以汽车安全角度重新审视汽车操作系统

黄一智
湖南大学 嵌入式与网络计算湖南省重点实验室
huangyizhi@hnu.edu.cn

汽车是移动的智能手机？

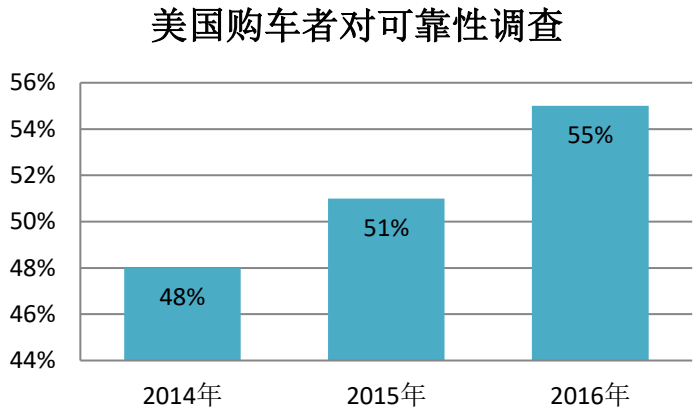
Tesla suffers complete network outage, internal systems and connectivity features down [Update: connectivity returning]

- 车联网
- ……

一切看起来都很美好，然而如果不能保证安全，再智能也没用！

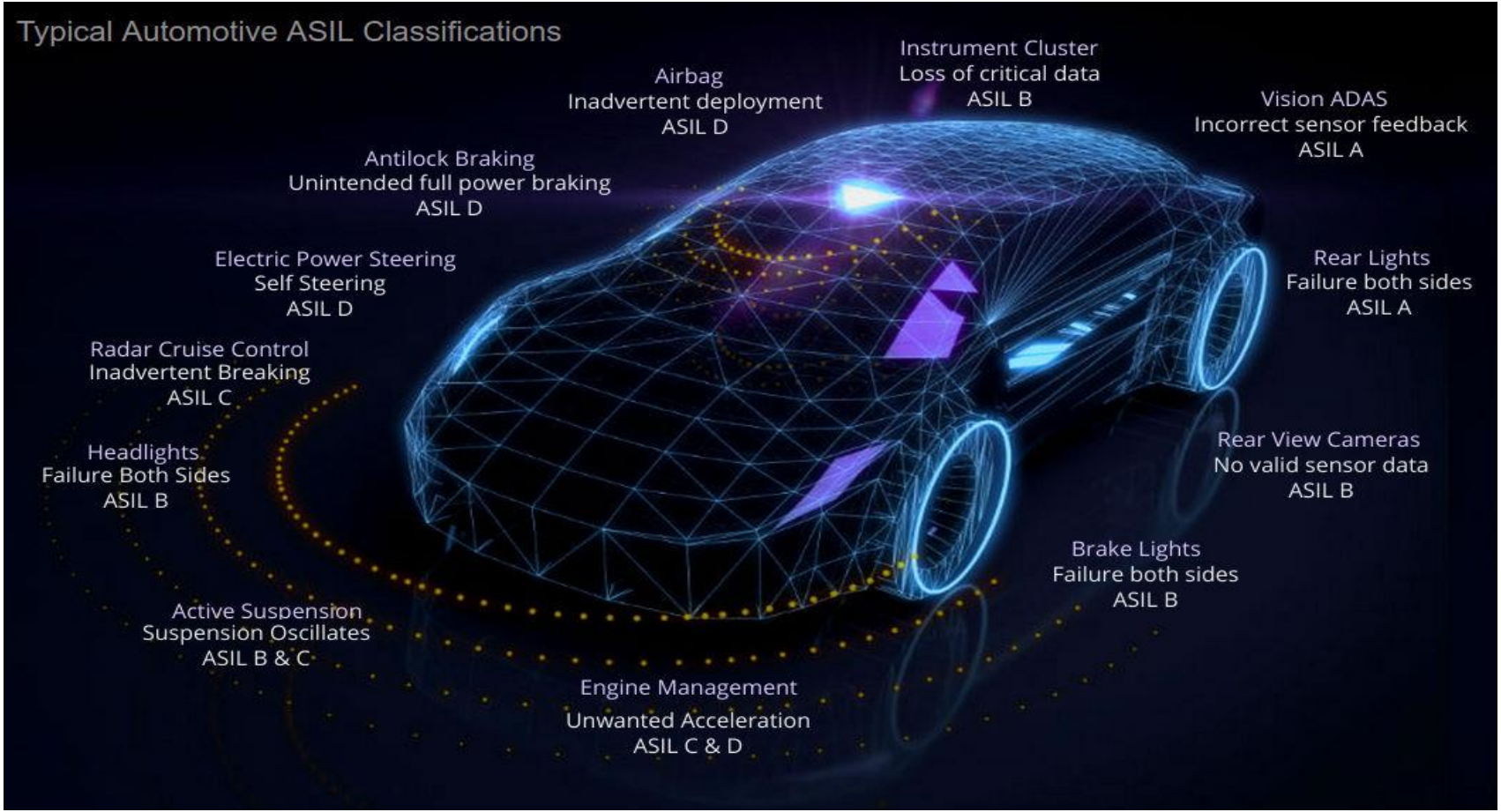
安全是汽车的第一属性

- 使用者最关心的属性



- 超过55%的美国购车者首要关心汽车可靠性（2016）
- 38%的加拿大受访者认为安全是最关注的特性，其次才是价格（2018）
- 超过84%的加拿大人在有孩子以后会更关注安全（2018）
- 超过81.8%的美国受访者关心自动驾驶中设备故障或系统失效后的安全情况（2014）

