车载操作系统总体技术要求研究报告

汽标委智能网联汽车分标委

资源管理与信息服务标准工作组

2021 年 7 月

目　录

前 言

随着智能网联汽车的飞速发展，传统的分布式的电子架构已经不 能满足整车的需求，电子架构正在朝着由分布式向集中式，由封闭到 开放，面向智能化、网联化和集中化的路径发展。2019 年 10 月，汽标委发布了《车用操作系统标准体系》，规范了车用操作系统定义，划分了车用操作系统边界，明确了车用操作系统分类，构建了车用操作系统标准体系，为车用操作系统标准化工作的开展提供了指导方向。车用操作系统是运行于车内的程序集合，管理硬件资源，提供软件平台和界面接口，为上层应用提供基础服务。车用操作系统从与整车正常运行是否相关的角度，分为车控操作系统和车载操作系统。车载操作系统主要应用于中控、仪表和 T-box，提供车载信息娱乐服务，可具备网联功能，提供导航、多媒体娱乐、语音、辅助驾驶、AI 等高级功能。近年来，随着汽车电子、云计算、大数据等技术的快速发展，车载操作系统的架构和技术功能不断演进，急需开展相关标准化需求研究，指导车载操作系统的研发、测试、示范和运行等。

本研究报告的车载操作系统研究范围是介于应用程序和硬件抽象层之间，主要由车载操作系统内核、资源抽象层、基础库、基础服务、运行时环境及程序运行框架组成。

本研究报告涉及的车载操作系统架构在《车载操作系统架构研究报告》中有相应的详细分析。

在此衷心感谢参加研究报告编写的各单位、组织及个人。

组织指导：汽标委智能网联汽车分标委

牵头单位：国汽（北京）智能网联汽车研究院有限公司、阿里巴巴（中国）有限公司

参与单位：斑马网络技术有限公司、上汽大众汽车有限公司、国汽智控（北京）科技有限公司、中国汽车技术研究中心有限公司、中国软件评测中心（工业和信息化部软件与集成电路促进中心）、东软集团（大连）有限公司、惠州市德赛西威汽车电子股份有限公司、大陆汽车车身电子系统有限公司、高通无线通信技术（中国）有限公司、长城汽车股份有限公司、北汽福田汽车股份有限公司、北京汽车股份有限公司、上海博泰悦臻电子设备制造有限公司、华为技术有限公司、中兴通讯股份有限公司、泛亚汽车技术中心、江苏智能交通及智能驾驶研究院、Elektrobit、上海机动车检测认证技术研究中心有限公司、安徽江淮汽车集团股份有限公司、东风商用车有限公司、一汽解放汽车有限公司、上海汽车集团股份有限公司零束软件分公司、东风日产乘用车公司、东风汽车集团有限公司技术中心、上汽通用五菱汽车股份有限公司、襄阳达安汽车检测中心有限公司、北京百度智行科技有限公司。

参与人员：刘卫国、霍克、王琳、刘大鹏、卜烨雯、金一华、刘克、满志勇、田思波、潘晏涛、吴含冰、张路、郭盈、周波、周海龙、李伟、伍宇志、唐侨、许劲、李俨，陈书平、石晓坤、毛雷、来冬清、钱国平、刘东、刘珺、秦文婷、顾照泉、周铮、陈晓、陈琨、江浩、刘翔海、王红燕、周鹏、王观、李兵、黄慧丽、桂绍靖、管杰、吴勇刚、郑岩、孙华、王圭、邹德英、武扬、程周、王振、彭杨、江恒、高海龙、彭伟、贾元辉。

1 术语定义及缩略语

1.1 术语与定义

下列术语与定义适用于本文件。

1.1.1 车用操作系统 vehicle operating system

运行于车内的系统程序集合，以实现管理硬件资源、隐藏内部逻辑提供软件平台、提供用户程序与系统交互接口、为上层应用提供基础服务等功能，包含车控操作系统和车载操作系统。

1.1.2 车载操作系统 on-vehicle operating system

运行于车载芯片上，管理和控制智能网联汽车车载软件、硬件资源的软件集合，为智能网联汽车提供除驾驶自动化功能实现以外的服务，包括车载信息娱乐、网联、导航、多媒体娱乐、语音、辅助驾驶、AI 等服务。

1.1.3 单系统架构 single system architecture

单个车载操作系统的架构，由车载操作系统内核、资源抽象层、基础库、基础服务、运行时环境、程序运行框架和车载操作系统安全模块组成，对底层硬件和上层应用程序提供统一的接口。

1.1.4 多系统架构 multisystem architecture

在同一套硬件之上运行多个车载操作系统单系统的架构，分为硬件隔离、虚拟机管理器、容器三类多系统基础架构，以及两类或三类基础架构的混合架构。

1.1.5 资源抽象层 resource abstraction layer

运行于车载操作系统内核上，为车载操作系统应用和基础服务提供SoC芯片平台硬件的资源抽象、整车信号的资源抽象、外部IoT设备的资源抽象和外围IC的资源抽象。

1.2 缩略语

下列缩略语适用于本文件。

ADAS：高级驾驶辅助系统（Advanced Driving Assistance System）

AI：人工智能（Artificial Intelligence）

API：应用程序编程接口（Application Programming Interface）

ASIL：汽车安全集成等级（Automotive Safety Integration Level）

OBD：车载诊断（On-Board Diagnostics）

OBU：车载单元（On-Board Unit）

OTA：空中下载（Over the Air）

RTOS：实时操作系统（Real Time Operating System）

V2X：车与外界的互联（Vehicle-to-Everything）

2 车载操作系统技术现状

2.1 车载操作系统技术比较

目前，应用于车载领域的操作系统有括QNX、Linux/AGL、RT-Linux、AliOS、AliOS RT、Android和鸿蒙OS等。各车载操作系统的对比参考如表1所示。

