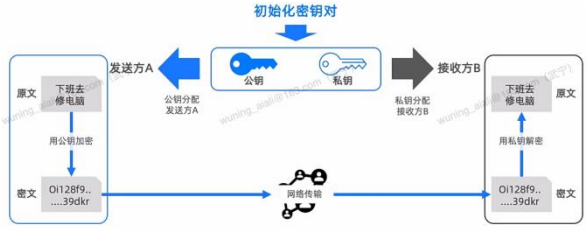
区块链学习笔记 -- 区块链核心概念

课程 1:

- 本质：去中心化的数据库
- 多项目技术的融合：

- 公钥加密原文，网络传输的是密文，接收方用私钥解密获取原文
  - 每个用户（节点）都可以获得公钥和私钥，公钥表示身份，私钥表示权利
  - 区块链中发送方用私钥对信息加密（信息摘要）生成签名，接收方通过公钥解密签名获得信息摘要
- 

课程 2: 区块链分类 & 架构相关

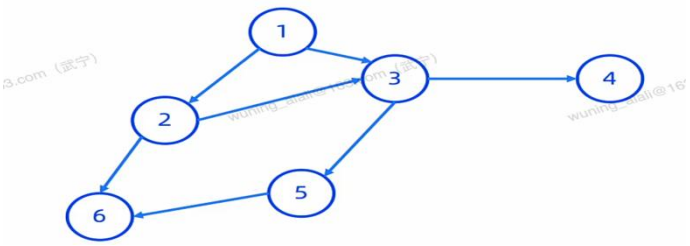
区块链分类：

- 按准入许可与否：许可链、非许可链
- 也可分为：公有链、联盟链、私有链

架构相关：

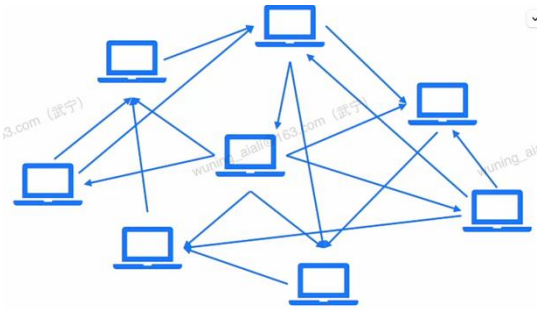
- 数据存储结构：
  - BlockChain Data Structure

  - DAG Directed Acyclic Graph Data Structure



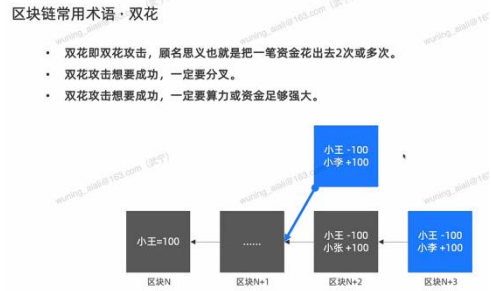
- 网络结构：

- P2P 网络机构：每个节点既可以充当服务端也可以充当客户端

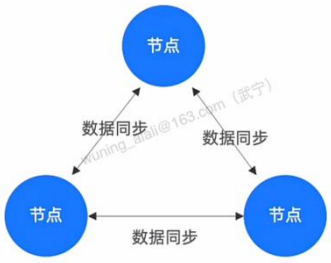


课程 3:

- 智能合约：定时脚本 + 伴有资产转移的合同履行
- 协议：网络传输协议
- 双花：黑客手段
  - 最长链表写入原则



- 共识算法：区块链系统中各分布节点对事务或状态的验证、记录、修改等行为达成一致确认的方法



[节点（N1、N2、...）]
↓ 出块（由某节点）
[新区块生成]
↓ 广播给节点们
[节点接收新区块]
↓ 验证合法性
[节点通过验证后存储区块]

  - 常见的共识算法：PoW、PoS(股权证明)、DPoS(委托权益证明)、RAFT 等
  - 节点负责生成、验证和存储区块，节点是运行区块链软件的计算机或设备

- 数字签名：用户同时创建一对公钥和私钥，公钥可以对外公开，私钥要自己保存。用户可以使用私钥对信息进行签名，其他用户可以根据该用户公布的公钥对信息进行验证。发送方加密，接收方解密，采用非对成加密算法。

