区块链相关技术

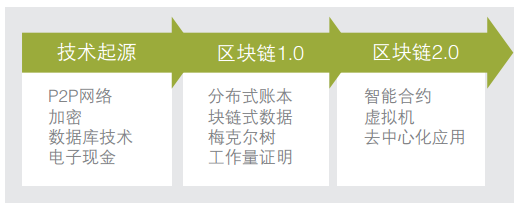
2.1 区块链的兴起

2.1.1 区块链的诞生

区块链技术起源于化名为“中本聪”（Satoshi Nakamoto）的学者在2008年发表的奠基性论文《比特币：一种点对点电子现金系统》。狭义来讲，区块链是一种按照时间顺序将数据区块一顺序相连的方式结合成的一种链式数据结构，并以密码学方式保证的不可篡改和不可伪造的分布式账本。广义来讲，区块链技术是利用块链式数据结构来验证与存储数据、利用分布式节点共识算法来生成和更新数据、利用密码学的方式保证数据传输和访问的安全、利用由自动化脚本代码组成的智能合约来编程和操作数据的一种全新的分布式基础架构与计算范式。

目前，区块链技术被很多大型机构称为是彻底改变业务乃至机构运作方式的重大突破性技术。同时，就像云计算、大数据、物联网等新一代信息技术一样，区块链技术并不是单一信息技术，而是依托于现有技术，加以独创性的组合及创新，从而实现以前未实现的功能。

至今为止，区块链技术大致经历了3个发展阶段，如图2-1所示。



2.1.2 区块链的设计思想

价值交互的基础是双方信任的建立。区块链技术的革命性在于它实现了一种全新的信任方式，通过在技术层面的设计创新，使得价值交互过程中人与人的信任关系能够转换为人与技术的信任，甚至于由程序自动化执行某些环节，商业活动得以更低成本的实现。

2.1.2.1 经济层面的设计思想

降低成本，是区块链技术的一个重要的设计思路。在区块链体系中，参与者可以不需要了解对方基本信息的情况进行交易，实现了“无需信任的信任”，改变了传统模式中以第三方为中心的信任模式。

这种设计模式有许多创新性，其中两项值得关注：

第一，交易信任由机器和算法确定。区块链通过构建一个信赖于机器和算法信任的交易体系，解决在匿名交易过程中的互相信任问题。所有参与者将在无需建立信任关系的环境中，通过密码学原理确定身份，依靠共识机制实现相互间的信任。

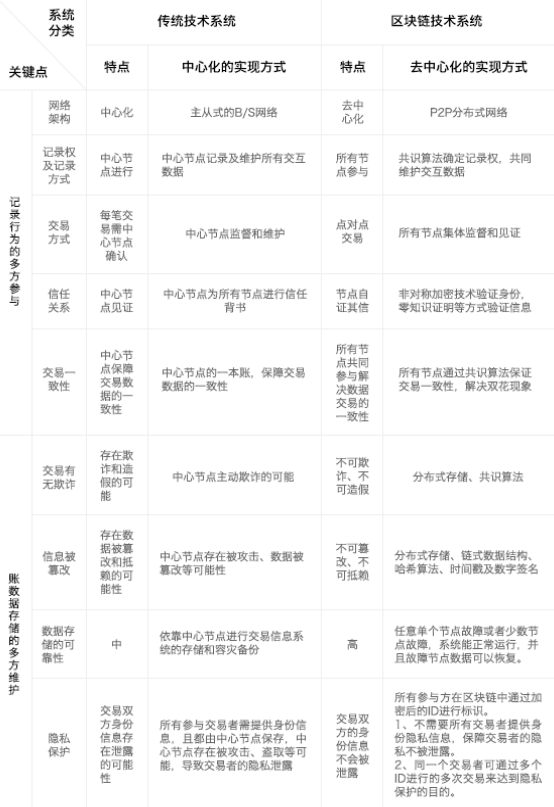
第二，交易过程可以由程序自动执行。区块链通过可编程的智能合约，自动执行双方所达成的契约，排除了认为的干扰因素，从制度上防止任何一方的抵赖。从而推动经济社会进入一种智能的状态，实现当前经济交易系统的质的飞跃。

基于区块链技术的“弱中心化”特征，现有的经济体系可以脱离当前通过制度约束或第三方机构背书，双方直接实现价值交付。这种“弱中心化”特征可以有效降低交易成本，提高交易效率，减少因交易一致性所引发的摩擦。

