软件方案设计说明书

**项目名称：浏览器V粉卡代理概要设计**

**文档修订历史**

|  |  |  |
| --- | --- | --- |
| 修订版本 | 修订作者 | 修订备注 |
| 20180329 | 江文 | 新增 |

## 术语表【可选】

|  |  |
| --- | --- |
| 名称 | 解释 |
|  |  |
|  |  |

**目录**

[术语表【可选】 1](#_Toc509932273)

[1 业务分析 3](#_Toc509932274)

[2 系统框架设计 3](#_Toc509932275)

[2.1 设计原理 3](#_Toc509932276)

[3 功能设计 5](#_Toc509932277)

[3.1逻辑流程图 5](#_Toc509932278)

[3.1 数据存储【可选】 7](#_Toc509932279)

[4 兼容性和可扩展性设计【可选】 7](#_Toc509932280)

[5 安全性设计【可选】 7](#_Toc509932281)

[6 性能设计【可选】 7](#_Toc509932282)

[7 功耗设计【可选】 7](#_Toc509932283)

[8 可测试性设计【可选】 7](#_Toc509932284)

# 业务分析

V粉卡的功能：通过设置免流代理，让流量走专属通道，实现通道里流量免费的功能。

免流原理：目前有运营商主动免流和代理方案两种方式进行免流。

1. 运营商主动免流（IP收敛方式）

原理：运营商会对vivo相关应用访问的域名/IP直接进行免流。每个应用访问的域名由专人收集。浏览器目前涉及的域名包括\*.vivo.com.cn等，涉及的ip包括省流加速的ip。

1. 代理方案（针对IP非收敛的应用）

原理：针对域名不确定或者访问域名是第三方的应用，使用代理的方式进行免流，免流SDK提供当前可用的代理信息、鉴权认证信息，由应用自己去设置代理和认证信息，针对WebView，SDK提供了专门的方法用于设置和清除代理。对于浏览器内核，涉及如下几个场景：网页打开、mediaplayer（包括exo mediaplayer和system mediaplayer）、省流加速切直连情况。

基于v粉卡免流原理中代理方案的使用，本概要设计主要包括如下几个方面：

1. 接收来自外壳的免流代理信息；
2. 向内核提供免流代理信息查询接口；
3. 控制内核各个模块（网页、media、加速）是否走免流通道；

# 系统框架设计

## 方案设计

**使用系统mediaPlayer走免流代理**



**外壳调用exo走免流代理**



**内核调用exo走免流代理**

Exo播放视频时，setDataSource时直接从ProxyServiceJavaBridge获取代理信息，并设置到playsdk中。

**网页走免流代理**



## 模块功能设计

**内核调用exo走免流代理**

内核判断当前使用exo时，将相关的代理信息设置进去。Exo建立连接时会将代理信息以及鉴权信息带上。

**系统mediaPlayer走免流代理**

系统MediaPlayer播放器无法设置代理，方案通过创建本地Server代理Mediaplayer的播放器请求，实现代理播放，具体接入方法见<播放器本地代理SDK使用说明.docx>

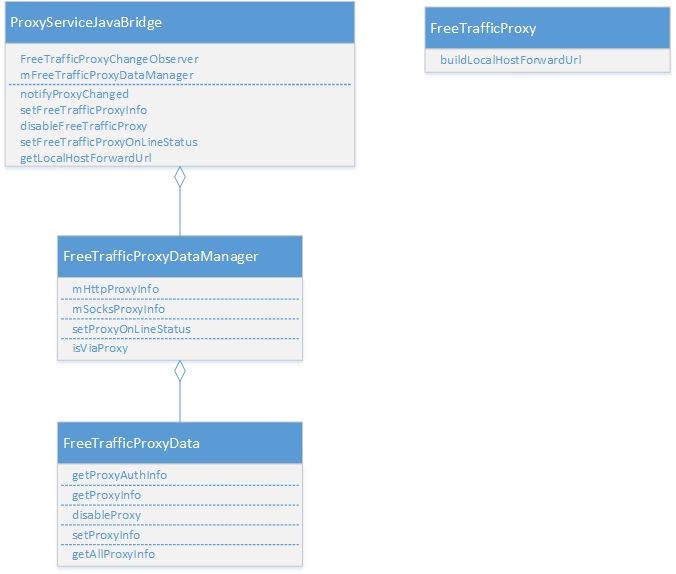
**外壳调用exo走免流代理**

外壳调用exo走免流代理的方式很简单，播放前直接通过调用exo的公共接口setProxy，并将type、port、domain、orderId、orderKey以map<String, String>形式作为参数带入即可设置好免流代理。Exo中的代理流程与内核调exo进行免流代理的流程相同。

**网页走免流代理**

网页走免流代理的流程与之前走vivo加速的流程无多大区别，不再重新进行设计。

**免流代理控制模块结构**



按照现有框架，在GlobalSetting中增加一套setStringMapValue的接口，用于外壳传递Map结构的数据到内核，通过该接口，向ProxyControllerBridge传递免流代理信息，进一步向代理配置模块传递，并建立免流代理。

# 兼容性和可扩展性设计【可选】

*无*

# 安全性设计【可选】

*无*

# 性能设计【可选】

*无*

# 功耗设计【可选】

无

# 可测试性设计【可选】

无