

ICS 03.060

CCS A 11

团 体 标 准

T/ZQX XXXX—XXXX

期货公司业务系统微服务架构设计指南

Guidelines for microservices architecture design of futures company
business systems

（征求意见稿）

XXXX—XX—XX 发布

XXXX—XX—XX 实施

中国期货业协会 发布

目 次

前言	II
引言	III
1 范围	1
2 规范性引用文件	1
3 术语和定义	1
4 缩略语	3
5 期货公司业务系统微服务架构设计指南	3
5.1 总体架构	3
5.2 等级概述	4
6 基础设施层	4
6.1 资源管理	4
6.2 资源编排	5
6.3 资源动态分配	6
7 微服务开发与运行层	7
7.1 服务开发	7
7.2 持续集成与发布（CI/CD）	7
7.3 服务发现	8
7.4 服务编排	9
7.5 通信协议	9
8 微服务治理层	10
8.1 服务限流	10
8.2 服务路由	11
8.3 服务容错	12
8.4 负载均衡	13
9 微服务支撑层	13
9.1 注册中心	13
9.2 配置中心	14
9.3 日志与审计中心	15
9.4 监控中心	16
9.5 服务安全	17
10 业务公共服务层	18
10.1 监管协同服务	18
10.2 数据治理服务	19
10.3 流程协同服务	19
附录 A（资料性附录） 应用案例	20
A.1 智能体开发平台	20
A.2 业务运营平台项目	20
参考文献	22

前 言

本文件按照 GB/T 1.1—2020《标准化工作导则 第一部分：标准化文件的结构和起草规则》JR/T 0352.1—2026《证券期货业标准化工作导则 第1部分：标准的名称》的规定起草。

请注意本文件的某些内容可能涉及专利。本文件的发布机构不承担识别专利的责任。

本文件由中国期货业协会提出并归口。

本标准起草单位：×××××。

本文件主要起草人：XXX、XXX、XXX、XXX、XXX、XXX。

引 言

随着金融科技深化与期货市场扩容，期货业务系统作为承载客户、结算、风控等核心职能的关键应用软件，正面临需求高频迭代与监管趋严的双重挑战。传统单体架构因耦合度高、扩展受限、交付缓慢，已难以支撑业务“敏态”与“稳态”双轨运行。微服务架构凭借独立部署、松耦合及弹性伸缩等优势，已成为期货行业系统演进的必然趋势。

然而，微服务在解构系统的同时，亦引入分布式事务复杂、调用链路追踪困难、运维治理成本骤增等挑战。期货行业作为强监管、高风险的金融领域，对数据强一致性、系统高可用及安全合规有着极严要求。若缺乏统一标准，微服务的无序拆分极易演变为“分布式单体”，引发严重的稳定性风险与运维灾难。

为指引业务系统微服务架构的科学规划与高效治理，特制定本指南。本指南立足期货公司业务特征与需求，系统性约定了基础设施层、微服务开发与运行层、微服务治理层、微服务支撑层、业务供提供服务层等层次的核心能力，旨在为期货机构提供可操作的架构参考，规避技术风险，确保系统在敏捷创新的同时满足金融级安全合规要求，推动期货行业IT架构的标准化与高质量发展。

期货公司业务系统微服务架构设计指南

1 范围

本文件规定了期货公司业务系统微服务架构及相关组成部分的功能要求。
本文件适用于期货公司业务系统微服务架构的设计、研发、建设、运行和维护。

2 规范性引用文件

下列文件中的内容通过文中的规范性引用而构成本文件必不可少的条款。其中，注日期的引用文件，仅该日期对应的版本适用于本文件；不注日期的引用文件，其最新版本（包括所有的修改单）适用于本文件。

GB/T 42568-2023 工业互联网平台 微服务参考框架
GB/T 43335—2023 自动化系统与集成 技术资源共享服务系统参考架构
JR/T 0099-2012 证券期货业信息系统运维管理规范
JR/T 0112-2014 证券期货业信息系统审计规范
JR/T 0158-2018 证券期货业数据分类分级指引
JR/T 0233-2021 证券期货业经营机构内部应用系统 日志规范
JR/T 0248-2022 证券期货业机构内部接口 证券交易
JR/T 0250-2022 证券期货业数据安全管理与保护指引
JR/T 0275-2023 证券期货业机构内部接口 资讯数据
JR/T 0295-2023 证券期货业信息安全运营管理指南

3 术语和定义

下列术语和定义适用于本文件。

3.1

微服务 `micro services`

可独立部署，并提供可实现某应用中特定功能的服务的制品。

[来源：GB/T42568—2023，3.1.3]

3.2

技术架构 `technology architecture`

整个或部分技术系统的可重复设计，表现为一组抽象构件及构件实例间交互的方法。

[来源：GB/T43335—2023，3.1.4]

3.3

微服务架构 `micro service architecture`

一种软件开发的架构和组织方法，其中软件由若干小型独立的服务组成，这些服务通过定义的应用程序编程接口进行通信。

3.4

组件 component

可以复用的软件组成部分（如一组相互关联的微服务、软件应用的某些复用部分等），也称作构件。

3.5

服务网格 service mesh

一组处理服务间大量进程以及相互网络通信的代理组件和任务管理组件。注：代理组件处理入站和出站数据包，任务管理组件控制流量，保障服务之间复杂调用的可靠性和易用性。

[来源：ITU-TJ.1306:2023, 3.2.12]

3.6

容器 container

一套软件，用于提供隔离、资源控制和可移植性应用程序的虚拟化处理。

[来源：ITU-TY.3535:2022, 3.2.1]

3.7

容器编排 container orchestration

对容器的部署和组织，提供调度和管理容器集群的能力，包括容器自动化部署、管理、弹性伸缩和容器网络管理等。

[来源：ITU-TJ.1306:2023, 3.2.3]

3.8

开发运维一体化 development and operations; DevOps

一组过程、方法与系统的统称，用于促进开发（应用程序/软件工程）、技术运维和质量保障部门之间的沟通、协作与整合。

[来源：ITU-TJ.1306:2023, 3.2.6]

3.9

持续集成与发布 continuous integration and continuous deployment; CI/CD

一种帮助团队成员频繁集成和发布其工作成果的软件开发实践方法。注：持续集成的每次集成都会经过自动检验，以尽快发现集成错误，缩短系统开发生命周期。持续发布能够自动将已经验证的代码发布到存储库，从而建立可以随时部署到生产环境的代码库。持续集成与发布在很大程度上需要依赖精细设计的自动化测试。

[来源：ITU-TJ.1306:2023, 3.2.4]

3.10

API 网关 API gateway

在服务端实现的、对服务调用者提供统一接入管理的系统，对外部调用提供了统一的出入口，屏蔽了内部服务的实现机制。

3.11**低代码开发平台 low-code development platform**

无需编程或通过少量代码就能快速生成应用程序的开发平台。注：低代码开发平台通过可视化的方式进行应用程序开发，使开发人员可以通过图形化的用户界面，使用拖拽组件和模型驱动的逻辑来创建各类应用程序。