表 1 车载操作系统对比

车载 OS	QNX	Linux	Android	AliOS	鸿蒙 OS
技术性能	微内核	宏内核	宏内核	宏内核	宏/微内核
	编译执行	编译执行	编译/解释混合执行	编译/解释混合执行	编译执行
可扩展性	高	高	高	高	高
是否可裁剪	否（微内核，无剪裁的必要）	是	是	是	是
是否开源	否	是	是，有风险	部分开源	部分开源
硬件支持	多	多	多	多	少
是否具备功能安全证书	AISL-D			AISL-D（RT-AliOS）	AISL-D

2.2 车载操作系统技术应用现状

车载操作系统可用于实现一芯多屏（多屏融合、多屏互动等功能）、单屏多系统（虚拟运行环境、多应用生态融合等功能）、及一芯多功能单元（信息娱乐、T-box等功能）的方案。

（1）一芯多屏应用

一芯多屏应用是车载操作系统多系统应用中最常见的使用场景。典型应用有理想One智能驾舱多屏系统。该系统使用了TI J6和高通S820a两块芯片支持了Linux和Android两套操作系统在四块屏幕上的显示。

（2）单屏多系统应用

单屏多系统应用主要用于对安全域非安全域有区分需求，并希望通过多系统扩展生态圈的使用场景。典型应用有南北大众新一代信息娱乐

系统CNS3。该系统使用Renesas RCar H3N芯片支持Linux和Android两套操作系统在一块屏幕上的融合显示。

（3）一芯多功能单元应用

为了充分发挥车载SoC的性能，满足汽车上对多人多屏多应用的场景需求，在车载系统上更多地引入虚拟机系统方案。

2.3 车载操作系统信息安全技术现状

目前主流的车载操作系统信息安全技术是基于ARM TrustZone的TEE技术。ARM TrustZone是基于硬件的安全功能，通过对硬件架构进行修改，在处理器层次引入了两个不同权限的保护域——安全域和普通域，任何时刻处理器仅在其中的一个环境内运行。同时这两个域完全是硬件隔离的，具有不同的权限，普通域中运行的应用程序或操作系统访问安全域资源受到严格的限制，反过来安全域中运行的程序可以正常访问普通域中的资源。两种域之间的硬件隔离和不同权限等属性为代码和数据提供了有效安全机制，普通域用于运行操作系统，提供了正常执行环境（Rich Execution Environment，REE）；安全域则使用安全小内核（TEE-kernel）提供可信执行环境（Trusted Execution Environment，TEE），机密数据可以在TEE中被存储和访问。即使普通域中操作系统被破坏入侵或ROOT，攻击者也无法读取或篡改TEE，保护关键业务与数据。TEE支持设备安全启动、安全升级保护、敏感信息保护、安全管理等，对关键业务实时保护，TEE不需要额外的硬件支持，可由软件控制自主切换。

（1）信息通讯的安全防护

系统可使用非对称加密技术、鉴权技术、证书管理和认证技术、安全协议技术、防火墙技术 、VPN等技术，对系统通信进行信息安全防

护。

（2）系统服务层的信息安全防护

系统可使用访问控制、恶意行为检测、加密存储等技术，对系统服务层进行信息安全防护。其中访问控制机制，依据安全策略控制用户、进程等主体对文件、数据库等客体进行访问。禁止不必要的服务（如FTP服务等），禁止非授权的远程接入服务，禁止 ROOT 用户直接登录，且限制用户提权操作，删除或禁用无用账号 。

（3）系统应用的信息安全防护

系统应用可通过混淆、加壳防护、防反编译、安全应用管理器等技术进行信息安全防护。

（4）密码加密安全

常见密码算法类型包括对称加密算法、非对称加密算法、杂凑算法。

- 对称加密算法：数据加密标准DES、高级数据加密标准AES、国际数据加密算法IDEA、分组密码算法SM4。
- 非对称加密算法：RSA、椭圆曲线密码算法ECC、数字签名算法DSA、椭圆曲线公钥密码算法SM2。
- 杂凑算法：MD5（不安全）、SHA1（160比特、不安全）、 SM3、SHA256，用于保障数据完整性、数字签名、MAC设计。

随着国密算法的引入，国密安全模块逐渐成为系统中的组成部分。国密安全服务通过固化在只读存储中的根证书和CA平台进行交互，当系统收到外部请求，服务先将请求报文中的证书部分进行校验比对，再通过证书附着的公钥对请求内容进行验签，以此确认报文真实可靠，通过这种方式可以服务对外部进入的软件升级包、控制命令、推送消息等进入的消息进

