



火龙果•整理
uml.org.cn

接入层篇

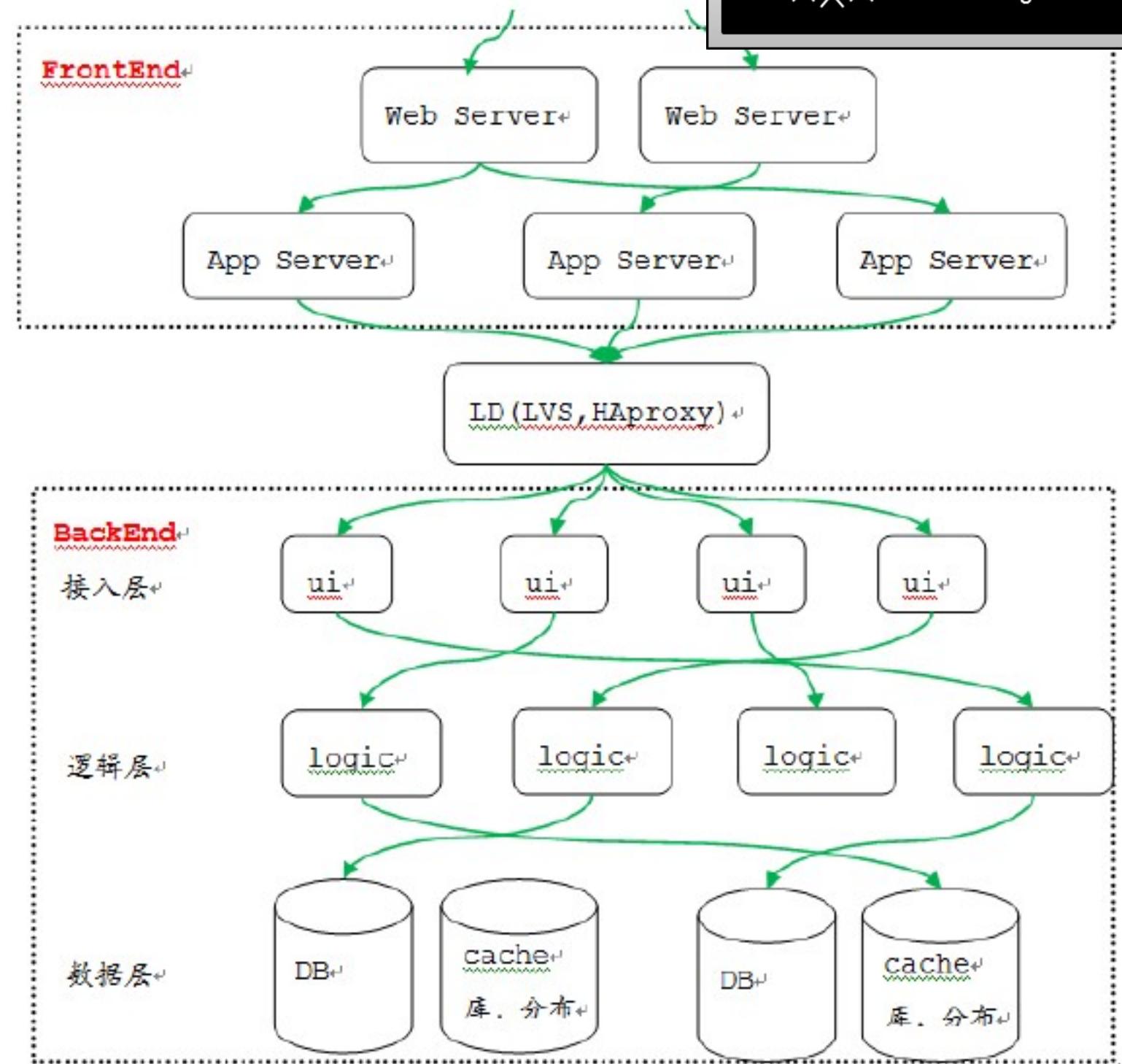
OutLine

- ::: 互联网产品移动后台技术架构
- ::: 接入层的作用是什么
- ::: 接入层Session如何设计?
 - Session复制、Session绑定、Session记录方式、Session高可用等
- ::: 接入层数据安全如何保证?
 - 对称加密、非对称加密、多种方法使用等
- ::: 接入层数据正确性如何保证?
- ::: 接入层高可用设计方案?
 - 无状态、动态扩展
- ::: 接入层高可用设计最佳实践是什么?
- ::: 我们的实践案例;

互联网产品移动后台技术架构

Data Flow

- webServer
- AppServer
- LD (LVS,HAProxy…)
- BackEnd Server
 - 接入层
 - 提交层
 - 业务处理层
 - 数据层
- 本文重点关注
 - BackEnd Server



58帮帮技术架构

线上情况

- 模块

- 30+
- JAVA/CPP

- 请求

- 10亿(IM)+30亿(!IM)
- 同时在线用户数突破100w+

- 机器

- 百台+



(单间出租) 朝阳区七彩商场附近单间出租 (冬暖夏凉, 卫生干净。)

房源编号: 58 日期: 2015-3-19 房源: 466

价格: 400 元/月 押一付一

整体: 1室 0厅 0卫 精装修

楼层: -1层/22层

出租: 主卧 其他 男女不限

区域: 朝阳 - 石佛营 - 七彩商场旁中水金海嘉苑

地址: 北京市石佛营136号(朝阳公园桥往东1100米)

入住: 随时

联系: 赵先生 (经纪人) 马上交谈 [立即在线咨询]

85821351
(电话归属地: 北京市)

