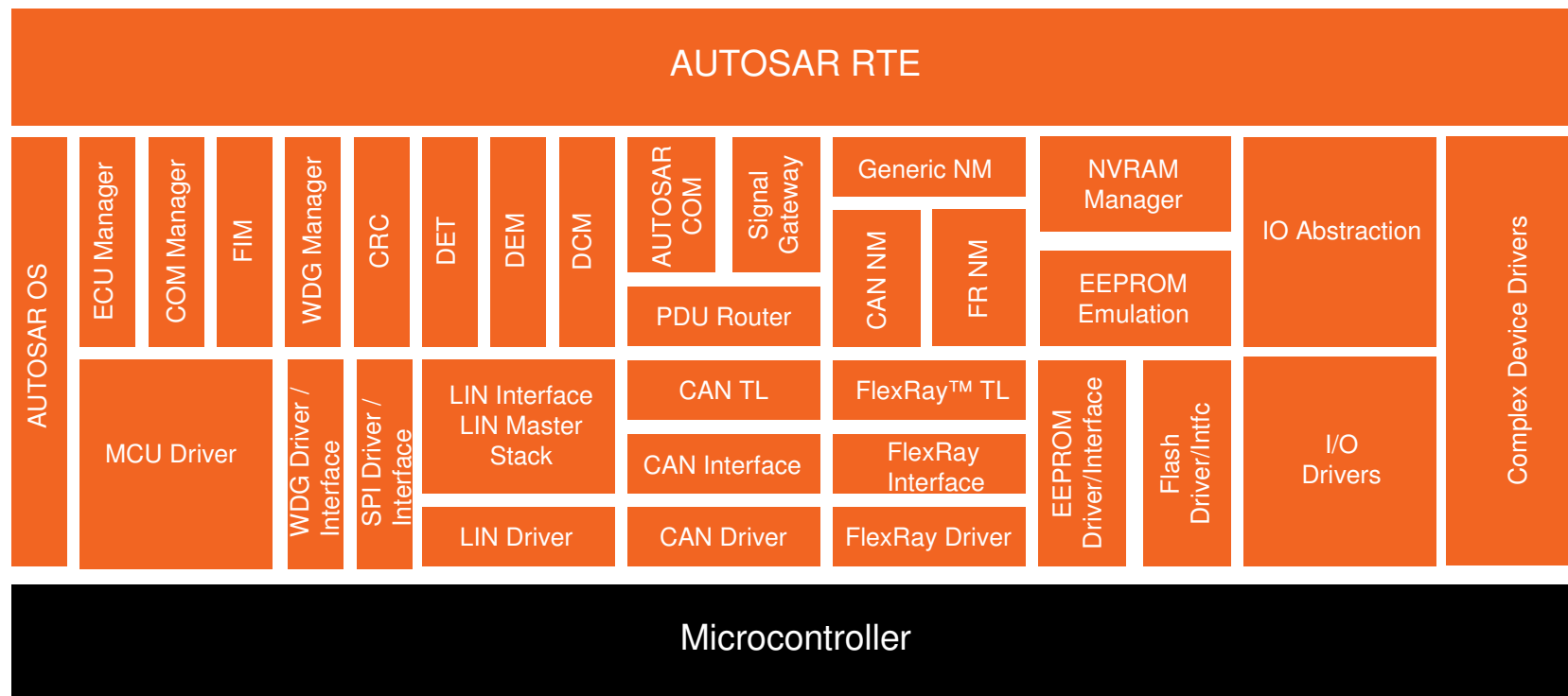
Source: dSpace



针对飞思卡尔控制器的AUTOSAR 软件



Freescal 提供的是 AUTOSAR 基础软件



AUTOSAR 软件产品选项

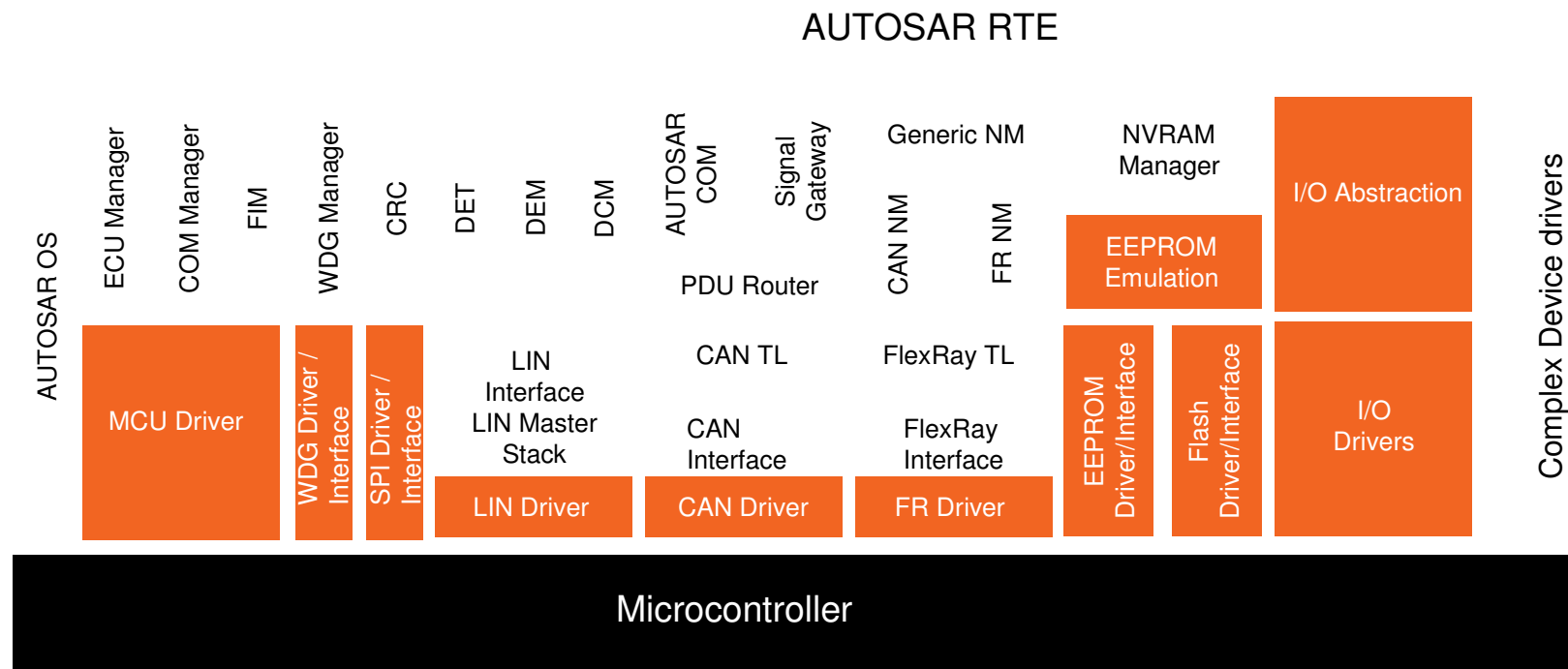
来自于 AUTOSAR 软件“产品包”：

- 1) SPAL Plus = MCAL + EEPROM emulation + IO abstraction layer
- 2) Operating system
- 3) Add-on services



