

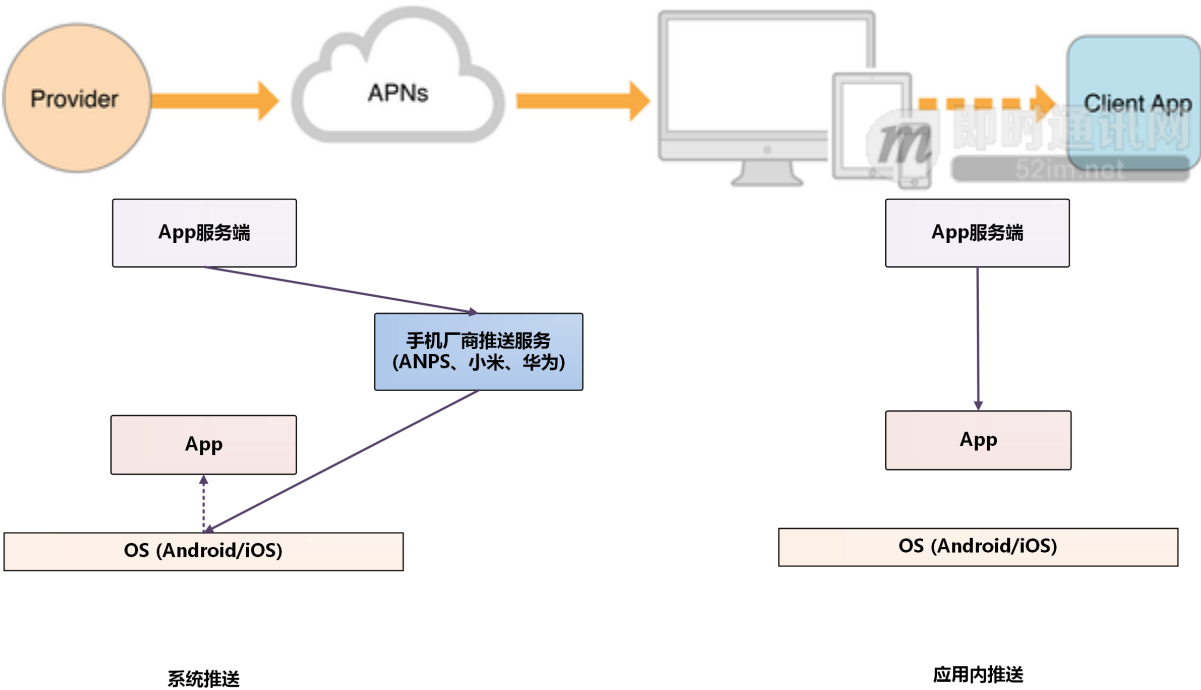
# 设计文档

## 互联网长连接服务实践

### 第一章 背景

通信是大多数App开发设计过程中的重要环节，其包含网络模型、传输协议、交互接口等多个组成部分。通信主要有请求响应(Request/Response)和发布订阅(Publish/Subscribe)两种模式。发布订阅主要用于推送场景，服务端虽为发起方，但需要客户端主动维系与服务端的数据连接。请求响应虽然是最普遍的交互场景，但很多大型App都是两种模式兼而有之。

移动端的推送可分为系统推送和应用内推送，应用内推送更多是推送应用数据，而系统推送主要用于用户级消息通知，受三方平台的推送频次和实时性约束。



网络模型从连接保持维度可以分为短连接和长连接。长连接大多采用MQTT、Websocket或其他基于TCP的私有协议，实现难度相对较大，但通信效率高，对应用内推送场景支持好。微信、滴滴等App以长连接为主，美团等App使用短连接为主。在日常使用上明显可以感受到微信在弱网环境下体验要更好。

优理宝App目前使用短连接与后端交互，应用内推送也是采用轮询的方式。长连接技术方案的启用意味着需要为此建立与日活客户规模相匹配的接入服务器集群，此外还需要连带规划建设包括监控统计、均衡容错、管控后台等一系列周边设施。花费巨大的技术投入若仅仅用来应对推送这一低频场景在投入回报上较为铺张，因此我们考虑基于此长连接做成双向通信，既支持发布订阅也支持请求响应，建设成面向App前后端开发人员的整体服务开发框架。

### 第二章 产品简介

互联网接入总线（简称eBus）是兴业证券自主研发的面向App前后端开发人员的整体服务开发框架。作为服务开发框架，总线包含有移动端SDK (Android、iOS、JS)、服务端SDK和管理后台。移动端SDK主要包含接口请求和主题订阅两个主要功能，服务端SDK包含互联网服务开发框架和推送接口，管理后台是开发运维人员的管控工具。