行验证，防止其被篡改。同时，组合AES、3DES以及国密的SM2和SM4构成的对称、非对称加密组合为通讯的私密性提供有效的保护。国密安全服务能够为总线通讯、网络连接、数据存储等其他模块提供加密、验签、签名和密钥存储、证书管理等功能。

2.4 用于车载环境的虚拟机技术现状

虚拟机技术包括 Type-1 和 Type-2 两种类型。

（1）Type-1

Type-1 型虚拟化技术通过提供系统虚拟机（例如模拟类似于真实硬件的整个系统），允许未经修改的客户机操作系统作为客户机运行，该模式需要借助硬件的虚拟化支持，例如 X86 架构 AMD-V/Intel VT，ARMv8 和 Power 架构的虚拟化 profile 等。其优势在于客户操作系统不需要进行任何修改就可以使客户操作系统正常运行，并且它们并不知道自己在虚拟化环境下运行。这种虚拟化技术使用简单，具有很好的兼容性，但这种虚拟化方式由于需要捕捉客户操作系统发出的敏感特权指令，进而通过虚拟机管理器来模拟系统的特权指令，这个过程将降低指令的执行速度。

（2）Type-2

由于 Type-1 技术对硬件性能的要求较高，并且运行效率不高。因此提出了 Type-2 虚拟化技术解决方案。在 Type-2 中，虚拟机系统（客户 OS）的内核需要经过特殊修改，把特权指令改成对虚拟化层 API 的调用。在 Type-1 的基础上，把客户操作系统进行了修改，增加了一个专门的 API，

这个 API 可以将客户操作系统发出的指令进行最优化，即不需要 Hypervisor 耗费一定的资源进行翻译操作，因此 Hypervisor 的工作负担变得非常的小，因此整体的性能也有很大的提高。Type-2 技术可以减少模拟执行特权指令造成的性能开销，另外这种技术还可以优化 I/O 访问和操作，运行速度基本可达到本机速度。

目前，可应用于车载环境的虚拟机典型技术方案有以下几种：

a) Xen 虚拟机

Xen 是一个开放源代码的 Hypervisor 平台，是在单个计算机上运行多达 100 个满特征的操作系统。操作系统必须进行显式地修改（“移植”）以在 Xen 上运行（但是提供对用户应用的兼容性）。这使得 Xen 无需特殊硬件支持，就能达到高性能的虚拟化。Xen 虚拟机可以在不停止的情况下在多个物理主机之间实时迁移。在操作过程中，虚拟机在没有停止工作的情况下内存被反复的复制到目标机器。虚拟机在最终目的地开始执行之前，会有一次 60-300 秒的非常短暂的暂停以执行最终的同步化，给人无缝迁移的感觉。类似的技术被用来暂停一台正在运行的虚拟机到磁盘，并切换到另外一台，第一台虚拟机在以后可以恢复。

典型的 Xen 的应用架构图如图 1 所示。

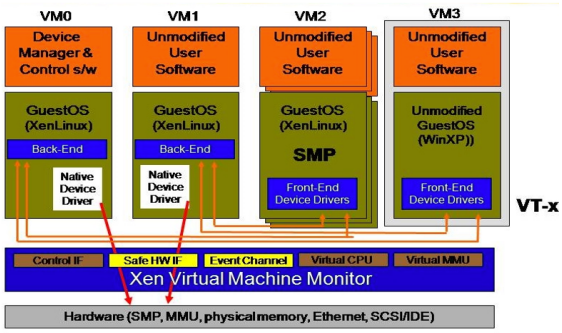

图1 Xen 虚拟机架构

b) QNX 虚拟机

由 QNX 提供的闭源虚拟机（如图 2 所示）是基于 type1 实时优先级的微内核管理程序，用于管理虚拟机。QNX 虚拟机管理程序可以更容易地获得安全关键部件，并通过隔离不同的客户端操作系统的非安全关键组件，以确保安全性。QNX Hypervisor 能够满足嵌入式零停机生产系统的精度要求。

QNX 虚拟机可以支持多个不同的架构，在这些架构中可以分别运行 QNX 的基础服务(Foundation),主机端(Host)，虚拟机端(GVM)。

基础服务模式:所有的物理设备的驱动都运行于 QNX 的主机端(Host)，所有的用户级应用都运于 QNX 虚拟机中(如显示管理，启动动画，静态画面等)

