

彻底解密企业级高可用高并发高可靠微服务架构 设计与实践（上）



主讲人：孙玄

2020.07.20

我的职业成长路线

NiX 奈学教育



奈学教育
奈学教育

创始人&CEO



转转

首席架构师
技术委员会主席
大中台技术负责人



58集团

技术委员会主席
高级系统架构师



百度

资深研发工程师



毕业

浙江大学



擅长领域

架构设计、大数据
机器学习、技术管理等



对外分享

业界顶级大会
百万年薪架构
直播大课品牌创始人

01

企业级微服务架构领域驱动设计 (DDD) 拆分方法论

02

企业级微服务架构实践方法论

03

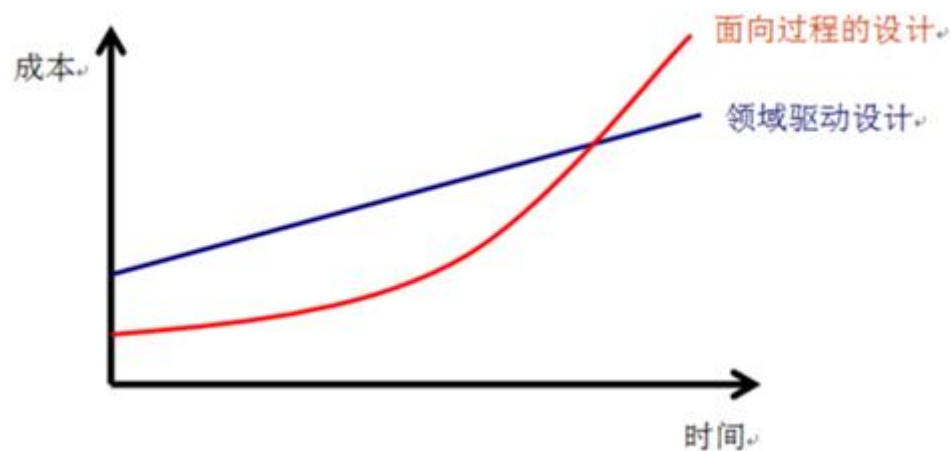
企业级微服务架构同步/异步业务场景案例剖析 (腾讯社交案例、阿里电商案例)