郑重提示: 请您在签订合同之前, 切勿支付任何形式的费用

58帮帮技术架构

定位

- 传统IM
- 满足58用户与商户沟通，获取信息

核心功能

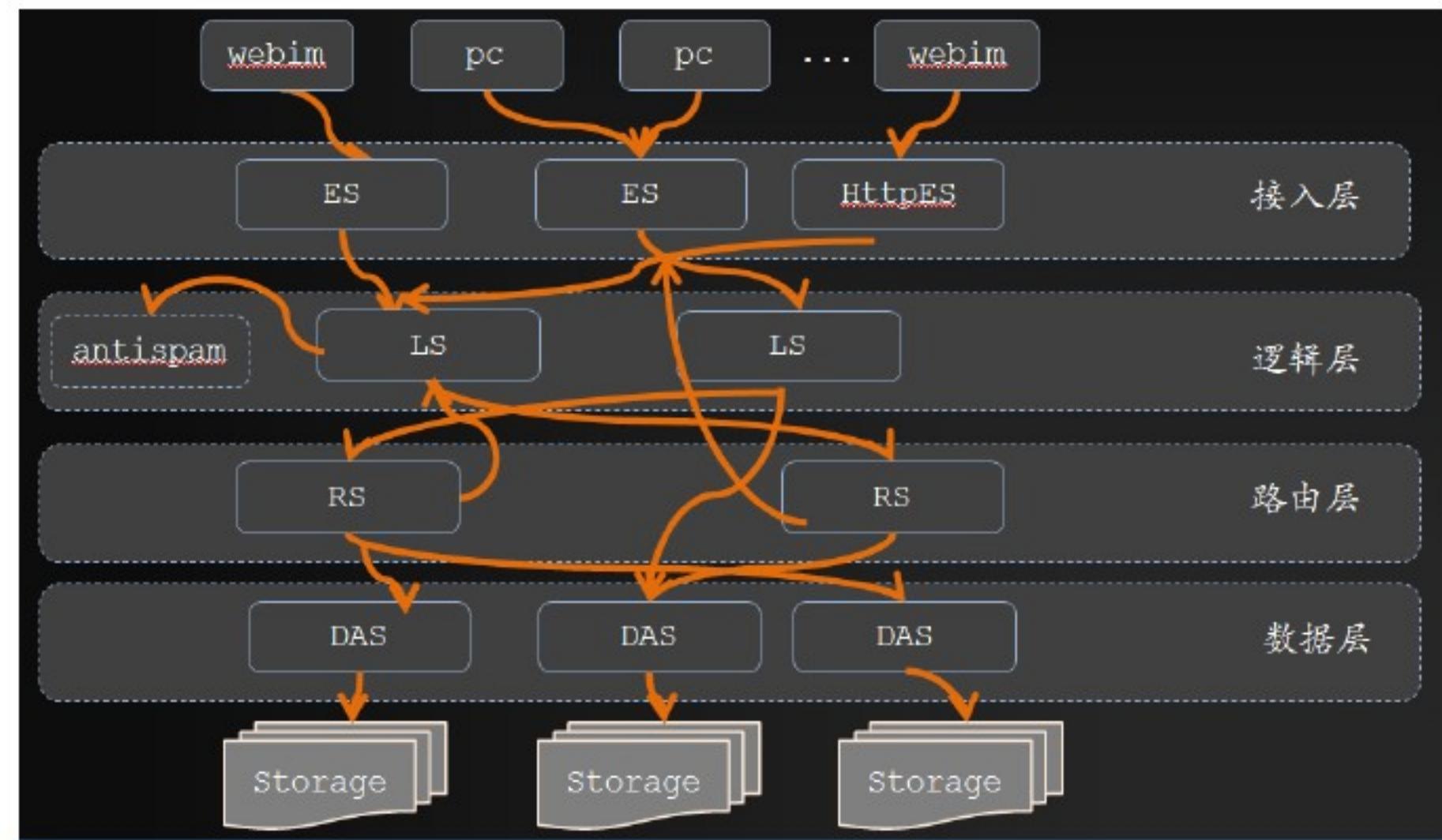
- 用户关系
- 添加好友
- 发送消息





58帮帮技术架构

传统IM技术架构



58帮帮技术架构

传统IM架构如何满足千万同时在线性能？

- 接入层、逻辑层、路由层、数据层
- 无状态设计
- 每层模块动态高扩展
- 模块冗余，高可用性保证
- 动态负载均衡，动态切换可用服务节点
- 优化效果
 - 单机线上支持50W+同时在线
 - 单机线上3w+qps

58转转技术架构

***** 中国移动 22:00 ④ 100% +



定位

- 全国最大的个人闲置交易平台

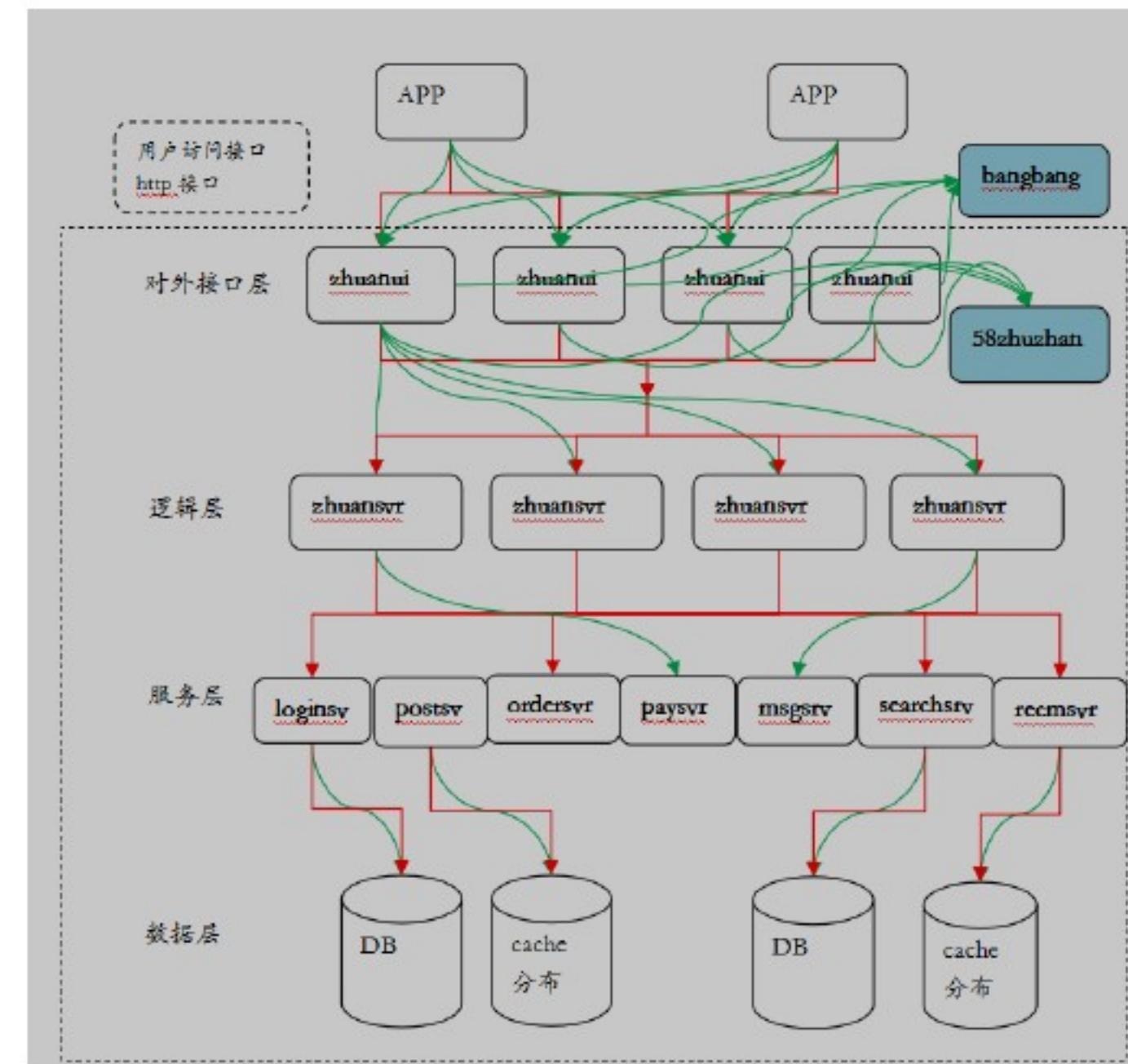
功能

- 用户
- 商品
- 社交
- 交易
- 圈子
- 推荐
- 搜索
- 运营
-

58转转技术架构

架构如何设计

- 功能多
- 业务复杂
- 高可用性
- 交易高安全
- 未来扩展
- 低耦合分层架构
 - 对外接口层
 - 逻辑层
 - 原子服务层
 - 数据层
-





百度空间feed系统架构

feed

2012年03月02日：“又到福来day，各位周末愉快~”

