区块链技术概述

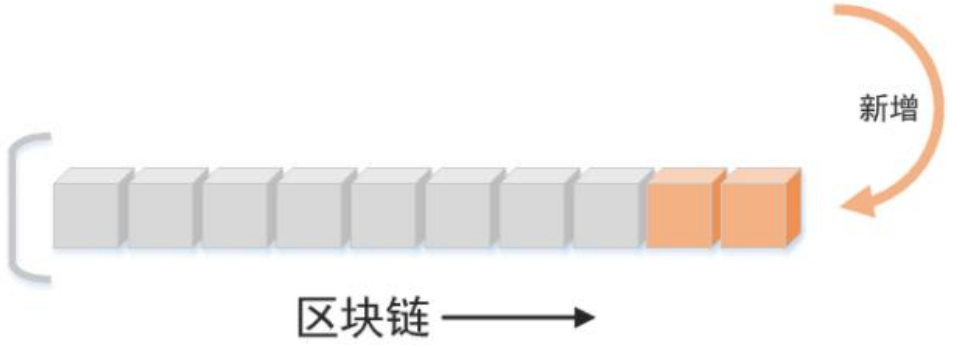
汇报人：戴家玉、顾雯雯、唐慧敏

華東師範大學
EAST CHINA NORMAL UNIVERSITY

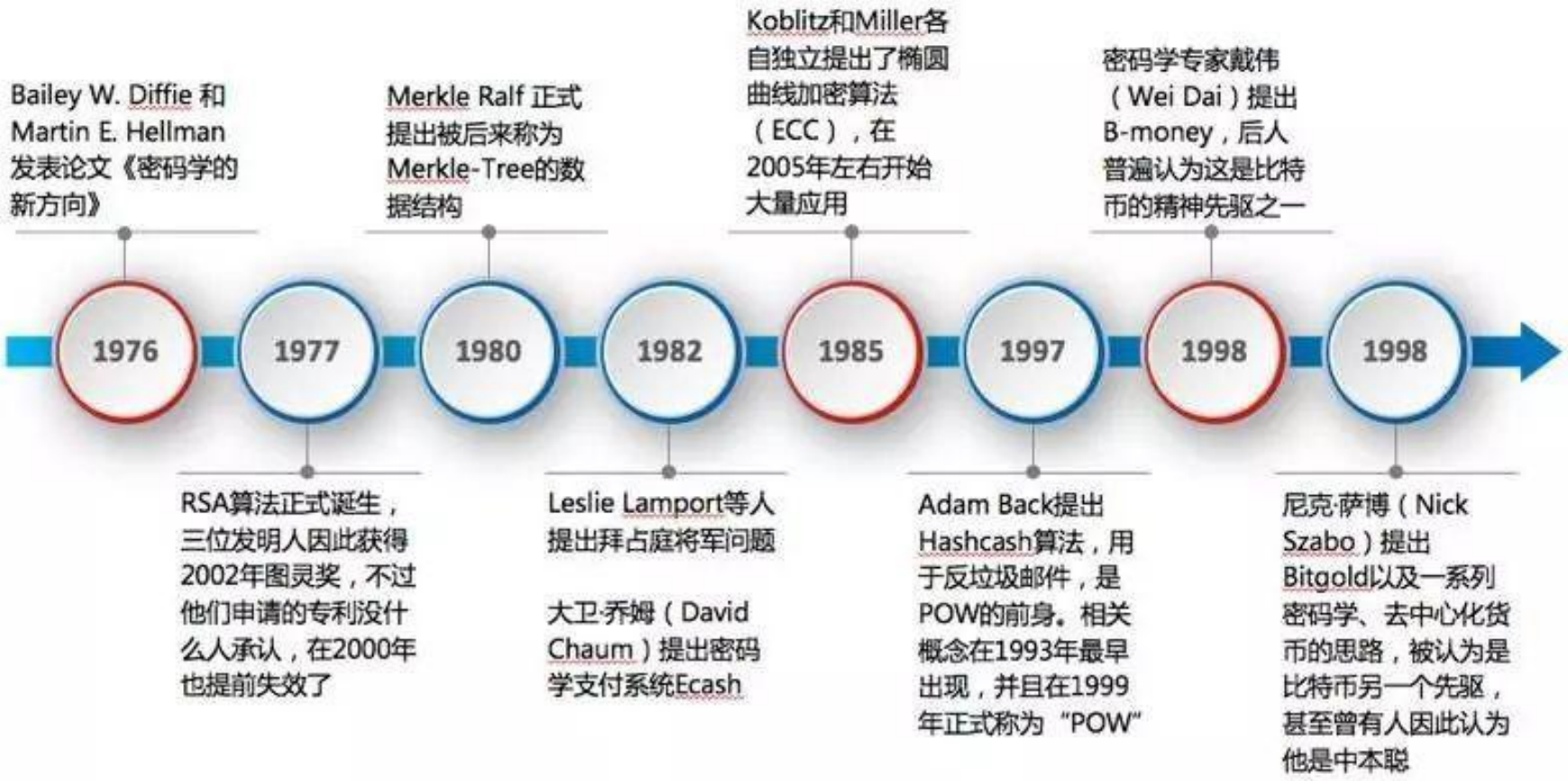
什么是区块链？

- **区块链是一种分布式账本技术，依靠智能合约等逻辑控制功能演变为完整的存储系统。**