✅ **数字签名的流程：**

以发送一封带签名的邮件为例：

1. **消息摘要**：对原始消息使用哈希算法（如SHA-256）生成一个摘要（固定长度的字符串）。
2. **签名生成**：发送者用自己的**私钥**对这个摘要进行加密，得到"数字签名"。
3. **消息发送**：将**原始消息 + 数字签名**一起发送给接收者。
4. **签名验证**（接收方）：
   - 使用发送者的**公钥**对签名进行解密，得到消息摘要。
   - 对收到的原始消息重新计算摘要。
   - 两个摘要比对：
     - 如果一致，说明消息未被篡改，且确实由持有私钥者发出。
     - 不一致则说明消息被改过或签名伪造。

- 加密算法：
  - 非对成加密：RSA、Elgamal、ECC
  - 对称加密： AES、DES、3DES
  - 国密算法：即国家密码局认定的国产密码算法

课程 4:

- 区块链的技术特征：

- 匿名性：

  - KYC -- Know Your Customer
  - 匿名级别：基础级、高级、极致级

- 开放性：
  - 账目的开放性
  - 组织结构的开放性

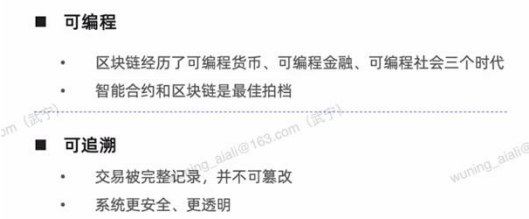
- 生态的开放性
- 开放程度：

- 开放性 vs 匿名性

- 匿名用户信息，开放系统信息

课程 5：区块链技术可编程可追溯

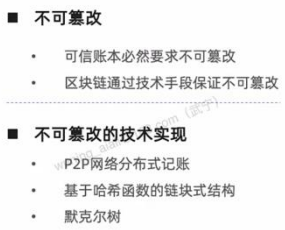
- 区块链的进化：
  - 区块链 1.0：可编程货币
  - 区块链 2.0：可编程金融
  - 区块链 3.0：可编程社会
- 区块链的可编程和可追溯：

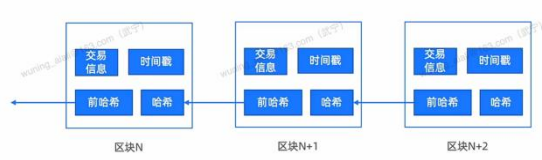


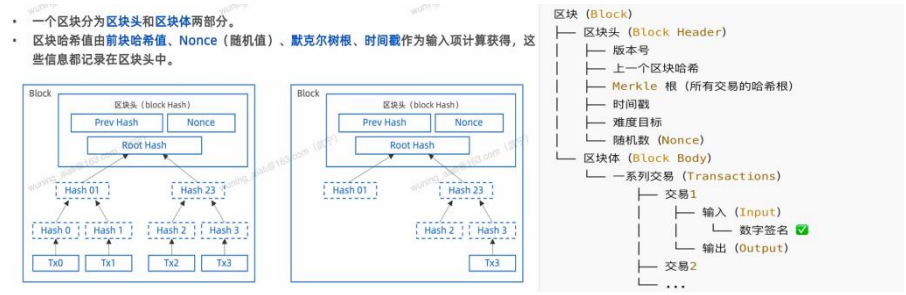
- 区块链和智能合约是最佳搭档：
  - 智能合约并非一定要依赖区块链，但却在区块链技术中得到了最佳实践。原因是区块链依赖于节点的多方认证的共识机制，对于智能合约的履行或生效关键就在于需要多方节点的监督或认证，而区块链刚好可以完美符合。
  - 智能合约在区块链中运行在一个沙箱环境中，为确保智能合约的执行不受环境干扰保证运行结果稳定，智能合约执行结果再通过区块链共识机制实现多方认证

课程 6：区块链不可篡改性：

- 不可篡改的技术实现：



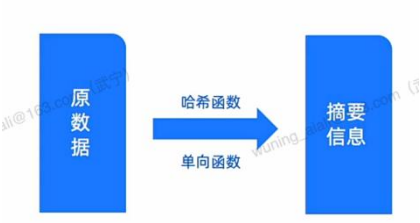
- SHA256：把**任意长度的输入**通过散列算法转换成**固定长度的输出；**哈希函数是防碰撞的
- 哈希指针组成的链块式结构：
- 

