**课程报告**

|  |  |
| --- | --- |
| **课程名称：** | **企业软件项目实训** |
| **学生姓名：** | **黄旸珉** |
| **学生学号：** | **201630664536** |
| **学生专业：** | **软件工程卓越班** |
| **开课学期：** | **2018-2019第二学期** |

**软件学院**

**2019年6月**

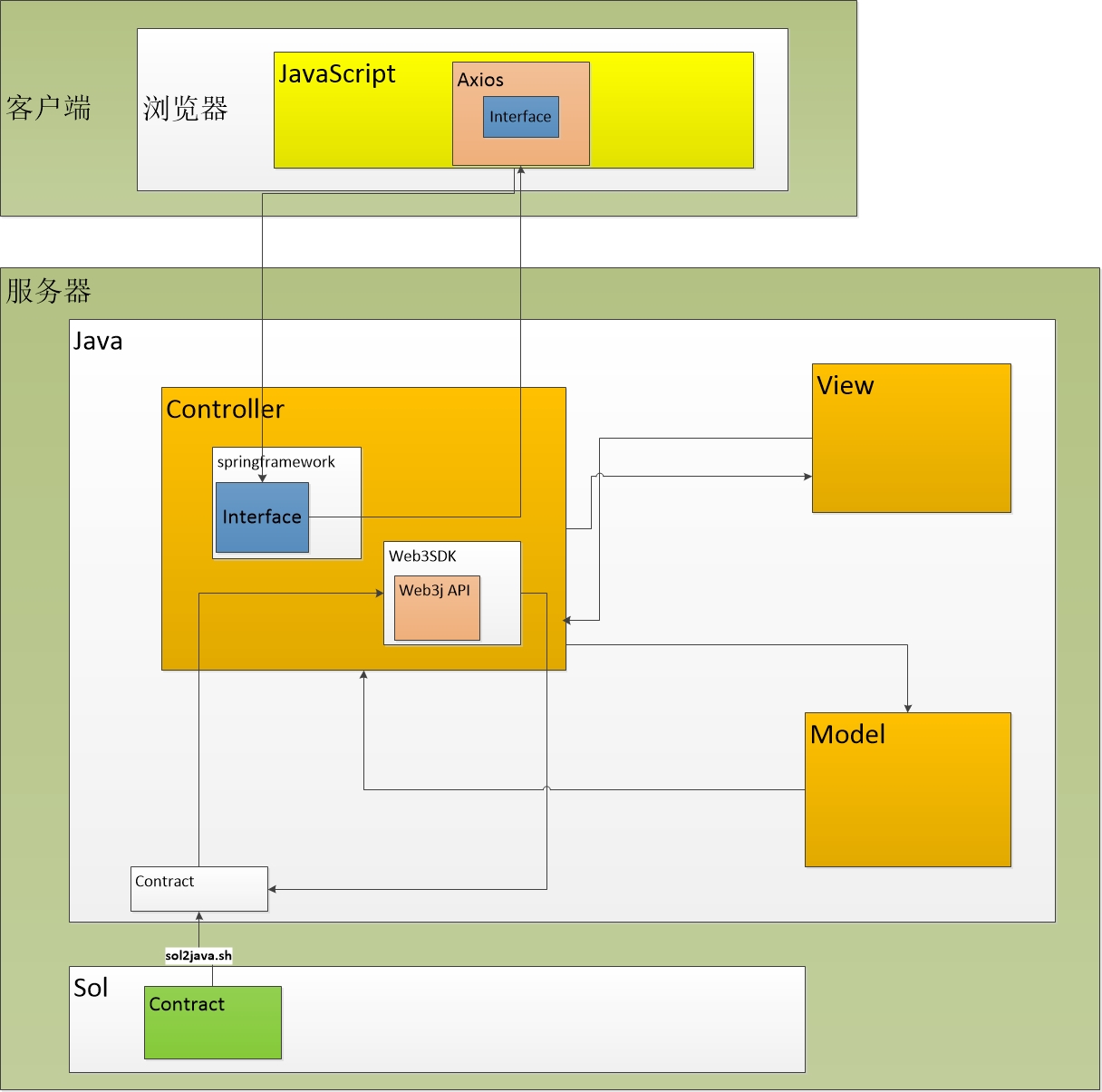
区块链实训工作总结及感想

经过了六个星期的企业实训，我们在微众银行的技术人员指导下成功地通过了区块链和联盟链的技术培训，不仅成功地将我们小组有关区块链在音乐方面应用的创意落地实现，更得到了微众银行各位老师的认可，更得到了证明我们学习成果和努力的区块链技术认证证书。回顾这六个星期的工作，我对区块链的概念认识比之前有了质的飞跃，更在实践的过程中学习到了不止于区块链技术的内容，比如团队合作的方法，精神，智能合约编写，以及与区块链衔接相关的Java开发等等。下面，我将以几个方面来阐述这六个星期中对于区块链技术的思考与学习内容。

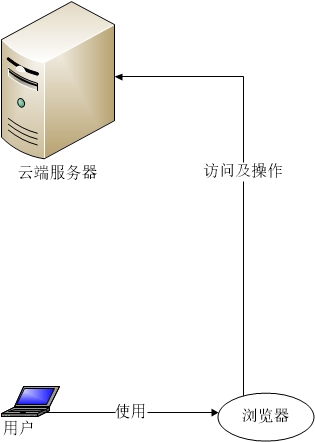
一、架构设计及区块链网络搭建

在六个星期的实训过程中，我主要负责了区块链网络在云端服务器上的搭建，其中包括了云端服务器部署节点，云端服务器搭建联盟链，云端服务器单群组多机构组网搭建等。同时，我也负责了对整体“音链”项目的架构设计。