- **从本质上讲，它是一个共享数据库，存储于其中的数据或信息，具有“不可伪造”“全程留痕”“可以追溯”“公开透明”“集体维护”等特征。**

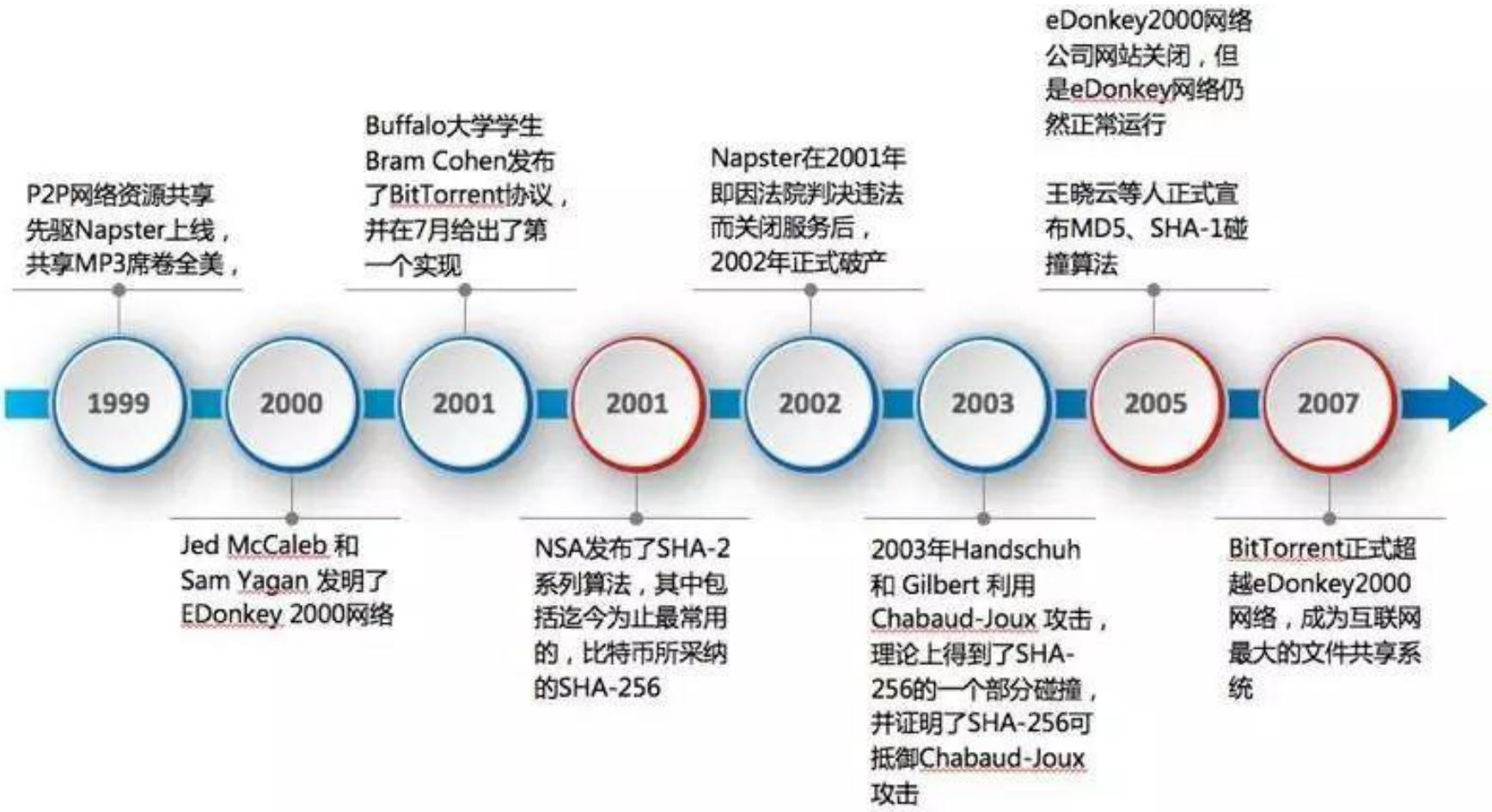
区块链中，交易信息以一个个信息块的形式记录，这些块以链条方式，按时间顺序连接起来。新生成的交易信息记录块，不断地被加到区块链中。