2.1.2.2 技术层面的设计思想

通俗的说，区块链可以看成是一套由多方参与的、可靠的分布式数据存储系统，其独特之处在于：一是记录行为的多方参与，即各方可参与记录；二是数据存储的多方参与、共同维护，即各方均参与数据的存储和维护；三是通过链式存储与合约，并且只能读取和写入，不可篡改。

在应用实践中，这种系统能够实现所有参与者信息共享、共识、共担，可以成为各种商业行为和组织机构的基础技术架构。具体与传统中心式系统对比如下表所示：



2.2 区块链的类型

根据区块链的开放程度，可以将区块链分为公有链、联盟链和私有链。但是随着区块链技术的快速发展，各种类型的链之间的界限慢慢也将变得模糊，特别是随着节点上所运行的智能合约所包含业务逻辑越来越复杂，私有链上的部分节点必须对外开放才能执行完整的业务逻辑，而部分共识及记账节点则会约束仅向许可节点开放保证效率和可控性，各种链之间的业务界限会逐渐模糊。三种链之间的对比，如表1.4所示。

表1.4 三种区块链对比

|  |  |  |  |
| --- | --- | --- | --- |
|  | **公有链** | **联盟链** | **纯私有链** |
| **参与者** | 任何人 | 授权的公司和组织 | 个体或一个公司内 |
| **记账人** | 任何人 | 参与者协调授权控制 | 自定 |
| **信任机制** | 工作量证明等 | 集体背书 | 自行背书 |
| **中心化程度** | 去中心化 | 多中心化 | 中心化 |
| **突出优势** | 信用的自建立 | 效率、成本优化 | 透明、可追溯 |
| **典型应用场景** | 比特币 | 清算 | 审计 |
| **承载能力** | 每秒3~20次 | 每秒100次以上 | 每秒1000次以上 |

2.3 区块链的特征

2.3.1 去中心化

去中心化是区块链最基本的特征，意味着区块链应用不依赖于中心化的机构，实现了数据的分布式记录、存储与更新。在传统的中心化网络中，业务运行高度依赖中心节点的稳健性与可信性，黑客若对单一的中心节点进行攻击即可破坏整个系统。而区块链的分布式架构中全网节点的权利和义务均等，系统中的数据是由全网节点共同基于密码学规则进行维护的，具有点对点、多冗余等特性，不存在单点失效的问题，因此其应对拒绝服务攻击的方式比中心化系统要灵活得多。即使一个节点失效，其他节点不受影响。

2.3.2 透明性

区块链系统的数据记录对全网节点是透明的，数据记录的更新操作对全网也是透明的，这是区块链系统值得信任的基础。由于区块链系统使用开源的程序、开放的规则和高参与度，区块链的数据记录和运行规则可以被全网节点审查、追溯，具有很高的透明度。

2.3.3 开放性

区块链的开放性是指，除数据直接相关各方的私有信息被加密外，区块链的所有数据对所有人公开（具有特殊权限要求的区块链系统除外）。任何人或参与节点都可以通过公开的接口查询区块链的数据记录或者开发相关应用，因此整个系统是开放的。

2.3.4 自治性

区块链采用基于协商一致的规范和协议，使整个系统中的所有节点能够在去信任的环境下自由安全地交换、记录以及更新数据，把对个人或机构的信任改成对体系的信任，任何人为的干预都将不起作用。

2.3.5 不可篡改性

区块链中有两套加密机制防止记录篡改，第一是采用默克尔树的方式加密交易记录，当底层数据发生改动时，必会导致默克尔树的根哈希值发生变化；第二是在创建新的区块时放入了前一区块的哈希值，这样区块之间形成链接关系，若想改动之前区块的交易数据，必须将该区块之前的所有区块的交易记录和哈希值进行重构，这是很难达到的，除非能够同时控制系统中超过51%的节点，否则单个节点上对区块中记录的修改是无效的，因此区块链的数据的稳定性和可靠性极高。

2.3.6 匿名性

区块链系统中虽然所有数据记录和更新操作过程都是对全网节点公开的，但其交易者的私有信息仍是通过哈希加密处理的，即数据交换和交易都是在匿名的情况下进行的。由于节点之间的数据交换遵循固定且预知的算法，因而其数据的交互无需双方存在相互信任的前提，可以通过双方地址而非身份的方式进行，因此交易双方无须通过公开身份的方式让对方产生信任。