[来源：ITU-TJ.1306:2023, 3.2.10]

4 缩略语

下列缩略语适用于本文件。

API：应用程序编程接口（Application Programming Interface）

CI/CD：持续集成与发布（Continuous Integration/Continuous Delivery）

DevOps：开发运维一体化（Development and Operations）

FTP：文件传输协议（File Transfer Protocol）

HTTP：超文本传输协议（Hyper Text Transfer Protocol）

HTTPS：超文本传输安全协议（Hyper Text Transfer Protocol Secure）

K8s：容器编排平台（Kubernetes）

MQTT：消息队列遥测传输协议（Message Queuing Telemetry Transport）

QPS：每秒查询率（Queries Per Second）

RPC：远程过程调用（Remote Procedure Call）

SFTP：安全文件传输协议（Secure File Transfer Protocol）

TLS：传输层安全协议（Transport Layer Security）

TPS：每秒事务处理量（Transactions Per Second）

XML：可扩展标记语言（Extensible Markup Language）

5 期货公司业务系统微服务架构设计指南**5.1 总体架构**

《期货公司业务系统微服务架构设计指南》总体与图1相符，覆盖基础设施层、微服务开发与运行层、治理层、支撑层及业务公共服务层，形成全栈式架构标准。

在基础设施层面，强调资源精细化管理与编排，支持基于业务负载的动态调度与预测性资源分配，保障交易系统在日盘、夜盘等高峰时段的稳定性。

开发与运行层聚焦标准化开发框架、CI/CD自动化流水线与多协议通信能力，提升研发效率与系统集成性。

支撑层强化注册中心、配置中心、日志与审计中心、监控中心与安全能力的协同能力，提升系统可观测性与运维智能化水平，并全面落实权限管理、数据安全、开发安全等要求。

治理层构建以限流、路由、容错、负载均衡为核心的运行保障机制，实现服务高可用。

业务公共服务层引入监管协同、数据治理与流程协同服务，推动合规自动化、数据资产

化与流程可视化。

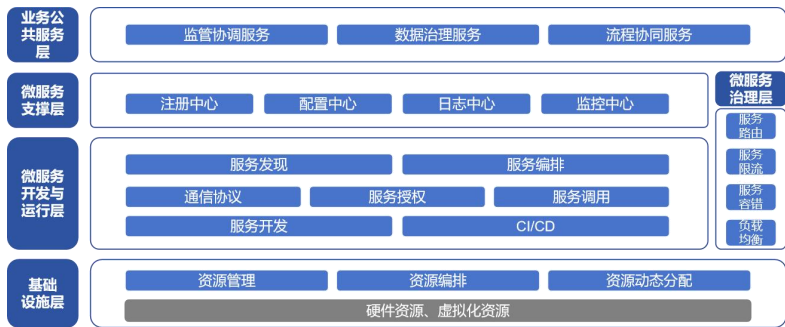


图1 期货公司业务系统微服务架构

5.2 等级概述

5.2.1 基础级

对应期货行业微服务平台的规范化建设阶段，平台已完成全面的微服务架构设计，具备服务注册发现、配置管理、熔断降级等核心基础能力，能够实现业务模块的标准化解耦与稳定交互，满足行业基础数据规范与系统运行稳定性要求，为期货业务提供可靠的底层技术支撑。

5.2.2 增强级

对应期货行业微服务平台的智能化升级阶段，平台在全面级基础之上深度融合智能化能力，具备智能运维、数据驱动决策等特性，能够实现故障自愈、弹性伸缩与业务场景的智能适配，通过技术能力的智能化演进，为期货行业数字化转型提供更高效、敏捷的技术赋能。

6 基础设施层

6.1 资源管理

6.1.1 硬件资源精细化管理

对服务器、存储设备、网络交换机等硬件资源进行全生命周期管理。

基础级能力要求如下：

- a) 自动发现底层物理服务器、CPU、GPU、内存、磁盘（本地/共享存储）、网络设备（网卡、带宽）等硬件资源；
- b) 实时监控硬件资源关键运行指标，包括但不限于 CPU 使用率、内存容量、磁盘 I/O 读写速率、网络带宽占用等，当指标超出预设阈值时自动触发告警，为资源调配和设备更换提供数据支撑。

增强级能力要求如下：

- a) 结合期货交易时段特性（如开盘高峰期、夜盘时段），建立硬件资源负载预测模型，提前识别性能瓶颈，辅助制定针对性扩容或调度方案，保障交易系统稳定性；
- b) 对硬件设备的维护记录、故障历史进行关联分析，生成设备健康度评分，为设备淘汰、更新提供决策依据，降低突发故障风险。

6.1.2 虚拟化资源池化管理

构建虚拟化资源池，对虚拟资源进行统一分配、回收与监控，支持资源的快速创建、销毁和迁移。

基础级能力要求如下：

- a) 构建统一的虚拟化资源池，通过虚拟化技术将物理服务器资源灵活划分为虚拟机、容器等虚拟资源，实现资源的动态分配、回收与监控；
- b) 支持虚拟资源的快速创建、销毁和跨物理节点迁移，满足期货微服务（如交易撮合、行情推送、风险控制）的弹性伸缩需求，尤其在交易量激增时可快速扩容。

增强级能力要求如下：满足基础级能力要求。

6.2 资源编排

6.2.1 业务导向的编排模板设计

针对期货行业不同业务场景，设计专用的资源编排模板。

基础级能力要求如下：

- a) 使用声明式模板（如 Kubernetes Manifest, Helm Chart, Docker Compose）定义微服务应用及其所需资源（CPU、GPU、内存、存储卷、网络配置、环境变量、配置映射、密钥）；
- b) 在模板中声明微服务间的依赖关系（启动顺序、健康检查依赖）以及对外部服务（如数据库、消息队列）的依赖。

增强级能力要求如下：

- a) 提供图形化界面（GUI）辅助用户创建、编辑和验证复杂的编排模板，降低使用门槛；
- b) 提供中心化的模板仓库，支持团队存储、版本控制、共享和复用经过验证的最佳实践模板；
- c) 期货行情分析微服务则侧重于大数据处理能力，模板可包含分布式计算集群和数据仓库资源；
- d) 期货交易执行微服务需要高性能计算资源、低延迟网络环境和大容量存储支持，可制定包含多台高性能服务器、高速网络交换机和分布式存储的编排模板；
- e) 开发人员可根据业务需求直接调用模板，快速构建微服务运行所需的资源环境。