后台采用Java Spring boot的MVC框架，前端采用Vue.js编写，智能合约使用Solidity语言编写。而部署的整体逻辑框架与物理框架如下图所示：



逻辑架构



物理架构

后台和区块链网络部署在云服务器端，通过访问前端，后台调用合约的结构进行部署。其中，我们使用到了Web3SDK以供后台访问节点，查询节点，并调用智能合约等。

而云服务器端的联盟链搭建以及单群组多机构的组网模式搭建过程如下：

1.安装依赖，及环境配置

1.1运行环境

运行环境如下：

阿里云服务器端Ubuntu 16.04

Java 8 （JDK-8u181）

1.2 openssl配置

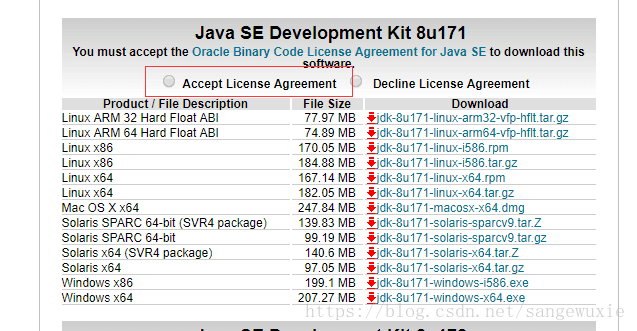
sudo apt install -y openssl curl

由于搭建联盟链的脚本文件Build\_chain.sh依赖于openssl和curl，使用该语句配置openssl。

1.3 Java安装

进入官网下载页面

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>



选择云服务器端所适合的JAVA SE版本，下载。

在/opt路径下用以下解压命令：

sudo tar -xvzf /mnt/share/tmp/jdk-8u181-linux-x64.tar.gz

在opt路径下使用ls命令查看解压是否成功：

2

在/bin目录下创建java软链接：

cd /bin

sudo ln –s /opt/jdk1.8.0\_181/bin/java java

随后设置Java环境：

vi ~/.bashrc

添加以下内容：

export JAVA\_HOME=/opt/jdk1.8.0\_181

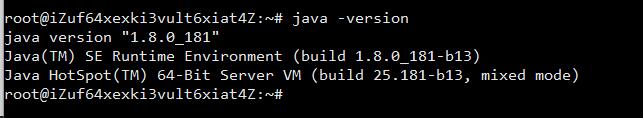
export JRE\_HOME=${JAVA\_HOME}/jre

export CLASSPATH=.:${JAVA\_HOME}/lib:${JRE\_HOME}/lib

export PATH=${JAVA\_HOME}/bin:$PATH

最后使用以下语句检验安装是否成功：

Java –version



1.4创建操作目录及下载脚本

使用以下命令创建fisco文件夹以进行后续搭建区块链工作：

cd ~ && mkdir -p fisco && cd fisco

使用以下命令下载脚本文件：

curl -LO https://github.com/FISCO-BCOS/FISCO-BCOS/releases/download/`curl -s https://api.github.com/repos/FISCO-BCOS/FISCO-BCOS/releases | grep "\"v2\." | sort -u | tail -n 1 | cut -d \" -f 4`/build\_chain.sh && chmod u+x build\_chain.sh

1.5配置控制台文件，以及控制台证书

使用如下指令获取及拷贝控制台文件：

bash <(curl -s

https://raw.githubusercontent.com/FISCO-BCOS/console/master/tools/download\_console.sh)

cp -n console/conf/applicationContext-sample.xml

console/conf/applicationContext.xml

使用如下指令配置控制台证书：

cp nodes/127.0.0.1/sdk/\* console/conf/

2.搭建4节点的单群组联盟链

在fisco目录下执行如下命令：

bash build\_chain.sh -l "127.0.0.1:4" -p 30300,20200,8545

其中，30300,20200,8545分别对应p2p\_port,channel\_port,jsonrpc\_port。由于链在本地运行，因此使用本机地址127.0.0.1，也可以使用云服务器端网卡局域网地址，在本ubuntu服务器端中，该地址为172.19.31.230。

运行成功后，返回如下信息：

Checking fisco-bcos binary...

Binary check passed.

==========================================================

Generating CA key...

==========================================================

Generating keys ...

Processing IP:127.0.0.1 Total:4 Agency:agency Groups:1

==========================================================

Generating configurations...

