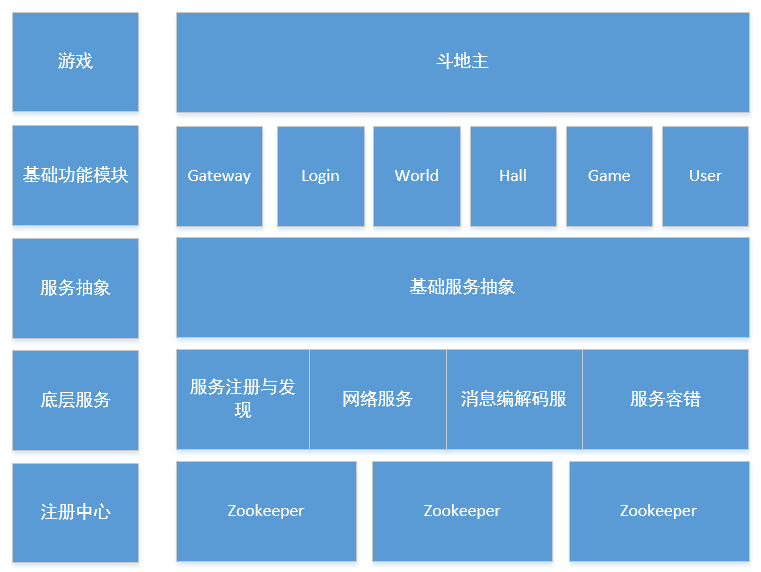
**Server游戏框架开发指南**

**简介：**

**Server**游戏开发框架是款为棋牌类游戏设计的游戏后端框架。**Server**被设计为分布式的，高可用的，可多机负载的，以方便游戏后期的扩容。

**Server**底层实现了网络通信，服务注册与发现，上层实现了Gateway，Game，User，Hall等基础功能模块，并提供统一的业务处理接口，模块配置接口，方便开发人员对游戏的功能进行具体的逻辑开发。

**模块架构：**



**功能详解：**

注册中心：

Server中，使用Zookeeper作为注册中心辅助组件，框架本省封装了注册服务只需要在配置中心配置，便可将业务服务注册至注册中心，其他需使用此服务的模块可以自动发现此服务的存在，并建立连接，发起请求调用。同时，注册中心负责维护模块的运行状态，一旦发现服务有异常，将主动通知依赖模块，关闭连接，取消服务依赖。

网络服务：

框架本身封装了网络服务，将网络通信抽象为接口调用。使用Netty作为网络通信基础框架，实现网络连接建立，消息收发，心跳保持等基本网络功能。实现消息单向发送（Oneway），请求与返回（Request&Response）的命令传递方式，方便上层应用基于这些模式实现消息应用。

