

1040604 OTA车端软件详细设计

| | | | |
|-----|-----------|--|------|
| 版本号 | | | |
| | 2024-7-10 | | 内容新增 |
| | 2024-7-16 | | 终版定版 |
| | | | |
| | | | |
| | | | |

- 1 目的(Objective)
- 2 范围(Scope)
- 3 缩写与定义 (Abbreviations and definitions)
 - 3.1 术场景补充说明语与缩略语描述
 - 3.2 名词缩写与定义描述
 - 3.3 文档与接口编号及命名规则
 - 3.4 预期读者 (Intended Readers)
- 4 架构阐述(Architecture Description)
 - 4.1 设计思想
 - 4.2 系统总体架构
 - 4.3 软件架构图
 - 4.3.1 车端软件分层
 - 4.3.2 子系统关系
 - 4.4 约束
 - 4.4.1 开发集成环境
 - 4.4.1 存储空间要求(Storge space requirements)
 - 4.4.2 性能约束(Performance constraints)
 - 4.4.3 软件约束(Code specification)
- 5 场景补充说明(Scenario description)
- 6 参照文件(Reference)

1 目的(Objective)

本文档是 整车OTA项目的软件详细设计指南，涵盖了UCM（升级主控模块）、Lite（升级从控模块）和Lite2hmi（升级从控模块至HMI系统）三个子系统。本设计指南致力于根据客户的具体需求，详细阐述每个子系统的技术实现方案。在整车OTA系统的需求和架构设计框架内，我们对子系统内部的各个模块进行了深入的实现层面的设计和说明（详见本章节下的子页面）。本文档旨在为开发团队提供清晰的指导方针，确保开发实现和测试工作能够顺利进行。同时，它也旨在为项目的持续优化和未来升级提供坚实的技术基础，以支持项目的长期发展和技术创新。

2 范围(Scope)

本章节专注于 整车OTA项目中的核心子系统——UCM、Lite及Lite2hmi的详细设计。我们详尽地介绍了各子系统的模块和组件，包括它们的功能概述、接口定义、架构布局以及时序设计。本文档专为以下专业人士设计：产品与解决方案策划者、软件开发工程师以及质量保证测试人员。我们的目标是提供必要的技术深度和指导，以确保项目从概念构思到最终实现的每个阶段都能得到精确的执行和严格的验证。通过本文档，读者将获得对项目技术层面的深刻理解，从而在开发和测试过程中做出明智的决策。

3 缩写与定义 (Abbreviations and definitions)

3.1 术场景补充说明语与缩略语描述

下列术语和缩略语适用于本文件，术语和缩略语描述如下所示：

| 缩写(Abbreviation) | 英文全称(Full Name) | 中文描述 (Chinese description) |
|------------------|-------------------------|----------------------------|
| OTA | Over-the-Air Technology | 远程无线升级技术 |
| FOTA | Firmware Over-The-Air | 固件空中升级技术 |

| | | |
|------------|-------------------------------|--|
| DOTA | Diagnostic Over-The-Air | 云诊断 |
| SOTA | Software Over-The-Air | 软件升级 |
| ECU | Electronic Control Unit | 电子控制器单元 |
| TSP | Telematics Service Provider | 远程通信服务供应商 |
| HMI | Human Machine Interface | 人机交互接口 |
| FBL | Flash Bootloader | 闪存引导加载程序 |
| CDN | Content Delivery Network | 内容分发网络服务器系统模块，用于管理升级包 |
| OTA Server | Over the air server | OTA管理平台，负责车辆的版本管理，任务发布，升级结果展示等 |
| MCU | Micro controller Unit | 微控制单元，作为单独升级模块 |
| APK | Android application package | OTA 人机交互Android应用程序包 |
| UCM | Update Control Master | OTA控制主模块，用来负责OTA升级主流程控制管理 |
| Lite | Update Control Lite | OTA 控制从模块，用来负责提供OTA升级所需的关键能力支持（如文件下载，车辆状态信息获取等）。 |
| Lite2hmi | Update Control Lite to HMI | OTA 控制从模块，用来负责对接HMI界面展现相关内容的交互 |
| UA | Updata Agent | 艾拉比差分升级模块，负责SOC的升级 |
| SOA | Service-Oriented Architecture | 面向服务的架构 |

