FAMS开发手册（v0.0.1）

# 目录

[目录 1](#_Toc37792107)

[修订（开发）记录 3](#_Toc37792108)

[一、FAMS简介 4](#_Toc37792109)

[二、FAMS总体架构 5](#_Toc37792110)

[2.1 业务层面系统划分 5](#_Toc37792111)

[2.2 服务层面系统划分 6](#_Toc37792112)

[三、环境汇总 7](#_Toc37792113)

[3.1 外部服务依赖 7](#_Toc37792114)

[3.2 开发环境 7](#_Toc37792115)

[3.2.1 Mysql 7](#_Toc37792116)

[四、模块划分 8](#_Toc37792117)

[4.1 总体模块划分 8](#_Toc37792118)

[4.2 fams-framework 8](#_Toc37792119)

[4.3 fams-springboot-starter 9](#_Toc37792120)

[4.4 fams-ace 交易前台 9](#_Toc37792121)

[4.5 fams-king 交易中台 9](#_Toc37792122)

[4.6 fams-queen 交易后台 9](#_Toc37792123)

[五、模块依赖 10](#_Toc37792124)

[六、服务治理 11](#_Toc37792125)

[6.1 服务注册中心MonitorService 11](#_Toc37792126)

[6.2 权限管理中心AuthService 11](#_Toc37792127)

[七、Framework设计 12](#_Toc37792128)

[八、交易前台 ACE 13](#_Toc37792129)

[8.1 ACE定位 13](#_Toc37792130)

[九、交易中台 KING 14](#_Toc37792131)

[9.1 KING定位 14](#_Toc37792132)

[9.2 KING公共组件 14](#_Toc37792133)

[9.2.1 StrategyContainer策略容器 14](#_Toc37792134)

[9.2.2 OrderExecutor订单处理机 15](#_Toc37792135)

[9.2.3 AccountContainer账户资产管理 15](#_Toc37792136)

[9.2.4 TradeAnalysisContainer 15](#_Toc37792137)

[十、交易后台 QUEEN 16](#_Toc37792138)

[10.1 QUEEN定位 16](#_Toc37792139)

[十一、Python服务 17](#_Toc37792140)

# 修订（开发）记录

| 日期 | 版本 | 修订说明 | 修订者 |
| --- | --- | --- | --- |
| 2020.03.04 | v0.0.1 | 新建 截止到2020.03.04 已有功能：   1. ignite集成cache 2. 序列化模块，框架搭建，规范化编码 3. 集成redis，集成mybatis | 姜瀚 |
| 2020.04.14 | v0.0.1 | 设计ace，king，queen框架 | 姜瀚 |

# 一、FAMS简介

金融资产管理系统FAMS（Financial Asset Manager System），旨在建立一个涵盖金融投资行为的前中台的全覆盖产品，对标系统为恒生O32系统。愿景为可以支持基金公司级的日常投资业务需求。

从业务角度讲，FAMS提供投前、投中、投后的管理。投前服务，将提供策略池、股票池、投前风控等功能；投中服务，将提供内置的指令流转模块支持投资指令的生命周期、投中市场/系统风险控制；投后服务，提供实时估值，持仓管理，持仓风险管理等功能。

从系统角度讲，FAMS提供包括交易、行情、风控、估值服务。交易模块包括，普通交易、策略交易、伪高频交易（内存交易）；风控风控包括事前风控、实时交易风控、事后风控等。

从架构角度讲，FAMS采用标准的分布式架构，支持高可用性、高可扩展。从服务粒度讲，FAMS主体架构是基于Java语言的Spring-Cloud微服务架构，同时支持dubbo类型的服务接入。可扩展性：采用松散的结构，服务之间弱依赖性，支持python版本和c/c++版本的微服务接入，以支持更多用户的习惯。