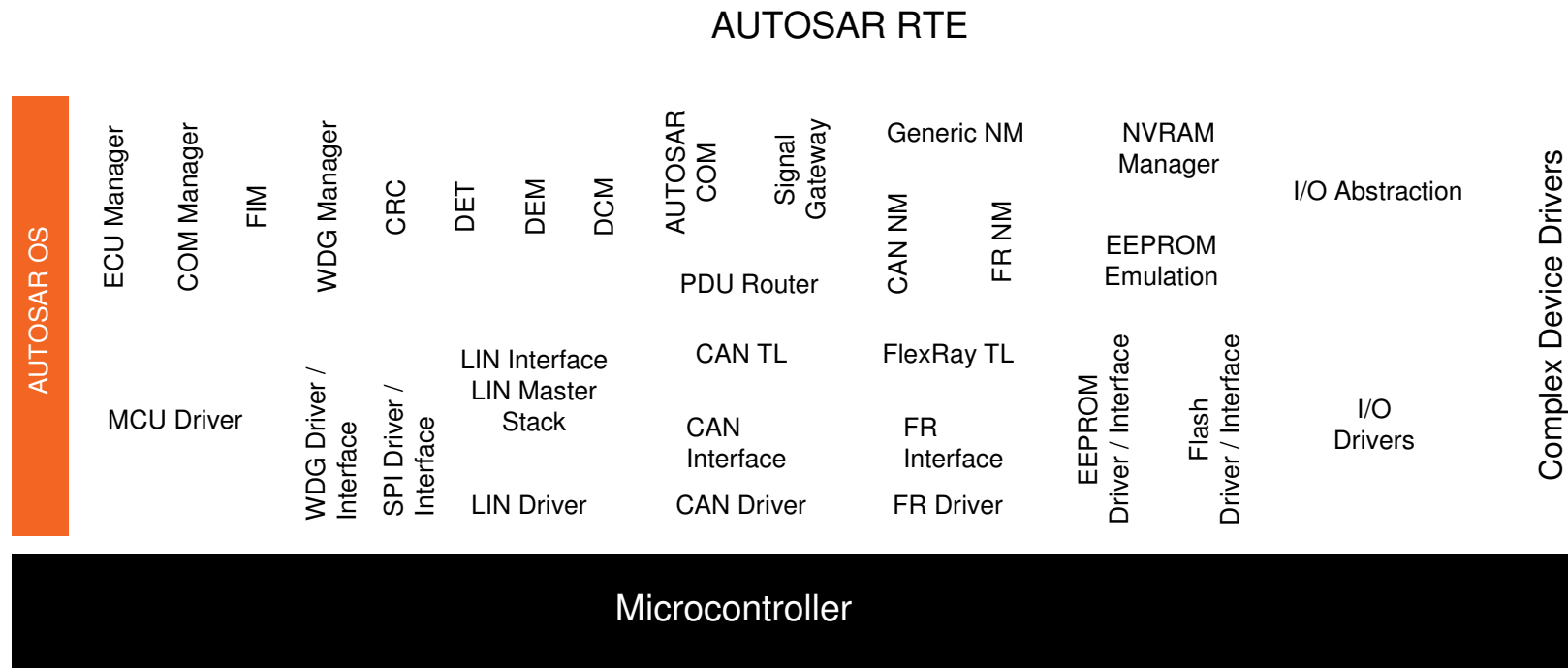
SPAL Plus

- ▶ MCAL + EEPROM emulation + IO abstraction layer
- ▶ 所有部件可使用符合 AUTOSAR 规范的配置工具配置



AUTOSAR 操作系统

- ▶ 可使用符合 AUTOSAR 规范的配置工具配置
- ▶ 有四种可调配的级别 1, 2, 3, 4, 以适合不同的应用要求
 - SC1 – 确定性的 RTOS 基础 (tasks, events, counters, alarms, messages)
 - SC2 – 基于时间的任务管理 (low-latency, precise timing for periodic tasks)
 - SC3 – 任务级的内存保护 (MMU/MPU), 以避免安全系统的内存冲突问题
 - SC4 – 任务级的时间和内存保护, 充分利用芯片的功能, 以提供汽车领域 RTOS 的安全保护



► 质量体系

- SPAL Plus team: CMM level 3 flow, certified to ISO 9001-2000
 - move to CMMI in preparation
- OS team: CMM level 3 flow, certified to ISO 90003-2004
 - comparable to ISO 15504 (SPICE), ISO 12207, and CMMI continuous

► 测试流程

- 单元测试
- 系统测试
- 代码审阅

► MISRA (PClint)

- Freescale AUTOSAR 驱动是部分符合 MISRA 标准的，所有不符合的部分都在原程序中采用以下形式的注释表明：

```
// MISRA Rule #<id> deviation: <reason>
```

模块测试计划 / 测试报告

6. Test Cases

6.1. Function M

AUTOSAR Re
WDG070, WDG0

Test descripti

The Wdg_Init function
are needed for contro
NULL pointer, the erro

Requirements

WDG003, WDG010
WDG066
WDG001, WDG028
WDGIF015
WDG025, WDG070
WDG009

Memory Map for test
Base Address = RES
7

WDG_TC008 WDG

15

MSB X

Input Parameters:

- Variable name : *ConfigPtr
Type : Wdg_ConfigType

Sr. No.	Test Case
1	WDG_TC00
2	WDG_TC04

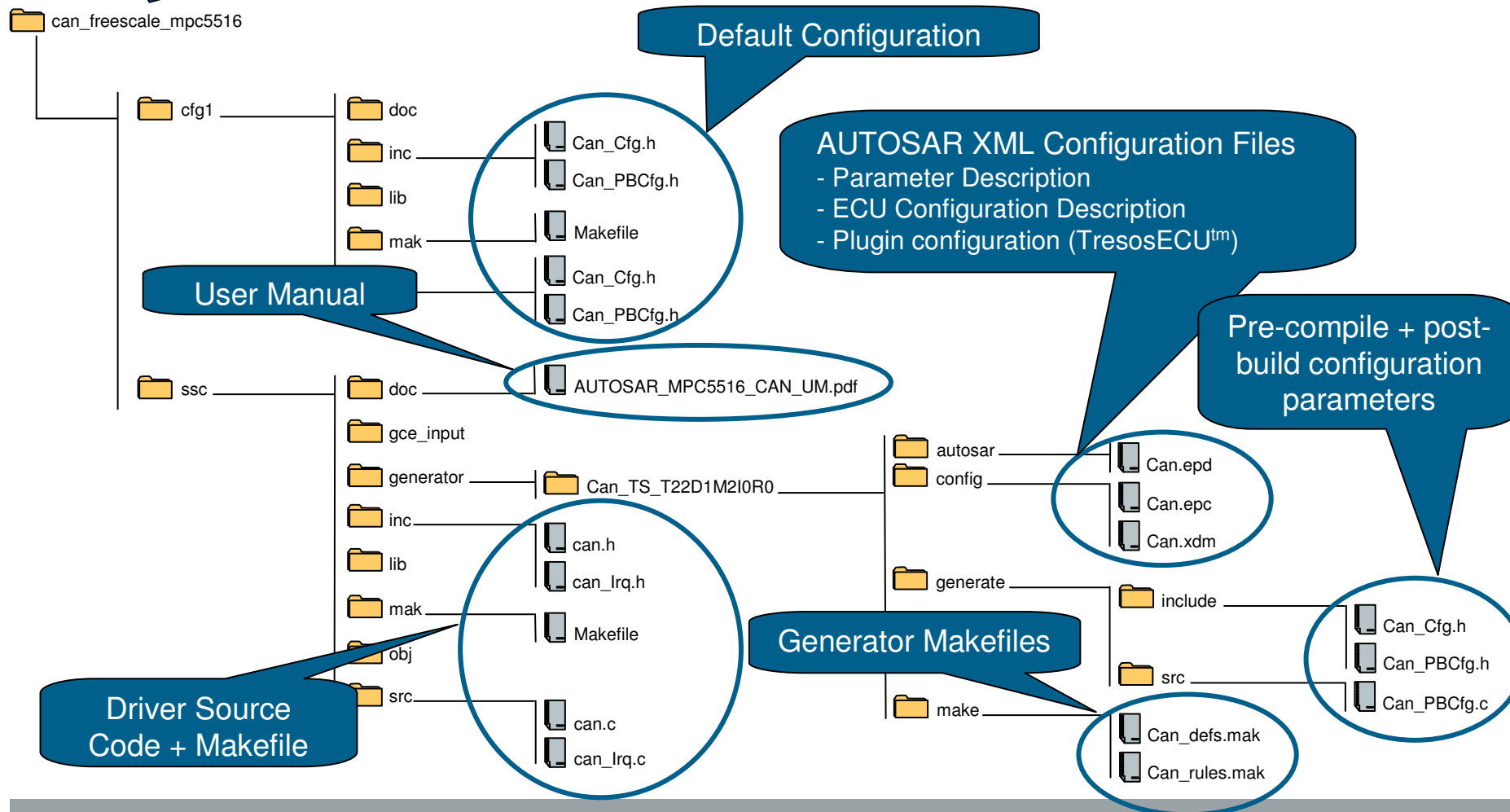
Input Parameters:

- Variable name : *ConfigPtr
Type : Wdg_ConfigType

Sr. No.	Test Case ID	Test Scenario	Expected Output	Actual Output	Result
1	WDG_TC001	<ul style="list-style-type: none">Enable development error detection switch before compilation.Disable Pre-compile support switch.Call Wdg_Init with NULL pointer.	Error should be reported to DET with following parameters, WDG_MODULE_ID (102), INSTANCE_ID (0), WDG_INIT_ID (0), WDG_E_PARAM_CONFIG (0x12)	Error was reported to DET with following parameters, WDG_MODULE_ID (102), INSTANCE_ID (0), WDG_INIT_ID (0), WDG_E_PARAM_CONFIG (0x12)	Pass
2	WDG_TC047	<ul style="list-style-type: none">Enable Pre-compile support switch.Call Wdg_Init with NULL pointer.	<p>The global variable "Wdg_CurrentMode" should be initialized to the default mode (as in the Wdg_Cfg.c).</p> <p>The Registers related to watchdog (HID0, TCR) should be initialized as per the configuration parameters' value.</p> <p>The status fields ENW, WIS and WRS in the TSR register should be cleared.</p> <p>The Watchdog Internal Status should be set to IDLE state after returning from the function.</p>	<p>The global variable "Wdg_CurrentMode" was initialized to the default mode (as in the Wdg_Cfg.c).</p> <p>The Registers related to watchdog (HID0, TCR) was initialized as per the configuration parameters' value.</p> <p>The status fields ENW, WIS and WRS in the TSR register was cleared.</p> <p>The Watchdog Internal Status was set to IDLE state after returning from the function.</p>	Pass