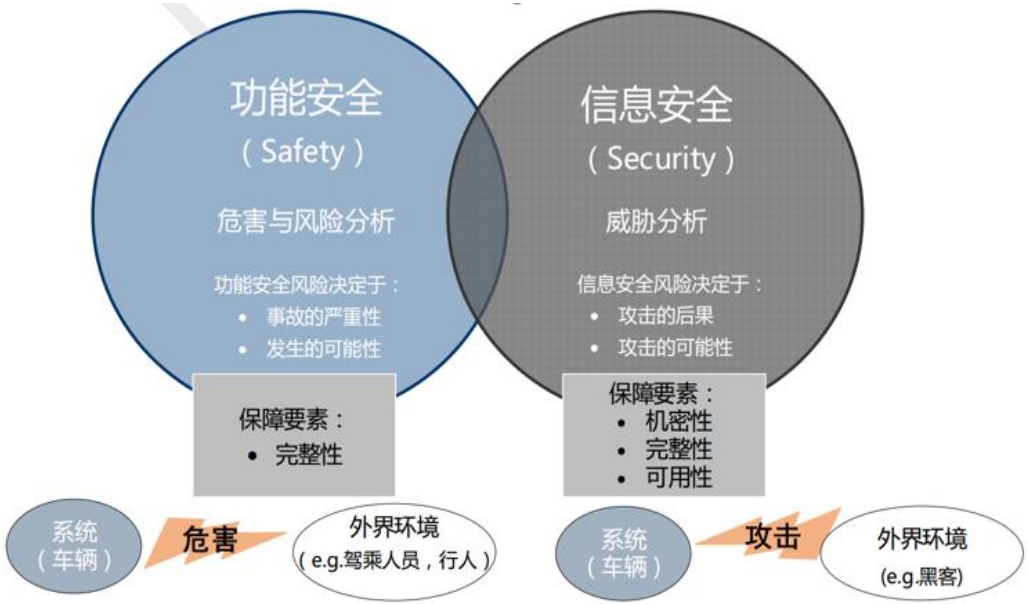
安全是汽车的第一属性


Typical Automotive ASIL Classifications
Instrument Cluster
Loss of critical data
ASIL B
Airbag
Inadvertent deployment
ASIL D
Vision ADAS
Incorrect sensor feedback
ASIL A
Antilock Braking
Unintended full power braking
ASIL D
Electric Power Steering
Self Steering
ASIL D
Rear Lights
Failure both sides
ASIL A
Radar Cruise Control
Inadvertent Breaking
ASIL C
Rear View Cameras
No valid sensor data
ASIL B
Headlights
Failure Both Sides
ASIL B
Brake Lights
Failure both sides
ASIL B
Active Suspension
Suspension Oscillates
ASIL B & C
Engine Management
Unwanted Acceleration
ASIL C & D

什么是汽车安全？

- 功能安全：避免因电气/电子系统故障而导致的不合理风险（ISO 26262）
  - ✓ OS保证功能完整性，减少由于软件错误引起的事故发生
  - ✓ OS的任务管理
  - ✓ OS的资源管理

- 信息安全：（ISO 21434）
  - ✓ Information Security
  - ✓ Cyber Security
  - ✓ Security IPC
  - ✓ 道高一尺，魔高一丈的问题