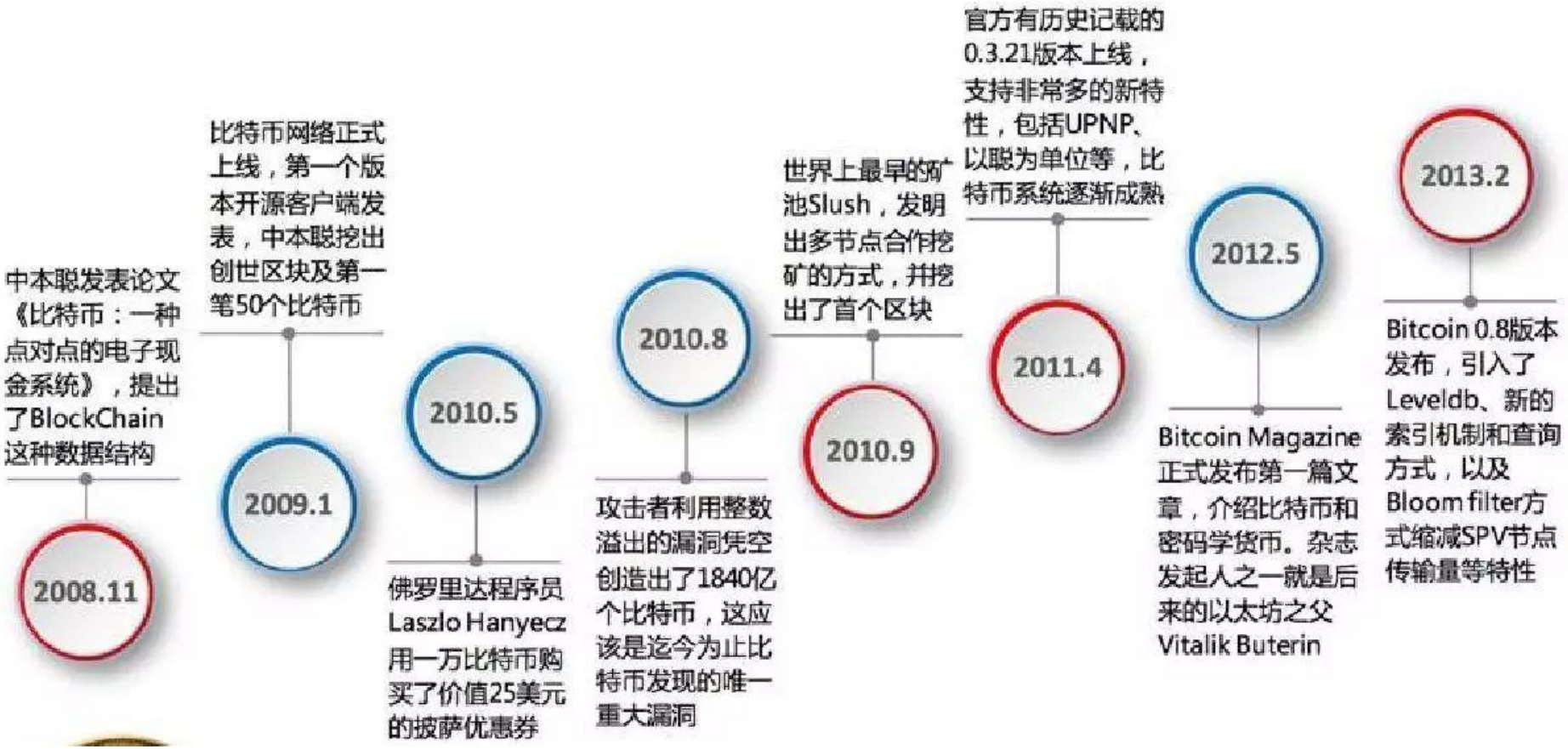
区块链的发展史


Bailey W. Diffie 和 Martin E. Hellman 发表论文《密码学的新方向》
1976
RSA算法正式诞生，三位发明人因此获得2002年图灵奖，不过他们申请的专利没什么人承认，在2000年也提前失效了
1977
Merkle Ralf 正式提出被后来称为Merkle-Tree的数据结构
1980
Leslie Lamport等人提出拜占庭将军问题
大卫·乔姆（David Chaum）提出密码学支付系统Ecash
1982
Koblitz和Miller各自独立提出了椭圆曲线加密算法（ECC），在2005年左右开始大量应用
1985
Adam Back提出Hashcash算法，用于反垃圾邮件，是POW的前身。相关概念在1993年最早出现，并且在1999年正式称为“POW”
1997
密码学专家戴伟（Wei Dai）提出B-money，后人普遍认为这是比特币的精神先驱之一
1998
尼克·萨博（Nick Szabo）提出Bitgold以及一系列密码学、去中心化货币的思路，被认为是比特币另一个先驱，甚至曾有人因此认为他是中本聪
1998

区块链的发展史


P2P网络资源共享先驱Napster上线，共享MP3席卷全美，
1999
Jed McCaleb 和 Sam Yagan 发明了EDonkey 2000网络
2000
Buffalo大学学生Bram Cohen发布了BitTorrent协议，并在7月给出了第一个实现
2001
NSA发布了SHA-2系列算法，其中包括迄今为止最常用的，比特币所采纳的SHA-256
2001
Napster在2001年即因法院判决违法而关闭服务后，2002年正式破产
2002
2003年Handschuh和 Gilbert 利用Chabaud-Joux 攻击，理论上得到了SHA-256的一个部分碰撞，并证明了SHA-256可抵御Chabaud-Joux攻击
2003
eDonkey2000网络公司网站关闭，但是eDonkey网络仍然正常运行
王晓云等人正式宣布MD5、SHA-1碰撞算法