Processing IP:127.0.0.1 Total:4 Agency:agency Groups:1

==========================================================

[INFO] FISCO-BCOS Path : bin/fisco-bcos

[INFO] Start Port : 30300 20200 8545

[INFO] Server IP : 127.0.0.1:4

[INFO] State Type : storage

[INFO] RPC listen IP : 127.0.0.1

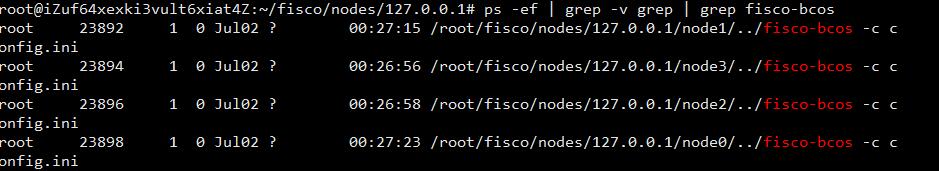
[INFO] Output Dir : /root/fisco/nodes

[INFO] CA Key Path : /root/fisco/nodes/cert/ca.key

执行以下指令启动四个节点：

bash nodes/127.0.0.1/start\_all.sh

成功后，执行以下指令检查进程启动情况：



3.机构群组的组网模式搭建过程

3.1下载安装部署工具

通过以下语句下载和安装：

cd ~/ && git clone https://github.com/FISCO-BCOS/generator.git

cd generator && bash ./scripts/install.sh

按照以下语句拉取最新fisco-bcos二进制文件到meta文件中：

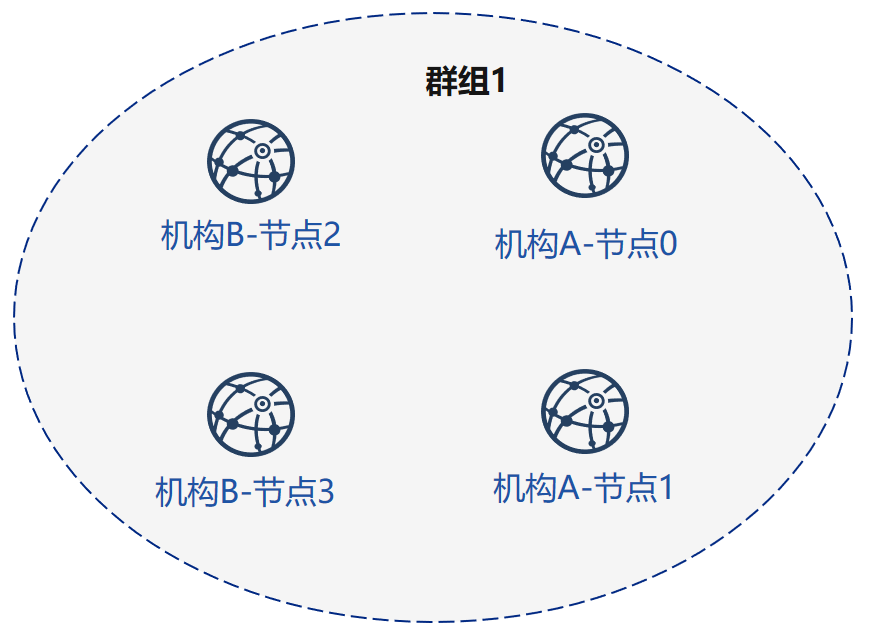
./generator --download\_fisco ./meta

使用以下语句检查二进制版本是否安装成功：

./meta/fisco-bcos -v

3.2组网模式结构

本项目构建的4节点2结构1群组的组网模式的区块链结构如图：



3.3机构，节点，群组，地址信息配置

每个节点的IP，端口号及所属信息如下表：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 机构 | 节点 | 所属群组 | P2P地址 | RPC/channel监听地址 |
| 机构A | 节点0 | 群组1、2 | 127.0.0.1:30300 | 0.0.0.0:8545/:20200 |
| 机构A | 节点1 | 群组1、2 | 127.0.0.1:30301 | 0.0.0.0:8546/:20201 |
| 机构B | 节点2 | 群组1 | 127.0.0.1:30302 | 0.0.0.0:8547/:20202 |
| 机构B | 节点3 | 群组1 | 127.0.0.1:30303 | 0.0.0.0:8548/:20203 |

需要注意的是，为了开发人员能够在开发后台的时候可以直接调试，我们允许外网进行访问，因此监听地址设置为广播地址，而对于后台访问云服务器的时候，其设置的P2P参数则为：



其中，云服务器端的公网ip为101.132.68.46。端口设置为搭建联盟链时设定的端口参数。

3.4机构初始化

使用以下命令初始化机构A和B

cp -r ~/generator ~/generator-A

cp -r ~/generator ~/generator-B

使用以下命令初始化链证书（在证书生成机构目录下操作）：

./generator --generate\_chain\_certificate ./dir\_chain\_ca

随后查看链证书及私钥情况：

6

需要注意的是，一条联盟链只能有一个唯一的链证书。

3.5机构A、B构建群组

由机构本地生成私钥agency.key，再生成证书请求文件，向证书签发机构获取机构证书agency.crt。

在证书生成机构目录下操作，生成机构A、B的证书，在生成之后发送链证书、机构证书、机构私钥至各自机构所属文件夹中，如通过文件拷贝的方式，从证书授权机构将机构证书发送给对应的机构，放到机构的工作目录的meta子目录下：

./generator --generate\_agency\_certificate ./dir\_agency\_ca ./dir\_chain\_ca agencyA

cp ./dir\_chain\_ca/ca.crt ./dir\_agency\_ca/agencyA/agency.crt ./dir\_agency\_ca/agencyA/agency.key ~/generator-A/meta/

./generator --generate\_agency\_certificate ./dir\_agency\_ca ./dir\_chain\_ca agencyB

cp ./dir\_chain\_ca/ca.crt ./dir\_agency\_ca/agencyB/agency.crt ./dir\_agency\_ca/agencyB/agency.key ~/generator-B/meta/

要注意的是，一条联盟链中只能用到一个根证书ca.crt，以及一个群组只能有一个群组创世区块group.x.genesis。

3.6分别修改配置文件

由于部署工具根据node\_deployment.ini的配置生成相关节点证书，以及生成节点配置文件夹等，因此需要根据云服务器端的情况进行部署和修改配置。

在机构A的文件夹generatorA下，修改conf/node\_deployment.ini，以以下内容配置相关节点，

：

[group]

group\_id=1

[node0]

; host ip for the communication among peers.

; Please use your ssh login ip.

p2p\_ip=127.0.0.1

; listen ip for the communication between sdk clients.

; This ip is the same as p2p\_ip for physical host.

; But for virtual host e.g. vps servers, it is usually different from p2p\_ip.

; You can check accessible addresses of your network card.

; Please see https://tecadmin.net/check-ip-address-ubuntu-18-04-desktop/

; for more instructions.

rpc\_ip=0.0.0.0

p2p\_listen\_port=30300

channel\_listen\_port=20200

jsonrpc\_listen\_port=8545

[node1]

p2p\_ip=127.0.0.1

rpc\_ip=0.0.0.0

p2p\_listen\_port=30301

channel\_listen\_port=20201

jsonrpc\_listen\_port=8546

EOF

同理，以相同形式部署机构B的conf/node\_deployment.ini：

[group]

group\_id=1

[node0]

; host ip for the communication among peers.

; Please use your ssh login ip.

p2p\_ip=127.0.0.1

; listen ip for the communication between sdk clients.

; This ip is the same as p2p\_ip for physical host.

; But for virtual host e.g. vps servers, it is usually different from p2p\_ip.

; You can check accessible addresses of your network card.

; Please see https://tecadmin.net/check-ip-address-ubuntu-18-04-desktop/

; for more instructions.

rpc\_ip=0.0.0.0

p2p\_listen\_port=30302

channel\_listen\_port=20202

jsonrpc\_listen\_port=8547

[node1]

p2p\_ip=127.0.0.1

rpc\_ip=0.0.0.0

p2p\_listen\_port=30303

channel\_listen\_port=20203

jsonrpc\_listen\_port=8548

EOF

3.7机构A、B各自生成节点并发送节点信息

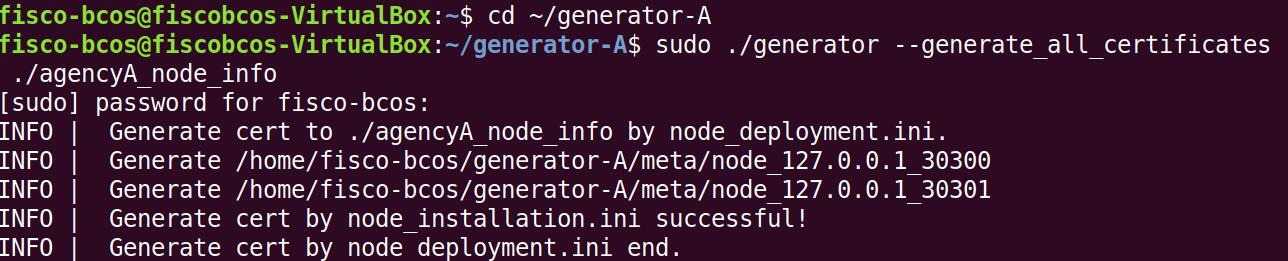
根据node\_deployment.ini，及机构meta文件夹下的机构证书与私钥分别在机构A、B的目录下生成节点证书及P2P连接信息文件。