01.企业级微服务架构领域驱动设计（DDD）拆分方法论

- Domain Driven Design（领域驱动设计）

DDD不是贫血模型、充血模型、成为领域专家……



DDD保证服务不会变成Shit，使公司降本增效

- Domain Driven Design（领域驱动设计）

DDD本质是服务高内聚低耦合

DDD落地方式按照领域的拆分

- Domain Driven Design（领域驱动设计）

Entity实体->Model

AggregateRoot聚合->Logic

Service服务->Controller服务

- Domain Driven Design（领域驱动设计）

微服务架构如何落地？

业务侧：领域驱动设计

架构侧：水平拆分设计

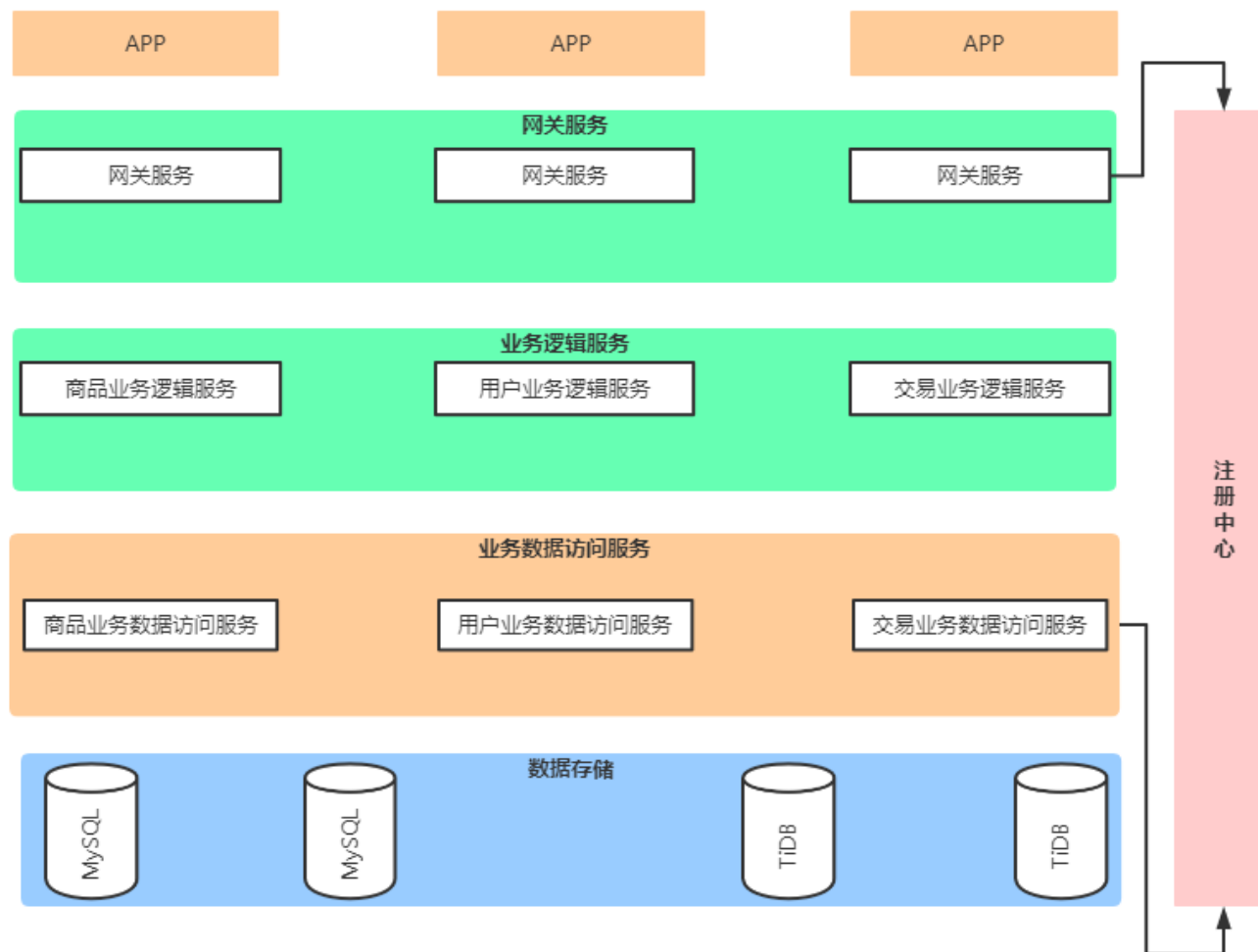


02.企业级微服务架构实践方法论

In short, the microservice architectural style [1] is an approach to developing a single application as a suite of small services, each running in its own process and communicating with lightweight mechanisms, often an HTTP resource API. These services are built around business capabilities and independently deployable by fully automated deployment machinery. There is a bare minimum of centralized management of these services, which may be written in different programming languages and use different data storage technologies.