注：

OTA-Master，完全等同于FOTA-Master，UC-Master，本文档范围内统一简称为UCM。

OTA-SubMaster，完全等同于FOTA-Lite，UC-Lite，本文档范围内统一简称为Lite。

Diagnostic Engine，完全等同于Vehicle Diagnostic Manager，本文档范围内统一简称为VDM。

3.2 名词缩写与定义描述

下列为本文各模块术语和缩略语，术语和缩略语描述如下所示：

| 缩写(Abbreviation) | 英文全称(Full Name) | 中文描述 (Chinese description) |
|------------------|----------------------------|----------------------------|
| XTM | XOTA Trigger Manager | 触发管理模块 |
| ECM | ECUs Config Manager | 零件配置管理模块 |
| EIM | ECUs Information Manager | 零件信息管理模块 |
| CVM | Version Check Manager | 版本检测管理模块 |
| DPM | Download Policy Manager | 下载策略管理模块 |
| TPM (预留) | Transfer Policy Manager | 传输策略管理模块 |
| SUM | Seamless Update Manager | 无感更新管理模块 |
| IPM | Installer Policy Manager | 安装策略管理模块 |
| CDM | Condition Analysis Manager | 条件分析管理模块 |
| LGM | Log Manager | 日志信息管理模块 |
| RCM | Reporting Cloud Manager | 上报云信息管理模块 |
| OCM | OTA-Client Manager | 车云客户端模块 |

| | | |
|-------|------------------------------------|-----------|
| UIM | User Interaction Manager | 人机交互管理模块 |
| DIM | Device Interface Manager | 设备信息管理模块 |
| VSM | Vehicle State Manager | 车辆状态服务模块 |
| VDM | Vehicle Diagnostic Manager | 车辆诊断服务模块 |
| UAM | Update Agent Manager | 更新代理服务模块 |
| DTM | Delegate Transfer Manager | 委托传输服务模块 |
| SM | Security Manager | 安全服务模块 |
| D2B | Device to Business Manager | 设备对业务服务模块 |
| MISC | Miscellaneous Manager | 杂项服务模块 |
| VSMA | Vehicle State Manager Adapter | 车辆状态适配模块 |
| VDMA | Vehicle Diagnostic Manager Adapter | 车辆诊断适配模块 |
| UAMA | Update Agent Manager Adapter | 更新代理适配模块 |
| TCA | Transport Channel Adatppter | 传输通道适配模块 |
| SMA | Security Manager Adapter | 安全适配模块 |
| D2BA | Device to Business Adapter | 设备对业务适配模块 |
| MISCA | Miscellaneous Adapter | 杂项适配模块 |

3.3 文档与接口编号及命名规则

编号命名规则旨在为事物或实体分配唯一标识，以便于组织、管理和辨识。本架构文档与接口编号规则约束如下：

- **文档编号规则：**
 - 文档编号应遵循以下格式：《项目_软件元素_车端软件详细设计文档》。
 - 例如：XXXX整车OTA咨询项目，如果文档描述的是整车UCM的软件详细设计，则文档编号为《一汽大众整车OTA咨询项目_UCM_车端软件详细设计文档》。
- **接口编号规则：**
 - 接口编号应遵循以下格式：SWE3_软件元素_一级组件名称_二级组件编号，其中组件编号默认从0001开始（保留四位有效数字）。
 - 例如：对于XXXX整车OTA咨询项目，如果接口属于UCM的一级组件（车云客户端模块，OCM），并且是该组件的第一个二级组件（车辆注册接口），则接口编号为：SWE3_UCM_OCM_0001。
- **服务接口分类命名规则：**

根据AutoSRA AP的SOA架构约定，SOA平台上服务之间的通信接口有Event、Method和Field三种形式，服务接口用于定义Event/Method/Field消息类型和具体的命名空间，与具体的通信协议无关。结合OTA的交互场景（本项目暂未使用Field类型），本项目我们约定每种服务的命名原则如下：

- **Event**

Event接口为事件通知，表示实际传输的数据，以数据为操作对象。命名规范遵循基础规范，同时为了避免接口命名的冲突，这里区分事件是来自UIM还是DIM，接口命名后缀如下：