2012啦！新年快乐！ 发布

表情 手机也能发hoho了，快来试试吧！ waphi.baidu.com 还能输入140字

好友动态 全部 文章 相册 分享 HOHO 设置

 solary 发表了新 文章 下午的时光 1小时前
坐在阳台书桌边上，泡上一杯竹叶青，吃点小饼干，翻看新买的一摞书：《数学的诱惑：日常生活中的数字游戏》，《你以为的就是你...》
[阅读全文](#) [分享](#) [评论\(0\)](#)

 一框知天下在 通辽吧 中发贴 【公告】通辽吧垃圾帖子回收站： 6小时前
需要删除的帖子请此处举报

 一框知天下在 贴吧合并吧 中发贴 【申请合并】申请将内蒙古通辽吧合并到通辽吧 7小时前

百度空间feed系统架构

feed系统关注问题

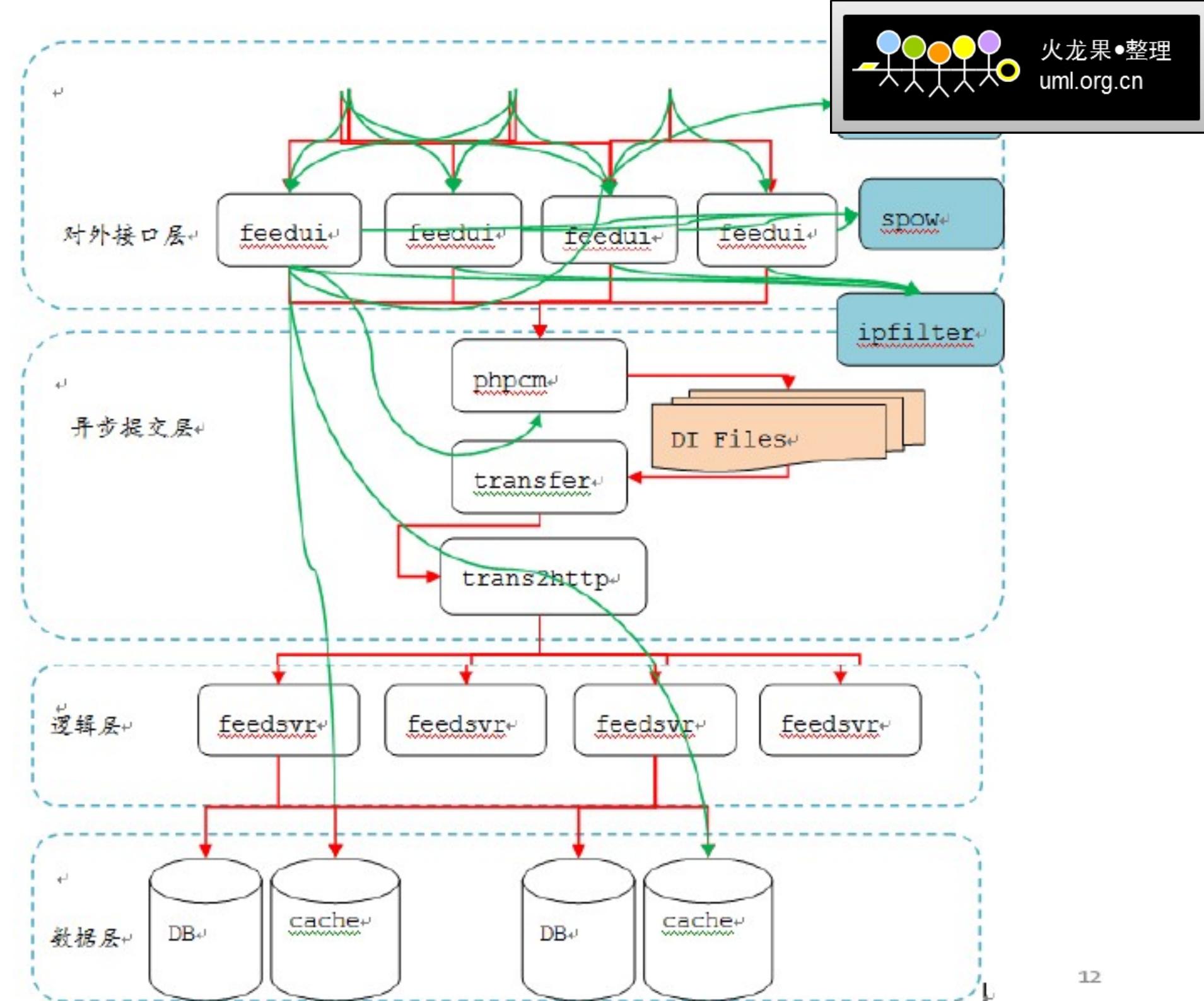
- 获取好友的feed
- 组合好友的feed聚类展示
- 一般按照feed发布时间倒序展现
- 推or拉
 - 拉



百度空间feed系统架构

feed系统最终架构

- 对外接口层
- 异步提交层
- 逻辑层
- 数据层



接入层的作用是什么

» 接入层作用

- 客户端海量长连接管理
- 建立与客户端通信的加密通道
- 数据合法性、正确性校验
- 整合成内部少量的长连接
- Session的管理
- 实施初步的攻防
- 请求转发到逻辑层

接入层session如何设计

Session

- 高可用主要基于服务无状态
- 事实上业务总是有状态的，为什么?
 - 电商网站，需要有购物车记录用户购买信息
 - IM系统中，需要记录用户当前登录状态、好友状态、消息情况等
- 这些有状态的信息会随用户操作页面发生更新
- Session是什么
 - 请求更新使用的上下文对象，称之为会话(Session)

接入层Session如何设计

单机环境设计

- 单机不存在Session共享的问题
- 处理比较简单
- Session放在本机内存
- 高可用无法保证
 - 服务进程挂掉
 - 宕机
 - Session丢失，不可用
- 怎么搞？

