

常见问题

背书

背书架构:

问题: 在一个网络中需要多少个 Peer 节点来对一笔交易进行背书?

回答: 为一笔交易进行背书所需 Peer 节点的数量取决于链码定义中所指定的背书策略。

问题: 一个应用程序客户端需要连接所有的 Peer 节点吗?

回答: 客户端只需要连接链码的背书策略所需数量的 Peer 节点即可。

安全和访问控制

问题: 我如何来确保数据隐私?

回答: 对于数据隐私，这里有很多个方面。首先，你可以使用通道来隔离你的网络，每个通道代表网络成员的一个子集，这些成员查看部署到该通道上的链码的相关数据。

其次，你可以使用 [私有数据](#) 来保护组织的数据隐私。一个私有数据集合规定了通道中的一部分组织可以背书、提交或者查询私有数据，而不需要额外创建一个通道。在这个通道上其他的参与者只会收到数据的哈希值。更多信息可以参考 [在 Fabric 中使用私有数据](#) 教程。注意，在本文档的“关键概念”章节也解释了 [什么时候应该使用私有数据而不是通道](#)。

第三，作为 Fabric 使用私有数据来对于数据进行哈希的替代方案，客户端应用可以在调用链码之前将数据进行哈希或者加密。如果你将数据进行哈希运算的话，那么你需要提供一个方式来共享原始的数据。如果你将信息进行加密的话，那么你需要提供一个方式来共享解密密钥。

第四，你可以通过在链码逻辑中增加访问控制的方式，将数据的访问权限限制在你的组织中的某些角色上。

第五，账本数据还可以通过在 Peer 节点上的文件系统加密的方式来进行加密，在数据在交换的过程中是通过 TLS 来进行加密的。

问题: 排序节点能够看到交易数据吗?

回答: 不，排序节点仅仅对交易进行排序，他们不会打开交易。如果你不想让数据经过排序节点，那么应该使用 Fabric 的私有数据功能。或者，你可以在调用链码之前在客户端应用中对数据进行哈希运算或加密。如果你对数据进行加密的话，那么你需要提供一种方式来共享解密的密钥。

应用程序端编程模型

问题: 应用程序客户端如何知道一笔交易的结果?

回答: 模拟交易的结果会在提案响应中通过背书节点返回给客户端。如果有多个背书节点，那么客户端会检查所有的反馈是否一样，并且提交结果和背书来进行交易的排序和提交。最终提交节点将会验证这笔交易是否有效，并且客户端会通过 SDK 获取交易的最终结果。

问题: 我应该如何查询账本的数据?

回答: 在链码中，你可以基于键来查询。键可以按照范围来查询，复合键还可以通过多参数的组合进行查询。比如可以用一个复合键（owner, asset_id）查询所有某个实体拥有的资产。这些基于键的查询可以用来对于账本进行只读查询，也可以在更新账本的交易中使用。

如果你将资产数据在链码中定义为 JSON 格式并且使用 CouchDB 作为状态数据库的话，你也可以在链码中使用 CouchDB JSON 查询语句来对链码的数据进行富查询。应用程序客户端可以进行只读查询，这些反馈通常不会作为交易的一部分被提交到排序服务。

问题: 我应该如何查询历史数据来了解数据的来源？

回答: 链码 API `GetHistoryForKey()` 能够返回一个键对应的历史。

问题: 如何保证查询的结果是正确的，尤其是当被查询的 Peer 节点可能正在恢复并且在获取缺失的区块？

回答: 客户端可以查询多个 Peer 节点，比较他们的区块高度、查询结果，选择具有更高的区块高度的节点。

链码（智能合约和数字资产）

问题: Hyperledger Fabric 支持智能合约吗？

回答: 是的。我们将这个功能称为链码。它是我们对于智能合约的实现，并且带有一些额外的功能。

链码是部署在网络上的程序代码，它会在共识过程中被链的验证者执行并验证。开发者可以使用链码来开发业务合约、资产定义以及共同管理的去中心化的应用。

问题: 我如何创建一个业务合约？

回答: 通常有两种方式开发业务合约：第一种方式是将单独的合约编码到独立的链码实例中。第二种方式，也可能是更有效率的一种方式，是使用链码来创建去中心化的应用，来管理一个或者多个类型的业务合约的生命周期，并且让用户在这些应用中实例化这些合约的实例。