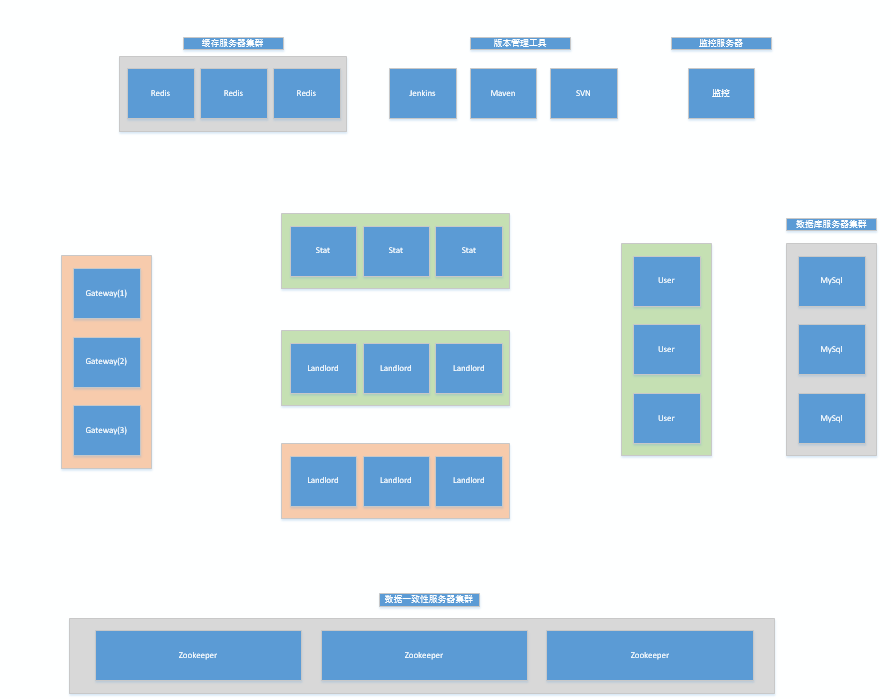
基础功能模块：

Server中实现了游戏中部分基础模块。包括游戏网关，登录，大厅，世界，游戏，和基础服务等。目前只是实现了部分基础服务功能，随后可任意扩展。

游戏：

基于基础模块，可以完成上称游戏应用的开发。

**服务架构：**



所有服务支持分布式部署，消息支持定向发送和负载均衡发送两种方式。模块分为有状态（橘色）和无状态（绿色）两种。无状态服务将数据存在分布式缓存（Redis）中，多机状态唯一。有状态服务将状态维护在本地内存，断电即消失，需要持久化的数据根据业务条件进行持久化。

模块功能如下：

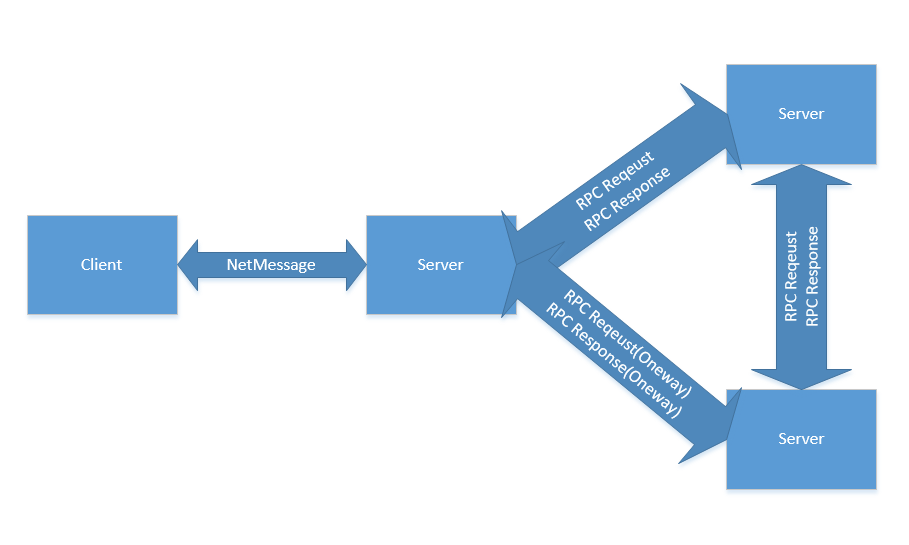
Gateway：服务网关。负责维护客户端连接状态，Client消息转发及路由，服务器消息推送至Client。

