

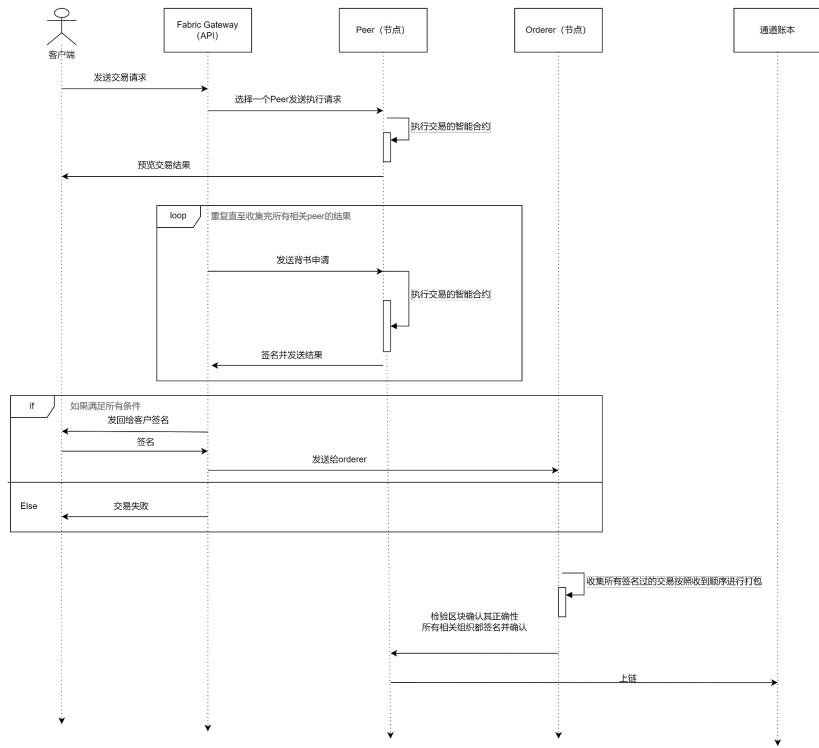
HyperLedger Fabric API

A dark blue diagonal gradient bar that starts from the bottom left and extends towards the top right, covering the lower half of the slide.

Fabric Gateway API

- 语言: 支持Go, Node, 或者Java
- 用于客户端于Fabric区块链进行交易方面的交互

交易流程

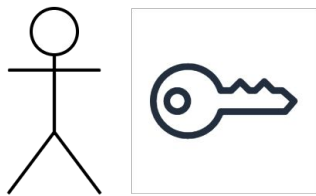


Peer节点: 拥有整个账本的记录, 负责执行合约、验证交易和区块
Orderer节点: 负责将所有收到的交易按顺序打包

执行请求: 选一个peer节点查询当前账本内容 (World State) 并返回基础结果
背书请求: 得到足够多的背书结果
背书结果: 通过连接一个Peer节点并在当前节点上执行智能合约得到
提交交易: 将交易发送给orderer打包

如何使用Gateway连接网络，调用链码，提交交易

前提：



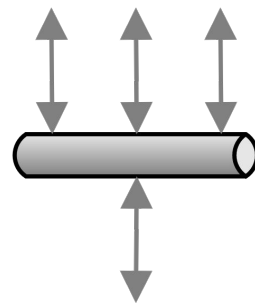
合法账户 及其公钥

1. 需要被当前通道所信任的账户



节点 (Peer)

2. 指定的ip address: port 和其TLS证书



通道，链码名称，合约方法名称等

1. 配置依赖

Maven:

```
<dependency>  
  <groupId>org.hyperledger.fabric</groupId>  
  <artifactId>fabric-gateway</artifactId>  
  <version>1.1.0</version>  
</dependency>
```

Gradle:

```
implementation 'org.hyperledger.fabric:fabric-gateway:1.1.0'
```

2. 构建Gateway, 实现客户端与Fabric网络连接方式的设置

1. 创建gRPC channel

- a. Fabric 网络实体之间通信使用的gRPC
- b. 在此API里, 创建Gateway对象必须提供一个gRPC channel 对象
- c. 一个gRPC channel可以用力啊创建多个Gateway对象

2. Gateway配置+设置

- a. Gateway支持我们以一个特定省份与Fabric网络进行通信

```
ManagedChannel grpcChannel = ManagedChannelBuilder.forAddress("gateway.example.org", 1337)
    .usePlaintext()
    .build();
```

1. 创建时需指定目标节点的IP、端口号、和TLS证书

```
Gateway.Builder builder = Gateway.newInstance()
    .identity(identity)
    .signer(signer)
    .connection(grpcChannel);
```

2. 设置identity是向Gateway中导入账户的身份证书

设置signer是向Gateway中导入账户与证书相匹配的私钥

然后把之前创建的gRPC channel导入

3. 调用链码&提交交易

3. 获取网络和链码

- a. 需指明需要连接到的通道名称, 链码名称, 参数才能调用

4. 执行交易和提交交易

- a. 执行交易: 向peer查询当前账本内容(World State)
- b. 提交交易: 提交新的背书、共识、验证、上链

```
try (Gateway gateway = builder.connect()) {  
    Network network = gateway.getNetwork("channelName"); Network接口是一个通道中所有Fabric节点的集合  
    Contract contract = network.getContract("chaincodeName"); Contract接口指一个特定的智能合约  
  
    byte[] putResult = contract.submitTransaction("put", "time", LocalDateTime.now().toString());  
    System.out.println(new String(putResult, StandardCharsets.UTF_8)); 提交交易  
  
    byte[] getResult = contract.evaluateTransaction("get", "time");  
    System.out.println(new String(getResult, StandardCharsets.UTF_8)); 执行交易  
} finally {  
    grpcChannel.shutdownNow().awaitTermination(5, TimeUnit.SECONDS);  
}
```