

01 OTA Client详细设计

- 1 文档介绍 Introduction
 - 1.1 范围说明 Scope
 - 1.2 参考文档 References
 - 1.3 读者对象 Target Audience
 - 1.4. 术语和缩写 Definitions and Abbreviations
- 2 架构设计 Architecture Design
- 3 静态设计 Static Design
 - 3.1 详细模块/组件框图 Detailed Module/Component Diagram (颗粒度L3~L4级)
 - 3.2 模块详细设计 Module Detailed Design (L4级别)
 - 3.2.1 OTA Client
- 4 动态设计 Dynamic Design
 - 4.1 业务流程图
 - 4.2 业务时序图
- 5. 数据库/数据结构设计 (可选)
- 6. 对外接口设计
 - 6.1 apl_service_init
 - 6.2 apl_service_run
 - 6.3 apl_service_stop
 - 6.4 apl_service_deinit
 - 6.5 obtain_vehicle_config_list
 - 6.6 obtain_vehicle_version
 - 6.7 notify_vehicle_status
 - 6.8 obtain_vehicle_task_status
 - 6.9 notify_vehicle_event
- 7. 单元测试
 - 7.1. Mock工具

Version / 版本	Date / 日期	Responsible / 编写人	Reviewer / 审核人	Review Date / 审核日期	Change / 修改内容

1 文档介绍 Introduction

1.1 范围说明 Scope

样例：This document mainly describes the software detailed design of Radio (Tuner). Radio provides functions such as FM/AM playing, searching, and favorite storing, and band pictures displaying, and so on. These functions are implemented by a number of different modules in cooperation with each other. For the specific modules involved in each function, as well as the interaction sequence and interaction interface, please refer to the following chapters for details.

