FISCO BCOS和Hyperledger Fabric的异同

FISCO BCOS和Hyperledger Fabric是联盟链最主流的两种技术，这两种技术我都深入地研究过原理并且大量实操过，一直想写一篇文章总结一下两者的异同，有些侧重于理论，有些侧重于使用过程中的实操，方便大家在技术选型的时候可以更加有依据。

写在前面：在讨论联盟链的时候，必然会涉及到组织或机构的概念，两者其实是一回事，比如，政府的税务局、公安局就是两个独立的组织，公司也是组织，大集团公司的子公司或内部的某个部门也可以定义为一个组织，这主要基于业务需求来确定。文章的剩余部分我就统一用“组织”这个词了。

另外，对两者的异同不会过于深入地讨论，否则发散出去一篇文章是完全不够的。

**相同点**

* 两者都是联盟链技术，都基于公钥基础设置（PKI），都有CA，都有节点准入控制，节点准入都需要证书，并非任何人都可以随意地加入一条联盟链，在组建联盟链的时候，往往各组织之间已经有了一定的信任基础，在线下的身份是彼此互相了解的，同时还会签订法律合同约定彼此在链上的行为，如果发现某组织在链上作恶，依法办事。联盟链的信任模型和公链是有本质不同的，公链是完全匿名的，彼此之间并不知道对方的链下身份，所以公链的参与各方是没有任何信任基础的，必须要用更加公平、可靠的共识算法来保证各方都有机会打包，且不会作恶。
* 都支持组织分组和网络隔离。Fabric中有channel的概念，FISCO中有group的概念，二者都是用于组织分组和网络隔离。

**不同点**

* 共识算法不同

FISCO目前支持RAFT和PBFT两种共识算法，Fabric目前支持KAFKA和RAFT两种共识算法，并不支持PBFT。Fabric很早的版本是支持过PBFT的（v0.6），后来因为某种原因不支持PBFT了。所以FISCO是支持崩溃容错和拜占庭容错的，而Fabric只支持崩溃容错。从这一点看，FISCO对共识算法的支持更为丰富。但通常来讲，联盟链各组织之间已经有一定信任基础，链下会通过合同和法律约束链上的行为，通常来讲，RAFT共识算法对联盟链来讲是足够的。

* 容器化支持

Fabric官方从一开始就是通过docker compose部署的，我后来在实操的过程中也通过kubernetes成功地部署过Fabric，Fabric对容器的支持比较完善，也有成熟的部署方案。Fabric所有的组件全都是容器化执行，包括peer, orderer,数据库couchdb，甚至链码本身就是一个新创建的容器，和peer容器是解耦的。

而FISCO从一开始主要是通过二进制方式部署，后来也支持docker部署，但稍显单薄。

容器化的优势就不多说了，大家都知道。

从容器化支持这个角度讲，Fabric支持得更好。

* 国密算法

FISCO对国密的支持是很成熟的。但是Fabric官方目前还并不止支持国密，所以在某些行业（比如政务）或者业务场景用Fabric存在一些合规风险，所以有些用Fabric的公司为了支持国密，一般都需要定制化开发支持国密，但是这并不是一个很好的方式，质量得不到很好的保证，也和Fabric主分支形成了分叉，以后代码合并也是一个大问题。不过最近一段时间以来，超级账本TWGC国密工作组一直在努力进行Fabric的国密开发，目标是把国密的支持加入Fabric的主分支，以后Fabric官方就支持国密了，但目前这项工作还在进行当中，还没实现。

从国密算法的角度看，FISCO是处于优势地位。

* 证书结构&组织结构

FISCO是三级证书结构：链证书、机构证书、节点证书。意味着有一个超级管理员的存在，即所有组织的根CA的存在，机构证书是由根证书签发的，这实际上有点违背区块链去中心化的应有之意，即在联盟链各机构上面还有一个联盟链委员会且有管理员，管理根CA，这是一种集中化的思想。

Fabric是两级证书结构：组织证书、节点证书。Fabric里没有链证书的概念。每个组织都有自己的根CA，互相没有隶属关系，组织与组织之间是平等、对等的关系，没有一个超级管理员来管理所有组织，这更加符合区块链去中心化的思想，个人认为真正的联盟链应该的证书结构和组织结构应该是Fabric这样的，这样各组织才能够没有顾虑地上链。

* 证书角色

FISCO的证书角色有：联盟链委员会管理员、机构、节点、SDK。但每个用户（即每个使用和调用联盟链的用户）是没有单独的证书的，只有私钥，也就是说在FISCO链上，一个用户只有地址的概念，没有证书，没有组织属性，这就意味着权限控制需要完全通过智能合约代码来控制。

Fabric的证书角色有：组织、节点（peer,orderer）、admin、client。admin和client即区块链的用户（admin是特殊的用户，拥有管理peer和orderer的权限），即用户不仅在链上有地址，还有证书，且可以通过智能合约的库获取该用户的MSPID即组织ID以及其他一些和本组织相关的信息，这样就可以在合约中非常方便地获知交易的发起者属于哪个组织，从而更加方便地进行权限控制。比如属于公安局的用户可以调用某些方法，属于税务局的用户又可以调用哪些方法，非常有利于权限控制，和线下的组织和业务逻辑做到了更好的映射。