#### 2.1 移动端SDK

无论RESTful还是SOAP，跨进程调用的本质是定位到服务端目标函数并传入实参和上下文以驱动相应逻辑的执行。任何远程调用，调用方都需要向目标进程传递三个信息：目标函数定位、实参和上下文。以HTTP调用为例，URL和Method是目标函数定位信息，实参是链接参数和Body中的数据，上下文则以Cookie或其他Header的形式携带。

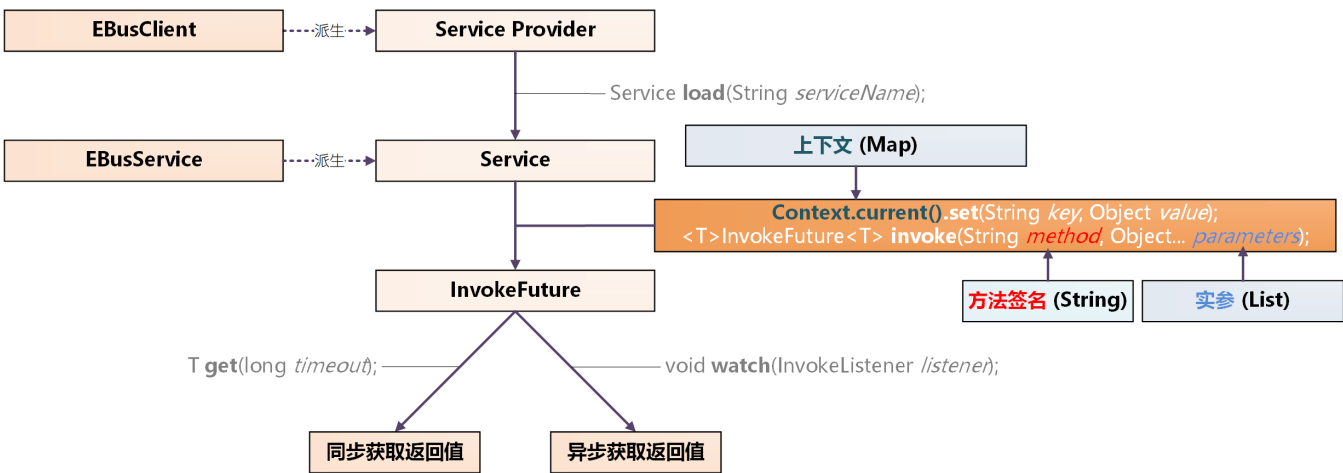
我们抽象了各类Web框架和RPC框架，设计了一套通用调用模型。该模型定义了服务提供者(Service Provider)、服务、调用凭据(Invoke Future)的接口，同时定义了RPC传递三信息的标准。eBus移动端SDK其实是该调用模型的实现，而早先我司的Lisa微服务则是该调用模型的服务端实现。

- A. 方法签名：标记服务实现类的指定方法的字符串，格式为：methodName(parameterType)，例如add(int, int)。同时为了简化使用，可以简写为add，前提是服务实现类不能有add的方法重载。
  - B. 实参
  - C. 上下文
- 上下文包含需要透传给服务实现类的数据，采用Map结构表示，为了方便在各平台上传输上下文的Value只能是基础类型(isPrimitive = True)。互联网长连接服务的移动端SDK便是基于通用调用模型构建而成的。开发者可以使用很简短的代码实现后端接口的调用，范例如下：

```
Service stockService = eBusClient.load("ulb-stock-service");
//
InvokeFuture<Double> future = stockService.<Double>invoke("query(String)", "601377");
//
Double result = future.get(1000);
SDK

StockService service = eBusClient.load("ulb-stock-service", StockService.class);
//
InvokeFuture<Double> future = service.query("601377");
//
Double result = future.get(1000);
```