2005
BitTorrent正式超越eDonkey2000网络，成为互联网最大的文件共享系统
2007

区块链的发展史


2008.11
中本聪发表论文《比特币：一种点对点的电子现金系统》，提出了BlockChain这种数据结构
2009.1
比特币网络正式上线，第一个版本开源客户端发表，中本聪挖出创世区块及第一笔50个比特币
2010.5
佛罗里达程序员Laszlo Hanyecz用一万比特币购买了价值25美元的披萨优惠券
2010.8
攻击者利用整数溢出的漏洞凭空创造出了1840亿个比特币，这应该是迄今为止比特币发现的唯一重大漏洞
2010.9
世界上最早的矿池Slush，发明出多节点合作挖矿的方式，并挖出了首个区块
2011.4
官方有历史记载的0.3.21版本上线，支持非常多的新特性，包括UPNP、以聪为单位等，比特币系统逐渐成熟
2012.5
Bitcoin Magazine正式发布第一篇文章，介绍比特币和密码学货币。杂志发起人之一就是后来的以太坊之父Vitalik Buterin
2013.2
Bitcoin 0.8版本发布，引入了Leveldb、新的索引机制和查询方式，以及Bloom filter方式缩减SPV节点传输量等特性

区块链的发展史


2013.3
0.8版本的一个bug导致比特币硬分叉，紧急回退到旧版，而后发布0.81版本进行解决
2013.5
第一台比特币ATM在美国加州圣地亚哥诞生
8月，德国正式承认比特币合法地位
2013.12
中国人民银行等五部委发布《关于防范比特币风险的通知》，明确比特币不具与货币等同的法律地位
2014.6
整个比特币网络的算力达到了100 PH/S
2015.8
高盛等机构估计，全美已经有160,000商家接受比特币进行交易
2016.1
整个比特币网络的算力达到了1EH/S，至今已经超过4EH/S
2016.7
研究指出，接受比特币交易的合法商家数量早已超过犯罪活动和非法交易者人数
2017.3
Github上与比特币相关的项目数量超过了10,000个
Code is not the law, but good software is good