# 二、FAMS总体架构

## 2.1 业务层面系统划分

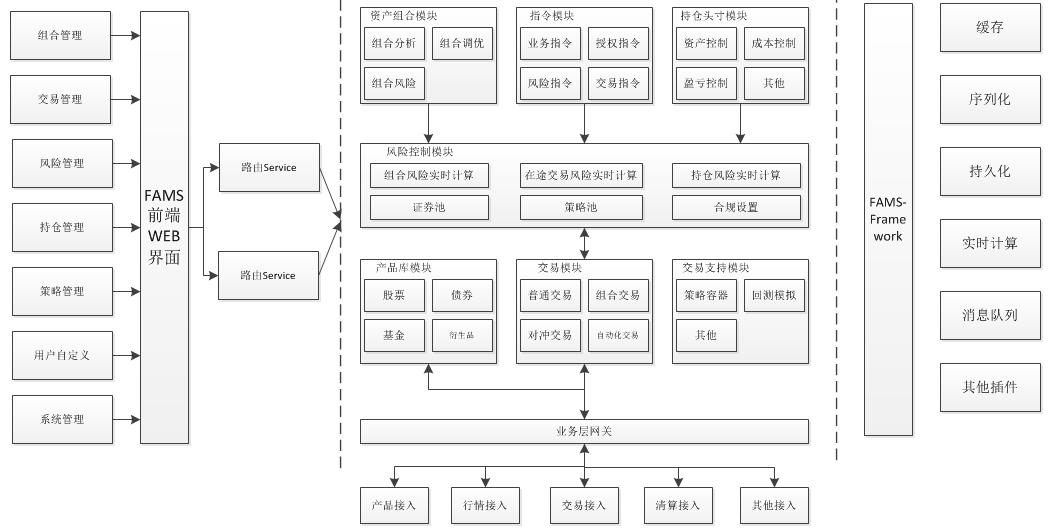


图2.1 业务层面系统划分

1. 前端用户可视层：完成用户的指令搜集任务。包括交易相关的普通委托、算法交易委托、策略交易委托等；组合管理相关的创建组合、组合试算、组合分析、组合风险评估等；风险管理相关的风险策略设定、风险指标设定等；策略管理相关的创建策略、加载策略等

2. 业务逻辑处理层：解析前台的用户指令，完成指令操作

3. 系统支持层：为了实现上层业务，而封装的一系列框架可用组件

## 2.2 服务层面系统划分

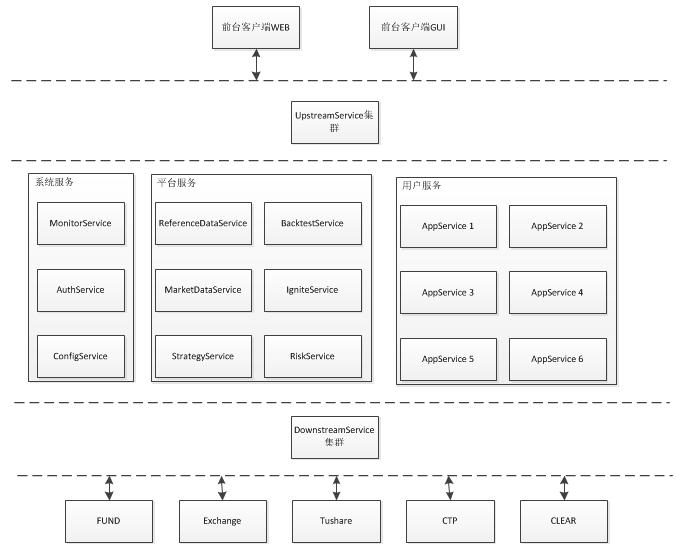


图2.2 service级系统划分

# 三、环境汇总

## 3.1 外部服务依赖

|  |  |  |
| --- | --- | --- |
| 外部依赖组件名称 | 版本 | 系统中功能 |
| Zookeeper | 3.5.6 | Kafka依赖，后续作为dubbo类服务的服务治理 |
| Kafka | 2.4.0 | 消息中间件 |
| Redis | 3.2.1 | 普通数据的缓存 |
| Mysql | 5.6.46 | 普通数据的存储 |
| Ignite | 2.7.6 | 内存计算、流数据处理 |

## 3.2 开发环境

### 3.2.1 Mysql

|  |  |
| --- | --- |
| Ip | 122.51.159.32 |
| Port | 3307 |
| Username |  |
| Password |  |