2.4 区块链的核心技术

2.4.1 共识机制

共识机制是区块链技术的一个核心问题，它决定了区块链中区块的生成法则，保证了各节点的诚实性、账本的容错性和系统的稳健性。常用的共识机制主要有PoW、PoS、DPoS、PBFT等。基于区块链技术的不同应用场景，以及各种共识机制的特性，主要可以从性能效率、资源消耗、容错性、监管水平等几个方面进行评价和比较。

1）PoW

PoW的定义简单来说就是工作端提交已知难于计算但易于验证的计算结果，而其他任何人都能够通过验证这个答案就确信工作端为了求得结果已经完成大量的计算工作。PoW工作量证明的主要特征是计算的不对称性。根据机器的运算资源来分配记账权，由于参与运算的不同节点根据自身的运算资源获取记账权，所以这些节点在竞争结束前都要一直进行哈希运算，资源消耗较高。而众多参与节点中最终只会产生一名记账者，性能效率比较低。其典型应用为比特币。

2）PoS

PoS指的是所有权证明，节点通过拥有的所有权的证明获得产生新区块的权利。根据节点持有的所有权的数量和时间来等比例地降低挖矿难度，这样节点记账权的获得难度与节点持有的权益成反比，与PoW所有机器同等挖矿难度相比，这种方法在一定程度减少了数学运算难度，各节点资源消耗减少，性能也得到了一些提升。但由于在挖矿时仍是基于哈希运算竞争的方式，可监管性弱，共识机制容错性也和PoW相同。典型应用为点点币（Peercoin）。

3）DPoS

PoW与PoS机制都能有效的解决记账行为的一致性共识问题，但PoW中拥有巨大算力的一方容易成为另一个中心，而PoS机制中会产生所有权比例越大的账户拥有的权力更大。DPoS机制致力于解决PoW机制和PoS机制的这类不足。DPoS机制中，可由区块链网络主体投票产生N个见证人来对区块进行签名，其根本特性是权益所有者保留了控制权从而使系统去中心化。通过信任少量诚信节点减少了确认要求，提高了交易速度，因此，性能、资源消耗都要优于PoS。其合规监管、容错性与PoS相似。其典型应用为比特股（BitShares）。

4）PBFT

PBFT的原理是基于异步网络环境下的状态机副本复制协议，与一般公有链的共识机制主要基于经济博弈原理不同。在PBFT算法中，不同的节点之间通过消息交换尝试达成共识，也是一种采用许可投票、少数服从多数来选举领导者进行记账的共识机制，但不同之处在于该共识机制允许拜占庭容错，本质上是数学算法实现了公式，因此区块的确认不需要像公有链一样在若干区块之后才安全，可以实现出块即确认。同时该共识机制允许强监管节点参与，具备权限分级能力，性能更高，耗能更低，该算法每轮记账都会由全网节点共同选举领导者，允许33%的节点作恶，容错性为33%。其典型应用为超级账本项目（Hyperledger）。

2.4.2 智能合约

智能合约（Smart Contract）由尼克·萨博（Nick Szabo）于1995年提出，他给出的定义是：“一个智能合约是一套以数字形式定义的承诺，包括合约参与方可以在上面执行这些承诺的协议。”

区块链中的智能合约可视作一段部署在区块链上由事件驱动，具有状态的，获得多方承认的，可自动运行、无须人工干预，且能够根据预设条件自动处理资产的程序，从本质上讲，智能合约的工作原理类似于计算机程序中的if-then语句，当一个预先编好的条件被触发时，智能合约执行便相应的条款程序。由于智能合约运行在图灵完备的虚拟机上，因此智能合约的具体条款可以根据应用场景由开发人员编写，其具体的技术细节又包括编程语言、编译器、虚拟机、事件、状态机、容错机制等。但由于智能合约本质上是一段程序，存在出错的可能性，因此需要做好充分的容错机制，通过系统化的手段，结合运行环境隔离，确保合约的正确执行。

2.4.3 安全技术

1）哈希算法

哈希算法也叫数据摘要或者散列算法，其原理是将一段信息映射成一个固定长度的二进制值，该二进制值称为哈希值。哈希值具有以下特点：

（1）若某两段信息相同，则他们经过哈希运算得到的哈希值也相同。

（2）若两段信息不同，即使只是相差一个字符，他们产生的哈希值也会不同且杂乱无章毫无关联。

要找到哈希值为同一值得两个不同输入，在计算上是不可能的，因此数据的哈希值可以被用以检验数据的完整性，可以把给定数据的哈希值理解为该数据的“指纹信息”。本质上，散列算法的目的不是为了“加密”而是为了抽取“数据特征”。