1.2 参考文档 References

Ducment Name / 文档名称	Date / 日期	Version / 版本	Location / 位置
《车云交互设计》	2024-7-24	v1.0	

1.3 读者对象 Target Audience

系统工程师，开发人员，测试人员，技术支持人员，及其他相关人员。

1.4. 术语和缩写 Definitions and Abbreviations

本章节对本文中使用的术语定义和缩写进行解释。

Definitions or Abbreviations / 术语或缩写	Explanation / 解释

2 架构设计 Architecture Design

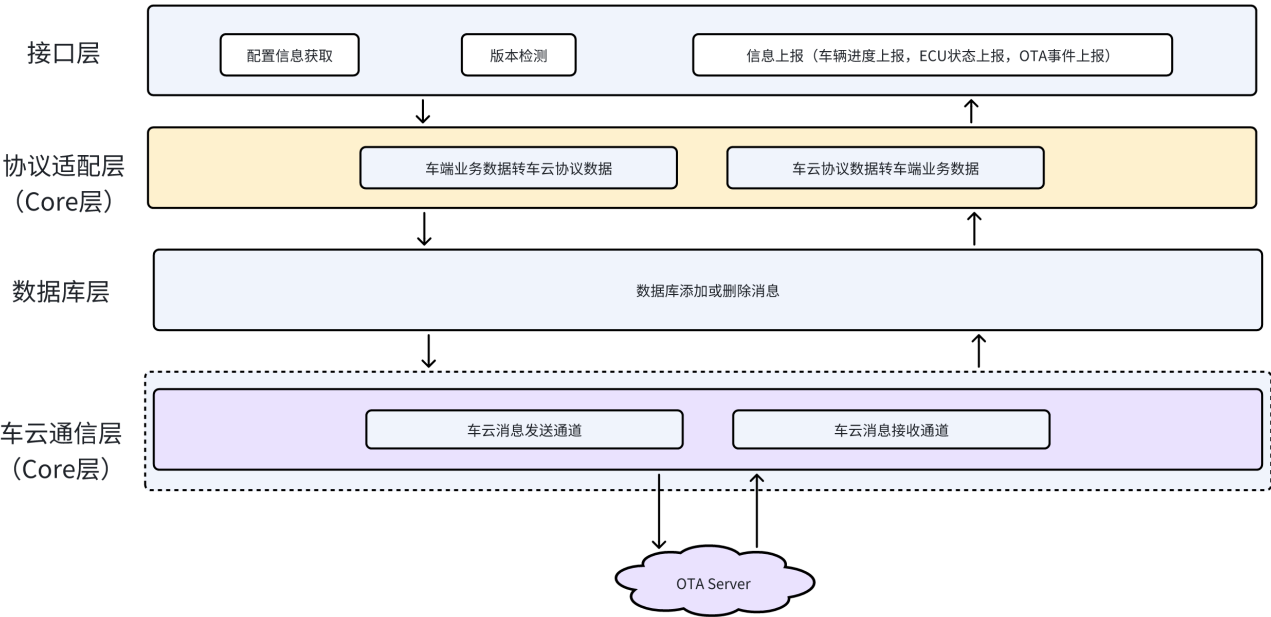
按照车云通信的需求，我们将OTA Client内部分为三层：

接口层：是OTA Client对外提供的功能接口，主要为业务提供车云通信接口。