6.2.2 跨资源类型的协同编排

实现不同类型资源（硬件资源、虚拟化资源、云资源等）的协同编排。

基础级能力要求如下：

- a) 保证跨资源操作的原子性（全部成功或回滚）或提供明确的错误处理和补偿机制，减少不一致状态；
- b) 将编排指令分发到不同的底层基础设施（本地集群、公有云）并协调其上的资源部署，实现统一控制面下的混合编排。

增强级能力要求如下：满足基础级能力要求。

6.2.3 编排流程自动化与可视化

编排流程可自动化执行，可视化呈现。

基础级能力要求如下：

- a) 根据编排模板自动执行资源的创建、配置和关联；
- b) 支持在部署失败或检测到问题时自动或一键触发回滚到上一个已知良好状态；

- c) 提供界面查看当前部署状态（如哪些资源已创建、状态是否健康）与期望状态的对比。

增强级能力要求如下：

- a) 提供详细的编排执行日志和时间线视图，清晰展示每个步骤的执行情况、耗时和结果，便于排障和审计；
- b) 通过图形化或脚本化方式定义复杂的、多步骤的编排 workflow（如金丝雀发布、蓝绿部署流程），将部署、测试、验证、切换等步骤串联并自动化。

6.3 资源动态分配

6.3.1 基于负载的动态资源调度

实时采集微服务的运行负载数据。

基础级能力要求如下：

- a) 基于预设指标（如 CPU 利用率、内存利用率、自定义应用指标如 QPS）自动增加或减少微服务应用的实例数量，以匹配当前负载；
- b) 基于历史资源使用分析，自动生成 Pod 资源请求（Requests）与限制（Limits）的调整建议与策略，优化单个容器的资源利用率。

增强级能力要求如下：

- a) 基于多个指标组合（如 CPU 和 QPS）进行扩缩容决策，并支持指标之间的逻辑关系（AND/OR）；
- b) 用户定义使用更复杂的扩缩容算法和行为（如扩缩容步长、冷却时间、最大/最小实例数阈值、基于时间窗口的预测）。

6.3.2 业务优先级驱动的资源分配

根据期货业务的优先级划分资源分配策略。

基础级能力要求如下：

- a) 用户自定义优先级类，为 Pod 赋予更细粒度的优先级数值，更灵活地控制抢占和调度顺序；
- b) 为 Pod 设置不同的 QoS 等级，调度器和资源管理器依据 QoS 等级在资源紧张时优先保障高优先级服务的资源供给，并可能驱逐或限制低优先级服务；
- c) 在高优先级 Pod 需要资源而节点资源不足时，调度器可驱逐（抢占）节点上足够多的低优先级 Pod（需考虑优雅终止）以腾出资源；
- d) 在管控机制下，允许高优先级服务在必要时突破配额限制；
- e) 将期货交易、风控管理等核心业务设置为高优先级，在资源紧张时优先保障这些业务的资源需求。

增强级能力要求如下：满足基础级能力要求。

6.3.3 预测性资源分配

利用机器学习和数据分析技术，对期货业务的资源使用趋势进行预测。

基础级能力要求如下：

- a) 分析历史资源使用模式（如日/周规律、业务高峰），涵盖历史交易数据、行情波动规律等信息，预测未来负载，并提前进行资源预分配或预热（如提前扩容实例、预拉取镜像）；
- b) 响应已知的业务事件（如定时批处理任务），主动触发预定义的扩缩容计划。

增强级能力要求如下：

- a) 集成机器学习模型，利用更复杂的时序数据和外部因素（如节假日、天气）进行更精准的负载预测，指导资源分配决策。

7 微服务开发与运行层

7.1 服务开发

7.1.1 框架与工具适配

提供专为期货行业定制的微服务开发框架，集成期货业务专属的工具集。

基础级能力要求如下：

- a) 支持多种开发语言，如Java、Python等；
- b) 集成交易指令处理、风险计算、清算逻辑等业务的专属基础模块，支持开发人员直接复用，减少重复编码。

增强级能力要求如下：

- a) 利用AI辅助代码生成，根据业务描述自动填充模板节点；
- b) 框架内可内置交易指令处理、风险计算、清算逻辑等基础模块，允许开发人员可直接复用，减少重复编码。

7.1.2 领域模块与模板

以期货行业通用业务领域对开发模板进行设计。

基础级能力要求如下：无要求。

增强级能力要求如下：

- a) 预置标准化业务模板（交易、风控、清算、行情），指南接口定义（输入/输出参数）与代码结构，如期货下单微服务模板，预设好订单校验、合约匹配、资金冻结等流程节点；
- b) 模板支持自定义扩展，允许开发人员根据特定业务需求（如个性化风控规则）添加自定义节点或修改流程逻辑结构。

7.1.3 开发流程规范

制定从需求分析到服务上线的全流程开发规范。

基础级能力要求如下：

- a) 从需求分析到服务上线的全流程规范，明确代码评审、单元测试、集成测试标准，要求记录业务逻辑与技术实现文档；
- b) 使用自动化开发工具（如 CI/CD 流水线），实现代码提交后自动触发测试、合规检查与部署流程，提升开发效率。

增强级能力要求如下：满足基础级能力要求。

7.2 持续集成与发布（CI/CD）

7.2.1 自动化流水线构建

构建端到端的CI/CD流水线。

基础级能力要求如下：

- a) 端到端的流水线涵盖代码提交、代码编译、单元测试、集成测试、代码扫描、镜像

构建、部署等环节自动化串联；

- b) 流水线触发规则需支持双模式配置：代码提交（如 Git 分支推送、Merge Request）自动触发（默认开启主干分支，开发分支可按需配置）、手动触发（支持指定版本、目标环境），且触发记录需留存日志，便于追溯；
- c) 针对期货系统高可用性要求，流水线需适配微服务集群部署场景，支持多环境（开发、测试、预发布、生产）差异化配置；
- d) 提供声明式部署模板（如K8s YAML/Helm Chart），定义微服务的资源需求、环境变量、健康检查、网络策略等配置，支持一键执行标准化部署；
- e) 实现测试、预发布、生产环境的配置隔离与自动化部署，确保环境间资源策略（如CPU/内存配额）和依赖服务（数据库、消息队列）的一致性；
- f) 发布策略需支持多模式适配，包括蓝绿部署、金丝雀发布等；
- g) 使用可视化模板编辑器，通过拖拽组件生成部署流程，降低开发运维门槛。

增强级能力要求如下：

- a) 流水线需集成期货行业特殊合规校验环节，如代码中涉及客户信息、交易数据处理的模块，需自动触发数据脱敏合规检查，未通过检查的流程需暂停并生成合规报告。

7.2.2 版本管理

对微服务进行版本管理。

基础级能力要求如下：

- a) 需遵循语义化版本规范（如主版本号；次版本号；修订号），版本信息需包含版本号、变更内容（需关联需求 ID / 缺陷 ID）、测试结论（通过率、未覆盖场景说明）、部署时间、部署负责人、环境信息，支持追溯近年所有版本记录；
- b) 当部署后出现交易异常、系统故障等问题时，支持触发版本回滚，回滚流程需自动校验目标版本完整性（如依赖组件、配置文件匹配度），回滚后生成回滚报告，记录回滚原因、影响范围及后续优化措施。

