理想

LiAuto HaloOS

理想星环 OS

技术架构白皮书 v1.0

目录

理想星环 OS 技术架构白皮书

1. 概述

1.1 什么是汽车操作系统

在 AI 浪潮引发技术革命、深刻重塑汽车产业的时代背景下，汽车操作系统正进化为承载和驱动车辆 AI 智能化的重要技术引擎。作为一个融合了实时性、安全性与可靠性要求的高度复杂的软件基座，它必须有效支撑各类智能应用的部署与运行。其根本任务是在车辆庞杂的硬件资源（包括高算力芯片）与多样的上层智能应用软件之间，扮演关键的“中央指挥官”和“资源总调度”角色，为实现车辆的各种功能提供基础环境和运行保障。

它需要精妙、高效地“向下”抽象和管理底层硬件。这包括但不限于：统一驱动和管理着异构的高性能计算芯片、各类存储单元、多样化的感知传感器阵列、车内通信网络，以及关系到行车安全与操控的执行器系统。这要求操作系统需要提供确定性的实时控制能力、高效的资源虚拟化与隔离机制，将这些物理硬件有机整合，确保其协同、稳定、安全地运作。

它需要安全、灵活地“向上”支撑不断演进的应用生态。 它需要为上层应用软件——从保障基础行车安全和驾乘舒适的智能车辆控制，到守护行车安全、提升通行效

率的智驾系统——开辟一片安全隔离、运行可靠、接口标准化、易于开发与部署的基础系统环境。这要求操作系统具备良好的可扩展性、兼容性以及强大的中间件服务能力。

因此，汽车操作系统是承载车辆网联化、AI 智能化转型的基础底座。 它的架构设计、技术先进性与安全性水平，不仅直接决定了整车电子电器架构的形态与成本，还深刻影响着功能的丰富度、性能表现和用户体验的优劣。

1.2 汽车操作系统发展趋势

汽车操作系统的演进历程与汽车硬件架构的变革息息相关，它们共同塑造了现代汽车的技术形态。回顾其发展，大致可划分为以下几个关键阶段：

【第一阶段：机械时代（~1970 年代之前）】

在这个时代，汽车主要由机械部件驱动和控制，几乎不涉及电子控制单元或软件。车辆的功能实现完全依赖于物理连接和机械联动，操作系统概念无从谈起。

【第二阶段：电子电器化时代（1970 年代 - 2010 年代末/2020 年代初）】

随着半导体技术的发展和汽车对功能性、安全性、舒适性要求的提升，电子电器技术开始应用于汽车。该阶段之内可进一步划分：

- **功能叠加阶段（1970 年代 - 2000 年代初）：** 以单功能控制器（ECU）的出现为标志。例如，电子燃油喷射、防抱死制动系统（ABS）、安全气囊等开始装车。这些早期 ECU 通常运行着裸机程序或者极其精简的、厂商专有的实时操作系统。
- **域集中/智能化阶段（2000 年代初 - 2022）：** 随着车功能持续爆炸式增长，ECU 数量急剧膨胀，这便推动了 EE 架构的域集中化，各个域内部的操作系统独立发展：
  - **车辆控制域（如动力、底盘、车身）：** AUTOSAR CP（Classic Platform）

逐步成为主流框架，提供规范化的接口与组件，将车控软件的开发流程标准化。

- **智能驾驶域：**负责环境感知、决策规划与执行控制，特点是多传感器融合所带来的数据密集与 AI 算法所带来的计算密集。因此操作系统需要支持高性能的异构计算，常采用 QNX、Linux 与 AUTOSAR AP（Adaptive Platform）结合。

【第三阶段：空间机器人时代（2022~）】

随着人工智能，特别是大语言模型等生成式 AI 技术的快速发展并向汽车领域渗透，车上的算力需求和软件复杂性指数级增长。因此硬件层面正朝着中央计算架构演进，软件层面正朝着整车各个域应用 AI 化、一体化发展。该变化正驱动着汽车操作系统的革命：

- **从 Smart 到 Intelligence 的跨越：**随着先进 AI 技术在汽车领域的深度融合，汽车操作系统正经历从“Smart”到“Intelligence”的跨越。这一转变要求操作系统需具备“Native for AI”的特性——能够智能地调度和管理 CPU、GPGPU、NPU 等异构计算资源，确保高性能、确定性与安全性；通过隔离机制保障不同安全等级、不同实时性要求的功能可靠共存；提供标准化的 API 和工具链，支撑上层 AI 应用的敏捷开发与快速迭代，让 AI 成为驱动汽车不断进化的内生动力。
- **面向“通用空间智能操作系统”的进化：**智能驾驶功能需要汽车在物理世界中安全、可靠、实时地运行，这使其成为空间机器人理念的最早、最大规模的落地场景之一。能够满足这种极致复杂场景的操作系统，其技术栈具备超越智能汽车本身的通用性。智能汽车操作系统有潜力成为“通用空间智能操作系统”，以面向未来更广泛的机器人、具身智能体等需要与物理世界深度交互的复杂 AI 系统。

2. 星环 OS 技术架构

星环 OS 是一款面向 AI 智能化业务的整车操作系统，以全域协同、软硬结合为创新内核，树立性能、安全、成本及效率的行业标杆，奠定空间机器人时代的系统基石。

- 资源集中与共享：算力池化、通信以太网化、全域调度、服务共享
- 极致性能：混合系统中关键链路的端到端实时性、确定性、可靠性

- 快速迭代：软硬解耦、软软解耦、服务化设计、定制化工具
- 高安全性：原生安全的体系化纵深安全防护，保护隐私数据和控车指令

星环 OS 由以下四个重要部分组成，可以比作人的大脑、小脑、神经系统和免疫系统。

- **智能驾驶系统**定位是车辆的**大脑控制系统**，它可以处理复杂思维过程，保证智能驾驶又快又好的工作。设计的重点在于**“智慧”**：
  - 通过任务调度、图像处理等优化措施，让 AI“思维敏捷”，跑得更快更稳。
  - 通过虚拟化技术实现算力共享，让一块芯片安全地同时处理多个任务，实现“一心多用”。
- **智能车控系统**定位是车辆**小脑控制系统，**它负责车辆的肢体控制,快速执行车辆的各种基础控制命令（比如开关车灯、调节空调、控制电机等）。设计的重点在于**"灵活"**：
  - 通过灵活架构适配，支持不同品牌、不同架构的芯片，实现 MCU 选择自由，缩短新 MCU 上车时间。
  - 关键性能指标与资源优化指标均做到了业界领先，让多个功能共享同一块硬件成为可能，支撑灵活降本。
  - 系统配套有自动配置工具、PC 端硬件模拟器工具以及覆盖资源、性能、诊断的一整套分析工具，实现研发效率跃升。

- **通信中间件**定位是车辆的**神经系统**，负责在域内与域间传递信息，完成车内各个模块（比如刹车、屏幕、雷达）之间的高效、可靠通信。设计重点在于**“通畅”**：
  - 效率上，统一了车内大大小小多种设备的应用层通信协议，实现“车同轨、书同文”，做到了各域通信解耦，提升效率。
  - 可靠性上，实现了关键特性、核心指标与资源优化均实现业界领先，结合双通道备份等技术大幅提升通信可靠性。
- **信息安全系统**定位是智能汽车的**免疫系统**，用来保护用户隐私数据和车控指令。通过软硬件协同实现，构建系统全面的安全体系，达成四个安全目标：系统完整性保护、数据加密与保护、身份认证与权限管控和可信执行环境，实现了原生安全的纵深防御体系。

星环 OS 的四大支柱并非独立模块，而是构成一个深度融合、协同运作的有机整体 —— 恰如人体的不同系统协调一致，共同支撑高级智能行为。以通信中间件这条“高速神经系统”为脉络，星环 OS 确保了智驾系统的复杂决策能够高效、可靠地传递给车控系统精准执行，同时保证了各个传感器与处理单元间的数据畅通无阻。

基于这种一体化架构，星环 OS 能够提供超越单个模块能力的系统级保障。例如，在确定性方面，通过全局任务调度、优先级管理以及确定性通信机制，星环 OS 能够保障从传感器感知、智驾“大脑”决策，到车控“小脑”最终执行的端到端任务链路具有可预测的稳定低时延，即使在多任务高并发的复杂工况下依然表现稳定可靠，这对于智能驾驶的安全性和平顺性至关重要。

同样地，在安全性方面，信息安全“免疫系统”并非孤立运行，而是贯穿于整个操作系统。统一的安全架构将身份认证、权限管控、完整性保护和数据加密等措施系统性地应用于所有组件、接口和通信链路中，结合可信执行环境和硬件安全模块（HSM）的底层支持，实现了从硬件可信根到上层应用，从系统启动到运行时的全方位、多层次纵深防御，有效抵御跨模块、跨系统的攻击渗透。

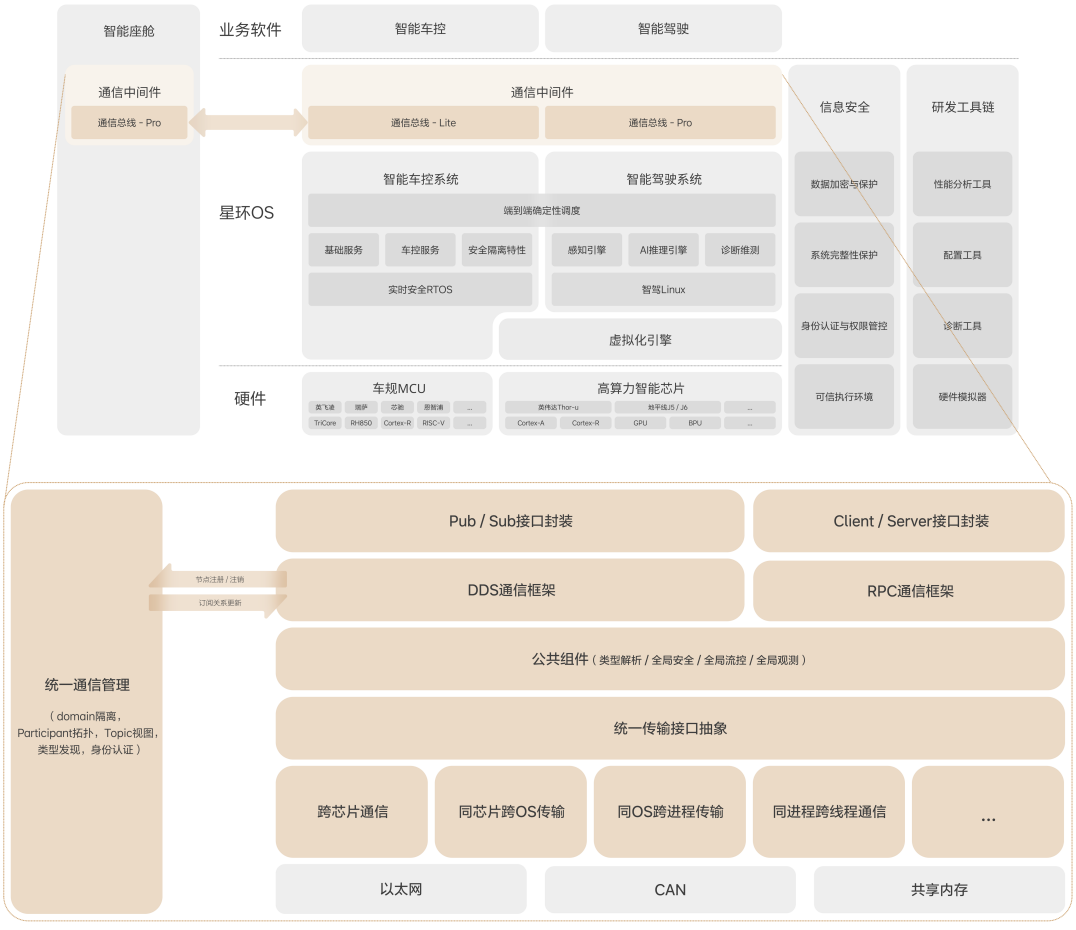
正是这种高度协同的有机整体性，使得星环 OS 不仅仅是功能的集合，更是一个能够全局优化、统一管理、系统性安全保障的强大平台，为智能汽车提供兼具高性能、高安全、高效率的核心支撑，并逐步构建未来空间机器人的坚实系统基座。