以汽车安全角度重新审视汽车操作系统

1、当前汽车操作系统能够维护现代汽车安全

2、当前汽车操作系统无法维护未来汽车安全

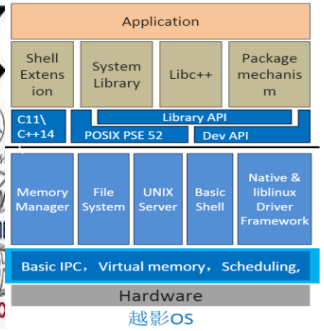
以汽车安全角度重新审视汽车操作系统

1、当前汽车操作系统能够维护现代汽车安全

2、当前汽车操作系统无法维护未来汽车安全

当前汽车操作系统——分布式多操作系统

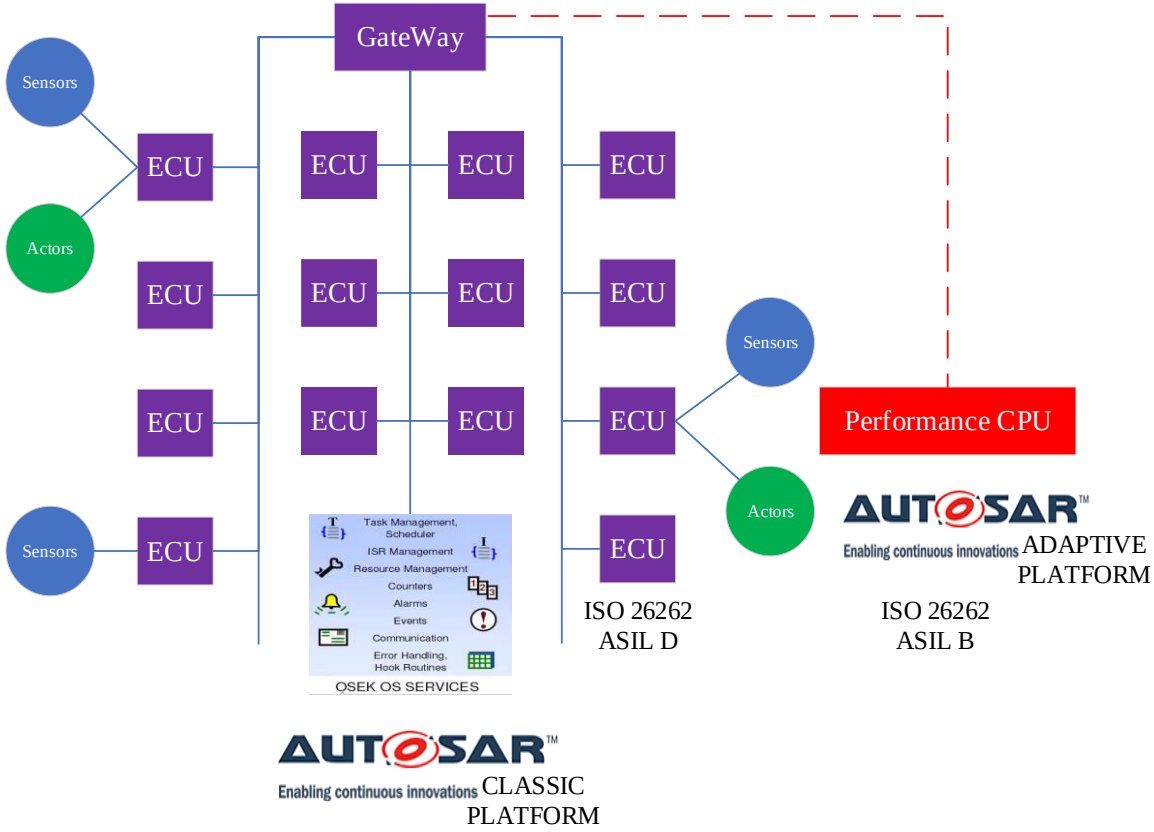
- 参与控制汽车功能的OS与负责车载计算/娱乐的OS

运行在汽车内部ECU上	运行在车载计算平台	
分布式Real-time OS	基于Linux kernel OS	类Unix Real-time OS
ERIKA 3  VECTOR TOPPERS Toyohashi OPen Platform for Embedded Real-time Systems	TESLA	QNX ACURA Audi HONDA NISSAN CHRYSLER GM HYUNDAI CHEVROLET LAND ROVER PORSCHE SAAB TOYOTA  越影OS
当前真正影响汽车安全		
• 硬实时操作系统 • 分布运行在汽车ECU上 • 汽车级标准约束（OSEK/VDX） • Autosar CP OS（Autosar Classic Platform OS）	• Tesla Version, 基于Linux内核的OS • 搭载Android Automotive OS的汽车 • Autosar AP OS（Autosar Adaptive Platform OS）	• QNX, 类Unix实时操作系统, NVIDIA DRIVE AGX支持 • 华为越影OS, 支持Linux接口的实时操作系统, 自研MDC平台支持 • Autosar AP OS（Autosar Adaptive Platform OS）

- 两类OS功能上有很强的隔离（具备辅助驾驶能力的汽车除外）

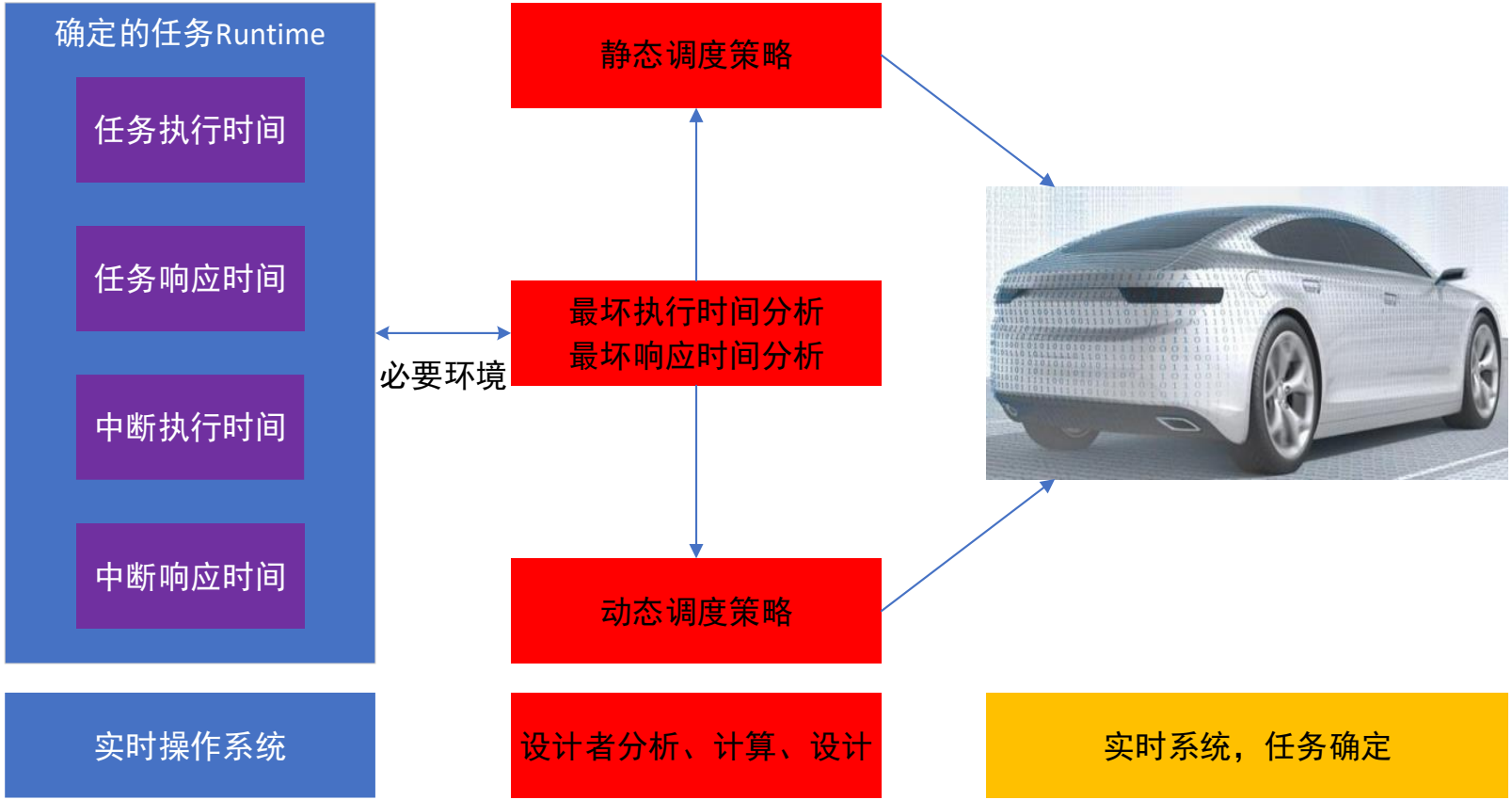
当前汽车操作系统——分布式多操作系统

- 第一类OS运行在ECU上
  - ✓ 硬实时OS（每个ECU上）
  - ✓ ECU性能相对低下
- 第一类OS遵循标准
  - ✓ OSEK/VDX
  - ✓ Autosar Classic Platform
  - ✓ ISO 26262 ASIL D
- 第一类OS称为Autosar CP OS
- 第二类OS运行在车载计算机
  - ✓ 多核异构
  - ✓ 软实时
- 第二类OS遵循标准
  - ✓ Autosar Adaptive Platform
  - ✓ ISO 26262 ASIL B
- 第二类OS称为Autosar AP OS
- 两类OS弱连接