增强级能力要求如下：满足基础级能力要求。

7.3 服务发现

7.3.1 动态发现

微服务实例可被动态发现。

基础级能力要求如下：

- a) 微服务实例启动时自动上报服务名称、网络地址、接口能力（如期货合约查询、下单操作）等元数据；
- b) 实时更新服务实例状态，当实例故障下线或恢复上线时，及时同步状态，保障调用方获取准确信息。

增强级能力要求如下：满足基础级能力要求。

7.3.2 发现策略优化

服务可依据多种策略进行优化。

基础级能力要求如下：

- a) 能够设定服务发现策略，如按地区（区分境内外期货交易节点等）、按负载（依据实例当前CPU、内存使用率）、按业务优先级（保障交易微服务优先调用低负载实

例)进行筛选服务实例,提升数据获取效率;例如,期货行情微服务调用,可优先选择距离近、负载低的实例,降低网络延迟,提升行情数据获取效率。

增强级能力要求如下:

- a) 能够根据期货业务周期(如开盘、收盘时段)动态调整发现策略,开盘高峰期优先保障交易微服务的低负载实例调用。

7.4 服务编排

7.4.1 可视化与代码化编排

通过服务编排进行业务场景的快速构建。

基础级能力要求如下:

- a) 提供可视化拖拽界面与DSL(领域特定语言)代码化编排双模式,支持微服务的构建,如客户信息微服务、资金校验微服务、交易执行微服务等;
- b) 可通过连线、配置参数等方式,组合微服务,快速编排构建完整业务流程,如期货下单流程。

增强级能力要求如下:

- a) 在设计阶段可对流程进行可视化预览或模拟运行,验证逻辑正确性;
- b) 满足复杂业务场景下的精细编排需求,比如自定义行情数据预处理与分发的编排逻辑。

7.4.2 流程管理与版本控制

对编排好的期货业务流程进行全生命周期管理。

基础级能力要求如下:

- a) 能够完成流程的创建、修改、停用、删除操作,提供流程定义仓库统一存储;
- b) 允许多版本流程共存(如V1生产环境运行,V2测试环境验证),支持按环境/流量比例灰度发布;例如期货业务流程优化时,新编排版本先在测试环境或部分交易节点试运行,验证无误后再全量替换,保障业务稳定过渡;
- c) 支持按交易节点、用户分组、业务类型分流灰度(如仅股指期货交易走新流程);
- d) 流程异常时可一键回退至历史稳定版本,保障业务连续性;
- e) 支持智能灰度决策,基于业务指标(如错误率<0.1%)自动扩大灰度范围至全量;
- f) 支持回滚影响预判,自动评估回滚风险,避免新流程中新增的服务未在旧流程中。

增强级能力要求如下:满足基础级能力要求。

7.4.3 动态调整与协同

业务流程可动态实时调整。

基础级能力要求如下:无要求。

增强级能力要求如下:

- a) 支持通过集成规则引擎或事件驱动架构(EDA),实现基于外部事件的自动流程调整,如行情波动剧烈时,自动触发风控微服务调用频率提升的编排调整;
- b) 支持流程定义的热加载,确保变更即时生效,不影响正在运行的流程实例。

7.5 通信协议

7.5.1 模式与格式定义

通信模式需适配机构要求。

基础级能力要求如下：

- a) 支持同步与异步两种通信模式，适配不同业务场景，并符合行业标准中对场外市场交易系统涵盖行情接口、订单接口、结算接口，以及内部接口涵盖账户管理、证券交易、咨询数据等接口的模式定义（参考JR/T 0155.1-2018、JR/T 0155.2-2018、JR/T 0155.3-2018、JR/T 0274—2023、JR/T 0248—2022、JR/T 0275—2023）；
- b) 具备统一的数据传输格式标准，如JSON、Protobuf等，确保微服务间数据结构清晰、可解析、可校验；
- c) 具备接口契约管理机制，实现通信协议的版本化、文档化与自动化校验；
- d) 具备多协议适配能力，如同时支持HTTP、RPC、MQTT、WebSocket等，满足不同性能与场景需求。

增强级能力要求如下：满足基础级能力要求。

7.5.2 可靠性保障

通信协议具备可靠性机制。

基础级能力要求如下：

- a) 在异步通信中需确保消息不丢失，接收方成功处理后返回确认，发送方未收到确认时触发重传；
- b) 同步通信设置超时重试、故障切换策略，当调用微服务遇网络波动或服务短暂不可用时，自动重试或切换备用实例；
- c) 重传策略可配置，包括重试次数、间隔时间、指数退避等，避免雪崩效应；
- d) 具备消息持久化机制，确保在服务重启或宕机情况下，未处理消息不丢失。

增强级能力要求如下：满足基础级能力要求。

8 微服务治理层

8.1 服务限流

8.1.1 限流策略制定

微服务架构需构建多层次的限流策略。

基础级能力要求如下：

- a) 使用多维度限流策略配置，如系统资源利用率（CPU、内存、线程池）、请求速率（QPS/RPS）、并发连接数、资源配额（如单位时间调用次数、数据量限制）；
- b) 按微服务粒度设置限流规则，允许为不同服务设定独立的限流阈值；
- c) 基于业务优先级的差异化限流策略，对核心服务（如期货下单、风控校验）配置更高的限流阈值，保障关键业务资源供给；
- d) 限流规则能够集中管理与统一配置，通过配置中心实现动态下发。

增强级能力要求如下：

- a) 限流能力可达到租户级/用户级，适用于多租户或机构客户场景，防止个别用户占用过多资源；
- b) 全局与局部限流具备协同机制，实现集群级与实例级限流的联动控制。

8.1.2 动态限流调整

限流策略可依据实时监控情况动态调整。

基础级能力要求如下：

- a) 基于实时监控数据，如微服务运行负载、系统资源利用率、业务流量趋势等数据，动态调整限流规则的能力，并实时生效；
- b) 在交易高峰期或突发行情场景下，自动提升核心交易类微服务的限流阈值，避免误限流导致业务中断；
- c) 当系统资源紧张时，自动降低非核心服务（如日志上报、统计分析）的限流阈值，实现资源动态再分配。

