# 项目背景

# 需求分析

## 服务器

|  |  |  |
| --- | --- | --- |
| 序号 | 需求 | 相关模块 |
| 1 | 使用C对接O32系统 | emsmodule/adapter/ufxapi |
| 2 | 多账户多策略同时运行 |  |
| 3 | 提供行情订阅服务 |  |
| 4 | 策略使用python编写 |  |
| 5 | 高可用性，任意模块能从任意状态恢复 |  |
| 6 | 提供策略回测功能 |  |
| 7 | 提供策略版本管理功能 |  |
| 8 | 支持自定义交易算法（完成Portfolio交易） |  |
|  |  |  |
|  |  |  |

## 前端

|  |  |
| --- | --- |
| 序号 | 需求 |
| 1 | 分用户角色显示不同界面（用户角色管理） |
| 2 | 风控员：   1. 风控规则界面（可以进行规则增删改查） 2. 实时风险警告界面（支持按条件过滤） 3. 策略盈亏、策略动态回撤（面向私募机构） |
| 3 | 交易员：   1. 策略列表（显示所有策略、说明、历史回撤、年化收益、SR等信息） 2. 策略运行界面（显示当前该交易员启动的策略【产品名称、策略类型、策略初始资金、当前资金、当日盈亏、浮动盈亏】 3. 策略明细界面（当日盈亏、浮动盈亏、持仓、当日成交、当日委托，当前信号与阀值图） |
| 4 | 研究员：   1. 嵌入编辑器到客户端，使用python进行开发（python语法高亮，并提示服务器导出的python函数【行情、交易等】） 2. 能够在仿真环境进行策略debug 3. 支持回测参数设置（交易费用、滑点） |
| 5 | 运维人员：   1. 显示各个模块之间链接情况，模块cpu、内存使用 2. 显示行情延时，交易adapter链接情况 3. 显示各个模块中队列长度 4. 热备模块运行情况，一键切换至热量化交易系统 |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

# 技术选型

动态加载算法：算法以go编译so形式加载？

动态加载策略：策略关键字绑定策略New函数，返回策略接口？

策略语言选择：R？Python？

实时行情数据源：redis？MySQL？

# 系统架构

## 行情模块

## 策略模块

## 风控模块

## 订单执行模块

## 订单管理模块

## 监控模块