 

**课程报告**

|  |  |
| --- | --- |
| **课程名称：** | **企业软件项目实训** |
| **学生姓名：** | **李嘉晨** |
| **学生学号：** | **201630664758** |
| **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**

1. **区块链技术原理；**
   1. **共识机制与共识算法**

在加密货币领域中，共识算法是每个区块链网络的关键要素，它们负责维护这些分布式系统的完整性和安全性。第一个被创建出的加密货币共识算法是[工作量证明](https://www.binance.vision/zh/blockchain/proof-of-work-explained)（PoW），它由中本聪设计并在[比特币](https://info.binance.com/en/currencies/bitcoin)上实现，用于实现拜占庭容错。

共识算法可以被定义为使区块链网络达成共识的机制。公共（去中心化的）区块链是作为分布式系统来构建的，由于它们不依赖于一个中央权威，因此分散的节点需要就交易的有效与否达成一致。这就是共识算法发挥作用的地方，确保所有节点都遵守协议规则并保证所有交易都以可靠的方式进行，即每个货币都只能花一次。

在深入研究不同类型的共识算法之前，先了解算法和协议之间的差异非常重要。

* + 1. **共识算法与协议**

算法和协议作为术语经常会被互换使用，但它们并非同一回事。简单来说，我们可以将协议定义为区块链的主要规则，而将算法定义为遵循这些规则的机制。

除了金融系统中的广泛应用，区块链技术还可以应用于各种各样的业务，并且可以适用于不同的用例。但无论什么情景，区块链网络都将建立在一个协议之上，该协议将定义系统应该如何工作，因此系统的所有不同部分和网络的所有参与者都需要遵循规则。

虽然协议确定了规则是什么，但算法告诉系统采取哪些步骤来遵守这些规则并最终产生所需的结果。例如，区块链的共识算法决定了交易和区块的有效性。因此，比特币和[以太坊](https://info.binance.com/en/currencies/ethereum)是协议，而工作量证明和[权益证明](https://www.binance.vision/blockchain/proof-of-stake-explained)是它们的共识算法。

进一步来说，比特币协议定义了节点间应如何交互，数据应如何在它们之间传输，以及区块验证成功的要求是什么。另一方面，公式算法负责负责验证余额和签名，确认交易，以及实际执行区块验证 - 所有这些都取决于网络共识。

* + 1. **不同类型的共识算法**

有几种类型的共识算法。最常见的是工作量证明（PoW）和权益证明（PoS）。在尝试平衡安全性与功能性和可扩展性时，每个方案都有各自的优缺点。

* + - 1. **工作量证明（PoW）**

工作量证明是第一个被建立的共识算法。它被比特币和许多其他加密货币所采用。工作量证明算法是[挖矿过程](https://www.binance.vision/zh/blockchain/what-is-cryptocurrency-mining)的重要组成部分工作量证明挖矿涉及到大量哈希（算力）的（挖矿）尝试，因此更多的算力意味着每秒更多的尝试。换句话说，哈希率的矿工有更多机会找到下一个区块（也叫做块哈希）的解。如果网络的分布式节点达成共识并且确认矿工提供的块哈希是有效的工作证明，则工作量证明共识算法确保该矿工仅能够验新区块里的交易并将其添加到区块链。

* + - 1. **权益证明（PoS）**

作为工作量证明的替代方案，权益证明共识算法在2011年被提出。尽管工作量证明和权益证明有着相同的目标，但它们存在根本的差异性特殊性。特别是验证新块的过程。

简单来说，权益证明共识算法用一种机制来取代工作量证明的挖矿，期中区块根据参与者的质押的币而验证。每个区块的验证者（也称为铸造者或者铸币者）由本身的加密货币投资决定，而不是分配的算力数量。每个权益证明系统可以以不同的方式来实现该算法，但是通常来说，区块链由随机的选举过程保护，该过程考虑了节点的财富、币龄（币被锁定或者是质押的时间）以及随机因素。

以太坊区块链目前基于工作量共识算法，但是最终Casper协议将被推出以将网络从工作量共识切换到权益共识，以尝试增加网络的可扩展性。

* + 1. **为什么共识算法对于加密货币至关重要**

就如同前面所说，共识算法对于维护加密货币网络的完整性和安全性至关重要。它们提供了一种分布式节点就正确版本的区块链达成共识的方法。对当前的区块链状态达成共识对数字经济系统的正常运作至关重要。

