

移动应用开发 - 服务端

服务端常用技术介绍 / 系统设计实例讲解

讲师介绍

廖文鑫

- 2018 至今，今日头条 App 服务端负责人，负责后端技术架构以及业务功能
- 2013 - 2018，微信团队后台高级工程师，负责基础业务以及基础服务开发



移动应用的服务端开发

会告诉你的事：

- 服务端开发的基本技能，涉及到的知识点 / 组件
- 如何实现一个简单的 IM 服务 / 信息流服务，应该考虑什么问题

不会告诉你的事：

- 计算机基础知识
- 具体如何搭建服务，如何使用开源的组件和服务
- 代码怎么写

内容

服务端技术介绍

- 解决什么问题
- 技术栈与常用的技术有哪些

业务系统设计

- 如何实现一个 IM 服务
- 如何实现一个信息流服务

服务端技术介绍





服务端解决什么问题？

数据

- 存储 - 用户资料 / 商品 / 文章 / 图片 / 视频
- 传输 - 消息投递 / 转账 / 上传 / 下载

计算

- 数据计算 - 地图导航 / 搜索 / 信息流推荐
- 提供算力 - 语音合成 / 语音实时转写 / 图像内容识别



服务端技术介绍

- 系统架构纬度
- 常用技术名词



服务端技术介绍

- 系统架构纬度
- 常用技术名词

服务端开发的技术栈





基础设施 / 支撑系统

基础设施

- 操作系统
- 网络设施
- 机房机架
- ...