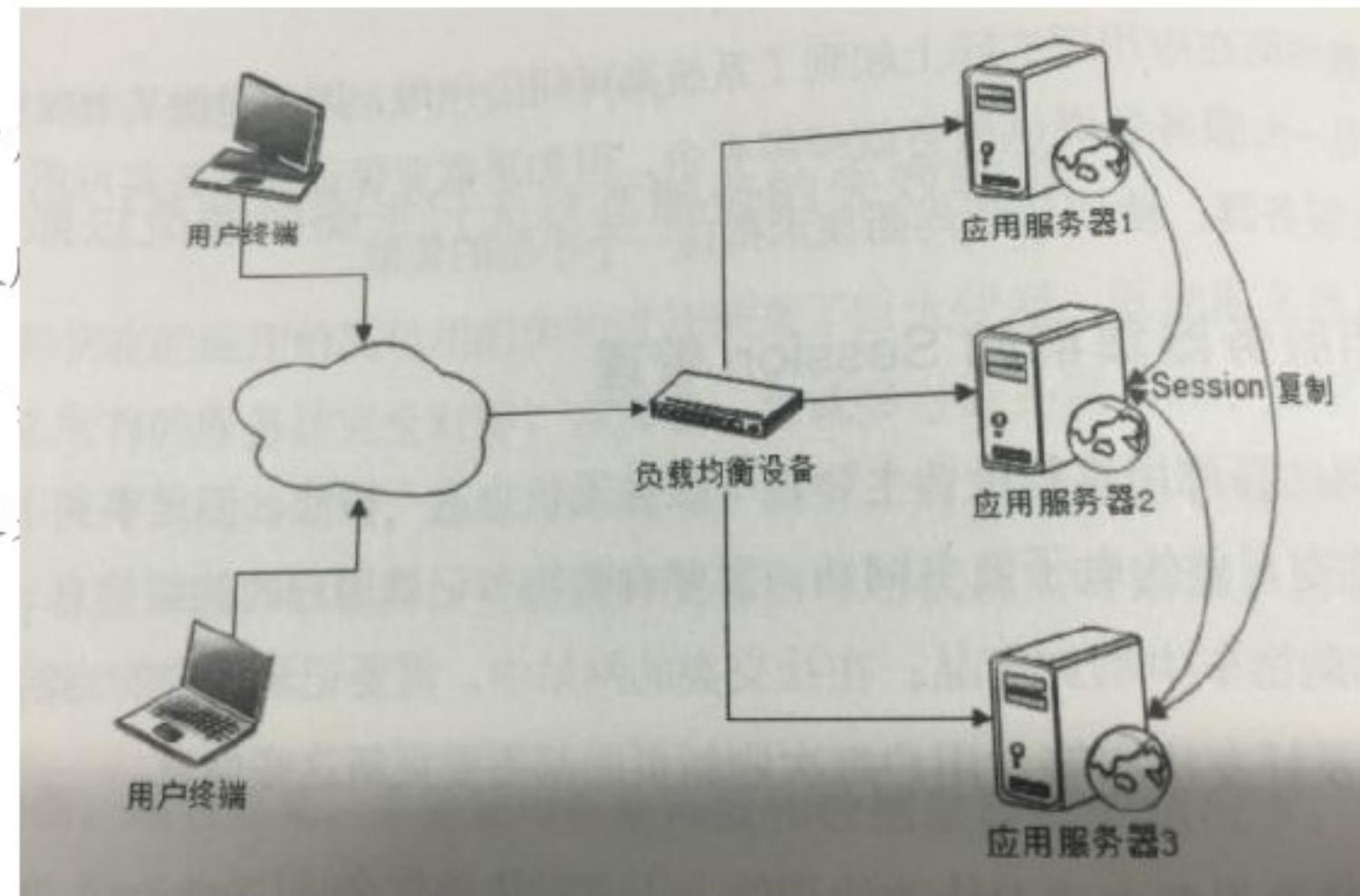


接入层Session如何设计

● 集群（多机）

- Session复制

- 在集群的每台接入机上实现
- 每台接入机上只有一台宿主机
- 宿机一台
- 用户请求直接到宿机



接入层Session如何设计

● 集群（多机）

- Session复制
 - 存在问题
 - 适用于接入层集群较少
 - 接入层集群量大
 - 大量的Session复制通信，占用服务器和网络资源
 - 每天机器存储全量用户Session，内存占用量大，甚至Out Of Memory
 - 大型网站接入层数千台，同时用户在线到千万（IM），不适合这个方案