对于机构A，依次执行以下语句：

./generator --generate\_all\_certificates ./agencyA\_node\_info

cp ./agencyA\_node\_info/peers.txt ~/generator-B/meta/peersA.txt

其中，由于机构生成节点时需要指定其他节点的节点P2P连接地址，因此，机构A需将节点P2P连接地址文件发送至机构B，对于机构B而言同理。其生成过程如图。



3.8由机构A生成该群组的创世区块

在机构A的目录下，修改conf文件夹下的group\_genesis.ini 内容如下：

[group]

group\_id=1

[nodes]

node0=127.0.0.1:30300

node1=127.0.0.1:30301

node2=127.0.0.1:30302

node3=127.0.0.1:30303

EOF

随后根据机构A的meta文件夹下配置的节点证书，生成group\_genesis.ini配置的群组创世区块，使用到了机构A的meta下有名为cert\_127.0.0.1\_30300.crt，cert\_127.0.0.1\_30301.crt，cert\_127.0.0.1\_30302.crt，cert\_127.0.0.1\_30303.crt的节点证书，也同样需要用到机构B的节点证书。

使用如下语句生成群组：

./generator --create\_group\_genesis ./group

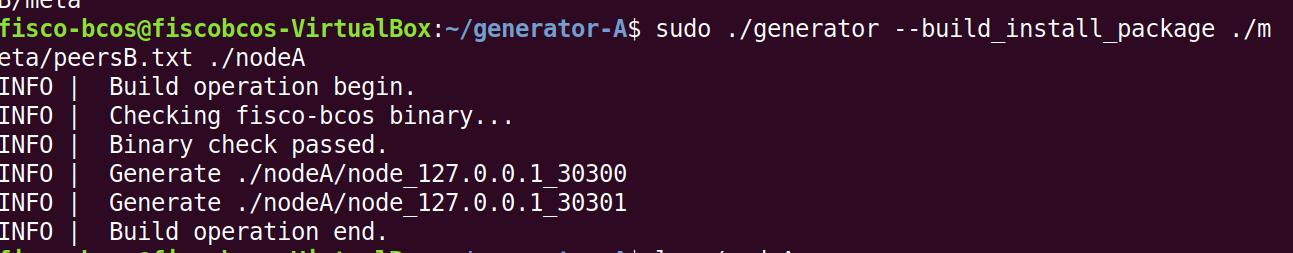
最后分发群组的创世区块到机构B：

cp ./group/group.1.genesis ~/generator-B/meta。

3.9各机构生成并启动所属节点

分别在各机构目录下根据用户配置的node\_deployment.ini文件生成相应的节点配置文件夹。执行如下命令：（以机构A为例）

./generator --build\_install\_package ./meta/peersB.txt ./nodeA

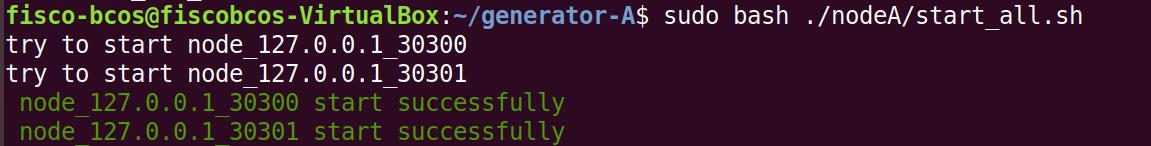


其中，monitor 为脚本，node\_...为对应地址服务器端口号的节点配置文件夹，scripts 为节点的相关工具脚本，start\_all.sh 和 stop\_all.sh 分别为批量节点启动和停止的脚本。

