面试话术文档

一、自我介绍（开场 1 分钟）

C（Context）：

我目前从事物联网平台研发方向，有 5 年以上 Java 后端开发经验。主要负责设备接入、数据处理、规则引擎以及微服务架构相关工作。

R（Role）：

在团队中，我主要承担系统架构设计与性能优化相关工作，参与过从单体到微服务的演进，以及监控体系建设。

A（Action）：

项目主要基于 Spring Cloud、Kafka、Redis、Elasticsearch、SkyWalking 等组件构建，

我主导了高并发接入网关、消息异步处理、以及链路追踪与监控平台搭建。

P（Problem）：

早期平台存在设备状态延迟、消息堆积、系统可观测性差的问题。

S（Solution）：

通过引入 Kafka 削峰、Redis 缓存加速、SkyWalking 链路追踪以及 Nacos 配置中心治理，系统稳定性显著提升。

目前可支撑 50 万台设备同时在线，消息吞吐超过 20W TPS。

��关键词总结:

Java / Spring Cloud / Kafka / Redis / DDD / 性能调优 / 可观测性 / IoT 架构

二、面试官提问：你们是怎么实现设备实时状态同步的?

C:

我们的平台要实时显示设备在线状态和告警信息，要求延迟低于 3 秒。

R:

我负责设备接入网关与状态同步模块的设计。

A:

使用 **Netty** 维护设备长连接；

通过 **MQTT 协议 + Kafka** 进行消息中转；

设备状态变化时实时推送到 Redis；

后端再通过 WebSocket 同步给前端。

P：

当设备断电或掉线时，状态不同步，出现“假在线”现象。

S：

引入 **心跳检测 + TTL 机制**：

设备每隔 10 秒上报心跳，Redis 维护状态缓存并设置过期时间；

当 TTL 到期无心跳续期，则自动切换为离线状态。

最终实现了**“秒级同步”**，保证实时性。

三、面试官追问：Kafka 消息堆积是怎么处理的？

C：

在高并发设备接入场景下，Kafka 消息消费速度一度落后生产速度。

R：

我负责消费组与处理逻辑优化。

A：

增加 Kafka 分区数量；

优化消费组数量与并发线程；

引入批量消费与异步提交 offset；

在消费端添加幂等 key。

P：

早期因为重复消费与偏移量提交延迟，导致业务重复执行。

S：

通过批量消费 + Redis 幂等控制，重复消费率降到 0，

消息延迟由 5s 降至 1s 以内。

四、面试官：你们是如何做接口幂等的？

C：

物联网平台存在大量重复上报、接口幂等性要求高。

R：

我负责幂等机制的实现。

A：

前端请求添加 **唯一请求 ID（UUID）**；

网关层通过 **Redis SETNX** 校验是否重复；

对业务关键操作（如设备绑定）在数据库层使用 **唯一约束**；

Kafka 消费端通过 **业务 key 去重表** 实现幂等控制。

P：

部分操作在高并发时仍可能出现重复执行。

S：

完善分布式锁控制，并引入延迟队列重试机制，

保证接口在**高并发场景下仍能保持一次性执行**。

五、面试官：那你在性能调优上具体做了哪些？

C：

系统在高峰期响应时间达 800ms，GC 频繁。

R：

我负责性能瓶颈分析与调优。

A：

使用 **JProfiler** 分析 GC 和堆内存；

优化 JVM 参数（G1 GC、堆大小、线程数）；

Redis 做热点缓存 + Caffeine 本地缓存；

优化数据库 SQL 索引和批量写入。

P：

发现部分对象频繁创建导致 GC 压力过大。

S：

通过对象池与缓存策略优化后，接口平均响应降至 200ms，吞吐量提升 3 倍。

六、面试官：DDD 是如何在你们项目中落地的？

C：

设备、告警、规则等领域逻辑复杂，业务边界模糊。

R：

我负责主导 DDD 模型落地与领域划分。

A：

划分限界上下文（设备域、规则域、告警域）；

每个上下文独立聚合根与仓储层；

使用领域事件机制（DomainEvent）实现领域间解耦；

应用层 orchestrate 不同上下文。

P：

最初跨域调用复杂，代码改动容易破坏其他模块。

S：

通过限界上下文与聚合模型，

系统模块化程度提高，业务修改周期缩短 30%。

七、面试官：你们是怎么做可观测性和监控的？

C：

平台服务多，链路复杂，定位问题困难。

R：

我负责监控体系建设。

A：

接入 SkyWalking 进行链路追踪；

Prometheus + Grafana 做指标监控与可视化；

Jenkins + GitLab CI 做持续集成自动部署；

利用告警规则自动推送钉钉通知。

P：

早期出现接口超时但无法快速定位。

S：

接入监控后，平均故障定位时间从 30 分钟缩短至 3 分钟。

八、面试官：你平时在团队中主要承担什么角色？

C：

团队约 10 人，负责物联网核心平台。

R：

我属于技术核心成员，主要负责架构设计、性能调优、以及新人技术辅导。

A：

定期做架构评审与代码走查；

参与技术选型与 POC 验证；

推动监控体系、CI/CD 落地。

P：

早期技术栈混乱、代码质量不统一。

S：

推动统一规范后，团队研发效率提升约 40%，

Bug 数量减少显著。

九、面试官：那如果你加入我们团队，你能带来什么价值？

C：

我理解贵公司同样有大量设备接入、高并发场景与数据一致性挑战。

R：

我希望能在架构与性能优化上发挥经验优势。

A：

推动高并发架构优化（消息队列削峰、缓存分层、异步化处理）；

建立可观测性体系（链路追踪、告警监控、日志聚合）；

落地 DDD 建模，提升可维护性与扩展性。

P：

很多团队缺乏经验导致架构复杂度上升。

S：

我能帮助团队实现从“能运行”到“可扩展、可监控、可优化”的体系化提升。

十、收尾总结（结束语）

我非常喜欢做复杂系统的架构设计与性能优化，尤其在物联网场景中，
把成千上万设备的实时数据稳定地接入与处理，这种挑战让我非常有成就感。
如果有机会加入贵公司，我希望能在高并发架构、可观测性建设和微服务治理方向上继续深耕，
一起推动系统从稳定性、可维护性到智能化监控的全面升级。

面试官总结印象

能力全面：从业务理解到技术落地

表达结构清晰：每个问题都有背景与结果

可量化成果：性能提升、延迟降低、可用性提升

稳定自信、有技术深度