- DIM 侧：
 - 服务端：以Evt（Event）结尾。
 - 消费端：以EvtHdl（Event Handle）结尾。
- UIM侧：
 - 服务端：以Evt2HMI（Event to HMI）结尾。
 - 消费端：以Evt2HMIHdl（Event to HMI Handle）结尾。

- **Method**

Method接口表示某种控制，如状态查询、车辆控制等。接口命名需要表达清楚该方法的含义，同时为了避免接口命名冲突，这里区分方法是来自UIM还是DIM，接口命名后缀如下：

- DIM侧：
服务端：以MthDHdl (Method DIM Handle) 结尾。
消费端：以MthDProxy (Method DIM Proxy) 结尾。
- UIM侧：
服务端：以MthUHdl (Method UIM Handle) 结尾。
消费端：以MthUProxy (Method UIM Proxy) 结尾。

通过这样的编号规则，可以确保每个文档和接口都有一个唯一且描述性的标识符，便于项目的组织、管理和辨识。同时，这也有助于在项目团队成员之间提供清晰的沟通基础，以及在项目的整个生命周期中维护文档和接口的一致性和双向追溯。

接口命名冲突示例：

如：UIM处理来自UCM的下载进度通知，会与UCM处理Lite的下载进度通知，在接口命名重复，所以在接口中使用后缀Evt/Evt2HMI 来区分。

同理：UCM处理来自UIM的安装请求，会与Lite处理UCM的安装请求，在接口命名上重复，所以，在接口中使用后缀MthUHdl/MthDHdl来区分。

3.4 预期读者 (Intended Readers)

- 解决方案工程师
- 项目经理
- 研发工程师
- FAE工程师
- 测试工程师

4 架构阐述(Architecture Description)

4.1 设计思想

在本项目中，我们的设计思想是构建一个灵活、可扩展且高效的软件系统，以适应不断变化的业务需求和技术环境。为了实现这一目标，我们采纳了分层架构设计方法，相较于传统的三层架构（界面层、业务逻辑层和数据访问层），我们根据OTA软件的复用性考虑，将整个系统从顶层到底层划分为四个主要层次：

- **应用层(Application layer)**：应用层是系统的顶层，其主要职责是集成和协调FOTA应用的各项功能。在这一层，我们将整合和调度底层组件层所提供的服务，确保OTA应用程序能够无缝执行其核心功能，同时提供用户友好的交互界面。
- **组件层(Component layer)**：组件层致力于为OTA业务提供一系列标准化的组件实现。这些组件包括车辆状态控制、车辆诊断刷写、文件传输等关键服务。我们的目标是最大化组件的复用性，以便在不同的应用场景中快速部署和扩展。
- **基础软件层(Basic software layer)**：基础软件层为OTA平台提供了坚实的基础软件平台。在这一层，我们将实现事件处理、进程间通信和信息埋点等基础软件功能，为开发人员提供一个高效的系统级实现框架，从而加速上层应用场景的构建。
- **平台适配层(Foundation layer)**：平台适配层是系统架构的底层，它为OTA平台提供了可移植的操作系统接口，以及常用功能和工具的抽象实现。这一层确保了系统的跨平台兼容性，为上层架构提供了稳定和统一的运行环境。

分层架构的核心宗旨在于达成“高内聚低耦合”的设计理想，并促进组件的高效复用。在这一架构中，我们精心构建了四个层次，以确保系统的主要功能和业务逻辑得到恰当的处理和优化。

