

零刻秒杀商城

B2B2C 多商户高并发秒杀电商平台

- ✓ 用户端、商家端、管理端形成完整业务闭环
- ✓ 核心解决高并发、超卖、异步下单、风控和 AI 客服
- ✓ 整合商品识图、RAG AI 客服、智能告警等能力
- ✓ 关键规模：约 50 张数据表、5 个状态机、14 个权限码



高并发秒杀

Redis Lua + 队列削峰
支撑大流量秒杀场景



完整业务闭环

商品 → 下单 → 支付 → 售后
全链路打通



企业级可靠性

熔断降级 / 容灾容错
可观测 / 可运维



智能化体验

RAG AI 客服 + 商品识图
更懂用户，更高效服务

目录

— 零刻秒杀商城 · 汇报结构

01

项目概述



平台定位、工程模块、基础设施

P.02

02

系统功能展示

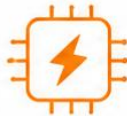


用户端、商家端、管理端闭环

P.03-05

03

核心技术亮点



秒杀链路、限流库存、订单一致性、AI 与监控

P.06-15

04

总结



价值回顾与后续规划

P.16

从业务场景到技术闭环

项目概述

P.02

从平台定位到工程全景

01



平台定位

B2B2C 秒杀商城

02



工程全景

用户端 / 商家端 /
管理端 / 后端 API

03



基础设施

MySQL · Redis ·
MQ · MinIO · InfluxDB

多端协同, 覆盖全角色



后端 API 核心

业务服务 / 网关 / 认证 / 权限 / 配置



服务治理

注册 / 配置 / 熔断

统一网关
高内聚、
易扩展

基础设施

MySQL
关系型数据库Redis
缓存 / 分布式锁RabbitMQ
消息队列MinIO
对象存储InfluxDB
时序监控存储稳定可靠
高性能支持

先建立全局地图, 再进入功能与技术细节