3. 核心系统介绍

3.1 通信中间件

3.1.1 系统说明

星环 OS 通信总线（VBS，Vehicle Bus System）是专为智能汽车领域打造的高效数据交互通信平台。该平台依托标准化的通信协议、模块化架构以及卓越的实时数据传输能力，为整车电子电气系统构建起一条实时且可靠的信息高速公路。凭借这一平台，智能驾驶、动力控制、信息娱乐、主动安全等关键服务得以实现无缝协同，为智能汽车的高效运行与功能拓展提供坚实保障。通信中间件的系统架构如下图所示：

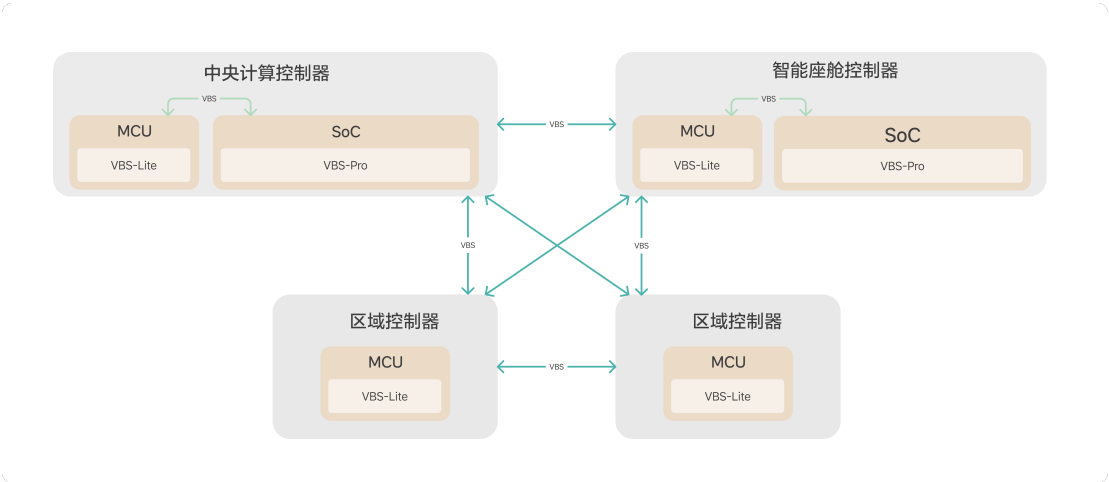


3.1.2 核心技术特性

3.1.2.1 支持全域统一部署

新一代电子电气架构给车载通信中间件带来两大技术难题：一方面，智能驾驶系统要求海量传感器数据能在毫秒级内确定性传输；另一方面，车控域芯片算力与存储空间有限，却需部署管理数百个通信主题（topic）。当前行业内的通信中间件仅能在限定域场景下解决部分问题，导致车载场景全域系统部署时通信协议割裂，工程化管理与维护成本很高。星环 OS 通信中间件基于车载场景实现定制化的 DDS 通信协议，实现面向 MCU 的轻量化设计，构建起真正全域统一的通信基座，核心特点如下：

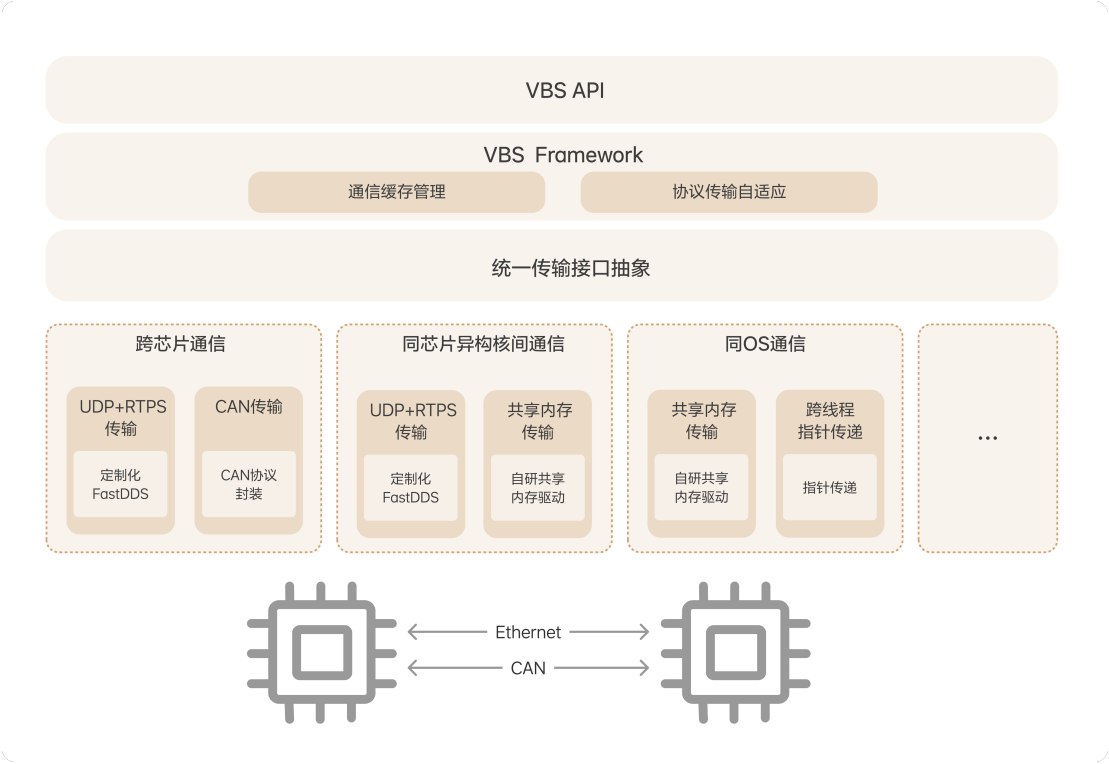
- **VBS Pro 版本：**运用无锁化设计与自适应序列化 / 反序列化等技术，实现跨进程零拷贝数据传输，提升数据传输效率；通过发送端消息过滤与定频消息去重等机制，有效减少无效数据传输。
- **VBS Lite 版本：**借助自定义通信协议、传输通道智能合并、逻辑通信端点等技术创新，降低对系统内存占用，满足了各类资源受限 MCU 部署场景需求。
- **通用特性：**VBS Pro 版本与 VBS Lite 版本均采用统一的跨域数据传输协议，这使得全域通信无需进行复杂的多协议间交互，简化了通信流程，提升了系统整体的兼容性与易用性。



3.1.2.2 多传输协议自适应

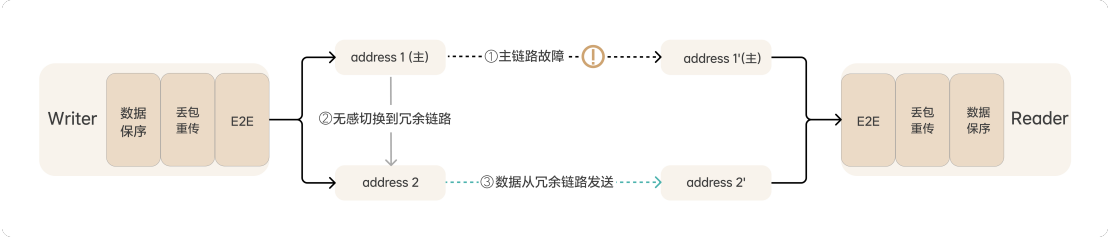
在车载汽车领域，业务功能部署存在无序性，而且底层介质协议呈现多样化，涉及以太网、CAN、共享内存等多种介质。在传统方案中，针对每种传输介质都需要定制独立的协议栈，应用程序也必须分别适配不同的协议栈，这无疑大幅增加了开发的复杂性与成本。为了解决上述难题，星环 OS 通信中间件设计实现了多传输协议自适

应方案，支撑业务使用统一接口层，底层可在跨芯片、芯片内异构核之间、核内多进程之间等不同场景下，自适应匹配到底层以太网、CAN、共享内存等传输介质上，从而有效简化开发流程，具体如下图所示：



3.1.2.3 可靠性机制增强

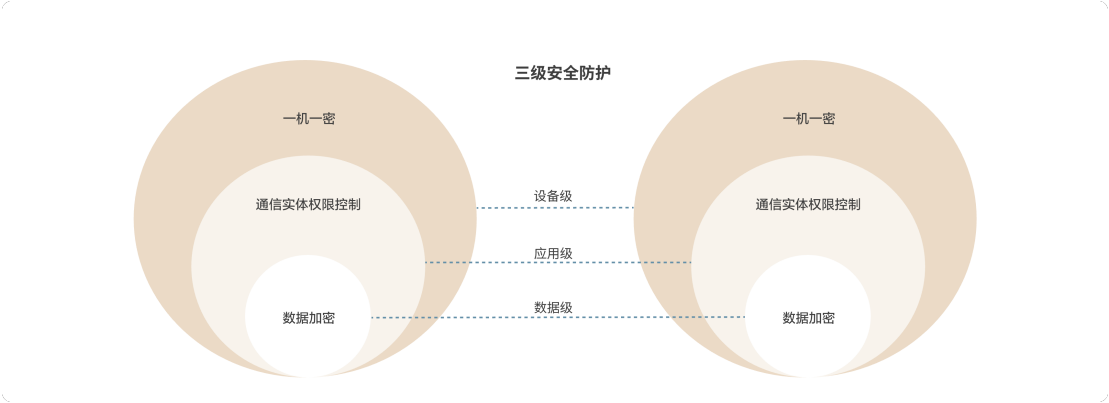
星环 OS 通信中间件不仅支持 E2E 校验、丢包重传、按序到达以及网络拥塞控制等基础传输可靠性保障机制，还实现了多路冗余传输方案、共享内存异常无感恢复等可靠性增强方案，确保关键指令（如主动安全相关指令）能够可靠到达，同时实现传输低延迟，以适应严苛的车规级环境。其中多路冗余传输方案的原理如下图所示：



