棋牌游戏服务器架构

## 1服务器简要说明

### 1：**游戏平台服**

主要为http通讯的api服务器，可按业务拆分。尽量用golang,java等语言实现，php相对应对一些特殊的需求比较繁琐。

功能：提供web pc端，手机端h5等各种周边业务的后端api接口。提供游戏app的登陆，支付，商场，商务，活动等非游戏逻辑的api接口。

只提供数据接口，不提供页面表现等。前后端分离便于维护及扩展。

### 2：**游戏大厅服**

分布式多开部署，具体客户端连接那个大厅有登陆平台服分配

1负责客户端的连接维护

2当前所在服务器的信息维护，转发游戏服与客户端的消息通讯

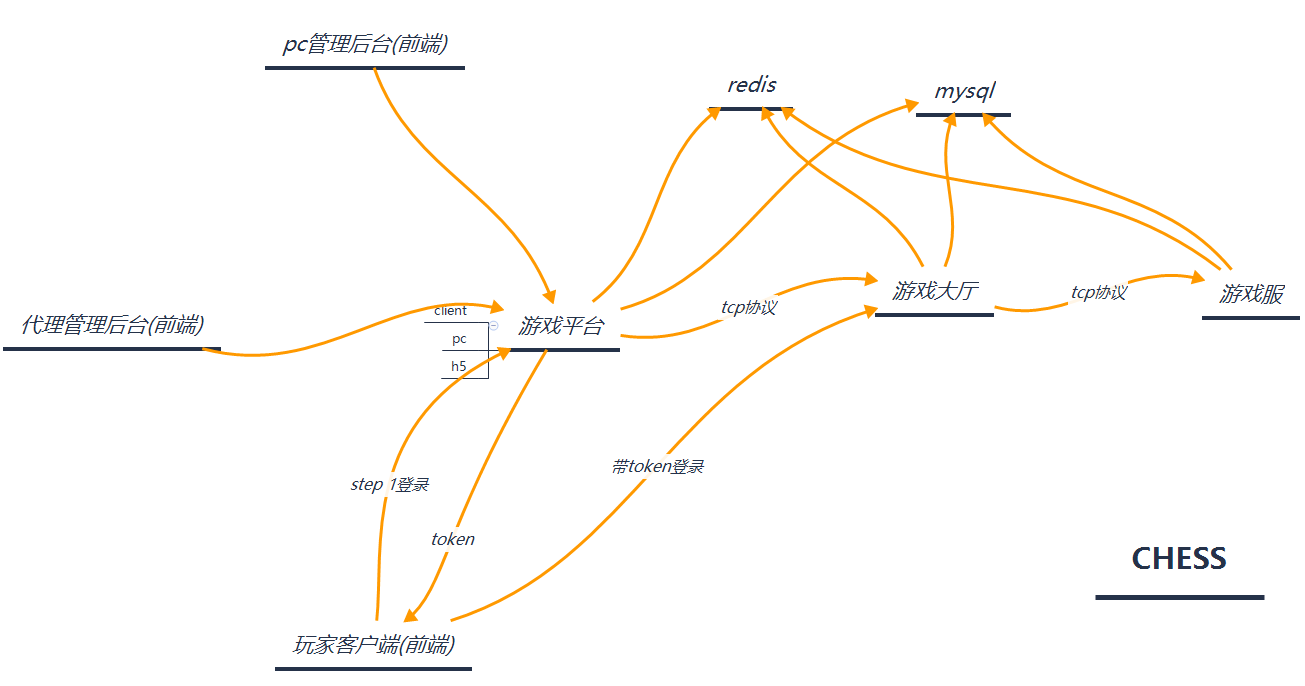
3 维护游戏服务器列表实时状态，不可进入的服务器显示维护中

4 维护玩家热切换服务器

5 各种实时状态信息维护

### **3：游戏服**

实现具体的游戏逻辑，启动后连接游戏大厅，注册本服务器提供的游戏信息



大概流程结构

## 2 热切换的主要流程

### 1一级热更

游戏服务器Lua逻辑或活动业务，参数，玩法等不设计c++代码的，直接reload脚本

### 2 单款游戏逻辑热切

启动预发布游戏逻辑服，后台设置生效(**可设计白名单对部分玩家生效**),启动服务器内部更新流程，非在玩玩家收到弹框提示，点击确定，切换新游戏服。非本游戏玩家或者在玩玩家不提示。在玩玩家大局结束后提示，走同样流程。

### 3 大组更新

涉及到大厅级别的热切换，发布预发布大厅服及相关游戏服，非游戏玩家弹框提示后切换新游戏大厅服。在玩玩家结束后踢出并提示。

### 4 平台服更新

无状态服务器直接重启

### 5 客户端更新

有玩法变更的，服务器先更新，客户端后更新。客户端的创房面板，由服务器配置文件控制(登陆刷新json配置)，客户端根据json配置动态生成创房UI面板。具体选项的意义，由游戏逻辑赋值json字段给游戏服务器逻辑处理

## 3 热更依赖基础设施

1：tcp协议兼容，走protobuf协议，维护及沟通成本会降低很多，新人介入门槛低

2：服务器尽量能嵌入lua开发部分逻辑，减少一些代理的配置玩法出尔反尔

3：http接口版本兼容，设计带版号的接口，或者访问带版本参数

4：游戏服务器设计一些自适应容灾设计，以免玩家出现意外逻辑卡住，极端情况提供后台GM接口处理，比如强制解散房间，踢人等等。

# 项目开发的一些个人爬坑建议

1：c++ 尽量设计框架跟服务，周边业务由golang,java,实现。经常变更的逻辑尽量嵌入lua脚本。这样才能应对商务跟产品经理各种突发情况，技术只有实现了实时热更才能应对。小公司产品好说话，大公司基本没有妥协的余地。

2：c++ 框架降低环境依赖，做到all in one，一处编译，随处运行。复杂的部署环境，可能出现不可预知的问题。极端情况也碰到过，实在不行docker部署。

3：涉及web开发的尽量前后端分离，专业的人做专业的事情比找一个全栈工程师更容易，效率，质量都远超全栈工程师。

4：涉及热更的业务尽量由服务器一端决定，客户端做成动态逻辑。比如创房面板UI设计成根据服务器json配置自动生成。消费，活动，商场，商务，周边信息等都由服务器配置，设计成动态，由商务或产品去更新配置。

5：程序员能不干的活尽量不干，只提供工具或者后台给策划跟产品，一定要把锅甩出去，身上背的锅太多了，会变成陀螺。

6：休闲类游戏要求不高，尽量用简单的技术框架。个人维护的项目中，有一个轻量级的c++框架：<https://gitee.com/hero1s/game_server.git>可以参考一下。

Golang 框架推荐简单易用的beego，

java 推荐 spring,

nodejs 推荐 koa2 用typescript 开发效率也是挺高的。主要是全栈的同学喜欢

其它的相对不推荐。

7：多跟同行交流，自己爬坑是代价最高的成长方式

以上都是个人根据之前做过的一个棋牌项目的爬坑的经验以及跟同行交流的心得。个人主要项目经历：类梦幻西游，类剑网3全3d武侠端游，刀塔传奇页游版，斗战神页游，神仙道山寨版，实时对战的塔防游戏，虫虫特工，棋牌等。

有兴趣或者疑问的可以私下交流

----- Toney