增强级能力要求如下：

- a) 结合历史流量数据与AI算法，提前预测流量高峰并预设限流策略；
- b) 限流策略具备灰度发布与回滚能力，便于验证策略效果并快速恢复。

8.1.3 限流算法应用

限流算法的使用应结合期货业务场景进行优化。

基础级能力要求如下：

- a) 支持使用主流限流算法，包括：令牌桶、漏桶、滑动窗口等，如可利用令牌桶算法控制交易指令请求频率，确保单位时间内处理的指令数量在系统承受范围内；通过滑动窗口算法统计短时间内的请求次数，实现精准限流，防止突发流量压垮微服务；
- b) 算法参数可配置，如令牌生成速率、桶容量、窗口大小等；
- c) 具备分布式环境下的集群级限流能力，基于 Redis 或分布式协调组件（如 ZooKeeper、Nacos）实现跨节点限流状态同步；
- d) 能够自定义限流算法插件机制，允许企业根据业务特性扩展专属限流逻辑；
- e) 配置限流触发后的响应策略，如返回特定错误码、记录日志、触发告警等。

增强级能力要求如下：满足基础级能力要求。

8.2 服务路由

8.2.1 路由规则管理

期货公司业务系统微服务的路由规则能被统一管理。

基础级能力要求如下：

- a) 统一的路由规则管理中心能实现路由策略的集中配置与管理；
- b) 能完成多条件路由规则配置，如服务名称、版本号、地理位置、请求来源、请求参数等；
- c) 能基于地理位置的智能路由，将用户请求导向延迟低、距离近的服务节点，提升响应速度；
- d) 路由规则能进行版本化管理与变更审计，确保可追溯、可回滚。

增强级能力要求如下：

- a) 具备规则优先级与冲突处理机制，明确多条规则匹配时的执行顺序；
- b) 具备图形化路由规则配置界面，降低运维与业务人员使用门槛。

8.2.2 智能路由决策

路由决策可通过智能化方式进行实现。

基础级能力要求如下：

- a) 基于实时服务状态进行动态路由，如依据服务健康度（心跳、存活状态）、实例负

载（CPU、内存、队列长度）、响应延迟与吞吐量等指标进行路由；

b) 具备权重动态调整机制，根据实例性能实时更新权重，实现更合理的流量分配。

增强级能力要求如下：

- a) 当某实例响应变慢或故障时，自动将后续请求切换至健康实例，实现故障隔离与高可用；
- b) 基于机器学习的智能路由预测模型，预测实例未来负载并提前优化路由路径；
- c) 支持亲和性路由（Affinity Routing），如会话保持、数据本地性优化，提升缓存命中率与处理效率。

8.3 服务容错

8.3.1 容错机制构建

微服务可通过熔断、降级、重试、隔离等机制实现故障隔离与服务自愈。

基础级能力要求如下：

- a) 具备完整的容错策略组合，包括：
 - 1) 熔断：当失败率超过阈值时自动熔断，防止雪崩；
 - 2) 降级：返回默认值或兜底逻辑，保障核心流程不中断；
 - 3) 重试：支持指数退避、随机延迟等策略，提升 transient 故障恢复能力；
 - 4) 隔离：限制资源占用，防止故障扩散；
- b) 具备容错策略按服务/接口级别配置，允许差异化设置。

增强级能力要求如下：

- a) 具备熔断状态的半开（Half-Open）模式，实现故障恢复后的试探性调用；
- b) 具备自适应熔断算法，基于实时指标动态调整熔断阈值。

8.3.2 故障自动检测

微服务运行状态可实时被感知，自动进行故障检测。

基础级能力要求如下：

- a) 具备心跳检测机制，实时感知微服务实例的存活状态；
- b) 具备主动健康检查，定期发起探测请求，验证服务可用性；
- c) 具备深度健康检查，不仅检查服务是否存活，还需验证核心业务接口（如下单、风控）是否正常响应；
- d) 具备故障检测结果与服务注册中心联动，自动剔除不健康实例。

增强级能力要求如下：

- a) 具备多维度健康评分模型，综合响应时间、错误率、资源使用等生成健康度评分；
- b) 具备故障预测能力，基于趋势分析提前发现潜在异常。

8.3.3 容错恢复处理

微服务在出现异常或故障时，可自动恢复。

基础级能力要求如下：

- a) 具备熔断恢复机制，在熔断超时后自动进入半开状态，尝试恢复调用；
- b) 具备降级恢复机制，当服务恢复正常或系统负载下降时，自动切换回正常处理逻辑；
- c) 具备容错事件的完整日志记录，包括熔断触发时间、原因、降级返回内容、重试次数与结果。

增强级能力要求如下：满足基础级能力要求。

8.4 负载均衡

8.4.1 负载均衡算法选型

可根据不同业务场景灵活选择或扩展负载均衡算法。

基础级能力要求如下：

- a) 具备多种负载均衡算法，包括轮询、加权轮询、随机、最小连接数、一致性哈希；
- b) 具备自定义负载均衡算法插件机制，允许企业扩展专属算法。

增强级能力要求如下：

- a) 能动态切换算法，根据运行时负载自动选择最优算法；
- b) 根据业务需求灵活选择负载均衡算法，如对请求处理耗时均匀的行情查询微服务，采用轮询算法平均分配流量；对交易处理微服务，依据实例性能设置权重，通过加权轮询算法将更多流量分配给高性能实例。

8.4.2 动态负载感知

负载情况可实时感知并进行资源分配优化。

基础级能力要求如下：

- a) 实时采集微服务实例的负载指标，包括 CPU 使用率、内存占用、请求响应时间等；
- b) 基于负载数据的动态流量调度，当某实例负载过高时，自动减少其流量分配；
- c) 具备负载预测能力，结合历史趋势预判未来负载变化；
- d) 支持负载均衡器集群部署，提升高并发场景下的处理能力与可用性；
- e) 与服务注册发现机制（如 Nacos、Eureka、Consul）深度集成，自动感知实例上下线，动态更新路由表。

增强级能力要求如下：

- a) 具备无状态负载均衡器设计，支持横向扩展与故障自动切换；
- b) 具备负载均衡策略的动态更新，无需重启即可生效；
- c) 具备边缘负载均衡能力，在靠近用户侧部署负载均衡节点，进一步降低延迟。

9 微服务支撑层

9.1 注册中心

9.1.1 高可用架构设计

注册中心是依据高可用架构及逆行设计。

基础级能力要求如下：

- a) 具备分布式集群部署模式，避免单点故障，注册中心在部分节点宕机时仍可正常提供服务；
- b) 具备数据多副本机制，实现服务注册信息的跨节点同步与持久化存储。

增强级能力要求如下：

- a) 具备跨机房/可用区部署能力，支持异地多活或灾备架构，提升整体容灾能力；
- b) 具备一致性协议（如 Raft、Zab），保障集群内数据一致性与节点选举的可靠性；
- c) 具备自动故障转移与脑裂防护机制，防止网络分区导致服务状态混乱。