3.1.2.4 多层级安全防护

星环 OS 通信中间件基于车载场景进行安全防护增强，实现三级安全防护，如下所示：

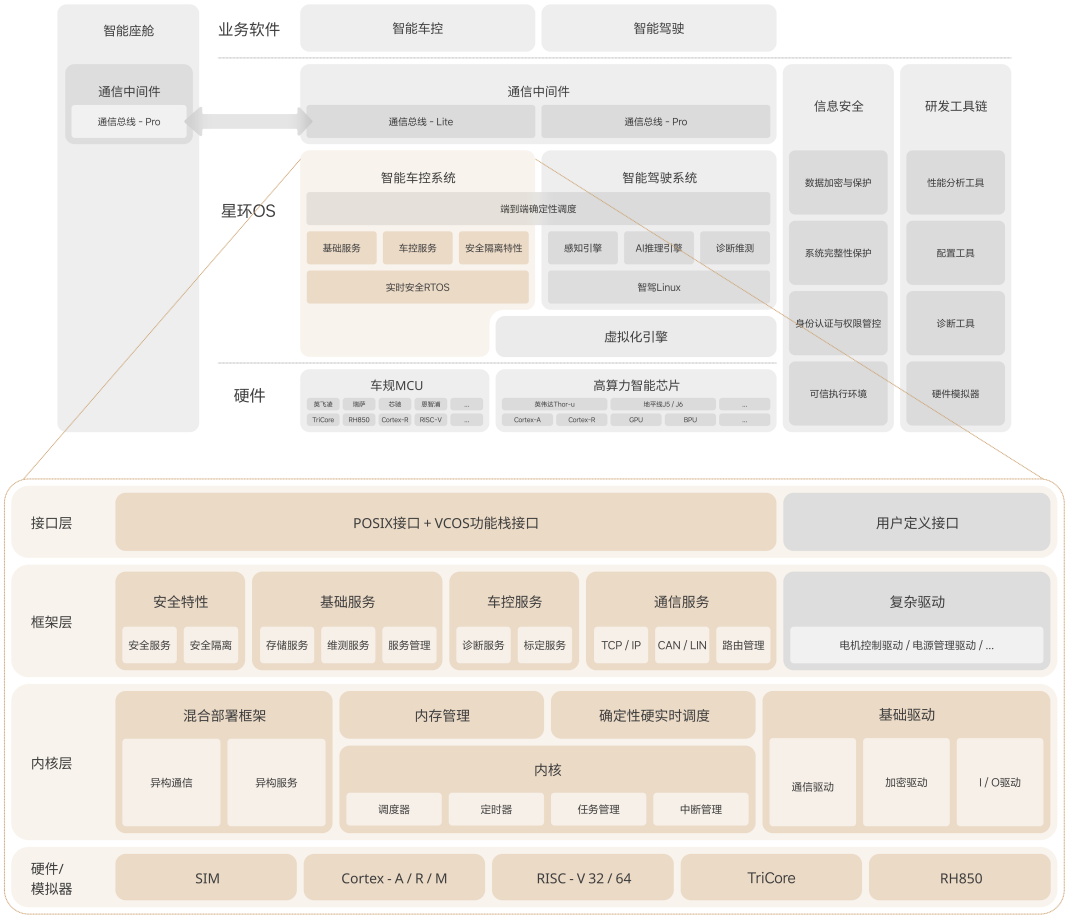
- **设备级：**采用一机一密 PKI 身份认证机制，确保非法设备无法探测到授权设备所提供的服务，从源头上阻止非授权设备接入网络，保障设备层面的安全。
- **应用级：**通过对通信实体应用进行权限控制，只有经过签名的可信应用之间才能建立通信，有效防止非可信应用干扰或窃取通信数据，保障通信过程的安全性与可靠性。
- **数据级：**运用会话级数据加密技术，即使报文被中间人截获，由于缺乏有效的解密密钥，也无法获取原文内容，全方位保障数据在传输过程中的保密性。



3.2 智能车控 OS

3.2.1 系统说明

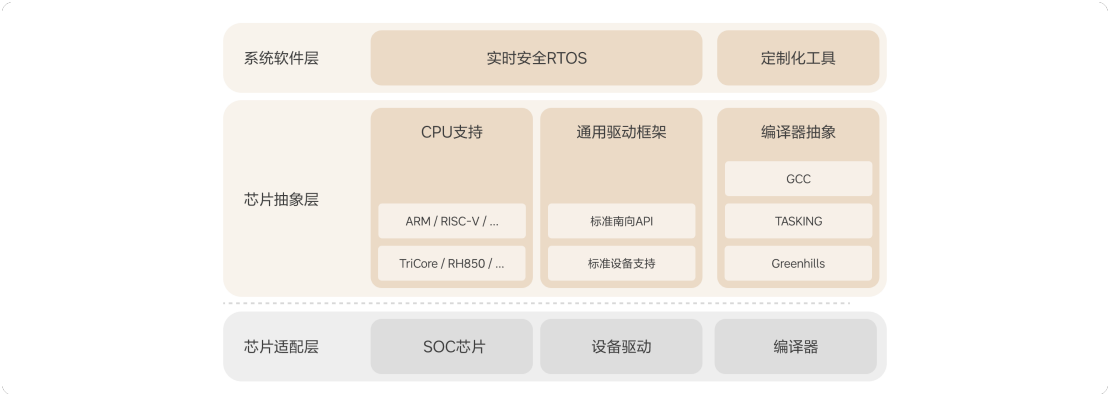
智能车控 OS（VCOS: Vehicle Control Operating System）是面向车辆控制的操作系统，支撑智能汽车高安全、高实时的核心车控业务部署。系统通过硬实时调度架构确保动力控制、底盘控制等关键任务精准响应，实现感知与决策系统的高效协同，构建从硬件到软件的全链路安全防护体系。智能车控 OS 在实时性、确定性和功能安全等维度显著领先业内系统，并配备覆盖开发调试、仿真验证的可视化工具链，有效提升车企在智能控制系统开发、测试及迭代的效率。智能车控 OS 的系统架构如下图所示：



3.2.2 核心技术特性

3.2.2.1 全方位软硬解耦

智能车控系统硬件选型的复杂性曾是制约快速迭代的瓶颈，为应对这一挑战，智能车控 OS 采用了彻底的软硬件解耦策略。内部设计实现了一个逻辑清晰、交互间接的芯片抽象层，通过多维度抽象建模（覆盖 CPU、驱动、编译等），有效屏蔽了底层硬件差异，为上层系统提供了一致且稳定的视图。这种“隔离”设计使得适配新硬件时，把原本需要的绝大部分修改提取到抽象层内部，并借助自动化代码生成工具，适配工作量被大幅削减。这使得车企能够将过去长达数月的芯片适配周期缩短至 4 周，显著提升了芯片选择灵活性，为供应链的韧性提供了坚实保障，架构原理图如下：



3.2.2.2 寻优算法保障硬实时

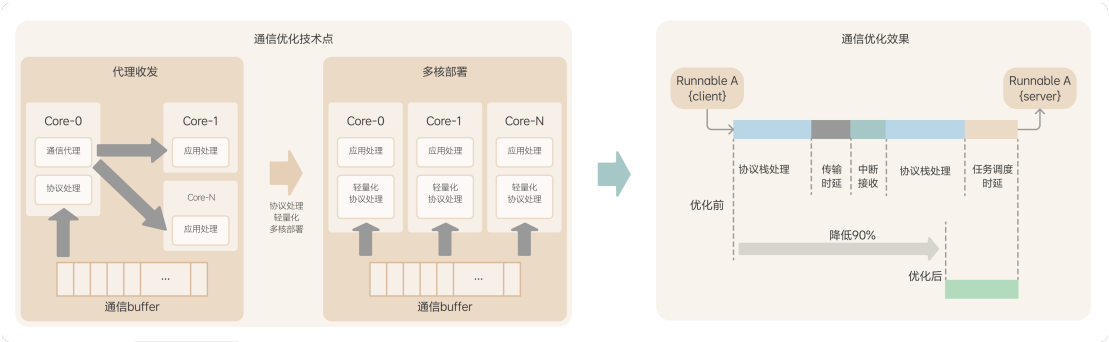
在复杂的跨域分布式场景中，会出现各种端到端实时性不达标的状况。智能车控 OS 在传统硬实时内核上进行了进一步强化，并借助一体化工具链和全局寻优算法支撑，进一步提升跨系统交互的端到端实时性，其核心能力如下：

- **硬实时内核：** 实现了微秒级中断处理和任务切换，保证极低且可预测的中断延迟与切换开销；基于抢占式优先级调度策略，最大限度减少任务阻塞与不确定性。
- **一体化工具链：**提供端到端时序分析与验证功能，协助开发者在设计阶段分析复杂系统的实时性表现；基于全局寻优算法生成最优系统配置，并确保系统可以按照优化后系统配置执行，将“事后调试”转化为“设计阶段保障”。

3.2.2.3 全链路压缩通信时延

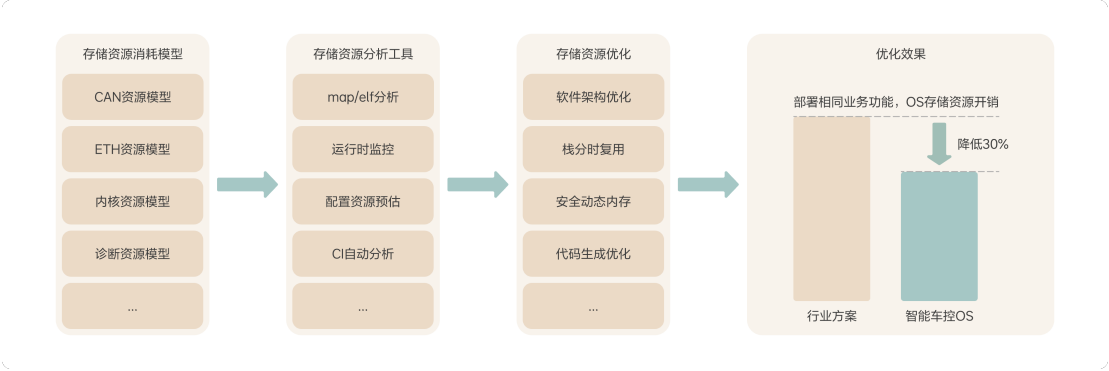
完成一次远端服务访问过程中，协议传输层和系统任务调度层的各个环节处理都会影响到最终用户接口的访问时延。因此智能车控 OS 内部基于自研确定性网络、轻