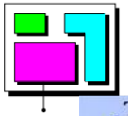
当前汽车操作系统的安全保证——实时性保证

- ECU中的任务执行必须是确定的， Autosar CP OS提供实时环境保证

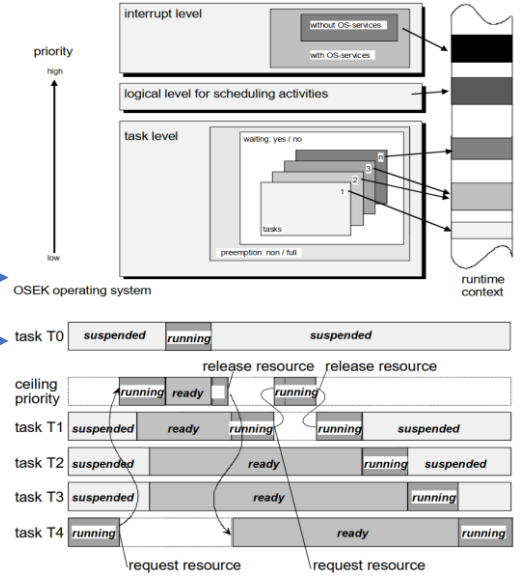


当前汽车操作系统的安全保证——实时性保证

- 当前Autosar CP OS的实时环境保证
  - 可抢占的多优先级、多任务、静态调度机制
  - 可预测的同步机制
  - 防护优先级倒置
  - 多核处理



OSEK/VDX



- 确保任务执行的确定性，保证安全

AUTOSAR CLASSIC PLATFORM

限制

- 启动后不支持加入附加核心
- 调度算法不允许动态分配任务
- 资源只允许本核使用不支持跨核心

以汽车安全角度重新审视汽车操作系统

1、当前汽车操作系统能够维护现代汽车安全

- 实时设计原则
- 静态设计原则
- 冗余设计原则

保证功能完整性，减少由于软件错误引起的事故发生

2、当前汽车操作系统无法维护未来汽车安全

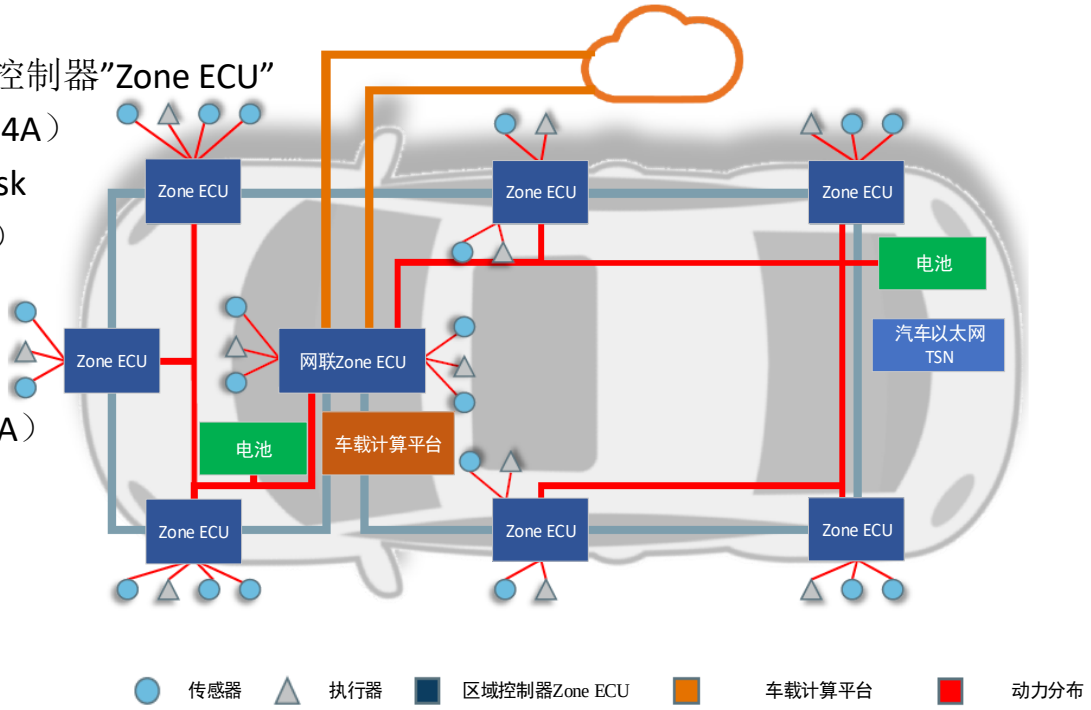
以汽车安全角度重新审视汽车操作系统

1、当前汽车操作系统能够维护现代汽车安全

2、当前汽车操作系统无法维护未来汽车安全

未来汽车E/E系统结构变迁——集中式的边端云结构

- 汽车内部E/E系统——区域控制
  - ✓ ECU大量减少，功能集中到区域控制器”Zone ECU”
  - ✓ 多核异构处理器（如NXP S32G274A）
  - ✓ 实现更多Functions，执行更多Task
  - ✓ 网联（Vehicle to everything, V2X）
- 车载计算平台
  - ✓ 高性能多核异构（CPU/GPU/FPGA）
  - ✓ 智能驾驶决策
  - ✓ 与汽车E/E系统强连接（控制）
  - ✓ ASIL B -> ASIL D
- 云端



