区块链网络架构：

一共三个组织：交易平台; 游戏公司; 第三方公正处（或监督处）

说明：**实际上区块链只部署了两个组织**

区块链网络接口：

游戏资产部分：

GameAsset{

AssetID：资产主键，保证唯一

Type：资产类型，可以是帐号，道具，或者之后是其他分类

Number：该类游戏资产的数量

GameCompany：该游戏资产的发行公司

GameName：资产所属的游戏名称

ReleaseTime：游戏资产首次发行时间

Owner：该资产的所有者（首次发行默认为游戏公司，有人认购再更改）

AssetInfo：对于游戏资产的描述，需要对游戏资产有精确的定义

TransactionInfo：用于记录每次交易的备注，相当于以太交易备注，按需写入信息，

增加灵活性

}

1.生成游戏资产

接口：generateAssets（json：GameAsset）{

return true or false

}

说明：发行公司，发行时间，所有者应该在资产初始化的时候应该由系统自动生成（交给SDK和 前端协商了）

2.游戏资产属性查询（查所有者，查描述啥的, SDK会处理成前端需要的信息）

接口：getGameAssetInfo（AssetID）{

return GameAsset

}

3.游戏资产所有权的更改

接口：changeGameAssetOwner(AssetID, newOwner, TransactionInfo){

return true or false

}

4.删除游戏资产（从数据库完备的角度应该增加删除操作）

接口：deleteGameAsset(AssetID){

return true or false

}

测试数据："{\"AssetID\":\"001\",\"Type\":\"money\",\"Number\":16,\"GameCompany\":\"Tencent\",\"GameName\":\"chiji\",\"ReleaseTime\":\"2018-04-29T05:05:13.359Z\",\"Owner\":\"Liyiming\",\"AssetInfo\":\"nothing\",\"TransactionInfo\":\"init\"}"

用户部分（用于对用户的信息索引，方便查询）：

User{

ID:主键，不能重复

Name：账户名称

Email：用户邮箱

Balance：账户余额

AssetLIst：用户拥有的资产列表

AssetForSale：用户待出售的资产

TransactionInfo：用户的交易信息

}

1.生成游戏资产

接口：generateUser（json：User）{

return true or false

}

2.用户信息查询

接口：getUserInfo（UserID）{

return User

}

3.用户信息修改

接口：changeUserName（UserID，newName）{

return true or false

}

接口：changeUserEmail（UserID，newEmail）{

return true or false

}

接口：changeUserBalance（UserID，newBalance）{

return true or false

}

接口：changeUserAssetList（UserID，newAssetList）{

return true or false

}

接口：changeUserAssetForSale（UserID，newAssetForSale）{

return true or false

}

接口：changeUserTransactionInfo（UserID，newTransactionInfo）{

return true or false

}