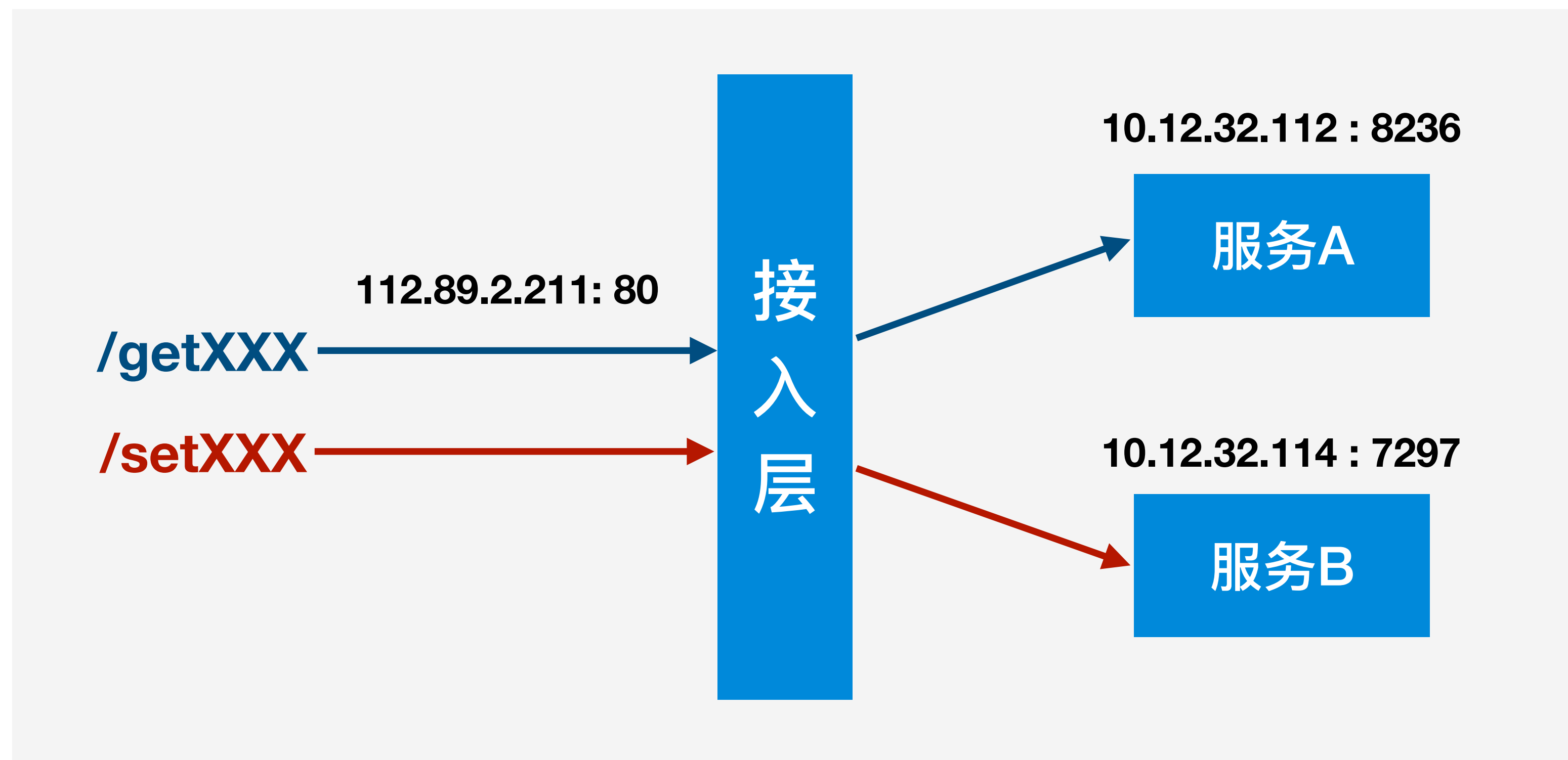
支撑系统

- 日志系统
- 监控报警
- 运维系统
- ...

接入层

作用

- 反向代理请求
 - nginx
- 负载均衡
 - 轮询 / 随机 / 权重
 - hash / 一致性 hash
 - ...



逻辑层

作用

- 根据业务逻辑，处理用户请求，得到结果

设计模式

- 单体式服务
- 微服务





单体式服务

所有的业务逻辑都集中在一个代码仓库，一个进程里

优点

- 简单，适合协作人数少，功能较少
- 维护 / 更新 方便

缺点

- 难以支持高复杂度系统



微服务

业务逻辑按功能或类型作划分，分别部署多个服务，使用网络通信交换数据

优点

- 多人团队协作成本更低
- 功能复用，系统解耦，故障隔离

缺点

- 影响新能，系统复杂性大增，对业务流程设计提出了更高的要求
- 对网络，基础框架要求更高



存储层

作用

- 数据的 存储 / 转发 / 查找

常用的存储

- SQL 类关系型存储
- NoSQL 类 Key-Value 存储
- 分布式文件系统
- 队列存储



服务端技术介绍

- 系统架构纬度
- 常用技术名词



HTTP / HTTPS

HTTP：超文本传输协议，数据交换协议的一种

- 万维网数据通信的基础
- 基于 TCP，协议简单

HTTPS：TLS 加持的 HTTP

- 提供网站服务器身份认证
- 保护数据交换的隐私以及完整性校验



JSON / XML / Protobuf

数据的封装和交换

JSON

```
{
  "name": "alice",
  "relation": {
    "family": [
      "xiaohong",
      "ligang"
    ],
    "friends": [
      "zhang3",
      "li4"
    ]
  }
}
```

XML

```
<?xml version="1.0"?>
<person name="alice">
  <family>
    <name>xiaohong</name>
    <name>ligang</name>
  </family>
  <friends>
    <name>zhang3</name>
    <name>li4</name>
  </friends>
</persion>
```

