

C/C++Linux服务器开发

高级架构师课程

三年课程沉淀

五次精益求精

十年行业积累

百个实战项目

十万内容受众

讲师:darren/326873713



扫一扫 升职加薪

班主任:柚子/2690491738

讲师介绍--专业来自专注和实力



Darren老师

曾供职于国内知名半导体公司（珠海扬智/深圳联发科），曾在某互联网公司担任音视频通话项目经理。主要从事音视频驱动、多媒体中间件、流媒体服务器的开发，开发过即时通讯+音视频通话的大型项目，在音视频、C/C++/GO Linux服务器领域有丰富的实战经验。



1 IM即时通讯项目框架分析和部署

- 0 即时通讯项目课程安排
- 1 即时通讯应用场景
- 2 即时通讯的基本架构
- 3 即时通讯分层架构
- 4 Teamtalk架构 (源码)
- 5 即时通讯项目重点
- 6 即时通讯项目意义和收获
- 7 即时通讯项目部署
- 8 通信协议解析

0 即时通讯项目课程安排

1. 即时通讯框架分析和部署
2. 登录服务器和消息服务器设计
3. 路由服务器设计
4. 数据库代理服务器设计
5. 文件服务器和docker部署
6. 产品上云公网发布/公网测试上线



1.1 即时通讯应用场景

即时通讯技术应用非常广泛，涉及教育、电商、金融、泛娱乐、生活服务、医疗健康、政企服务、游戏聊天、在线客服等等行业，以下以部分典型应用为例。

（一）教育



（二）电商



（三）金融



（五）招聘



（六）医疗健康



（七）政企服务



提供：即时通讯、用户点对点聊天、主播互动、实时弹幕、实时监控、远程控制、多人互动、位置共享



1.2 即时通讯提供商

容联

融云

环信 [http api](http://api.easemob.com) [环信即时通讯 REST API 概览 \[IM开发文档\] \(easemob.com\)](http://api.easemob.com)

网易云信 阿里

QQ通讯 qq sdk

详细对比: https://blog.csdn.net/weixin_33739646/article/details/87996637

每天发信息不过10000条

大部分场景下自研成本比使用第三方的高

自研的优势: 可控



1.3 即时通讯开源项目

C++: teamtalk

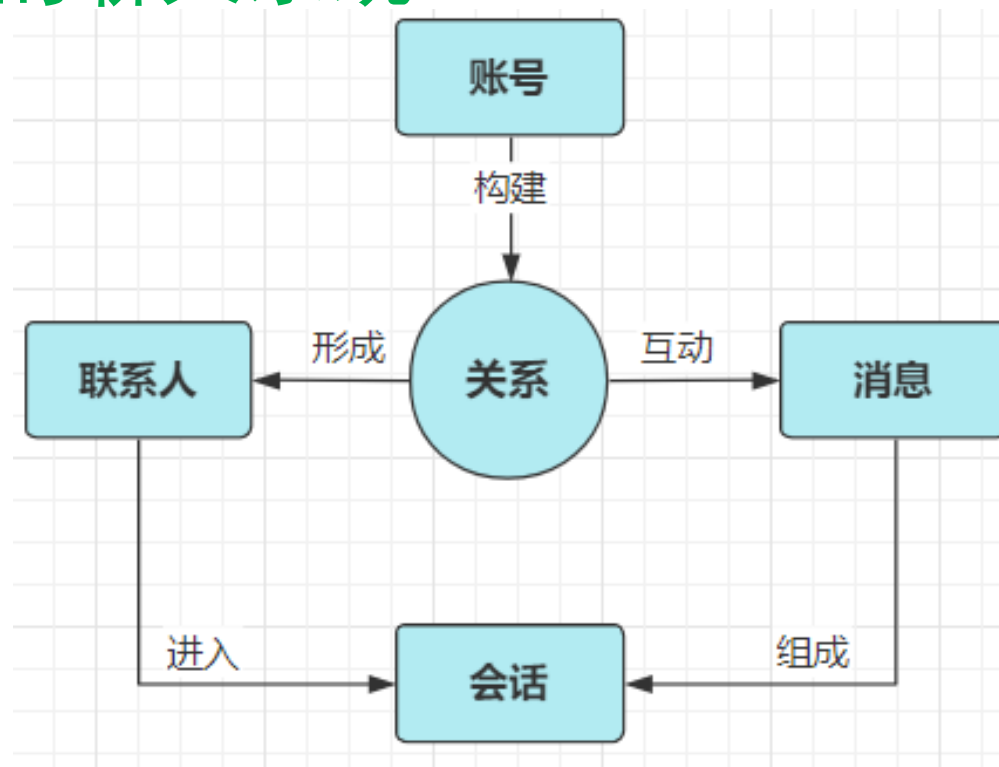
Go: [GoBelieve](#)

