

# [SCA] Driver App多市场个性化业务治理技术方案设计

## 1、需求概况

### 1.1 需求背景

目前drive app业务中，针对不同国家市场有不同的业务逻辑，随着国家数量的增加，业务功能的差异化管理成本越来越高。

当前使用的实现方式主要是通过if-else if-else判断，国家的增加使得多分支的处理复杂度变高，分支的判断处理与业务代码耦合度比较高，降低后续的扩展性和维护性。

为此我们针对国家化差异进行框架开发，通过不同的粒度控制功能，在实现上提供统一的接口，接口实现上根据国家业务进行定制，业务组件只需要调用相对应的接口，即可达到差异化管理目的。

### 1.2 需求目的

针对市场功能的差异化，完成不同粒度下的功能管理

### 1.3 需求功能

该需求主要功能为国家差异化管理，差异化的配置上有支持功能国家列表，不支持功能国家列表，和默认配置，实现上主要通过注解和依赖注入的方式，具体可见方案设计

## 2、技术方案设计

### 2.1 整体方案设计

方案实现上主要通过注解+字节码插桩/依赖注入的方式。

注解阶段，获取@CountryDiff 标注的方法和类，并获取其父类接口，生成以父类接口为key，value为注解方法信息的映射表列表，

在应用初始阶段，会调用SpxDiff框架初始化，初始化时传入当前国家，并加载由注解生成的映射表，在获取对应的功能的接口时，通过DiffFactory和国家配置，动态生成接口实现产品集。

### 2.2 详细方案设计

#### 2.2.1 产品集接口设计

将同一产品集，抽象其功能成一个接口，例如IdScanMode和BrScanMode都实现自IScanMode接口，在接口中定义具体方法，在具体实现类方法中添加@CountryDiff接口，并添加国家配置信息，各个组件在编译阶段，能够识别此产品，并添加到对应的产品集和实现类的映射表中。

#### 2.2.2 SpxDiff框架初始化

SpxDiff框架会对外提供init接口，init方法中需要传入当前国家，init方法调用会加载产品集的映射表。

映射表的初始化可通过字节码插桩实现，字节码插桩使用Gradle Plugin插件方式，在Transform阶段，实现对映射表的扫描，并通过字节码方式，插入到SpxDiff类中。

### 2.3 接口设计

#### 2.3.1 注解接口设计

注解接口，定义为方法级别的接口，具体参数如下，suppor为支持的国家列表，default判断是否为默认实现

#### 2.3.2 注解映射表设计

映射表设计为Map<String,List<DiffMeta>>结构，其中map为实现的接口名称，value为DiffMeta结构的集合

#### 2.3.3 Transform ASM字节码扫描

借助于Gradle Plugin插件，在Transform阶段，实现对上述生成的映射表的扫描，并通过字节码插入手段，最终将此映射表初始化到SpxDiff类中。具体实现上，主要包含两个阶段：

1. 通过继承ASM ClassVisitor，完成对匹配的目标接口的收集，缓存到Plugin的CacheConfig中；
2. 通过集成ASM MethodVisitor，将CacheConfig中收集到的启动任务，以字节码的方式，自动注册到SpxDiff中。

#### 2.3.4 Diff框架对外提供的接口

Diff框架最终对外提供get()接口，调用方传入接口类(如IScan.class)，内部会通过DiffFactory创建对应的产品对象

#### 2.3.5 DiffFactory设计

在SpxDiff接口中会有DiffFactory实例，具体接口如下，SpxDiff框架外部提供了初始化接口，在DiffFactory中会有创建接口实例类的create方法。

### 2.4 数据存储设计

暂无

### 2.5 监控方案设计

暂无

### 2.6 部署方案设计

暂无

## 3、风险评估

由于在接入框架时，需要替换原来逻辑代码，可能会存在逻辑遗漏的情况，考虑引入单元测试，确保接入框架后前后逻辑一致。

编译时增加默认实现判空逻辑。