Host 模式，所有物理硬件的驱动均运行于 QNX 主机端，用户的应用程序和驱动程序都运行在 Host 端，这种模式下通常只能运行一个 Linux 客户端。

多用户模式，所有的物理驱动均运行于 QNX host 端，用户级应用和驱动都运行于主机 Host 端,这种模式通常可以支持 2 个以上 Linux 客户端。

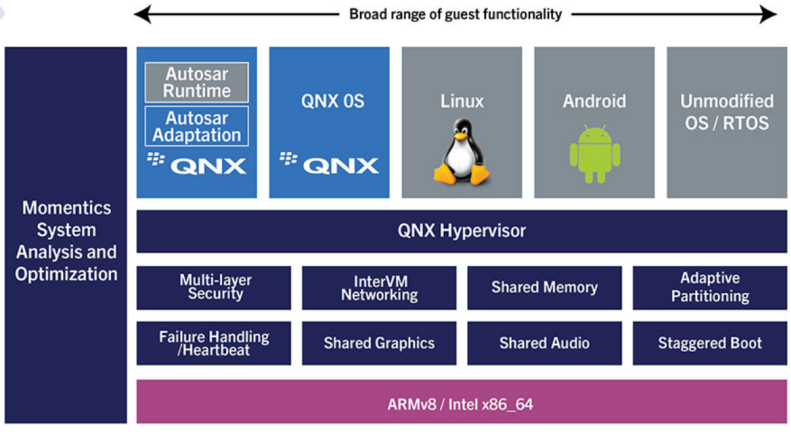

图 2 QNX 虚拟机架构

c）ACRN 虚拟机

ACRN 是一个比较成熟、稳定的基础虚拟化技术开源方案，由 Intel 联合东软开发的基于 Intel 芯片的虚拟化技术，目前只能应用于 Intel 系列芯片。ACRN 可以灵活的支持逻辑分区、共享和混合模式。

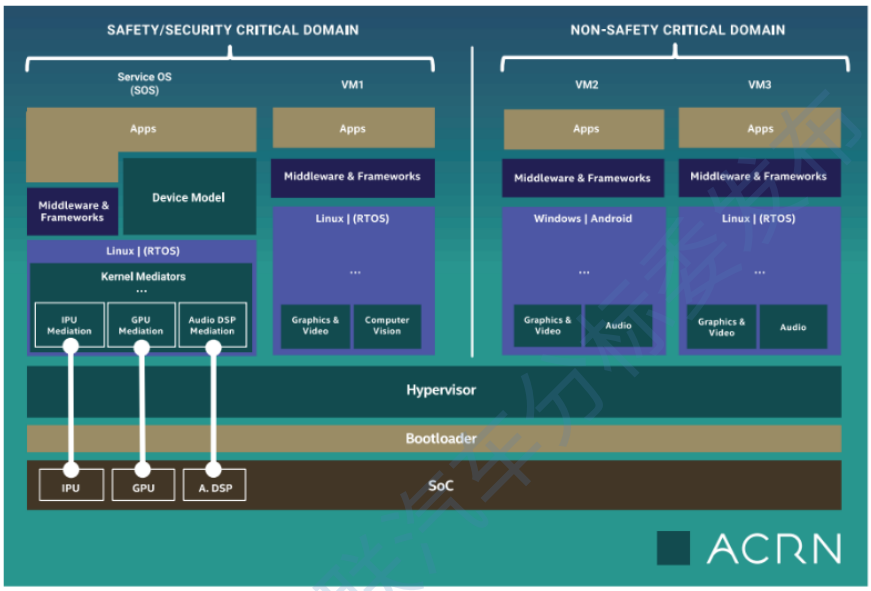

图3 ACRN 虚拟机架构

如图 3 所示，硬件资源可以被分成两个区域部分，一部分可以直接被 Hypervisor 启动，甚至可以在 VM 启动之前。若果没有提前启动的 VM，Service VM 是第一个启动的，并且可以直接获取硬件资源。

ACRN 为嵌入式和车载应用量身定制的虚拟化方案，追求灵活性、轻量级，并且在开发阶段对代码量有严格控制。由于代码量小，在做实时性、稳定性和功能安全的认证时就会比较方便。ACRN 在设计上考虑了功能安全的要求和实时性要求，在开源社区里也在推动符合功能安全认证的开发模式，是针对车载的技术方案。它直接运行在芯片上，有 Service OS 的概念，这个概念是为了把 I/O 设备支持单独拿出来放在 OS 里，现在开源的 Service OS 是基于 Linux 的。在车载领域有很多 I/O 资源的共享，下图是

具体落实下来后可能的架构。Service OS 会把仪表盘做在里面，ADAS 的显示功能做在 ADAS 的 VM 里，Android、中控和后台都有单独的虚拟机。从 I/O 的延迟来讲，实时性能不是最好，因为 I/O 访问要经过 Service OS。

d) COQOS 虚拟机

COQOS 虚拟机（如图 4）由 OpenSynergy 公司推出，不仅可用于座舱，也可以用于自动驾驶系统，对自适应 AUTOSAR 也有对应，并且也通过了 2018 版的 ASIL B 级认证。

最新的 COQOS Hypervisor SDK 围绕安全高效的虚拟机管理程序构建，可在一个 SoC 上同时运行多用途操作系统以及 RTOS 和 AUTOSAR 兼容软件。