goim

Java: openfire



2.0 即时通讯-用户眼中的聊天系统



用户账号：用户唯一标识

联系人列表：比如好友关系

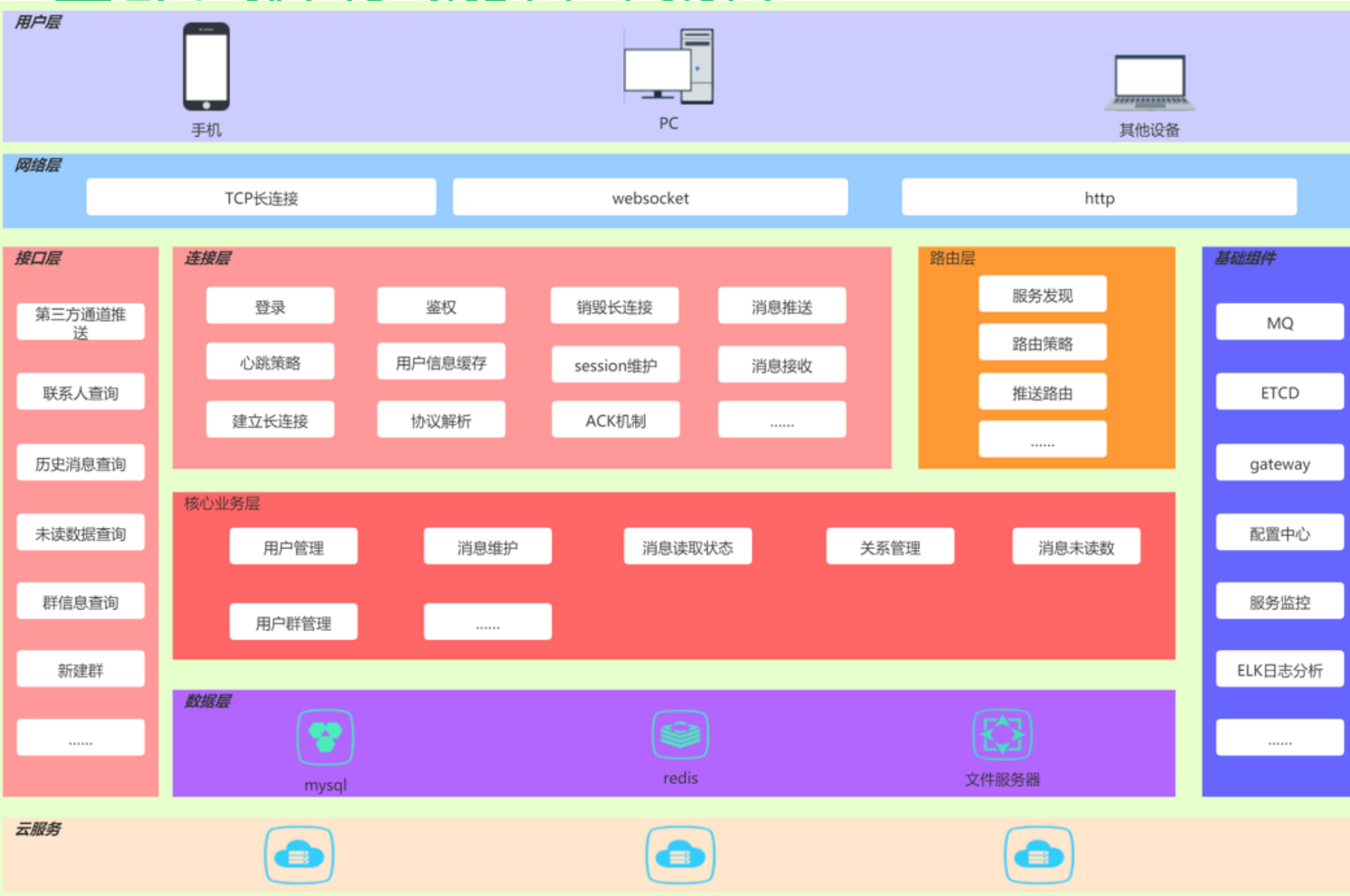
消息：聊天互动产生消息

聊天会话：你和对方的聊天记录组成一个聊天会话，在会话可以看到你们之间的互动消息。



2.1 即时通讯的基本架构

客户端与服务器端进行网络通信、收发消息



连接层为客户端收发消息提供出入口。主要的任务：保持海量用户连接；解析协议，对传输内容进行编码；维护session；推送消息。

核心业务层(逻辑层)负责IM系统各项功能的核心逻辑实现。

路由层负责消息投递

数据层负责IM系统相关数据的持久化存储，包括消息内容、账号信息等。

外部接口服务最常用的第三方系统推送服务有苹果手机自带的 APNs (Apple Push Notification service) 服务、安卓手机内置的谷歌公司的 GCM (Google Cloud Messaging) 服务等。



ce.

2.2 IM技术的特点

1. **实时性**：保证消息实时触达。（轮询与长连接）（消息提示推送，和我们阅读的消息是不一样的。）
2. **可靠性**：保证消息的不丢失和不重复。（ACK 机制）
TCP 只能保证消息数据链路可靠，不能保证业务可靠。
3. **一致性**：保证同一条消息在多人、多终端展现顺序的一致性。（消息序号生成器）
4. **安全性**：保证数据传输安全、数据存储安全、消息内容安全。（WWS、HTTPS、TLS、AES 等）

参考：

[微信新一代通信安全解决方案：基于TLS1.3的MMTLS详解-](#)



