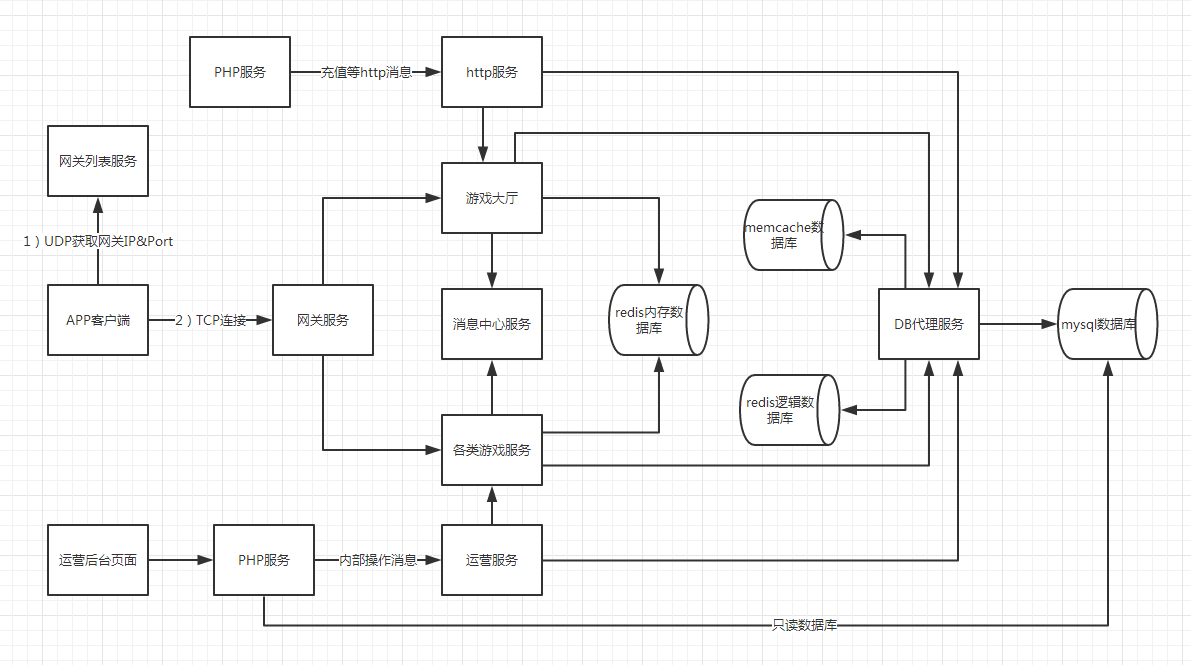
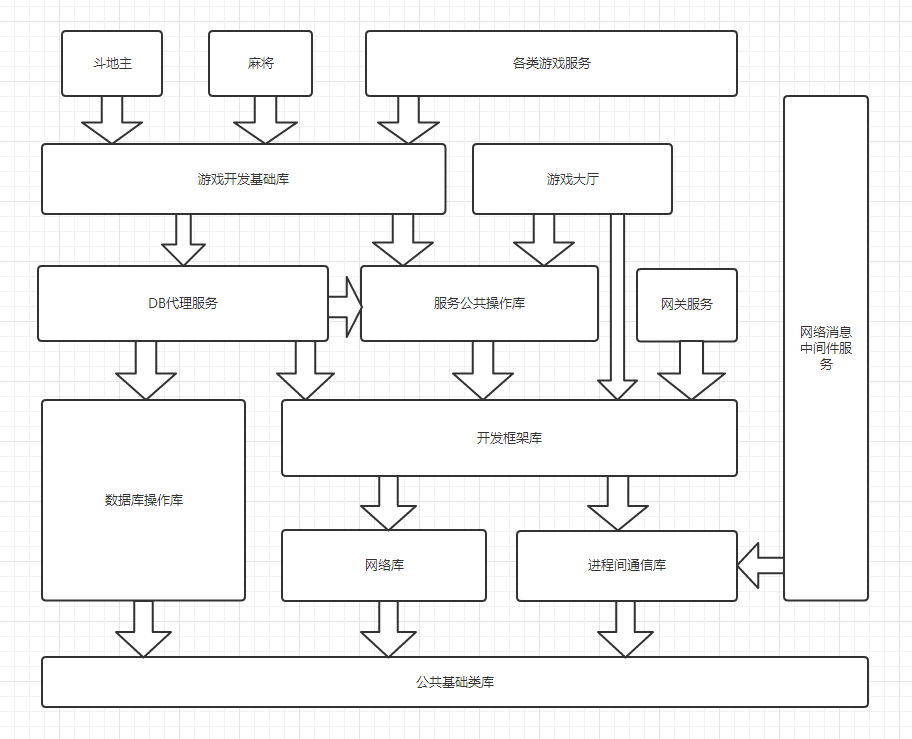
**游戏架构说明**