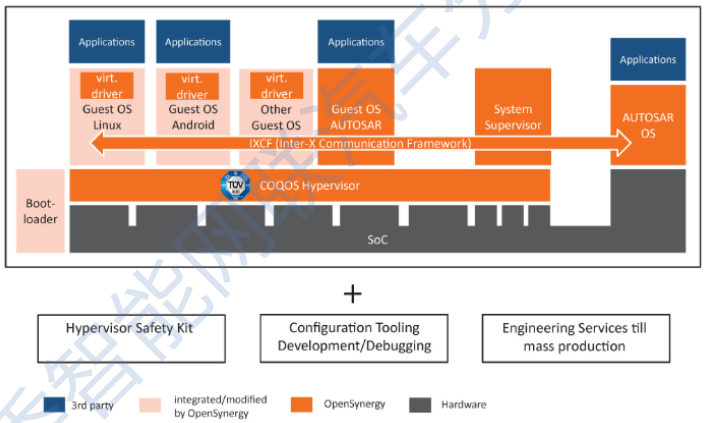

图4 COQOS 虚拟机架构

e) 哈曼虚拟机

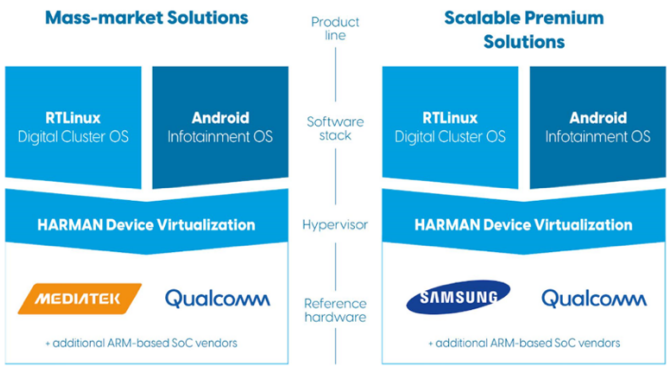

图5 HARMAN 虚拟机架构

HARMAN 虚拟机（如图 5）可以被用作各种硬件/软件平台配置的抽象层，每个软件平台可以在各自的参考硬件上使用，每种硬件平台已经有一个 A 或 B 样本可用于快速汽车制造商原型和车辆集成集群，在给定参考硬件上的不同的仪表和信息娱乐系统组合可以允许汽车制造商在不同市场、品牌和型号上进行区分。

3 车载操作系统技术演进趋势

3.1 单系统向多系统转变

随着车辆的功能从单一的安全驾驶功能逐步向智能化、娱乐化、个性化过渡，单一系统已无法满足多样的功能需求。多系统的架构设计和技术方案在不断演进中，从而支持不同的车载应用。

3.2 单核向多核转变

多核芯片的出现让车载操作系统设计变得更加复杂。在单核 CPU 中，并不需要考虑 CPU 间负载均衡的问题，无论线程如何切换，CPU 始终处于工作状态，并不会影响程序运行的总时间。但对于多核 CPU，系统必须考虑负载均衡的问题，避免出现负载小的 CPU 出现空闲等待的现象。

3.3 更丰富的基础服务

传统车载 ECU 的服务能力较为专一，仅提供如驾驶、娱乐等某一方面的基础服务，因此各 ECU 的操作系统功能也较为单一。今后的车载操作系统通过如下方式提供更加丰富的基础服务：

- 某些传统模块通过增加能力承载更多的服务， 例如传统的车载操作系统仅仅用于多媒体娱乐，加入导航功能后为驾乘人员提供了行车线路，带来了驾乘便利；在增加了 ADAS 等辅助驾驶能力后，为驾乘人员的安全驾驶提供了帮助。
- 随着某些模块能力的提升，模块自身可承载更强的服务能力，比如定位和地图服务，随着定位的能力越来越强，地图的精度和实时性越来越高，将导航功能逐渐升级成辅助驾驶能力，甚至能够为自动驾驶的关键决策提供原始数据。

车载操作系统逐步可提供的典型的新基础服务包括：

1）互联服务

传统车载系统随 ECU 固化，除非特殊情况返厂，一旦出厂基本不会进行修改或者功能变更。而车载系统在支持网络互联的功能后，一方面系统的 OTA 功能能够为系统固件升级以及上层应用、模块的更新提供服务，形成一个持续性的软件服务平台。其次通过车载网关访问公共互联网和云服务平台，为各种复杂应用提供更多的网络接入和服务功能。而通过专网互联，能够为主机厂或者监管部门提供车辆特别是运营车辆的实时数据。同时 V2X 的通讯能力也为车路协同提供了更便捷的渠道。另一方面，通过定制的上层应用，手机、笔记本等便携设备和车载系统产生互动，内容包括电子商务等支付业务以及定制化的个人服务。

2）辅助驾驶服务

传统车载系统提供的人机互动相对较少，主要集中于主机面板，需要通过驾乘人员的手部操作以及观看输出结果获得反馈信息。而今后的系统则提供了更丰富的人机交互方式，如通过语音、手势对车辆系统下达指令可以为驾乘安全提供更多保障；系统还能通过车内传感器对驾乘人员的身份进行识别，提供相应个性化的驾乘环境，以及为了提供更好的视觉体验提供的抬头显示以及 AR 导航等等。

3）AI 服务

随着车载系统的算力提高和高速总线的出现，各个感知设备提供各种数据，如视觉、指纹、压力传感器获取驾乘人员的信息，车载操作系统需有高效安全的数据采集和分析机制进行分析和计算。车载操作系统需具备在端侧进行数据训练的能力从而在不上传用户数据的情况下让系统具备越来越个性化的智能，根据一定的规则和算法制定策略，分析比对驾驶习惯、驾乘人员状态、外部环境，提供最佳的驾舱环境配置策略。在人机交互和主动服务上随着驾乘人员对于系统使用的深入而越来越智能，给驾乘人员带来优质安全的体验。