典型的哈希算法有MD5（Message Digest Algorithm，消息摘要算法第五版）、SHA1/SHA256和SM3（SM3密码杂凑算法）等，各算法特点的对比，如表1.5所示。

目前区块链主要使用SHA256，国内某些特定业务场景使用国密SM3，亦是比较符合国家安全和监管的选择。SHA256和SM3这两种算法的效率和安全性大致相当，但由于不同业务场景的安全性标准有别，未来不排除仍需探索更优算法的可能性。

表1.5典型哈希算法的特点

|  |  |  |  |
| --- | --- | --- | --- |
| **加密算法** | **安全性** | **运算速度** | **输出大小（位）** |
| **MD5** | 低 | 快 | 128 |
| **SHA1** | 中 | 中 | 160 |
| **SHA256** | 高 | 比SHA1略低 | 256 |
| **SM3** | 高 | 比SHA1略低 | 256 |

2）非对称加密算法

非对称加密算法是区块链基础技术之一，区块链中使用非对称加密的公私钥对来构建节点间信任。非对称加密算法由对应的一对唯一的密钥（即公开密钥和私有密钥）组成，任何获悉用户公钥的人都可用用户的公钥对信息进行加密与用户实现安全信息交互。由于公钥与私钥之间存在依存关系，只有持有私钥的用户本身才能解密该信息，任何未经授权的用户甚至信息的发送者都无法将此信息解密。

常用的非对称加密算法主要有RSA、ECC以及SM3，其特点比较如表1.6所示。具体算法的技术细节将在本书后续章节进行讨论。

表1.6 常用非对称加密算法的特点

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **加密算法** | **成熟度** | **安全性** | **运算速度** | **资源消耗** |
| **RSA** | 高 | 低 | 慢 | 高 |
| **ECC** | 高 | 高 | 中 | 中 |
| **SM3** | 高 | 高 | 中 | 中 |

2.4.4 分布式存储

1）区块数据结构

在区块链中，数据以区块的方式永久储存。区块链的时间戳解决了区块的排序问题，新区块生成时便记录着上一个区块通过哈希计算得到的哈希值，实现了区块密码学链接。每一个区块记录了其创建期间发生的所有交易信息。区块的数据结构一般分为区块头和区块体，其中，区块头部结构记录了版本号、前一个区块的哈希值、默克尔树的根值、时间戳、目标特征值和随机数值；区块体部分则包含了经过验证的、区块创建过程中产生的所有交易信息。区块主标识符是它的加密哈希值，一个通过SHA256算法对区块头进行二次哈希计算而得到的数字指纹，产生的32字节哈希值被称为区块头哈希值。第二种识别区块的方式是按照该区块在区块链中的位置，即“区块高度”，如第一个区块，其区块高度为0。区块链的数据结构，如图1.9所示。

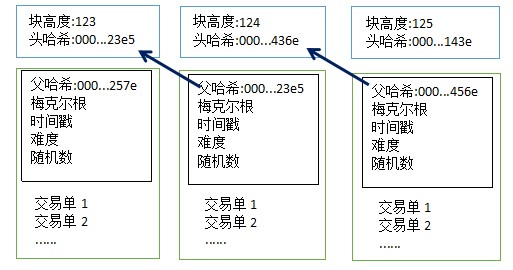


图1.9 区块链数据结构

2）数据库

区块链中关系型和非关系型两种数据库均可采用。其中，关系型数据库采用关系模型来组织数据，支持各种SQL（Structured Query Language，结构化查询语言）功能，功能性强，支持事务性，读写性能一般，可扩展性弱，在数据存在海量并发情况下表现较差。而非关系型数据库中键值对数据库的数据结构组织形式简单，读写性能很高，支持海量并发读写请求，可扩展性强，操作接口简单，支持一些基本的读、写、修改、删除等功能，但不支持复杂的SQL功能和事务。

根据数据库的部署形式，可分为单机型和分布式两种。其中，单机型数据库保证强一致性和较好的可用性。分布式数据库在物理部署上遵循了分布式架构，能提供高并发的读写性能和容错，有很强的可用性和分区容错性，但由于需要进行数据同步，分布式架构的数据一致性较弱，只能保证最终一致性。

在区块链中，如果待存储的是一些字符串、json对象，可以使用扩展账本结构链存储；如果是图片、视频等较大的多媒体文件，可以将文件的哈希值存储在链上，而原文件可使用云存储存储到云端。

