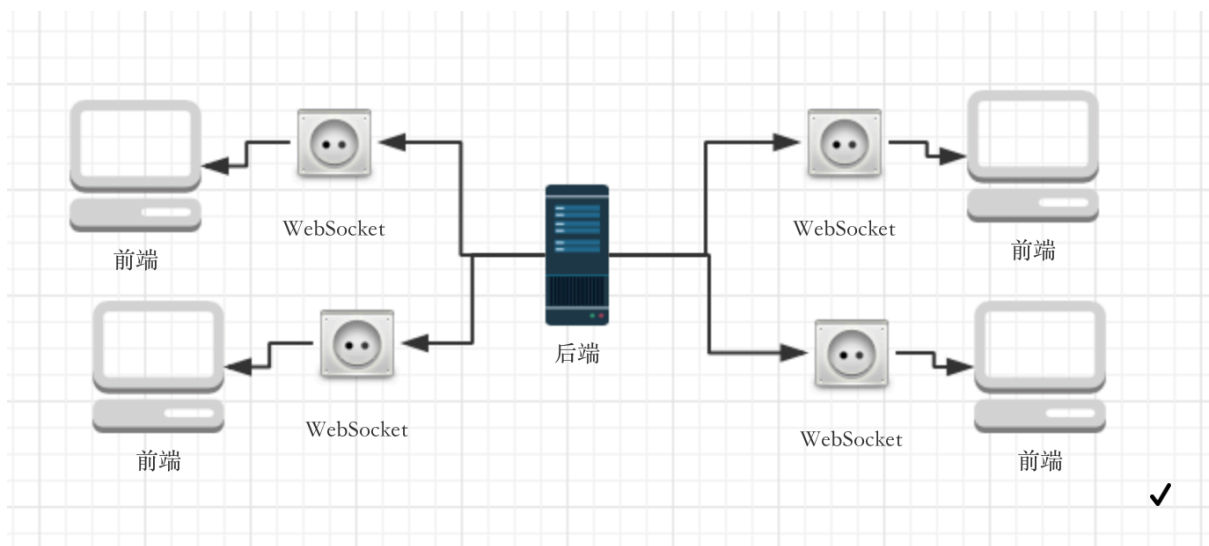


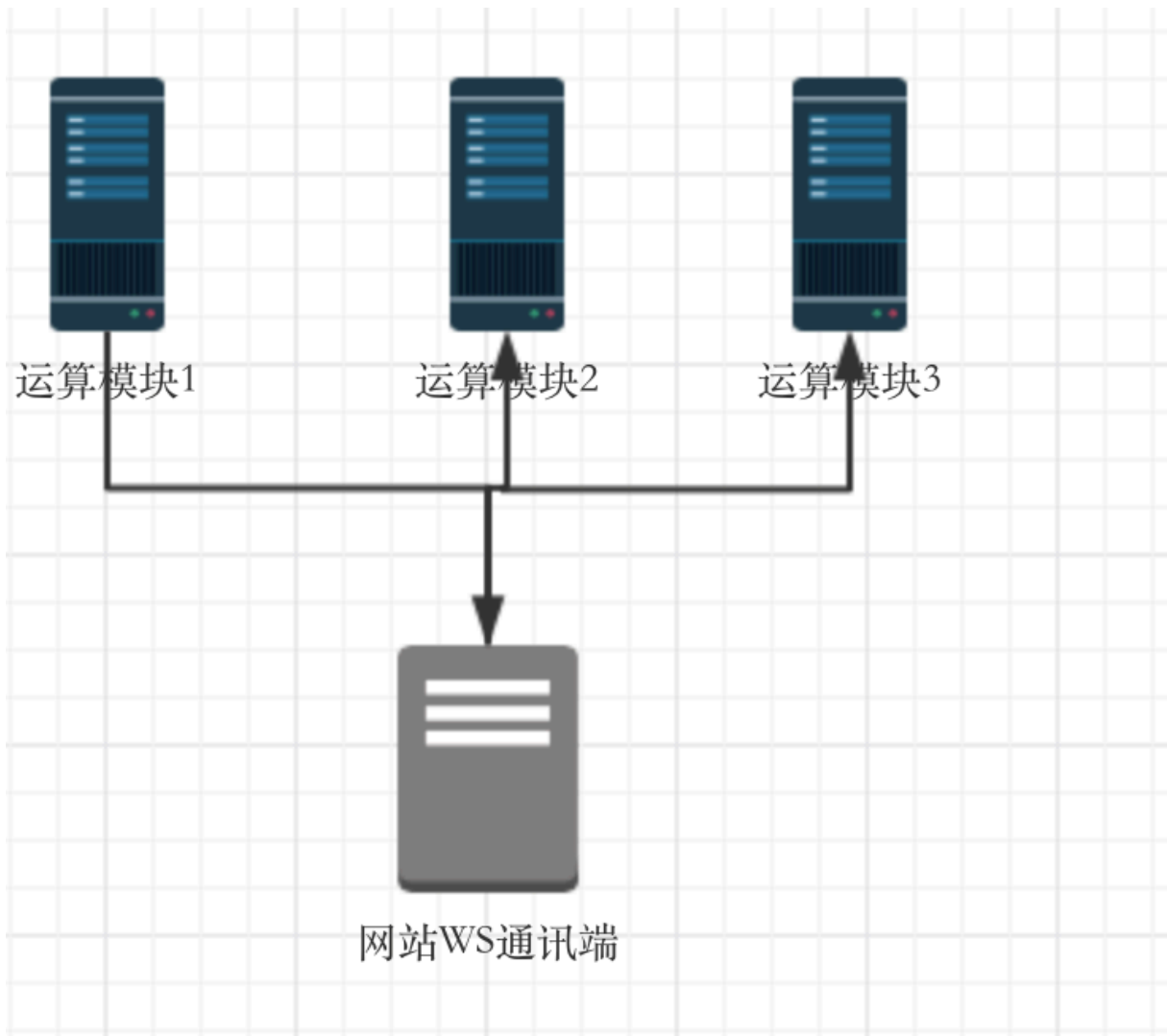
架构说明书



为了确保游戏的实时通讯，游戏必须保证使用websocket。此处比较方便的有socket.io（或者其他亦可），socket.io是Node.js专用。

但是这张图中有个问题，后端集成度较高，比如我们选择socket.io，基本意味着我们需要用Node作为我们的后端。

此时我们把Server拆开，内部细分。



服务端（简称site）收到由websocket接收到的信息，进行消息转发（此处可能是另一个新的websocket服务器或者http服务器，也可以是一个独立模块或者语言），计算模块按照规定的接口实现传送数据包，进行与site的交互操作，各单元可以独立运行，可以是不同语言，只要采用统一通信协议，统一数据包即可。site开发必须保证site的时序正确（基本上我们的流程控制死的还算可以，应该不会有太大问题）。

比如如图，我们可以假定运算模块1用于抽卡逻辑，需要存储每个人的卡堆和手牌等信息便于计算，提供相应接口（虽然不知道会不会用到）……。

运算模块2用于连锁时的时序，情况判断与计分。

运算模块3用于……（此处还没想好）

这是我上次自己定义的数据包格式，可以参考：https://github.com/csvwolf/Sky_Message_Routing_Middleware

为了方便开发，建议使用Git作为版本控制，各单元最好能进行独立的单元测试，考虑到时间紧迫，能跑起来万岁！！

分工：

背景设计（用于口胡）：1人

前端：1-2人

中间件：1人

各运算模块：1个一人