四个工程模块 + 六类基础设施支撑完整平台



三端前端统一接入后端 API，底层设施支撑秒杀、电商、AI 与监控

三端前端



用户端 client-web
React 19 / TypeScript / Vite
面向消费者



商家端 merchant-web
React 18 / Ant Design 5
商家与员工



管理端 admin-web
React 18 / Ant Design 5
平台管理员



后端模块 seckill-mall-system

Spring Boot 3.5 / Java 21 / 统一 REST API / 端口 8080

client API

merchant API

admin API



MySQL



Redis



RabbitMQ



InfluxDB



MinIO



Milvus



Vision Server

→ 业务数据

→ 缓存/锁

→ 消息队列

→ 监控数据

→ 图片/文档

→ 向量检索

→ 视觉识别



平台不是单一秒杀接口，而是由前端、后端与基础设施共同组成的完整工程系统。

系统功能展示

P.03-05

用户端、商家端、管理端的业务闭环



从页面功能看业务闭环，从角色视角看系统价值

用户端把发现、抢购、支付和售后串成完整购物链路

从发现好物到安心售后，打造丝滑、可靠、智能的购物体验



智能体验



AI 客服
RAG 知识问答
智能咨询



拍照识图
商品识别
相似推荐



3D 商品模型
360° 预览
真实可视化



图片放大镜
细节放大
精准查看



覆盖用户完整购物旅程，提升转化、满意度与复购率



商家端把经营动作沉淀成可授权、可追踪的管理流程



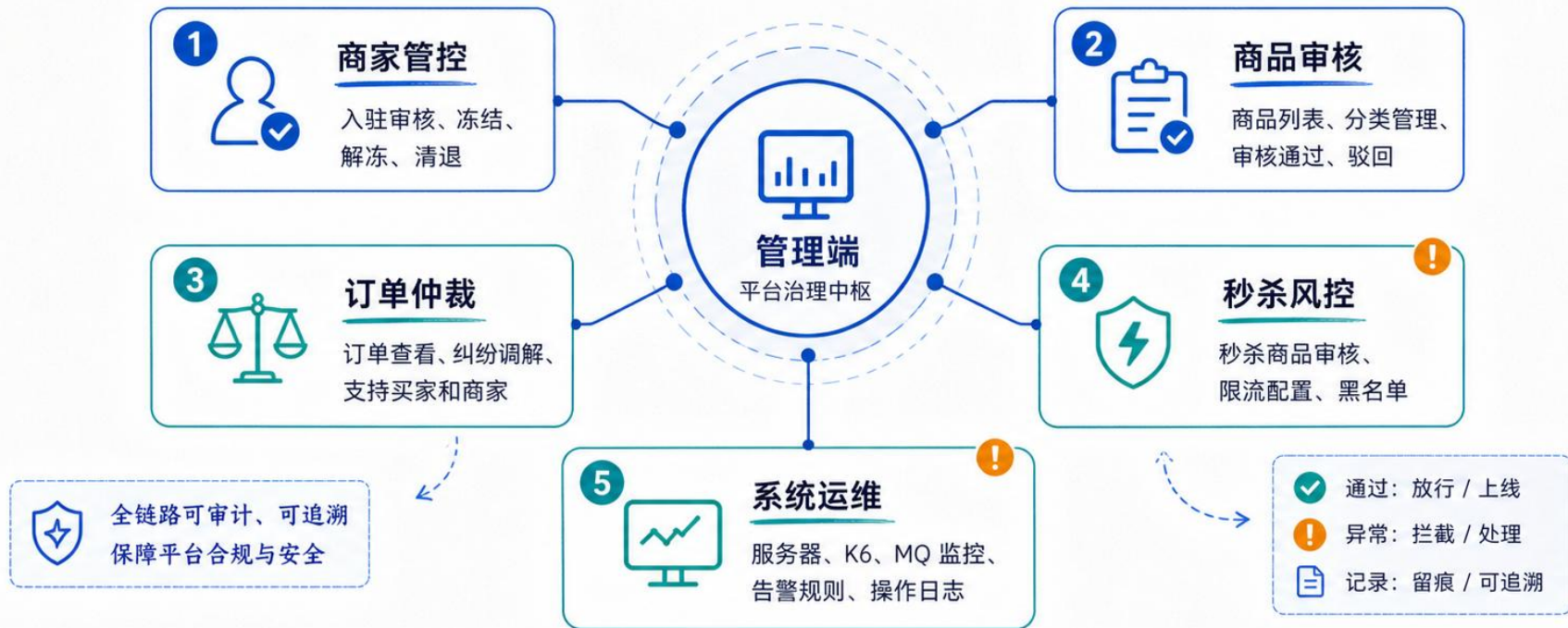
商家端不是“功能堆叠”，而是把商品、订单、售后、AI、财务和权限串成 **经营闭环**。