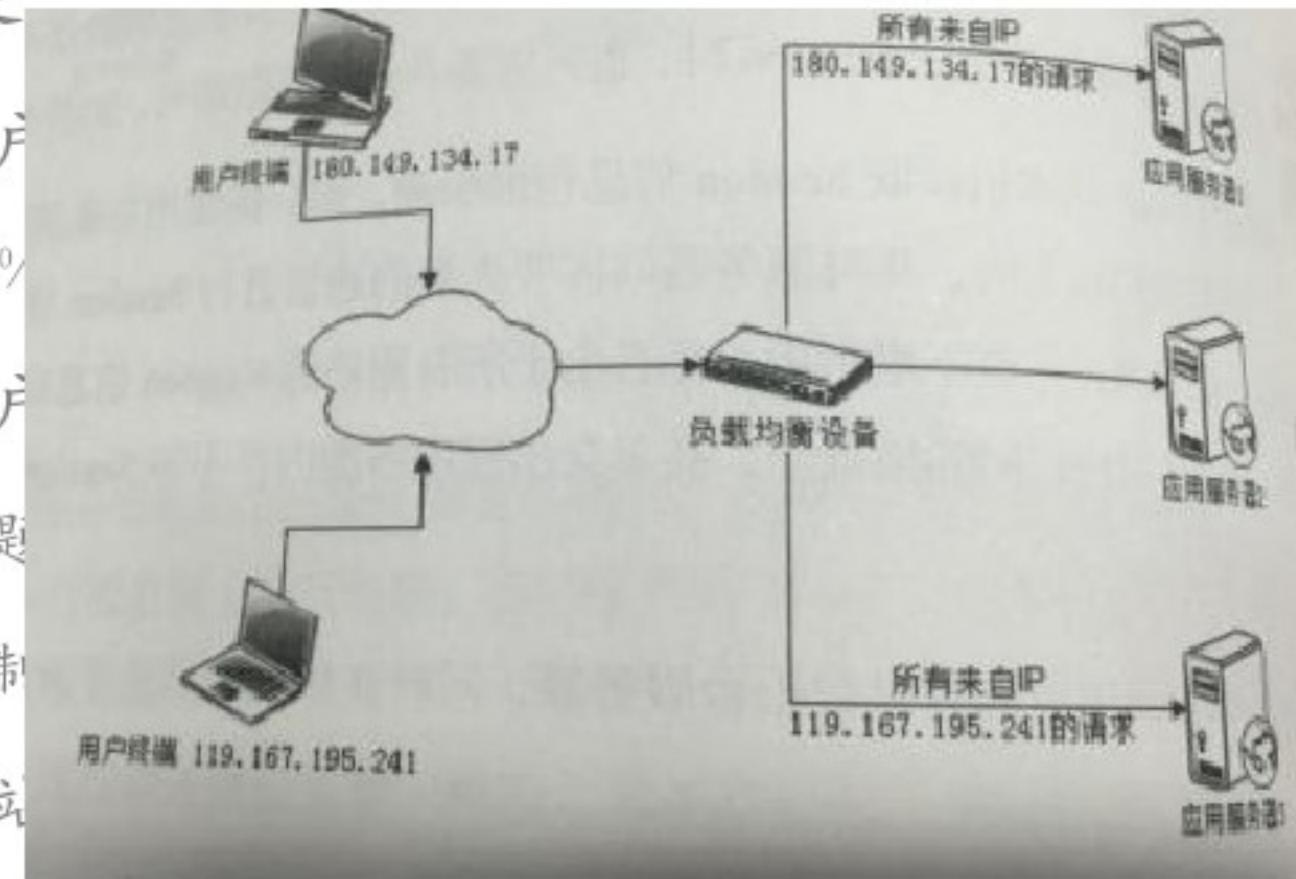


接入层Session如何设计

● 集群（多机）

- Session绑定

- 根据用户
 - uid%
 - ip%
 - 特定用户
 - 单点问题
 - 复制
 - 较少网站



载均衡到特定接入层

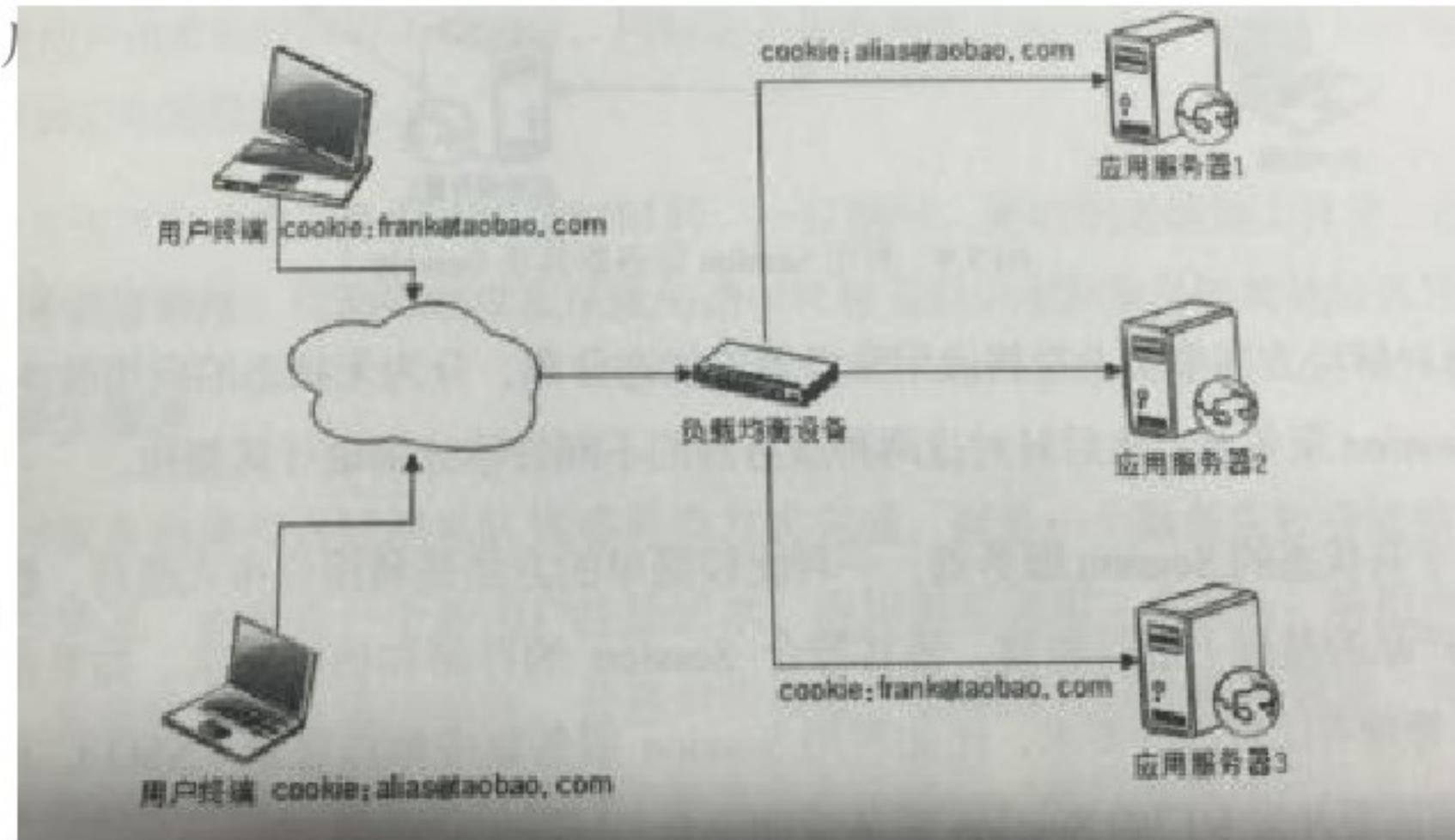


接入层Session如何设计

● 集群（多机）

- 利用

-



拿着Session，服务

接入层Session如何设计

● 集群（多机）

- 利用Cookie记录Session
 - 存在问题
 - Cookie中记录信息大小收限制
 - » 100KB
 - 每次请求都要传输Cookie，性能受影响
 - 用户关闭、清理掉Cookie，用户请求不正常
 - 方案简单，支持服务端的线性伸缩
 - 方案可用性高
 - 很多网站都有使用