量化协议栈、协议栈多核部署等关键技术，大幅缩短了远端访问时延。在跨域控制器传感器资源共享的部分典型场景下，端到端访问时延减少 90%，实现与本地设备基本一致的访问效果，优化效果如下：



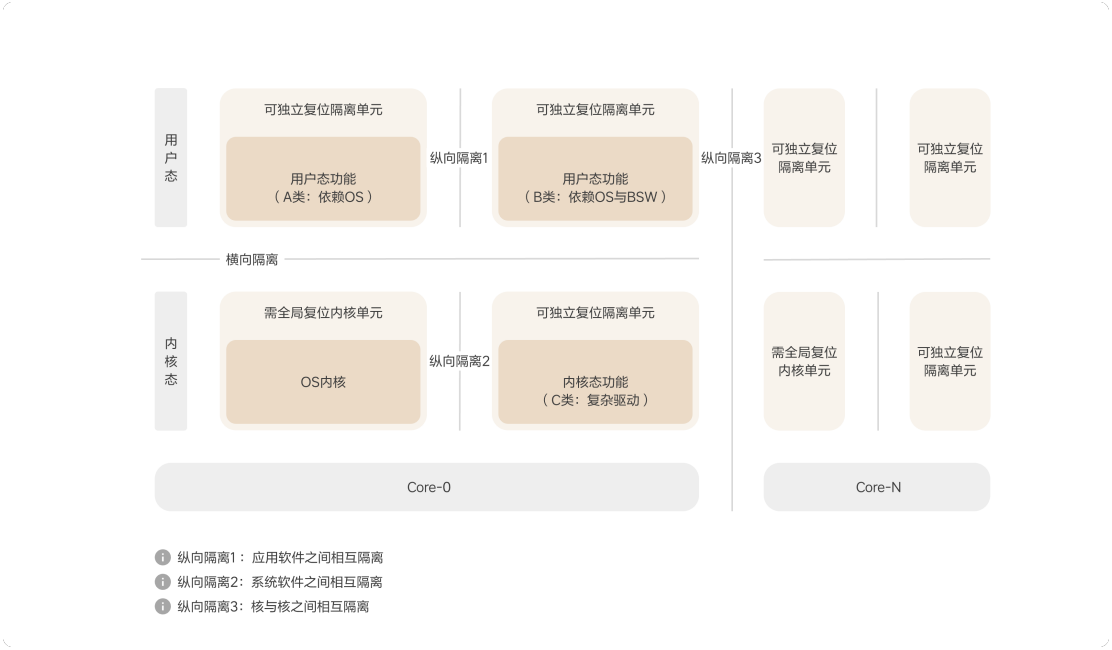
3.2.2.4 多维度存储资源优化

传统 OS 通过一些零散功能点做存储资源优化，没有形成全流程闭环的优化逻辑，难以持续迭代优化。智能车控 OS 通过建立资源消耗模型，借助智能车控 OS 的资源分析工具，并结合资源池化等诸多优化机制，形成全流程可持续的优化方案，将 OS 的资源开销相比行业领先方案优化 30%，具体方案与效果如下图所示：



3.2.2.5 轻量级安全隔离

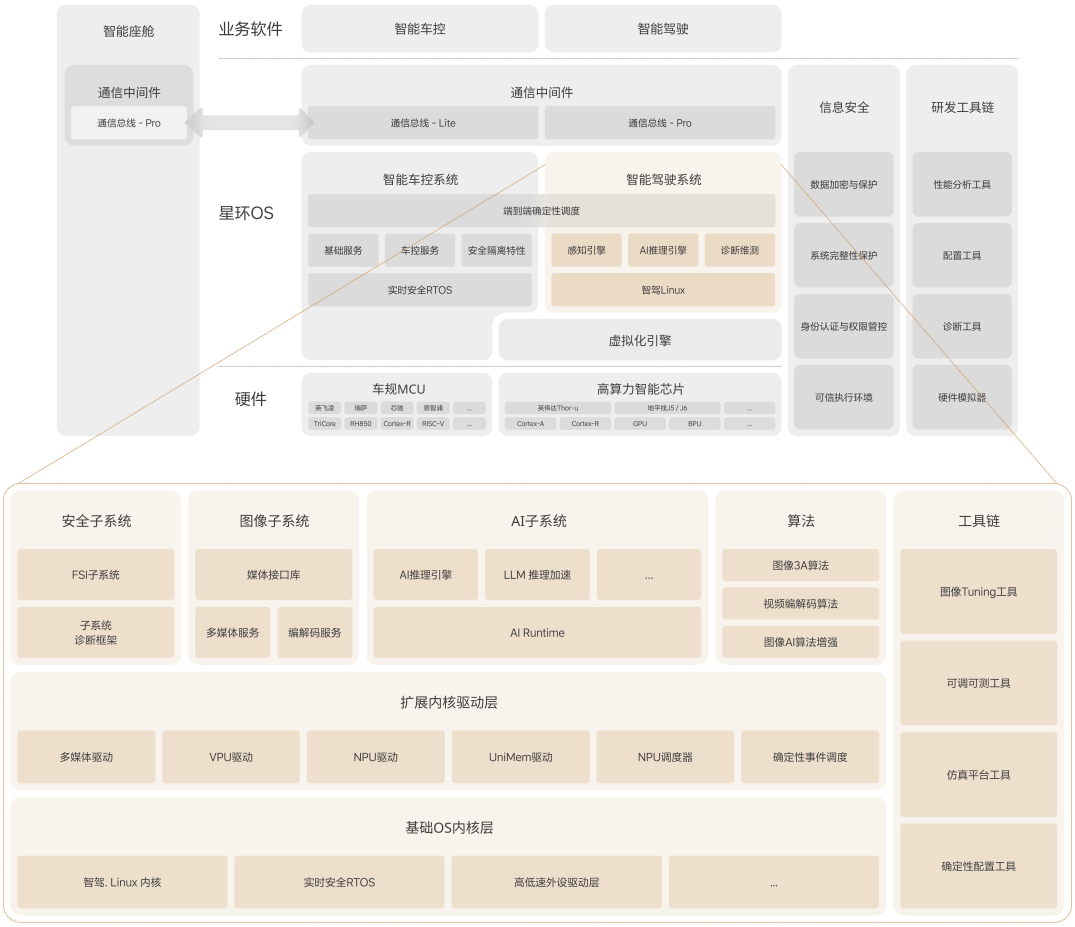
传统操作系统常采用内核级纵向隔离，不仅资源开销大，且高度依赖特定硬件功能，例如复杂的 MMU 配置等。智能车控 OS 创新地在解耦底层硬件特殊功能的依赖基础上，构建了一套轻量级软件解耦框架。该框架实现了核与核、系统软件间、以及应用层级间的三种纵向隔离机制，充分满足车载业务在功能隔离与独立复位方面的核心需求，通过最大化的轻量化设计，实现隔离安全性与资源效率间的最佳平衡，具体原理和效果如下图所示：



3.3 智能驾驶 OS

3.3.1 系统说明

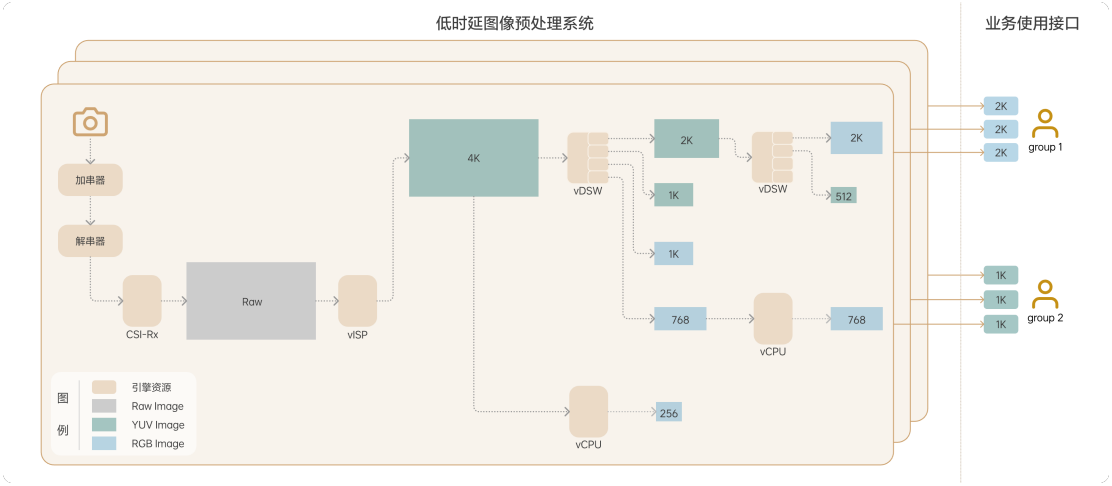
智能驾驶 OS 是为智能驾驶场景打造的专用操作系统。系统内部对底层图像处理、AI 推理加速、视频编解码等专用硬件能力进行高效封装，在最大化释放底层硬件性能的同时，向上层应用提供一套简洁易用的接口，助力上层系统聚焦于业务与算法的实现，快速实现产品的迭代演进。同时，智能驾驶 OS 系统在异构图调度确定性、软硬协同的故障诊断与恢复机制等方面采用了诸多创新性设计，为智能驾驶场景提供了卓越的实时性、确定性和安全性保障。智能驾驶 OS 还同步提供了配套开发的仿真、调试和可观测性等完整工具链，显著提升开发者在开发、测试及维护阶段的工作效率。智能驾驶 OS 的系统架构如下图所示：



3.3.2 核心技术特性

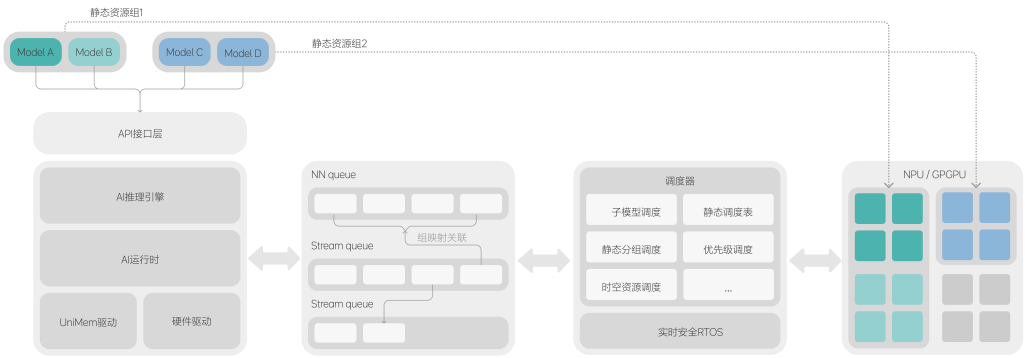
3.3.2.1 低时延图像预处理

图像预处理为智能驾驶端侧系统提供核心输入信息，其重要性不言而喻。由于预处理产生的图像质量、时延及稳定性直接决定了整个智驾系统的性能表现，因此该处理系统构成了智驾 OS 内部的关键功能。为了确保最优效果，智能驾驶 OS 的图像预处理子系统会基于用户配置的多路图像组，动态地规划图像传感器到专用处理硬件（IP）的计算流水线，从而保障最低的端到端处理时延。其工作原理如下图所示：