Freescal SPAL Plus 软件包内容

Example: CAN Driver for MPC5516



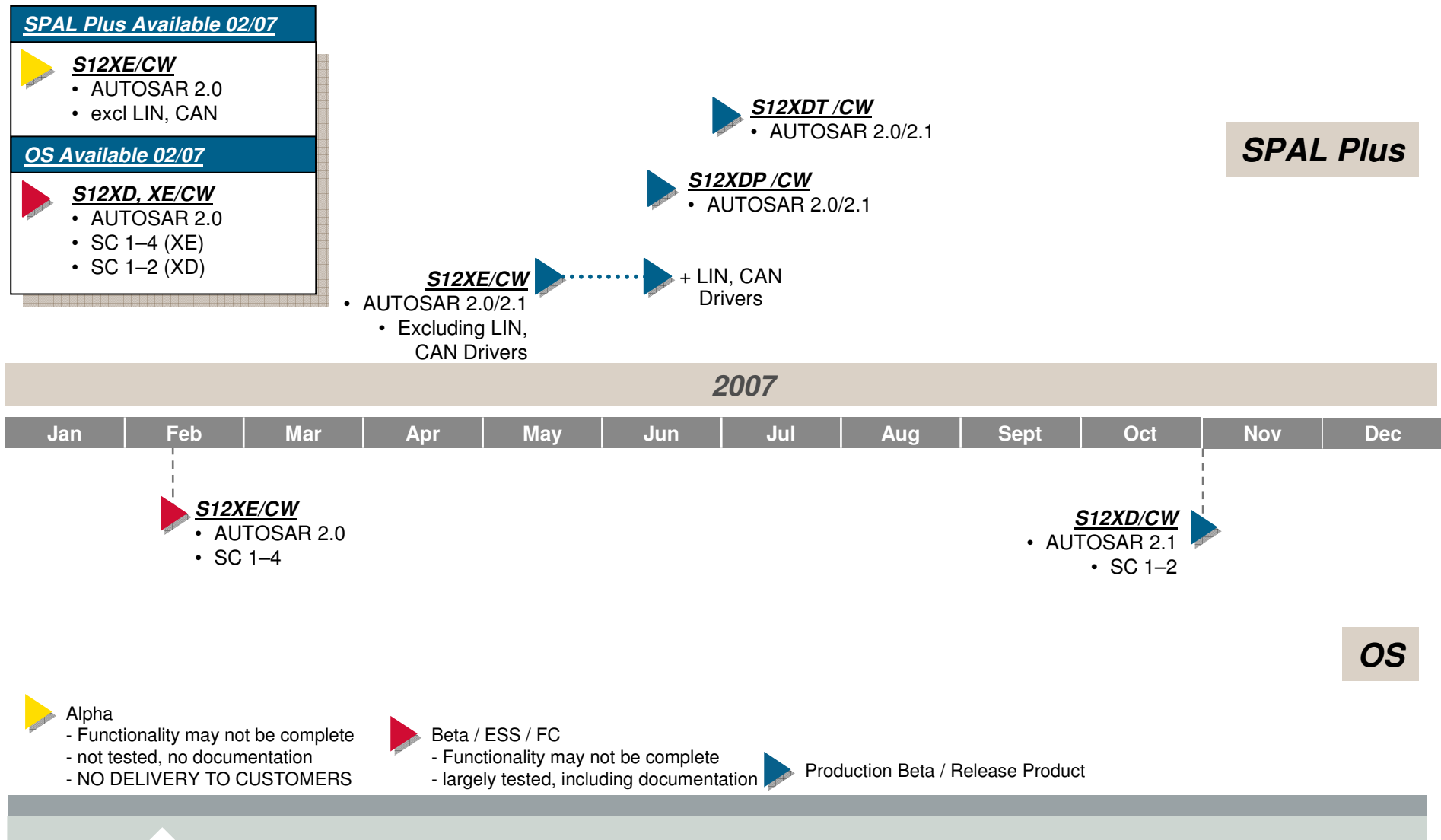
FreescalE 提供的 AUTOSAR 软件

► 以下 Freescale 处理器结构将支持 AUTOSAR 标准的产品级软件:

- 16-bit : S12XE family
- 16-bit : S12XD family
- 32-bit : MPC556x family
- 32-bit : MPC551x family