Ccore层的协议适配层：用于车端业务数据与车云协议数据之间的相互转化，减少业务数据与车云协议的耦合性，从而提升系统的整体灵活性。

Ccore层的车云通信层：用于适配车云消息通信通道，主要负责消息的收发，同时保证车云通信的链路安全（支持TLS安全传输）。

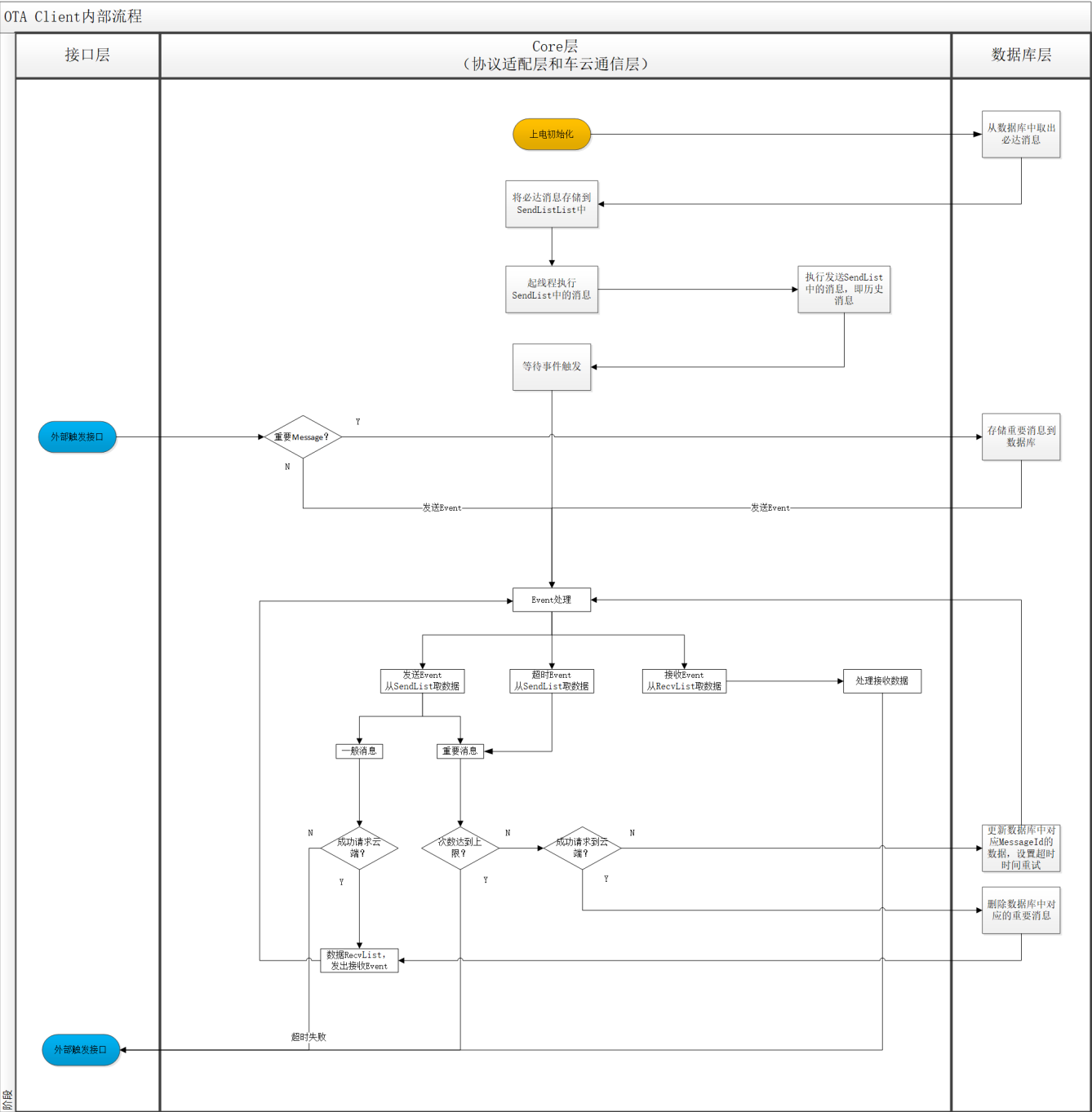
数据库层：对重要消息入库存储或删除数据库中的已消费的消息。



3 静态设计 Static Design

3.1 详细模块/组件框图 Detailed Module/Component Diagram (颗粒度L3~L4级)

详细设计模块总体框图L3-L4级，可作为《~~XXXXXXXXXX~~》中“模块/组件框图”细化（注：如果没有细化的必要，则可以直接使用概要设计中的模块/组件框图），从总体上表达软件内部模块结构以及层次关系。



3.2 模块详细设计 Module Detailed Design (L4级别)

3.2.1 OTA Client

概要设计模块链接:

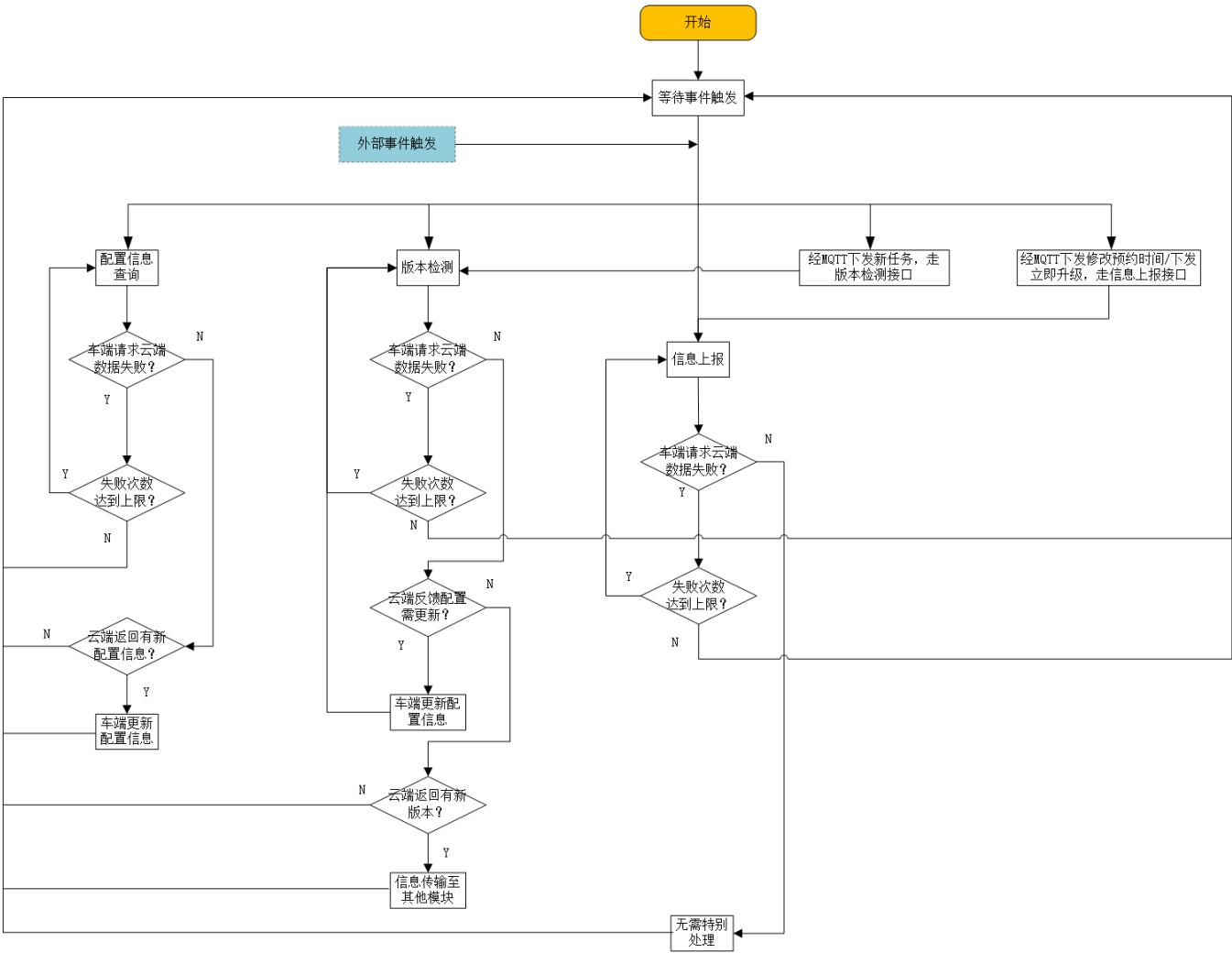
4 动态设计 Dynamic Design

4.1 业务流程图

OTA Client模块不仅简化了业务模块与OTA管理平台之间的通信流程，还确保了通信的安全性和可靠性。这种设计使得OTA系统能够更加稳定地运行，同时也为业务模块提供了一个安全、高效的通信环境。在本项目中，OTA Client提供了一系列的标准接口，这些接口支持FOTA的全流程管理。以下是OTA Client模块对外提供的具体接口及其用途：