3.3.2.2 基于车端优化 AI 推理

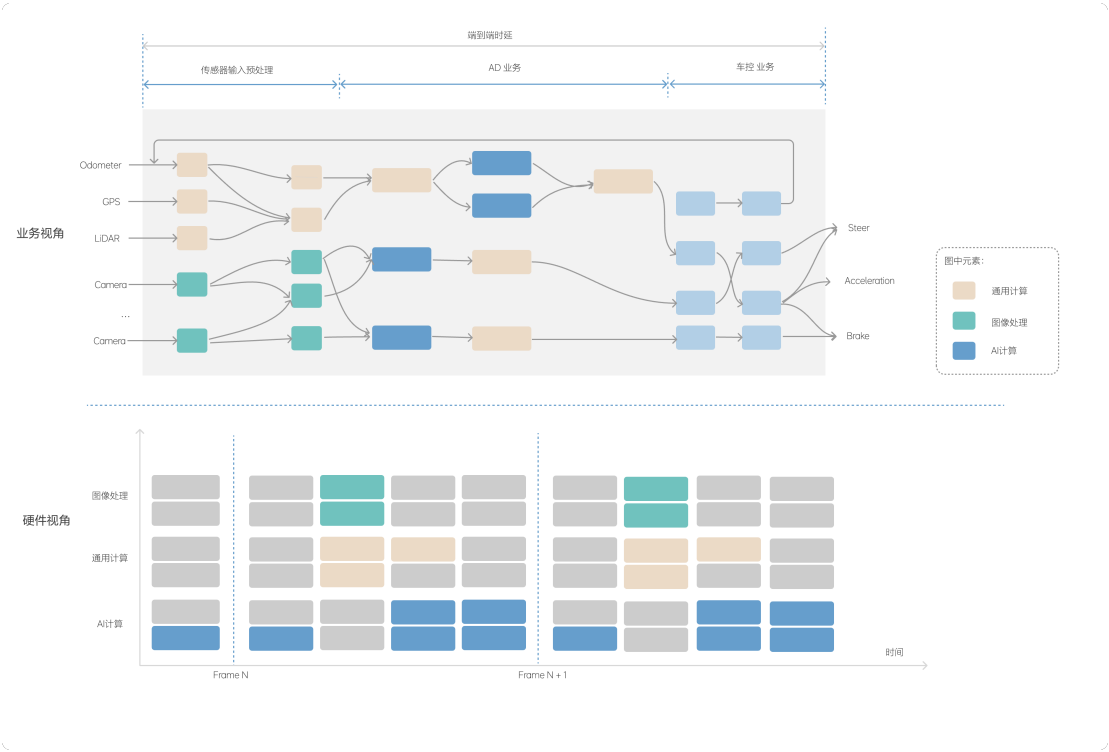
智能汽车中 AI 模型推理的应用日益增多，遍及车控、智驾等多个领域，导致模型数量不断增长。然而，不同领域业务对 AI 推理在资源占用、实时性、确定性和安全性方面的要求差异巨大，这使得 AI 算力推理加速变得异常复杂。为了应对这些挑战，智能驾驶 OS 开发的 AI 推理子系统通过多层级 AI 计算资源预编排以及多种内置调度模式（例如：子模型调度、优先级调度、时空资源调度等）相结合的方式，能够根据具体场景灵活选择策略，有效满足端侧 AI 推理的需求。其架构原理如下图所示：



3.3.2.3 端到端确定性调度能力

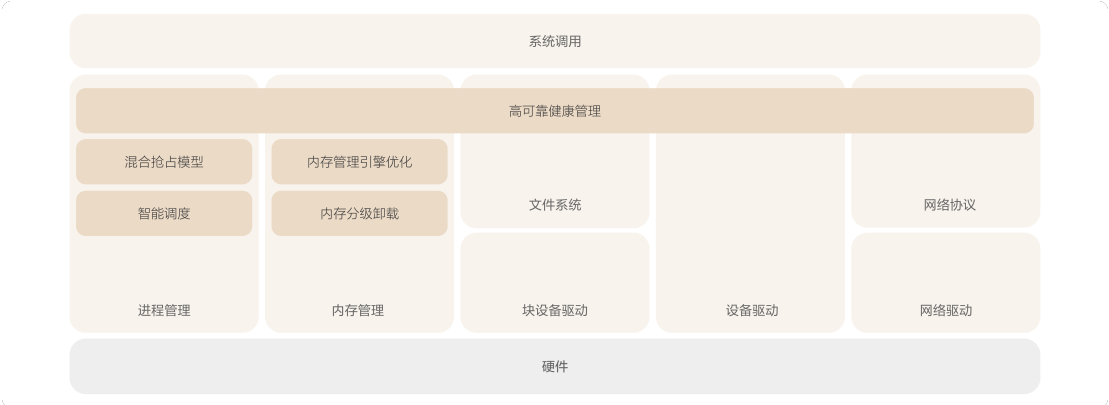
作为智能驾驶系统的关键底座之一，智驾 OS 致力于实现两大核心目标：极致的性能表现与关键链路执行的严格确定性。后者不仅体现在极低的时延抖动，更要求执行过程确定且可回放，这是最大限度保障行车安全与生命安全的基础。为实现这一确定性，智驾 OS 会构建覆盖从传感器输入到控制器输出全过程的异构计算任务图，并将任务精确映射、分配至底层硬件单元，有效管理资源竞争，最终确保端到端调度的

可预测性。核心原理的简化架构如下：



3.3.2.4 定制化的智驾 Linux 内核

在智能驾驶领域，定制化 Linux 内核具有关键作用。由于智能驾驶系统对实时性、可靠性、安全性及特定硬件支持的要求极高，标准通用 Linux 内核通常难以满足这些需求，因此需要进行深度定制和功能增强，以构建适用于智能驾驶场景的系统内核。



定制化智驾 Linux 内核主要采用以下核心技术方案：

- **实时性和性能提升：**采用定制化内核混合抢占模型，提升智能驾驶场景的调度实时性；支持可编程调度算法扩展，满足车载多样化业务需求，全方位优化性能表现。

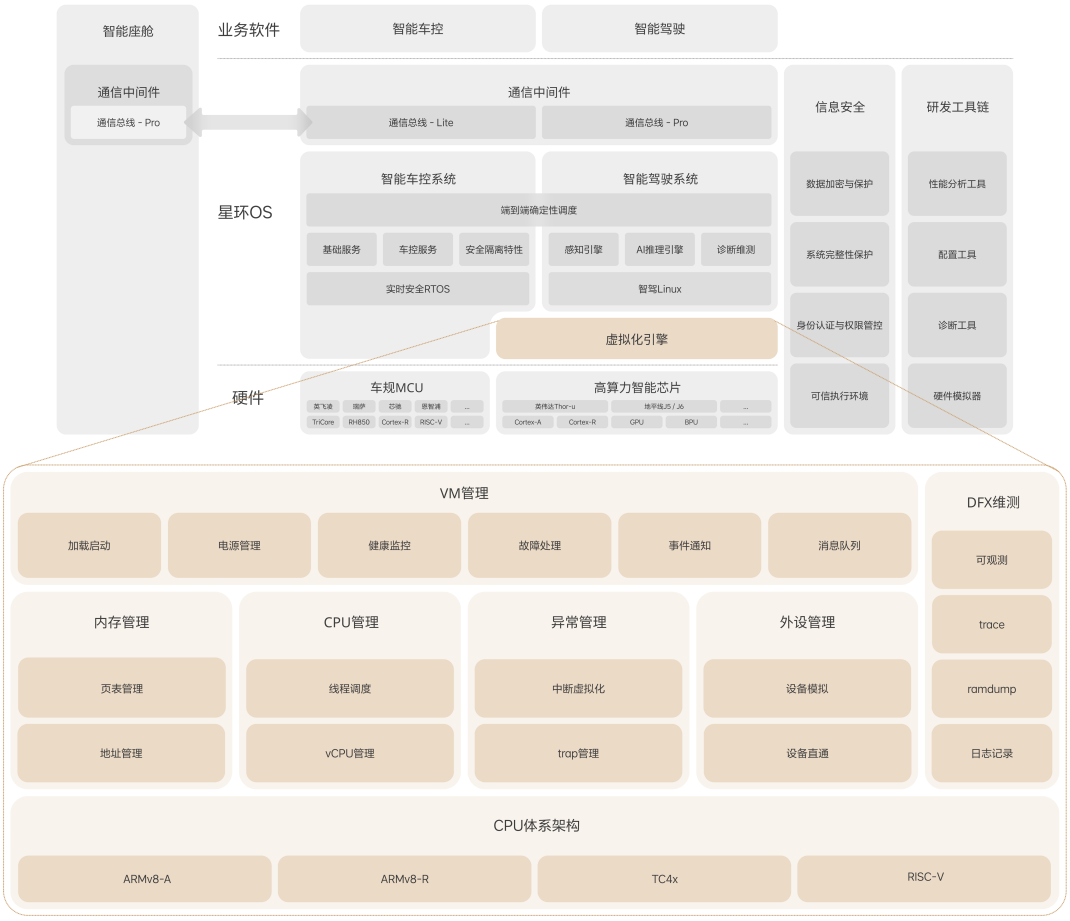
- **极致资源利用与管控：** 采用用户无感知的智能内存分级卸载器有效减少内存占用，配合动态大页技术和优化的页面回收算法，显著提升内存管理效率。

- **高可靠的健康管理：**采用实时高效的健康诊断框架，为智能驾驶业务提供实时功能安全(FHTI)保障，包括软硬件故障的精确定位能力、故障严重等级和功能状态诊断评估以及保障系统可用性的多级故障处理机制。

3.4 虚拟化引擎

3.4.1 系统说明

星环 OS 虚拟化引擎（LiVisor）是用于构建车端 AI 计算中心的虚拟化底座，它通过对 CPU、内存、NPU 以及 I/O 外设进行资源池化管理与安全隔离，在集中式硬件平台上支持车端智能驾驶、智能车控等多域业务的并发运行与协同。在底层充分结合硬件特性进行软硬联合定制化设计，满足不同客户机对冷热启动、跨域通信、高速外设访问等特性的高性能需求。