上述示例代码中没有服务端域名IP信息、也没有URL和传输协议，屏蔽了网络模块的底层技术细节，极大地简化的App开发人员的工作。



2.2 服务端SDK

服务端SDK提供了编写eBus服务的开发框架和推送接口。

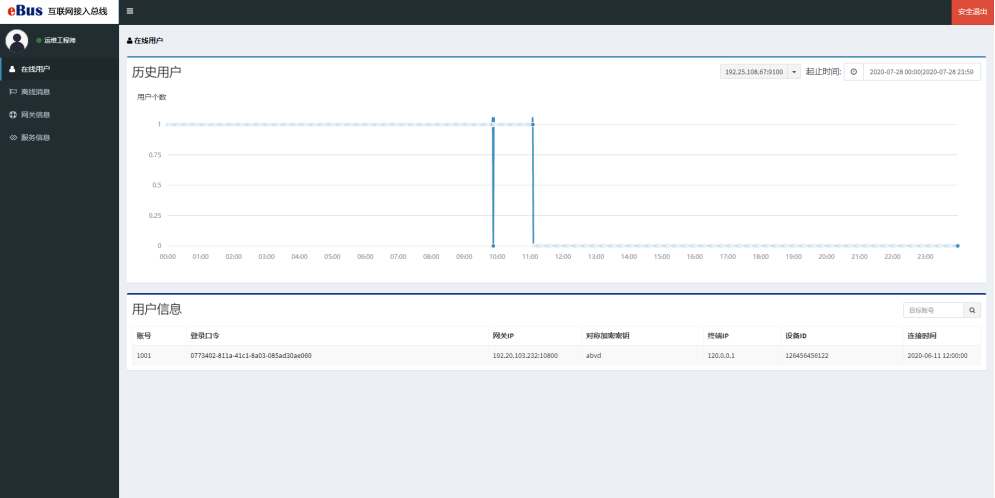
```
@Service
@EBusService(name="ebus_demo_service")
public class EBusDemoService {
    /**
     */
    public int plus(int i, int j) {
        return i + j;
    }
}
```

开发者仅仅需要为服务实现类添加@EBusService注解即可将本地Bean发布至互联网供移动端代码调用。同时，服务端SDK包含了ebus\_push\_service、ebus\_auth\_service等微服务，用于向App推送应用内消息。

接口 功能 备注  
send 发送在线消息 仅在线用户可以收到，离线报错  
post 投递离线消息 离线用户登录后将收到，可以查询历史

2.3 管理后台

管理后台是总线的运维平台。通过管理后台可以查看网关信息、在线用户等信息，同时服务端SDK会将后端服务的注释信息提交到管理后台，开发者无需额外编写接口文档，移动端开发人员可直接在线查看后端接口。



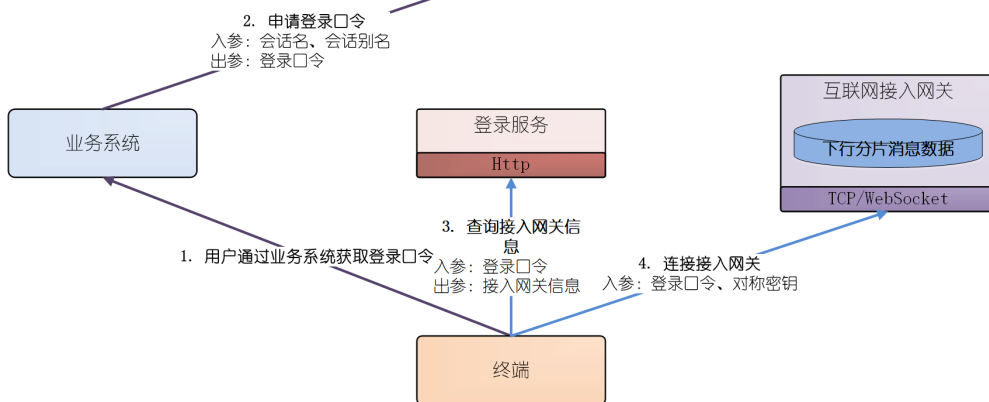
The screenshot shows the 'eBus 互联网接入总线' management console. The '离线消息' (Offline Messages) section is active. It features a table for message details and a pagination bar at the bottom.

账号	信息	创建时间
SuIb123456	["body":"离线推送测试","type":"msg"]	2020-07-01 16:12:57
SuIb123456	["body":"离线推送测试","type":"msg"]	2020-07-01 16:12:48
SuIb123456	["body":"test股票涨停11111","type":"msg"]	2020-07-01 16:09:14
SuIb123456	["body":"test股票涨停11111","type":"msg"]	2020-07-01 16:08:02
SuIb123456	["body":"test股票涨停11111","type":"msg"]	2020-07-01 16:07:58
SuIb123456	["body":"test股票涨停11111","type":"msg"]	2020-07-01 16:06:53
SuIb123456	["body":"test股票涨停11111","type":"msg"]	2020-07-01 15:59:11
test_topic2	--mqtt_pub,sample<\${clientId}#EÖ	2020-06-11 13:27:39
test_topic5	--mqtt_pub,sample<\${clientId}#EÖ	2020-06-11 13:27:33
test_topic4	--mqtt_pub,sample<\${clientId}#EÖ	2020-06-11 13:27:33