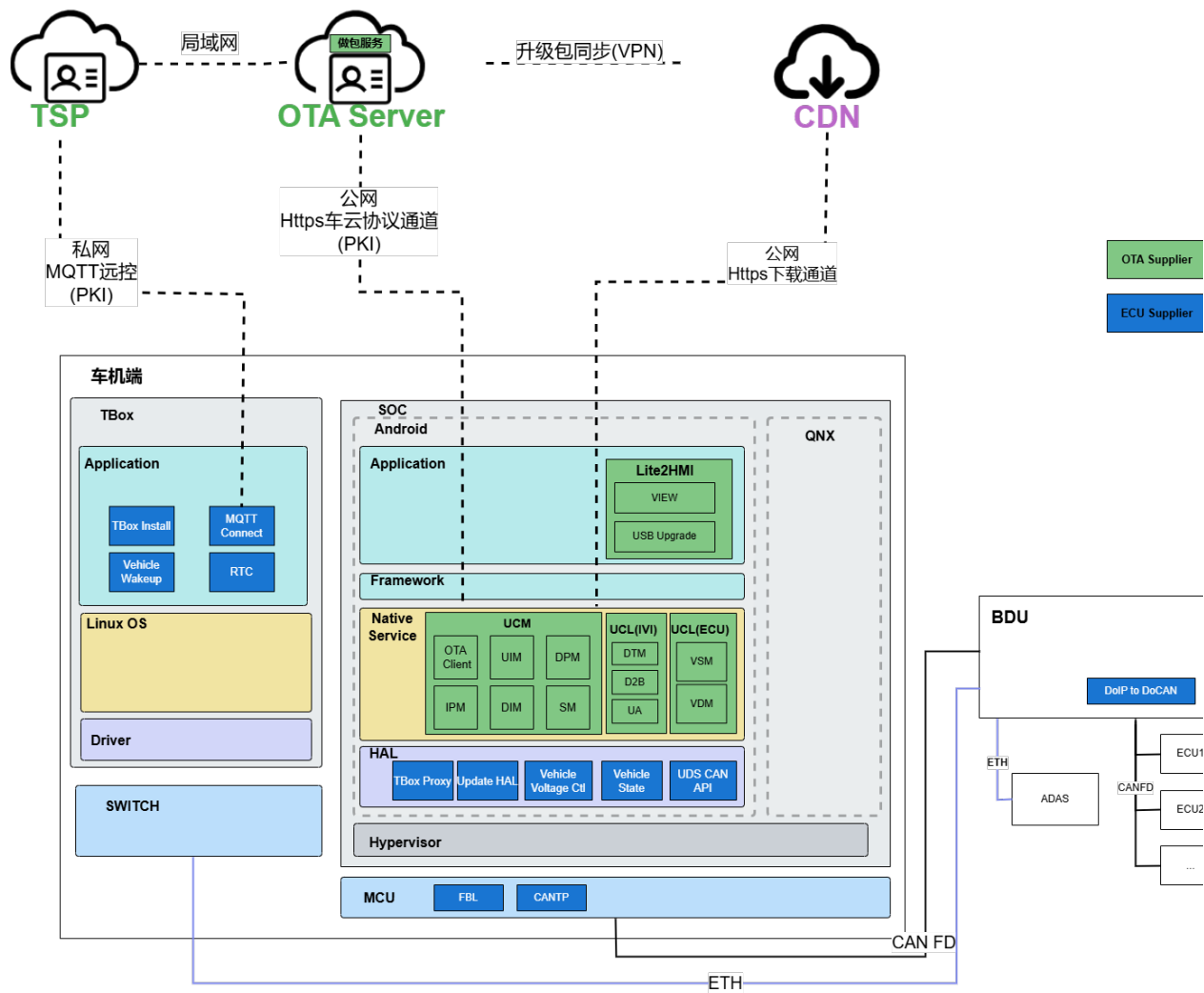
分层架构的原则是基于将系统分解为多个独立但相互协作的层次，以提高软件的可维护性、可扩展性和复用性。以下是分层架构设计的一些核心原则：

- **关注点分离**：每个层次应该专注于其特定的职责，避免跨层次的职责交叉。这有助于减少复杂性，使得每个层更容易理解和维护。
- **模块化**：系统应该被划分为多个独立的模块或组件，每个模块都有明确的接口。这样可以提高代码的复用性，便于替换和升级。
- **松耦合**：层次之间的依赖关系应该尽可能地弱化。这意味着一个层次的变化不应该对其他层次产生太大影响。
- **高内聚**：每一层都应该向上层提供清晰的抽象，隐藏下层的实现细节。这有助于上层开发者专注于业务逻辑，而不必关心底层的具体实现。
- **标准化接口**：层次之间的通信应该通过定义良好的接口进行。这有助于确保不同层次之间的兼容性和一致性。
- **可扩展性**：架构应该设计得足够灵活，以支持未来的扩展。这包括水平扩展（同层内增加更多的组件/模块）和垂直扩展（层与层之间允许扩充层）。
- **可测试性**：每一层都应该独立可测试，以便于进行单元测试和集成测试。这有助于提高软件质量，减少缺陷。
- **安全性**：安全性应该在所有层次中得到考虑和实施，确保数据和系统的安全。
- **性能优化**：在设计时考虑性能需求，确保系统在满足功能的同时，也能提供良好的性能。