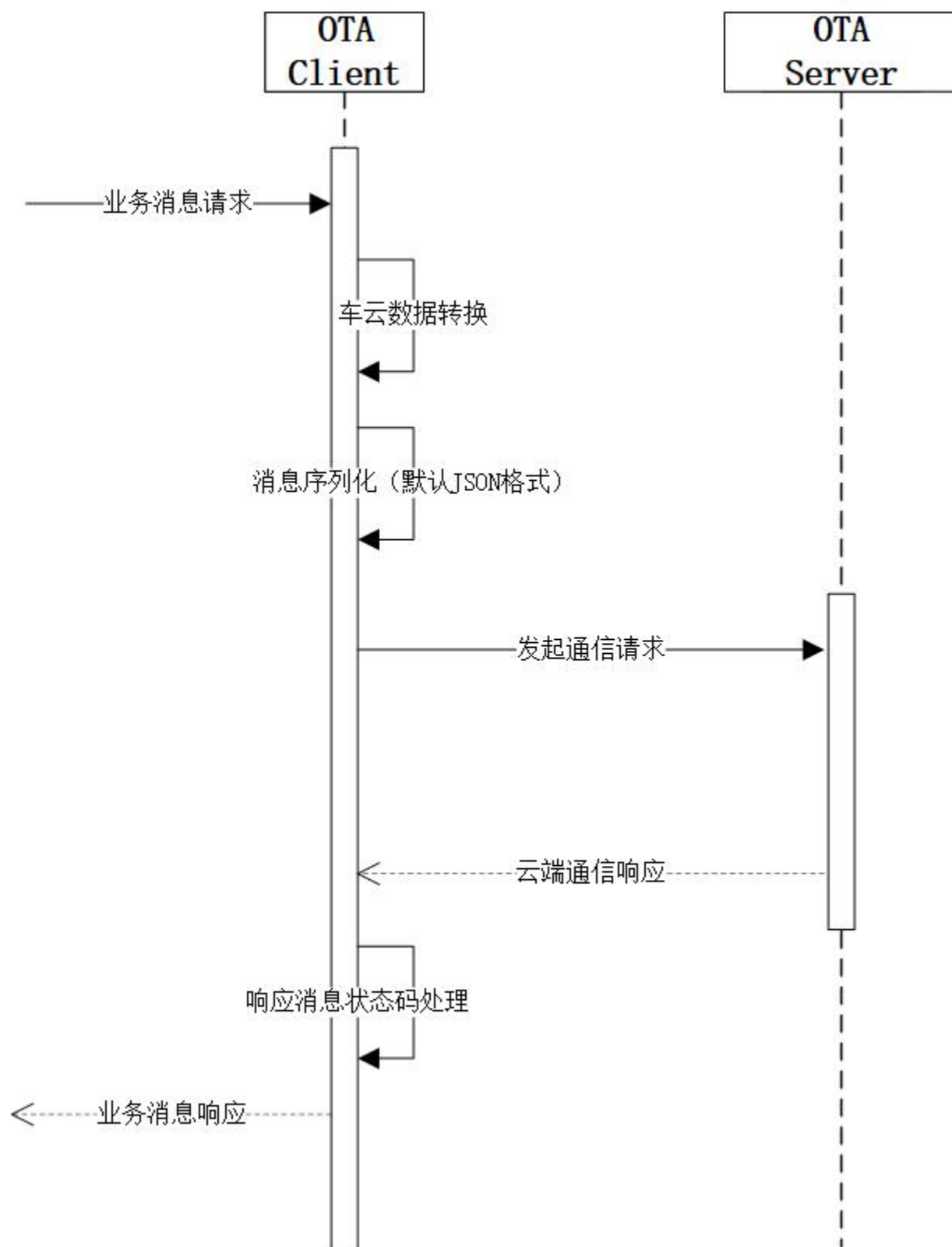
- 1. 配置信息获取：
 - 接口用于从OTA管理平台获取与车辆相关的零件配置信息。
 - 这些信息确保车辆中哪些零件被用于本次OTA升级。
- 2. 版本检测：
 - 接口用于向OTA管理平台发送版本检测请求。
 - OTA管理平台会检查携带的车辆当前的ECU信息，并确定哪些ECU需要被升级。
- 3. 车辆信息同步：
 - 接口用于将车辆的当前状态信息（如大版本号）同步到OTA管理平台。
 - 有助于OTA管理平台跟踪车辆的升级信息，并为后续的升级任务做准备。

不同的业务流程可能是基础接口的组合，整体业务流程图如下：



4.2 业务时序图

车云交互是阻塞式的HTTP请求，交互链路采用双向认证方式，OTA Client会做消息的序列化及反序列化操作。其他业务消息请求给到OTA Client之后，OTA Client结合车云协议所需数据，会组装成标准车云协议数据格式。



5. 数据库/数据结构设计（可选）

全局的（不属于某个模块的）设计数据库表结构或者数据结构，数据库请用ER图表示

当车云通信的时候，可能会出现断电的情况，因此需要将车云通信的数据做落盘存储，当发送成功或发送失败达到指定的次数，可将待发送的消息清除。

创建的数据表是apl_msg，表的字段及类型如下所示

字段	id	retry_cnt	arriver_rty_cnt	state	flag	len	expiration	lastSendTime	msgId	msgdata
类型	INTEGER PRIMARY KEY AUTOINCREMENT UNIQUE	INTEGER	INTEGER	INTEGER	INTEGER	INTEGER	INTEGER	INTEGER	VARCHAR (20)	TEXT
含义	自增的PRIMARY KEY	消息重试次数，无论此次是否发送成功	必达消息重试次数，当消息重试且不成功的时候，此值才增加	状态（保留）	表示消息的重要性，如必达消息	msgdata的长度	过期时间（保留）	消息最后发送的时间	消息Id	车云消息