个人认为Fabric的证书对于权限控制更加的方便。

* TLS证书

FISCO中TLS证书和identity证书是同一个证书。

Fabric中TLS证书是单独的证书，用于通信加密和安全。

* 打包执行的机制

FISCO和以太坊是以太坊是先打包、再执行，意味着执行一定是串行的。

Fabric是先预执行，再打包，再验证，再提交，意味着预执行阶段是并行的，这是为了性能更高。具体就不深入了，有兴趣的可以看Fabric的设计文档。

<https://arxiv.org/pdf/1801.10228.pdf>

* 排序节点

Fabric有单独的排序节点orderer，只用于共识、排序、打包，不存储区块数据和链码。

FISCO没有单独的排序节点，而是分为共识节点和观察者节点，共识节点既负责共识、排序、打包，也存储区块数据和智能合约。

* gas

FISCO由于大量借鉴以太坊，所以有gas的概念。

Fabric是和以太坊完全不同的设计，没有gas的概念。

* 数据库表

FISCO:一个群组内的一张KVTable表可以被多个合约共享使用。

Fabric:每个channel和chaincode是一张单独的表，不能共享。

* 群/环签名

FISCO支持，Fabric不支持。

* 子分组

Fabric支持在一个分组（channel）中把一些组织抽出来进行进一步分组（private data collection），打个比方，A,B,C,D四个组织在一个channel中，互相通信，共同维护账本，执行合约，但由于A和B之间需要单独执行某些业务交易，且涉及商业机密并不想让C和D知道，这时候就可以把A,B分为一个更小的分组，即private data collection，A,B之间共享的数据对C,D不可见，但数据hash是在C,D上有存证的，起到了存证和见证的效果。

FISCO不支持这一功能。

这就让Fabric更适用于一些更复杂的商业场景更加。

* 智能合约语言和开发

FISCO大量借鉴以太坊，且采用和以太坊相同的solidity语言开发智能合约，所以可以很方便地实现各种通证标准，比如ERC20, ERC721, ERC1155，且有工业级的成熟的第三方库比如openzeppelin可供使用，对于通证相关的开发，FISCO是比较方便的。但solidity除了在各种通证标准库方面有优势之外，在其他方面并不具有优势，因为solidity缺少大量现有的基础库，比如字符串的拼接和比较这种功能的库，从一开始是没有的，还需要自己造轮子，导致开发效率很低，总之，solidity的第三方库是很缺乏的，也没有形成一些标准，各种库在网上分布也比较零散，这一块是solidity开发的一个痛点。除此之外，solidity开发还有一些其他坑，比如stack too deep（栈太深）这个坑我已经遇到很多次了，另外就是solidity语言有很多的限制（当然这其中也有EVM和gas limit，交易大小，区块大小的限制），比如在最简单的对json的解析都是很难实现的。但solidity语言的单元测试，如果在没有用到Table.sol, KVTable.sol这种预编译合约的情况下，可以通过以太坊的openzeppelin-test environment, openzeppelin-test helper,waffle,mocha等来进行，比较方便。FISCO的生态对以太坊的生态有一定程度的相通。

Fabric则主要是采用go语言作为智能合约开发的语言，go语言是相对成熟的语言，不仅用在智能合约开发中，也用到了以太坊客户端Geth，k8s，docker的开发当中，所以，go语言是经过了大量的工业级产品检验的，且有大量成熟的三方库，很多基础功能完全不需要自己造轮子，且单元测试非常方便，所以，在Fabric中开发智能合约和开发普通的程序没有什么太明显的区别，这对程序员是很友好的。

另外一个小细节，Fabric和FISCO都是支持event的，这有利于智能合约和链下世界进行交互，比如预言机，具体不详述了。

* 组件设计的精细程度

Fabric的设计个人认为更加模块化，peer,orderer,CA server, CA client,chaincode等组件都是分得很清晰的，且还充分地考虑了高可用特性，比如Fabric CA server，既可以选择leveldb(如果没记错的话)，还可以选择mysql来做持久化。这种感觉不太容易说清楚，结合自己的经验，就感觉Fabric的各个组件做得很细，很扎实，也解耦，而FISCO要相对粗放一点。

* 官方文档语言

Fabric是全英文，且很多句子需要多读几遍才能透彻理解其精髓，对英语水平的确有一定要求，我认识的一些朋友都看不下去，我的经验是：耐着性子多读几遍，一定会有很多的收获，会被Fabric的文档深深地折服，Fabric的文档写得非常好，用词很精准（英语的单数复数都体现着不同的含义）。且Fabric的文档不仅仅说了Fabric怎么用，Fabric文档中有两章”Introduction”和”Key Concepts”是关于联盟链总体思想和哲学层面的，我个人认为做联盟链的人都应该看看这两章，看完之后对联盟链会有更深刻的理解，堪称联盟链最好的教科书。个人认为Fabric的文档写得非常好，其实网上很多人遇到的各种Fabric的问题都是因为文档没有读透，没有理解透，导致很多人觉得Fabric的上手门槛很高。

FISCO是中文，但也有英文版本，不过咱们中国人肯定是看中文，没有难度，容易看懂，注重实操，即怎么使用，但是在联盟链的思想和哲学层面和Fabric的文档相比还有很大的差距，算是实用主义吧，有利于快速上手，用起来。

* 技术生态

Fabric和FISCO都有比较完善的技术生态，比如都有各自对应的去中心化数字身份的组件（Indy和WeIdentity），区块链浏览器等等。

* 技术支持

Fabric由国外开发，技术支持不是那么给力，一般只能通过英文聊天室或者Stackoverflow进行提问，回复得也比较慢。

FISCO由于是国内开发的，加入官方的微信群，就可以问各种技术问题，通常都会有及时的回复。

就先写这么多吧。