3 即时通讯分层架构

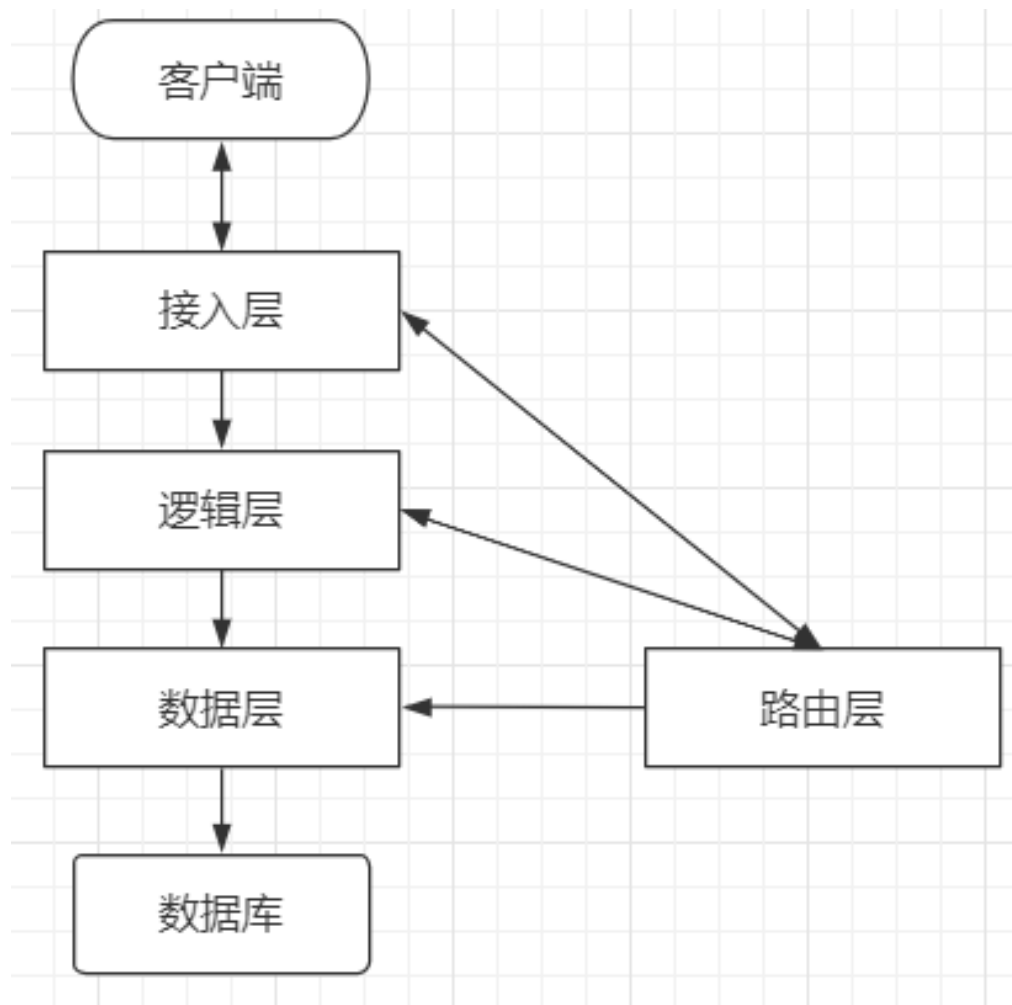
即时通讯系统常用分层架构

接入层：entry/gate

逻辑层：logic

数据层：data

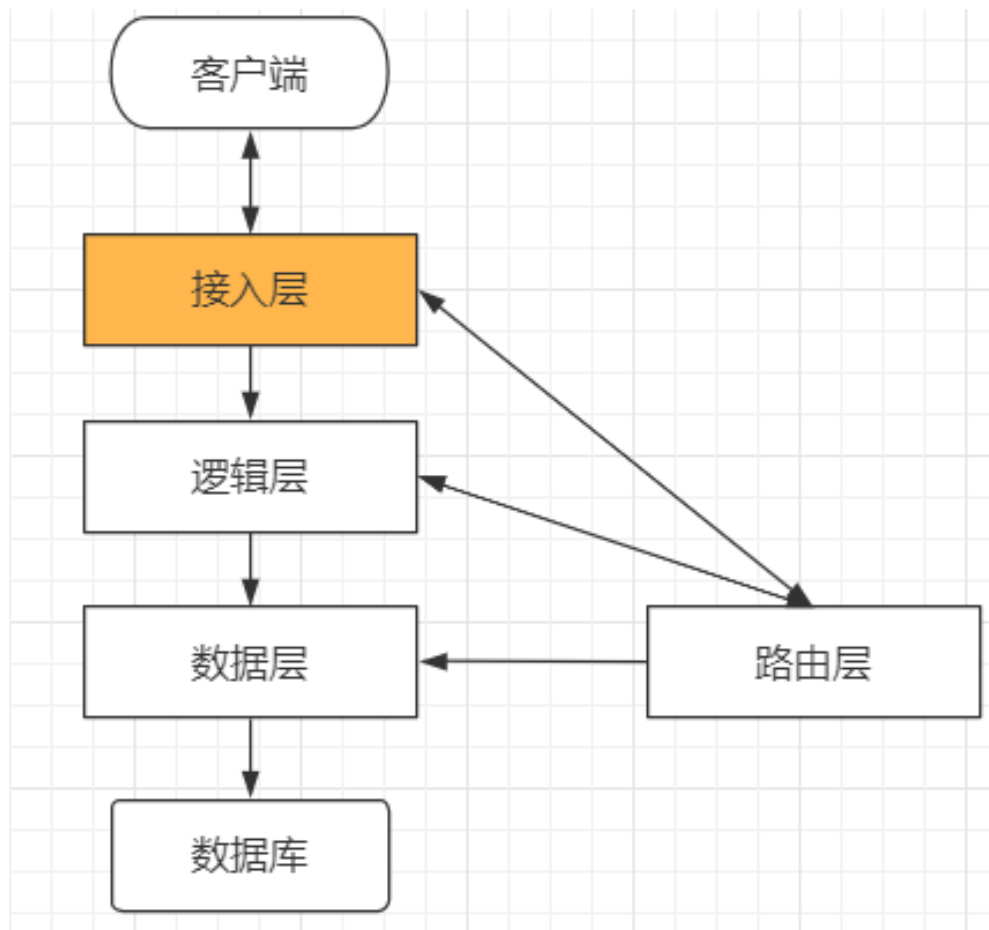
路由层：route



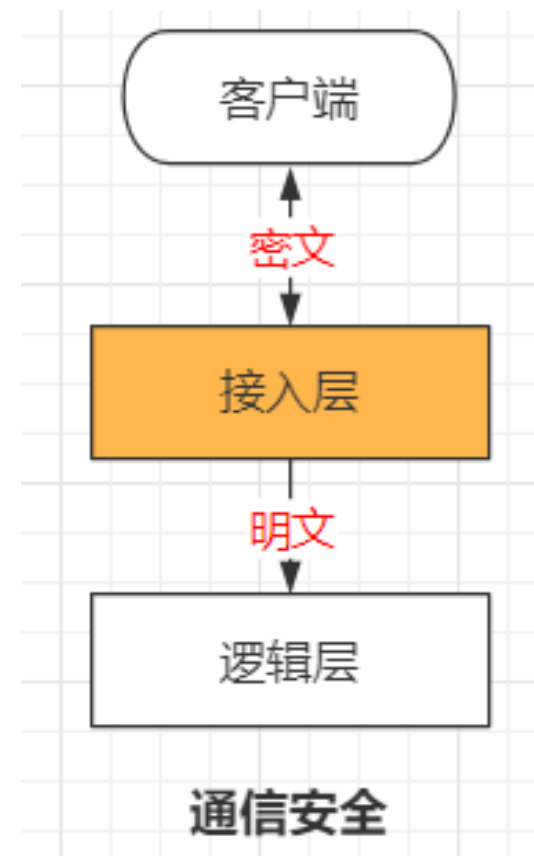
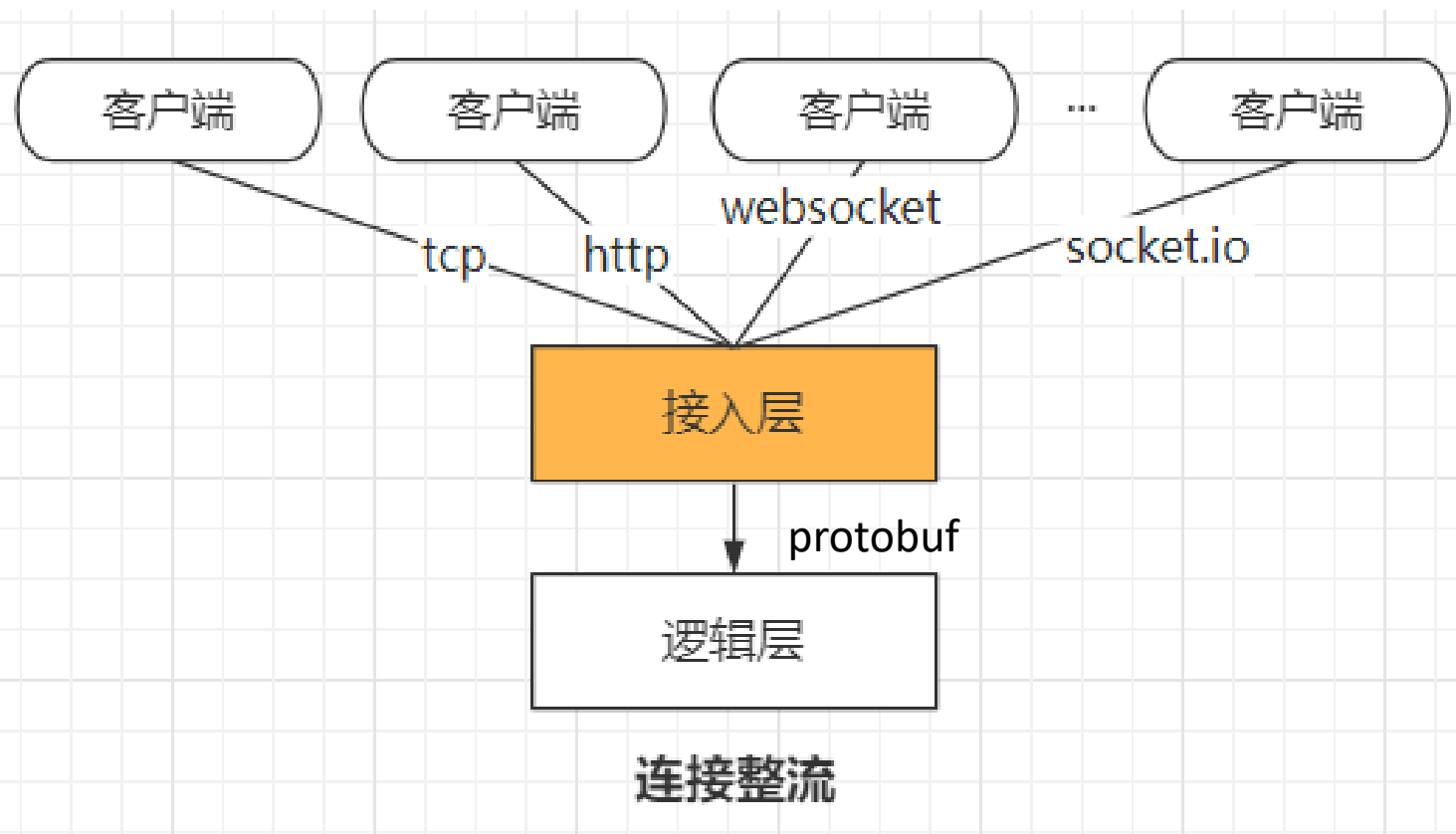
3.1 即时通讯分层架构-接入层

