华南理工大学

课程报告

课程名称：	企业软件项目实训
学生姓名：	吴峻羽
学生学号：	**201630665892**
学生专业：	软件工程
开课学期：	**2018-2019 第二学期**

软件学院

2019 年 6 月

目 录

前 言.....1

一、区块链简介.....1

1.1 多角度看区块链.....1

1.2 区块链技术发展.....2

二、 区块链技术原理.....3

2.1 密码学原理.....3

2.2 区块数据结构.....7

2.3 分布式结构.....9

三、区块链技术选型.....12

3.1 公有链.....12

3.2 私有链.....13

3.3 联盟链.....13

四、区块链应用开发.....14

4.1 智能合约.....14

4.2 开发平台.....15

4.3 DAPP.....19

要 点.....20

总 结.....21

参考文献.....22

微众银行企业实训课程——区块链学习报告

前 言

“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。

一、区块链简介

1.1 多角度看区块链

1.1.1 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）。

1.1.2 从数学的角度来看

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。拜占庭帝国军队的将军们必须全体一致的决定是否攻击某一支敌军。问题是这些将军在地理上是分隔开来的，并且将军中存在叛徒。叛徒可以任意行动以达到以下目标：欺骗某些将军采取进攻行动；促成一个不是所有将军都同意的决定，如当将军们不希望进攻时促成进攻行动；或者迷惑某些将军，使他们无法做出决定。如果叛徒达到了这些目的之一，则任何攻击行动的结果都是注定要失败的，只有完全达成一致的努力才能获得胜利。

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。

1.1.3 从效果的角度来看

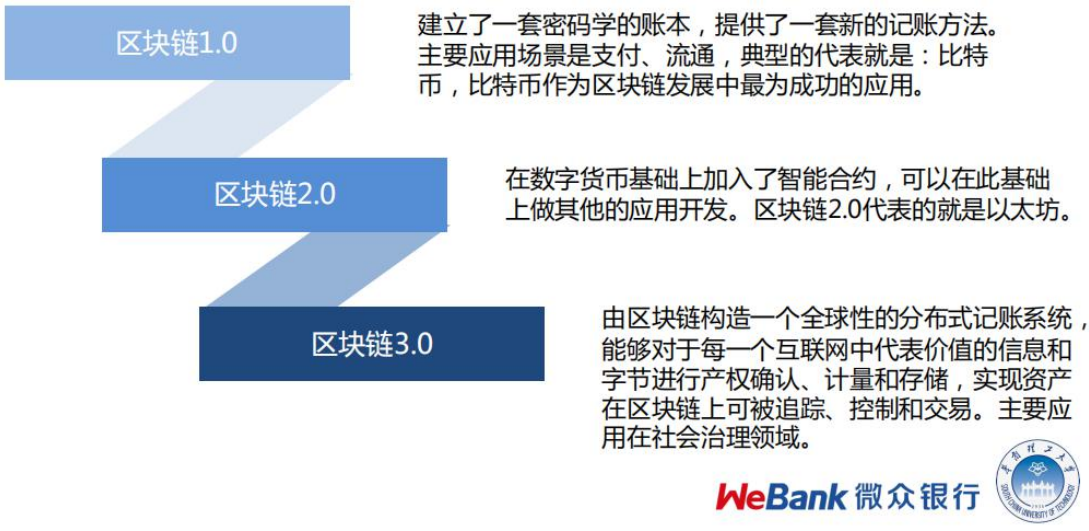
区块链可以生成一套记录时间先后的、不可篡改的、可信任的数据库，这套数据库是去中心化存储且数据安全能够得到有效保证的。区块链技术重新定义了网络中信用的生成方式：在系统中，参与者无需了解其他人的背景资料，也不需要借助第三方机构的担保或保证，区块链技术保障了系统对价值转移的活动进行记录、传输、存储，其最后的结果一定是可信的。

1.1.4 小结

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。

1.2 区块链技术发展

1.2.1 三个阶段



1.2.2 可靠数据库

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。

二、区块链技术原理

2.1 密码学原理

2.1.1 哈希函数

哈希(hash)，就是把任意长度的输入，通过散列算法，变换成固定长度的输出，该输出就是散列值。

哈希函数有几个特点:

- 同样的原始信息用同一个哈希函数总能得到相同的摘要信息
- 原始信息任何微小的变化都会哈希出面目全非的摘要信息
- 从摘要信息无法逆向推算出原始信息

哈希函数在区块链中，可用于生成各种数据的摘要，当比较两个数据是否相等时，只需要比较其摘要。例如，比较两个交易是否相等，只需要比较两者的

hash 值，达到快速验证的效果。

同时，应用哈希函数还可以防止篡改。传递一个数据，要保证它在传递过程中不被篡改，只需要同时传递它的摘要即可。收到数据的人将这个数据重新生成摘要，然后比较传递的摘要和生成的摘要是否相等，如果相等，则说明数据在传递过程中没有被篡改。

2.1.2 非对称加密

现今密码算法可分为对称密码（Symmetric Cryptology）和非对称密码（Asymmetric Cryptology）；其区分依据主要是所采用的密钥间的关系。在对称密码中，加密密钥和解密密钥是完全相同的，或彼此之间容易互相推导。在非对称密码算法，或称为公钥密码（Public Key Cryptology）中，加密密钥和解密密钥是不同的，从加密密钥推导出解密密钥在计算上是不可行的（Computationally infeasible）。把密钥分为公钥和私钥，公钥是公开的所有人都可以认领，私钥是保密的只有一个人知道。