3.4.2 核心技术特性

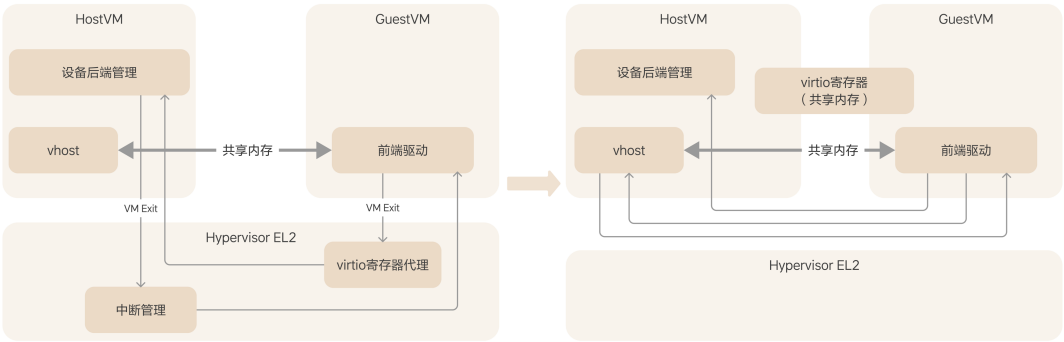
3.4.2.1 系统级安全隔离

传统虚拟化技术主要针对云端场景设计，而车载业务属于安全关键系统（Safety-Critical），对隔离性要求更为严格。为此，星环 OS 的虚拟化引擎 LiVisor 采用静态分区技术，确保 CPU、内存、中断及独占外设等资源具备更强的隔离性。

- **CPU 隔离，**以物理 CPU(PCPU)为粒度做隔离，静态部署逻辑 CPU(VCPU)到物理 CPU(PCPU)上，避免 CPU 资源争抢，保障实时性。
- **内存隔离，**建立全局静态可配内存资源池，基于 CPU 的 Stage 2 MMU 特性实现虚拟机内存空间硬隔离，消除内存越界访问风险。
- **中断隔离，**虚拟化系统中断控制器，按虚拟机粒度划分中断与路由策略，杜绝中断风暴跨虚拟机传播。
- **独占类外设隔离，**结合 CPU 的 Stage 2 MMU 特性及白名单机制对虚拟机独占设备启用直通模式，实现独占类外设 I/O 空间硬件级隔离。

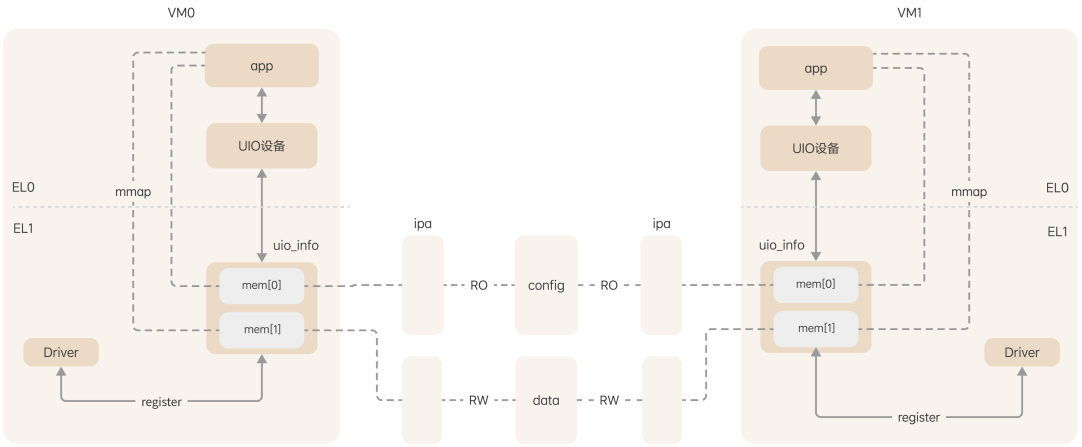
3.4.2.2 高效资源共享

针对车载域融合场景中 UFS、以太网控制器等非硬件虚拟化设备的共享需求，传统 virtio 半虚拟化方案会导致 30%~40%的 I/O 性能损耗。星环 OS 虚拟化引擎重构 vhost 控制面框架，推出基于 virtio 的增强技术 VM Exit-Less。通过消除虚拟机上下文切换（VM-Exit），显著减少数据传输时延，极大提升设备虚拟化吞吐性能。



3.4.2.3 高速跨域互通

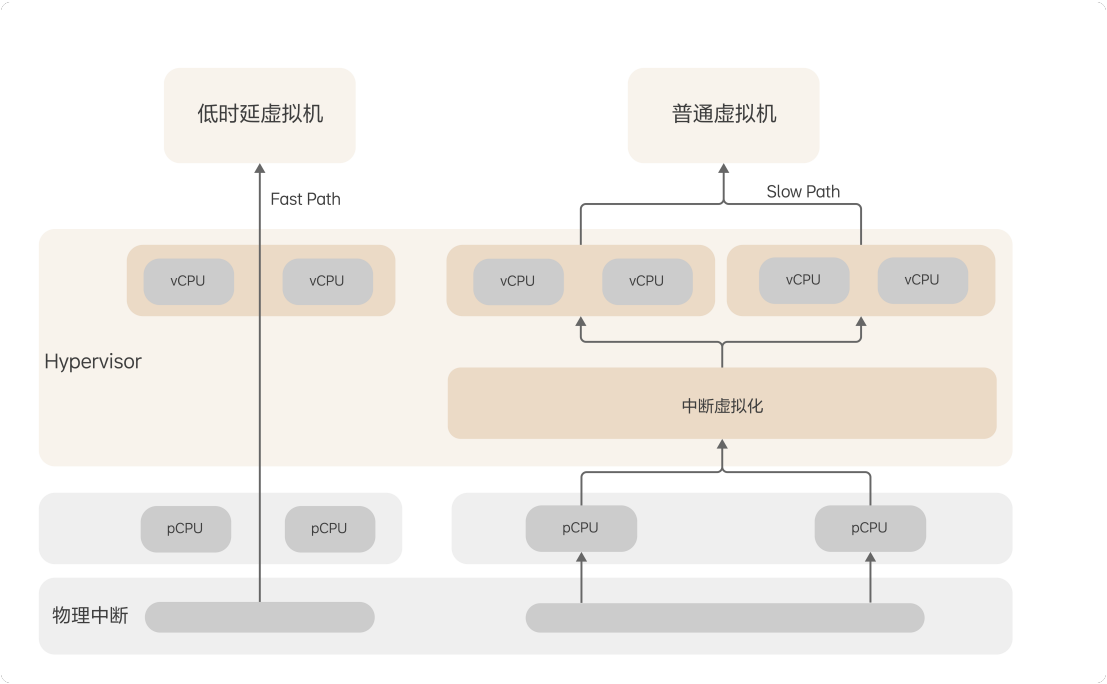
传统基于网络的虚拟化通信手段不仅通信时延较高，还会额外消耗网络带宽资源。星环 OS 虚拟化引擎采用了共享内存机制，实现了跨 VM 间的零拷贝通信，确保各类跨 VM 通信场景的实时性，其架构原理如下图所示：



3.4.2.4 性能损耗更低

传统虚拟化方案因高中断延迟、频繁缺页异常、TLB Miss 高及 vCPU 上下文切换开销大等痛点问题，导致虚拟机性能与实时性严重下降。星环 OS 虚拟化引擎通过以下核心技术构建完整实时虚拟化方案。

- **vCPU 绑定**，将 vCPU 和 pCPU 一对一静态绑定，减少 CPU 的频繁切换带来的上下文切换损耗。
- **大页预映射**，对虚拟机物理内存进行预映射，消除运行时缺页现象。利用内存大页，减少 Stage 2 内存页表、降低 TLB Miss，提升虚拟机访存性能；
- **中断直通**，控制面-数据面相结合，实现关键中断高效处理：
  - **控制面**：Guest OS 对 vGIC 的每次读写都会陷入到 Hypervisor 内的 vGIC 模块，保障安全。
  - **数据面**：设备中断直通目的 VM，无需经过 Hypervisor 绕行转发，保障性能。

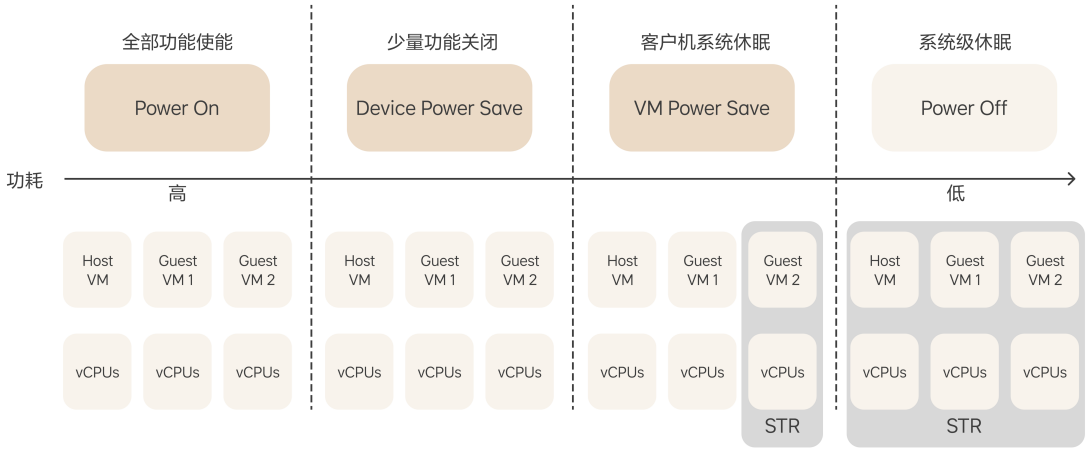


3.4.2.5 冷热启动加速

车载系统的启动速度与休眠唤醒效率，直接决定用户对“上车即用”体验的满意度。LiVisor 通过定制化 VM 级并行启动、细粒度 VM 级休眠唤醒及全域休眠唤醒技术，确保用户在上下车、临时离车等场景中始终感受无缝衔接的流畅交互。