接入层的作用：

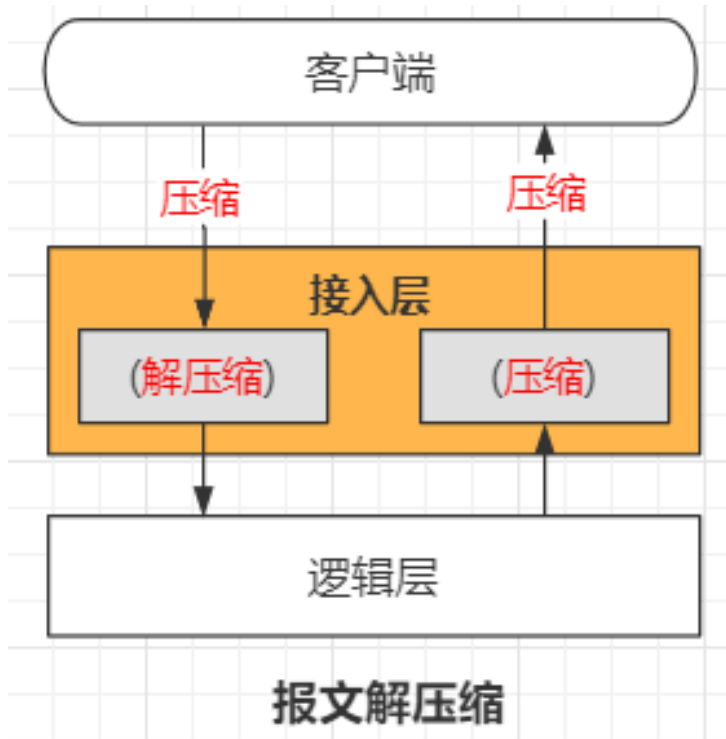
- 连接整流
- 通信安全
- 报文解压缩
- 初步防攻击



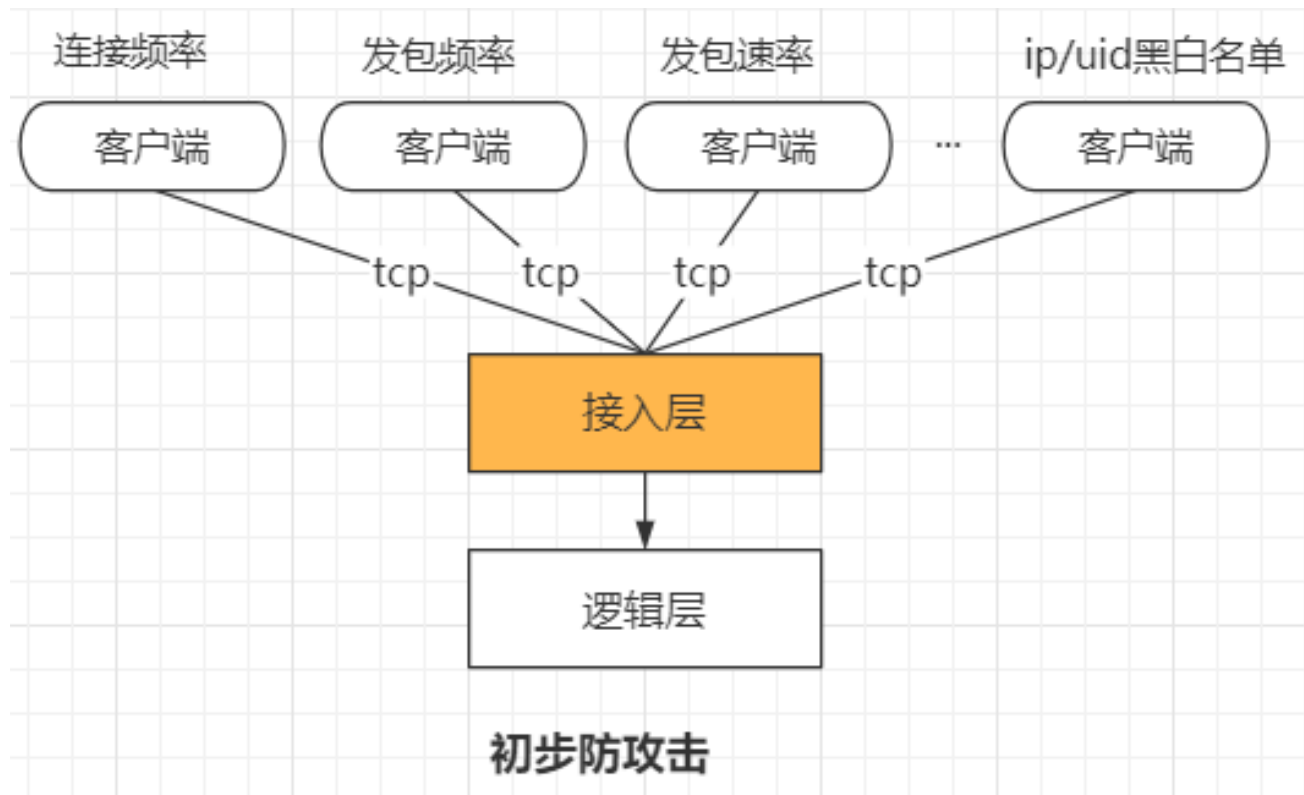
3.1 接入层-连接整流和通信安全



3.1 接入层-报文解压缩和初步防攻击

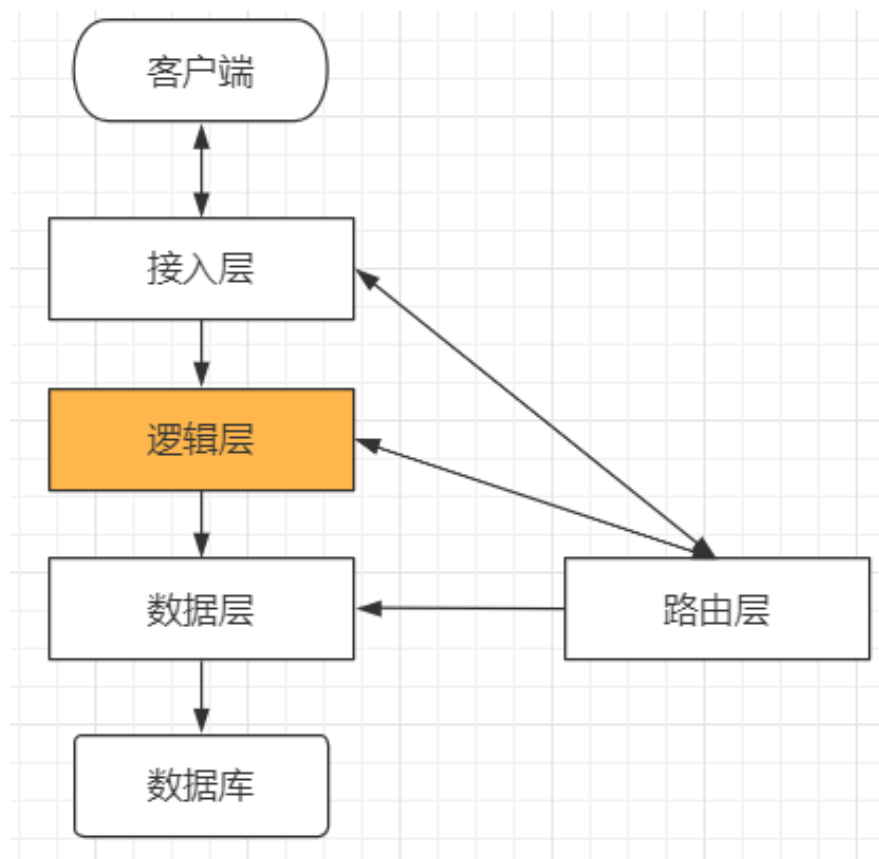


zip

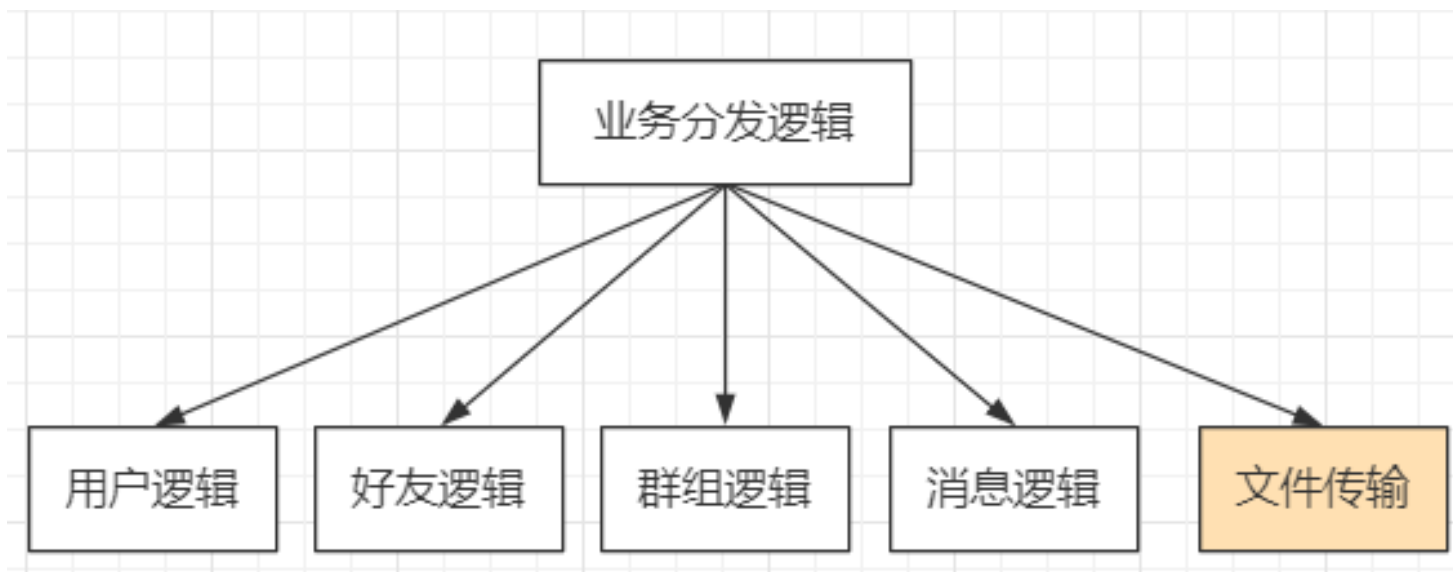


3.2 即时通讯分层架构-逻辑层

1. **用户逻辑**：用户登录、用户退出、用户信息查询、用户更新签名、用户分组创建等
2. **好友逻辑**：添加好友、删除好友、拉取好友列表、好友添加备注等
3. **群组逻辑**：创建群、加入群、删除群、删除成员等
4. **消息逻辑**：单聊文字消息、单聊语音消息、群聊文字消息、群聊语音消息、拉取离线消息等
5. **其他**，比如文件传输（需要接收点击接收）、图片传输（直接显示）等