工作量证明共识算法被认为是拜占庭将军问题的最佳解决方案之一，它可以将比特币创建为拜占庭容错系统。这意味着比特币对攻击具有很强的抵抗力，例如51％的攻击（或多数攻击）。不仅仅因为网络是去中心化的，而且还因为工作量证明算法。挖矿过程中涉及的高成本使矿工很难并且不太可能投入资源来破坏网络。

* 1. **分布式结构——开源的、去中心化的协议**

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。

**分布式记账——会计责任的分散化（Distributedaccountability）**

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？

* 1. **非对称加密算法核心技术**

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）。

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**[1]**（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。

* 1. **智能合约**

智能合约（Smartcontract）是一种特殊协议，在区块链内制定合约时使用，当中内含了程式码函式(Function)，亦能与其他合约进行互动、做决策、储存资料及传送以太币等功能。智能合约主力提供验证及执行合约内所订立的条件。智能合约允许在没有第三方的情况下进行可信交易。这些交易可追踪且不可逆转。智能合约概念于1994年由一名身兼计算机科学家及密码学专家的学者尼克‧萨博 (NickSzabo)首次提出。[2]

智能合同的目的是提供优于传统合同方法的安全，并减少与合同相关的其他交易成本。

**安全问题**

智能合约是“执行合约条款的计算机交易协议”。区块链上的所有用户都可以看到基于区块链的智能合约。但是，这会导致包括安全漏洞在内的所有漏洞都可见，并且可能无法迅速修复。

这样的攻击难以迅速解决，例如，2016年6月TheDAOEther的漏洞造成损失5000万美元，而开发者试图达成共识的解决方案。DAO的程序在黑客删除资金之前有一段时间的延迟。以太坊软件的一个硬分叉在时限到期之前完成了攻击者的资金回收工作。

以太坊智能合约中的问题包括合约编程Solidity、编译器错误、以太坊虚拟机错误、对区块链网络的攻击、程序错误的不变性以及其他尚无文档记录的攻击。

2018年4月22日，BeautyChain智能合约出现重大漏洞，黑客通过此漏洞无限生成代币，导致BEC的价值接近归零。同月25日，SmartMesh出现疑似重大安全漏洞，宣布暂停所有SMT交易和转账直至另行通知，导致损失约1.4亿美金。28日，EOS被指可能存在BEC代币合约类似的整数溢出漏洞，但没消息详细说明。5月24日，BAI交易存在大量异常问题，损失金额未知。8月22日，GODGAME合约被黑客入侵，GOD智能合约上的以太坊总数归零。

1. **联盟链和公有链的异同；**
   1. **联盟链**

如果不同的公司一起经营区块链，称联盟链。联盟链是具有由预设节点控制的共识过程的区块链。区块链可以允许每个人或仅参与者访问或采用混合访问方法。例如，根哈希及其API（应用程序接口）可以向公众开放; 外部各方可以使用API进行一定数量的查询，并获得与区块链状态相关的信息。这种区块链可以被视为“部分分散的”。

为了成为一个联盟而不是一个私人链，参与公司必须平等地参与该链的共识和决策过程。从技术上讲，可以使用奇偶校验等解决方案轻松实现联盟链。但是，有关合作伙伴对财团的平衡控制至关重要。由于这种治理往往在这些项目中被忽视，所有努力的结果往往又是一个集中控制的私人链。

联盟链与其公有链的不同之处在于它们需要权限，因此，不是任何具有互联网连接的人都可以访问联盟链。这些类型的区块链也可以被描述为半分散的。对联盟链的控制不授予单个实体，而是授予一组经批准的个人。联盟链共识流程可能与公有链不同。联盟链的共识参与者可能是网络上的一组预先批准的节点，而不是任何人都能参与该过程。因此，联盟链具有公共区块链中固有的安全特征，同时还允许对网络的更大程度的控制。

联盟链是部分私有的。关于这与完全私有系统的区别如何，这一直存在一些混淆。 Vitalik Buterin提供了一个非常简单的定义：

“到目前为止，很少强调联盟链和私有链之间的区别，尽管它很重要：前者提供了公共区块链提供的'低信任'和'单一高度可信实体'模型之间的混合体。私有链，而后者可以更准确地描述为具有一定程度的加密可审计性的传统集中系统。”**[3]**