管理端聚焦平台治理：审核、仲裁、风控、财务和监控



统一治理标准 → 保障平台公平 → 降低运营风险 → 提升平台健康度



通过标准化治理体系，实现 **公平有序** + **风险可控** + **运营高效** + **数据透明** 的平台生态。

03

核心技术亮点

高并发秒杀、状态一致性与智能化能力

P.06-15



P.06

秒杀链路



P.07

限流库存



P.08

订单一致性



P.09

状态机



P.10-15

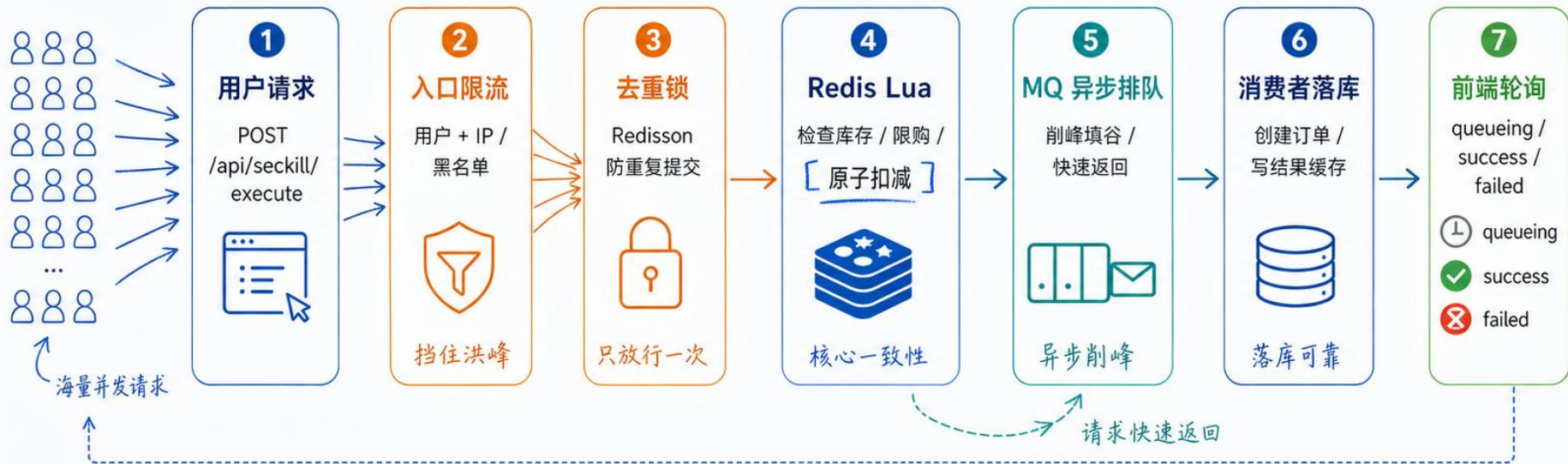
AI 与监控



这一章是汇报主体：从性能、可靠性到智能化闭环

秒杀完整链路

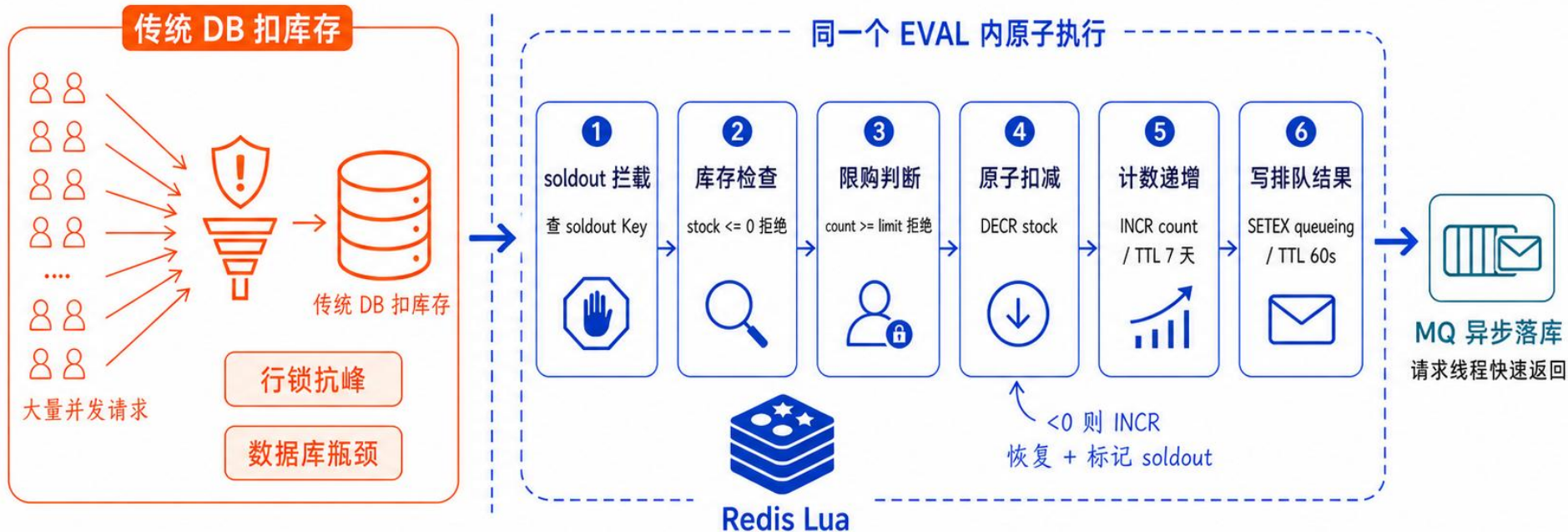
一次秒杀请求经过挡流、抢资格、排队和结果轮询



抢到的是 **下单资格**；高频竞争留在 Redis，订单落库交给 MQ 异步削峰。

≡ Lua 把库存检查、限购判断和扣减合并成一次原子操作 ≡

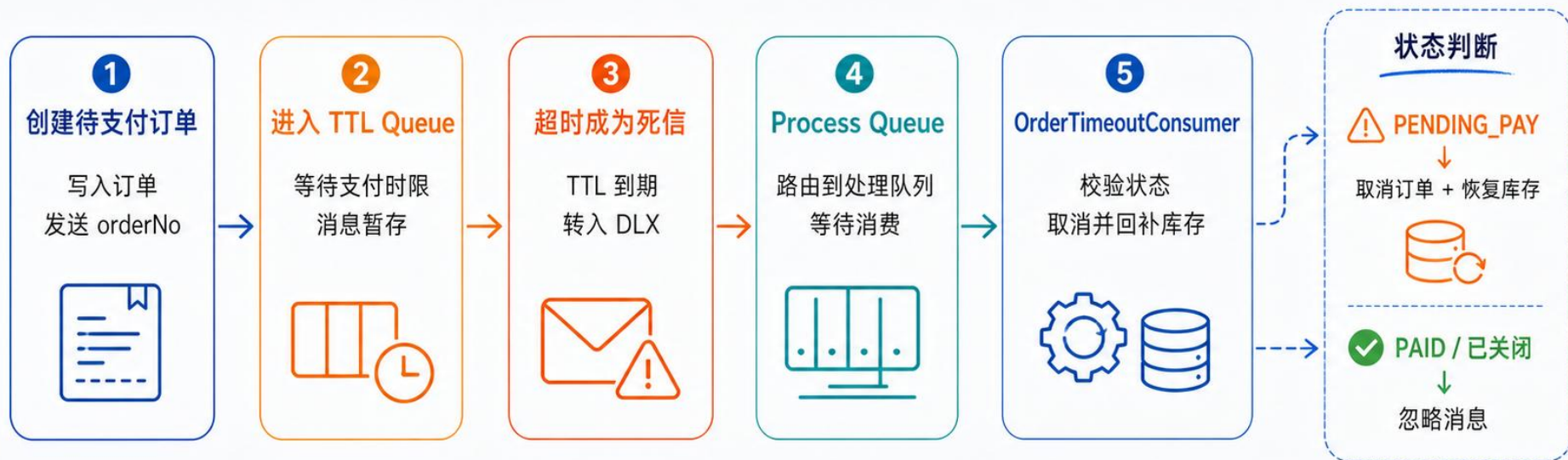
Redis Lua 在单线程模型中原子执行，避免高并发库存竞争打到数据库



最热的库存竞争留在 **Redis**；最重的订单写入交给 **MQ**。

订单超时取消依靠 TTL + DLX 完成最终一致性兜底

RabbitMQ 延迟队列让未支付订单自动释放库存



兜底逻辑只处理“仍未支付”的订单；消息可能延迟，但库存最终会被可靠释放。

枚举守卫 + Spring StateMachine 双重校验业务状态流转



状态字段仍保存在数据库 Integer 字段，兼容现有表结构。



枚举方法如 canPay()、canShip()、canCancel() 做快速业务校验。



Spring StateMachine 定义权威状态流转关系，避免 Service 层散落硬编码。