3.4 提供适配硬件和应用的统一接口

传统车载 ECU 模块的硬件平台和操作系统基本自成一体，各个模块的软件接口和系统基本不能兼容，替换不同厂商的模块和系统往往就意味着局部架构的改变。今后的操作系统能够提供统一的接口，为今后产品的封装和模块化提供有力的支持。

3.5 安全性要求更高

传统的车载系统平台比较封闭，功能较单一，且分布式的架构使得系统相对安全。随着系统的复杂度和开放程度越来越高，架构越来越集中化，以及服务越来越网联化，使得整个系统功能越来越丰富的同时，也为外部入侵提供了途径，增加了安全威胁面和攻击面，信息安全和功能安全的重要性也慢慢凸显出来。

4 车载操作系统技术标准化需求

4.1 多系统技术标准化

由于不同的服务和应用对车载操作系统的实时性、安全性和功能等要求不同，多系统的架构设计和技术方案在不断演进中，包括硬件隔离技术、虚拟机技术等。但目前对多系统的技术要求尚未形成统一共识，车厂很难根据自身对多系统的安全性和灵活性等需求选择合适的多系统技术方案。

4.2 多核技术标准化

多核技术的复杂度较高，运行在不同内核上的应用为了互相访问、相互协作，需在 IPC 机制、共享内存的数据结构和同步原语等方面进行标准化。

4.3 基础服务技术标准化

目前对车载操作系统可提供的基础服务尚未统一其定义及技术能力要求，例如高精度定位具体可实现的精度和可支持的场景是什么等，不便

于车厂基于自身应用需求选择适合的车载操作系统。因此，需要对基础服务的技术要求进行标准化。

4.4 接口技术标准化

统一的上层和底层接口技术是车载操作系统实现与硬件和上层应用程序解耦的关键，从而适配不同的硬件平台和兼容不同的应用。

4.5 安全技术标准化

车载操作系统的信息安全和功能安全十分重要，包括网联网关的安全防护和主动检测、应用模块隔离和系统的访问控制、功能备份和异常恢复等各种防护手段，都逐步会成车载系统重要的安全屏障。因此，急需对车载操作系统的功能安全和信息安全提出新的要求。

5 多系统技术要求建议

5.1 虚拟机技术要求

（1） 硬件系统资源分配和共享（静态和动态）；

（2） 外设资源的分配和共享（静态和动态）；

（3） 域隔离，保证操作系统之间的独立性；

（4） 支持多操作系统间通信，支持系统间的数据共享；

（5） 支持系统安全和功能安全相关方案；

（6） 管理各操作系统优先级，实现自适应分配最大化利用硬件资源，并满足对时间敏感需求和系统服务质量的要求；

（7） 管理各操作系统生命周期，包括启动，运行，关闭，重启等状态；

（8） Hypervisor的存在应保证其实时性，不应对各操作系统性能产生影响；

（9） 具有不同安全性和安全级别的应用程序可以在相同的硬件上运行，通过硬件或软件分区相互保护；

（10）全局电源管理。

5.2 硬件隔离技术要求

硬件隔离技术支持将硬件资源通过硬件分区的方式进行划分和管理，硬件资源的所属分区拥有对该资源的访问和管理权限，其他分区不能对该资源进行操作。针对具备内存保护单元的多核 SoC，可创建两个系统的硬件分区，将不同的硬件资源划分到两个分区，并将不同 CPU 内核划分到两个分区，实现两个分区上的 CPU 分别独立运行各自操作系统。

硬件隔离技术的具体要求包括：

（1）管理系统资源，如 SoC 外围设备(Resource)，内存区域(Memory)和引脚等。

（2）允许将资源划分为与不同执行环境关联的不同所有权组。

（3）允许所有者配置对资源的访问权限。

（4）提供硬件强制隔离。

6 多核技术要求建议

车载操作系统面向多核芯片硬件平台时，在资源共享、内核调度算法、

IPC、CPU 负载均衡等方面需支持相应的技术要求，确保让每个内核独立访问某种资源，不会被其他内核上的应用程序争抢。

7 基础服务技术要求建议

7.1 互联服务

车载操作系统应支持与 T-Box、OBD、Tuner、V2X-OBU 等多车载终端互联服务。车载操作系统应支持与个人移动端、云端的信息交互互通。系统应支持通过无线（如 Wi-Fi、蓝牙）或有线（如 USB）等方式，与个人移动端投屏互联，实现娱乐智能服务、辅助控制服务等；系统应内置常用 MQTT、CoAP、HTTPS 等网络协议，具备与云端互通能力，连接各种云服务。

7.2 地图及定位服务

车载操作系统应支持地图服务（位置服务），为包括地图在内的各类应用程序提供位置相关的 API。地图服务模块可结合 GNSS、各类传感器以及车辆信号数据，采用复杂的数据融合及滤波算法，实现准确的位置信息求解，为包括地图在内的各类应用程序提供位置相关的 API。云端可实时获取上传的位置数据，部署各类服务，将位置信息与第三方服务或数据挖掘的数据进行融合，为用户提供所需的与位置相关的增值和个性化的服务。