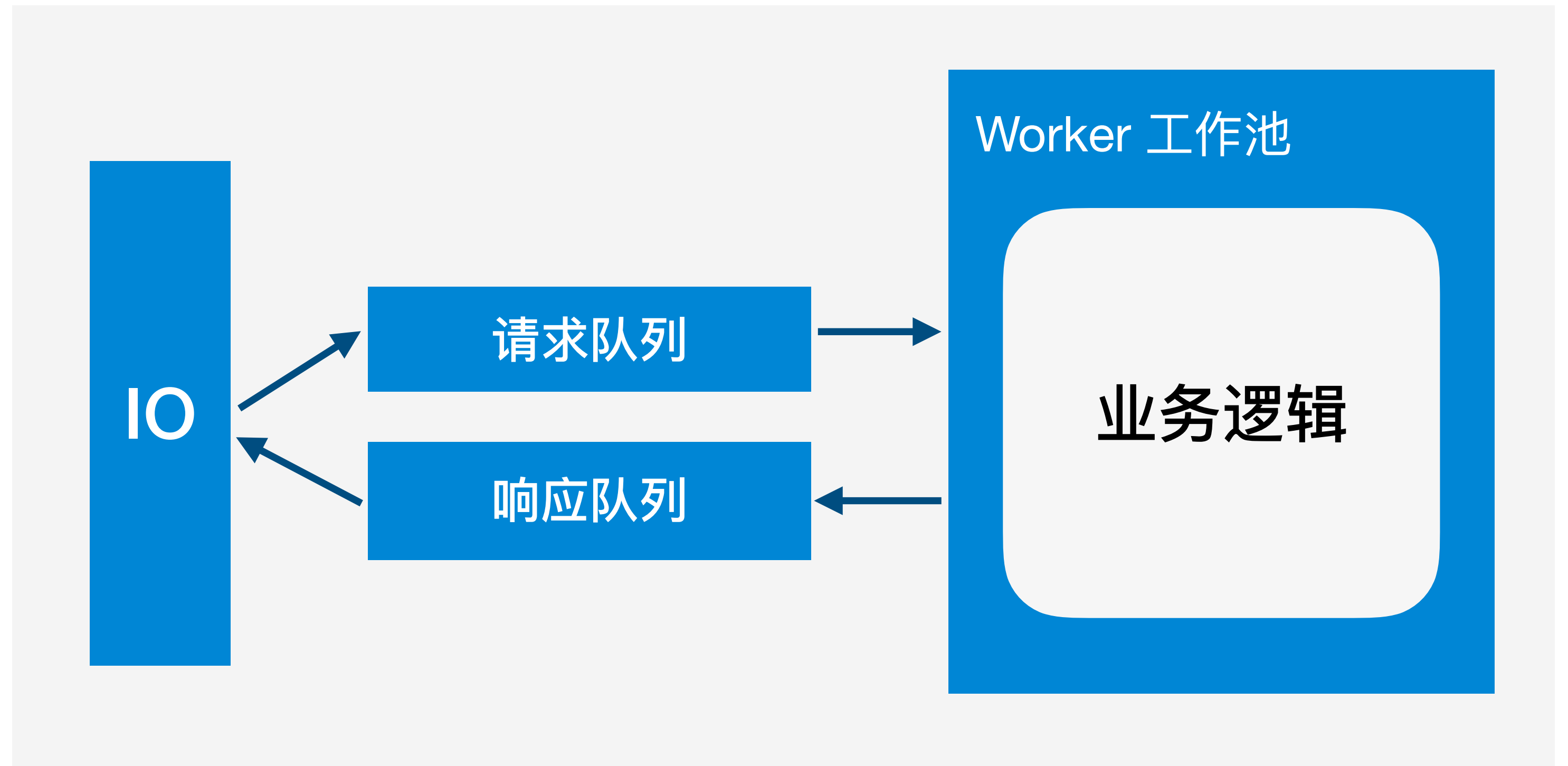
Protobuf


```
message Person {
  required string name = 1;
  repeated Person family = 2;
  repeated Person friends = 3;
}
```