3.2 逻辑层设计核心-扩展性



同一个服务也可以做

划分越细，需要开发人员越多。

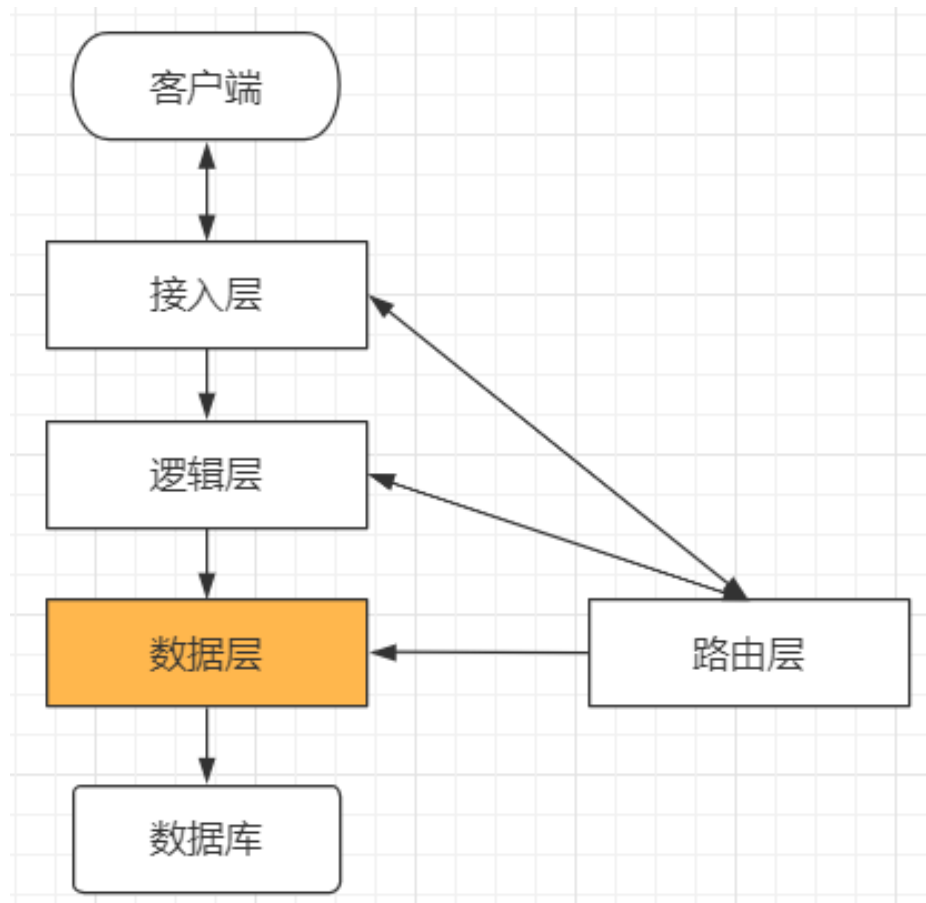
文字单聊消息逻辑 文字群聊消息逻辑

1. 水平扩展各个逻辑模块，**消息逻辑1** 消息逻辑2 ...消息逻辑n
2. 无缝添加新的逻辑服务，**比如文件传输**

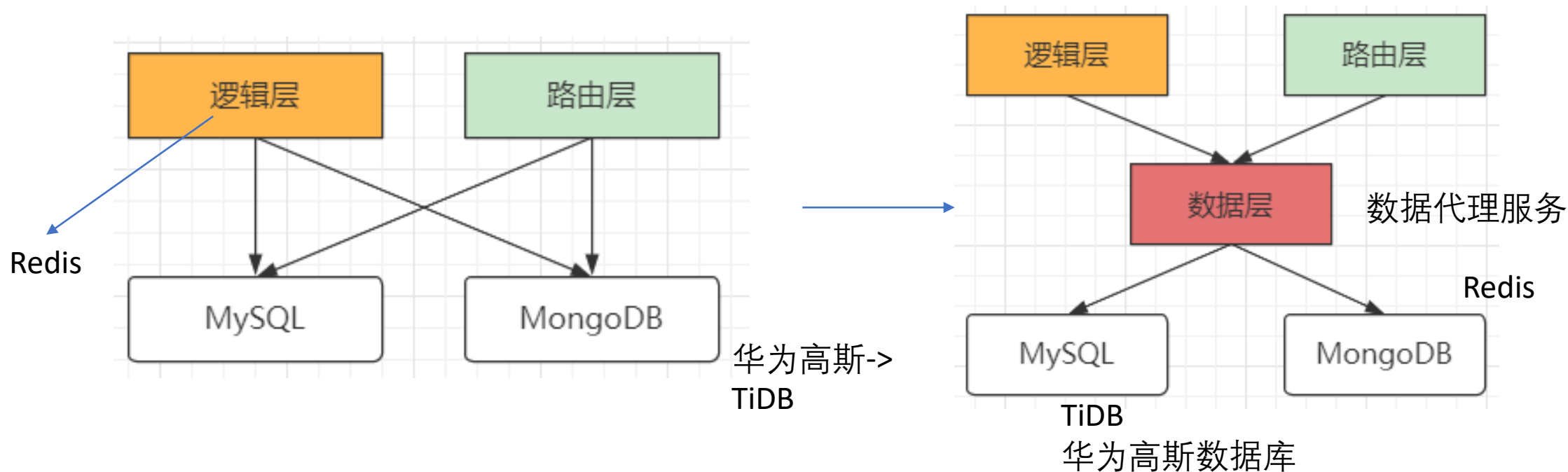


3.3 即时通讯分层架构-数据层（数据代理层）

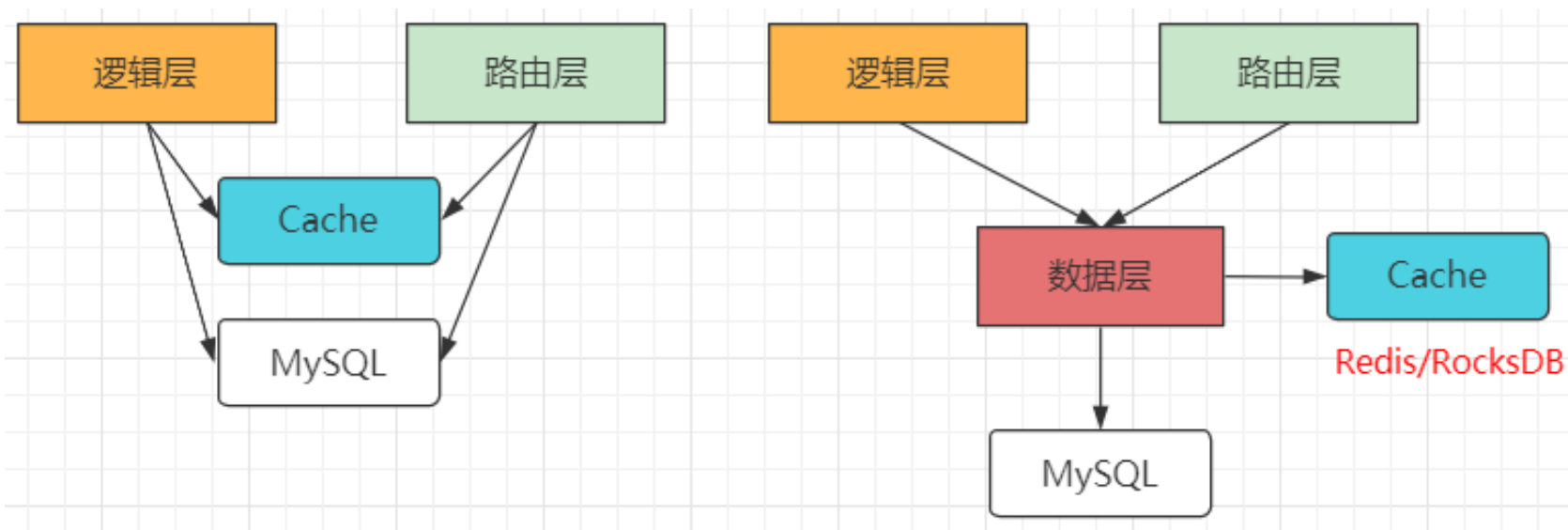
1. 对上游屏蔽存储引擎
2. 对上游屏蔽cache层
3. 对上游提供友好接口