最后，启动节点（以机构A为例）：

bash ./nodeA/start\_all.sh

可以看到启动成功，对于机构B而言同理。

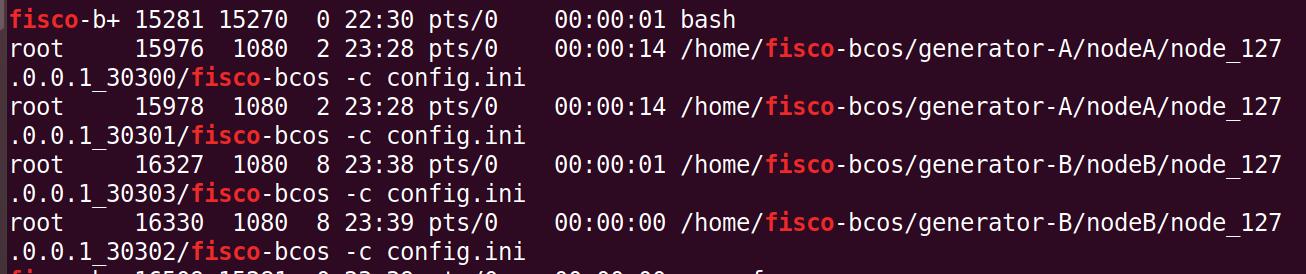


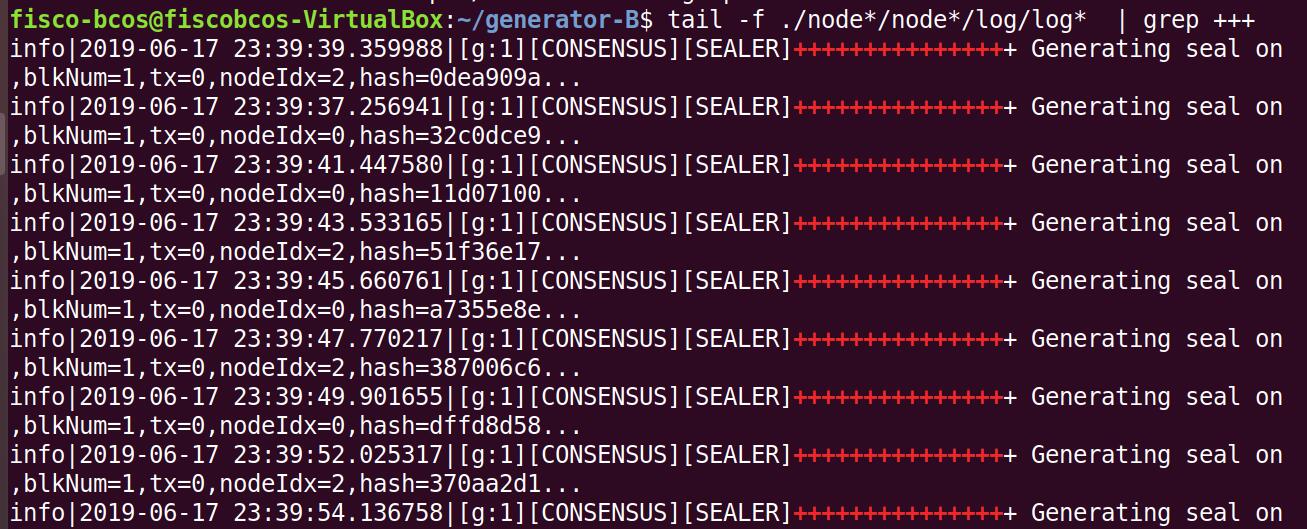
最后，通过以下命令查看进程和节点log验证搭建情况：

ps -ef | grep fisco

tail -f ./node\*/node\*/log/log\* | grep +++

分别显示情况如下：





至此，区块链网络2机构单群组搭建完成。

二、区块链技术原理

“区块链”技术最初是由一位化名中本聪的人为比特币（一种数字货币）而设计出的一种特殊的数据库技术，它基于密码学中的椭圆曲线数字签名算法（ECDSA）来实现去中心化的P2P系统设计。但区块链的作用不仅仅局限在比特币上。现在，人们在使用“区块链”这个词时，有的时候是指数据结构，有时是指数据库，有时则是指数据库技术，但无论是哪种含义，都和比特币没有必然的联系。[[1]](#endnote-1)

从数据的角度来看：区块链是一种分布式数据库（或称为分布式共享总账，DistributedShared Ledger），这里的“分布式”不仅体现为数据的分布式存储，也体现为数据的分布式记录（即由系统参与者来集体维护）。简单的说，区块链能实现全球数据信息的分布式记录（可以由系统参与者集体记录，而非由一个中心化的机构集中记录）与分布式存储（可以存储在所有参与记录数据的节点中，而非集中存储于中心化的机构节点中）。

从效果的角度来看：区块链可以生成一套记录时间先后的、不可篡改的、可信任的数据库，这套数据库是去中心化存储且数据安全能够得到有效保证的。

区块链是一种把区块以链的方式组合在一起的数据结构，它适合存储简单的、有先后关系的、能在系统内验证的数据，用密码学保证了数据的不可篡改和不可伪造。它能够使参与者对全网交易记录的事件顺序和当前状态建立共识。[[2]](#endnote-2)

如今的区块链技术概括起来是指通过去中心化和去信任的方式集体维护一个可靠数据库的技术。其实，区块链技术并不是一种单一的、全新的技术，而是多种现有技术（如加密算法、P2P文件传输等）整合的结果，这些技术与数据库巧妙地组合在一起，形成了一种新的数据记录、传递、存储与呈现的方式。简单的说，区块链技术就是一种大家共同参与记录信息、存储信息的技术。过去，人们将数据记录、存储的工作交给中心化的机构来完成，而区块链技术则让系统中的每一个人都可以参与数据的记录、存储。区块链技术在没有中央控制点的分布式对等网络下，使用分布式集体运作的方法，构建了一个P2P的自组织网络。通过复杂的校验机制，区块链数据库能够保持完整性、连续性和一致性，即使部分参与人作假也无法改变区块链的完整性，更无法篡改区块链中的数据。

区块链技术涉及的关键点包括：去中心化（Decentralized）、去信任（Trustless）、集体维护（Collectively maintain）、可靠数据库（ReliableDatabase）、时间戳（Time stamp）、非对称加密（AsymmetricCryptography）等。

区块链技术重新定义了网络中信用的生成方式：在系统中，参与者无需了解其他人的背景资料，也不需要借助第三方机构的担保或保证，区块链技术保障了系统对价值转移的活动进行记录、传输、存储，其最后的结果一定是可信的。

区块链技术原理的来源可归纳为一个数学问题：拜占庭将军问题。拜占庭将军问题延伸到互联网生活中来，其内涵可概括为：在互联网大背景下，当需要与不熟悉的对手方进行价值交换活动时，人们如何才能防止不会被其中的恶意破坏者欺骗、迷惑从而做出错误的决策。进一步将拜占庭将军问题延伸到技术领域中来，其内涵可概括为：在缺少可信任的中央节点和可信任的通道的情况下，分布在网络中的各个节点应如何达成共识。区块链技术解决了闻名已久的拜占庭将军问题——它提供了一种无需信任单个节点、还能创建共识网络的方法。

区块链技术的本质是一种互联网协议。

设想一下，如果现在我们想要在互联网世界中建立一套全球通用的数据库，那么我们会面临三个亟待解决的问题，这三个问题也是设计区块链技术的核心所在：

问题一：如何建立一个严谨的数据库，使得该数据库能够存储下海量的信息，同时又能在没有中心化结构的体系下保证数据库的完整性？

问题二：如何记录并存储下这个严谨的数据库，使得即便参与数据记录的某些节点崩溃，我们仍然能保证整个数据库系统的正常运行与信息完备？

问题三：如何使这个严谨且完整存储下来的数据库变得可信赖，使得我们可以在互联网无实名背景下成功防止诈骗？

针对这三个核心问题，区块链构建了一整套完整的、连贯的数据库技术来达成目的，解决这三个问题的技术也成为了区块链最核心的三大技术。此外，为了保证区块链技术的可进化性与可扩展性，区块链系统设计者还引入了“脚本”的概念来实现数据库的可编程性。我们认为，这四大技术构成了区块链的核心技术。

关于如何建立一个严谨数据库的问题，区块链的办法是：将数据库的结构进行创新，把数据分成不同的区块，每个区块通过特定的信息链接到上一区块的后面，前后顺连来呈现一套完整的数据，这也是“区块链”这三个字的来源。

 区块（block）：在区块链技术中，数据以电子记录的形式被永久储存下来，存放这些电子记录的文件我们就称之为“区块（block）”。区块是按时间顺序一个一个先后生成的，每一个区块记录下它在被创建期间发生的所有价值交换活动，所有区块汇总起来形成一个记录合集。

区块结构（BlockStructure）：区块中会记录下区块生成时间段内的交易数据，区块主体实际上就是交易信息的合集。每一种区块链的结构设计可能不完全相同，但大结构上分为块头（header）和块身（body）两部分。块头用于链接到前面的块并且为区块链数据库提供完整性的保证，块身则包含了经过验证的、块创建过程中发生的价值交换的所有记录。