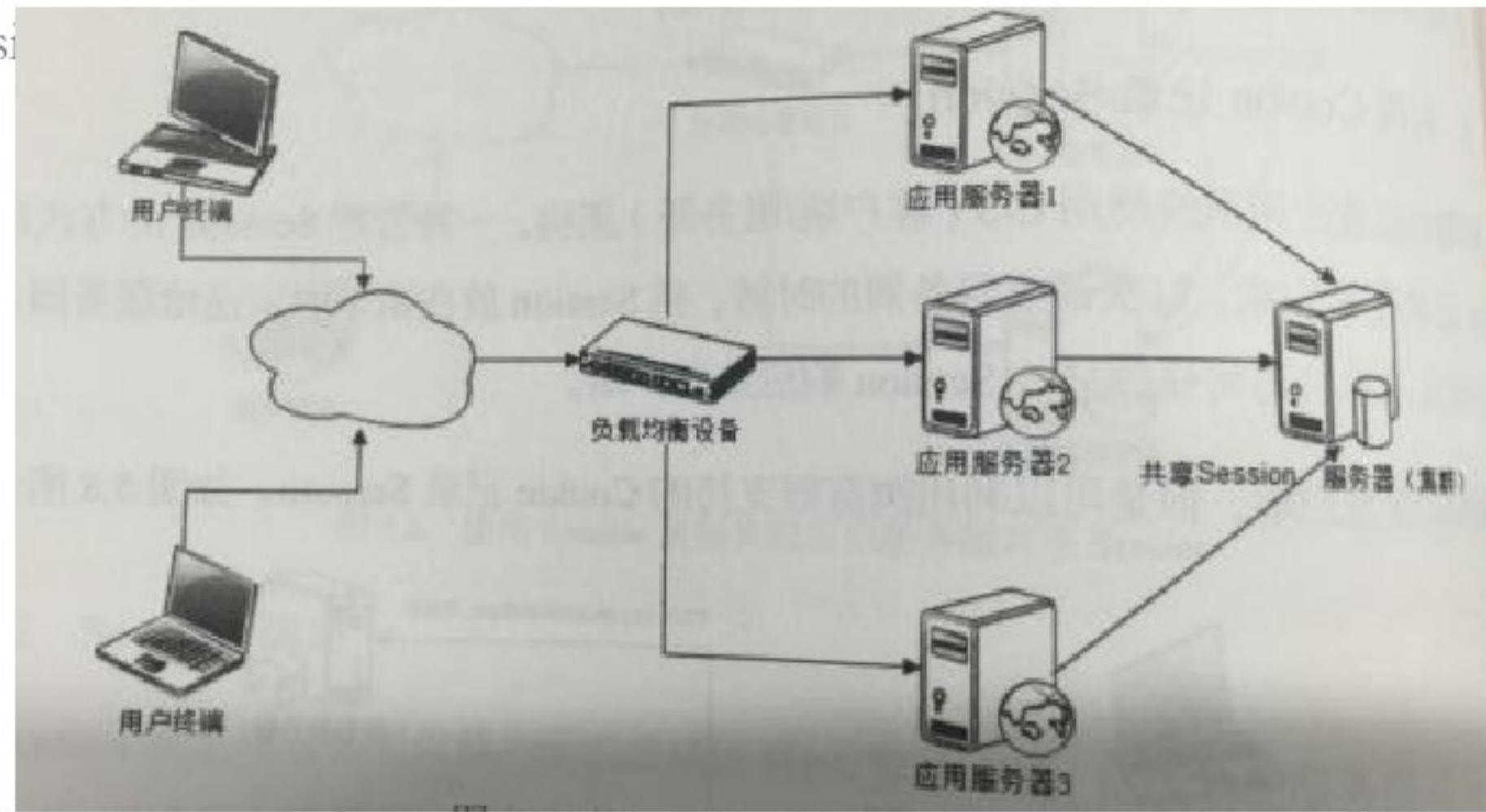


接入层Session如何设计

高可用Session设计

- Session

-
-
-



接入层数据安全如何保证?

● 接入层安全性

- 接入层是客户端和服务端的Interface
- 数据安全重要性不言而喻
- 怎么保证数据安全性?
 - 加密?
 - 加密的安全通道建立

接入层数据安全如何保证?

复杂网络环境下客户端高效滴与服务端建立安全信道方法

- 解决客户端与服务端实现加密会话问题
- 适用于一切客户端
- 58帮帮
- 58转转
-

接入层数据安全如何保证?

名称解释

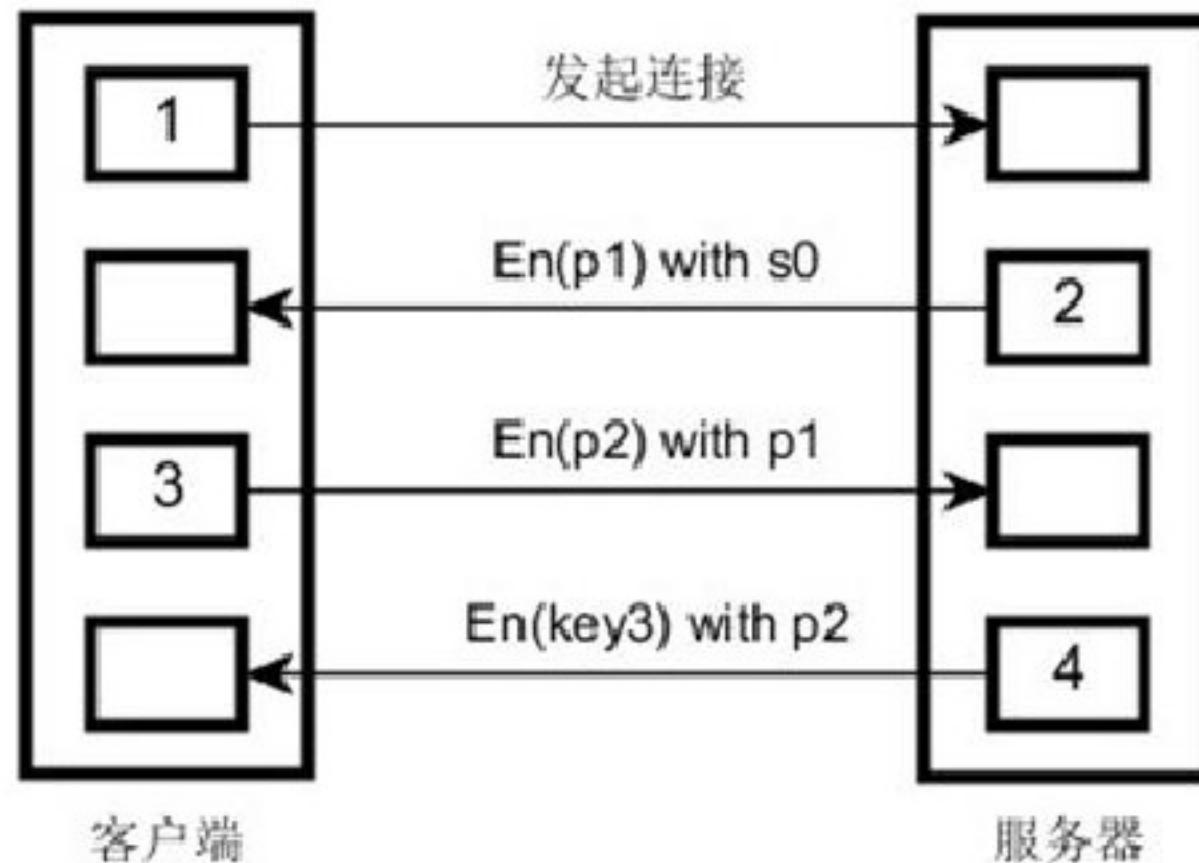
- 对称加密算法:
 - 加密和解码使用同一密钥的加密方案 (AES)
- 非对称加密算法
 - 加密和解密使用一对密钥 (由两个满足一定关系的密钥组成的密钥对) 的中不同的密钥的加密方法 (RSA)
- 公钥
 - 非对称加密算法中公开给大众保密的密钥
- 私钥
 - 非对称加密算法中留给个人保密的密钥
- 会话状态
 - 描述客户端与服务器的一次连接的所有信息集合

接入层数据安全如何保证?

技术实现方案

- 客户端和服务器之间的所有请求都必须加密，为了提高效率，使用对称加密算法；
- 对称加密密钥使用非对称加密算法经过两次协商确定
- 安全信道的建立必须满足
 - 任何第三方无法伪造服务器
 - 在破解客户端代码的情况下，即使截获其他用户发送的加密请求，也无法解密

接入层数据安全如何保证?