- 区块结构：
- 

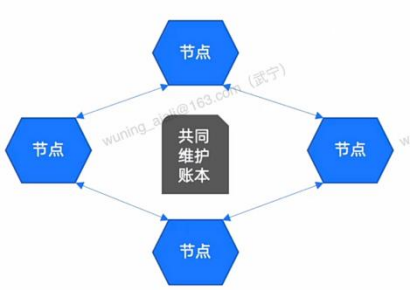
- 默克尔树：平衡二叉树
  - 当交易发生变化时，Hash 会相应发生链式变化，Root Hash 最终也会改变
  - 验证右图中 Tx3 是否在该区块链上，只需要知道：Hash2 Hash01 节点即可完成验证，无需获取所有节点的信息。

课时 7：区块链的其他属性

- 数据完整性（精确性和可靠性）技术保证：
  - 哈希算法验证数据真伪：利用哈希算法防碰撞的特性，只有相同的输入才会有相同的输出。
  - 数字签名技术验证数据修改权限，数字签名可以验证数据持有者的身份，确保只有持有私钥的用户才可以对数据进行修改
  - P2P 网络进行分布式多节点数据备份（数据全量备份），防止数据丢失或单点故障。



- 数据自治性：
  - 人类意识：投票、信任、承诺、协作、判定 均可以通过智能合约在区块链领域进行应用
- 集体维护性：区块链具有天然的开放性、共治性，区块链的参与者基于区块链的协议共同维护区块链系统

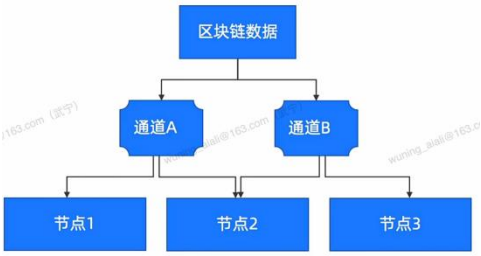


- 不可抵赖性：行为不可抵赖 + 行为发生时间不可抵赖

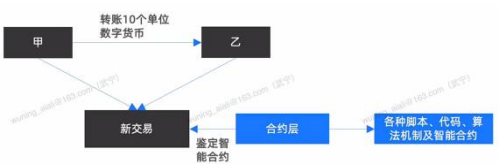
  - 如何通过数字签名确保交易不可抵赖？就像你用自己独有的“图章”盖在合同上，合同传给大家了，大家都能验证那个章是你专用的，那你就不能再说“不是我盖的”。

课程 8：区块链分层架构：

- 通道层：主要用于联盟链，做业务数据隔离，如卖水果的商家不希望和不想混淆卖拖拉机的商家的数据



- 共识层：通过 共识机制 确定记账权（解决谁负责出块的问题）
  - PoW： 谁先破解区块 Hash 谁拥有记账权
  - RAFT： 选出 leader 节点，由 leader 节点节点进行出块，其他节点跟随，从而达成共识
  - DPoS(委托权益证明)：通过投票选出若干超级节点，超级节点轮流出块，投票机制是动态筛选的，如果节点所得不好，也有可能被投出去
- 通信层：通信机制：RPC HTTP IPC（InterProcess Communication）
- 合约层：各种脚本、代码、算法机制及智能合约，是区块链可编程的基础
  - 智能合约包含：转账交易信息、虚拟资产的转移信息等，无须第三方担保，通过事务自动执行



- 应用层：将传统的应用从中心化存储数据库部署到去中心化的数据库中

■ **分层架构的作用**

- 通道层：业务隔离
- 共识层：确定记账权
- 通信层：节点间通信
- 合约层：脚本、代码、算法及智能合约
- 应用层：应用案例

附：

引申思考，我一直认为 AI 和 Web3 并不冲突，AI 会成为 Web3 的催化剂。AI 定位在提速工具，Web3 则是虚拟现实的具体产品，AI 可以帮助 Web3 提升迭代和学习的效率。

关于效率，资金使用效率提升和生产工具升级，都会带来社会生产效率的提升，带来社会供给量的增加，带来产品的价格下降，带来社会的福利的增加。

关于信息，在 AI 和 Web3 的驱动下信息的茧房不会消失但会变少。分布式账本的思路提供了去中心化、公开透明、不可篡改、可信任的数字技术，是文类文明从信息不透明逐渐走向信息透明的见证。AI 的领域对于信息的版权是无法维护的，亦是用开源的信息供给给用户开源的信息，所以无论是 AI 还是 Web3 都构建在数据开源的基础上。

关于 Token，在 Web3 中，Token 指代币，数字资产。在 AI 中，Token 指的是一个词、一个像素，数字文本 | 数字图像。共同之处，都是使用数字来抽象具体的事物。为什么都使用 Token 这个词，我的理解是因为两个领域都有从具体的人类物品到抽象的计算机表达之意。