# 四、模块划分

## 4.1 总体模块划分

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 模块名称 | 模块项目 | 模块说明 |
| 1 | 消息序列化 | fams-bo | 系统内消息的序列化模块 |
| 2 | 公共包模块 | fams-common | 系统的公共工具模块，封装通用工具类 |
| 3 | FAMS框架模块 | fams-framework | 底层框架封装和实现 |
| 4 | FAMS启动模块 | fams-springboot-starter | 框架的starter |
| 5 | 交易前台 | Fams-ace | 包括订单管理、行情服务等 |
| 6 | 交易中台 | Fams-king- | 包括估值、风控、策略模型、回测、组合交易等 |
| 7 | 交易后台 | Fams-queen- | 包括账户体系、证券产品库、指令流转等 |

## 4.2 fams-framework

|  |  |  |
| --- | --- | --- |
| 序号 | 模块名称 | 模块说明 |
| 1 | fams-framework-common | Framework模块的公共包 |
| 2 | fams-framework-core | Framework模块的核心包，封装框架所有都需要的插件，比如数据库模块、消息队列模块、缓存队列等 |
| 3 | fams-framework-ignite | 封装ignite相关业务的实现，FAMS使用ignite作为分布式缓存、流式数据处理（交易和风控）、机器学习（策略）、大数据处理（回测框架） |
| 4 | fams-framework-service | Fams系统级的服务 |

## 4.3 fams-springboot-starter

|  |  |  |
| --- | --- | --- |
| 序号 | 模块名称 | 模块说明 |
| 1 | fams-framework-ignite-springboot-starter | Framework-ignite模块的启动包，封装  @EnableIgnite：启用ignite节点  @EnalbeIgniteCache:ignite节点启用cache |
| 2 | fams-framework-service-springboot-starter | 框架服务相关启动包  @EnableFAMSServerRegister：标识服务为注册中心  @EnableFAMS：标识服务为FAMS类服务  @FAMSEventListener：标识方法为事件接收方 |
| 3 |  |  |

## 4.4 fams-ace 交易前台

## 4.5 fams-king 交易中台

|  |  |  |
| --- | --- | --- |
| 序号 | 模块名称 | 模块说明 |
| 1 | 交易中台公共组件 | 包括行情容器MarketDataContainer、策略容器StrategyContainer、交易分析容器TradeAnalysisContainer等 |
| 2 | 事件型回测引擎 | 基于事件（Bar或Tick行情）触发的回测框架 |
| 3 |  |  |

## 4.6 fams-queen 交易后台

# 五、模块依赖

截止到2020.02.26，service-starter不再依赖ignite-starter

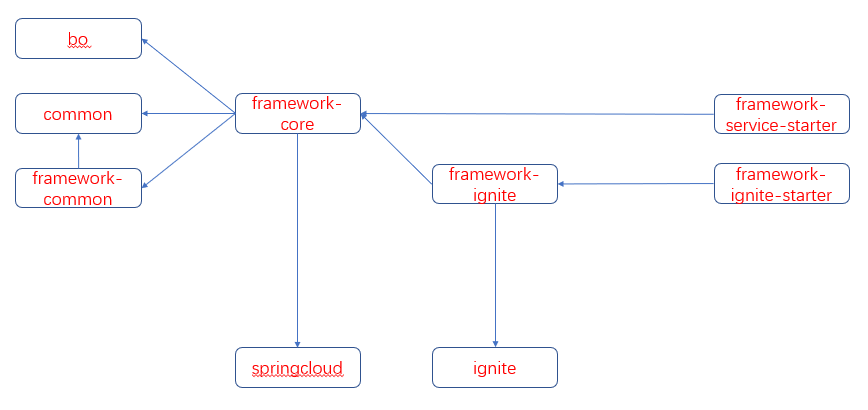


图5.1 各模块工程依赖

# 六、服务治理

本模块提供的服务，是FAMS平台的系统级微服务，用于支持整个平台的正常运行

## 6.1 服务注册中心MonitorService

用于管理当前平台中的service，提供负载均衡，服务治理，以及service状态异常的后续处理。一个FAMS系统中，必须存在至少一个active状态的MonitorService。当已存在active状态的MonitorService时，新加入的MonitorService将作为备用服务，暂时不支持同时存在多个active的MonitorService。主要是考虑到当前环境，存储的信息必须要实现多个服务的同步，才能保证MonitorService的正确性。