Hall：游戏大厅服务。负责分配维护游戏服务器，建立牌桌，分配牌桌等功能。

Game：游戏服务。负责游戏的进行。Player游戏状态的维持，牌桌状态维持等游戏功能。

User：用户基础服务。连接数据库，提供用户信息查询，用户信息更新等基础服务接口。

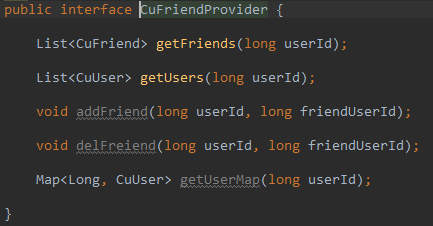
**消息传递：**



Client到Server之间，使用Protobuf消息编码进行消息传递，具体消息内容将在随后介绍。

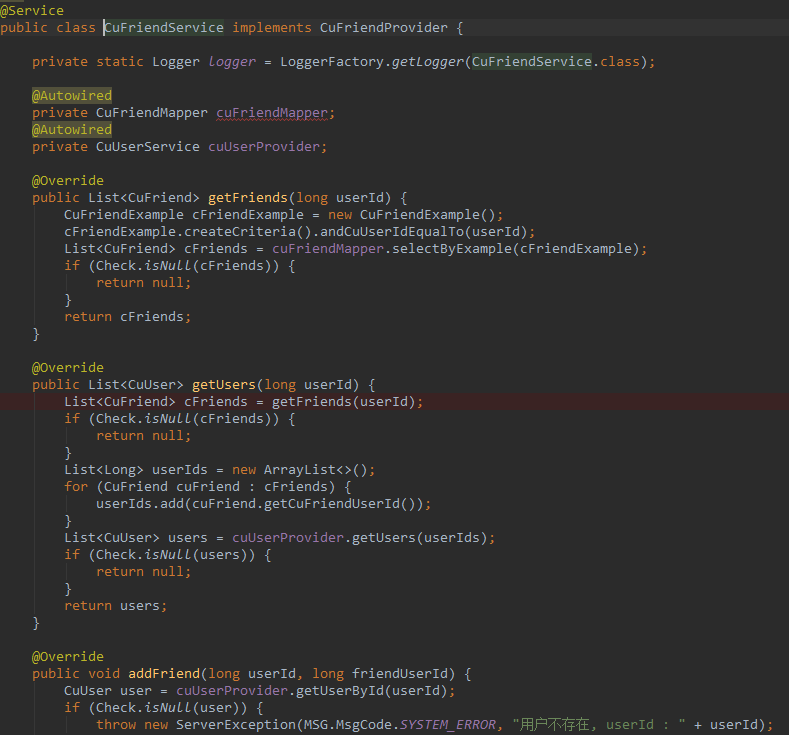
Server之间采用接口调用方式机型通信。消息分为Oneway（没有返回）和有返回两种模式。RPC消息为请求返回式消息，主要用于基础服务访问及应用类型服务访问。