订单 (6 状态 + 6 事件)、商品 (3 状态 + 4 事件)、商家 (5 状态 + 5 事件)、售后 (9 状态 + 10 事件 + 3 分支路径)、AI 文档审核 (6 状态 + 5 事件) 共 5 类状态机。



例如：订单 PENDING_PAY(0) → PAID(1) → SHIPPED(2) → COMPLETED(3) 正常链路，CANCEL / REFUND / TIMEOUT 旁路。



状态流转不仅用于校验，也能结合 StateMachineAuditListener 自动审计日志形成追责闭环。

Enum.canXxx()

枚举守卫 (快速校验)

- ✓ canPay()
- ✓ canShip()
- ✓ canCancel()
- ...

StateMachine

Config

状态机配置
(权威流转)



Service 层统一状态治理



双重校验，更安全、更可靠！

以「订单」状态流转为例

正常链路



旁路：取消 / 退款 / 超时



订单状态机

6 状态 + 6 事件



- 事件 (6)
- PAY (支付)
 - SHIP (发货)
 - COMPLETE (完成)
 - TIMEOUT (超时)
 - CANCEL (取消)
 - REFUND (退款)



商品状态机

3 状态 + 4 事件



- 事件 (4)
- LIST (上架)
 - DELETE (删除)
 - TAKE_DOWN (下架)
 - RESTORE (恢复)



商家状态机

5 状态 + 5 事件

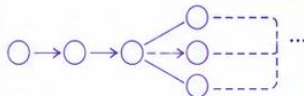


- 事件 (5)
- APPLY (提交)
 - REJECT (拒绝)
 - EXIT (退出)
 - APPROVE (通过)
 - SUSPEND (停业)



