复用文档

识别可复用组件

- 系统的构件
    - 系统体系结构：
        - netty框架，而不是用socket重新写
        - 前后端通过EventBus机制解耦
        - 将功能划分为不同的低耦合、高内聚的模块单独实现（channel中有一系列顺序流过的handler来分别实现不同的功能，而不是一个handler实现所有的功能）
    - UI设计构件
    - 配置文件读取构件(CM构件)
    - 数据库连接构件
    - 日志记录模块(PM构件)
    - 速率限制许可证模块(License模块)
- 项目管理和文档
    - 项目管理模式
    - 文档构件：项目管理文档，项目计划表
    - 测试数据：测试用例设计

可复用构件开发与发布

生成jar包 - PM构件 https://github.com/bookish-component/PM - 接收应用程序的性能指标（指标名称，指标数值） - 每分钟自动生成性能报告 - 性能报告输出到单独的性能文件，文件名包括性能报告时间 - CM构件 https://github.com/bookish-component/CM - 从文件中读取参数配置 - 提供查询接口 - 动态加载 - LICENSE构件 https://github.com/bookish-component/License - 每收到一个请求，计数加1 - 根据已经收到的消息数量和预设的数值，判断是否可以继续提供服务 - Throughput - Capacity - DATABASE构件 https://github.com/bookish-component/SQLHelper

冗余检查

- 过程冗余：由于登陆消息和聊天消息公用一条pipeline，所以消息在channel中流动时会有冗余，但这种冗余反而降低了设计的复杂度，是可接受的。