未来汽车操作系统面临的挑战

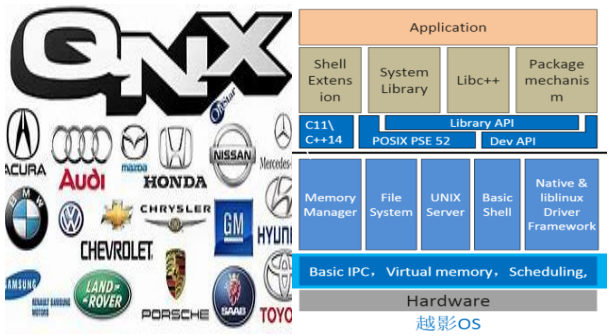
运行在汽车E/E系统

运行在车载计算平台

分布式Real-time OS

基于Linux kernel OS

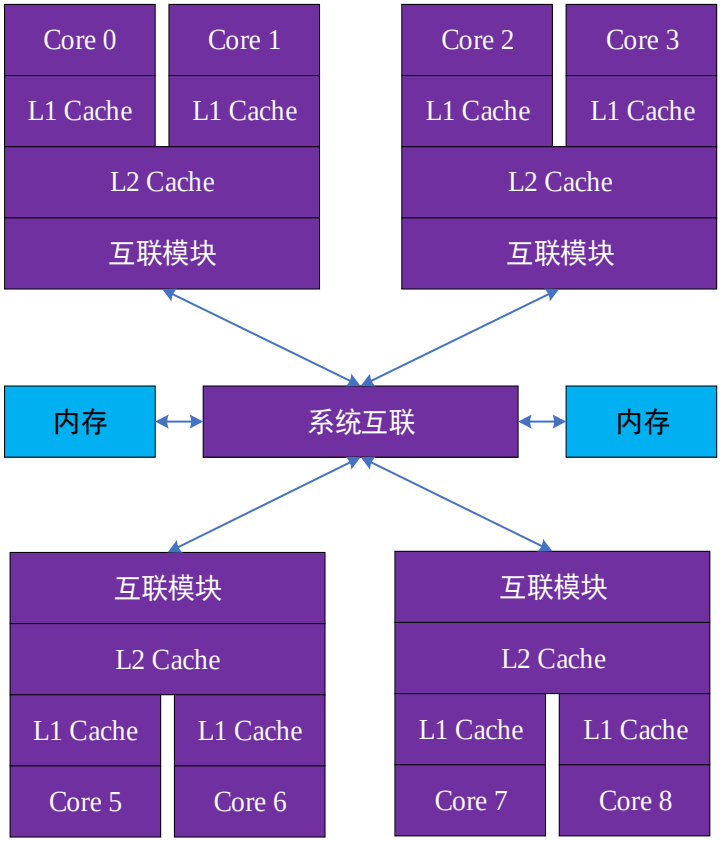
类Unix Real-time OS



- 所有的OS都会影响汽车的行驶安全
- 所有的OS都必须提供必要实时环境
- 所有的OS都必须达到最高安全等级

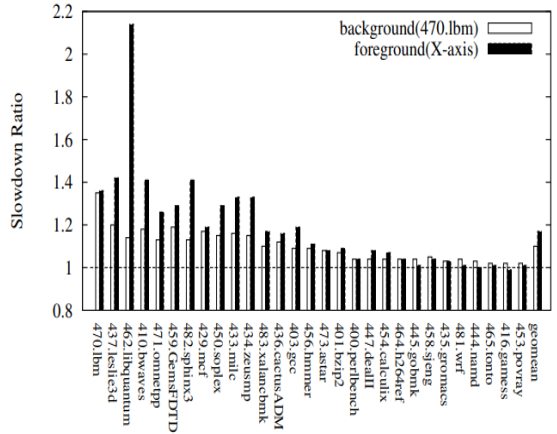
当前汽车操作系统无法维护汽车E/E系统的实时性

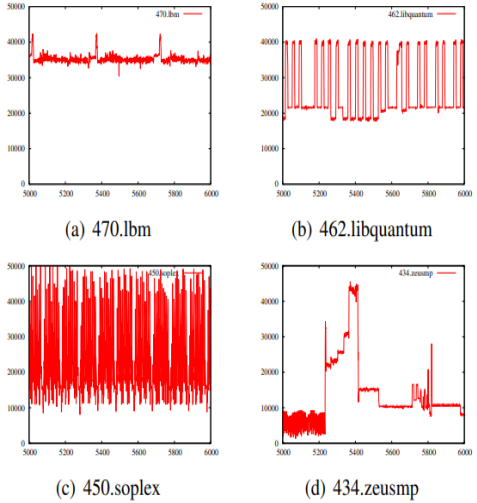
- 多核系统中大量任务执行的不确定性
  - ✓ 共享Cache的竞争（Cache Line被多任务相互驱逐）
    - ➢ Core 0上两个Task访问L1的同一Cache Line
    - ➢ Core 0上两个Task访问L1的不同Cache Line
  - ✓ 车载计算平台上，任务在不同核心上的迁移
    - ➢ Core 0上的Task迁移到Core 1
    - ➢ Core 0上的Task迁移到Core 2
  - ✓ 车载计算平台上，任务的并行执行
    - ➢ Core 0上串行执行Task A、Task B
    - ➢ Core 0和Core 1并行执行Task A、Task B
  - ✓ 应用程序自身问题
    - ➢ 全局原子操作
  - ✓ ……