问题: 我应该如何创建资产？

回答: 用户可以使用链码（对于业务规则）和成员服务（对于数字通证）来设计资产，以及管理这些资产的逻辑。

在大多区块链解决方案中由两种流行的方式来定义资产：无状态的 UTXO 模型，账户余额会被编码到过去的交易记录中；账户模型，账户的余额会被保存在账本的状态存储空间中。

每种方式都带有他们自己的好处及坏处。本区块链技术不主张任何一种方式。相反，我们的第一个需求就是确保两种方式都能够被轻松实现。

问题: 支持哪些语言的链码开发？

回答: 链码能够使用任何的编程语言来编写并且在容器中执行。当前，支持 Golang、node.js 和 java 链码。

也可以使用 **Hyperledger Composer** 来构建 Hyperledger Fabric 应用。

问题: Hyperledger 有原生的货币吗？

回答: 没有。然而，如果你的网络真的需要一个原生的货币的话，你可以通过链码来开发你自己的原生货币。对于原生货币的一个常用属性就是交易会引入余额的变动。

近期发布版本中的不同

问题: 我在哪里能够看到在不同的发布版本中都有哪些变动?

回答: 发布版本中的变动记录在 [版本发布](#) 中。

问题: 如果还要其他问题的话, 我在哪里可以获得技术上的帮助?

回答: 请使用 [StackOverflow](#) 。

排序服务

问题: 我有一个正在运行的排序服务, 如果我想要转换共识算法, 我该如何做?

回答: 这个是不支持的。

问题: 什么是排序节点系统通道?

回答: 排序节点系统通道 (有时被称为排序服务系统通道) 是排序节点初始化时被启动的通道。它被用来编排通道的创建。排序节点系统通道定义了联盟以及新通道的初始配置信息。在通道被创建的时候, 在联盟中定义的组织、`/Channel` 组中的值 and 策略以及 `/Channel/Orderer` 组中的值 and 策略, 会被合并到一起形成一个新的初始的通道定义。

问题: 如果我更新了我的应用程序通道, 我是否需要更新我的排序系统通道?

回答: 一旦一个应用程序通道被创建, 它的管理独立于其他任何的通道 (包括排序节点系统通道)。基于所做的改动, 变动可能需要也可能不需要被放置到其他的通道。一般来说, MSP 的变动应该被同步到所有的通道, 而策略的变动一般是针对一个特定通道的。

问题: 我可以有一个既作为一个排序节点又作为应用程序角色的组织吗?

回答: 尽管这是可能的, 但是我们强烈不建议这样配置。默认的 `/Channel/Orderer/BlockValidation` 策略允许任何具有有效的证书的排序组织来为区块签名。如果一个组织既是排序节点又是应用程序的话, 那么这个策略就应该被更新为只有被授权来排序的证书的子集才可以为区块签名。

问题: 我想要实现一个针对于 Fabric 的共识, 我应该如何开始?

回答: 一个共识的插件需要实现在 `consensus`包中定义 `Consenter` 和 `Chain` 接口。有一个基于 `raft` 的插件。你可以学习更多的内容帮助你实现。排序服务的代码可以在 `orderer`包中找到。

问题: 我想要改变我的排序服务配置, 比如批处理的超时时间, 当我启动了网络之后, 我该如何做?

回答: 这属于网络的配置。请参考 [configtxlator](#) 。

BFT

问题: 什么时候会有 BFT 版本的排序服务?

回答: 目前还没有具体的时间。 We are working towards a release during the 2.x cycle, i.e. it will come with a minor version upgrade in Fabric.