7.3 语音服务

车载操作系统应支持语音服务，其架构可分为语音形象、语音编程框

架、语音服务模块、语音开放平台和语音自学习系统五部分。

- 语音形象：语音形象是一个特殊的系统内置应用，提供了所有交互展示型的功能，并向上层的语音类应用提供 API。比如：当有语音输入（ASR）操作时，语音形象应用会将识别出来的文字显示在屏幕上。当有语音播报（TTS）时，语音形象应用会处于播报态。语音形象应用通过接收下层语音框架上报的多种状态信息（唤醒态、识别态、处理态、播报态等），展示相应状态所对应的可视化界面。
- 语音编程框架：提供语音 API 支持基于 CAF 编写的语音应用（CloudApp）或车载小程序语音应用。
- 语音服务模块：语音框架还提供了多个语音服务，主要是提供本地语音语义能力和与云端能力对接，主要包括：声学前端和语音唤醒模块（WUW），语音识别模块（ASR），语音合成模块（TTS），语音声纹模块（VPR），语义理解模块（NLU），对话管理模块（DM），语义生成模块（NLG）等。整个架构是一套端云一体组件化可插拔的技术架构。
- 语音开放平台：支持第三方应用开发者自助式编写离线/在线的 NLU、DM、NLG 等技能，通过技能管理还可进行可视化编辑、管理等工作。

- 语音自学习系统：通过将语音唤醒、语音识别等用户交互的音频数据进行定期的回流，并对获取的大量语音数据进行分析和训练，可以得到更好的相关模型参数，通过闭环优化系统，定期将这些模型参数及相关资源进行下发，使最终用户可以感觉到语音定期会更新很多新的能力，语音唤醒、语音识别、语义理解等效果也变得更好，而且能提供很多个性化的优化效果，用户体验也会变得更好。

7.4 多媒体服务

车载操作系统应支持基础的图像、音频、视频的编解码播放和控制服务。

7.5 云服务能力

7.5.1 自动升级服务（OTA）

车载操作系统应支持自动升级服务（OTA），通过网络自动检测到新版本（如新功能、安全补丁等）并下载安装，为新功能、安全补丁的发布提供升级通道。

自动升级服务架构如图 6 所示，由客户端与服务端两部分构成。客户端以系统级服务的方式运行在操作系统中，支持服务端检测系统版本、下载升级包、安装升级包等操作。服务端应包括如下模块：

- 操作控制台：主要为配置管理员提供发布版本、上传升级包的操作功能；

- 版本查询服务：向客户端提供版本检测查询服务，要能稳定支撑大量设备的并发查询请求；
- 升级包下载服务：向客户端提供升级包下载服务，通常需要依赖分布式的文件存储并接入数据中心以提高下载速度；
- 数据存储：结构化的版本数据存储以及非结构化的升级包文件存储；
- 其他：升级过程全链路数据分析模块、版本升级效果统计模块等，以帮助掌握升级过程并跟踪新版本升级情况和效果。

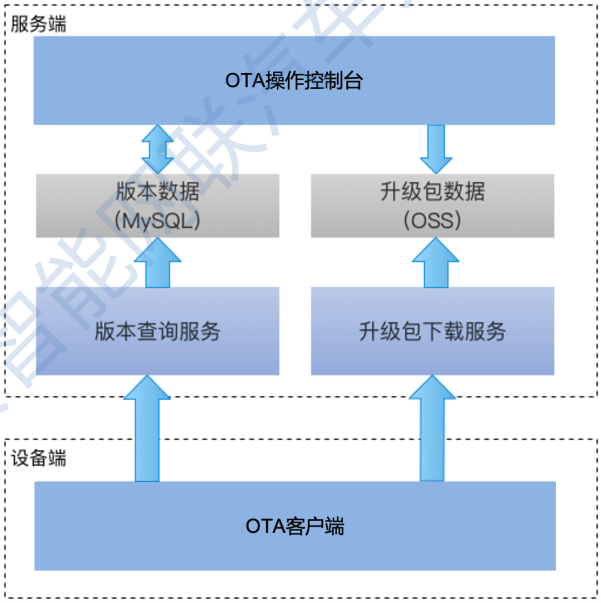

图 6 自动升级服务（OTA）架构

7.5.2 账号

帐号包括系统账号、系统中安装的应用账号和服务帐号。不同的帐号服务可有各自的独立的用户界面、操作流程以及鉴权方式。

系统账号是面向互联网汽车的车载操作系统中云+端资源的串接器，

通过帐号统一接入服务端的服务资源和计算资源，实现数据共享和服务融合，以及多移动端的信息流转和互通。系统帐号应提供分层次的应用授权访问机制，通过服务端鉴权保证服务端访问的安全性和合法性。系统帐号应支持对本地资源的管理，通过服务端鉴权保证在本地增加、删除以及切换帐号对应本地用户资源并访问该资源的安全性和合法性。

车载操作系统应支持的账号管理机制包括：

(1) 统一管理和开放帐号服务

(2) 匹配系统帐号和本地用户资源管理

(3) 支持多系统帐号

(4) 帐号管理与帐号服务解耦

(5) 多帐号数据隔离

7.6 辅助驾驶服务

车载操作系统可支持简单的辅助驾驶服务，如 360 环式、倒车影像、盲区提示、车窗控制等。结合 CAN 消息的硬件抽象、标定、电源策略、时间策略和亮度策略，基于图像拼接算法等技术，完成单一车控功能或组合式车控功能。