**个人开源项目：https://github.com/limingfan2016/game\_service\_system**

**架构图：**



**框架图：**



**APP客户端：**

1、APP客户端从网关列表服务获取网关服务的IP和端口号，和网关服务建立TCP连接；

2、通过TCP连接向游戏大厅、子游戏服务收发消息；

3、除了网关服务暴露给客户端之外，其他所有的服务对外封闭；

**网关列表服务：**

1、可部署多个实例，客户端可通过域名发送UDP消息获取网关服务信息；

2、定时主动从redis内存数据库获取所有最新的在线有效的网关服务信息（IP、端口、在线人数）；

3、接收客户端的UDP消息，根据策略（比如：选择在线人数最少的或者同城的网关服务）返回网关服务信息（IP、端口号）；

**网关服务：**

1、网关只负责和客户端建立连接，只转发消息；

2、定时把IP、端口号、在线人数、更新时间戳等信息更新到redis内存数据库；

**游戏大厅：**无数据状态，实现游戏大厅的各种功能（如玩家登录验证、活动、消息、背包、聊天、邮件等等）；

**各类游戏服务：**各种子游戏服务，如牛牛、斗地主、炸金花、捕鱼等；

**Http服务：**负责和第三方平台的对接，收发http消息，如充值、登录等等；

**消息中心服务：**负责内部消息的路由转发，如游戏服务到大厅服务；

**运营服务：**接收管理运营后台对游戏的管理信息，实现对游戏和玩家的操作控制；

**DB代理服务：**所有消息对db的读写都统一通过db代理服务；

**数据库：**  
1、mysql数据库可分库分表存储玩家数据，以memcached做为mysql的内存缓存DB，大大提高了数据库的读写能力；

2、mysql以主备机方式部署，做为数据备份和故障切换；

3、redis数据库存储必要的少量交互数据，方便服务间交互；

**安全：**

1、整套架构对外只暴露了网关，当受到外部攻击时，只需要变更网关部署即可，其他所有的服务无需任何改变；

2、数据库内部隐藏，所有对数据库的操作，均由DB代理服务验证执行，外部无权限；

**说明：**

1、主体架构：N网关+N服务+N数据库代理+内存DB（Redis、MemCached）+Mysql数据库，基于该架构可建立集群，高效率处理大规模、高并发消息；

2、所有的服务均可部署多个实例，任意实例可分布式部署在同节点或者跨节点上，所有服务实例可任意扩容和收缩；

3、各服务在同节点间的通信采用共享内存，以达到同节点服务间通信的最大性能；

4、每个网络节点部署一个消息中间件服务，跨节点服务通信通过消息中间件，服务间通信透明便捷；

5、已实现丰富的各种类库可直接使用，包括基础库（配置文件读写、日志、多线程、多进程、锁、对象引用计数、内存池、免锁消息队列、免锁数据缓冲区、进程信号、共享内存、定时器等等基础功能组件），网络库（socket、TCP、UDP、epoll机制、连接自动收发消息等），数据库操作库（mysql，redis、memcache API 封装可直接调用），开发框架库（消息调度处理、自动连接管理、服务开发、游戏框架、服务间消息收发、消息通信等等），消息中间件服务（不同网络节点间自动传递收发消息）等多个功能组件、服务；

6、开发框架采用多线程（分别有定时器线程、网络线程、业务线程）模式，网络非阻塞异步模型，消息异步处理方式，极大的提高消息处理性能；

7、开发框架细节已封装在底层，业务开发人员只需专注业务逻辑实现，一定程度上提高了开发效率及降低了开发人员的技能要求；

**缺点：**

1、纯异步开发方式，对开发人员有一定的技术门槛要求；

2、目前只支持C++语言开发业务逻辑，开发效率相对不高；

**改进：**

1、游戏业务逻辑支持使用lua、python、go等语言开发；

2、平台不变，使用C++开发，保持平台的运行高性能，平台稳定之后变更较小；

3、支持其他语言开发业务逻辑，开发效率大大提高并且对开发人员的要求门槛降低，弥补了C++开发效率低要求门槛高的问题，如此整套体系就比较完善了；