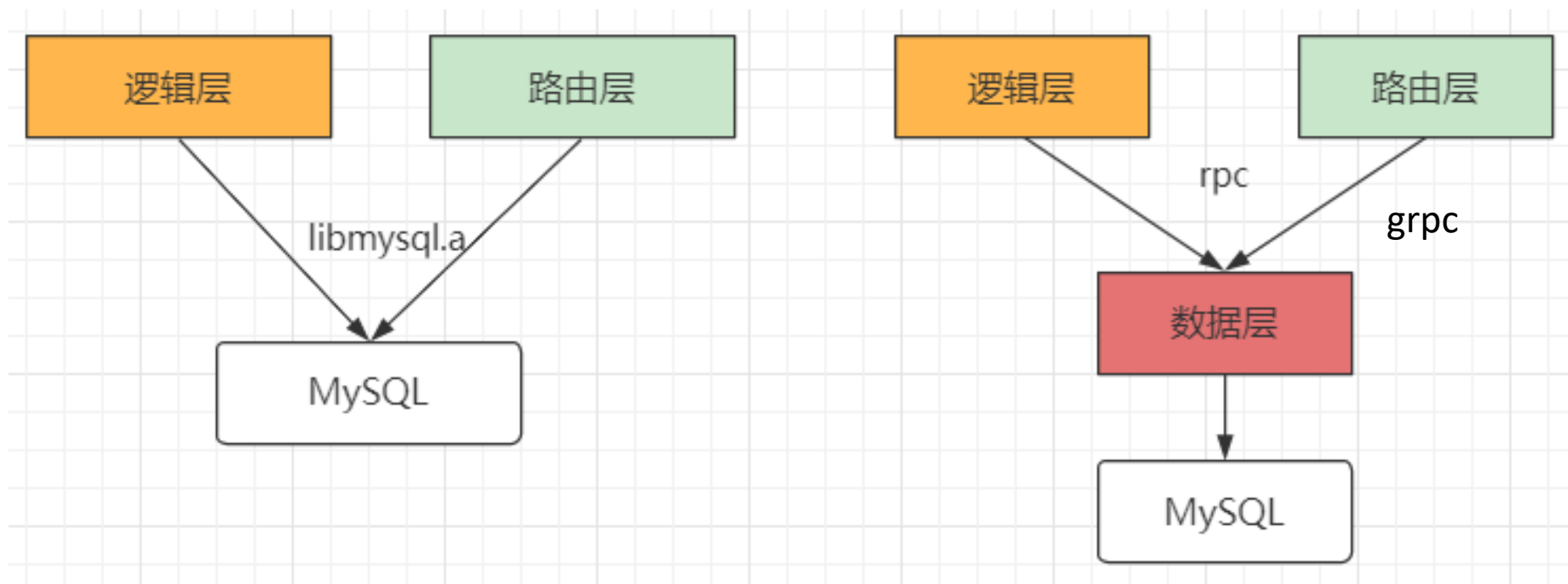
3.3.1 数据层-屏蔽存储引擎



3.3.2 数据层-屏蔽cache层



3.3.3 数据层-提供友好接口

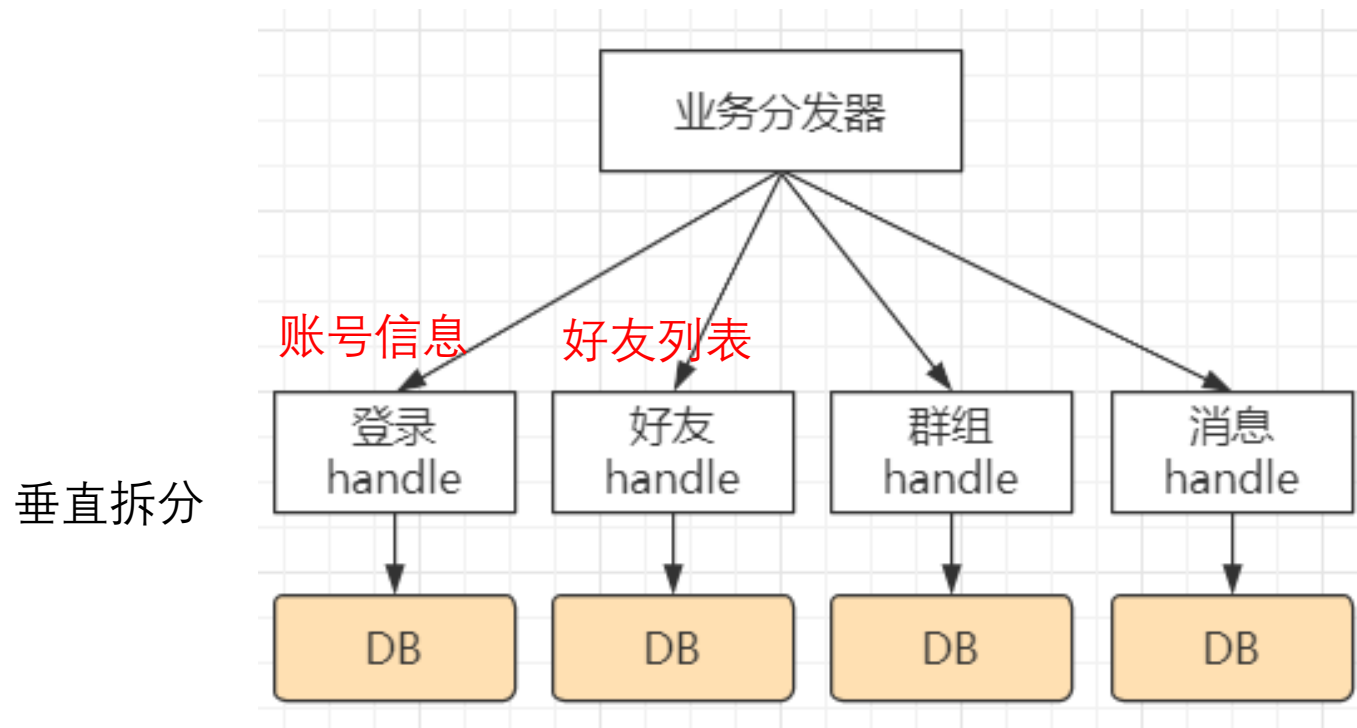


3.3.4 数据层-扩展性

数据存储
业务量小，单个数据库。

垂直拆分

水平拆分