除此之外，本架构中还约定：“同一层中，不同模块间不允许直接接口调用，必须采用消息或者基础软件层中的数据库接口交互。不同层中，不推荐上层直接调用下层的接口，应尽量采用消息、基础软件层和平台适配层的接口交互”。

4.2 系统总体架构

下图主要描述该项目车端架构图：



系统中各个模块功能描述如下：

| 名称 | 描述 | 实现方 |
|-----|--|-----|
| UCM | OTA主控程序主要负责控制业务流程;车云交互，处理外部刺激，收集零件信息,解析和控制下载策略和安装策略 | 艾拉比 |
| UCL | 主要负责响应UCM控制指令，负责收集该域下的零件交互完成零件信息收集，升级包下载，升级能力调用等业务完成该域下零件OTA业务 | 艾拉比 |
| HMI | 负责人机交互接受用户指令(检测，下载，立即安装，预约安装等)，显示OTA业务流程 | 艾拉比 |
| VDM | 实现车辆总线数据收发、UDS(ISO14229)诊断数据收发、UDS刷写数据收发，定制化脚本解析及执行的功能模块 | 艾拉比 |

| | | |
|-------------|--|-----|
| DoIP协议栈 | Diagnostic Communication Over Internet Protocol基于以太网协议的诊断通信 | 艾拉比 |
| DTM | DTM委托传输服务(Delegate Transfer Manager)是服务组件中的一个关键模块，旨在提供高效可靠的传输能力(车内文件下载、车云文件下载和车内文件上传等) | 艾拉比 |
| UA | 差分升级模块，负责车机soc的升级 | 艾拉比 |
| Integration | 域内更新代理该项目负责管理了IC，DMS，TBOX等下挂件升级和控制了车机升级（UA：刷写非当前区域+Active：激活） | 艾拉比 |
| D2B | 设备提供的服务功能(Device to Business)，OTA系统中有些业务需要UCM宿主ECU设备提供接口功能，OTA系统把这类接口功能归类到D2B服务模块，由D2B服务模块统一给OTA业务提供服务 | 艾拉比 |
| SM | 安全模块，与KPI对接完成文件解密验签等安全相关功能 | 艾拉比 |