不是允许任何具有互联网连接的人参与交易过程的验证或仅允许一个公司完全控制，而是预先确定一些选定的节点。一个联盟平台提供许多与私有区块链相关的相同利益 - 例如效率和交易隐私 - 而不需要仅与一家公司合并。您可以将其视为信任长老理事会。理事会成员通常是已知的实体，他们可以决定谁可以读取区块链分类账。

联盟区块链平台具有许多与私有区块链相同的优势，但在一个组的领导下运行而不是单一实体。该平台非常适合组织协作。想象一下，中央银行根据国际金融规则协调其活动。或者联合国将其交易分类账和投票系统外包给区块链，允许每个国家代表一个验证节点。可能性是无止境

* 1. **公有链**

对于公有链，任何个人都可以访问该信息，提交有效确认的交易，并参与其中的共识程序。公有链通过POW机制或POS机制等方法整合经济激励和加密数字验证。遵守这些规则，每个人都可以获得与对共识程序所做贡献成比例的经济激励。整个公有链系统是开放和透明的，被视为一个开源系统。而且，每个节点的身份可以保持匿名，因此保护系统中节点的匿名性和隐私性。关于上述特征，公有链被认为是完全分散的。比特币，以太坊和莱特币都采用公有链系统。

这种类型的区块链没有访问限制，意味着任何有互联网连接的人都可以成为公共区块链的参与者。换句话说，世界上任何人都能够读取区块链中包含的数据，并且允许世界上的任何人在公共区块链上执行交易。重要的是，对于谁可以参与区块链的共识过程也没有限制，区块链是确定可以向区块链添加区块的个人或实体的过程。公共区块链被认为是完全分散的，对区块链的控制不在任何单个个体或实体的控制之下。

区块链旨在在任何资产交换场景中安全地切断中间人。它通过设置一个对等事务块来实现。在将每个事务写入系统之前，每个事务都被验证并与附属于区块链的每个节点同步。在此之前，下一笔交易无法继续前进。任何拥有计算机和互联网连接的人都可以设置为一个节点，然后与整个区块链历史同步。

虽然这种冗余使公共区块链非常安全，但它也使它变得缓慢而浪费。运行每笔交易所需的电力是天文数字，并随着每个额外的节点而增加。好处是每个交易都是公开的，用户可以保持匿名。

当网络需要分散时，公共区块链是最合适的。如果分类帐或个人匿名的完全透明度是期望的好处，那也是很好的。与现有的会计系统和方法相比，成本更高，速度慢于私有链，但速度更快，成本更低。对于像比特币这样的加密货币来说，这是一个很好的权衡。安全性是他们用户的关键，分散的网络是项目的核心，他们在金融行业的竞争对手仍然比公共区块链网络更加昂贵和慢，尽管与私有区块链相比，它的速度很慢。

1. **信任链是如何建立的？**

信任链，或称数字证书链，是一连串的数字证书，由根证书为起点，透过层层信任**[4]**，

“A certificate chain is an ordered list of certificates, containing an SSL Certificate and Certificate Authority (CA) Certificates, that enable the receiver to verify that the sender and all CA's are trustworthy. The chain or path begins with the SSL certificate, and each certificate in the chain is signed by the entity identified by the next certificate in the chain.”

使终端实体证书的持有者可以获得转授的信任**[5]**，

“我们在网络上可以信任谁或什么？每台连到网络的计算机都包含可信任的根CA列表。这些根CA签发的证书可以用来为其他CA或服务器签名证书。任何证书都需有一个“信任链”，让系统看到它所信任的任何一个根证书。”

以证明身份。基于信息安全的考虑，在进行电子商务或使用政府服务时，交易的另一方用户，以根证书为基础，凭借对签发机构的信任，相信当时持有信任链终端的证书持有者确为其人，并透过公开密钥加密确保通信保密、透过数字签名确保内容无误、以及保证对方无法抵赖。

公开密钥基础建设已经在X.509及RFC 5280指定了使用信任链的认证路径验证算法**[6]**。

“In general, a chain of multiple certificates may be needed, comprising a certificate of the public key owner (the end entity) signed by one CA, and zero or more additional certificates of CAs signed by other CAs. Such chains, called certification paths, are required because a public key user is only initialized with a limited number of assured CA public keys.”