假设 A 要发送一封 Email 给 B，他不想让任何其他人在传输中看到 Email 的内容，做法就是使用 B 的公钥对 Email 加密，只有 B 的私钥能够解密（B 的私钥唯一性保证信件不会泄露）。

某天，有黑客冒充 A 给 B 发送 Email，并且也用 B 的公钥加密，导致 B 无法区分这封邮件是否来自 A。此时 A 可以用自己的私钥加密，那么 B 收到邮件后如果用 A 的公钥可以解密邮件，那么证明这封信肯定来自于 A。

总结：

- 公钥的作用：对内容本身加密，保证不被其他人看到
- 私钥的作用：证明内容的来源
- 公钥和私钥是配对关系，公钥加密就用私钥解密，反之亦然，用错的密钥来尝试解密会报错

2.1.3 数字签名

假设 A 用自己的私钥对 Email 加密发送，这存在下面问题：

对文件本身加密可能是个耗时过程，比如这封 Email 足够大，那么私钥加密整个文件以及拿到文件后的解密无疑是巨大的开销。应用数字签名可以解决这个问题：

a. A 先对这封 Email 执行哈希运算得到 hash 值简称“摘要”，取名 h1

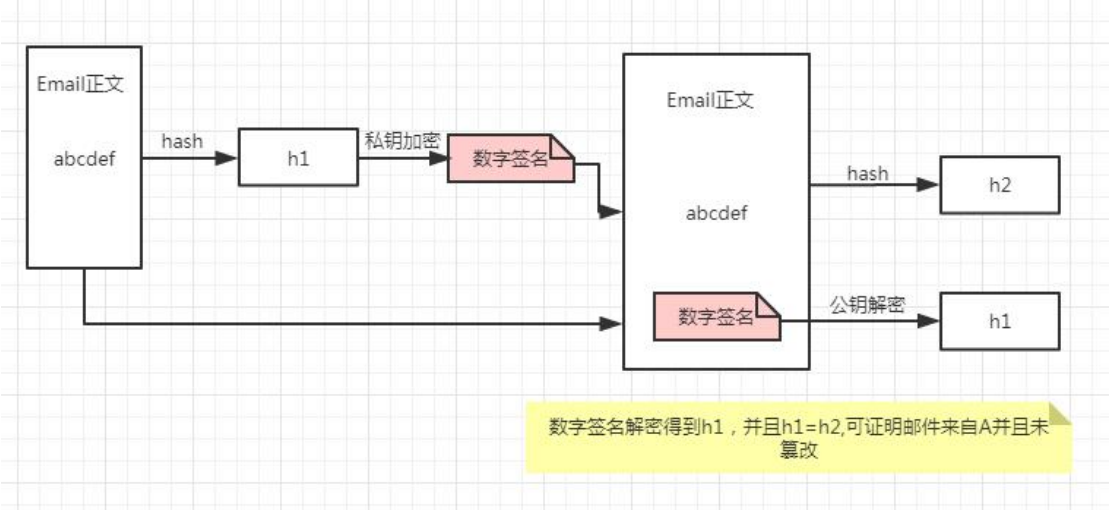
b. 然后用自己私钥对摘要加密，生成的东西叫“数字签名”