9.1.2 元数据管理

微服务的注册发现需涵盖其元数据的信息。

基础级能力要求如下：

- a) 微服务关键元数据的集中管理，包括、服务名称、版本号、接口定义、网络地址、健康状态、权重、标签（Tag）等；
- b) 元数据的动态更新机制，服务实例上线、下线或配置变更时，实时同步至注册中心；
- c) 元数据版本控制，记录变更历史，支持回溯与审计；
- d) 多维度查询能力，支持按服务名、版本、标签、环境等条件检索服务实例。

增强级能力要求如下：

- a) 自定义元数据扩展字段，满足业务级标识（如“核心服务”、“VIP客户专用”）需求；
- b) 元数据变更通知机制，主动推送更新事件至订阅方。

9.2 配置中心

9.2.1 统一配置管理

微服务的配置能力可统一管理。

基础级能力要求如下：

- a) 集中式配置仓库，实现所有微服务配置的统一存储与管理，涵盖数据库连接参数、服务端口、业务规则配置等；
- b) 配置可按维度分类管理，包括环境（开发、测试、预发布、生产）、服务名称、业务模块、租户/客户（多租户场景）；
- c) 具备配置版本管理能力，记录每次变更的版本号、操作人、变更内容，支持回滚；
- d) 具备配置变更审计日志，满足合规与安全审计要求；
- e) 具备配置模板机制，支持通用配置的复用与继承。

增强级能力要求如下：满足基础级能力要求。

9.2.2 动态配置更新

配置规则可动态更新，满足期货业务规则调整（如手续费计算规则变更）、系统参数优化的快速响应需求。

基础级能力要求如下：

- a) 在不重启微服务进程的前提下，支持动态更新应用内部的配置值；
- b) 提供回调接口或事件监听机制，允许微服务在配置更新后执行自定义逻辑；
- c) 微服务实例能向配置中心或监控系统上报配置更新的结果（成功/失败），便于运维人员追踪变更的落地情况。

增强级能力要求如下：满足基础级能力要求。

9.2.3 配置权限管控

配置功能具备权限管控。

基础级能力要求如下：

- a) 支持基于角色的配置访问控制，可根据用户角色（如开发、测试、运维等）分别授予配置项的查看、修改及发布等不同权限；
- b) 通过加密通道（如HTTPS、TLS）与配置中心进行通信，防止配置信息在传输过程中被窃听或篡改；
- c) 支持记录微服务实例访问配置中心的详细日志（如服务名、实例ID、请求时间、请

求的配置项），并与配置中心的审计日志关联，形成完整的“谁在何时访问了什么配置”的审计链条。

增强级能力要求如下：

- a) 支持配置权限的细粒度控制，可按环境、命名空间、应用等维度灵活授权；
- b) 对期货交易、风控等核心业务的敏感配置（如交易限额、风险阈值），采用加密存储与双重认证机制。

9.3 日志与审计中心

9.3.1 日志采集与聚合

日志中心需完成分散日志信息的采集、汇聚与整合。

基础级能力要求如下：

- a) 规定统一的微服务日志的输出路径（如/var/log/app/*.log）和文件命名规范；
- b) 服务间调用时，自动将trace_id等分布式追踪标识注入到日志上下文中；当一个请求跨越多个服务时，所有相关服务的日志都能通过相同的trace_id被关联起来，实现跨服务日志的聚合；
- c) 基于规则过滤，以减轻日志中心的传输和存储压力；
- d) 使用常见日志格式（如JSON、纯文本等）的解析与字段标准化，并具备自定义解析规则的能力；
- e) 多租户场景下，日志采集与标签隔离，可按应用、环境等维度区分来源；
- f) 提供图形化配置界面，便于非运维人员便捷管理采集策略。

增强级能力要求如下：满足基础级能力要求。

9.3.2 日志存储与检索

日志的存储与检索需同步符合证券期货业业务系统的要求及规范。

基础级能力要求如下：

- a) 具备日志数据的分级存储策略，可自动将近期热数据存放于高性能存储，冷数据归档至低成本大容量存储；
- b) 提供跨节点、跨时间范围的高速日志检索，支持关键词、服务名、日志级别、时间范围等多维度组合查询；
- c) 安全级别为高的应用系统，日志保存时间建议至少为 36 个月；安全级别为中或者低的应用系统，日志保存时间建议至少为 12 个月；IT基础设施和系统的日志保存时间建议至少为 12 个月（参考JR/T 0233—2021《证券期货业经营机构内部应用系统日志规范》）。

增强级能力要求如下：

- a) 能导出、上下文浏览与可视化展示检索结果，辅助故障定位与业务分析；
- b) 在日志发送到采集器之前，支持对日志数据进行压缩（如gzip）和批量发送，以减少网络带宽占用和I/O开销，间接支持高效的数据传输至存储系统。

9.3.3 日志分析与告警

微服务需依托汇聚后的日志数据开展实时分析，精准识别异常并触发合规告警。

基础级能力要求如下：

- a) 能基于规则和模式进行日志实时分析，可识别错误、异常、高频事件等典型日志场景；

- b) 能设置灵活的多条件告警规则，并可通过邮件、短信、集成通知等方式及时触达相关人员；
- c) 提供日志统计、聚合分析及仪表盘展示能力，支持业务异常与系统性能的常态监控；
- d) 日志与链路、指标等数据能关联查询，实现全栈可观测性问题的闭环定位。

增强级能力要求如下：

- a) 提供告警收敛、依赖关联、分级推送等机制，减少误报与重复告警；
- b) 能基于机器学习算法实现日志异常自动检测与智能告警抑制。

9.3.4 日志审计

微服务架构下关键操作日志的审计合规能力，支持内控审计、监管检查与安全事件追责。

基础级能力要求如下：

- a) 所有涉及用户身份认证、权限变更、敏感数据访问、交易指令提交、配置修改的操作，应生成审计日志；
- b) 审计日志应与普通运行日志物理或逻辑隔离；
- c) 审计日志禁止修改、删除，仅支持追加与只读查询；
- d) 审计日志查询权限应严格受限，仅授权人员（如合规官、内审员、监管人员）可访问，且所有查询操作本身也需记录审计日志；
- e) 经营机构内部审计部门应按照本文件对日志的记录、转储、存储、备份、销毁过程按照审计规范进行审计，审计报告建议至少保存 3 年（参考JR/T 0233—2021《证券期货业经营机构内部应用系统日志规范》）。

增强级能力要求如下：满足基础级能力要求。

9.4 监控中心

9.4.1 全链路追踪

监控范围需覆盖期货业务微服务全链路。

基础级能力要求如下：

- a) 能全链路调用追踪（Tracing），从请求入口到服务响应出口，追踪每个微服务调用链路，记录请求耗时、参数传递、服务间调用关系等信息；
- b) 支持通过可视化链路图展示微服务调用拓扑，直观呈现系统运行状态，方便排查期货交易、行情推送等复杂业务流程中的性能问题；
- c) 能在微服务中自动或手动埋点，捕获关键业务流程的执行数据；
- d) 能采集多维度指标，包括：性能指标（响应时间、吞吐量、错误率）、资源指标（CPU、内存）、业务指标（交易笔数、风控触发次数、行情更新频率）。