区块结构有两个非常重要的特点：第一，每一个区块上记录的交易是上一个区块形成之后、该区块被创建前发生的所有价值交换活动，这个特点保证了数据库的完整性。第二，在绝大多数情况下，一旦新区块完成后被加入到区块链的最后，则此区块的数据记录就再也不能改变或删除。这个特点保证了数据库的严谨性，即无法被篡改。

顾名思义，区块链就是区块以链的方式组合在一起，以这种方式形成的数据库我们称之为区块链数据库。区块链是系统内所有节点共享的交易数据库，这些节点基于价值交换协议参与到区块链的网络中来。

区块链是如何做到的呢？由于每一个区块的块头都包含了前一个区块的交易信息压缩值，这就使得从创世块（第一个区块）到当前区块连接在一起形成了一条长链。由于如果不知道前一区块的“交易缩影”值，就没有办法生成当前区块，因此每个区块必定按时间顺序跟随在前一个区块之后。这种所有区块包含前一个区块引用的结构让现存的区块集合形成了一条数据长链。“区块+链”的数据存储结构如下图所示。

我们引用《区块链：互联网金融的终局》（肖风）的一段话来总结区块链的基本结构：“人们把一段时间内生成的信息（包括数据或代码）打包成一个区块，盖上时间 戳，与上一个区块衔接在一起，每下一个区块的页首都包含了上一个区块的索引数据，然后再在本页中写入新的信息，从而形成新的区块，首尾相连，最终形成了区块链。”这个结构的神奇之处：区块（完整历史）+ 链（完全验证）= 时间戳

“区块+链”的结构为我们提供了一个数据库的完整历史。从第一个区块开始，到最新产生的区块为止，区块链上存储了系统全部的历史数据。

区块链为我们提供了数据库内每一笔数据的查找功能。区块链上的每一条交易数据，都可以通过“区块链”的结构追本溯源，一笔一笔进行验证。

区块+链=时间戳，这是区块链数据库的最大创新点。区块链数据库让全网的记录者在每一个区块中都盖上一个时间戳来记账，表示这个信息是这个时间写入的，形成了一个不可篡改、不可伪造的数据库。我们认为，时间戳是区块链中一项伟大的技术创新，它可以证明什么呢？

我们有了区块+链的数据之后，接下来就要考虑记录和存储的问题了。我们应该让谁来参与数据的记录，又应该把这些盖了时间戳的数据存储在哪里呢？在现如今中心化的体系中，数据都是集中记录并存储于中央电脑上。但是区块链结构设计精妙的地方就在这里，它并不赞同把数据记录并存储在中心化的一台或几台电脑上，而是让每一个参与数据交易的节点都记录并存储下所有的数据。

1.关于如何让所有节点都能参与记录的问题，区块链的办法是：构建一整套协议机制，让全网每一个节点在参与记录的同时也来验证其他节点记录结果的正确性。只有当全网大部分节点（或甚至所有节点）都同时认为这个记录正确时，或者所有参与记录的节点都比对结果一致通过后，记录的真实性才能得到全网认可，记录数据才允许被写入区块中。

2.关于如何存储下“区块链”这套严谨数据库的问题，区块链的办法是：构建一个分布式结构的网络系统，让数据库中的所有数据都实时更新并存放于所有参与记录的网络节点中。这样即使部分节点损坏或被黑客攻击，也不会影响整个数据库的数据记录与信息更新。

区块链根据系统确定的开源的、去中心化的协议，构建了一个分布式的结构体系，让价值交换的信息通过分布式传播发送给全网，通过分布式记账确定信息数据内容，盖上时间戳后生成区块数据，再通过分布式传播发送给各个节点，实现分布式存储。

从硬件的角度讲，区块链的背后是大量的信息记录储存器（如电脑等）组成的网络，这一网络如何记录发生在网络中的所有价值交换活动呢？区块链设计者没有为专业的会计记录者预留一个特定的位置，而是希望通过自愿原则来建立一套人人都可以参与记录信息的分布式记账体系，从而将会计责任分散化，由整个网络的所有参与者来共同记录。

区块链中每一笔新交易的传播都采用分布式的结构，根据P2P网络层协议，消息由单个节点被直接发送给全网其他所有的节点。

区块链技术让数据库中的所有数据均存储于系统所有的电脑节点中，并实时更新。完全去中心化的结构设置使数据能实时记录，并在每一个参与数据存储的网络节点中更新，这就极大的提高了数据库的安全性。

通过分布式记账、分布式传播、分布式存储这三大“分布”我们可以发现，没有人、没有组织、甚至没有哪个国家能够控制这个系统，系统内的数据存储、交易验证、信息传输过程全部都是去中心化的。在没有中心的情况下，大规模的参与者达成共识，共同构建了区块链数据库。可以说，这是人类历史上第一次构建了一个真正意义上的去中心化体系。甚至可以说，区块链技术构建了一套永生不灭的系统——只要不是网络中的所有参与节点在同一时间集体崩溃，数据库系统就可以一直运转下去。

我们现在已经有了一套严谨的数据库，也有了记录并存储这套数据库的可用协议，那么当我们将这套数据库运用于实际社会时，我们要解决最核心的一个问题（问题三）是：如何使这个严谨且完整存储下来的数据库变得可信赖，使得我们可以在互联网无实名背景下成功防止诈骗？

什么是非对称加密？简单来说，它让我们在“加密”和“解密”的过程中分别使用两个密码，两个密码具有非对称的特点：（1）加密时的密码（在区块链中被称为“公钥”）是公开全网可见的，所有人都可以用自己的公钥来加密一段信息（信息的真实性）；（2）解密时的密码（在区块链中被称为“私钥”）是只有信息拥有者才知道的，被加密过的信息只有拥有相应私钥的人才能够解密（信息的安全性）。[[3]](#endnote-3)

简单的总结：区块链系统内，所有权验证机制的基础是非对称加密算法。常见的非对称加密算法包括RSA、Elgamal、D-H、ECC（椭圆曲线加密算法）等。在非对称加密算法中，如果一个“密钥对”中的两个密钥满足以下两个条件：1、对信息用其中一个密钥加密后，只有用另一个密钥才能解开；2、其中一个密钥公开后，根据公开的密钥别人也无法算出另一个，那么我们就称这个密钥对为非对称密钥对，公开的密钥称为公钥，不公开的密钥称为私钥。在区块链系统的交易中，非对称密钥的基本使用场景有两种：1、公钥对交易信息加密，私钥对交易信息解密。私钥持有人解密后，可以使用收到的价值。2、私钥对信息签名，公钥验证签名。通过公钥签名验证的信息确认为私钥持有人发出。