2.5 主流区块链技术介绍

2.5.1 比特币

比特币（BitCoin）是基于区块链技术的一种数字货币实现，比特币网络是历史上首个经过大规模、长时间验证的数字货币系统。

2.5.1.1 重要概念

1. 账户/地址

比特币采用了非对称的加密算法，用户自己保留私钥，对自己发出的交易进行签名请确认，并公开公钥。

比特币的账户地址其实就是用户公钥经过一系列Hash（Hash160，或先进行SHA256，然后进行RIPEMD160）及编码运算后生成160位（20字节）的字符串。

一般地，对账户地址串进行Base58Check编码，并添加前导字节（表明支持哪种脚本）和4字节校验字节，以提高可读性和准确性。

1. 交易

交易是完成比特币功能的核心概念，一条交易可能包括如下信息：

* 付款人地址：合法的地址，公钥经过SHA256和RIPEMD160两次Hash，得到160位Hash串。
* 付款人对交易的签名确认：确保交易内容不被篡改。
* 付款人资金的来源交易ID：哪个交易的输出作为本次交易的输入。
* 交易金额：多少钱，与输入的差额为交易的服务费。
* 收款人地址：合法的地址。
* 收款人公钥：收款人的公钥。
* 时间戳：交易何时能生效。

网络中节点收到交易信息后，将进行如下检查：

* 交易是否已经处理过。
* 交易是否合法，包括地址是否合法、发起交易者是否输入地址的合法拥有者、是否是UTXO。
* 交易的输入之和是否大于输入之和。

如果检查都通过，则将交易标记为合法的未确认交易，并在网络内进行广播。

图xxx 比特币交易的例子

1. 交易脚本

脚本（script）是保障交易完成（主要用于检验交易是否合法）的核心机制，当所依附的交易发生时被触发。通过脚本机制而非写死交易过程，比特币网络实现了一定的可扩展性。比特币脚本语言是一种非图灵完备的语言，类似于Forth语言。

一般每个交易都会包括两个脚本：输入脚本（scriptPubKey）和认领脚本（scriptSig）。输出脚本一般由付款方对交易设置锁定，用来对能动用这笔交易输出（例如，要花费交易的输出）的对象（收款方）进行权限控制，例如限制必须是某个公钥的拥有者才能花费这笔交易。认领脚本则用来证明自己可以满足交易输出脚本的锁定条件，即对某个交易的输出（比特币）的拥有权。

1. 区块

比特币区块链的一个区块主要包括如下内容：

|  |  |  |
| --- | --- | --- |
| 字段名称 | | 字段说明 |
| 神奇数 | | 神奇数是一个等于0xD9B4BEF9的常数，用于区分区块间的间隔，长度为4字节。 |
| 区块大小 | | 区块大小是指从区块开始到区块结束的字节数，长度为4字节。 |
| 区块头部信息 | 版本号 | 版本号是区块的版本号，长度为4字节。 |
| 前一个区块地址 | 前一个区块HASH值，长度为32字节。 |
| Merkle树根节点 | 记录了当前区块中所有交易Merkle树的根节点的HASH值，长度为32字节。 |
| 时间戳 | 记录了当前区块生成的时间，按照UNIX时间格式，长度为4字节。 |
| 目标值 | 当前区块生成所达成目标值的特征，用于挖矿时的工作量证明和比特币的“发行”，长度为4字节。 |
| 随机数 | 当前区块工作量证明的参数，长度为4字节。 |
| 交易计数 | | 当前区块所记录的交易数，长度为1至9字节。 |
| 交易详情 | | 记录了当前区块保存的所有交易细节，长度不确定。 |

可见，要对区块链的完整性进行检查，只需要检查各个区块头信息即可，无需获取具体的交易内容，这也是简单交易验证（simple payment verification,SPV）的基本原理。另外，通过头部的链接，提供时序关系的同时加大了对区块中数据进行篡改的难度。

2.5.1.2 基本交易过程

每次发生交易，用户需要将新交易记录写到比特币区块链网络中，等网络确认后即可认为交易完成。每个交易包括一些输入和一些输出，未经使用的交易的输出（unspent transaction output,UTXO）可以被新的交易引用作为合法的输入，被使用过的交易的输出（spent transaction output,STO）则无法被引用作为合法输入。

一笔合法的交易，即引用某些已存在的交易的UTXO作为交易的输入，并生成新的输入的过程。

在交易过程中，转账方需要通过签名脚本来证明自己是UTXO的合法使用者，并且指定输出脚本来限制未来本交易的使用者（为收款方）。对每笔交易，转款方需要进行签名确认。并且，对每一笔交易来说，总输入不能小于总输出。总输入相比总输出多余的部分称为交易费用（transac fee），为生成包含该交易区块的矿工所获得。目前规定每笔交易的交易费用不能小于0.0001BTC，交易费用越高，越多矿工愿意包含该交易，也就越早被放到网络中。交易费用在奖励矿工的同时，也避免了网络受到大量攻击。