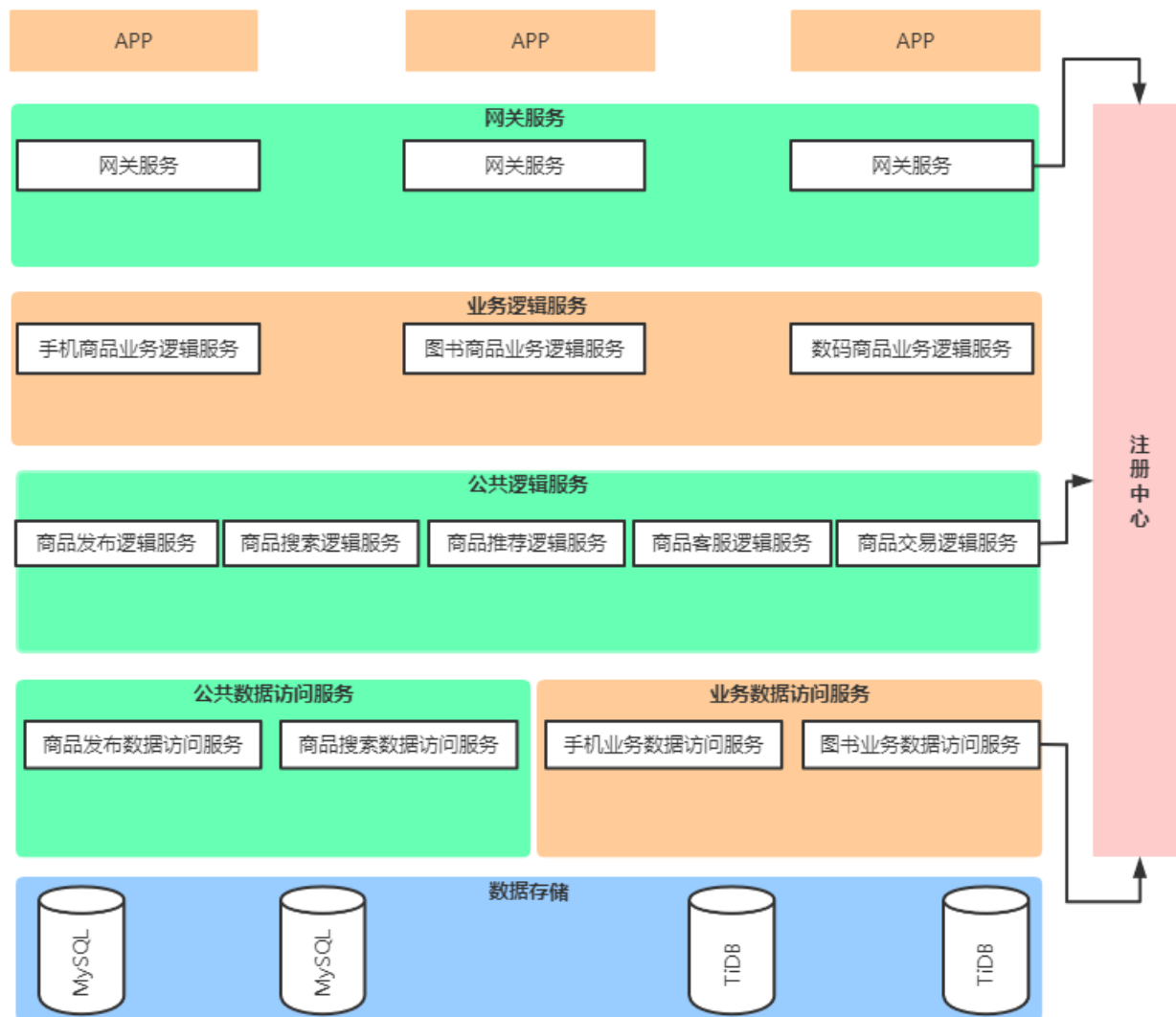
企业级微服务架构实践方法论

- 二手交易平台微服务实践



企业级微服务架构实践方法论

- 二手交易平台微服务进一步实践

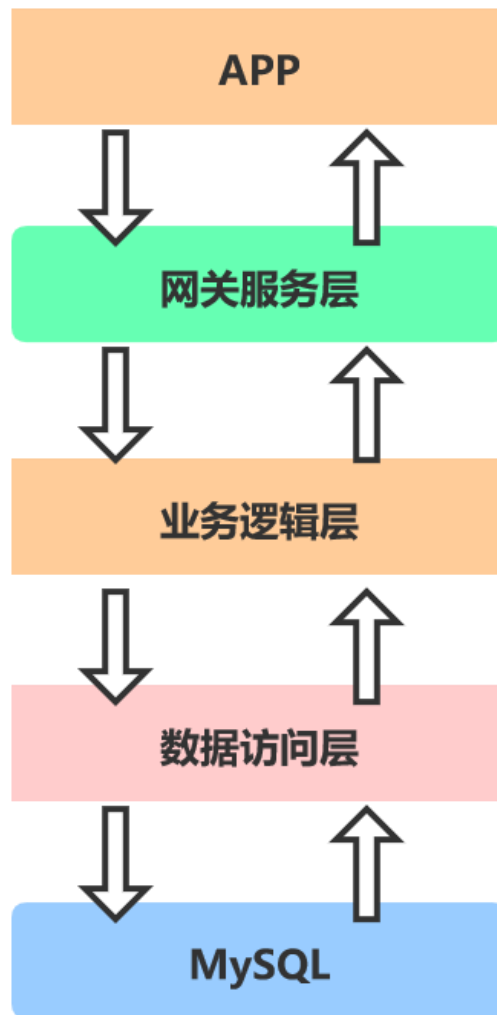




03.企业级微服务架构同步/异步业务场景案例剖析（腾讯社交案例、阿里电商案例）

企业级微企业级微服务架构同步/异步业务场景案例剖析

- 同步架构



企业级微服务架构同步/异步业务场景案例剖析

- 异步架构

- 异步目的

- ✓ 提升吞吐量

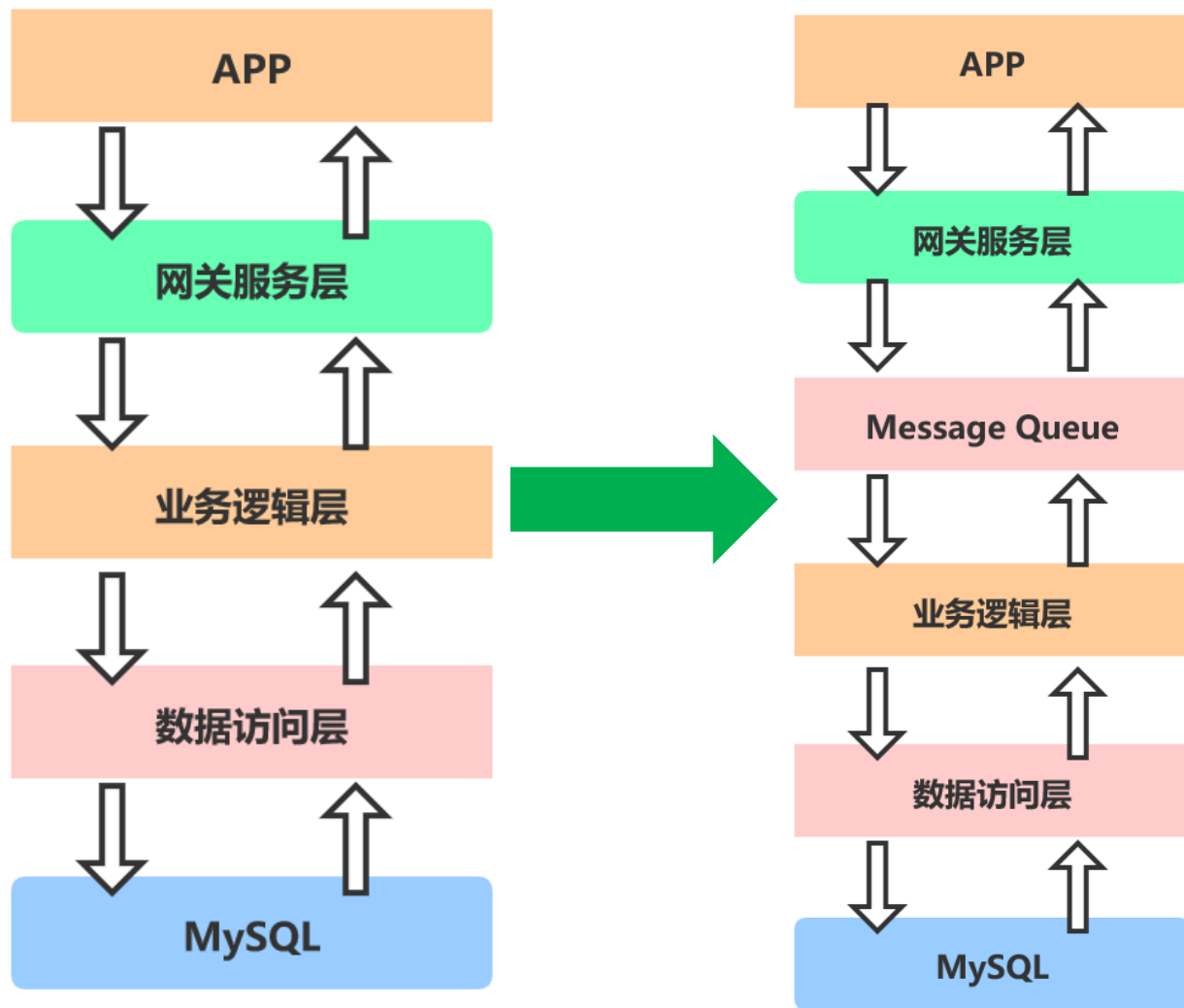
- 异步手段

- ✓ 消息队列

- 适用场景

- ✓ 请求类型

- ✓ 业务类型



● 企业级代码



```
Project
├── message
│   ├── annotation
│   │   ├── ClassUtils 2019-11
│   │   └── MessageHelper 20
│   └── render
│       ├── Rcmd4SearchRender
│       └── RcmdResultRender
├── validation
└── components
    ├── CateListService 2019
    └── ...

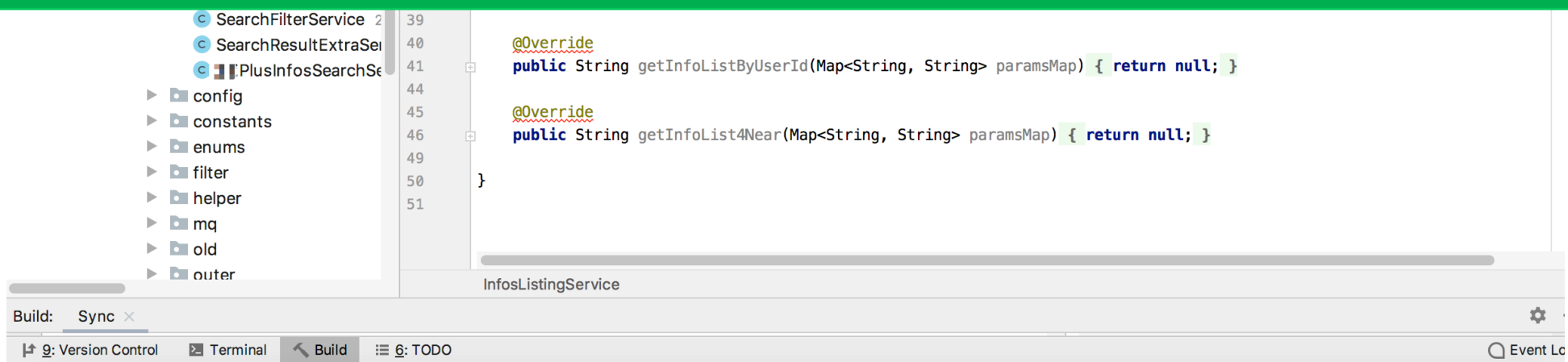