区块链的五大特性

◆ **去中心化**

区块链最大的特性是“去中心化”，去中心化意味着所有操作都部署在分布式账本上，而不再部署在中心化机构的服务器上。通俗来说，去中心化就是无需第三方介入，任何人都可以访问整个数据库及其完整的历史记录，实现人与人、点对点的交易和互动。

◆ **不可篡改**

由于采用密码学原理将数据上链，且后一个区块包含前一个区块的时间戳，按时间顺序排序，因此区块链可以具备不可篡改或者篡改成本非常高的特性。不可篡改意味着一旦数据写入到区块链,任何人都无法轻易擅自更改数据信息。

只有掌握整个系统51%节点，才能对区块链信息进行篡改。

区块链的五大特性

◆ **可追溯**

区块链本身是一个块链式数据结构，链上的信息依据时间顺序环环相扣，这就使得区块链具有可追溯性。运用到生活上，就是产品的种植、生产、运输、销售、监管等所有信息均被记录在区块链上。一旦发生任何问题，就可以往前追溯，检测每个环节，以确保产品的安全性。

◆ **开放性**

由于区块链是去中心化的，所有网络节点都可以参与区块链网络数据的记录维护。这要求区块链网络必须是开放的。区块链网络只有开放了，才能保证所有人都可以参与进来，才能保证数据的安全性。

区块链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。区块链公有链就是充分展示区块链开放透明的例子。

区块链的五大特性

◆ **匿名性**:

如果说去中心化是很多人了解区块链的动力，那么匿名性则是很多人选择区块链的重要原因。区块链运用哈希运算、非对称加密、私钥公钥等密码学手段，在实现数据完全开放的前提下，保护个人交易隐私。

区块链的匿名性目前也屡受质疑，原因是部分不法分子利用区块链开展洗钱、资产盗取等非法行为，但由于区块链具备匿名性，仅通过地址无法获知不法分子相关身份信息，导致不法分子可以不被发现、逍遥法外，引发监管难题。目前各大项目均通过加强技术防范，降低甚至避免不法行为的发生。