2.5.1.3 创新设计

1. 避免作恶

避免作恶基于经济博弈原理。在一个开放的网络中，无法通过技术手段来保证每个人都是合作的。单可以通过经济博弈来让合作者得到利益，让非合作者遭受损失和风险。

比特币网络中所有试图参与者（矿工）都首先要付出挖矿的代价，进行算力小号，越想拿到新区块的决定权，意味着抵押的算力越多。一旦失败，这些算力都会被没收掉，称为沉没成本。当网络中存在众多参与者时，个体试图拿到新区块决定权要付出的算力成本是巨大的，意味着进行一次作恶付出的代价已经超过可能带来的好处。

1. 负反馈调节

比特币网络中矿工越多，系统就越稳定，比特币价值就越高，但挖到矿的概率会越低。反之，网络中矿工减少，会让系统更容易被攻击，比特币价值降低，但挖到矿的概率提高。

因此，比特币的价格理论上应该稳定在一个合适的值（网络稳定性也会稳定在响应的值），这个价格乘以挖到矿的概率，恰好达到矿工的收益预期。

从长远角度看，硬件成本是下降的，但每个区块的比特币奖励每隔4年减半，最终将在2140年达到2100万枚，之后将完全依靠交易的服务费来奖励矿工对网络的维护。

1. 共识机制

传统共识问题往往是考虑在一个相对封闭的分布式系统中，允许同时存在正常节点、故障如何快速达成一致。

对于比特币网络来说，它是完全开放的，可能面对各种攻击情况，同时基于Internet的网络质量只能保证“尽力而为”，导致问题更为复杂，传统的一致性算法在这种场景下难以使用。

因此，比特币网络不得不对共识的目标和过程进行一系列限制，提出了基于proof of work的共识机制。

首先是不实现面向最终确认的共识，而是基于概率、随时间逐步增强确认的共识。现有达成的结果在理论上可能被推翻，只是攻击者要付出的代价随时间而指数级上升，被推翻的可能性随之指数级下降。

此外，考虑到Internet的尺度，达成共识的时间相对比较长，因此按照区块来进行阶段性的确认（快照），从而提高网络整体的可用性。

最后，限制网络中共识的噪音。通过进行大量的Hash计算和少数的合法结果来限制合法提案的个数，进一步提高网络中共识的稳定性。

2.5.2 以太坊

作为公有区块链平台，以太坊将比特币针对数字货币交易的功能进一步进行了拓展，面向更为复杂和灵活的应用场景，支持了智能合约这一重要特性。

从此，区块链技术的应用场景，从单一基于UTXO的数字货币交易，延伸到图灵完备的通用计算领域。用户不在受限于仅能使用比特币脚本所支持的简单逻辑，而是可以自行设计任意复杂的合约逻辑。这就为构建各种多样化的上层应用开启了大门，意义重大。

2.5.2.1 主要特点

以太坊区块链底层也是一个类似比特币网络的P2P网络平台，智能合约运行在网络中的以太坊虚拟机里。网络自身是公开可接入的，任何人都可以接入并参与网络中数据的维护，提供运行以太坊虚拟机的资源。

与比特币项目相比，以太坊区块链的技术特点主要包括：

* 支持图灵完备的智能合约，设计了变成语言Solidity和虚拟机EVM。
* 选用了内存需求较高的哈希函数，避免出现强算力矿机、矿池攻击。
* 叔块激励机制，降低矿池的优势，并减少了区块产生间隔（10分钟降低到15秒左右）。
* 采用账户系统和世界状态，而不是UTXO，容易支持更复杂的逻辑。
* 通过Gas限制代码执行指令数，避免循环执行攻击。
* 支持PoW共识算法，并计划支持效率更高的PoS算法。

2.5.2.2 重要概念

1. 智能合约

智能合约是以太坊中最为重要的一个概念，即以计算机程序的方式来缔结和运行各种合约。最早在上世纪90年代，Nick Szabo等人就提出过类似的概念，但一直因为缺乏可靠执行智能合约的环境，而被当做一种理论设计。区块链技术的出现，恰好补充了这一缺陷。