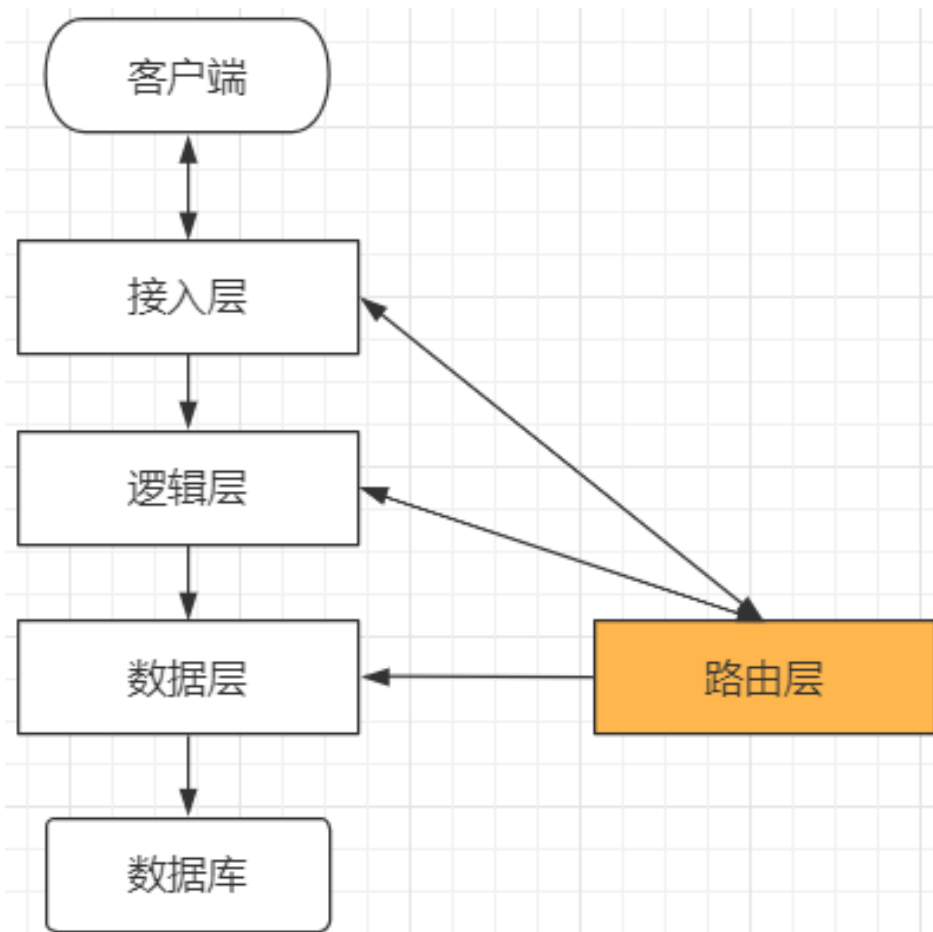
水平拆分,可以通过映射的方式

消息DB1 消息DB2 消息DB3



3.4 即时通讯分层架构-路由层

1. 路由消息：消息投递
2. 内存存储：用户临时数据，比如用户状态信息(在线/离线)



3.4.1 路由层-路由消息

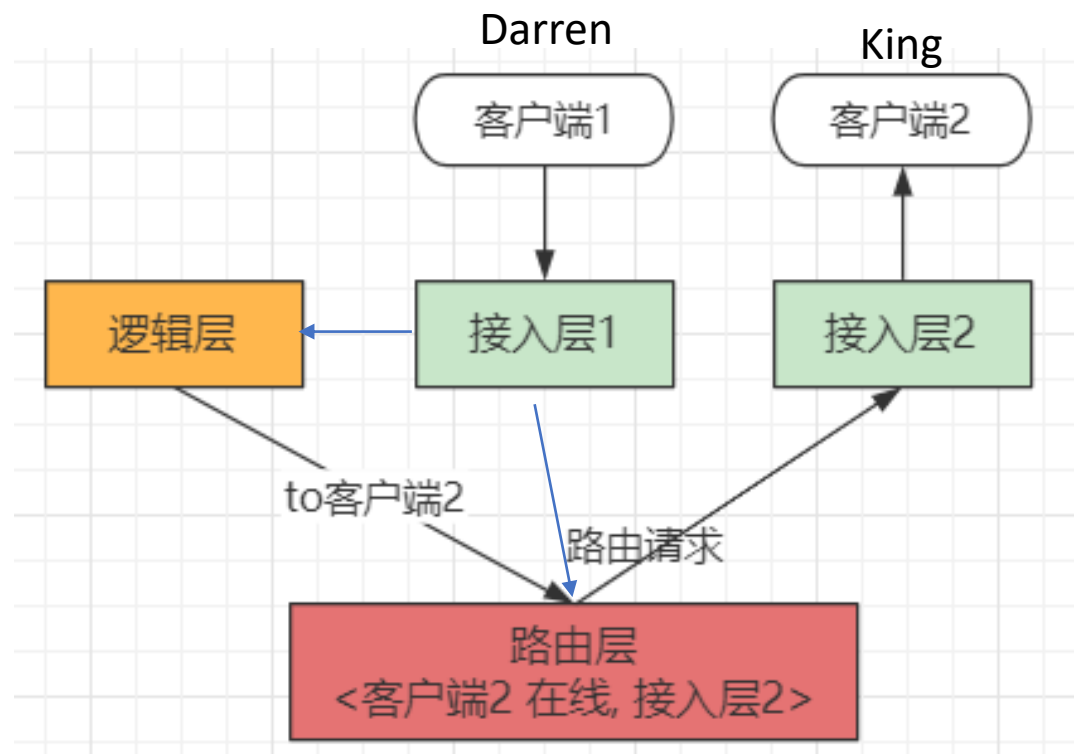
1. 为什么路由层要存储
用户状态信息(在线、隐身)