LiVisor 支持三级渐进式低功耗策略，功耗控制粒度从外设级覆盖至系统级：

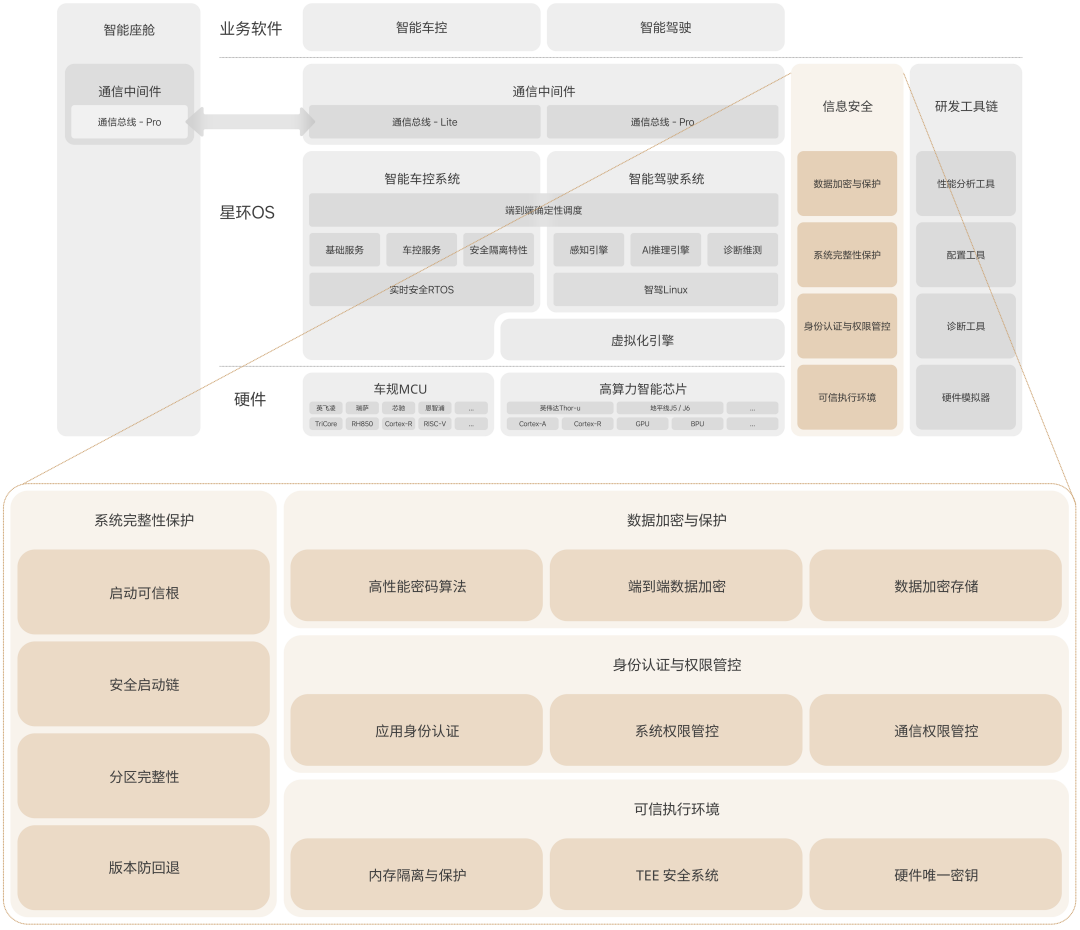
- 外设级休眠：客户虚拟机动态挂起（Runtime Suspend）非必要外设，降低局部功耗。
- 虚拟机级休眠：客户虚拟机按整体业务挂起休眠，包括该虚拟机使用的 CPU 以及独占外设。
- 系统级休眠：系统级挂起休眠，非 AON 电源域软硬件资源均进入低功耗状态。



3.5 信息安全

3.5.1 系统说明

星环操作系统的信息安全体系是面向智能网联汽车构建的一整套多层级安全防护机制，涵盖数据加密与保护、系统完整性保护、身份认证与权限管理和可信执行环境等关键能力，旨在保障车辆关键功能稳定运行和用户隐私不泄露。核心功能架构如下图所示：



3.5.2 核心技术特性

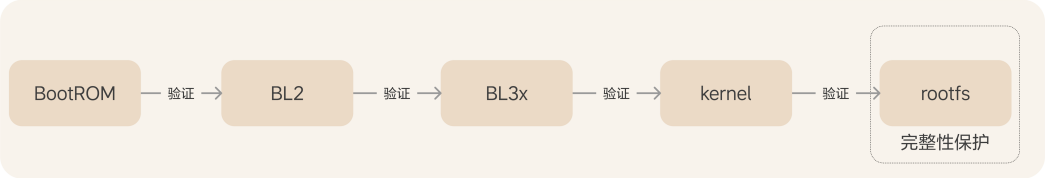
3.5.2.1 数据加密与保护

数据加密与保护功能旨在防止未经授权的访问与篡改，确保数据在存储和传输过程中的机密性、完整性与可用性，降低数据泄露风险。智能汽车涉及大量的用户隐私数据，星环 OS 实现了覆盖全场景的数据加密能力，以保护用户隐私数据。

数据加密的基础是密码学算法。星环 OS 实现了多种密码学算法，涵盖了各类对称加解密算法、非对称签名验签算法、密钥交换算法和哈希算法等，并且通过软硬结合实现提高了算法性能，相较于纯软件实现性能提高了 4 倍。基于这些高性能算法实现了端到端加密，确保数据传输过程中不泄露；同时实现了对应用透明的存储加密，在保护数据的同时降低应用接入成本。通过覆盖全场景的数据加密能力，星环 OS 实现了对关键数据的全程保护，最大程度避免用户隐私泄露。

3.5.2.2 系统完整性保护

系统完整性确保运行的软件都是经过认证的合法软件，防止系统被非法篡改后进入不可控状态。



系统启动流程中的每一步，都包含对下一级启动对象的合法性校验，如此一环扣一环构成启动的信任链。只有正确通过签名校验的镜像文件才可被加载并运行，包括启动引导程序、内核、固件等镜像文件。启动链条的第一级来源于硬件的启动可信根。在启动过程的任何阶段，如果签名校验失败，则启动流程会被终止。对于尺寸较大无法一次性校验的磁盘分区，星环 OS 在文件系统层面实现了分区粒度的完整性保护，实时发现非法篡改行为，维护系统环境的可信状态。

为防止攻击者刷入存在已知漏洞的低版本系统，利用防篡改存储实现版本防回退机制，确保设备仅能运行最新的安全固件，从而有效阻止通过旧版本系统实施的攻击行为。

3.5.2.3 身份认证与权限管控

身份认证与权限管控的目标是让正确的应用，访问正确的资源。智能汽车包含了大量敏感资源，如控制车辆运动的接口，用户的隐私数据等，一旦这些资源被滥用，都会对用户的行车安全和隐私安全造成威胁。星环 OS 实现了应用级的应用身份认证和权限管控能力，确保应用和资源之间的正确访问关系，以应对权限滥用带来的安全风险。

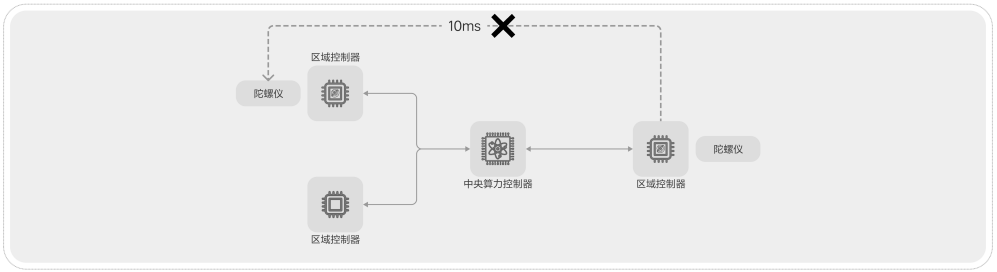
3.5.2.4 可信执行环境

可信执行环境（TEE）是一个保护敏感数据和代码的硬件安全世界，它为应用程序提供一个隔离的执行环境，确保它们在运行时免受外部攻击的影响。星环 OS 充分利用了硬件的安全能力，基于可信执行环境实现了系统的安全信任根，将整个系统的安全水平提升至硬件芯片级别。

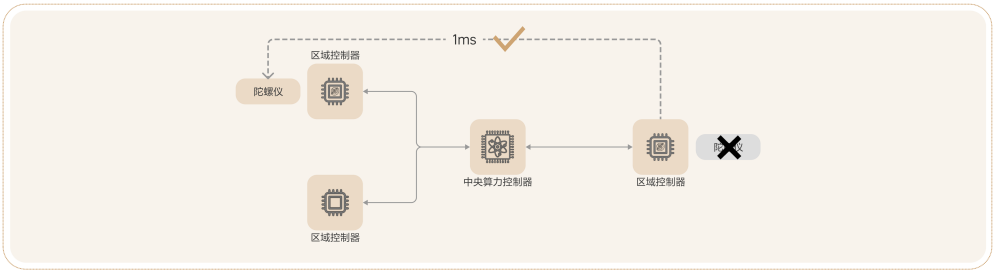
可信执行环境技术将系统的工作状态分为安全世界和非安全世界，TEE 安全系统是运行在安全世界中的软件核心。它驱动了硬件安全能力实现了物理内存隔离，保护系统中最核心的敏感数据和代码。即使非安全世界中的系统（如 Linux）被攻破，攻击者也无法窃取、篡改可信执行环境中的数据和代码。TEE 安全系统负责管理每台设备唯一的硬件根密钥，该密钥仅在安全世界中可访问，无法被导出或复制。基于这把密钥，TEE 安全系统派生出一系列用于数据加密和签名的密钥或证书，保证密钥无法被窃取。

4. 创新场景案例

4.1 传感器跨域共享



传统架构传感器冗余配置，成本浪费



实现传感器跨域共享，并通过确定性网络和资源服务化框架，实现数据访问延迟小于1ms。

【场景描述】

随着汽车智能化、AI 化功能的极大丰富，车辆搭载的传感器数量和种类急剧增加，如用于不同目的的摄像头、雷达、激光雷达以及惯性测量单元（IMU）等。

- **功能多样化驱动传感器增加：** 例如，摄像头不仅服务于智能驾驶的感知，也广泛应用于智能座舱的行车记录、360 环视；IMU 既用于车机导航定位，也作为智驾定位融合算法的关键输入，同时还可能用于车身稳定系统的姿态控制。
- **传统架构导致冗余与成本浪费：** 在传统的、基于功能域或独立 ECU 的分布式架构下，不同功能模块往往需要配置各自独享的传感器硬件，即使这些传感器在物理特性上可能完全相同。这种传感器资源形成“数据孤岛”、无法共享的模式，直接导致了硬件成本增加与线束复杂度提升。

因此，实现传感器的跨域共享，是优化整车成本、简化硬件架构、降低集成复杂度的迫切需求。

【技术挑战】

想要实现传感器的高效、可靠共享，就需要操作系统克服这些技术挑战：

- **超低访问时延（<1ms）：** 对于某些应用场景，如底盘控制系统实时使用远端 IMU 数据，要求从传感器数据产生到远端控制器应用层获取到的端到端时延必须控制在 1 毫秒以内。这对整个数据通路（采集、传输、处理）提出了极高的性能要求。
- **远程设备透明访问：** 操作系统需要将物理传感器设备抽象为标准化的软件服务，并利用高效的通信中间件和实时内核调度能力，实现了远端资源的“本地化”透明访问。应用程序无需关心传感器物理位置，即可通过统一 API 调用所需数据，如同访问本地设备一样便捷。

【解决方案及效果】

星环 OS 站在整车视角优化了内核、网络协议栈、资源服务化等多个组件，构建出完整的传感器共享解决方案，实现一套设备、全车共享：

- **确定性网络通信：**优化车载以太网通信协议栈，显著降低了跨域数据传输的延迟和抖动。
- **资源服务化框架：**将物理传感器封装为标准服务，并通过统一 API 实现了远