增强级能力要求如下：

- a) 能进行日志聚合与结构化处理，便于与监控系统联动分析；
- b) 能实时展示微服务间的调用关系与流程执行路径；
- c) 能使用无侵入式探针技术（Agent-based），减少对业务代码的改造。

9.4.2 实时监测与预警

对关键指标进行实时监控及告警。

基础级能力要求如下：

- a) 具备实时监控仪表盘，可视化展示微服务运行状态与关键指标；
- b) 具备多级预警机制，支持通过邮件、短信、企业微信、钉钉等方式发送告警；

c) 可配置告警阈值，如响应时间、错误率、熔断触发次数等。

增强级能力要求如下：

- a) 能进行告警去重、抑制与升级机制，避免告警风暴；
- b) 具备智能异常检测算法（如基线预测、趋势分析），实现潜在故障的提前预警。

9.4.3 故障定位

基于监控数据，实现故障快速定位。

基础级能力要求如下：

- a) 调用链路能够可视化展示，清晰展示服务间调用关系，快速定位性能瓶颈或故障节点；
- b) 对指标与日志、链路进行关联分析，实现故障问题的根因分析。

增强级能力要求如下：

- a) 进行历史数据回溯与对比分析，用于性能趋势判断与容量规划；
- b) 定期生成系统健康报告，包含性能瓶颈、调用热点、资源利用率等；
- c) 具备自动化诊断建议功能，基于监控数据推荐优化方案（如扩容、代码优化、缓存策略调整）。

9.5 服务安全

9.5.1 权限管理

服务权限进行细粒度设计。

基础级能力要求如下：

- a) 依据期货业务角色（交易员、风控员、系统管理员等）、功能权限（下单、撤单、风控规则配置、系统参数修改等），设计细粒度服务授权模型，实现对微服务接口级的访问控制；
- b) 为每个微服务接口打上权限标签，并与用户角色/属性进行匹配校验，如期货下单微服务的下单接口，仅开放给具备交易权限的角色调用；
- c) 使用动态策略引擎，实现复杂授权逻辑的灵活配置与外部管理；
- d) 对期货交易、风控等核心微服务，可结合双向证书认证，保障调用双方身份真实可信；
- e) 具备多因素认证（MFA）集成能力，提升高敏感操作的安全性；
- f) 权限策略能够动态更新，无需重启微服务即可生效；
- g) 权限变更具备实时通知机制（如通过配置中心推送），确保所有服务节点及时加载最新策略；
- h) 具备权限变更审计日志，记录谁在何时修改了哪些权限，满足合规与追溯要求。

增强级能力要求如下：满足基础级能力要求。

9.5.2 通信与数据安全

服务间通信与数据传输需保证安全。

基础级能力要求如下：

- a) 所有跨网络微服务调用默认启用 TLS 加密传输，禁止明文通信，尤其在跨机房或云边协同场景；
- b) 敏感数据（客户身份信息、银行账号、交易密码）在存储与日志中须加密或脱敏处理，符合《个人信息保护法》要求；

- c) 关键业务数据（如委托单、成交回报）传输过程启用完整性校验机制（如数字签名），防止篡改；
 - d) 配置中心、注册中心等核心组件的敏感配置项（如数据库密码、API密钥）采用加密存储，运行时动态解密；
 - e) 数据安全管理与保护可遵循JR/T 0250—2022《证券期货业数据安全管理与保护指引》相关要求；
 - f) 对涉及敏感数据或核心业务的微服务，应在架构设计阶段评估在基础设施层面实施物理隔离（如独立网络、独立存储设备）的必要性，以防范软隔离或权限控制机制在极端情况下失效带来的风险。具体可参照《证券公司信息隔离墙制度指引（2019年修订）》等相关规范；
 - g) 对于不支持动态发现或采用私有通信协议的传统单体架构系统，应通过统一的API网关或专用适配器进行对接。所有内部服务（无论新旧架构）对集群外部的暴露必须经过统一网关，并在此层面实施强制性的域名与接口白名单访问控制，严格防范通过Host Header注入等手段绕过边界防护、非法访问内部未公开服务的安全风险。
- 增强级能力要求如下：满足基础级能力要求。

9.5.3 安全开发与漏洞管理

服务需进行安全开发，并对漏洞进行管理。

基础级能力要求如下：

- a) 建立漏洞响应机制，对发现的高危漏洞（如Log4j类事件）支持热修复或快速回滚；
- b) 微服务依赖的第三方库版本受控，在使用第三方组件时宜充分考虑第三方组件的安全性和合规性（参考JR/T 0295—2023《证券期货业信息安全运营管理指南》）；
- c) 集成静态代码扫描（SAST）与软件成分分析（SCA）工具，在CI/CD流水线中自动检测漏洞与开源组件风险。

增强级能力要求如下：

- a) 建立微服务安全编码规范，禁止硬编码密钥、禁用不安全函数、防范常见漏洞（SQL注入、XSS）；
- b) 关键服务（如交易、清算）上线前建议通过第三方渗透测试与安全评估。

10 业务公共服务层

10.1 监管协同服务

通过数据报送引擎与监管规则中台，实现监管数据自动化合规报送与监管规则实时同步、验证，满足证监会等机构监管要求。

基础级能力要求如下：

- a) 满足多监管机构（如证监会、协会、监控中心）的差异化数据报送要求，提供可动态配置的报表模板引擎，支持XML/JSON等多种输出格式；
- b) 提供报送流程自动化管理，包括数据预校验（字段完整性、业务逻辑）、国密SM2加密与签名、基于FTP/SFTP的可靠传输及断点续传，保障报送成功率不低于99.9%。

增强级能力要求如下：

- a) 能够实时接收与处理监管机构发布的风险参数与规则变更，实现业务逻辑热加载，生效延迟不超过5分钟；
- b) 提供规则模拟验证环境（沙箱），支持业务人员对拟生效规则进行仿真测试，并输

出合规性与风险评估报告；

- c) 支持监管政策库的版本化管理与语义化检索，便于业务人员跟踪和理解监管要求；
- d) 提供报送监控看板，实时展示报送任务状态、失败原因及重试机制，支持自动告警与人工介入处理；
- e) 规则变更影响范围可以自动分析并可视化，提升规则发布的可靠性和可控性。

10.2 数据治理服务

依据证券期货业数据相关标准，围绕主数据管理与敏感数据处理，构建规范、安全的数据治理体系。

基础级能力要求如下：

- a) 提供企业级主数据管理能力，至少覆盖客户、账户、合约三类主数据，实施统一编码规范与数据标准，并支持按流程发起主数据变更，记录完整审计日志；
- b) 数据标准遵循业务属性、技术属性、管理属性的架构进行标准规范（参考JR/T 0319—2024《证券期货业数据标准属性框架》）；
- c) 按照业务类型进行数据类型的划分，一般划分为交易、监管、信息披露、其他（参考JR/T0176.3—2021《证券期货行业数据模型》）。