c. 把数字签名加在 Email 正文后面，一起发送给 B（当然，为了防止邮件被窃听你可以用继续公钥加密，这个不属于数字签名范畴）

d. B 收到邮件后用 A 的公钥对数字签名解密，成功则代表 Email 确实来自 A，失败说明有人冒充

e. B 对邮件正文执行哈希运算得到 hash 值，取名 h2

f. B 会对比第 4 步数字签名的 hash 值 h1 和自己运算得到的 h2，一致则说明邮件未被篡改。



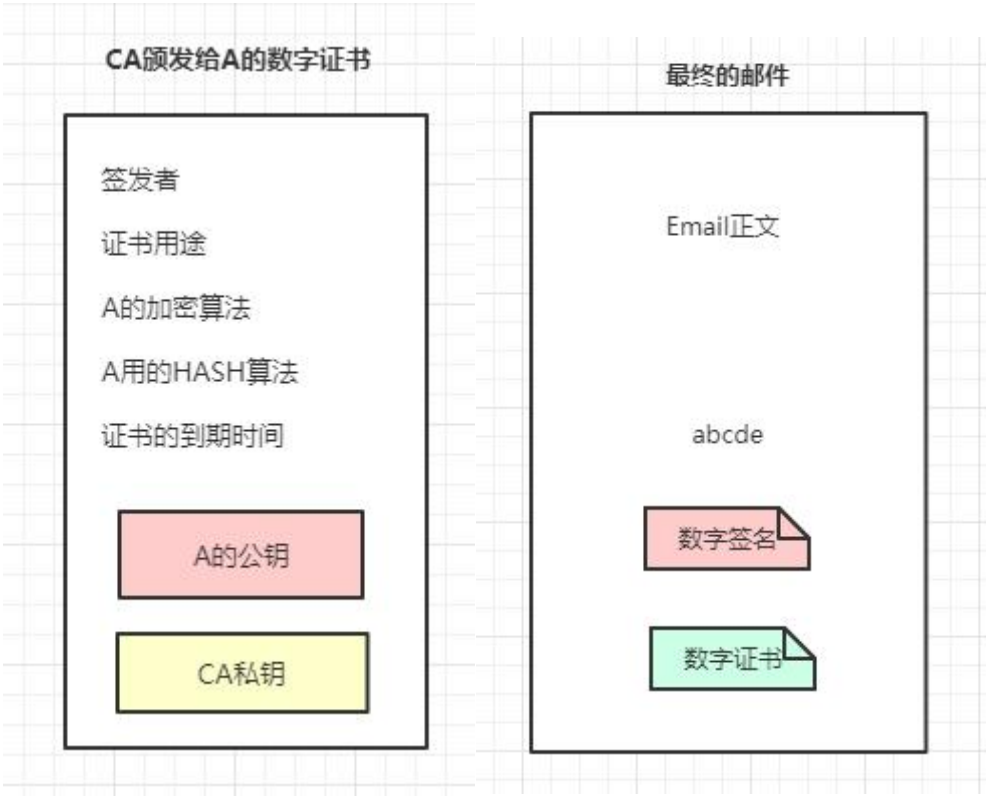
总结：

- 数字签名是利用算法（不一定是非对称算法）对原文 hash 值加密，然后附着到原文的一段数据。数字签名的作用就是验证数据来源以及数据完整性，解密过程则称为数字签名验证。

2.1.4 数字证书

公钥是公开的并且可以自行导入到电脑，如果有人比如 C 偷偷在 B 的电脑用自己公钥替换了 A 的公钥，然后用自己的私钥给 B 发送 Email，这时 B 收到邮件其实是被 C 冒充的但是他无法察觉。解决办法就是**数字证书**。那么数字证书是怎么生成的呢？以及如何配合数字签名工作呢？

a. A 去找"证书中心"（certificate authority，简称 CA），为公钥做认证。证书中心用自己的私钥，对 A 的公钥和一些相关信息一起加密，生成"数字证书"（Digital Certificate）：



b. A 在邮件正文下方除了数字签名，另外加上这张数字证书

c. B 收到 Email 后用 CA 的公钥解密这份数字证书，拿到 A 的公钥，然后验证数字签名，后面流程就和图 1 的流程相同，不再赘述。

2.1.5 PKI 体系

在使用公钥体制的网络环境中，必须向公钥的使用者证明公钥的真实合法性。因此，在公钥体制环境中，必须有一个可信的机构来对任何一个主体的公钥

进行公证，证明主体的身份以及它与公钥的匹配关系。目前较好的解决方案是引进数字证书(Certificate)机制，并由此构建形成公钥基础设施（Public Key Infrastructure、PKI）来提供服务。

PKI 是由公钥密码技术、数字证书、证书认证中心和关于公开密钥的安全策略等基本成分共同组成，管理密钥和证书的系统或平台。一个完整的 PKI 系统必须具备证书认证机构（Certificate Authority、CA）、数字证书库、密钥备份及恢复系统、证书作废系统和应用接口（API）等基本组成部分。

2.2 区块数据结构

区块链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。

2.2.1 区块

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。

2.2.2 区块结构

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。

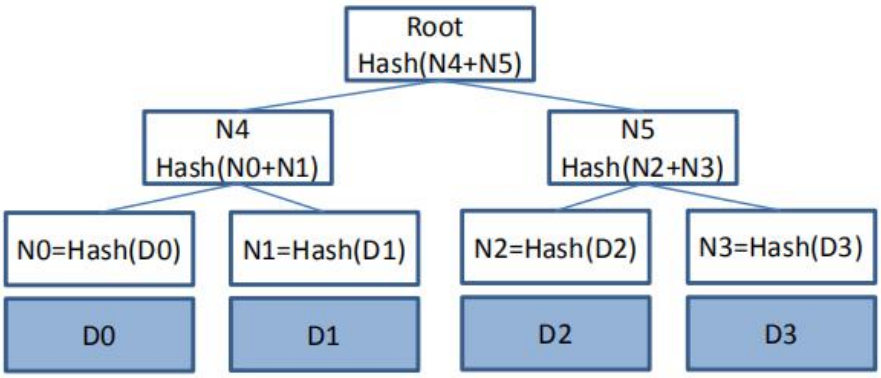
区块结构（BlockStructure）有两个非常重要的特点：第一，每一个区块上记录的交易是上一个区块形成之后、该区块被创建前发生的所有价值交换活动，这个特点保证了数据库的完整性。第二，在绝大多数情况下，一旦新区块完成后被加入到区块链的最后，则此区块的数据记录就再也不能改变或删除。这个特点保证了数据库的严谨性，即无法被篡改。

2.2.3 Merkle 树

Merkle tree（默克尔树），常叫它 merkle 树，是一种哈希二叉树。在计算机科学中，二叉树是每个节点最多有两个子树的树结构，每个节点代表一条结构化数据。通常子树被称作“左子树”（left subtree）和“右子树”（right subtree）。二叉树常被用于实现数据快速查询。

Merkle 树由一个根节点、一组中间节点和一组叶节点组成。叶节点包含存储数据或其哈希值，中间节点是它的两个孩子节点内容的哈希值，根节点也是由它的两个子节点内容的哈希值组成。所以 Merkle 树也称哈希树。

区块链中每个区块都会有一个 Merkle 树，它从叶子节点（树的底部）开始，一个叶子节点就是一个交易哈希。叶子节点的数量必须是双数，但是并非每个块都包含了双数的交易。如果一个块里面的交易数为单数，那么就将最后一个叶子节点（也就是 Merkle 树的最后一个交易，不是区块的最后一笔交易）复制一份凑成双数。从下往上，两两成对，连接两个节点哈希，将组合哈希作为新的哈希。新的哈希就成为新的树节点。重复该过程，直到仅有一个节点，也就是树根。根哈希然后就会当做是整个块交易的唯一标示，将它保存到区块头，然后用于工作量证明。