RPC

远程过程调用

- 封装网络收发 / 传输协议
- 专注业务逻辑开发
- gRPC / bRPC / thrift





非关系型数据库

简单，性能高，横向拓展性高

Redis

- 性能高，支持多种数据结构，kv / list / set 等等，有一定的持久化能力

memcached

- 性能高，简单易用，作为cache使用

MongoDB

- 查询功能强大，性能好



关系型数据库

支持复杂的数据关系存储和查询

Mysql

- 性能卓越，服务稳定，开源



消息队列

解耦业务逻辑，削峰限流

Kafka

- 性能很高，消息理论上不会丢

RabbitMQ

- 支持AMQP

ZeroMQ

- 超轻量的事件组件

业务系统设计






业务系统设计

- 设计一个 IM 服务
- 设计一个信息流服务



业务系统设计

- 设计一个 IM 服务
- 设计一个信息流服务

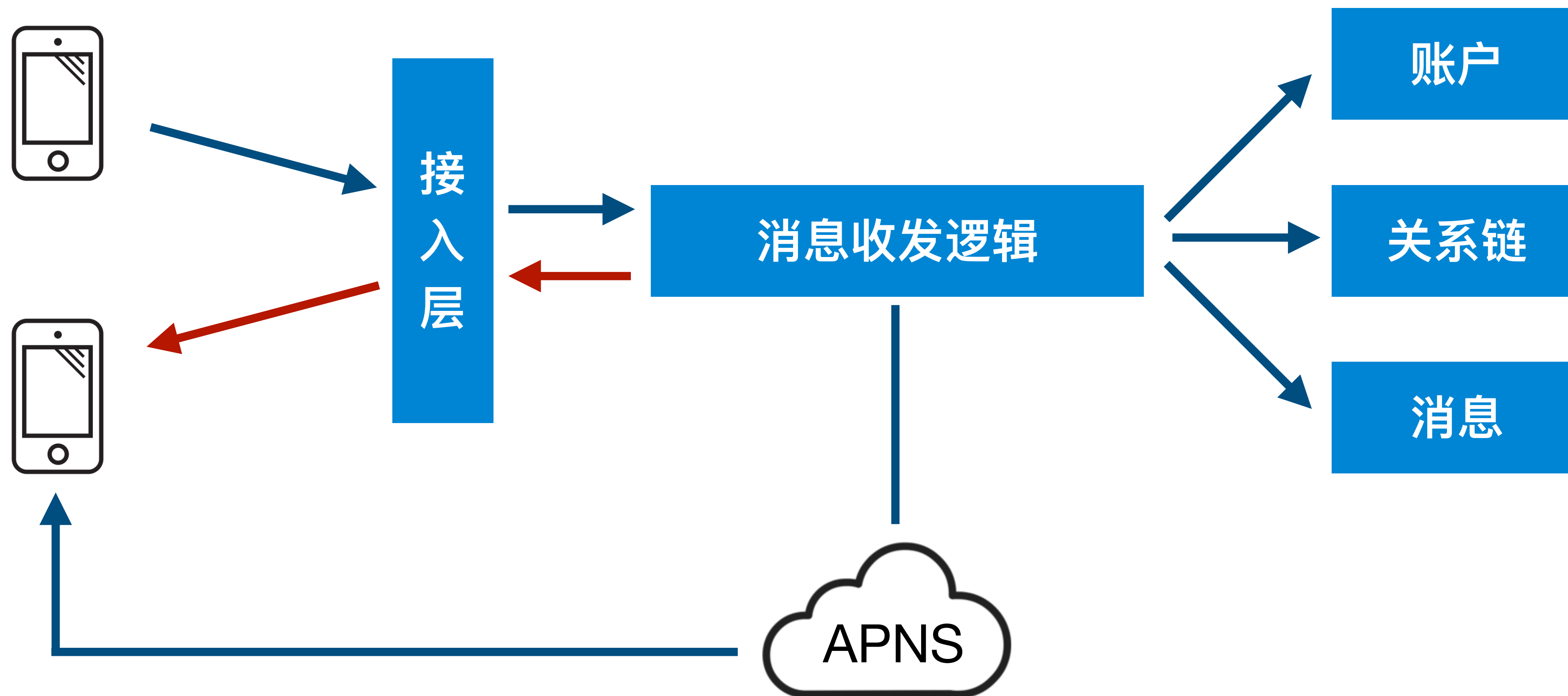


设计一个 IM 服务

需要考虑的问题

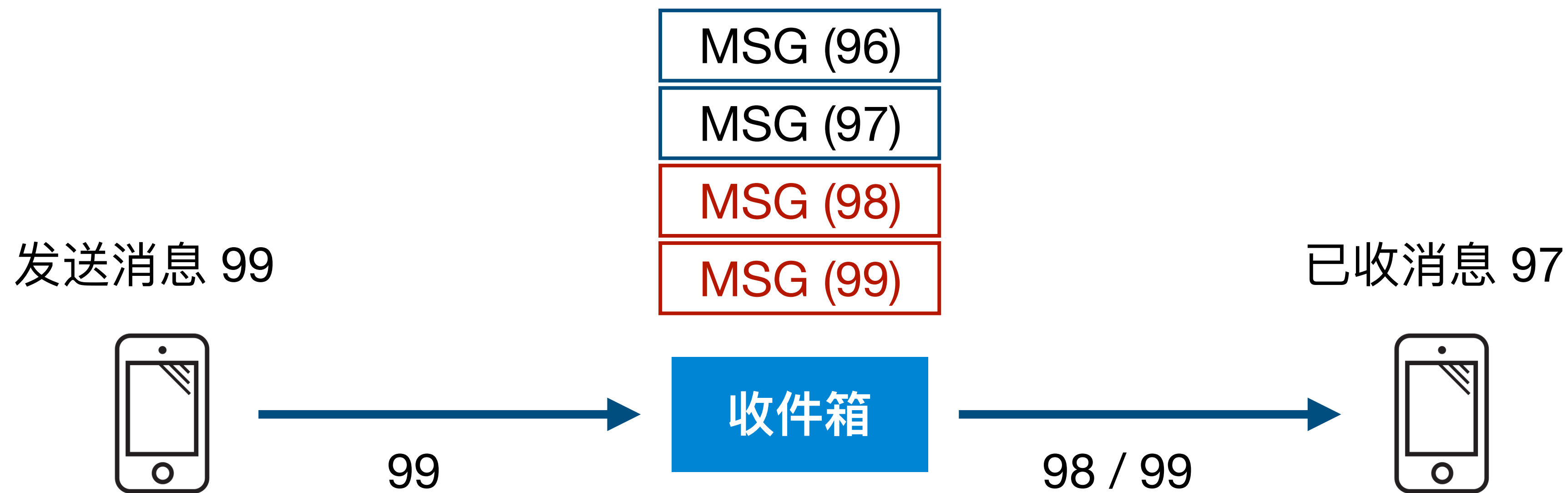
- 账户 / 消息 / 关系链 存储选型
- 消息模型：读扩散 / 写扩散
- 消息收发逻辑
- 新消息推送

