

02 Master App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)
- 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
- 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

OTA Master App承担启动，运行环境准备和持久化数据加载、ECU信息数据存储和更新、OTA Manager模块初始化、IVI UA模块初始化、ECU UA模块初始化、Log搜集和打包管理（Log Service）、配置文件管理（Config Service）、Download Agent模块初始化、等功能。

1.2 参考文档 References

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。

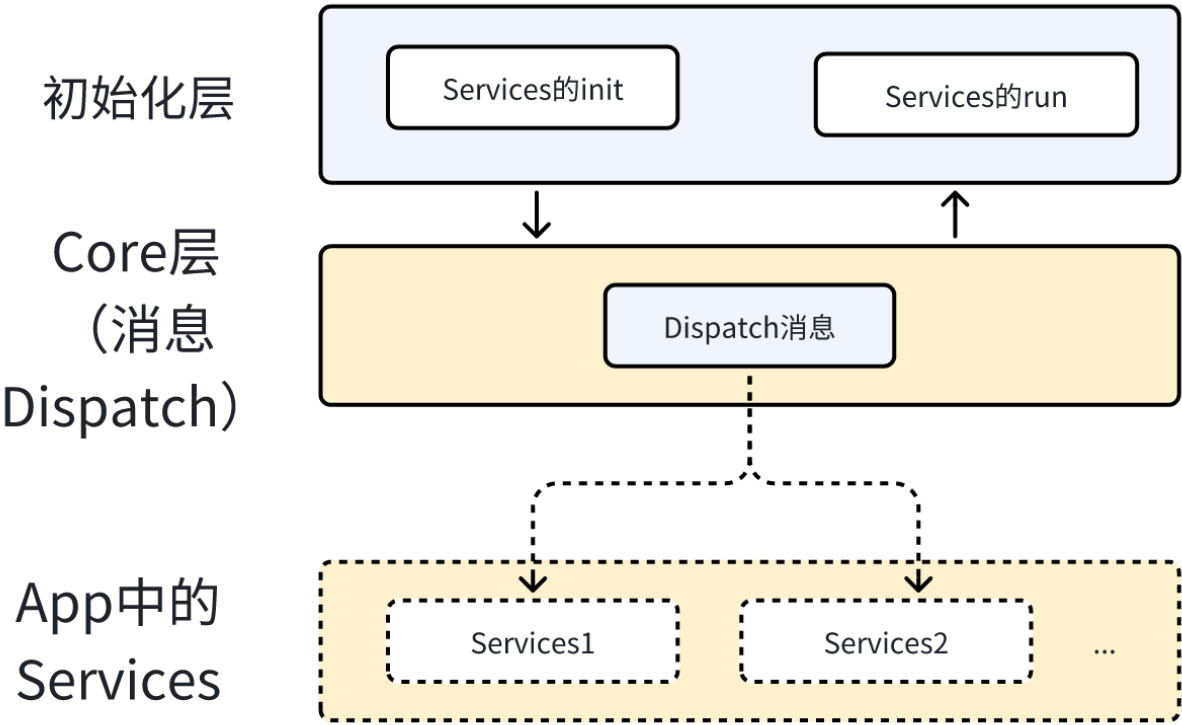
1.4. 术语和缩写 Definitions and Abbreviations

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

Master App的作用类似串珠子，将各个Service串起来组成完成完整的OTA业务。



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《》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。