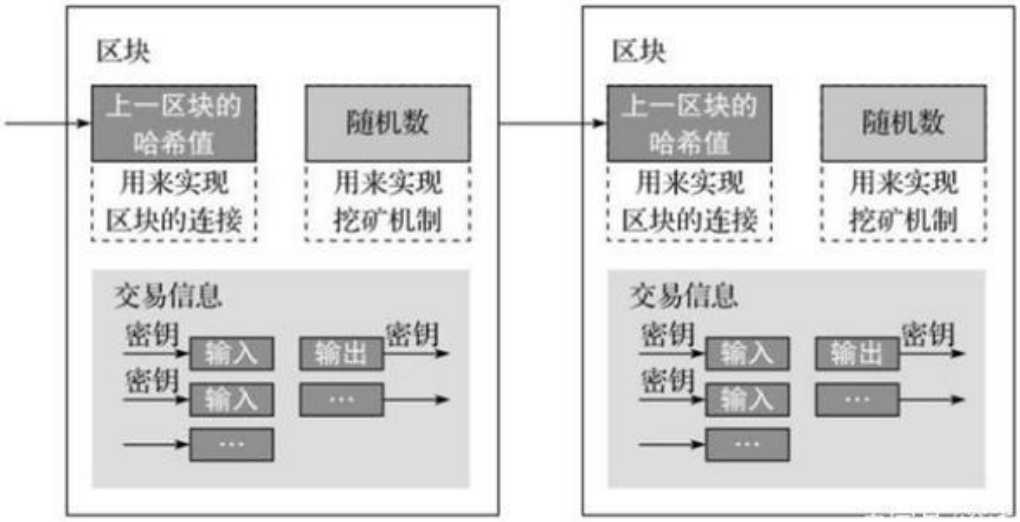


Merkle 树

Merkle 树的优点就是一个节点可以在不下载整个块的情况下，验证是否包含某笔交易，并且只需要一个交易哈希，一个 Merkle 树根哈希和一个 Merkle 路径。

2.2.4 链式结构

成熟的区块链系统（如比特币系统）大约每 10 分钟会创建一个区块，这个区块包含了这段时间里全网范围内发生的所有交易。每一个区块都保存了上一个区块的哈希值，使得每个区块都能找到其前一个区块，这样就将这些区块连接起来，形成了一个链式的结构。



在当前区块加入区块链后，所有节点就立即开始下一个区块的生成工作：（1）把在本地内存中的交易信息记录到区块主体中；（2）在区块主体中生成此区块中所有交易信息的 Merkle 树，把 Merkle 树根的值保存在区块头中；（3）把上一个刚刚生成的区块的区块头的数据通过 SHA256 算法生成一个哈希值填入到当前区块的父哈希值中；（4）把当前时间保存在时间戳字段中；（5）难度值字段会根据之前一段时间区块的平均生成时间进行调整，以应对整个网络不断变化的整体计算总量，如果计算总量增长了，则系统会调高数学题的难度值，使得预期完成下一个区块的时间依然在一定时间内。

2.3 分布式结构

2.3.1 点对点技术

点对点（P2P）技术又称对等互联网络技术，是一种网络新技术，依赖网络中参与者的计算能力和带宽，而不是把依赖都聚集在较少的几台服务器上。P2P 网络通常用于通过 Ad Hoc 连接来连接节点。

区块链系统采用 P2P 网络架构，整个系统没有中心化的硬件或者管理机构，任意节点之间的权利和义务都是均等的，每个节点通过多播实现节点识别和数据传播等功能。当发生交易或找到合法区块时，可以通过 P2P 网络发送给每一个节点。全网总账本是由全节点集体维护的，每个全节点都能获得一份完整数据 库的拷贝，单个节点篡改账本是不可能的，从而保证了区块链系统的安全性。

2.3.2 分布式账本技术

分布式账本是分布在多个节点或计算设备上的数据库。每个节点复制并保存一个相同的分类帐副本。网络的每个参与节点都独立地更新自己。

分布式分类账技术的突破性特性是，分类账不由任何中央机构维护。对分类账的更新由每个节点独立构建和记录。然后节点对这些更新进行投票，以确保大多数成员同意所达成的结论。对一份分类帐进行的投票和协议被称为共识，并通过一致算法自动执行。一旦达成一致意见，分布式分类账将自动更新，最新的、商定的分类账版本将分别保存在每个节点上。

分布式账本技术大大降低了信任成本，因此可以减轻人们对银行、政府、律师、公证人和监管合规官员的依赖，为信息的收集和传播提供了一种新的范式，并将彻底改变个人、企业和政府的交易方式。

区块链是分布式分类器技术的一种形式。不是所有的分布式账簿都使用一个区块链来提供安全有效的分布式共识。DLT 技术上是分散的，并且依赖于类似区块链的共识原则。但是，原则上，实施者对其实现方式拥有更高的控制权，可以决定支持其服务的网络的结构、目的和功能。因此，DLT 虽然在技术上是去中心化的，但其企业组织却未必是。而区块链不仅要求技术上和结构上是去中心化的，而且其组织和发展也是如此。