IM 服务整体架构



消息模型

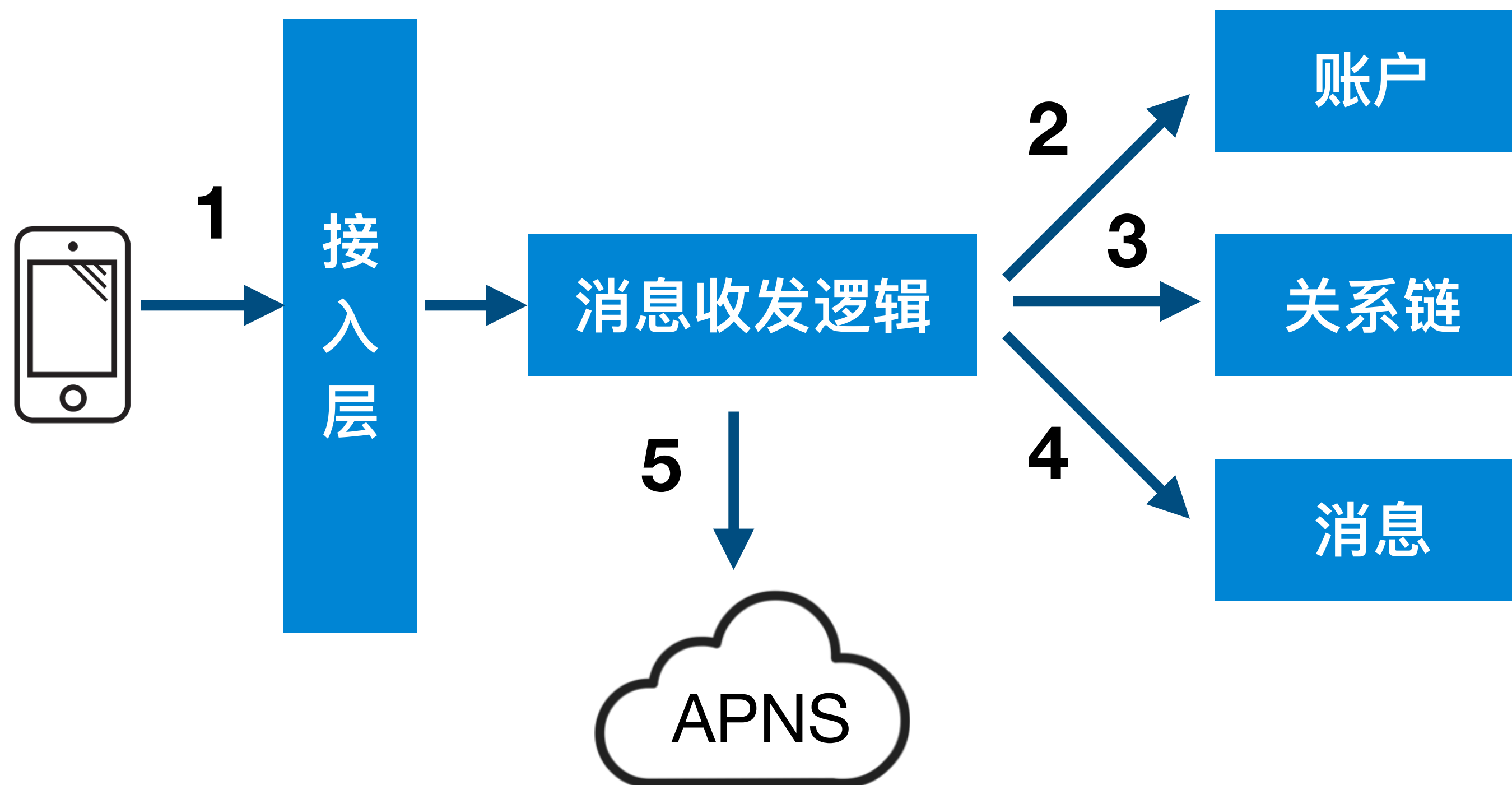
存储转发



发消息逻辑

大体流程