7.7 AI 服务

随着车机芯片或者异构 AI 芯片的算力增强，智能座舱的智能化程度越来越高，对于视觉感知、语音交互、网联通信等功能需求的不断提升，

车载操作系统的 AI 能力也许相应提升。

车载操作系统需支持结合 TensorFlow、MXNet、Caffe、PyTorch 等多种机器学习框架，同时再利用量化推理、自动计算调度优化、异构编译执行等技术，来支撑本地 AI 功能开发。基于以上技术可设计语音识别、自然语言处理、图像识别、文本分类、地图导航、搜索引擎等 AI 模块引擎，支持 AR 导航、盲区检测，驾驶员疲劳检测，FaceID，手势识别、语音图像理解等人机交互 AI 应用，满足这些应用的各类性能指标要求。

8 接口技术要求建议

8.1 面向硬件的接口

面向硬件的接口主要有面向 SoC 的硬件接口、面向整车信号的接口，面向外围 IC 的接口以及面向 IoT 设备的互联接口等。面向硬件的接口将操作系统面向车规级芯片的接口标准化，使得操作系统在调用硬件资源时，不需要了解硬件的内部逻辑及资源分配情况，只需要按照标准的接口进行参数配置即可实现对硬件的调用，从而实现对于不同硬件的适配，保证系统较高移植性。其中，多核异构架构适配性技术要求包括异构平台硬件体系的融合驱动和异构平台并行计算框架。

8.2 面向应用的接口

面向应用程序的接口主要有 UI 控件、图形图像处理、多媒体、网络、蓝牙、摄像头、音频、机器学习、位置、车身信号、本地数据库、应用管理、系统等。面向应用程序的接口使得应用程序在调用操作系统的基础服务时，不需要了解其内部逻辑，只需要按照统一的接口进行参数配置即可

实现应用程序与操作系统的交互，从而进一步实现应用程序与操作系统的解耦，促进应用生态发展。

9 安全技术要求建议

9.1 信息安全

鉴于车载操作系统内核功能的复杂性，车载操作系统应构建完整的信息安全防护机制，降低内核漏洞对系统安全的危险，强化内核的安全防护能力。系统应具备多域隔离、一致性检查、安全启动、安全存储、安全输入、加密文件系统等技术，应具备系统镜像完整性及合法性验证机制、系统镜像防回退校验功能。

a) 内核完整性保护：可分为静态完整性保护和运行时完整性保护，确保内核不被篡改或在篡改后能够被迅速发现。

b) 数据隐私及加密：根据车载FOTA特点提供密钥方案和文件级加密方案，提高不同量产车辆产品的安全性，提升本机对data分区不同文件的加密保护。

c) 安全隔离机制：提供空间隔离机制，将内核与外部的驱动模块进行安全隔离，阻断攻击者从物理存储中获取安全容器内的文件内容。

d) 安全启动机制：源于硬件可信根的校验链，逐级校验下一阶段代码和文件系统的完整性，防止系统运行的代码被恶意篡改。

e) 安全存储机制：密钥管理模块与硬件安全能力结合，进一步提升文件和密钥存储的安全性。

9.2 功能安全

车载操作系统从应用角度分为中控系统、仪表系统和 T-box 系统等，应确保车载操作系统满足 ISO26262 安全标准，不同应用车载操作系统的功能安全等级要求不同，如仪表需满足 ASIL-B 的安全等级要求。

针对仪表等功能安全等级要求较高的系统软件，为防止系统内部多层级、多模块的单点软件失效等软件故障或瞬时硬件故障造成严重后果，可对系统软件进行冗余设计，来确保系统功能的稳定可靠。

在系统软件冗余设计中，关键任务处理时，可使用两种不同的冗余通道软件算法来执行相同的任务，比较两种冗余通道的输出结果，如发现差异，则生成故障消息。再者，也可在时间维度上以冗余的方式执行安全关键任务，使用同一软件算法（限定相同执行条件）在一定时间内上执行多次同样任务，比较多次执行结果，如出现差异，则执行相关的纠正操作，消除瞬时故障。

10 总结及标准化项目建议

基于对车载操作系统技术发展现状和演进趋势的研究，本报告提出了车载操作系统基础服务、接口、车载操作系统安全以及车载操作系统多系统的技术标准化建议，具体标准化项目建议及必要性分析如表 2 所示。智能网联汽车的应用正在高速发展期，车载操作系统技术作为智能网联汽车的支撑技术，具体的功能模块和接口也会随着系统的不断演进而演进，可以通过团标和行标标准化来进行，以适应快速变化的市场需求。

表 2 标准化项目建议

标准化项目建议	必要性	启动时间建议
车载操作系统技术要求及测试方法	通过规定车载操作系统的基本技术要求、功能要求和性能要求、功能安全要求、与硬件和应用程序接口的兼容性及测试方法，保证产品的质量。目前业界已有相应实践积累，建议启动国标。	优先级低
车载操作系统信息安全要求及测试方法	信息安全要求和测试方法是保证智能网联汽车的网络安全和数据安全前提，建议尽快启动国标。	优先级高