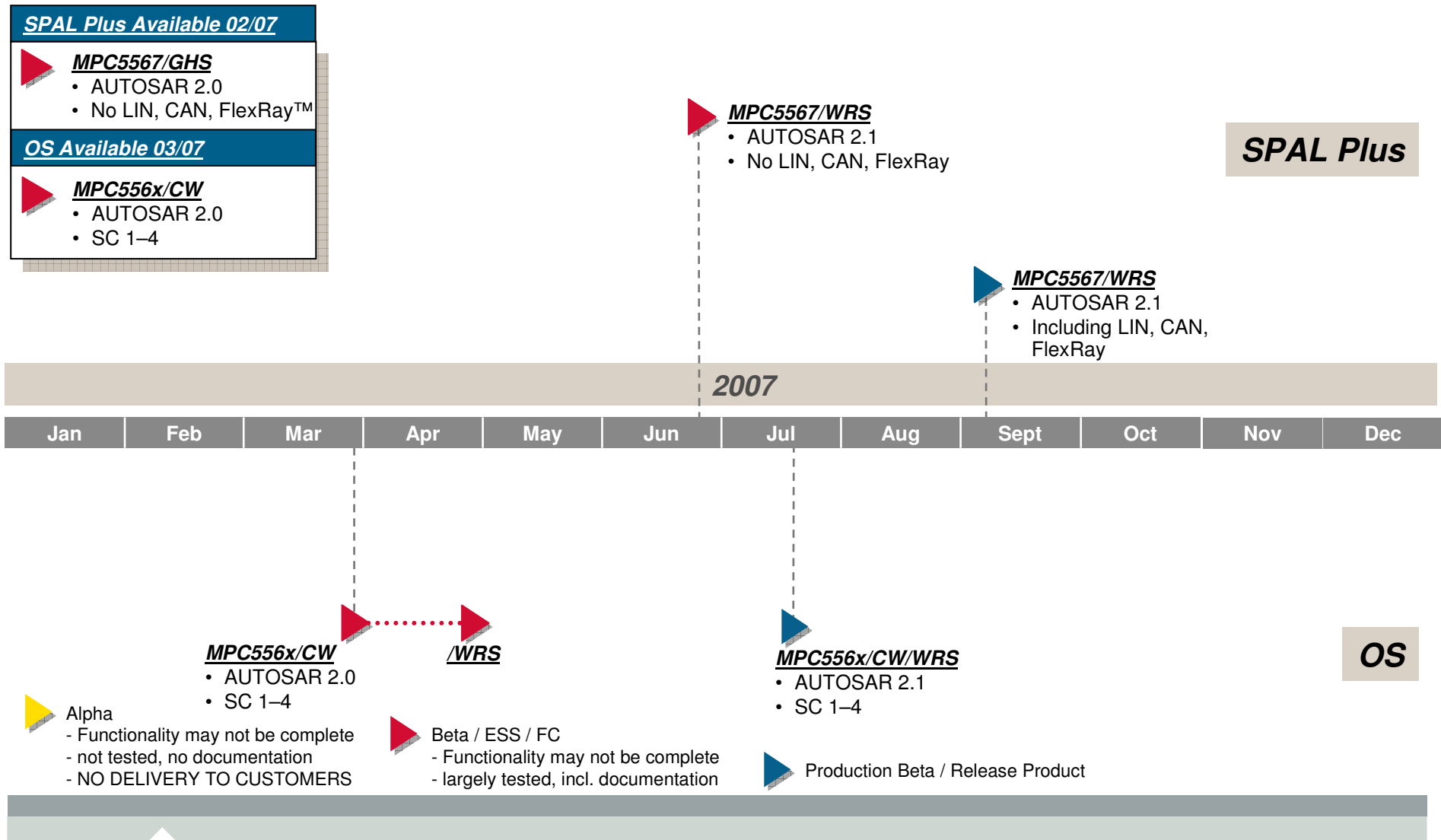
S12X AUTOSAR OS/SPAL Roadmap 2007

(subject to change)