增强级能力要求如下：

- a) 关键数据血缘追溯，可分析主数据变更对业务规则和风控策略的潜在影响；
- b) 具备敏感信息自动识别与保护机制，对PII数据实施最小化采集和存储，落地加密与脱敏策略；
- c) 按监管要求设置数据归档策略，归档前执行完整性校验（如MD5比对），失败时触发人工干预流程；
- d) 基于机器学习的数据质量自动侦测与清洗建议；
- e) 提供数据资产地图与血缘可视化工具，提升数据可信度和使用效率；
- f) 多级数据归档策略及低成本冷热数据分层管理，兼顾合规与存储经济性。

10.3 流程协同服务

提供可视化流程设计、事件一致性保障及分级容灾切换能力，确保业务流程高效运转与核心业务连续可用。

基础级能力要求如下：

- a) 提供低代码可视化流程设计能力，支持跨多个业务域的工作流编排，包括并行分支、条件路由和人工审批节点；
- b) 基于本地事务表与异步补偿机制，保障分布式环境下关键业务流程的最终一致性，发生异常时支持自动回滚或恢复；
- c) 核心业务分级容灾策略，对高等级业务（如结算、报送）实现秒级故障自动切换与数据补偿对账；
- d) 流程版本管理与灰度发布，可在不影响线上业务的情况下部署和测试新流程。

增强级能力要求如下：

- a) 提供流程运行监控与性能分析看板，实现对流程实例状态、耗时及异常情况的实时可视化管理；
- b) 流程节点效率度量和瓶颈分析，为持续优化业务流程提供数据支撑。

附 录 A

（资料性附录）

应用案例

A.1 智能体开发平台

某期货公司智能体开发平台基于微服务架构建设，是一套为智能体的设计、开发、部署、运维、迭代提供全生命周期支持的工具链与基础设施；其采用微服务架构实现，本质是将智能体开发全流程中的复杂功能拆解为独立、可扩展的服务单元，通过标准化接口协同工作，最终满足企业、开发者对智能体的定制化、敏捷化开发需求；基于这一架构，平台无需从零开发智能体核心能力（如记忆、工具调用、任务规划），而是通过“搭积木”的方式实现快速迭代，可按需接入新的大模型（如DeepSeek、通义千问）、工具（如 CRM 系统、数据分析工具）或业务系统；

在功能解耦层面，智能体开发涉及“NLP 对话、技能组件管理、权限控制、部署运维”等多维度能力，微服务可将这些能力拆分为对话引擎服务、技能组件服务等独立模块，避免了单体架构“牵一发而动全身”的问题，让各功能模块的更新与维护更具独立性；从独立扩展来看，不同场景下智能体的负载差异较大，客服对话类智能体需应对高并发流量，办公辅助智能体则更注重运行稳定性，微服务架构支持“按需扩容”，例如仅针对对话引擎服务进行资源扩充以应对峰值请求，有效减少不必要的资源浪费；在技术栈选择上，各模块可根据自身功能需求选用适配的技术方案，像对话引擎可采用 Python/TensorFlow 组合以适配自然语言处理需求，数据存储模块则支持多种数据源及多种接入方式，实现高效数据管理，无需强制统一技术栈，大幅提升了各模块的开发效率与适配性；而在容错与高可用保障方面，即便单个微服务出现故障（如OCR功能服务暂时下线），也不会导致整个平台陷入崩溃，通过熔断、降级等机制，能优先保障对话交互等核心功能的正常运行，减少故障对整体开发流程的影响；

围绕微服务架构的设计思路，该平台已完成核心业务模块的拆分，具体涵盖用户与权限服务、智能体设计编排服务、技能与插件服务、对话引擎服务、数据存储与分析服务以及业务集成服务；这些模块通过标准化接口相互协同，共同支撑起智能体从设计到运维的全生命周期开发需求，为企业与开发者提供稳定、灵活的智能体开发支撑。

A.2 业务运营平台项目

某期货公司业务运营平台项目在技术上采用微服务架构构建，基于领域驱动的思想设计，借助容器化技术编排部署；通过微服务架构，将复杂的业务系统拆分为松耦合的服务单元，每个服务聚焦特定业务域，既能独立开发、测试与部署，又能通过标准化接口协同工作，有效降低系统复杂度，支持团队并行开发，提升迭代效率；

基于领域驱动的设计思路，平台深入梳理期货业务协作场景中的核心领域（如客户协作、信息流转、任务管理等），通过限界上下文明确领域边界，构建贴合业务逻辑的领域模型与聚合关系，使代码结构与业务语言高度对齐，不仅便于业务人员与技术人员的高效沟通，更能灵活响应业务规则的调整与新需求的拓展；

而容器化编排部署则借助 Kubernetes 等工具，将各微服务封装为标准化容器，实现环

境一致性与资源的动态调度，可根据业务负载自动扩缩容，保障高峰期系统稳定性；支持蓝绿部署、金丝雀发布等策略，降低版本迭代的风险；同时通过统一的编排管理，简化多环境部署流程，提升运维效率，为平台的高可用、可扩展提供坚实的基础设施支撑。

参 考 文 献

- [1] 《证券期货业场外市场交易系统接口 第 1 部分：行情接口》（JR/T 0155.1—2018）2018 年由中国证券监督管理委员会发布，自发布之日起施行。
- [2] 《证券期货业场外市场交易系统接口 第 2 部分：订单接口》（JR/T 0155.2—2018）2018 年由中国证券监督管理委员会发布，自发布之日起施行。
- [3] 《证券期货业场外市场交易系统接口 第 3 部分：结算接口》（JR/T 0155.3—2018）2018 年由中国证券监督管理委员会发布，自发布之日起施行。
- [4] 《证券期货业机构内部接口 账户管理》（JR/T 0274—2023）2023 年由中国证券监督管理委员会发布，自发布之日起施行。
- [5] 《证券期货业信息安全运营管理指南》（JR/T 0295—2023）2023 年由中国证券监督管理委员会发布，自发布之日起施行。
- [6] 《证券期货业数据标准属性框架》（JR/T 0319—2024）2024 年由中国证券监督管理委员会发布，自发布之日起施行。
- [7] 《证券期货行业数据模型》（JR/T 0176.3—2021）2021 年由中国证券监督管理委员会发布，自发布之日起施行。
- [8] 《证券期货业经营机构内部应用系统日志规范》（JR/T 0233—2021）2021 年由中国证券监督管理委员会发布，自发布之日起施行。
- [9] 《证券公司信息隔离墙制度指引（2019 年修订）》2019 年 11 月 8 日由中国证券业协会发布，自发布之日起施行。
-