软件方案设计说明书

**项目名称：多代理代码重构**

**文档修订历史**

|  |  |  |
| --- | --- | --- |
| 修订版本 | 修订作者 | 修订备注 |
| 20180328 | 张守年 | 新增 |

## 术语表【可选】

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

**目录**

[术语表【可选】 1](#_Toc490492496)

[1 业务分析【可选】 3](#_Toc490492497)

[2 系统框架设计 4](#_Toc490492498)

[2.1 模块功能设计 5](#_Toc490492499)

[2.2 数据监控 6](#_Toc490492500)

[3 兼容性和可扩展性设计【可选】 7](#_Toc490492501)

[4 安全性设计【可选】 7](#_Toc490492502)

[5 性能设计【可选】 7](#_Toc490492503)

[6 功耗设计【可选】 7](#_Toc490492504)

[7 可测试性设计【可选】 7](#_Toc490492505)

# 业务分析【问题背景】

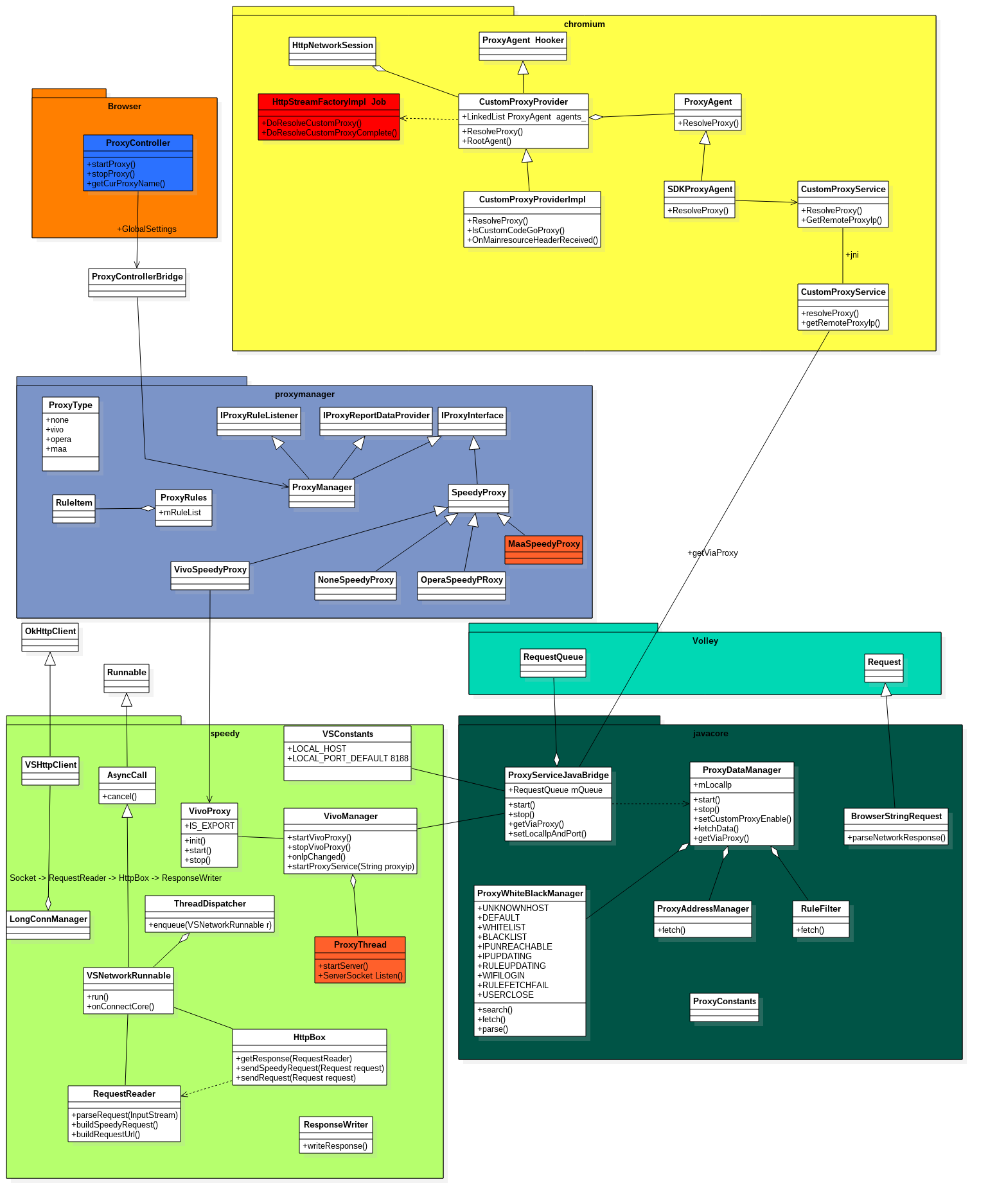
V2.3内核版本将大量的位于UI层的功能移植回内核，同时省流加速功能也移植回内核，代理的结构在内核没有进行适配，特别是与各个模块之间的引用依赖关系，包括内部针对多种代理共存和直连的交互关系，同时针对各类情况又要引入不同的埋点上报，导致目前代理存在两个主要问题：

1 对内结构比较混乱，存在一些冗余的类结构，流程不清，命名混乱。

2 对外各类依赖混乱，暴露的接口不统一，对埋点的一些处理，对Util工具类的一些依赖。

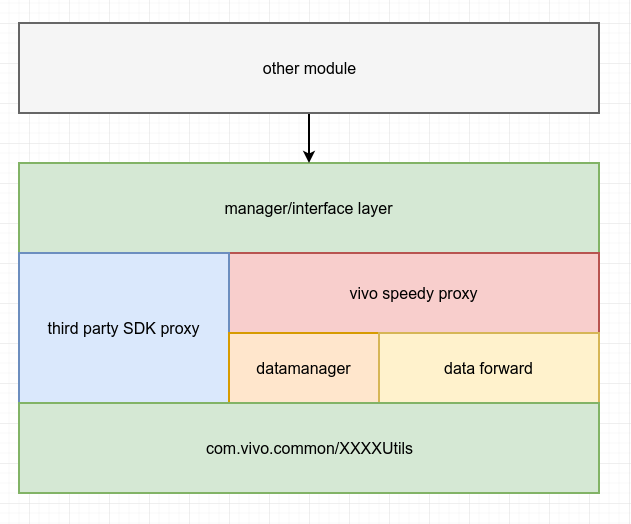
所以2.4本期主要任务是梳理省流模块对不同代理类型的管理，对外暴露的接口规范，对外重点模块的交互，接口的收拢统一。

# 系统框架设计



上图是当前的一个结构梳理，主要是左下蓝色和绿色的部分，这里无论是命名还是结构上都存在一些冗余，看起来让人迷惑的地方。

重构后整体层次上如下所示：



# 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