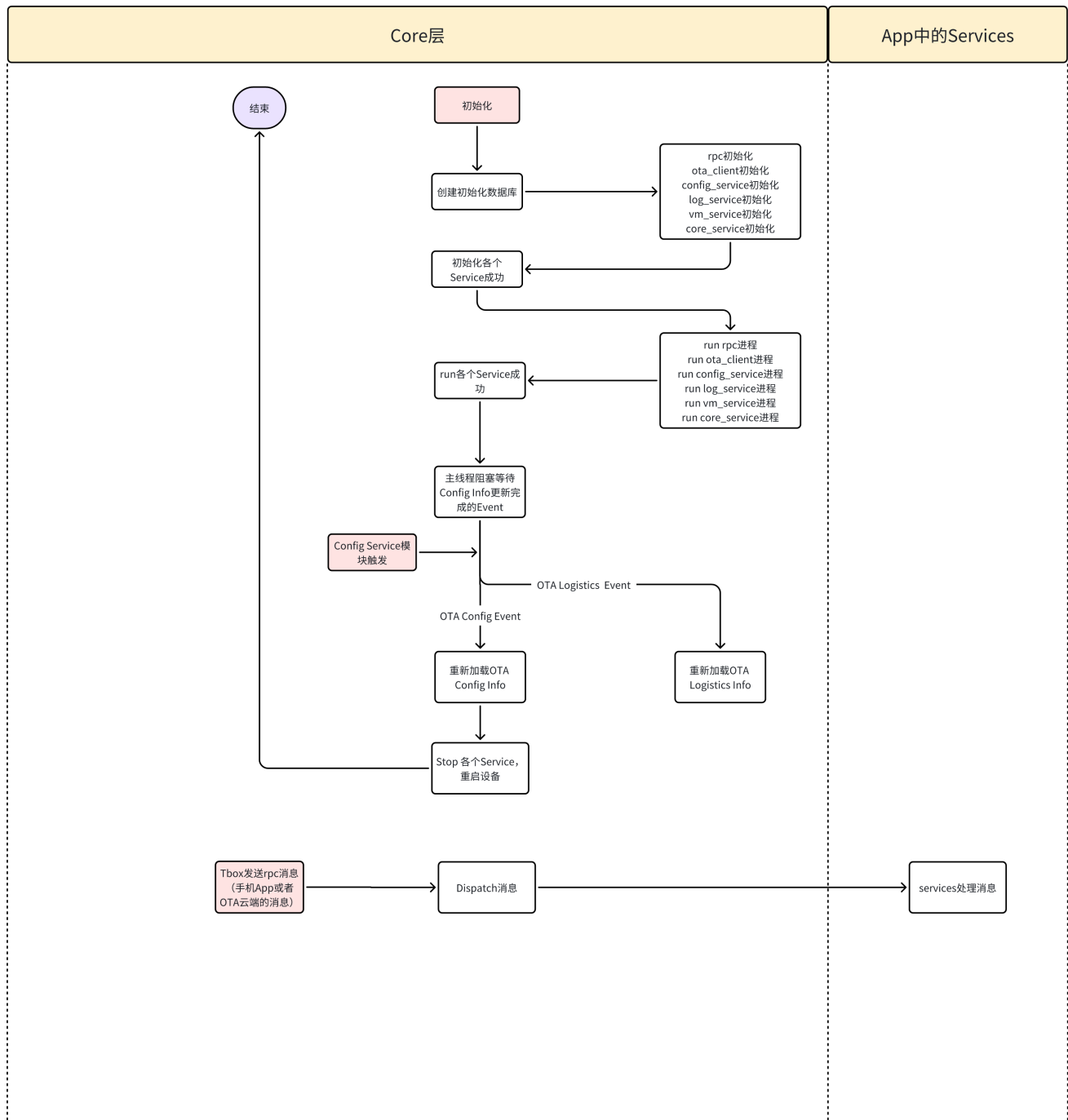
3.2 模块详细设计 Module Detailed Design（L4级别）

概要设计模块链接：

4 动态设计 Dynamic Design

4.1 业务流程图

Master App中包裹着各个Service。Master App会init并run这个services。从手机App或者从云端来的消息会先到Master App中，再由Master App分发到各个Services。