目前主要功能：

1、服务上线、心跳、状态变化、掉线时，会由MonitorService做服务的治理（已完成）

2、服务下线时，如果所有该服务的实例都已退出，MonitorService会做一系列的善后工作，如删除缓存数据等（开发中）

3、提供其他服务的问询和应答

## 6.2 权限管理中心AuthService

# 七、Framework设计

# 八、交易前台 ACE

## 8.1 ACE定位

交易前台ACE是扮演客户前端和外部系统的“前线冲锋者”角色，一方面负责把客户请求收集并发送给最终“决策者”交易中台KING，另一方面在KING分析、降解等一系列交易决策后，ACE只负责执行最终的交易指令。

ACE最重要的组成部分是OMS（订单管理系统）和MDLH（Market Data Line Handle），以及交易清算系统Clear & Settlement Service

# 九、交易中台 KING

## 9.1 KING定位

交易中台KING是扮演系统的“决策者”角色，是整个系统中最重要的部分，它是整个系统的大脑和指挥官。KING在收到前台业务请求后，根据事先的编程，业务请求拆分并具体实现为多个交易指令单元。例如前台收到的组合策略单，交易前台ACE只做组合策略的目标设定，至于如何组合，如何择时委托，如何控制风险等，均由中台KING来进行决策。

KING最重要的组成部分是风险控制服务Risk Manager Service、策略交易服务Strategy Trading Service、组合交易服务Portfolio Trading Service、估值测算中心Valuation & Calculation Center、交易结果分析 Trading Analysis Service等。

功能类的服务还有：事件驱动型的回测服务Backtest Service

## 9.2 KING公共组件

### 9.2.1 StrategyContainer策略容器

用于加载自定义交易策略的模块，采用观察者模式设计。

|  |  |
| --- | --- |
|  |  |
| StrategyConfig | 策略模块参数配置，由StrategyAutoConfigure加载，目前需要主动import，后期预计移动到一个完整的autoConfigure中 |
| OrderExecutor | 订单驱动模块。当策略发出交易信号后，由此模块进行后续的驱动，详见9.2.2小节 |
| AccountContainer | 账户管理器，处理账户的资金划转和计算，详见9.2.3 小节 |
| TradeAnalysisContainer | 交易结果分析器，分析当前的交易指标，详见9.2.4 小节 |

### 9.2.2 OrderExecutor订单处理机

用于处理委托和成交的订单驱动器。

处理两种事件：

1. 上游的订单委托：handlerOrder，当回测时，直接生成成交订单并通知处理成交订单handlerExecutor，如果是真实交易时，会发送给风控系统进行风险控制，最终发送前台oms并委托到交易所；
2. 下游的订单成交：handlerExecutor，收到成交回报后，推进到资产账户的管理（AccountContainer）

以及一个对外接口,用于生成当前为止的交易分析报告TradingAnaylsisContainer

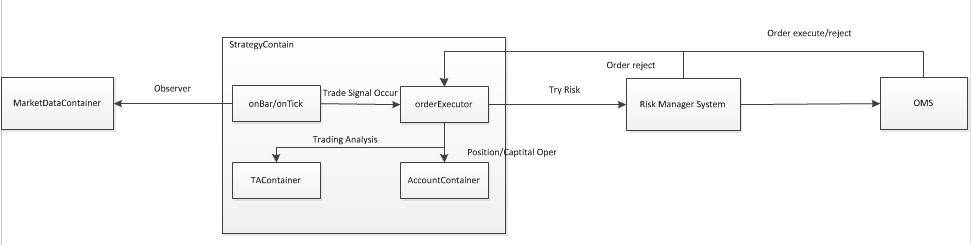


图9.1 自动化策略交易（StrategyContainer部分）

### 9.2.3 AccountContainer账户资产管理

当发生交易时，进行现金资产和股份的处理

### 9.2.4 TradeAnalysisContainer

当日交易结束时，产生交易结果分析，可以自定义一系列指标用于分析

# 十、交易后台 QUEEN

## 10.1 QUEEN定位

交易后台QUEEN是扮演系统的“保障者”角色，是系统的重要串联者和大量数据的拥有者。KING和ACE日常决策的数据来源于QUEEN，可以是QUEEN独立服务提供的，也可以是KING或ACE自主获取的，但数据的管理一定是由QUEEN或QUEEN的组件完成的。

QUEEN最重要的部分是产品库Refenerce Data Service、账户管理系统Account Manager Service等

# 十一、Python服务