售后状态机

9 状态 + 10 事件
+ 3 分支路径



- 事件 (10)
- 申请 / 受理 / 协商 / 收货 / 检测 / 同意 / 拒绝 / 退款 / 换货 / 关闭



AI 文档审核状态机

6 状态 + 5 事件



- 事件 (5)
- SUBMIT (提交)
 - REJECT (驳回)
 - ARCHIVE (归档)
 - PASS (通过)
 - RESUBMIT (重新提交)

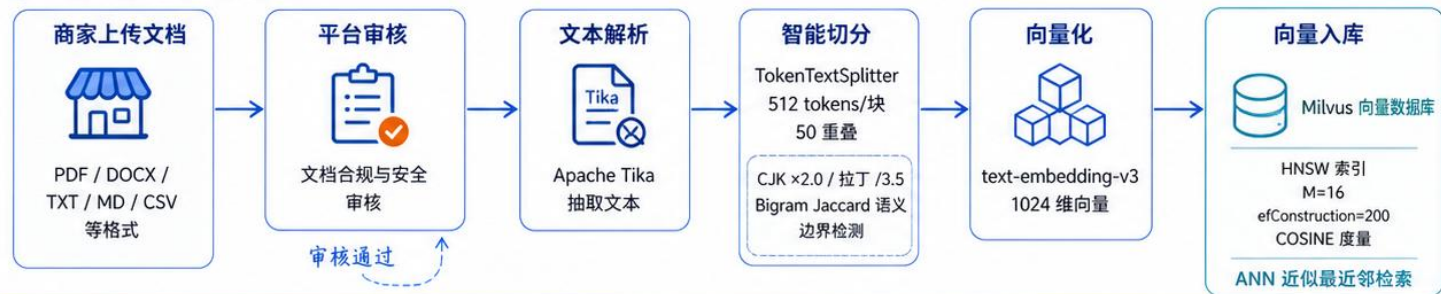


通过 枚举守卫 + Spring StateMachine 双重校验，构建 统一、可控、可审计 的业务状态流转体系！

校验更严、流转更稳、
审计可追，责任可追溯！

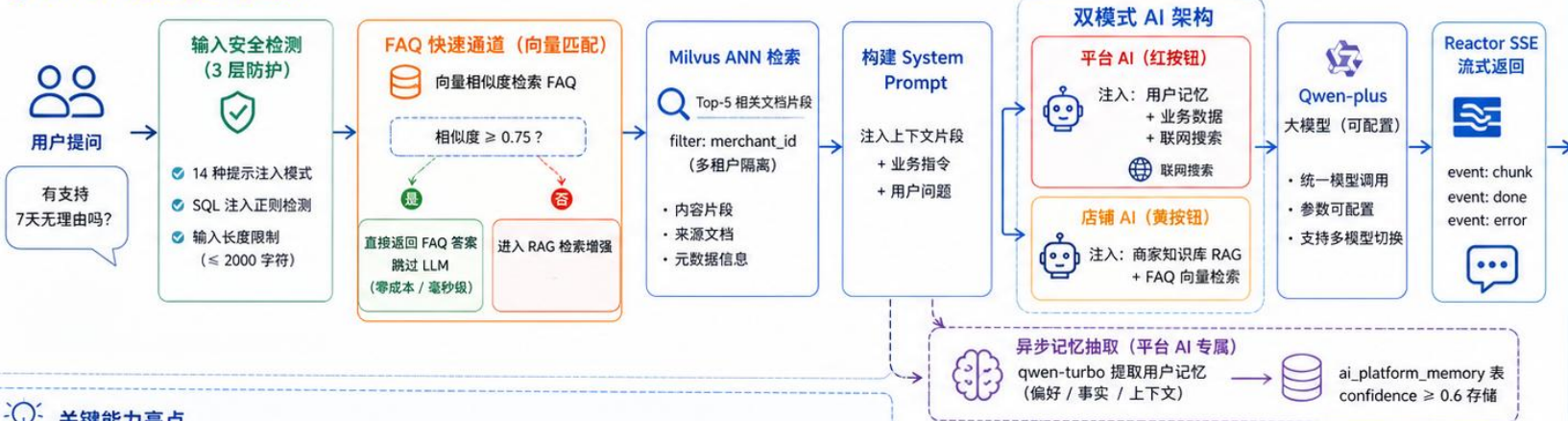
RAG 客服把商家知识变成可检索、可追踪、可审计的问答能力

1 知识库构建流程（离线）



- ✓ 多租户隔离
filter: merchant_id
- ✓ 可追踪
文档来源 + 片段元数据
- ✓ 可审计
操作日志 + 命中日志
- ✓ 安全合规
内容审核 + 敏感过滤