我们可以看出，从信任的角度来看，区块链实际上是数学方法解决信任问题的产物。过去，人们解决信任问题可能依靠熟人社会的“老乡”，政党社会的“同志”，传统互联网中的交易平台“支付宝”。而区块链技术中，所有的规则事先都以算法程序的形式表述出来，人们完全不需要知道交易的对手方是“君子”还是“小人”，更不需要求助中心化的第三方机构来进行交易背书，而只需要信任数学算法就可以建立互信。区块链技术的背后，实质上是算法在为人们创造信用，达成共识背书。

脚本可以理解为一种可编程的智能合约。如果区块链技术只是为了适应某种特定的交易，那脚本的嵌入就没有必要了，系统可以直接定义完成价值交换活动需要满足的条件。然而，在一个去中心化的环境下，所有的协议都需要提前取得共识，那脚本的引入就显得不可或缺了。有了脚本之后，区块链技术就会使系统有机会去处理一些无法预见到的交易模式，保证了这一技术在未来的应用中不会过时，增加了技术的实用性。

一个脚本本质上是众多指令的列表，这些指令记录在每一次的价值交换活动中，价值交换活动的接收者（价值的持有人）如何获得这些价值，以及花费掉自己曾收到的留存价值需要满足哪些附加条件。通常，发送价值到目标地址的脚本，要求价值的持有人提供以下两个条件，才能使用自己之前收到的价值：一个公钥，以及一个签名（证明价值的持有者拥有与上述公钥相对应的私钥）。脚本的神奇之处在于，它具有可编程性：（1）它可以灵活改变花费掉留存价值的条件，例如脚本系统可能会同时要求两个私钥、或几个私钥、或无需任何私钥等；（2）它可以灵活的在发送价值时附加一些价值再转移的条件，例如脚本系统可以约定这一笔发送出去的价 值以后只能用于支付中信证券的手续费、或支付给政府等。

三、联盟链和公有链的异同

公有链的概念为：世界上所有人都可以阅读和发送交易。如果他们合法都有希望看到自己被包括在内。世界上任何人都能参与到共识形成过程——决定在链条上添加什么区块以及现状是怎样的。作为中心化或准中心化信任的替代品，公有链受加密经济的保护，加密经济是经济激励和加密图形验证的结合，用类似工作量证明或权益证明的机制，遵循的总原则是人们影响共识形成的程度和他们能够影响的经济资源数量成正比。这类区块链通常被认为是“完全去中心化“。

而联盟链则是这样的，共识形成过程由预先选择的一系列的节点所掌控，例如，设想一个有15个金融机构的团体，每个机构都操作一个节点，为了使区块生效，其中的10个必须签署那个区块。阅读区块链的权利可能是公开的，或仅限于参与者，也有混合的路径，比如区块的根散表和应用程序编程接口一起公开，使公共成员可以进行一定量的查询，重获一部分区块链状态的加密图形证明。这类区块链被认为是“部分去中心化”。 [[4]](#endnote-4)

私有链：书写许可对一个组织保持中心化。阅读许可可能是公开的或者限制在任意程度。应用很可能包含对单个公司内部的数据库管理，审查等，因此公共的可读性在很多情况下根本不必要，但在另一些情况下人们又想要公共可读性。

私有链/联盟链可能和公有链毫无联系，他们仍然通过投资以太坊软件开发，对以太坊整体生态系统有利。经过一段时间，这会转变成软件改善，知识共享和工作机会。

公共区块链的优点可以归结为两大类：

1. 保护用户，免受开发者的影响

在公共区块链中程序开发者无权干涉用户，所以区块链可以保护使用他们开发的程序的用户。从天真的角度来看，的确难以理解为何程序开发者会愿意放弃自己的权限。然而，较为超前的经济分析为此提供了两个理由：借用Thomas Schelling的话语，妥协是一种力量。第一，如果你明确地选择做一些很难或者不可能的事情，其他人会更容易信任你并与您产生互动，因为他们自信那些事情不大可能发生在他们身上。第二，如果你是受人或其他外界因素的强迫，无法去做自己想做的事，你大可说句“即使我想，但我也没有权力去做”的话语作为谈判筹码，这样可以劝阻对方不去强迫你去做不情愿的事。程序开发者们所面临的主要的压力或者说风险，主要是来自政府，所以说“审查阻力”，便是公共区块链最大的优势。

2.网络效应

公共区块链是开放的，因此有可能被许多外界用户使用和产生一定的网络效应。举一个特定的例子，就拿域名托管来说吧。现在，如果A想卖给B一个域名，就有个需要待解决的风险问题：如果A首先出售了域名，但B可能还没给钱；或者如果B给钱了，但A还没出售域名。为解决这个问题，我们要设立中心化的托管中介，但须支付三到六个百分点的手续费。然而，如果我们在区块链上拥有一个域名系统，并使用这个区块链的货币，那么我们可建立交易费低至0的智能合约：A向该系统出售域名，系统马上将域名出售给首先支付资金的人，而且因为这系统是建立在公共区块链上所以值得信任。但注意为了使交易过程更高效，要将来自完全不同行业的完全不同的资产寄放在同一公共数据库上——这在私有区块链上是不可能轻易做到的。同样的例子可以是土地登记和产权保险，但注意若想可交互操作，要使用能被公共区块链验证的私有区块链，这样可通过跨链完成交易。

与公有链相比，联盟链有几个显著的特点：

1.交易速度非常快

联盟链本质上还是私有链，由于其节点不多，就算少量的节点也都具有很高的信任度，并不需要每个节点来验证一个交易，达成共识容易，交易速度自然也就快很多。

2.部分去中心化

与公有链不一样，联盟链在某种程度上只属于联盟内部的成员所有，且很容易达成共识，因为毕竟联盟链的节点数是非常有限的，所以只是部分去中心化。

3.可控性较强

公有链是一旦区块链形成，将不可篡改，这主要源于公有链的节点一般是海量的，比如比特币节点太多，想要篡改区块数据，几乎不可能，而联盟链，只要所有机构中的大部分达成共识，即可将区块数据进行更改；