4.3 软件架构图

4.3.1 车端软件分层

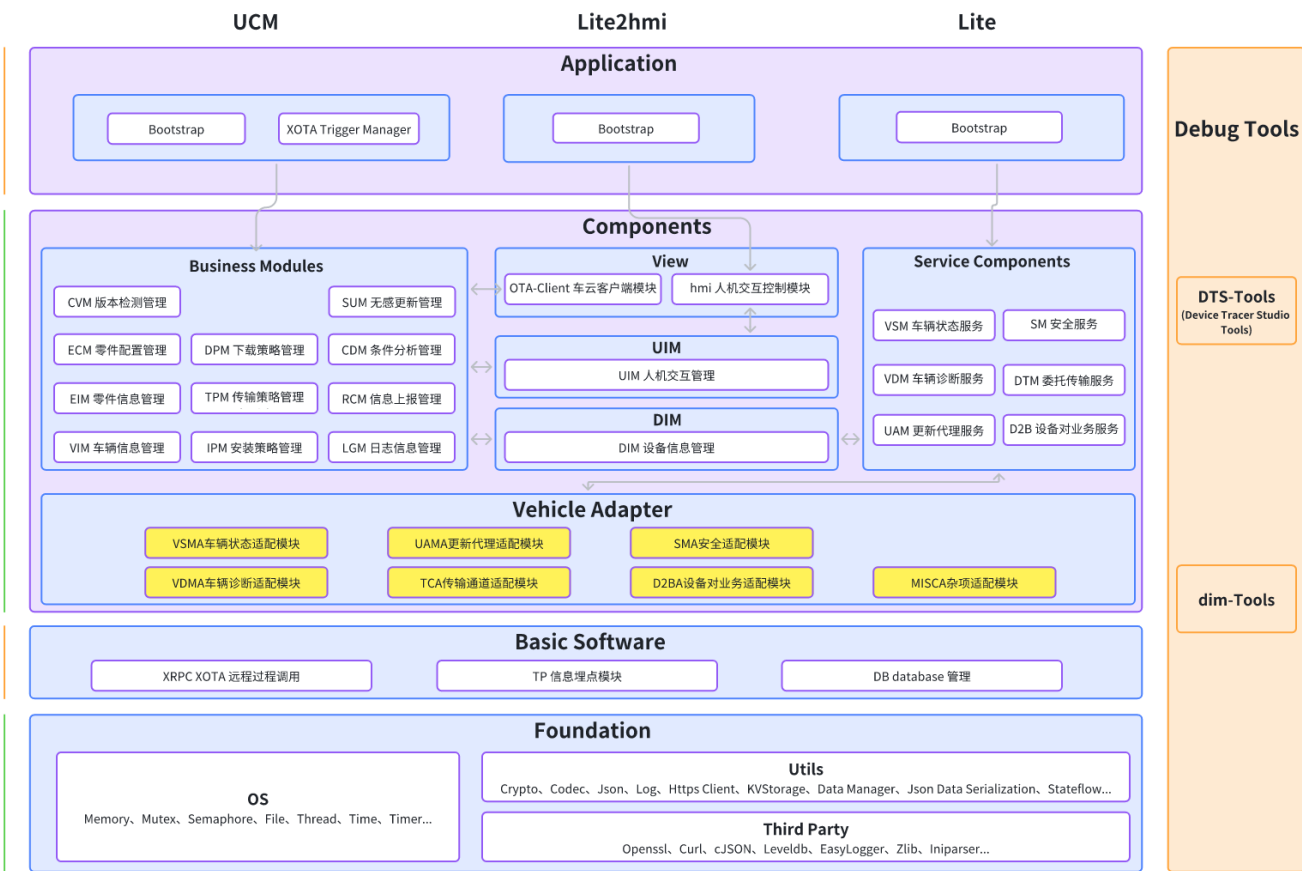





图 车端软件分层

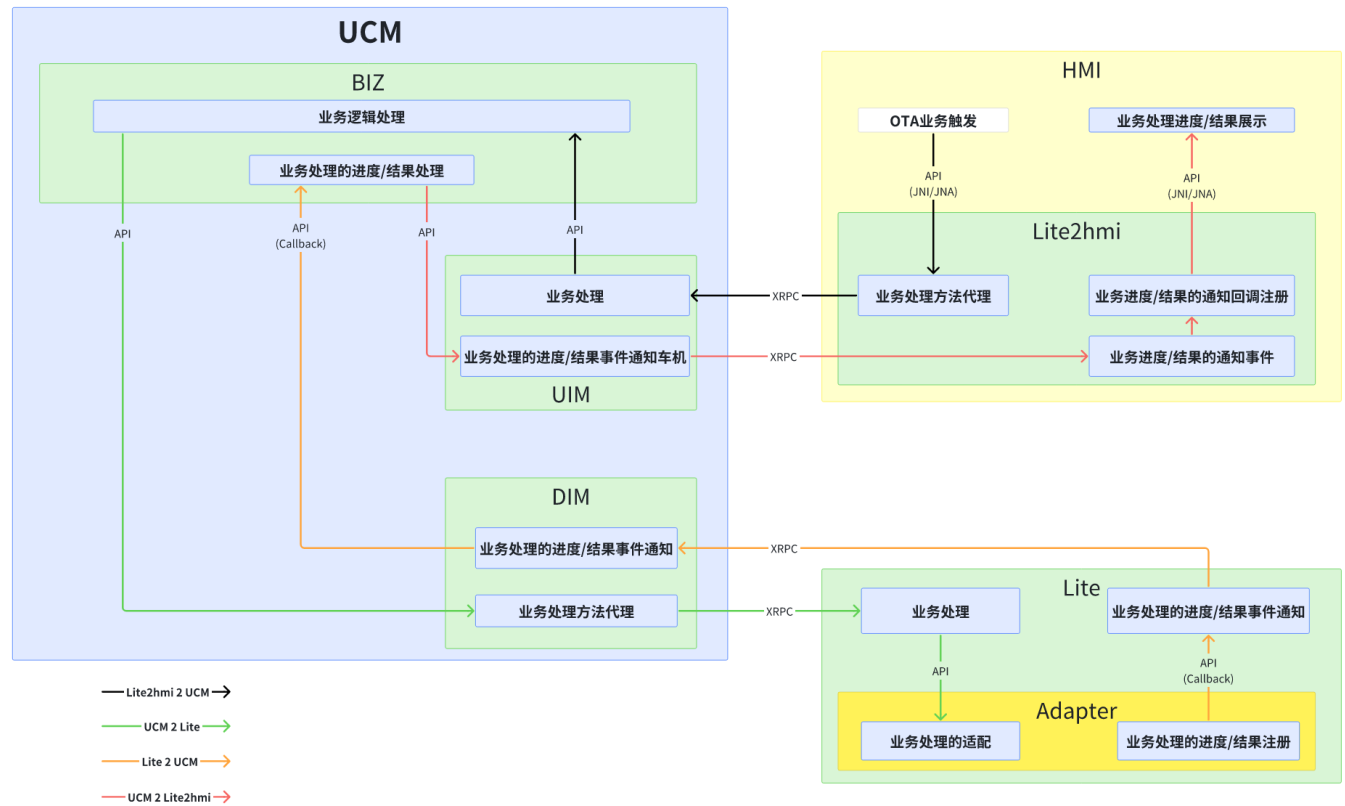
注：

整车中UCM子系统的软件分层示意图参照 当前页面的UCM软件详细设计子页面的第4章节部分（）

整车中Lite子系统的软件分层示意图参照 当前页面的Lite软件详细设计子页面的第4章节部分（）

整车中Lite2hmi子系统的软件分层示意图参照 当前页面的Lite2hmi软件详细设计子页面的第4章节部分（）

4.3.2 子系统关系



4.4 约束

4.4.1 开发集成环境

基于本架构设计文档，OTA支持的操作系统环境描述如下：

| 操作系统 | 版本号 |
|---------|------------|
| Linux | 3.4 |
| Android | M, N, O, P |
| QNX | 7.0 |

4.4.1 存储空间要求(Storage space requirements)

宿主ECU供应商需根据OTA业务需求，分配文件存储空间，用于缓存从OTA管理平台下载的升级文件。存储空间还需符合如下要求：

- 预留存储空间由ECU供应商、车厂和艾拉比根据实际业务情况共同协商定义。

- 建议有独立分区，该分区被OTA应用独占使用，用于存放升级包。
- 该存储空间具备一定的安全防护条件，确保数据库、配置文件、升级文件不被用户篡改，或者被第三方窃取。

4.4.2 性能约束(Performance constraints)

| 配置参数 | 参数要求 | 备注 |
|------|-------------|---------------|
| ROM | 20MB | 软件自身占用内存空间大小 |