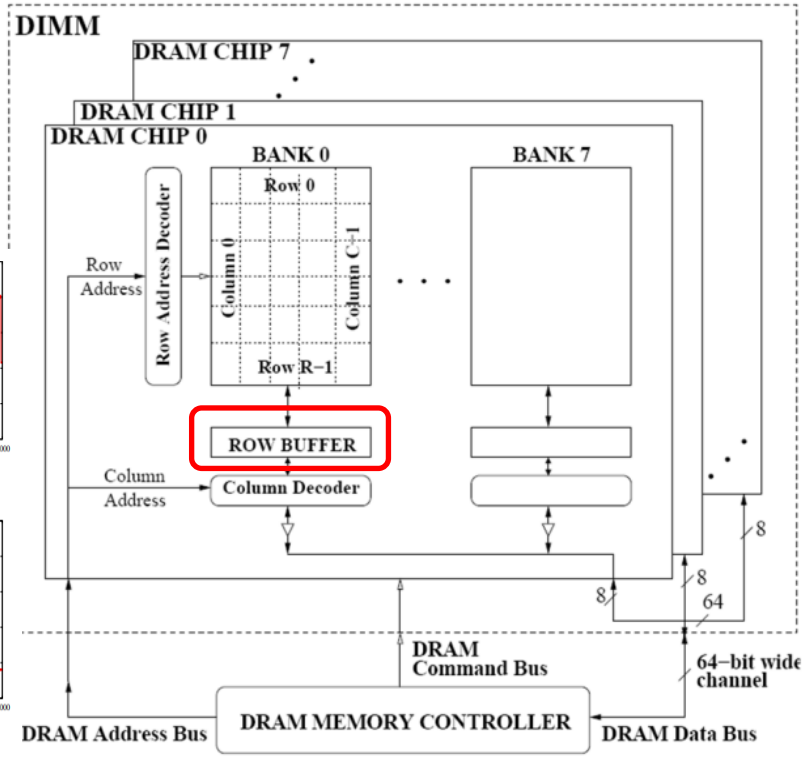


当前汽车操作系统无法维护车载计算平台的实时性

- 共享DDR内存系统下多任务执行相互干扰的不确定性
  - ✓ 多任务对ROW BUFFER的驱逐（Bank Conflict）
    - ➢ 即使只有两个任务，也会使任务执行非常不稳定
  - ✓ 多级并行的延迟差异
  - ✓ 以MMU或者MPU为基础的内存隔离/保护手段无法解决（核心问题物理地址与DDR BANK的映射）







- 现有Autosar OS内存保护标准无法解决

4.5.2.1 [SRS_Os_11005] The operating system shall prevent an OS-Application from modifying the memory of other OS-Applications

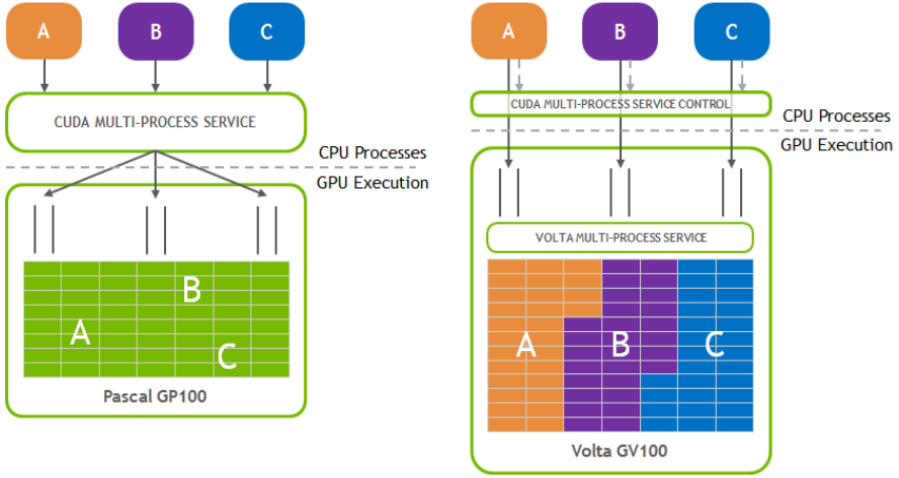
[

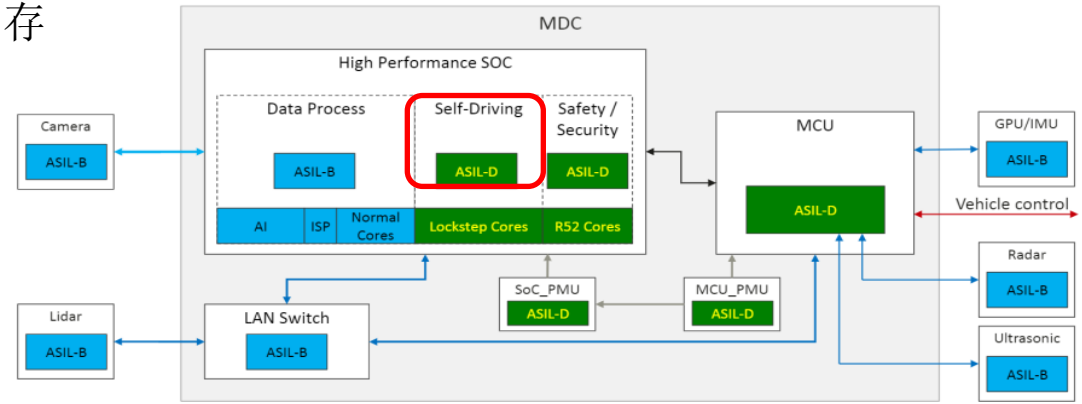
Type:	Valid
Description:	The operating system shall provide the ability of partitioning OS-Applications with respect to memory and prevent an OS-Application from modifying the memory of other OS-Applications.
Rationale:	Where multiple OS-Applications (of different software integrity) are resident on the same processor, their memory will be globally writable by any code. This means that the data of one OS–Application could be corrupted by another unrelated OS-Application (i.e. there is fault propagation between OS-Applications). For example a task of an OS-Application may overflow its stack, causing static data of an unrelated OS-Application to be corrupted, causing it to fail. To permit reasoning about adequate independence between the functions of different integrity levels, it is essential that this is prevented at runtime. Note that SRS_Os_11003 is different: It only detects fault rather than preventing a memory access error from generating a fault.
Use Case:	--
Dependencies:	Note that satisfying this requirement implies the satisfaction of the stack monitoring requirement as a stack overflow cannot occur if the stack is bounded by memory write access control. The write access protection needs appropriate hardware support.
Supporting Material:	