以太坊支持通过图灵完备的高级语言（包括Solidity、Serpent、Viper）等来开发智能合约。智能合约作为运行在EVM中的应用，可以接受来自外部的交易请求和事件，通过触发运行提前编写好的代码逻辑，进一步生成新的交易和事件，可以进一步调用其他智能合约。

智能合约的执行结果可能对以太坊网络上的账本状态进行更新。这些修改由于通过了以太坊网络中的共识，一旦确认后无法被伪造和篡改。

1. 账户

相对于比特币采用了UTXO模型记录整个系统的状态，任何人都可以通过交易历史来推算出用户的余额信息。而以太坊直接用账户来记录系统状态。每个账户余额信息、智能合约代码和内部数据存储等。以太坊支持在不同账户之间转移数据，以实现更为复杂的逻辑。

具体来看，以太坊账户分为两种类型：合约账户（contracts accounts）和外部账户（externally owned accounts,EOA）:

* 合约账户：存储执行的智能合约代码，只能被外部账户来调用激活。
* 外部账户：以太币拥有者账户，对应到某公钥。

当合约账户被调用时，存储其中的智能合约会在矿工处的虚拟机中自动执行，并消耗一定的燃料。燃料通过外部账户中的以太币进行购买。

1. 交易

交易在以太坊中是指从一个账户到另一个账户的消息数据。消息数据可以是以太币或者合约执行参数。

以太坊采用交易作为执行操作的最小单位。每个交易包括如下字段：

* to：目标账户地址。
* value：可以指定转移的以太币数量。
* nonce：交易相关的字串。
* gasPrice：执行交易需要消耗的Gas价格。
* startgas：交易消耗的最大gas值。
* signature：签名信息。

类似于比特币网络，在发送交易时，用户需要交纳一定的交易费用，通过以太币方式进行支付和消耗。

1. 以太币

以太币（Ether）是以太坊网络中的货币。

以太币主要用于购买燃料，支付给矿工，以维护以太坊网络运行智能合约的费用。以太币最小单位是wei，一个以太币等于10^18个wei。

以太币同样可以通过挖矿来生成，成功生成新区快的以太坊矿工可以获得5个以太币的奖励，以及包含在区块内交易的燃料费用。

1. 燃料

燃料（gas），控制某次交易执行指令的上限。每执行一条合约指令会消耗固定的燃料，当某个交易还未执行结束，而燃料消耗完时，合约执行终止并回滚状态。

gas可以跟以太币进行兑换。需要注意的是，以太币的价格是波动的，单运行某段智能合约的燃料费用是固定的，通过设定gas价格等进行调节。

2.5.2.3 创新设计

1. 智能合约

以太坊采用EVM作为智能合约的运行环境。EVM是一个隔离的轻量级虚拟机环境，运行在其中的智能合约代码无法访问本地网络、文件系统或其他进程。

对同一个智能合约来说，往往需要在多个EVM中同时运行多份，以确保整个区块链数据的一致性和高度的容错性。

智能合约编写完毕后，用编译器编译为EVM专用的二进制格式（EVM bytecode），由客户端上传到区块链中，之后在矿工的EVM中执行。

1. 共识

以太坊目前采用了基于成熟的PoW共识的变种算法Ethash协议作为共识机制。

为了防止ASIC矿机矿池的算力攻击，跟原始PoW的计算密集型Hash运算不同，Ethash在执行时候需要消耗大量内存，反而跟计算效率关系不大。这意味着很难制造出专门针对Ethash的新票。

以太坊有计划在未来采用更搞笑的PoS作为共识机制。相对于PoW机制来讲，PoS机制无需消耗大量无用Hash计算，但其共识过程的复杂度要更高。

1. 降低攻击

由于以太坊网络中的交易更加多样化，也就更容易受到攻击。

以太坊网络在降低攻击方面的核心设计思想仍是通过经济激励机制防止少数作恶：

* 所有交易都要提供交易费用，避免DDoS攻击；
* 程序运行指令数通过gas来限制，所消耗的费用超过设定上限时就会被取消，避免出现恶意合约。

这就确保了攻击者试图消耗网络中虚拟机的计算资源时，需要付出经济代价；同时难以通过构造恶意的循环或不稳定合约代码来对网络造成破坏。

2.5.3 超级账本

超级账本（hyperledger）是Linux基金会于2015年发起的首个面向企业应用场景的开源分布式账本平台。

如果说以比特币为代表的数字货币提供了区块链技术应用的原型，以太坊为代表的智能合约平台延伸了区块链技术的功能，那么进一步引入权限控制和安全保障的超级账本项目则开拓了区块链技术的全新领域。超级账本首次将区块链技术引入到了分布式联盟账本的应用场景，这就为未来基于区块链技术打造高效率的商业网络打下了坚实的基础。

