

已有的服务器引擎功能:

1. center->服务发现
2. dbproxy->数据库代理
3. hub->逻辑节点,提供与任意其他 hub 节点通信的能力,对于需要即时响应的逻辑提供与客户端直连的功能
4. gate->网关层,提供与客户端连接的功能,支持 tcp\enet\websocket

飞行棋的服务器架构:

在 abelkhan 的基础上实现了飞行棋游戏的逻辑

1. login->登录服,同时包含为玩家分配 player 服的功能
2. player->玩家数据服,提供玩家数据维护和存储的功能,游戏功能访问的发起点开房间\匹配等
3. match->匹配服,提供为玩家匹配游戏的功能
4. room->房间服,提供开房间功能,支持战场外聊天
5. game->游戏服,游戏主逻辑服

飞行棋开发过程中暴露的问题:

1. 玩家数据落地需要显式的调用与 dbproxy 的 rpc,带来开发上的不便
2. 对玩家的消息响应提供连接 id(current\_client\_uuid)需要程序显式的通过连接 id 获取玩家对象
3. 不同服之间的数据传递,需要自定义 rpc 调用

一些想法,在 abelkhan 的基础封装一套 ecs 架构的数据管理层,提供:

1. 玩家消息响应函数中,直接获取玩家实例
2. 托管数据落地,程序只需要按规范实现 to\_db\_object 接口
3. 不同服务器之间数据传递,程序只需要 await 调用 get/load 接口