当前汽车操作系统无法维护车载计算平台的实时性

- GPU实时性的巨大挑战

  - ✓ 操作系统不维护GPU的任务调度
    - ➢ GPU硬件决定
  - ✓ GPU任务不可抢占
    - ➢ 多任务下都能获得GPU资源，且不可控
  - ✓ 集成GPU与多核CPU共享内存
    - ➢ 内存竞争问题更严重
  - ✓ GPU仅达到ASIL-B
    - ➢ 智能驾驶的关键决策计算





当前汽车操作系统无法保障未来汽车实时性

不确定的任务Runtime

任务执行时间

任务响应时间

中断执行时间

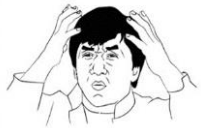
中断响应时间

多核/异构/内存竞争/...

AUTOSAR CP/AP OS

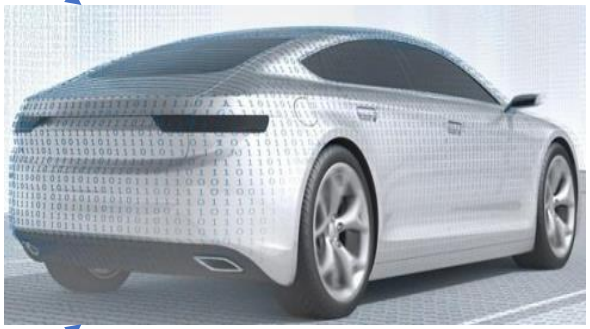
静态调度策略

最坏执行时间分析不准
最坏响应时间分析不准



动态调度策略

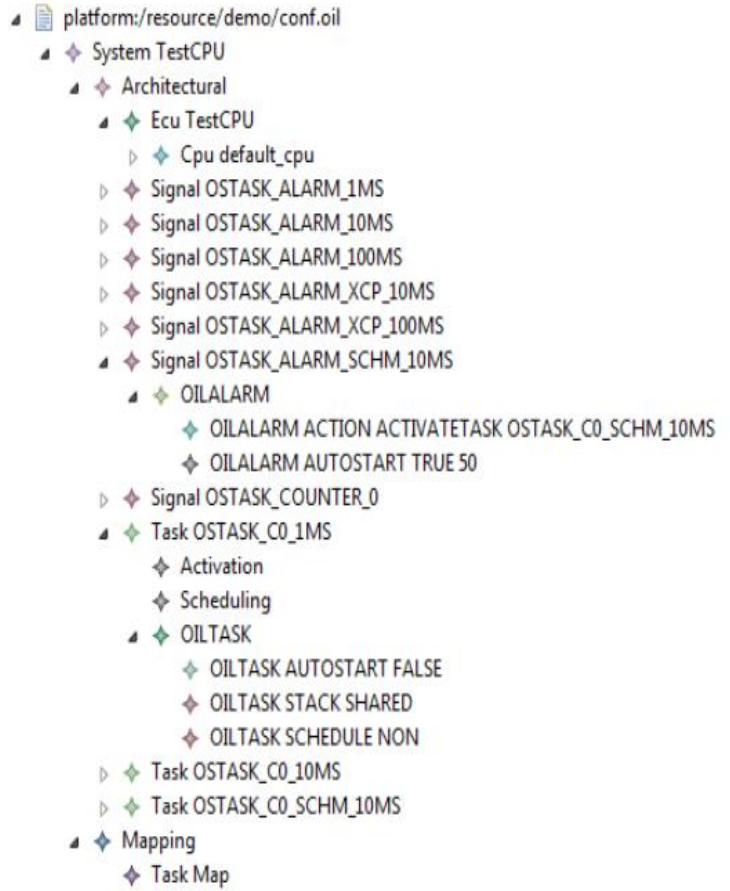
设计者无法准确分析、设计合理的任务排布



系统实时性无法保障

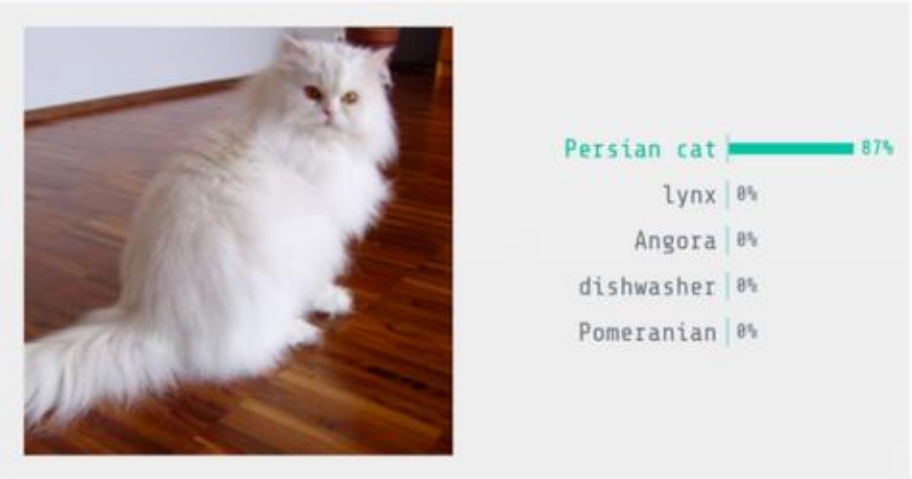
AUTOSAR CP OS静态设计在联网下的隐藏危机

- AUTOSAR CP OS中任务优先级、类型、调度是静态的
- AUTOSAR CP OS中任务变量、内存资源使用是静态的
- 联网状态下有外部被逆向的风险

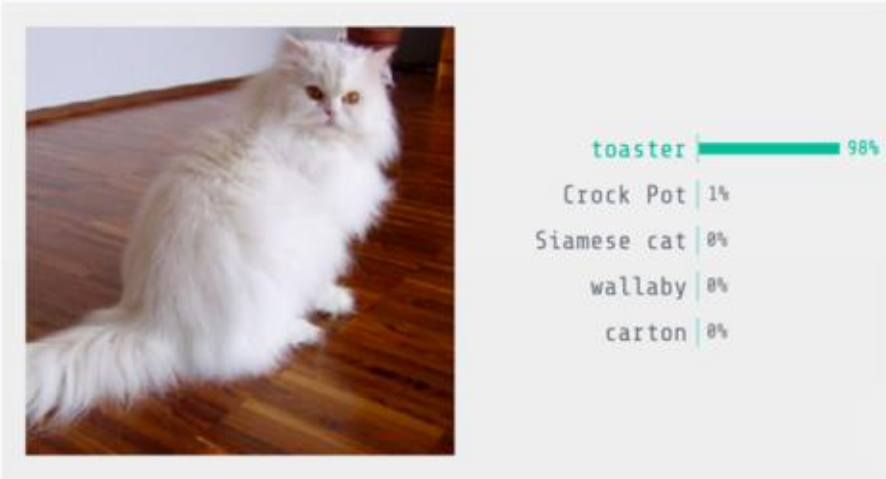


智能算法/应用引发的安全问题

Original Image



Hacked Image



- 智能算法/应用的准确性堪忧
  - ✓ 特斯拉的白色致命BUG
- 智能算法/应用容易被欺骗

- 应用的错误不应该引起系统的故障

来自汽车工程师的问题

老黄，我们的开发准则对于临界区资源要做保护，但是我们的工程师无法鉴别临界区资源，用了大量的锁，系统性能很差

老黄，我们的工程师经常搞出死锁的问题，然后导致发动机熄火

老黄，为什么会发生内存泄漏，以前我们都是一开始定义好的