2 用户实时问答流程（在线）



核心参数配置

- 切分策略
512 tokens/块
50 重叠
- Embedding
text-embedding-v3
1024 维
- 向量数据库 (Milvus)
HNSW 索引
M=16
efConstruction=200
COSINE 度量
- FAQ 阈值
相似度 ≥ 0.75
- 检索 Top-K
Top-5
- 输入长度限制
≤ 2000 字符
- 记忆置信度阈值
confidence ≥ 0.6

关键能力亮点

- ✓ FAQ 零成本直达 毫秒级响应
- ✓ RAG 检索增强 高准确率
- ✓ 向量数据库 高性能 ANN 检索
- ✓ 双模式 AI 灵活可配
- ✓ 记忆沉淀 个性化体验
- ✓ 安全可控 可追溯可审计

图例说明

- 知识库构建（离线）
- 向量操作
- AI / 大模型
- 存储 / 数据库
- 检测 / 安全
- 用户问答实时流程

商品识图把图片转换为可检索的商品候选结果

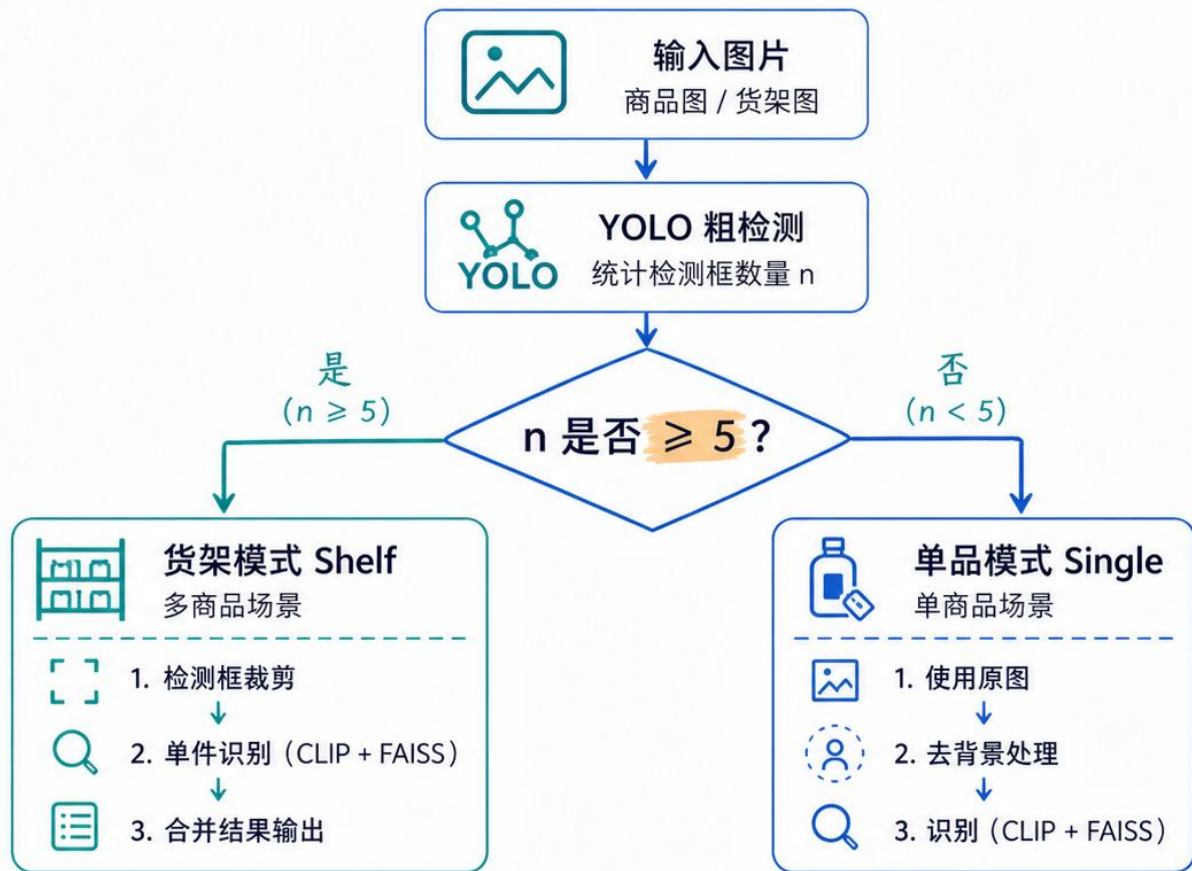
上传图片 → 目标检测 → 特征编码 → 向量检索 → 候选结果