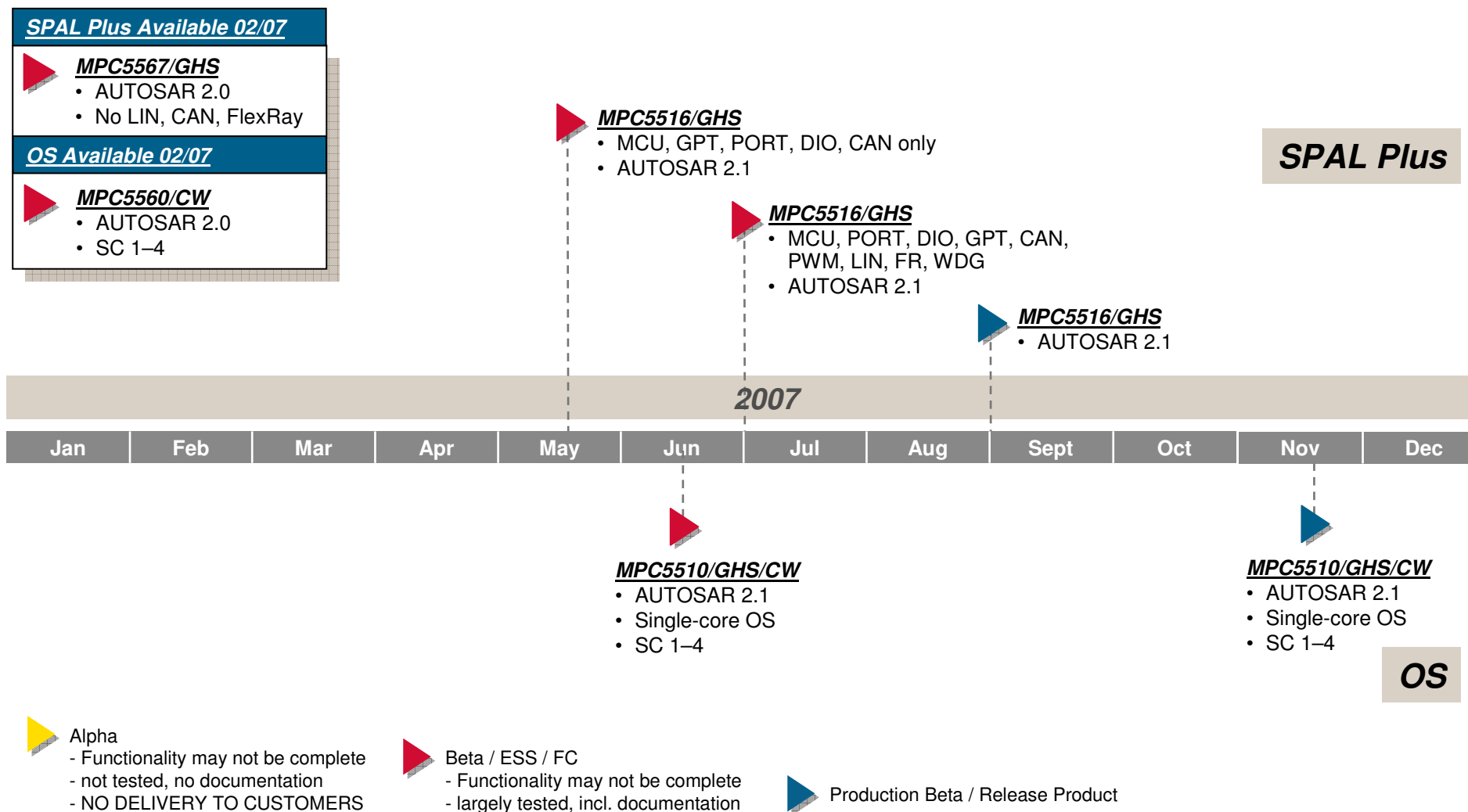
MPC556x AUTOSAR OS/SPAL Roadmap 2007

(subject to change)



MPC551x AUTOSAR OS/SPAL Roadmap 2007

(subject to change)



AUTOSAR 软件商业模式

AUTOSAR 标准软件产品

- SPAL Plus
- Operating system
- Add-on services

服务和咨询

- To “customize” standard products
- To migrate from legacy software

质量保证

- CMM/SPICE compliant development processes
- Process documentation available upon request

标准支持

- Basic phone and e-mail support
- Support fee is percentage of list price

特别支持

- Dedicated resources at Freescale
- Agreed upon definitions of severity levels and resolution times
- Custom service level agreements with customer; pricing to be negotiated

免版税的价格模式

- **Project Buyout**—Pay per **project** per **core**; example: AUTOSAR product for 2009 BMW Door Module using S12XE
- **Core Buyout**—Pay per **core**; use in unlimited projects. Example: AUTOSAR product for S12XE

责任

- Restricted liability in line with software pricing

为什么选择 Freescale 的 AUTOSAR 软件？

► 优化—我们长期的承诺

- Freescale 掌握芯片技术的细节，直到如何最好的使用芯片资源以提供软件模块的优化。

► 质量

- 采用已经验证的开发流程 (CMM/SPICE)，提供 AUTOSAR 软件给主要的汽车客户，以驱动质量和成本的有效性

► 紧随芯片的发展蓝图

- Freescale 支持 S12X 和 MPC55xx MCU 产品系列的 AUTOSAR 软件；客户可选择我们的 16- and 32-bit MCU 产品系列，加上 AUTOSAR 软件的支持。

► 全球支持

- 通过我们的全球客户支持体系架构。

► 软件服务团对

- 帮助将 Freescale AUTOSAR 软件集成于客户的软件架构体系和客户的模块开发中

- ▶ **AUTOSAR** 描述了一个适用于汽车领域的完整的软件架构，包括基础软件包，灵活的配置方法，以及软件工具的接口和应用指导
- ▶ 可现用的软件模块，加上完备的配置方法将改变您使用 **Freescale** 微控制器的方式
 - 减少 **ECU** 开发时间
 - 增强基础软件模块的重复使用率和质量保证
 - 增强应用软件模块的重复使用率和质量保证
- ▶ **Freescale** 通过提供针对于我们**16位**和**32位**汽车级微控制器的可生产级的**AUTOSAR** 软件，减轻了客户向 **AUTOSAR** 软件转变的难度
 - 操作系统
 - **SPAL Plus** 软件包
 - 增值服务