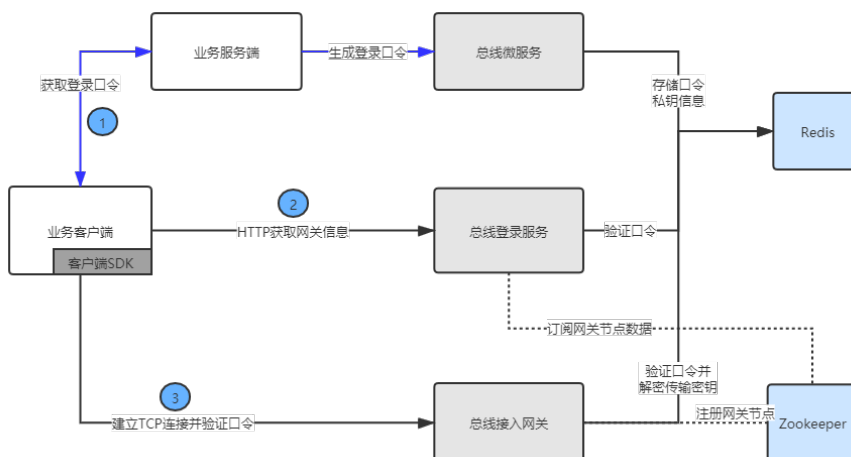
合计19394条信息, 每页10个信息

### 第三章 系统设计

#### 3.1 总体设计



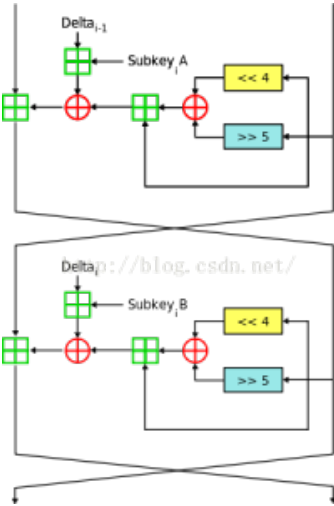
- A. 登录服务  
登录服务面向移动端SDK提供HTTP接口用于下发分配的网关信息，登陆服务承担负载均衡、重连协调、身份认证功能。
- B. 接入网关  
与客户端建立长连接，通信基于MQTT协议做安全传输，移动端通过pub上传服务请求、通过sub获取服务响应和推送消息。
- C. 总线微服务  
向内部各应用提供消息推送、终端在线状态、登录口令申请等服务。
- D. 请求消息队列  
移动端对后端服务的调用请求都会由接入网关写入RocketMQ，并由对应的服务提供者消费并处理，处理结果再走实时推送至对应网关并下发到移动端。



### 3.2 通道安全

MQTT协议有登录过程，但传输不支持加密，payload内容可能被窃取。因此总线的登录过程在设计时包含了通道加密逻辑。

- A. App开屏时调用App后台RESTful接口获取总线登录公钥Pub;
- B. App后台调用eBus微服务用用户账户Account申请登录公钥Pub;
- C. App调用总线登录服务RESTful接口传入MD5(Pub)获取接入网关IP等信息;
- D. App随机生成对称加密密钥Secret，并使用Pub加密Secret，通过MQTT Login协议连接指定网关，传入加密后的Secret;
- E. 接入网关通过私钥解密出Secret，后续与App通信的payload一律使用Secret加密;



### 第四章 结语

互联网接入总线是我们实践App长连接通信模式的关键步骤，其为移动端开发同学提供了除了HTTP以外另一种通信开发模式。极大地降低了App的开发成本，新开发App无需申请域名、加密证书、搭建网关、均衡负载和编写接口文档。此外eBus的运维平台精准地提供了在线用户报表、服务API文档和历史消息，实现了高效的运维管控。

互联网接入总线是我们涉足长连接通信这一复杂课题的重要实践，其后续的推广和普及相信会有效提升我司整体技术水平和研发效率。

**资源下载：**

[ebus客户端SDK（安卓和iOS）](#) [JavaScript Demo](#) [Moe Demo](#) [安卓版Demo](#)