2.5.3.1 核心特性

超级账本Fabric架构的核心特性主要包括：

* 解耦了原子排序环节与其他复杂处理环节，消除了网络处理瓶颈，提高可扩展性。
* 解耦交易处理节点的逻辑角色为背书节点、确认节点，可以根据负载进行灵活部署。
* 加强了身份证书管理服务，作为单独的Fabric CA项目，提供更多功能。
* 支持多通道特性，不同通道之间的数据彼此隔离，提高隔离安全性。
* 支持可插拔的架构，包括共识、权限管理、加解密、账本机制等模块，支持多种类型。
* 引入系统链码来实现区块链系统的处理，支持可编程和第三方实现。

2.5.3.2 重要概念

1. 节点

节点（Peer）的概念最早来自于P2P分布式网络，意味着在网络中担任一定职能的服务或软件。节点功能可能是对等一致的，也可能是分工合作的。

在超级账本Fabric网络中，Peer意味着在网络中负责接收交易请求、维护一致账本的各个fabric-peer实例。这些实例可能运行在裸机、虚拟机甚至容器中。节点之间彼此通过gRPC消息进行通信。

按照功能角色划分，Peer可以包括三种类型：

* 背书节点（endorser）：负责对来自客户端的交易提案进行检查和背书。
* 确认节点（committer）：负责检查交易请求，执行交易并维护区块链和账本结构。
* 提交节点（submitter）：负责接收交易，转发给排序者。

1. 排序者

排序者（orderer）也称为排序节点，负责对所收到的交易在网络中进行全局排序。

Orderer主要提供了Broadcast和Deliver两个接口。前者代表客户端将数据（交易）发送给Orderer，后者代表从Orderer获取排序后构造的区块结构。

1. 成员身份管理

CA节点负责对Fabric网络中的成员身份进行管理。

Fabric网络目前采用数字证书机制来实现对身份的鉴别和权限控制，CA节点则实现了PKI服务，主要负责对身份证书进行管理，包括生成、撤销等。

2.5.3.3 整体架构

超级账本Fabric的整体框架如图XX所示

其中账本是最核心的结构，负责记录应用信息，应用则通过发起交易来向账本中记录数据。交易执行的逻辑通过链码来承载。整个网络运行中发生的时间可以被应用访问，以触发外部流程甚至其他系统。权限管理则负责整个过程中的访问控制。

账本和交易进一步地依赖核心的区块链结构、数据库、共识机制等技术；链码则依赖容器、状态机等技术；权限管理利用已有的KPI体系、数字证书、加解密算法等诸多安全技术。

底层由多个节点组成P2P网络，通过gRPC通道进行交互，利用Gossip协议进行同步。

2.5.3.4 典型工作流程

典型的交易处理过程示例如图XX所示

在整个交易过程中，各个组件的主要功能如下：

* 客户端（App）：客户端应用使用SDK来跟Fabric网络打交道。首先，客户端从CA获取合法的身份证书来加入网络内的应用通道。发起正式交易前，需要先构造交易提案提交给Endorser进行背书，客户端收集到足够的背书支持后可以利用背书构造一个合法的交易请求，发给Orderer进行排序处理。
* Endorser节点：主要提供ProcessProposal方法供客户端调用，完成对交易提案的背书处理。收到来自客户端的交易提案后，首先进行合法性和ACL权限检查，检查通过则模拟运行交易，对交易导致的状态变化（以读写集形式记录，包括所读状态的键和版本，所写状态的键值）进行背书并返回结果给客户端。
* Committer节点：负责维护区块链和账本结构（包括状态DB、历史DB、索引DB等）。该节点会定期地从Orderer获取排序后的批量交易区块结构，对这些交易进行落盘前的最终检查（包括交易消息结构、签名完整性、是否重复、读写集合版本是否匹配等）。检查通过后执行合法的交易，将结果写入账本，同时构造新的区块，更新区块中的BlockMetadata[2]（TRANSACTIONS\_FILTER）记录交易是否合法等信息。同一个物理节点可以仅作为Committer角色运行，也可以同时担任Endorser和Committer这两种角色。
* Orderer：仅负责排序。为网络中所有合法交易进行全局排序，并将一批排序后的交易组合生成区块结构。
* CA：负责网络中所有证书的管理（分发、撤销等），实现标准的PKI架构。