POSIX PSE51

C++11

AUTOSAR Adaptive Platform OS API

- 汽车工程师应该专注于汽车系统
  - ✓ 很难了解过多计算机细节
  - ✓ 现有OS API的抽象程度对汽车工程师不友好
  - ✓ 缺乏在多核以及设计更多任务的经验

抽象程度更高的API/中间件？
自动化程度更高的开发组件？

- 计算机的问题应该交由操作系统和上层软件

以汽车安全角度重新审视汽车操作系统

1、当前汽车操作系统能够维护现代汽车安全

2、当前汽车操作系统无法维护未来汽车安全

- 多核系统竞争
- 多应用内存访问竞争
- 异构系统管理
- 算法或应用安全
- 静态设计产生的信息安全问题
- 开发者能力

以汽车安全角度重新审视汽车操作系统

1、当前汽车操作系统能够维护现代汽车安全

2、当前汽车操作系统无法维护未来汽车安全

3、能够维护未来汽车安全操作系统发展趋势

- 解决多核异构带来任务运行的不确定性问题（性能隔离？管理？）
- 安全的动态资源管理/调度方法
- 解决更高级的资源管理、任务抽象的实现
- 解决联网带来的诸多遗留问题和安全隐患
- ……

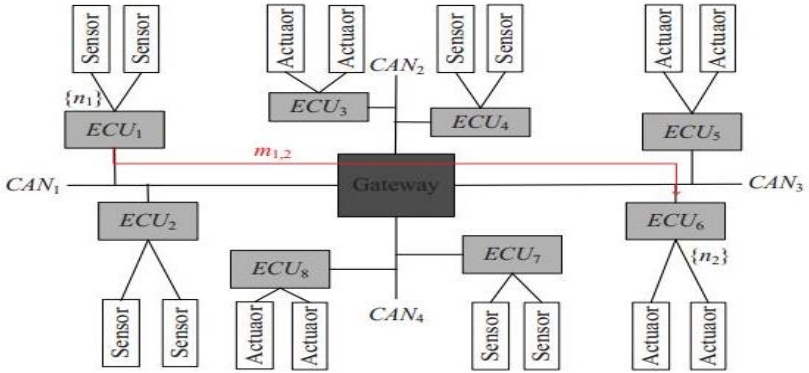
交互与智能固然重要，但是安全才是首要问题

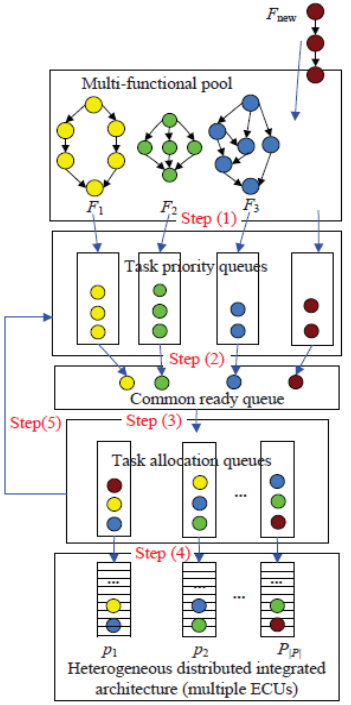
我们实验室的一些相关工作

- 参研TOPPERS/ATK2操作系统
  - ✓ 日本名古屋大学 Hiroaki Takada 教授主持研发
  - ✓ Autosar OS/Real-time OS

TOPPERS
Toyohashi OPen Platform for Embedded Real-time Systems

- 以功能安全为约束的静态、动态任务调度
  - ✓ 功能可以建模为DAG，Task为DAG节点
  - ✓ 运行时中央网关计算任务执行的ECU





Thank you!

For more info please contact us:

Phone: +86 151-9729-0063

Email: huangyizhi@hnu.edu.cn