**开发示例：**



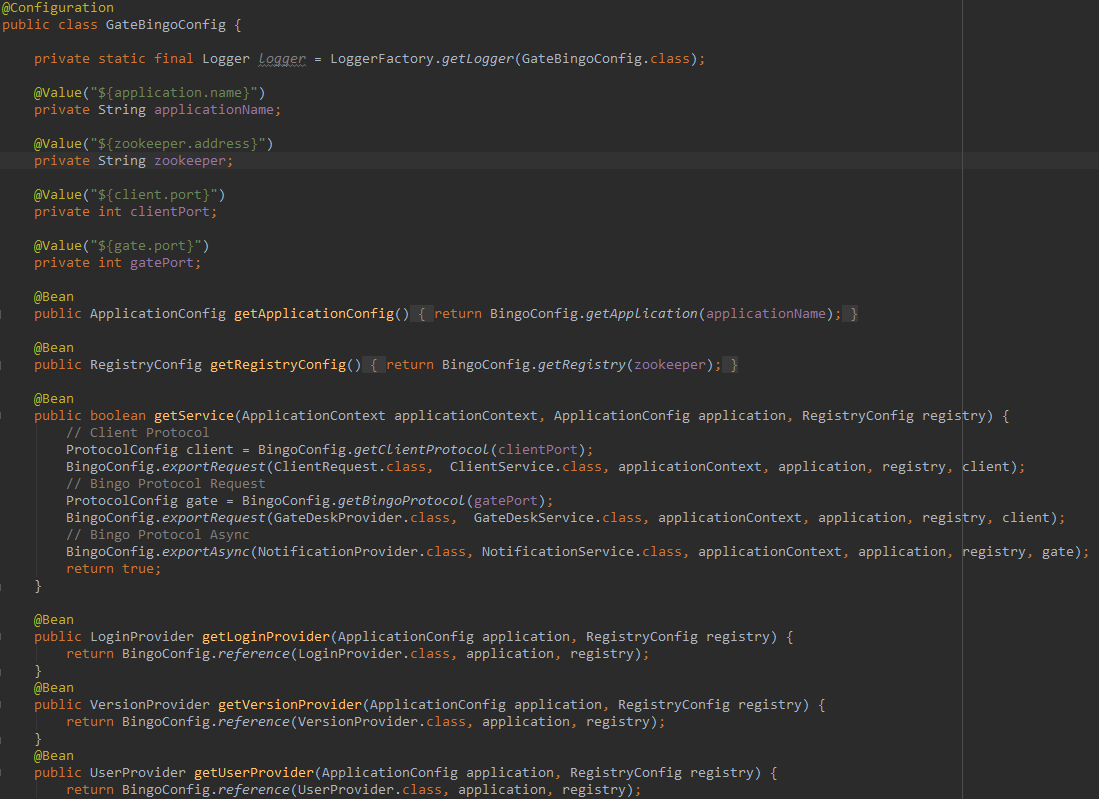
定义服务接口：

定义标准的Java接口用于服务提供接口。



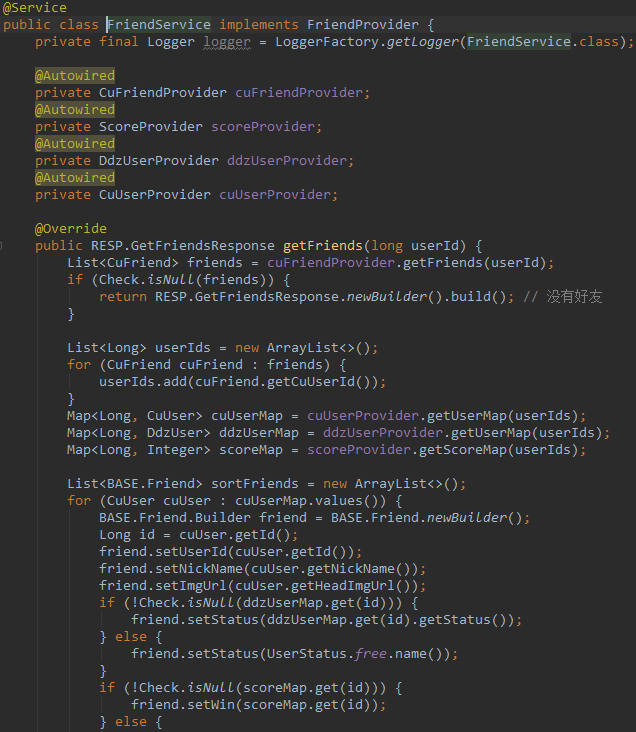
服务实现：

根据接口实现方法。



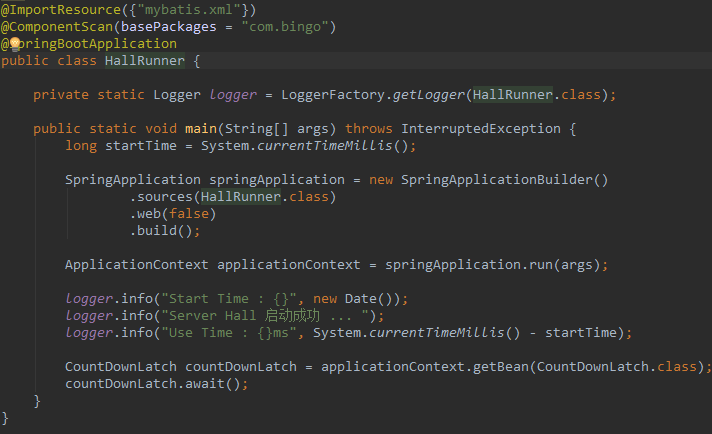
服务配置：

配置此模块依赖的服务及提供的服务，将接口配置到配置代码中，用于服务的注册与发现。



服务引用：

在需要使用服务器的地方,使用@Autowired引用配置好的服务,即可访问远端服务。



服务配置，启动：

模块服务使用Springboot做全局管理。

使用springApplication启动服务。