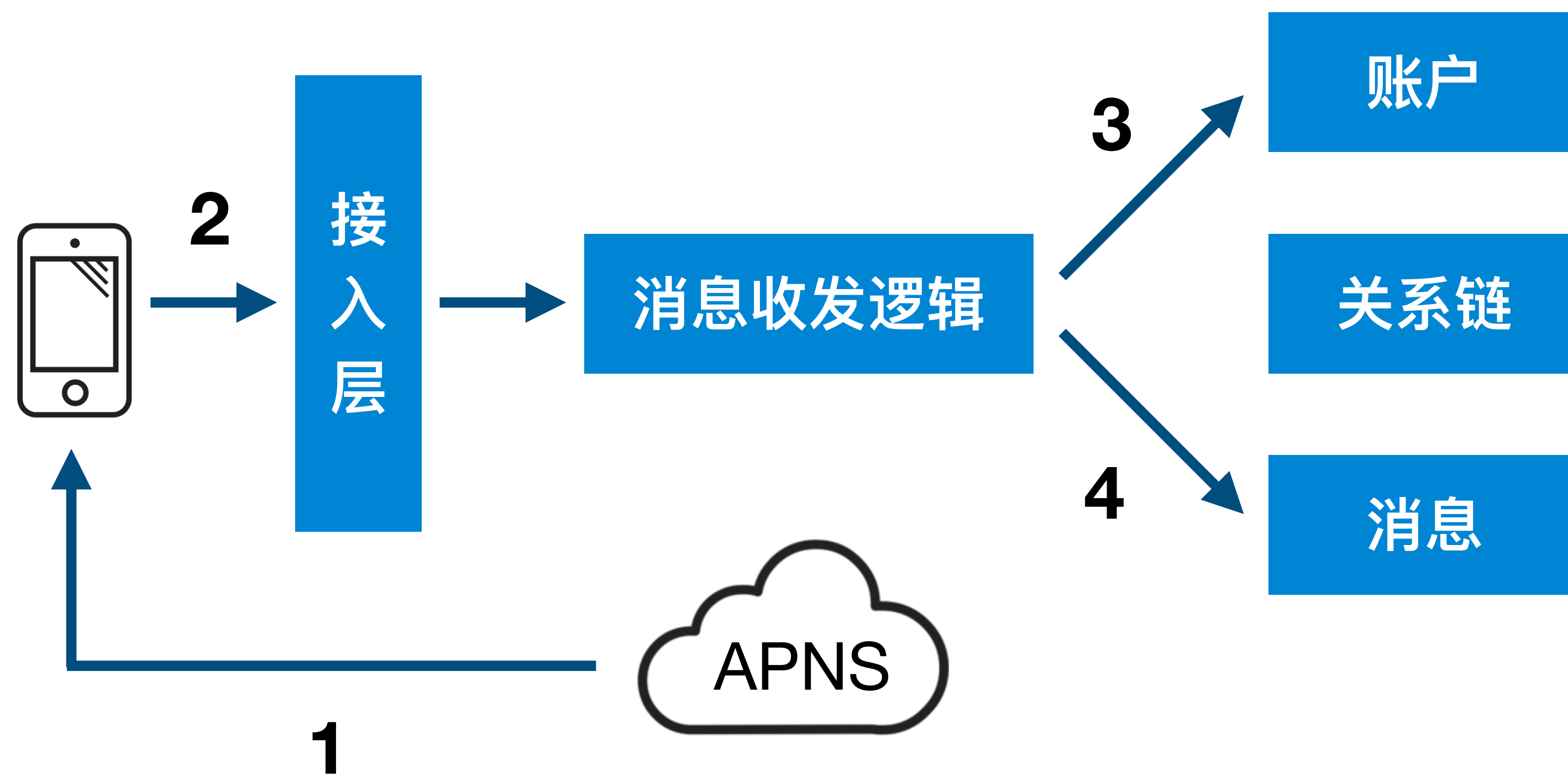
1. 发送消息请求
2. 检查账户
3. 检查关系链
4. 将消息写入接收者的收件箱
5. 发送最新消息通知