其中，会透过证书吊销列表及OCSP检查手上得到的证书是否已被证书机构在到期前撤消。另一方面，证书机构签发新的证书时，也可能透过证书透明度公布签发证书的记录，让公众查核，避免有其他机构在未得到当事人同意下滥发欺诈证书伪冒身份。CA/浏览器论坛通过了DNS证书颁发机构授权协议，参与的证书机构会在签发证书前透过域名系统检查是否已获授权。在互联网中，任何机构都可以登记域名以设立服务器，供大众连接沟通并进行电子商务或使用政府服务。虽然公开密钥加密可以确保通信保密、数字签名可以确保内容无误、以及保证对方无法抵赖；但如果数字证书未获得可供信任的数字证书认证机构数字签名（即自签证书），对方的真实身份仍然可疑**[7]**（除非通信双方早已互相认识并预先透过安全渠道交换数字证书）。

“An https web site is only secure to the extent that the web site is operated by someone in contact with the person who registered the domain name, and the communication between you and the website is encrypted to prevent eavesdropping. No other surety is implied. When you visit a secure website, Firefox will validate the website’s certificate by checking that the certificate that signed it is valid, and checking that the certificate that signed the parent certificate is valid and so forth up to a root certificate that is known to be valid. This chain of certificates is called the Certificate Hierarchy.”

数字证书认证机构在公开密钥加密基建担任了非常重要的角色，计算机软件安装并信任了其根证书，根据其私钥签发的下层证书都可（基于数字签名）被自动信任，如果是中介证书，则再下层的终端实体证书也一样被自动信任，此即构成了一条信任链**[8]**

“证书是在层次结构中创建的，其中每个个别证书都会链接到核发证书的 CA。此链接连至 CA 的证书。接着，CA 的证书会链接至核发 CA 原始证书的 CA。在找到根 CA 的证书之前，会一直重复这个程序。根 CA 的证书在本质上会受到信任。数字签名会借由信任此层次结构 (也称为“信任链接”(Chain of Trust)) 来验证实体。您可以使用 MMC 嵌入式管理单元来查看任何证书的链接，只要按两下任何证书，然后按一下 [证书路径] 选项卡即可。”

1. **Gas在智能合约中的作用；**

“gas”是以太坊使用的特殊单位的名称。它衡量一个动作或一系列动作需要执行多少“工作”：例如，计算一个Keccak256密码散列，每计算一次散列需要30个gas，每256位数据被哈希。Ethereum平台上的一项交易或合同可以执行的每项操作都会花费一定数量的天然气，其运营所需的计算资源比计算资源要求较少的运算需要更多的天然气。

gas的重要性在于它有助于确保提交给网络的交易支付适当的费用。通过要求交易支付每个操作的执行（或导致合同执行），我们确保网络不会因为执行大量对任何人无价值的密集工作而陷入困境。这与比特币交易费用不同，它仅基于交易的千字节大小。由于以太坊允许运行任意复杂的计算机代码，所以短的代码实际上可能导致大量计算工作的完成。所以衡量直接完成的工作非常重要，而不是仅仅根据交易或合同的长度选择费用。

所以，如果gas基本上是交易费用，那么你如何支付？这是一个棘手的地方。虽然gas是一个可以测量物质的单位，但gas并没有任何实际的标志。也就是说，你不能拥有1000gas。相反，gas只存在于以太坊虚拟机内部，作为正在执行多少工作的计数。在实际支付gas时，交易费用是ether的一定数量，以太坊网络上的内置令牌和矿工奖励生产块的令牌。

起初这可能看起来很奇怪。为什么不直接用ether衡量成本？答案是，就像比特币一样，以太网的市场价格可能会迅速变化！但是计算的代价并不是因为以太的价格变化而上升或下降的。所以将计算价格与以太币的价格区分开来是很有用的，这样每次市场走势就不需要改变操作成本。

这里的术语有点混乱。EVM中的操作具有gas成本，但gas本身也具有以ether的gas价格。每笔交易都规定了每个gas单位愿意支付的gas价格，从而使市场能够决定gas价格和计算成本（以天然气计量）之间的关系。这是两者的总和，即所用gas总量乘以gasprice，得到交易支付的全部费用。