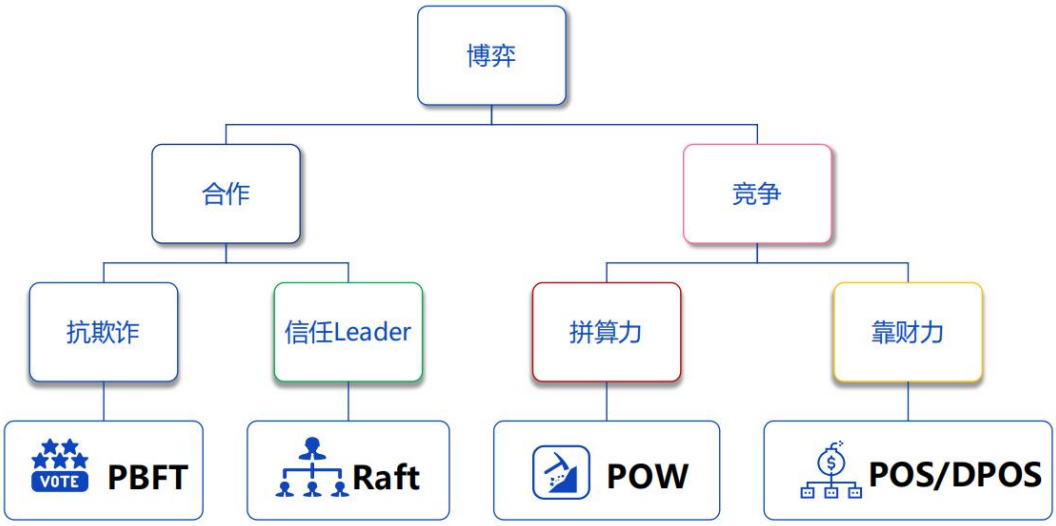
2.3.3 共识机制

共识机制是一种多方协作机制，用于协调多参与方达成共同接受的唯一结果，且保证此过程难以被欺骗，且持续稳定运行。

在区块链上，每个人都会有一份记录链上所有交易的账本，链上产生一笔新的交易时，每个人接收到这个信息的时间是不一样的，有些想要干坏事的人就有可能在这时发布一些错误的信息，这时就需要一个人把所有人接收到的信息进行验证，最后公布最正确的信息。

目前常用的几种共识机制：

- 工作量证明机制(Proof of Work - POW)是我们最熟知的一种共识机制。就如字面的解释，POW 就是工作越多，收益越大。这里的工作就是猜数字，谁能最快的猜出这个唯一的数字，谁就能做信息公示人。
- 权益证明机制(Proof of Stake - POS)也属于一种共识证明，它类似股权凭证和投票系统，因此也叫“股权证明算法”。由持有最多（token）的人来公示最终信息。
- 拜占庭共识算法（Practical Byzantine Fault Tolerance- PBFT）也是一种常见的共识证明。它与之前两种都不相同，PBFT 以计算为基础，也没有代币奖励。由链上所有人参与投票，少于（N-1）/3 个节点反对时就获得公示信息的权利。
- 授权股权证明（Delegated Proof of Stake - DPOS）。持有币的人可以进行投票选举，选举出一些节点做为代表来记账，类似于全国人民代表大会制度。
- Raft 算法通过选举一个领导人，然后给予其全部的管理复制日志的责任来实现一致性。领导人从客户端接收日志条目，把日志条目复制到其他服务器上，并且当保证安全性的时候告诉其他的服务器应用日志条目到他们的状态机中。



共识算法的博弈选择

三、区块链技术选型

根据应用场景和参与者的不同，可将区块链分成三类：公有链、联盟链和私有链。

3.1 公有链

3.1.1 全民参与

公有链（Public blockchain）中任何节点都向所有人开放，任何节点都可以自由加入和退出区块链系统，都可以发送和确认交易，都可以参与共识过程，没有中心化的机构，人人都可以参与到其中。

比特币、以太坊是当下最广为人知的公有链，它是全球所有节点、每个人都可以参与的区块链，也就是说公有链上的行为是公开的，而且不受任何人控制，是完全去中心化的区块链。

3.1.2 主要特点

- 访问门槛低
- 去中心化
- 用户匿名性
- 数据公开透明且无法篡改

3.2 私有链

3.2.1 小范围运行

私有链（Private blockchain）是完全私有的区块链，只有单一组织拥有写入权限，可以制定和修改区块链规则，信息一般不公开。在某些应用场景下，并不希望链被所有人参与，写入权限由某个公司或机构控制，只有被允许的节点才可以参与并查看所有数据，可用于协调企业内部各部门之间的工作。私有链最大的优势是加密审计和公开身份信息，可以起到监控作用，发生错误能够找到来源，机构或公司内部开发系统一般会采用私有链。蚂蚁金服就是应用私有链的显著代表。

3.2.2 主要特点

- 交易速度快
- 交易成本低
- 发生错误可以纠正
- 节点连接方便
- 隐私保护

3.3 联盟链

3.3.1 多方协作

联盟链（Consortium blockchain）介于公有链和私有链之间，由若干组织构成利益相关的联盟，约定区块链规则。联盟链中多个机构共同参与管理某个区块链，每个机构都运行着一个或多个节点，其中的数据只允许系统内不同的机构进行读写与发送交易，并且共同记录交易数据。其中每个节点的权限都完全对等，在不需要完全互信的情况下就可以实现数据的互换。节点的加入与退出需要联盟授权，只允许有限的、经过授权的节点参与共识过程。

联盟链的各个节点通常有与之对应的实体机构组织，通过授权后才能加入与退出网络。各机构组织组成利益相关的联盟，共同维护区块链的健康运转。目前比较有影响力的联盟链系统有 Corda、Fabric、FISCO BCOS 等。