6. 对外接口设计

OTA Client对云端的接口有获取获取配置信息、版本检测、信息上报等接口，OTA Client能提供的对外接口和具体的业务有关。对外接口如下：

6.1 apl_service_init

函数	int apl_service_init(INIT_PHASE phase);	
同步/异步	同步	
可重入	不可重入	
参数 (in)	初始化时传入的参数	typedef enum { INIT_PHASE_INSIDE=0, INIT_PHASE_RESTORE, INIT_PHASE_OUTSIDE, INIT_PHASE_COUNT, }
参数 (inout)	无	
返回值	int	0: 成功. 其他: 失败.
描述	服务初始化	

6.2 apl_service_run

函数	int conf_service_run(void);	
同步/异步	同步	
可重入	不可重入	
参数 (in)	无	
参数 (inout)	无	
返回值	int	0: 成功. 其他: 失败.
描述	服务运行	

6.3 apl_service_stop

函数	int apl_service_stop(void);	
同步/异步	同步	

可重入	不可重入	
参数 (in)	无	
参数 (inout)	无	
返回值	int	0: 成功. 其他: 失败.
描述	服务停止	

6.4 apl_service_deinit

函数	int apl_service_deinit(void);	
同步/异步	同步	
可重入	不可重入	
参数 (in)	无	
参数 (inout)	无	
返回值	int	0: 成功. 其他: 失败.
描述	服务去初始化	

6.5 obtain_vehicle_config_list

函数	int obtain_vehicle_config_list(json_object **M1_RES);	
同步/异步	同步	
可重入	可重入	
参数 (inout)	M1_RES	返回的物流清单信息和OTA全局配置信息
返回值	int	0: 成功. 其他: 失败.
描述	获取物流清单信息，请求内容是Common字段	

6.6 obtain_vehicle_version

函数	int obtain_vehicle_version(json_object *M2_REQ,json_object **M2_RES);	
同步/异步	同步	
可重入	可重入	
参数 (in)	M2_REQ	ECU版本信息
参数 (inout)	M2_RES	升级指导文件信息
返回值	int	0: 成功. 其他: 失败.
描述	车辆搜集自身的车辆版本信息，然后上报到云端，请求查看是否有新版本	

6.7 notify_vehicle_status

函数	int notify_vehicle_status(json_object *M3_REQ);	
同步/异步	同步	
可重入	可重入	

参数 (in)	M3_REQ	符合车云协议的状态上报数据
返回值	int	0: 成功. 其他: 失败.
描述	请求上报车辆状态信息	

6.8 obtain_vehicle_task_status

函数	int obtain_vehicle_task_status(json_object *M4_REQ, json_object **M4_RES);	
同步/异步	同步	
可重入	可重入	
参数 (in)	M4_REQ	查看任务状态的请求
参数 (out)	M4_RES	查看任务状态的响应
返回值	int	0: 成功. 其他: 失败.
描述	查看任务状态信息	

6.9 notify_vehicle_event

函数	int notify_vehicle_event(json_object *M5_REQ);	
同步/异步	同步	
可重入	可重入	
参数 (in)	M5_REQ	符合车云协议的事件上报数据
参数 (out)	None	
返回值	int	0: 成功. 其他: 失败.
描述	请求上报车辆事件信息	

7. 单元测试

7.1. Mock工具

在某些情况下，不容易产生某些数据或状态。为了简化测试，采用Mock工具进行单元测试，当前在仓库中有单元测试的代码。

Gerrit代码地址是

```
ssh://yinyongbo@gerrit.vc.in:31250/hu/android/faw-vw/j01/fota/service/ota_client
```

测试步骤如下：

Step1：Cmake编译测试程序

```
cd ota_client/unittest/  
mkdir build  
cd build  
cmake ../  
make  
cp ../crt/* . -rf //
```

Step2：准备API请求所需的JSON文件

依次准备api2.json、api3.json、api4.json和api5.json文件，作为ota_client的输入

Step3: 开始测试

```
./ota_client_test
```