尽管这很棘手，但了解这个区别是很重要的，因为这会导致以太坊交易对最初的学习者来说最混乱的一件事情：您的交易没有用完，交易也没有足够高费用。如果我在我的交易中设定的gasprice太低，那么没有人会在第一时间去管理我的交易。它不会被矿工包括在区块链中。但如果我提供一个可以接受的天然气价格，那么我的交易就会产生如此多的计算工作，以至于合并后的天然气成本超过了我所附加的费用数额，那么这个天然气就会被计算为“花费”，我不会收回。矿工将停止处理交易，恢复所做的任何更改，但仍将其作为“失败的交易”包含在区块链中，收取费用。这看起来可能很苛刻，但是当你意识到矿工真正的工作是在执行计算的时候，你可以看到他们永远也不会获得这些资源。所以，即使你设计糟糕的交易用完了，你付给他们的工作也是公平的。

提供太多的费用也不同于提供太多的ether。如果你设置了一个非常高的gasprice，那么你只需要付出很少的代价，就像在比特币中设置超高的交易费用一样。你肯定会被排在最前面，但你的钱已经没有了。但是，如果您提供了正常的gasprice，并且只需要支付比您购买gas所需的更多的ether，那么超额部分将退还给您。矿工只收取你实际工作的费用。你可以把煤气价格看作矿工的小时工资，把煤气成本看作是工作时间表。

gas还有许多其他的微妙之处，但这应该给你基本的东西！gas是使以太坊中的复杂计算“安全”的关键机制，因为任何失控的程序只会在请求运行的人提供的资金的情况下持续下去。当资金停止时，矿工们就停止工作。而你在程序中犯的错误只会影响付费使用它的人-网络的其他部分不会因为你的错误而遭受性能问题。当性能问题消耗掉所有的ether时，他们只会得到一个大的薪水！如果没有这个关键技术，通用区块链的想法将是完全不可能的。

1. **EVM中的数据存储结构；**

**以太坊虚拟机(EVM)**

以太坊虚拟机(EVM)是智能合约的运行环境。它是一个完全独立的沙盒，合约代码在EVM内部运行，对外是完全隔离的，甚至不同合约之间也只有有限的访问权限

* 1. **账户**

以太坊中有两种不同类型但是共享同一地址空间的账户：外部账户由一对公私钥控制，合约账户由账户内部的合约代码控制。

外部账户的地址是由公钥（经过hash运算）决定的，而合约账户的地址在此合约被创建的时候决定的（由合约创建者的地址和发送到此合约地址的交易数决定，这就是所谓的“nonce”）

不管是哪种类型的账户，EVM的处理方式是一样的每个账户都有一个持久的key-value类型的存储，把256字节的key映射到256字节的value

此外，每个账户都有以“Wei”为单位，在交易过程中会被修改的资产(balance)信息

* 1. **交易**

交易是一个从账户发往另一个账户（可以是同一个账户或者是special zero-account）的消息。它包含二进制数据（交易相关的数据）和 Ether。

如果目标账户包含代码，代码会被执行，交易相关的数据将作为参数

如果目标账户是地址为0的账户zero-account, 交易会创建一个新的合约。如上文提到的，合约地址不是一个地址为0的地址，而是一个由交易发送者和交易数来决定的地址。这样的一笔（到zero-account）交易的相关参数会被转化为EVM字节码

然后被执行，输出结果就是被永久存储的合约代码。这意味着为了创建一个合约，并不需要发送真实的合约代码，代码可以被自动创建

* 1. **Gas**

创建之后，每笔交易都需要一定数量的gas，用于限制交易所消耗的工作量，即交易是需要付出代价的（避免DDoS攻击）。EVM执行交易的过程中，gas会按一个特殊规则逐渐减少

费用的多少是由交易发起者设置，至少需要从发起账户支付gas\_price \* gas用费。如果交易执行完毕费用还有剩余的，将退回到发起账户。

如果交易完成之前费用耗尽，将会抛出一个out-of-gas的异常，所有的修改都会被回滚

更多关于gas的理解和讨论可以戳这里

* 1. **storage,memory,stack**

每个账户都有一个持久的内存空间，称之为storage,storage以key-value形式存储，256字节的key映射到256字节value，合约内部不可能枚举storage(内部元素)，读取或者修改storage操作消耗都很大(原文是 It is not possible to enumerate storage from within a contract and it is comparatively costly to read and even more so, to modify storage. )。 合约只能读取和修改自己的storage里的数据。