2. 用户id - 接入层 对应关系

Map<客户端2, 接入层2>

消息投递

踢人功能

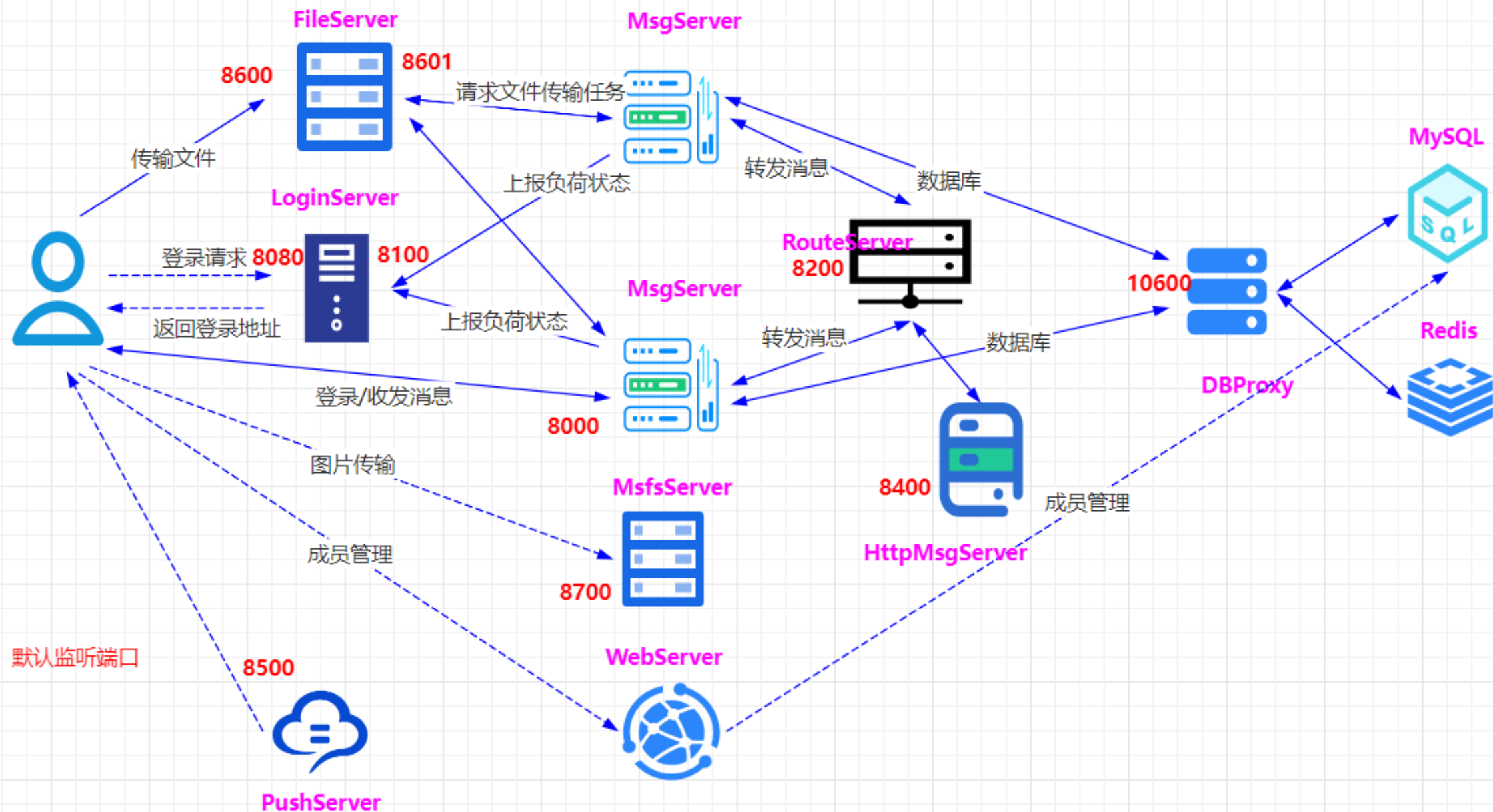


4 Teamtalk架构

为什么学习Teamtalk:

■ CloudTalk基于Teamtalk进行二次开发

■ xx公司基于Teamtalk进行二次开发



5 TeamTalk (简称TT) 即时通讯项目

1. Android/iOS/PC:各种客户端。
2. LoginServer: 主要负责负载均衡的作用，当收到客户端的请求时，分配一个负载最小的MsgServer给客户端。
3. MsgServer: TT的主要服务端，负责维护各个客户端的连接，消息转发等功能。
4. RouteServer: 负责消息路由的功能，当msg_server发现某个用户不在本服务器内，而又有消息需要发给他，就会将消息转发给route_server，route_server会将消息发给相应的msg_server，由此可知，route_server也维护了一定的用户状态。
5. DBProxy: 在TT中负责了主要的业务逻辑，主要与存储层打交道，提供mysql以及redis的访问服务，屏蔽其他服务器与mysql与redis的直接交互。
6. FileServer: 文件服务器，提供客户端之间得文件传输服务，支持在线以及离线文件传输
7. MsfsServer: 图片存储服务器，提供头像，图片传输中的图片存储服务。
8. PushServer: 负责Android、IOS客户端提醒消息的推送，类似微信的锁屏提醒消息。
9. Webserver:简单的管理功能。



6.1 我们能从中学到什么

1. 基础组件封装: thread、ringfifo等
2. Epoll网络库封装(客户端 服务端)
3. HTTP模块封装
4. HTTP api实战
5. HTTPs
6. Websocket
7. protobuf实战
8. Json使用
9. MySQL连接池
10. Redis连接池
11. 线程池
12. 数据库实战
13. 小文件存储系统原理
14. 音视频集成到即时通讯
15. 即时通讯技术的原理和实战
 1. 消息单聊 (消息id)
 2. 消息群聊 (分表的设计)
 3. 消息推送和拉取



6.2 即时通讯重点

实时性

- http
- socket
- websocket

数据库

- 数据库表设计
- 分表设计
- 密码存储方式
- 最近会话表
- 未读消息如何体现
- 聊天消息分表问题

单聊消息发送流程

- 怎么保证数据的不丢失
- 消息序号在哪里生成
- 消息序号生成方式
- 消息如何封装
- 消息发送时的seq有什么作用
- 消息发送后服务器怎么应答?
- 接收端收到数据后如何应答

群成员管理

- 如何创建群
- 如何删除群
- 怎么使用redis管理群成员

客户端的单聊消息未读计数是怎么实现的

- 服务器怎么保留消息未读计数
- 客户端的未读消息计数从何而来
- 客户端未读消息计数清0时向服务器发送了什么，服务器又是怎么清除未读消息计数

文件传输原理

- 文件传输分为在线传输离线传输
- 在线传输和离线传输有什么区别

群聊消息

- 如何推送群聊
- 群消息计数器
- 群会话如何更新（每有一个人发送消息，则其他人都需要更新会话消息）

MySQL连接池设计

- 为什么使用连接池
- 连接池设置多大合适?

redis连接池设计

- 为什么使用连接池
- 连接池设置多大合适?

并发能力

- 如何做到百万并发
- 如何做到千万并发



7.1 即时通讯项目部署

文档<https://www.yuque.com/docs/share/e848ccf0-ae02-4a46-9753-e0cb559ff568?#> 《1-2 即时通讯项目编译和配置》

- 1.保证redis正常运行；
- 2.保证mysql正常运行；
- 3.保证php正常运行；
- 4.保证nginx正常运行；
- 5.保证php正常运行，这里的php主要是web管理平台的使用；
- 6.protobuf版本库目前server端是2.0的版本，Android_av使用了3.0的版本
- 7.web服务器需要nginx支持；
- 8.服务不启动的时候主要查看log是否正常。



7.2 配置文件重点1

loginserver.conf 负载均衡服务

HttpPort: 对应提供HTTP服务的端口, 默认8080

MsgServerPort: msg服务器上报监听的端口, 默认8100

msgserver.conf 消息服务

ListenPort: 连接msg服务器需要的端口

LoginServerIP1: 上报msg服务器负载的地址

LoginServerPort1: 上报msg服务器负载的端口

RouteServerIP1: route服务器地址和端口

RouteServerPort1:

PushServerIP1: push推送服务器地址和端口

PushServerPort1:

FileServerIP1: 文件传输服务器地址和端口

IpAddr1: 对外提供的往外ip, 需要上报给login 服务器

routeserver.conf 路由服务

ListenIP=0.0.0.0 该服务监听ip

ListenMsgPort=8200 该服务监听端口

dbproxyservice.conf 数据库中间件服务

ListenPort: 该服务监听的端口

ThreadNum: 线程池线程数量

DBInstances=teamtalk_master,teamtalk_slave配置数据库实例

#teamtalk_master

teamtalk_master_host=127.0.0.1

数据库ip

teamtalk_master_port=3306

数据库端口

teamtalk_master_dbname=teamtalk

数据库名字

teamtalk_master_username=root

用户名

teamtalk_master_password=123456

密码

teamtalk_master_maxconnct=16

连接池最大连接数量

#未读消息计数器的redis

unread_host=127.0.0.1

redis地址

unread_port=6379

redis端口

unread_db=1

db索引

unread_maxconnct=16

连接池最大连接数量



7.2 配置文件重点2

fileserver.conf 文件传输服务

ClientListenIP=0.0.0.0 该服务监听的地址
ClientListenPort=8600 该服务监听的端口,
供msg server连接

httpmsgserver.conf HTTP reset api服务

ListenPort=8400 该服务监听
ConcurrentDBConnCnt=4 db_proxy服务连接通道数量
DBServerIP1=127.0.0.1 db_proxy服务地址
DBServerPort1=10600
DBServerIP2=127.0.0.1
DBServerPort2=10600

RouteServerIP1=localhost route路由服务地址
RouteServerPort1=8200

msfs.conf 文件存储服务

ListenIP=0.0.0.0 #可以监听多个IP,用;分割
ListenPort=8700 该服务端口
BaseDir=./tmp 存储地址
FileCnt=0 最大文件数量
FilesPerDir=30000 每个目录最大存储文件
GetThreadCount=32 下载线程
PostThreadCount=1 上传线程

pushserver.conf 推送服务

ListenIP=127.0.0.1
ListenPort=8500 该服务监听端口

CertPath=apns-dev-cert.pem
KeyPath=apns-dev-key.pem
KeyPassword=tt@mogujie

#SandBox
#1: sandbox 0: production
SandBox=0



8 通信协议解析