通过 AI 识图流程，将用户图片快速转换为可检索向量，精准匹配相似商品，提升找货效率与转化率。



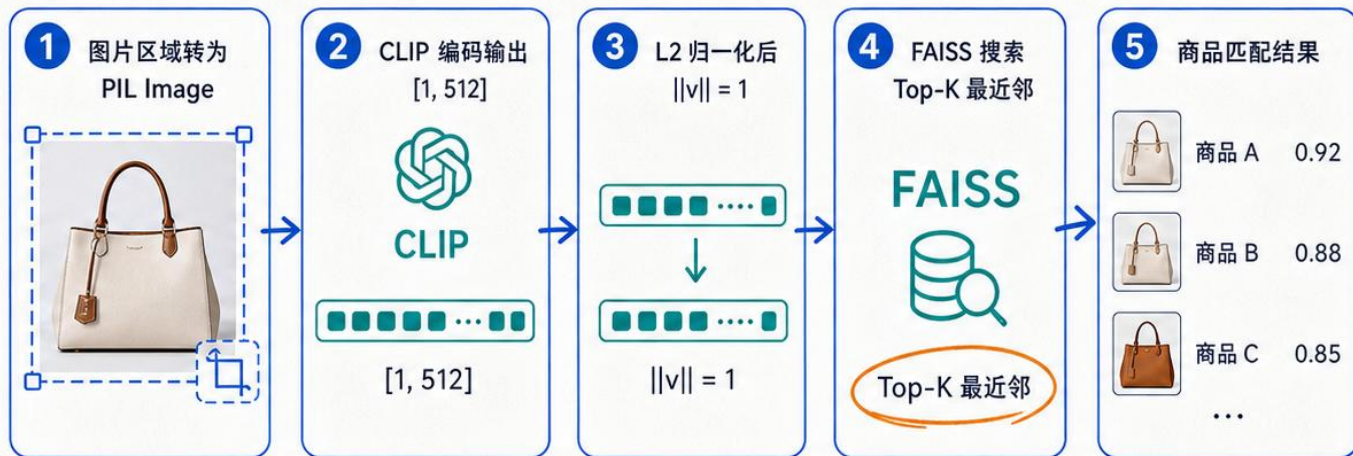
YOLO 分流让货架图和单品图走不同识别策略



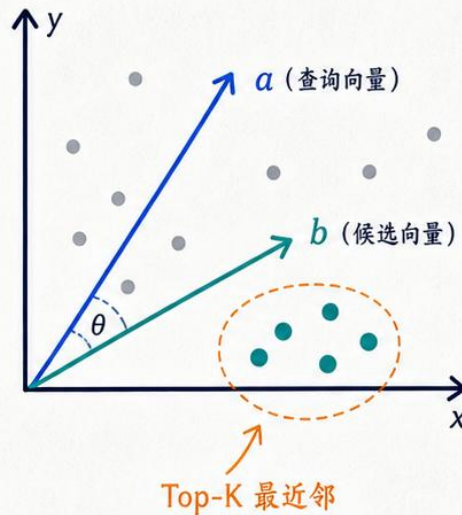
关键说明

- ✓ YOLO 粗检测统计检测框数量 n
- ✓ 判断条件: n 是否 ≥ 5
- ✓ 货架模式 Shelf: 多商品, 先裁剪再识别
- ✓ 单品模式 Single: 单商品, 原图 + 去背景
- ✓ 分流避免一种流程硬套所有图片

CLIP + FAISS 用向量相似度完成商品匹配



向量空间相似度示意 (二维空间)



向量相似度与距离关系

$$\cos\theta = a \cdot b$$
$$L2^2 = 2 - 2\cos\theta$$



- 向量已 L2 归一化后 ($\|a\| = \|b\| = 1$)
- $\cos\theta$ 越大 (越接近 1), 表示越相似
- 在归一化条件下, 最小的 L2 距离等价于最大的 $\cos\theta$