| RAM | 20MB ~ 25MB | 软件运行时占用内存空间大小 |
| CPU | 6000DMIPS | |

4.4.3 软件约束(Code specification)

代码规范严格按照

- 《艾拉比C/C++编码规范 5》。
- Helix QAC MISRA-2012(c++)/2016(c) 规范。

5 场景补充说明(Scenario description)

在没有HMI（Lite2hmi）集成的OTA整体升级方案中，对主控系统的影响主要体现在以下几个宏观层面：

- **信息展示方式：**由于HMI通常负责OTA过程中的信息展示，缺少HMI意味着需要寻找其他方式来实现信息的可视化。例如，可以考虑通过OTA云端系统来展示OTA状态。然而，这种方式可能受到网络环境的限制，导致信息展示的实时性不如HMI直接展示。
- **任务触发机制：**HMI在OTA流程中还扮演着任务触发的关键角色，包括触发检查更新、下载和安装步骤。在没有HMI的情况下，需要考虑替代方案来触发这些任务。这可能涉及到使用后台静默任务执行或者TSP推送来实现OTA任务的触发。

在设计没有HMI集成的OTA方案时，需要综合考虑以上因素，确保OTA过程的流畅性和用户的良好体验。同时，也要考虑到系统的可靠性和安全性，确保OTA升级的稳定性和数据的保护。

6 参照文件(Reference)

| 序号 | 文件名 | 版本号 | 存储位置 |
|----|----------------|--------|------|
| 1 | 产品功能需求矩阵 | V1.6.0 | |
| 2 | 产品功能需求规格说明书 | | |
| 3 | 5.0 OTA平台接口设计书 | | |
| 4 | | | |
| 5 | | | |
| 6 | | | |