V3 -> V4展望的架构变化

V3存在的问题

1. 目前公司服务已经按照业务已经拆分，如收费、微信、安检等，但服务之间通信仍通过httpClient rest请求，服务间接口调用复杂，且没有保证事务原子性，导致被调用服务报错回滚，调用方未做处理时，提交了事务，导致数据不一致（如微信，POS出现的单边账问题）

2. 为了方便前端和服务与服务之间调用Logic接口，目前做法是把Logic名称和路径全部注册到redis中，再通过nginx转发，这种方式导致不同服务的Logic重名会覆盖路径，还需要人为控制服务前后启动顺序

3. 服务之间耦合性强，收费服务业务多压力大，POS、微信服务调用收费服务接口时经常会超时，导致调用方服务也会卡死

4. 服务共用同一个库，因为耗时业务导致的锁库会影响所有服务的运作（奥德导出报表锁库影响所有业务执行）

5. 服务与服务之间调用缺少监控，难以定位问题出现点，解决业务卡慢问题难度大，监控体系不健全

V4的解决方案

通过Nacos服务发现和注册，将所有服务统一管理起来，通过OpenFeign实现声明式服务间调用，通过Seata实现分布式事务

微服务架构中会设计一个Gateway网关在里面，像pc前端、H5等等，不用去关心后端有多少服务，所有请求都往网关走，网关会根据请求中的一些特征，将请求转发给后端的各个服务。

通过Sentinel对服务进行熔断和降级保护，避免被调用方服务瘫痪影响调用方服务，调用方在调用接口报错时，可以设置降级方案存下中间数据方便后续恢复

进行分库，避免长耗时业务影响其他服务的运行

Nacos 0.8.0版本完善了监控系统,有专业清晰的可视化界面，可以设置监控指标，还有自带的钉钉告警