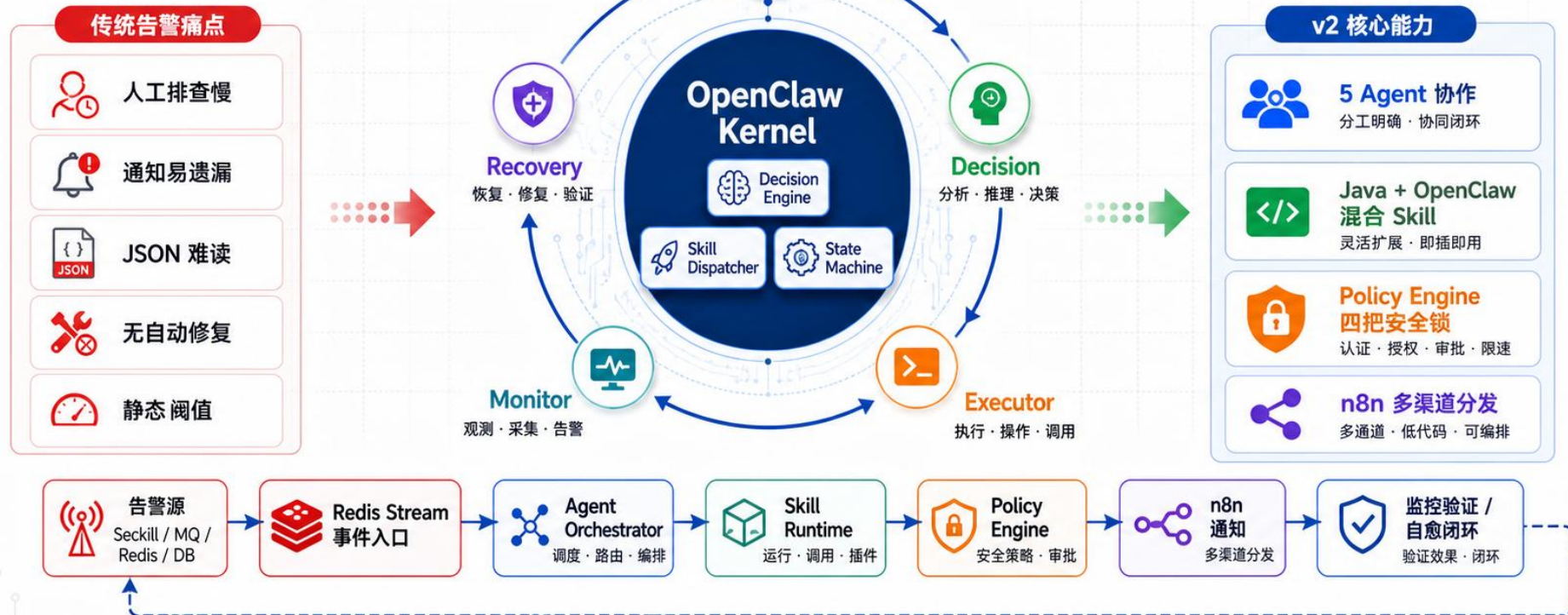
归一化后, **内积越大** 表示语义越相似, FAISS 按**相似度**返回候选商品。



AI 告警操作系统 v2

Multi-Agent 自愈型智能运维平台 | 零刻秒杀商城

🔄 从“人工运维”到“AI 自治”：OpenClaw 不只是工具，而是 **Kernel**。



⚠️ **Redis OOM 95% → 分析·修复·验证 < 5s**

总结

P.16

价值回顾与后续规划



≡ 零刻秒杀商城完成了从秒杀引擎到智能电商平台的闭环 ≡



业务上

三端协同覆盖用户、商家、平台管理员的核心流程。



技术上

高并发秒杀链路用 Redis、Lua、MQ 和补偿机制解决库存与订单压力。



治理上

RBAC、租户隔离、状态机、审核和操作日志保证平台可管。



智能化上

RAG 客服、视觉识图、推荐系统、用户记忆、智能告警提升用户体验和运维效率。

≡ 后续可扩展方向 ≡



从“能抢”到“能卖、能管、能运营、能智能服务”，构建可演示、可扩展、可观测的智能电商平台！