4.数据不会默认公开

不同于公有链，联盟链的数据只限于联盟里的机构及其用户才有权限进行访问，其他人即使能连接上网络，也无法获得数据。

5、交易成本很低

与私有链一样，可以进行完全免费或者至少说是非常廉价的交易。如果一个实体机构控制和处理所有的交易，就省掉了区块链网络中常见的“矿工费”，这就不需要花费额外的费用。

随着应用场景的需求更复杂，区块链技术变得越来越复杂，无论是公有链、联盟链还是私有链都没有绝对的优劣，往往需要根据不同的场景来选择合适的区块链类型。

四、区块链技术中分布式存储的优点

分布式存储的优点可以概述为以下的六个优点：

1. 高性能

一个具有高性能的分布式存户通常能够高效地管理读缓存和写缓存，并且支持自动的分级存储。分布式存储通过将热点区域内数据映射到高速存储中，来提高系统响应速度;一旦这些区域不再是热点，那么存储系统会将它们移出高速存储。而写缓存技术则可使配合高速存储来明显改变整体存储的性能，按照一定的策略，先将数据写入高速存储，再在适当的时间进行同步落盘。[[5]](#endnote-5)

2. 支持分级存储

由于通过网络进行松耦合链接，分布式存储允许高速存储和低速存储分开部署，或者任意比例混布。在不可预测的业务环境或者敏捷应用情况下，分层存储的优势可以发挥到最佳。解决了目前缓存分层存储最大的问题是当性能池读不命中后，从冷池提取数据的粒度太大，导致延迟高，从而给造成整体的性能的抖动的问题。

3. 多副本的一致性

与传统的存储架构使用RAID模式来保证数据的可靠性不同，分布式存储采用了多副本备份机制。在存储数据之前，分布式存储对数据进行了分片，分片后的数据按照一定的规则保存在集群节点上。为了保证多个数据副本之间的一致性，分布式存储通常采用的是一个副本写入，多个副本读取的强一致性技术，使用镜像、条带、分布式校验等方式满足租户对于可靠性不同的需求。在读取数据失败的时候，系统可以通过从其他副本读取数据，重新写入该副本进行恢复，从而保证副本的总数固定;当数据长时间处于不一致状态时，系统会自动数据重建恢复，同时租户可设定数据恢复的带宽规则，最小化对业务的影响。[[6]](#endnote-6)

4. 容灾与备份

在分布式存储的容灾中，一个重要的手段就是多时间点快照技术，使得用户生产系统能够实现一定时间间隔下的各版本数据的保存。特别值得一提的是，多时间点快照技术支持同时提取多个时间点样本同时恢复，这对于很多逻辑错误的灾难定位十分有用，如果用户有多台服务器或虚拟机可以用作系统恢复，通过比照和分析，可以快速找到哪个时间点才是需要回复的时间点，降低了故障定位的难度，缩短了定位时间。这个功能还非常有利于进行故障重现，从而进行分析和研究，避免灾难在未来再次发生。多副本技术，数据条带化放置，多时间点快照和周期增量复制等技术为分布式存储的高可靠性提供了保障。

5. 弹性扩展

得益于合理的分布式架构，分布式存储可预估并且弹性扩展计算、存储容量和性能。分布式存储的水平扩展有以下几个特性：1) 节点扩展后，旧数据会自动迁移到新节点，实现负载均衡，避免单点过热的情况出现；2) 水平扩展只需要将新节点和原有集群连接到同一网络，整个过程不会对业务造成影响；3) 当节点被添加到集群，集群系统的整体容量和性能也随之线性扩展，此后新节点的资源就会被管理平台接管，被用于分配或者回收。

6. 存储系统标准化

随着分布式存储的发展，存储行业的标准化进程也不断推进，分布式存储优先采用行业标准接口(SMI-S或OpenStack Cinder)进行存储接入。在平台层面，通过将异构存储资源进行抽象化，将传统的存储设备级的操作封装成面向存储资源的操作，从而简化异构存储基础架构的操作，以实现存储资源的集中管理，并能够自动执行创建、变更、回收等整个存储生命周期流程。基于异构存储整合的功能，用户可以实现跨不同品牌、介质地实现容灾，如用中低端阵列为高端阵列容灾，用不同磁盘阵列为闪存阵列容灾等等，从侧面降低了存储采购和管理成本。

五、总结

在六个星期的企业实训过程中，我学习到了许多有关区块链的知识，更在微众银行的几位老师指导下亲身参与和实践了有关区块链和联盟链的项目，将有关音乐版权和区块链结合的创意落地实践，而且得到了各位老师的认可。区块链是一项尚未完全成熟的技术，其分布式存储的技术能够很好地表现出高性能，并且对于数据的安全性等有了更高层次的保障。同时，我们更意识到虽然最原始的区块链技术起源于虚拟货币及公有链项目，但公有链的项目方往往以融资为目的，其用户则是以价格交易获利为目标，导致各方更多是关注币价的涨跌而非区块链的真正应用能力。由于公有链的代币实质上是“类货币”与“类证券”，已经被中国的监管部门严厉叫停。当潮水退去、大浪淘沙后，联盟链技术已肩负起推动区块链技术继续前行的重任。2018年，业界更是提出“公众联盟链”的发展路线，呼吁联盟链应该积极开放开源，从较为封闭的联盟内或公司内走向大众，让普罗大众真正感受到区块链带来的体验提升、效率提升、成本下降、信任增强、数据互换、责任追溯等好处，实现分布式商业的愿景。希望能够在未来，在区块链技术日趋成熟的时候，将其技术推广到各行业中，让区块链技术更好地服务大众，造福大众。

**参考文献：**

1. **傅晓阳.区块链技术应用探索.中国金融,2018(2):73-74.** [↑](#endnote-ref-1)
2. **朱建明,付永贵.区块链应用研究进展.科技导报,2017,35(13):70-76.** [↑](#endnote-ref-2)
3. **蒋春凤.非对称加密算法.内江科技,2012(8):148-148.蒋春凤.非对称加密算法.内江科技,2012(8):148-148.** [↑](#endnote-ref-3)
4. **高志豪.公有链和联盟链的道法术器.金卡工程,2017(3):35-39.** [↑](#endnote-ref-4)
5. **杨传辉, 大规模分布式存储系统: 原理解析与架构实战. 北京: 机械工业出版社, 20132001.** [↑](#endnote-ref-5)
6. **张健,汪洋,刘丹丹.分布式一致性算法Yac.计算机应用,2017,37(9):2524-2530.**张健,汪洋,刘丹丹.分布式一致性算法Yac.计算机应用,2017,37(9):2524-2530. [↑](#endnote-ref-6)