Rcmd4SearchRender.java
19
20
21
22
23
24
25
26
27
28
29

InfosListingService.java
19
20
21
22
23
24
25
26
27
28
29

@Override
public String getListingInfoList(Map<String, String> paramsMap) {
    IListingService bean = ApplicationContextContainer.getInstance().getBean(IListingService.class);
    ListingVo listingInfoList = bean.getListingInfoList(paramsMap);
    return ScfResponse.getResponseJsonString(listingInfoList.getCode(), listingInfoList, listingInfoList.getErrMsg(),
        listingInfoList.getTotalCount());
}

@Override
public String getListingInfoList4PC(Map<String, String> paramsMap) {
    IListingService bean = ApplicationContextContainer.getInstance().getBean(IListingService.class);
}
```

正式VIP课程全部落地企业级代码!!!



```
SearchFilterService 2
39
40
41
42
43
44
45
46
47
48
49
50
51

SearchResultExtraSer
PlusInfosSearchSe

config
constants
enums
filter
helper
mq
old
outer

InfosListingService
39
40
41
42
43
44
45
46
47
48
49
50
51

@Override
public String getInfoListByUserId(Map<String, String> paramsMap) { return null; }

@Override
public String getInfoList4Near(Map<String, String> paramsMap) { return null; }

}

Build: Sync x
9: Version Control Terminal Build 6: TODO Event Lc
```


场景Balance架构设计思维模型

01

企业级微服务架构领域驱动设计 (DDD) 拆分方法论

02

企业级微服务架构实践方法论

03

企业级微服务架构同步/异步业务场景案例剖析 (腾讯社交案例、阿里电商案例)

NiX 奈学教育



欢迎关注本人公众号
“架构之美”