3.3.2 主要特点

- 交易速度快
- 运行成本低
- 可多方合作
- 可信度高

四、区块链应用开发

4.1 智能合约

智能合约（Smart Contract）是一套以数字形式定义的承诺，承诺控制着数字资产并包含了合约参与者约定的权利和义务，由计算机系统自动执行。智能合约程序不只是一个可以自动执行的计算机程序，它本身就是一个系统参与者，对接收到的信息进行回应，可以接收和储存价值，也可以向外发送信息和价值。这个程序就像一个可以被信任的人，可以临时保管资产，总是按照事先的规则执行操作。

4.1.1 智能合约的思想

- 将现实世界的逻辑在区块链上实现
- 合约的内容和生命周期被共识确认，是大家认可的条款
- 在所有节点上保证逻辑的一致性
- 在所有节点上产生和维护一致的数据
- 合约还是有可能有 Bug 的
- “Code is Law”是个理想目标

4.1.2 图灵完备

在可计算性理论里，如果一系列操作数据的规则（如指令集、编程语言、细胞自动机）可以用来模拟单带图灵机，那么它是图灵完备的。这个词源于引入图灵机概念的数学家艾伦·图灵。图灵机是图灵为了研究可计算问题而构思的抽象计算模型——将人们传统的使用纸笔进行数学运算的过程，进行抽象，由一个虚拟的机器替代人们进行无数次的数学运算，也就是现代计算机的雏形。

图灵完备的语言特点：

- 具备强大的可编程能力
- 支持多种数据类型，int，string，map，array
- 支持判断，循环，跳转，分支，且可以应对停机问题
- 支持接口，继承等面向对象的特性

4.1.3 Solidity

Solidity 是一种面向对象的高级语言，用于实现智能合约，是当今最主要的智能合约开发语言。Solidity 是图灵完备的高级语言，支持循环、函数调用等用法。作为一门静态类型语言，Solidity 拥有丰富的数据类型，支持整形、字符串、数组、Map 等，同时也支持继承、库引用等高级用法。Solidity 拥有大量的参考实现以及广泛的开发者。

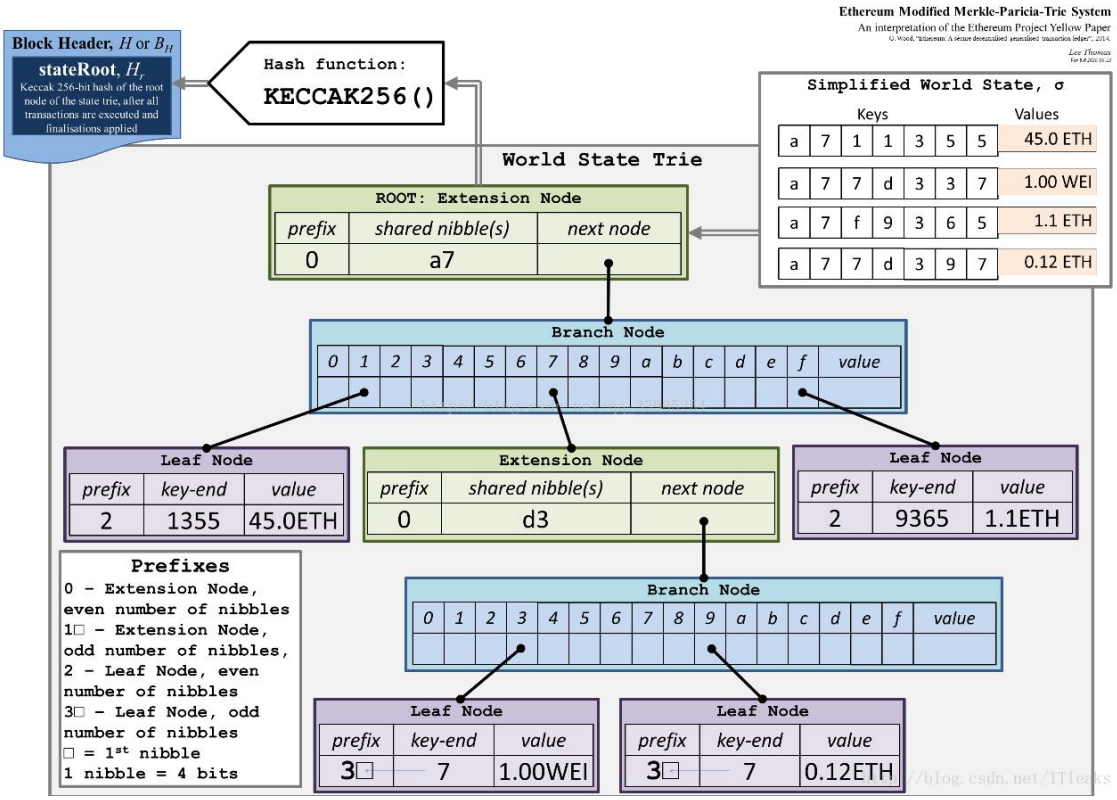
4.2 开发平台

4.2.1 以太坊

以太坊（ethereum）是一个全新开放的区块链平台，它允许任何人在平台中建立和使用通过区块链技术运行的去中心化应用。就像比特币一样，以太坊不受任何人控制，也不归任何人所有——它是一个开放源代码项目，由全球范围内的很多人共同创建。和比特币协议有所不同的是，以太坊的设计十分灵活，极具适应性。在以太坊平台上创立新的应用十分简便，随着 Homestead 的发布，任何人都可以安全地使用该平台上的应用。