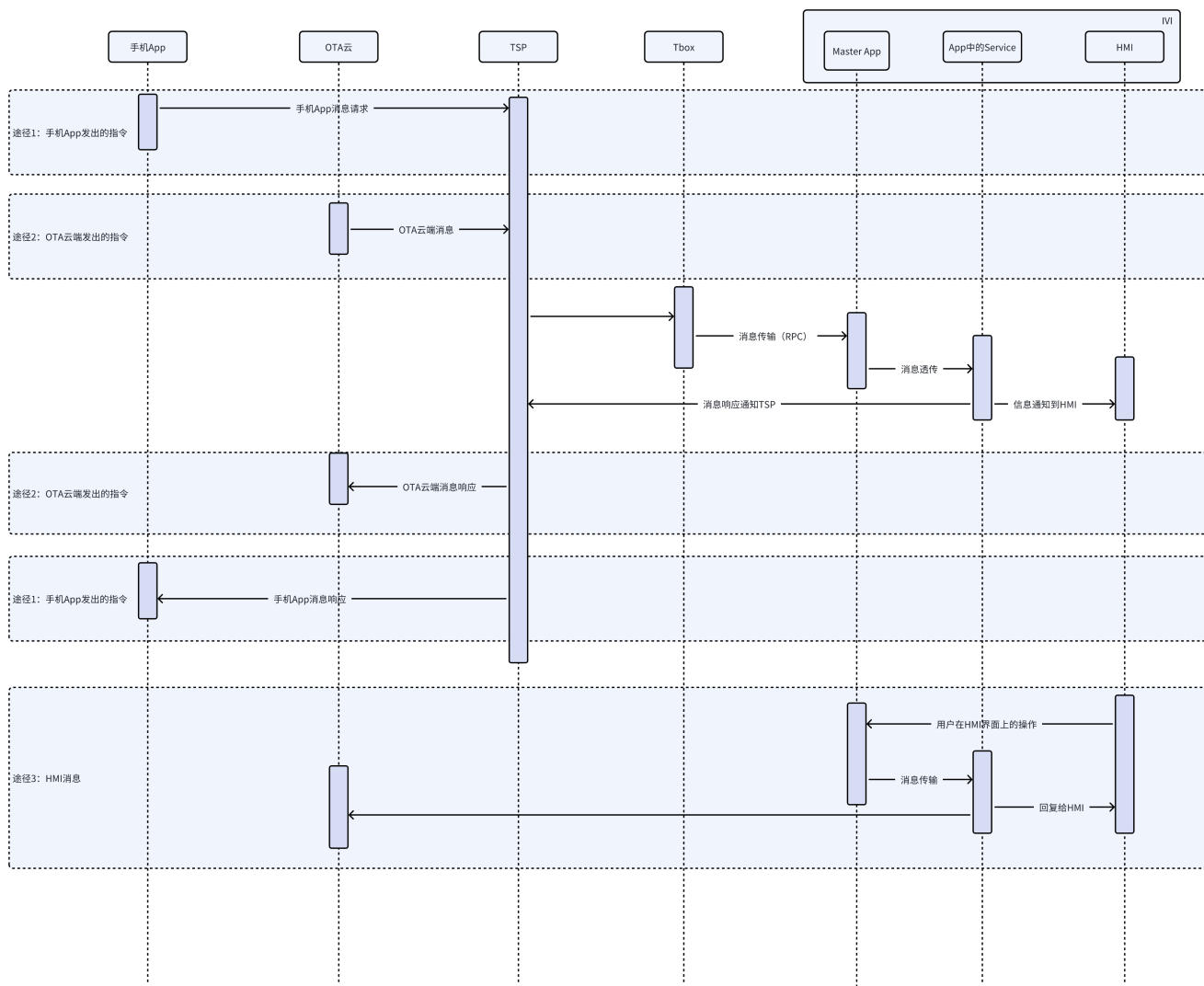
4.2 业务时序图

OTA中的业务触发途径有3条：

途径1：手机App下发的消息。手机App的消息经TSP流转到Tbox，Tbox经RPC将消息传输到OTA Master App。OTA Master App将消息处理后回复响应给Cloud，最后给手机App。

途径2：OTA云下发的消息。同途径1，最终会流转到TSP。工厂升级。

途径3：用户在HMI界面上的操作。HMI中的操作先到OTA Master App，OTA Master App将消息处理后回复响应给OTA云。



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

Master App主要功能是串起各个Service，每个Service有自己的数据持久化存储的处理，Master App中无数据持久化的处理。

6. 对外接口设计

//Tbox和HMI App的接口

手机App或者OTA云端下发的消息

消息来源	消息事件
手机App	设置预约时间
手机App	查询任务预约时间
手机App	取消预约
手机App	立即升级

用户在UI界面上的操作

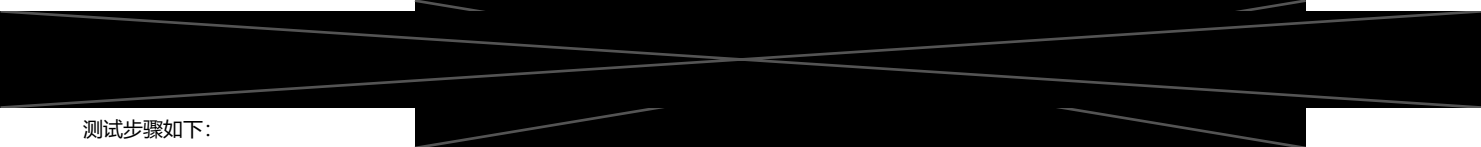
消息来源	消息事件
HMI	设置预约时间
HMI	查询任务预约时间
HMI	取消预约
HMI	立即升级
HMI	夜间升级开启
HMI	夜间升级关闭

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



测试步骤如下：

Step1: Cmake编译测试程序

```
rm -rf tmpbuild/virtualbox/x86_64/tmp/fota_master.ok
rm -rf tmpbuild/virtualbox/x86_64/build/fota_master/
cd ~/workspace/build/
make x86_64
```

Step2: 初始化测试环境

```
cd ~/workspace/build/tmpbuild/virtualbox/x86_64/build/fota_master/unittest
source env.sh //binetclogloglogdata
```

Step3: 配置数据放入etc目录下

由于车云协议更改等因素配置数据可能会调整

[etc.rar](#)

Step4: 开始测试

脚本需调整，如指定IP地址和工作目录

```
cp ../build/fota_master ./bin/
./start_test.sh start //fota_master
./start_test.sh stop //fota_master
```