约定公私钥对0 : $(p0, s0)$

协商公私钥对1 : $(p1, s1)$

协商公私钥对2 : $(p2, s2)$

对称密钥3 : key3

公密钥

(p)public 公钥
(s)security 私钥

0为事先约定
1和3为服务器随机生成
2为客户端随机生成

接入层数据安全如何保证?

技术实现方案

- 约定公私钥对0：写死在代码中的公私钥（公私钥池，服务器每次选一个，并告诉客户端每次选中的是池中的哪一个），用于客户端验证请求的确来自服务器；
- 协商公私钥对1：服务器随机生成的协商密钥；
- 协商公私钥对2：客户端随机生成的协商密钥；
- 对称密钥3：服务器随机生成的对称密钥，用于最终的对称加密

接入层数据安全如何保证?

❖ 技术实现方案

- 客户端

- C->S
- S->C
- C->S
- S->C
- 通讯使用对称密钥3

接入层数据安全如何保证?

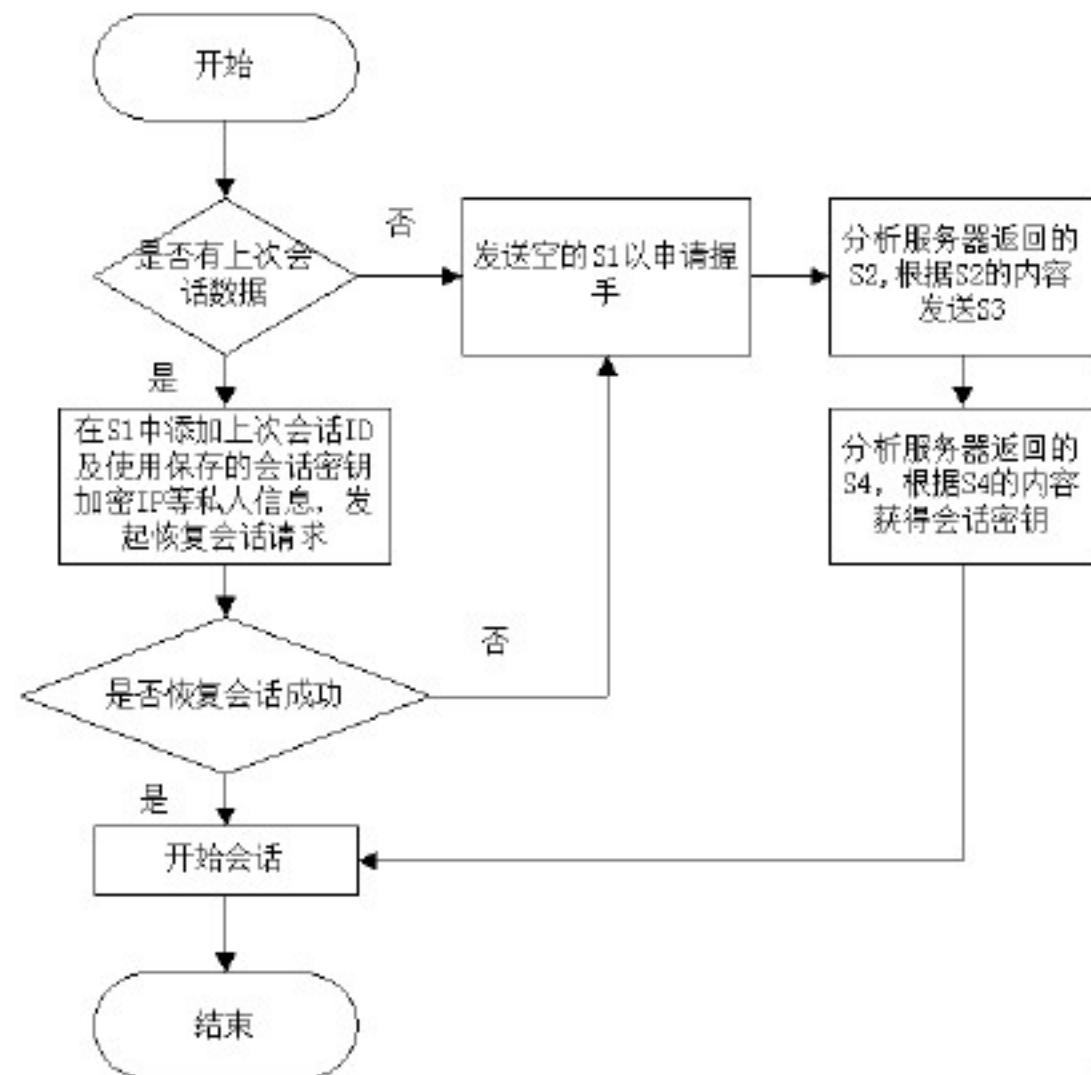
❖ 技术实现方案

- 如何进行快速会话恢复

- 第一步
- 第二步

接入层数据安全如何保证?