4.2.2 MPT

以太坊中的世界状态是通过 MPT 树实现的，MPT 树是其在内存中的连接关系。为了持久化存储，MPT 树的连接关系转化为一系列的 K-V 对。以太坊将这些 K-V 对存储在 levelDB 中。 在以太坊中，使用了一种特殊的十六进制前缀(hex-prefix, HP)编码，所以在字母表中就有 16 个字符。这其中的一个字符为一个 nibble。



从前面结构图可以看出，Merkle Patricia Tree 有 4 种类型的节点：

- 叶子节点（leaf），表示为[key,value]的一个键值对。和前面的英文字母 key 不一样，这里的 key 都是 16 编码出来的字符串，每个字符只有 0-f 16 种，value 是 RLP 编码的数据
- 扩展节点（extension），也是[key，value]的一个键值对，但是这里的 value 是其他节点的 hash 值，通过 hash 链接到其他节点
- 分支节点（branch），因为 MPT 树中的 key 被编码成一种特殊的 16 进制的表示，再加上最后的 value，所以分支节点是一个长度为 17 的 list，前 16 个元素对应着 key 中的 16 个可能的十六进制字符，如果有一个[key,value]对在这个分支节点终止，最后一个元素代表一个值，即分支节点既可以搜索路径的终止也可以是路径的中间节点。分支节点的父亲必然是 extension node
- 空节点，代码中用 null 表示

MPT 的叶子节点（账户信息）是按照账户的地址（Address）的字典排序形成。也就是说，MPT 树上的一个叶子节点路径上的所有的 key 组成的是账户地址。在内存中，节点和节点的连接关系可以通过“指针”完成。为了将连接关系持久化，需要将节点内容生成对应的，唯一的“地址”。其他引用节点内容的节点，只需要记录引用节点的地址即可。以太坊中，节点内容的地址就是节点内容的 hash。比如一个由“key”以及一个“hashNode”组成的节点，节点内容是 rlp（key，hashNode），对应的地址是内容的 hash。在存储中，存储的是 hash 到 rlp 的 KV 对。其他引用节点的“连接”用 hash 值代替，从而隐性的实现“连接”。

因此，一个叶子节点的改变，将导致节点内容的改变，节点的“地址”也会发生变化。也就是说，父亲节点和该节点的“连接”地址发生变化。因为父亲节点的内容包括“连接”地址，所以父亲节点的内容也发生变化，父亲节点的“地址”也随着变化。随即，父亲的父亲也发生变化，一直变化到 MPT 的树根。

总的来说，一个账户的改变，导致 MPT 树，从叶子到树根，整条路径上的所有节点的 KV 对都发生变化，需要更新到存储中。

4.2.3 EVM

以太坊虚拟机（environment virtual machine，简称 EVM），作用是将智能合约代码编译成可在以太坊上执行的机器码，并提供智能合约的运行环境。它是一个对外完全隔离的沙盒环境，在运行期间不能访问网络、文件，即使不同合约之间也有有限的访问权限。

以太坊底层通过 EVM 模块支持合约的执行与调用，调用时根据合约地址获取到代码，生成环境后载入到 EVM 中运行。通常智能合约的开发流程是用 solidlity 编写逻辑代码，再通过编译器编译元数据，最后再发布到以太坊上。

4.2.4 账户

以太坊中有两类账户（它们共用同一个地址空间）：外部账户由公钥-私钥对（自然人）控制；合约账户由和账户一起存储的代码控制。外部账户的地址是由公钥决定的，而合约账户的地址是在创建该合约时确定的（这个地址通过合约创建者的地址和从该地址发出过的交易数量计算得到的，也就是所谓的“nonce”）。无论帐户是否存储代码，这两类账户对 EVM 来说是一样的。

每个账户都有一个键值对形式的持久化存储。其中 key 和 value 的长度都是 256 位，我们称之为存储。此外，每个账户有一个以太币余额（balance）（单位是“Wei”），余额会因为发送包含以太币的交易而改变。

4.2.5 交易

交易可以看作是从一个帐户发送到另一个帐户的消息。它能包含一个二进制数据（合约负载）和以太币。

如果目标账户含有代码，此代码会被执行，并以 payload 作为入参。如果目标账户是零账户（账户地址为 0），此交易将创建一个新合约。如前文所述，合约的地址不是零地址，而是通过合约创建者的地址和从该地址发出过的交易数量计算得到的（所谓的“nonce”）。这个用来创建合约的交易的 payload 会被转换为 EVM 字节码并执行。执行的输出将作为合约代码被永久存储。这意味着，为创建一个合约，用户不需要发送实际的合约代码，而是发送能够产生合约代码的代码。

4.2.6 Gas

一经创建，每笔交易都收取一定数量的 gas，目的是限制执行交易所需要的工作量和为交易支付手续费。EVM 执行交易时，gas 将按特定规则逐渐耗尽。

gas price 是交易发送者设置的一个值，发送者账户需要预付的手续费为 gas_price * gas。如果交易执行后还有剩余，gas 会原路返还。无论执行到什么位置，一旦 gas 被耗尽（比如降为负值），将会触发一个 out-of-gas 异常。当前调用帧（call frame）所做的所有状态修改都将被回滚。

这可以使以太坊区块链免受无关紧要或恶意的运算任务干扰，比如分布式拒绝服务（DDoS）攻击或无限循环 。交易的发送者必须在激活的“程序”每一步付款，包括运算和记忆储存。费用通过以太坊自有的有价代币，以太币的形式支付。