收消息逻辑

大体流程

1. 接收新消息通知
2. 发起收消息请求
3. 检查账户
4. 根据消息编号，读取未收下去的消息





业务系统设计

- 设计一个 IM 服务
- 设计一个信息流服务

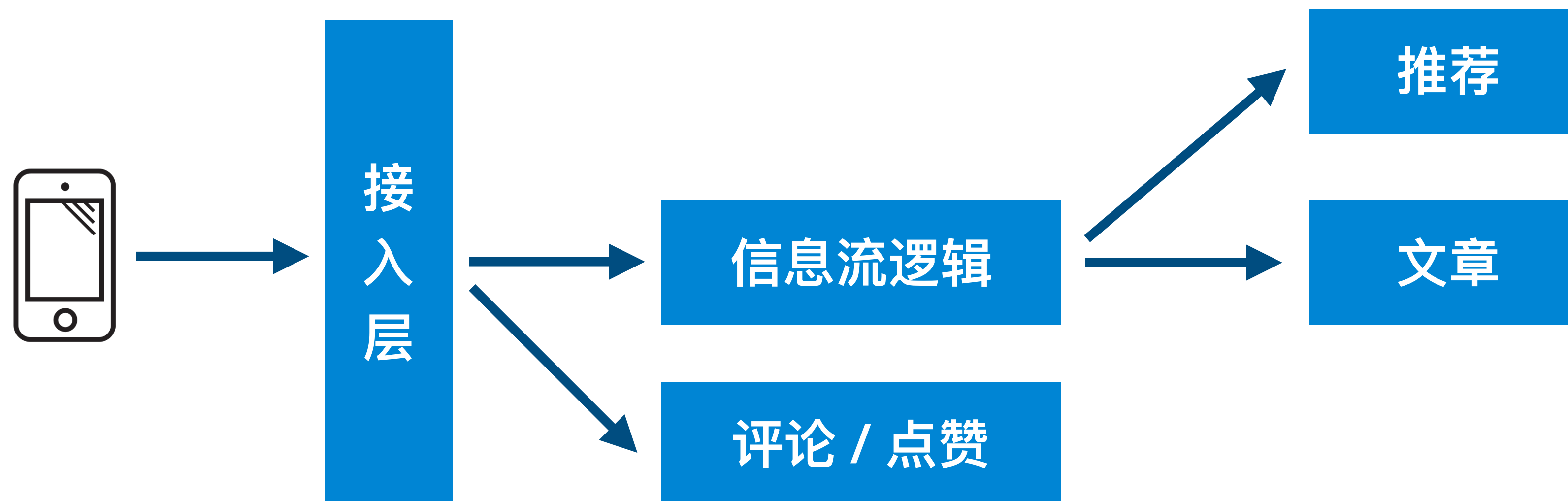


设计一个信息流服务

需要考虑的问题

- 文章推荐
- 文章数据存储与读取
- 评论 / 点赞

