说明

pc页面使用前端框架AnjularJs和HTML静态模板做页面展示逻辑，js和HTML等文件均通过CDN进行访问加速，不直接从服务器获取，

pc和app均使用HTTPS连接网关层获取json数据，网关使用聚合的方式分别调用各个微服务处理业务逻辑，自身不做实际业务逻辑的处理。

微服务负责具体的垂直的业务线，各个微服务之间不相互调用，底层数据保持一定程度的独立性，如有必须可做一定程度的数据冗余，冗余标准按垂直业务相关为准

微服务均可使用公共组件或中间件，目前没有相关限制

**第一层：网关层**

* 含义和作用：流量控制、降级|容错、API监控、微服务聚合
* 应用情况：不再对外输出HTML格式的模板，使用HTTPS接收请求和对外输出json数据，使用dubbo作为PRC框架调用和聚合微服务接口，目前网关层按业务线分为成了8个

**第二层：微服务**

* 含义和作用：每个微服务均负责某一块垂直的业务，相关业务只会在对应的这一个微服务内部处理，不存在多个微服务处理同一个业务的情况
* 目前按垂直业务划分为9个微服务

**MQ异步任务**

* 含义和作用：

1.为了提供更快的前端服务响应，将部分业务转移为前端响应完用户请求后系统异步执行；

2.事件驱动：某个业务触发一个系统事件，其他业务线通过监听此事件处理自己的相关后续业务

3.数据同步：如需将一些数据冗余到自己的微服务内部，通过异步任务来实现冗余可大大减轻前端业务响应压力和业务逻辑复杂度

**公共组件和中间件**

* 含义和作用：一些独立的程序功能，为所有微服务或系统任务提供功能上的支持

**类的分层命名规范**

* handler、manager、business、service

在web-api层聚合service的类叫manager

只有RPC的服务类才能叫service

微服务提供给web-api代理service接口的类叫handler

PRC内部逻辑处理类使用business命名