4.2.7 数据存储结构

每个账户有一块持久化内存区称为存储。存储是将 256 位字映射到 256 位字的键值存储区。在合约中枚举存储是不可能的，且读存储的相对开销很高，修改

存储的开销甚至更高。合约只能读写存储区内属于自己的部分。

第二个内存区称为内存，合约会试图为每一次消息调用获取一块被重新擦拭干净的内存实例。内存是线性的，可按字节级寻址，但读的长度被限制为 256 位，而写的长度可以是 8 位或 256 位。当访问（无论是读还是写）之前从未访问过的内存字（word）时（无论是偏移到该字内的任何位置），内存将按字进行扩展（每个字是 256 位）。扩容也将消耗一定的 gas。 随着内存使用量的增长，其费用也会增高（以平方级别）。

EVM 不是基于寄存器的，而是基于栈的，因此所有的计算都在一个被称为栈（stack）的区域执行。 栈最大有 1024 个元素，每个元素长度是一个字（256 位）。对栈的访问只限于其顶端，限制方式为：允许拷贝最顶端的 16 个元素中的一个到栈顶，或者是交换栈顶元素和下面 16 个元素中的一个。所有其他操作都只能取最顶的两个（或一个，或更多，取决于具体的操作）元素，运算后，把结果压入栈顶。当然可以把栈上的元素放到存储或内存中。但是无法只访问栈上指定深度的那个元素，除非先从栈顶移除其他元素。

4.3 DAPP

4.3.1 去中心化应用

DAPP 是去中心化应用（Decentralized Application）的简称。DApp 是一种互联网应用程序，与传统的 App 最大的区别是：DApp 运行在去中心化的网络上，也就是区块链网络中。网络中不存在中心化的节点可以完整的控制 DApp，而中心化的 App 则需要请求某台服务器来获取数据、处理数据。

4.3.2 DAPP 特点

- 开源
- 内部货币
- 去中心化的共识机制
- 无单点故障缺陷

DAPP 天然是分布式应用，因此避免了单点故障。区块链上的用户数据通常是用加密方式存储，数据的所有权归属用户，而非 DAPP 的开发者。DAPP 的后端程序是部署在区块链上的智能合约，智能合约是一组预定义的业务规则，具备确定性（Deterministic）执行的特征，能有效降低信任成本。DAPP 中消耗的资源由数字货币经济模型予以补偿或激励。

要 点

1. 区块链技术原理；
2. 联盟链和公有链的异同；
3. 链式存储和 MPT 存储
4. Gas 在智能合约中的作用；
5. EVM 中的数据存储结构；

总 结

随着区块链技术的进一步发展，其“去中心化”功能及“数据防伪” 功能在各个领域逐步受到重视。在法律、零售、物联、医疗等领域，区块链可以解决信任问题，不再依靠第三方来建立信用和信息共享，提高整个行业的运行效率和整体水平。区块链的应用范围将逐渐扩大到整个社会。

2016 年，国务院印发《“十三五”国家信息化规划》，区块链与大数据、人工智能、机器深度学习等新技术，成为了国家布局的重点。根据工信部发布的 2018 年区块链白皮书数据显示，截至 2018 年 3 月份，我国以提供区块链技术或服务为主营业务的公司已经达到 456 家，产业已初步形成规模。随着我国区块链技术的不断发展，区块链应用领域的不断拓展，未来国内区块链行业将呈现良好态势。

然而，区块链概念作为技术落地到具体的应用场景，从目前看来，和其他技术驱动的应用场景一样，并不能替代人的管理和运营能力而独立存在。其中有技术的门槛，更重要的是组织管理和知识储备的门槛。以电商的区块链场景应用为例，涉及物流、供应链、供货商、消费者售后服务、货币支付等，这些场景不仅与技术相连，更与人密切相关，然而区块链并不能解决人性的问题。此外，区块链的场景应用还涉及在成本上与原有模式相比是否存在竞争力的问题，如计算力成本过高则无法可持续。

参考文献

[1] 区块链技术原理，胡捷，http://www.360doc.com/content/18/0221/20/16022863_731276817.shtml

[2] 哈希函数，https://cloud.tencent.com/developer/news/326665

[3] DES 加密算法编程实现，华南理工大学软件学院，陈春华

[4] 通俗理解数字签名、数字证书和 https，https://www.jianshu.com/p/4932cb1499bf

[5] 数字证书及其应用，华南理工大学软件学院，陈春华

[6] 区块链记账原理，https://learnblockchain.cn/2017/10/25/whatbc/

[7] 剖析区块链，https://baijiahao.baidu.com/s?id=1613806947644891526&wfr=spider&for=pc

[8] 比特币之 Merkle 树，https://blog.csdn.net/suresand/article/details/79521905

[9] 区块链密码学基础，许可，华南理工大学软件学院，微众银行

[10] 区块链基础，许可，华南理工大学软件学院，微众银行

[11] 区块链理论研究进展，单进勇、高胜，密码学报，2018, 5(5)

[12] 区块链技术架构概要，微众银行

[13] 共识机制是什么，https://www.jianshu.com/p/a059c898acfb

[14] In Search of an Understandable Consensus Algorithm，Diego Ongaro and John Ousterhout, Stanford University

[15] Solidity 官方文档，https://solidity-cn.readthedocs.io/zh/develop/

[16] 以太坊官方文档，http://ethereum-homestead.readthedocs.io/en/latest/

[17] 以太坊 MPT 原理，https://blog.csdn.net/ITleaks/article/details/79992072

[18] 以太坊 MPT 存储，https://kuaibao.qq.com/s/20180829G1YTMR00?refer=cp_1026

[19] 智能合约和 DAPP，https://www.jianshu.com/p/5e7df3902957