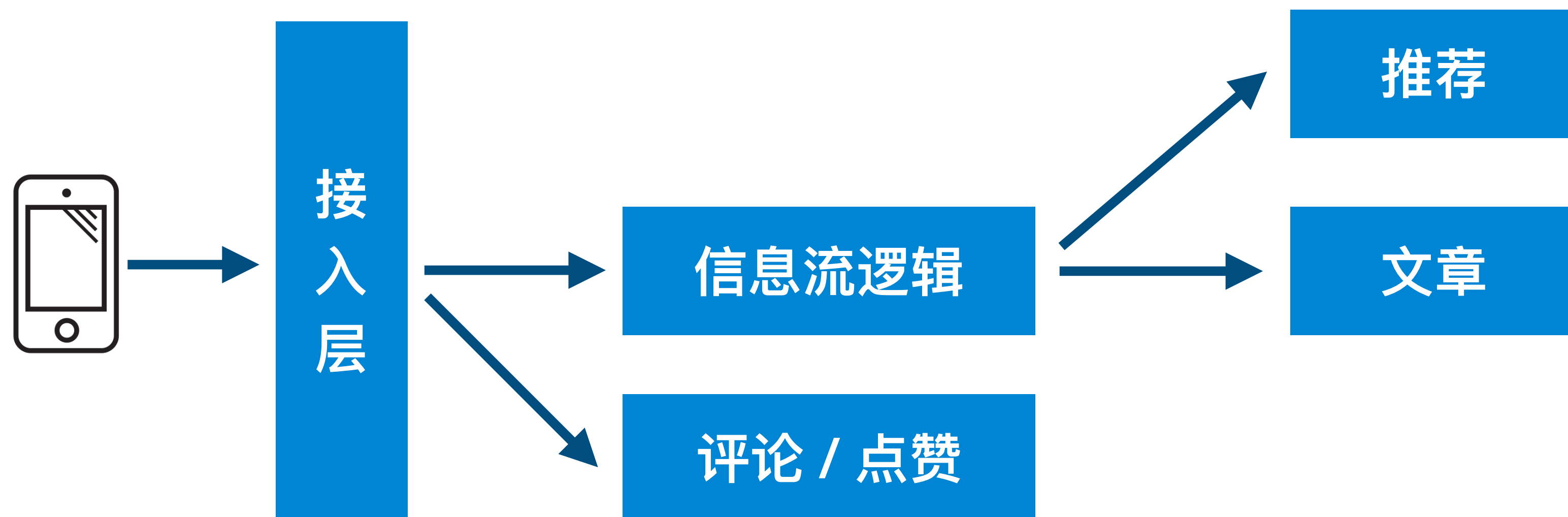
信息流服务整体架构



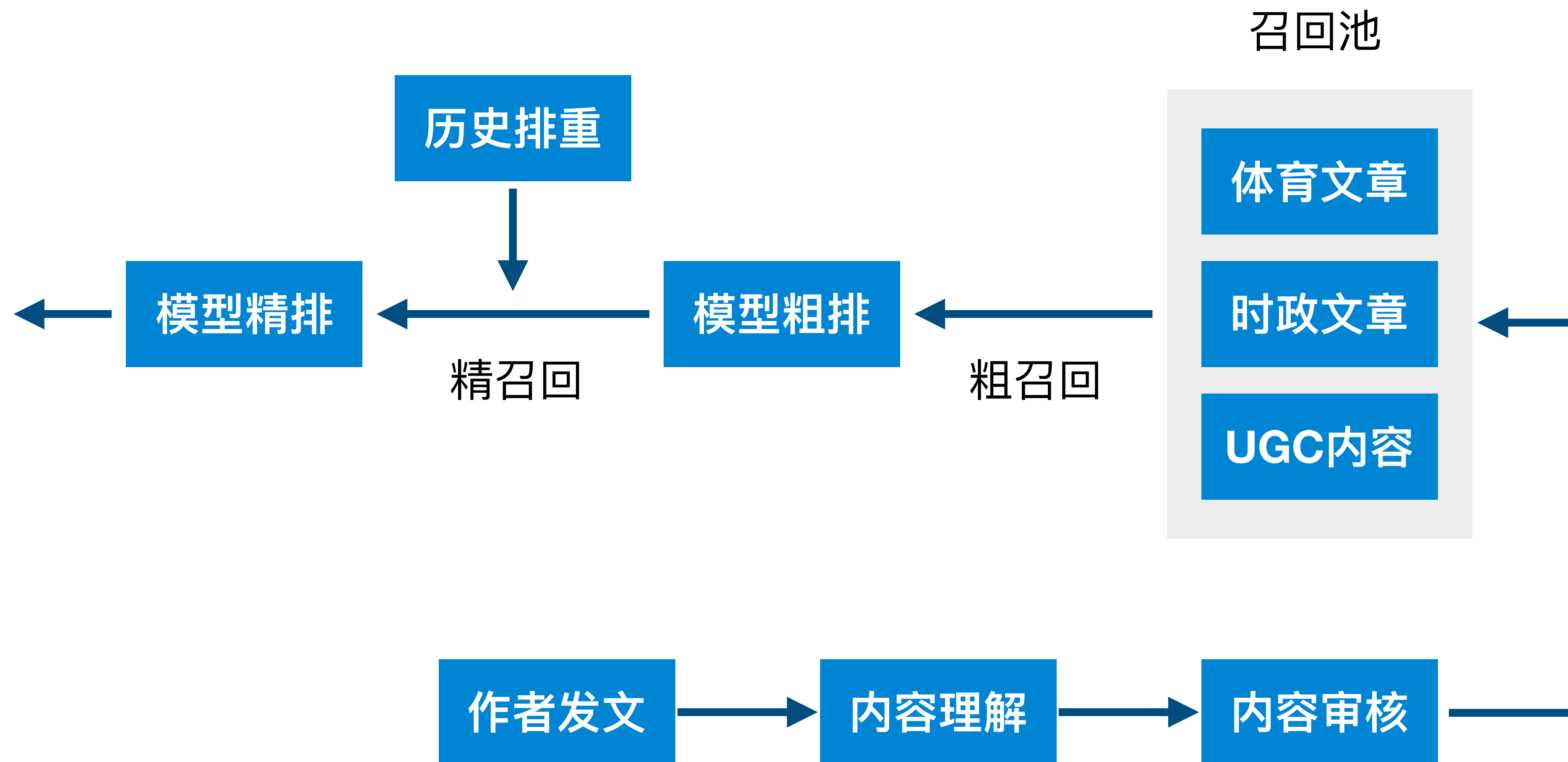
信息流逻辑

大体流程

1. 从推荐获取本次要出的文章列表
2. 获取文章的信息
3. 获取文章的评论/点赞信息
4. 打包后返回给客户端



文章推荐流程





Q&A

THANKS.