第二种内存空间称之为memory,里面存储着每个消息调用时合约创建的实例。memory是线型的，可以以字节级别来处理，但是限制为256字节宽度，写入可以是8或256字节宽度。当读取或写入一个预先未触发的指令的时候会消耗memory的空间，消耗空间的同时，必须支付gas。memory消耗的越多，需要的gas越多（按平方级增长）。

EVM不是一个注册的机器而是一个堆栈机器，所以所有的计算指令都在stack空间里面执行。stack最多只能容纳1024个长度不超过256字节的指令元素。只能用下述方法，从顶部访问stack：可以拷贝最顶部的16个元素中的一个到stack的最顶部，或者将最顶部的那个元素与其下面的16个元素之一互换。所有其它操作从stack最顶部取出两个（或一个，或更多，取决于操作）元素，然后把结果push到stack顶端。当然将stack中的元素移到memory或者storage也是可以的，但是不能直接访问stack中间的元素（必须从头部开始访问）

* 1. **指令集合**

EVM的指令集合控制的很小，这样可以避免错误的执行引发问题。所有的指令都是操作最基本的数据类型，256字节。而且都是最常见的逻辑，算法，字节和比较运算。有条件或无条件的跳转都可以。此外，合约可以访问当前区块的属性，比如区块编号和时间戳。

* 1. **消息调用**

合约之间可以通过消息调用的方式进行相互调用或者另一个给另一个无合约账户（外部账户）转币。消息调用很像交易，两者都有源账户，目标账户，数据（data payload），Ether,费用和返回数据。实际上每笔交易都由一个可创建更多调用的顶级调用组成。

合约可以决定内部消息调用的时候发送多少手续费，保留多少。如果在内部消息调用的时候抛出out-of-gas异常（或者其它异常），这个会被一个错误值标记，放到stack顶部。如此，只有和消息一起发出的手续费才会被消耗。在Solidity中这种情形默认会引发一个异常，以便异常“冒泡”到stack最顶端

如上所述，被调用的合约会接收到一个刚创建的memory实例，并且可以访问调用参数，调用参数被存储在一个被为calldata的隔离的区域。执行完毕后，被调用的合约将返回数据存储在调用合约预先创建的内存中。

调用被限制在1024深度，这意味着复杂的操作应尽量使用循环代替递归调用。

* 1. **代理调用/调用代码和库**

存在一种称为delegatecall的特殊的多样性的消息调用，和消息调用一样，只是代码目标地址是在执行时根据上下文决定的，调用者和调用参数不会改变他们的值。

这意味着合约可以在运行的时候动态的从另一个地址加载代码。存储、当前地址和资产仍然和调用的合约相关联，只有代码来自被调用的地址。

这样可以实现Solidity库的特性:反复使用的库代码可以被应用到合约的storage来实现复杂的数据结构。

**参考文献：**

**[1] N. Koblitz, Elliptic Curve Cryptosystems, Mathematics of Computation, vol. 48, pp. 203-209, 1987.**

**[2] Nick Szabo, Smart Contracts: Building Blocks for Digital Markets. www.fon.hum.uva.nl. 2017.**

**[3] V. Buterin, On Public and Private Blockchains, 2015.** [**https://blog.ethereum.org/2015/08/07/on-public-and-private-blockchains/**](https://blog.ethereum.org/2015/08/07/on-public-and-private-blockchains/)

**[4]How certificate chains work. Symantec.** <https://knowledge.symantec.com/support/ssl-certificates-support/index?page=content&actp=CROSSLINK&id=SO16297>

**[5]数位凭证：你能相信谁？趋势科技全球技术支持与研发中心.** <https://blog.trendmicro.com.tw/?p=12275> **2015-05-14 .**

**[6]RFC 5280. IETF .** <https://tools.ietf.org/html/rfc5280>

**[7]Secure Website Certificate. Mozilla.** <https://support.mozilla.org/zh-CN/kb/secure-website-certificate>

**[8]链结信任与凭证授权单位. 微软开发者网络** <https://docs.microsoft.com/zh-cn/dotnet/framework/wcf/feature-details/working-with-certificates>