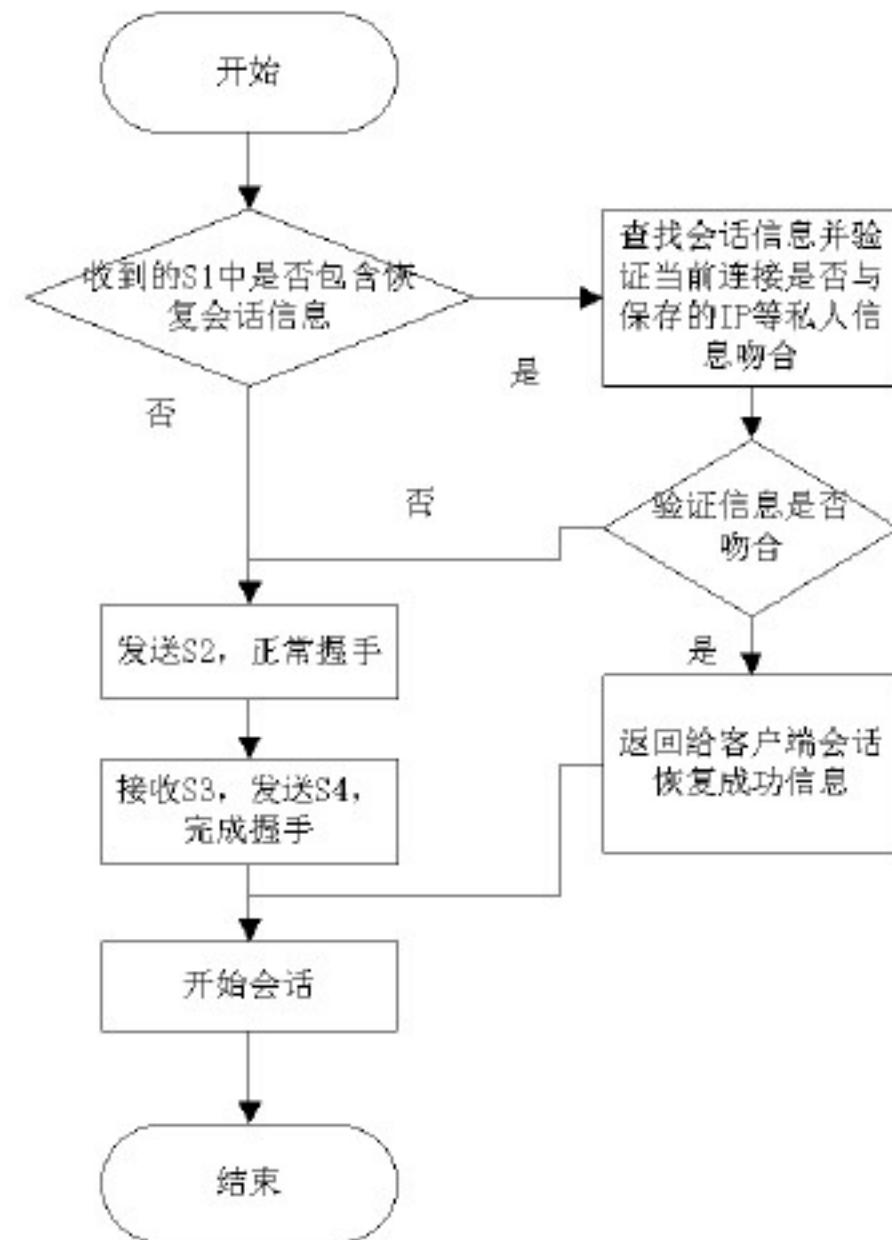
客户端握手流程





接入层数据安全如何保证?

服务端握手流程



接入层数据安全如何保证?

使用HTTPS

- HTTPS

- 提供了数据安全的加密方式
- 单向加密
- 双向加密

- 使用场景

- 交易
- 支付
- 金融
-

接入层数据安全如何保证?

❖ 使用HTTPS

- HTTPS
 - 单向加密
 - 不安全
 - 中间人攻击

接入层数据安全如何保证?

使用HTTPS

- HTTPS
 - 双向加密
 - 安全
 - 客户端证书
 - 配合
 - 接口分级
 - HTTPS
 - HTTPS+短信验证

接入层数据正确性如何保证?

❖ 数据加密

- 解决数据明文的问题
- 即使截获，无法破解明文
- 篡改无法避免
- 数据正确性需要保证
- 如何保证？

接入层数据正确性如何保证?

如何保证

- 数字签名
 - md5sum做签名
 - 其他双方约定规则做签名
- 服务端收到后
- 先按照规则生成md5sum值
- 和数据包里的md5sum值比较是否一致
- 一致说明没问题
- 不一致数据被篡改
- 需要丢弃

接入层数据正确性如何保证?

数字签名举例

- 第一步，设所有发送或者接收到的数据为集合M，将集合M内非空参数值的参数按照参数名ASCII码从小到大排序（字典序），使用URL键值对的格式（即key1=value1&key2=value2…）拼接成字符串stringA。
- 重要规则：
 - 参数名ASCII码从小到大排序（字典序）；
 - 如果参数的值为空不参与签名；
 - 参数名区分大小写；
 - 验证调用返回或微信主动通知签名时，传送的sign参数不参与签名，将生成的签名与该sign值作校验。

接入层数据正确性如何保证?

❖ 数字签名举例

- 第二步，在stringA最后拼接上key得到stringSignTemp字符串，并对stringSignTemp进行MD5运算，再将得到的字符串所有字符转换为大写，得到sign值signValue。



接入层数据正确性如何保证?

数字签名举例

```
<xml>
<appid>wxd930ea5d5a258f4f</appid>
<mch_id>10000100</mch_id>
<device_info>1000</device_info>
<body>test</body>
<nonce_str>ibuaiVcKdpRxkhJA</nonce_str>
<sign>9A0A8659F005D6984697E2CA0A9CF3B7</sign>
<xml>
```

appid : wxd930ea5d5a258f4f
mch_id : 10000100
device_info : 1000

RxkhJA

4-100001008nonce str-ibuaiVcKdpR

9A0A9CF3B7"

接入层数据正确性如何保证?

❖ 数字签名

- 安全进一步提升
 - 约定固定字符串，参与加密
 - securityStr=x33@#\$/3
 - 只有双方知道

高可用接入层如何设计?

模块和数据分离

- 接入层模块无状态
 - 动态伸缩
 - 冗余
- Session等数据分布式存储
 - 数据冗余保证
 - 高可用性保证

高可用接入层最佳实践是什么

❖ 模块和数据分离

❖ Session绑定

- 每个Session同步复制

❖ 不存储Session

- 接入层

我们的实践案例

项目背景

- 全国最大的C2C交易平台
 - 同之前叙述

我们的实践案例

转转接入层设计

- 设计目标

- 高可靠性
- 灵活扩展接口而不修改代码
- 安全性高

 - 交易、订单、支付

- 效果

- 负责海量APP端的接入
- 负责接入请求的合法性校验和安全校验
- Logic层专注逻辑实现

我们的实践案例

转转接入层设计

- 基于java的反射机制
- 配置uri到接口的关系和权限
- 使用单例模式初始化接口对象
- 权限校验
- 远程调用

安全性

- 双向HTTPS

```
--  
name: LoginBannerService  
scfurl:tcp://zzlogic/LoginBannerService  
scfclass:com.bj58.zhuanzhuan.logic.contract.ILoginBannerService  
--  
  
=====method cluster ======  
#priorit the right to access some function: 0 is no uid can access,  
----  
#[k=v]: common key = value, which every object will contain,  
#next time the history common kv will be discard.  
#ShareLogicService  
-[service:ShareLogicService]  
--  
name:getShareWXConfig  
scfmethod:getShareWXConfig  
priority:0  
--  
  
#ListingLogicService  
-[service:ListingLogicService]  
--  
name:getRecommendInfoFromErShouAndZhuan  
scfmethod:getRecommendInfoFromErShouAndZhuan  
priority:0  
--
```

- 使用单例模式初始化接口对象
- 权限校验
- 远程调用

我们的实践案例二

项目背景

- 58帮帮

- 58商户和用户沟通平台
- 海量连接管理
- 整流
- Session控制
- 请求转发
- 反作弊
 - 连接频率、发包频率、发包速率等
 - 对IP/UID等指标实施封禁



本课总结

- ::: 互联网产品通用技术架构
- ::: 接入层的作用是什么
- ::: 接入层Session如何设计?
 - Session复制、Session绑定、Session记录方式、Session高可用等
- ::: 接入层数据安全如何保证?
 - 对称加密、非对称加密、多种方法使用等
- ::: 接入层数据正确性如何保证?
- ::: 接入层高可用设计方案?
 - 无状态、动态扩展
- ::: 接入层高可用设计最佳实践是什么?
- ::: 我们的实践案例;