程资源的“本地化”透明访问。

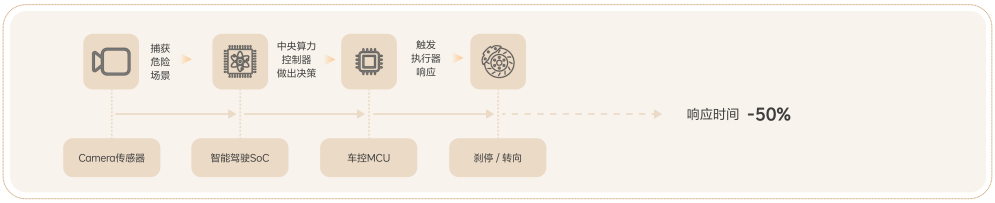
关键技术指标：

- 成功打破“数据孤岛”，实现了传感器在**跨控制器、跨核间的安全、高效共享。**
- 远端传感器访问的端到端时延，从传统方案通常需要的 10ms 以上，**稳定缩短并控制在 1ms 以内**，性能提升明显。

星环 OS 通过以上技术，有效减少了汽车上的摄像头、IMU 等传感器数量。

4.2 AEB/AES 快速反应

以120公里/小时的速度行驶时，系统反应时间每提升30ms，可增加1m的安全距离。



端到端关键事件链，具有全局视角与协同优化能力，响应速度提升1倍，动作稳定性提升5倍。

【场景描述】

理想非常关注用户安全，是行业内最早一批开始做自动紧急制动 AEB 与自动紧急避让 AES 的厂商。AEB/AES 是智能汽车的最后防线，其有效性高度依赖于系统极快且极其稳定的反应速度。在 120km/h 高速下，系统反应时间每缩短 30ms，即可增加约 1 米宝贵的安全距离，直接减少了碰撞概率。AEB/AES 的响应涉及一套完整的“感知-决策-执行”链路——摄像头捕获危险场景 -> 中央计算单元决策 -> 触发执行器响应制动/转向。然而，在传统分布式架构或缺乏系统级优化的方案中，各环节往往独立调度、通信延迟不定、缺乏有效协同，如同“三个独立部门顺序汇报”，导致端到端总时延过长且抖动剧烈，难以满足这类安全关键功能所需的确定性超低延迟要求。

【技术挑战】

为实现安全关键链路的端到端确定性与超低时延，对星环 OS 提出了系统性的技术挑战：

- **全局高精度时间同步：**必须在整车范围内，实现跨计算节点（传感器、计算单元、执行器）的精确时间同步。这是实现分布式系统协同感知、决策与控制，以及进行确定性调度的基础。
- **端到端任务链协同与优化：**操作系统需要具备全局视角，能够理解并管理跨越多个处理器核、甚至多个控制器的完整任务依赖关系，并基于此进行端到端的时延分析、资源分配和协同调度优化。
- **关键性任务的调度与隔离：**关键任务与非关键任务共享相同的硬件资源，操作系统需要严格隔离不同实时性要求的任务，并进行可预测的混合关键性实时调度，确保最高优先级的安全任务不受干扰。

【解决方案及效果】

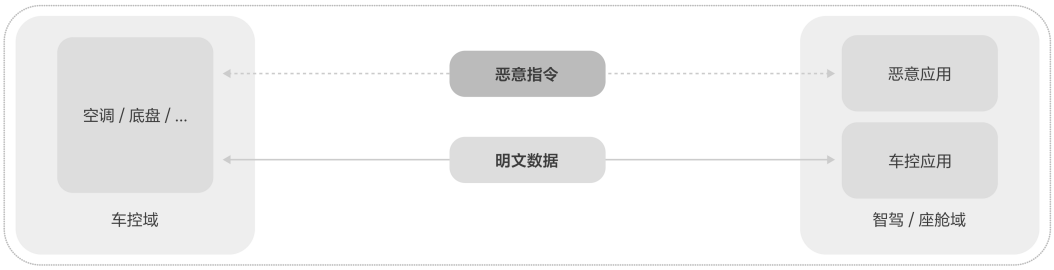
星环 OS 集成了高精度时间同步、硬实时内核、端到端确定性调度框架、确定性通信管理能力等多项技术，从系统软件层面确保了关键链路的确定性与低延迟。

关键技术指标：

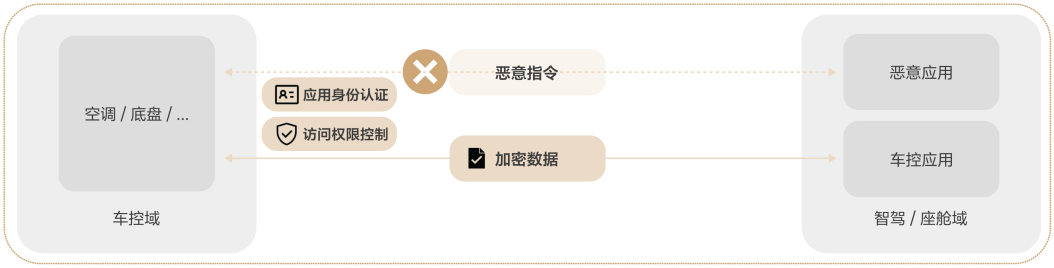
面向端到端场景关键事件链处理，全局视角与协同优化能力，实现端到端时延显著下降：

- **响应速度：**相比传统方案，端到端时延降低约一半，即整体反应速度提升 1 倍。
- **动作稳定：**端到端时延抖动减小至原先的 1/5，确保了在各种工况下响应的高度一致性和可靠性。

4.3 通信链路安全防护



通信链路缺乏完善的信息安全机制，有恶意指令侵入风险。



构建具有应用身份认证、访问权限控制和数据加密的完善信息安全机制
软硬件结合实现，关键加解密算法性能提升4倍。

【场景描述】

智能汽车内部各控制器需通过通信中间件频繁交互数据与指令，协同完成复杂功能。若通信中间件缺乏完善的内建安全机制，恶意应用或被攻破的系统组件可能利用此缺陷，向通信总线注入伪造或恶意的控制指令，从而实现非法控车，对用户行车安全构成严重威胁。因此，确保车内通信的真实性、完整性和机密性，是保障整车安全的基础。

【技术挑战】

在复杂的车载网络环境中实现端到端的安全通信，面临关键的技术挑战：

- **应用身份认证**：身份是合法性判定的基础。车内通信基于以太网，一端的通信实体难以根据网络报文信息辨别对端身份；
- **密码学算法性能**：对通信数据加密是保障通信安全的通常做法，由于在通信过程中增加了额外的加解密操作，密码学算法性能直接影响通信时延；

【解决方案及效果】

星环 OS 通过内置于操作系统层面的安全机制和软硬件协同设计，有效应对上述挑战，构建了可信且高效的车内通信环境：

- 基于 PKI 证书体系，实现动态的应用身份授予与认证机制，让每个应用都有

了唯一的身份，防止恶意应用伪造合法身份控制车辆系统。

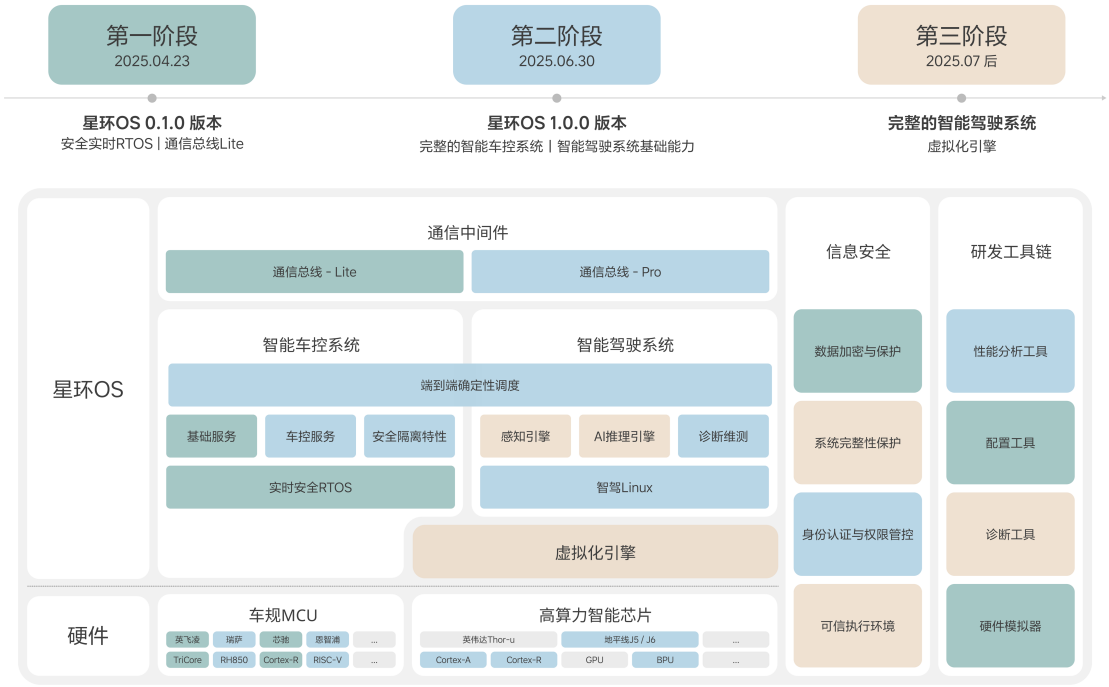
- 基于通信实体的权限控制机制，精细化限定应用与控制指令之间的关系，防止越权访问，杜绝因权限滥用导致的车辆误控或用户隐私泄露。
- 基于通信数据加密机制，实现数据端到端加密传输，确保用户敏感信息在传输过程中无法被中间人截取或篡改。

关键技术指标：

- 通过软硬结合的方式实现，密码学算法性能相较于软件方式提升了 4 倍。

5. 开源计划

星环 OS 会从 2025 年 4 月底逐步开放源码：



星环 OS 开源的主要目的是促进行业合作，旨在破解行业“重复造轮子”的困局，通过生态共建实现车企之间、车企与芯片厂商之间的互利共赢，最终普惠每个用户。星环 OS 采用宽松型的 Apache License，不会通过开源收取费用，不干涉代码的使用方式，也不控制使用者的数据。

站在车企视角，操作系统是实现智能化和差异化竞争的关键，自研操作系统已成为领先车企的必需品。理想汽车的星环 OS 需求直接源于其造车实践、用户反馈和产

品规划，并通过大规模自有车辆的实际运行进行验证和快速迭代，不断提升 OS 质量。这对于解决行业普遍性问题具有重要价值。

站在芯片供应商视角，星环 OS 为加速新芯片的集成适配做了四方面努力：一是原生适配 ARM A/R 核、TriCore、RISC-V 等主流 CPU 架构；二是兼容 MCAL、HAL 等主流驱动框架；三是可以与芯片厂商深度协同开发关键驱动并联合优化；四是通过 PC 端系统模拟器实现软硬件并行开发。这些能力可以缩短新款芯片量产上车的周期。

因此，星环 OS 欢迎更多车企以及产业链上下游伙伴加入星环 OS 的生态建设中。一群人一起走，将会走得更稳、走得更远。