区块链的设计思想

◆ **共识机制**

共识机制是指定义共识过程的算法、协议和规则，区块链的共识机制具备“少数服从多数”以及“人人平等”的特点。

◆ **密码学原理**

在区块链技术中应用了大量的密码学的知识，如：公钥、私钥、哈希、对称加密、非对称加密、同态加密、签名、零知识证明等。

◆ **分布式存储**

一是区块链每个节点都按照块链式结构存储完整的数据

二是区块链每个节点存储都是独立的、地位等同的，依靠共识机制保证存储的一致性。

区块链技术

- ✓核心技术
- ✓区块链体系结构

核心技术

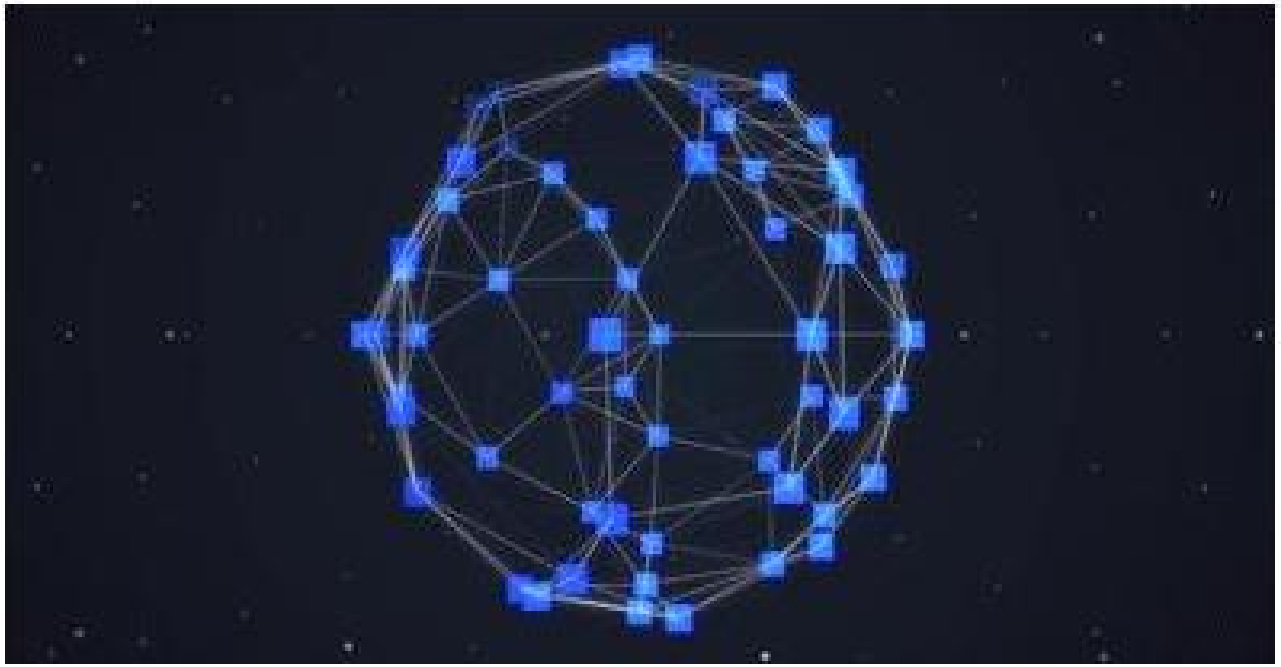
共识机制

分布式存储

密码学原理

智能合约

共识机制



- ✓ 选择一个独特的节点来产生一个区块
- ✓ 使分布式数据记录不可逆

共识机制

常见的共识机制及其应用

密码学原理

- ✓ 哈希算法
- ✓ 对称加密
- ✓ 非对称加密
- ✓ 数字签名

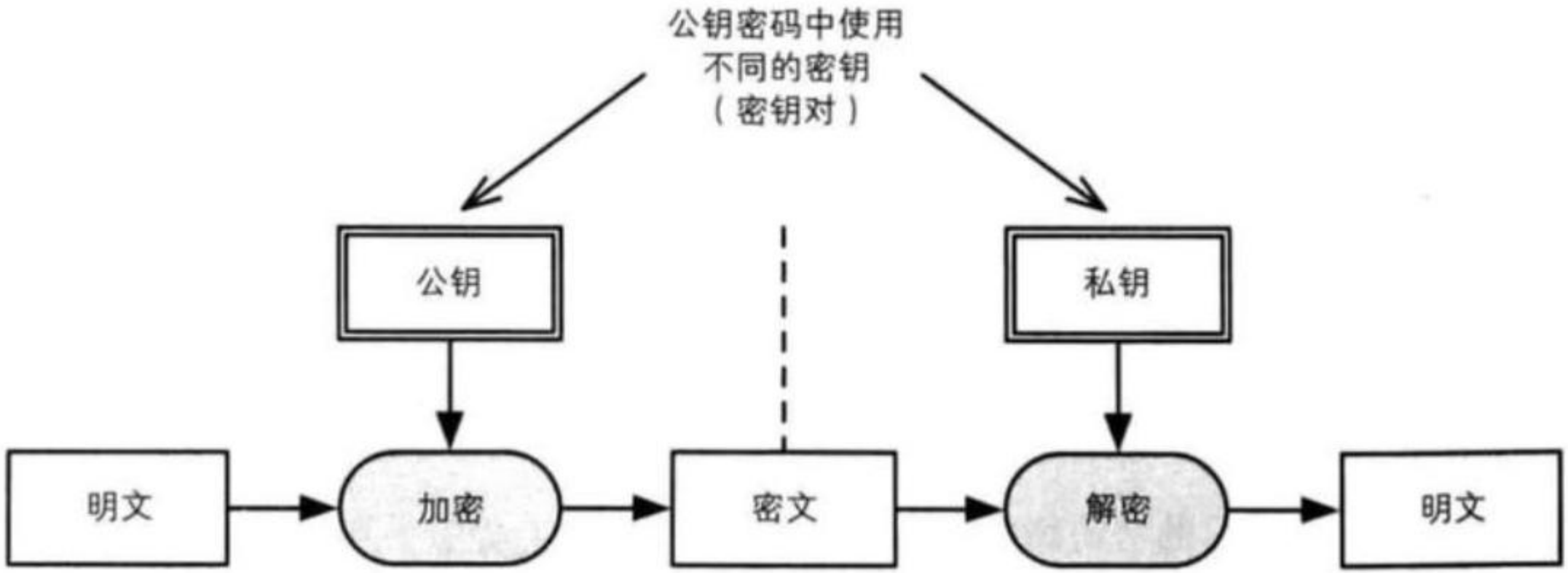
非对称加密原理：

公钥：信息的真实性

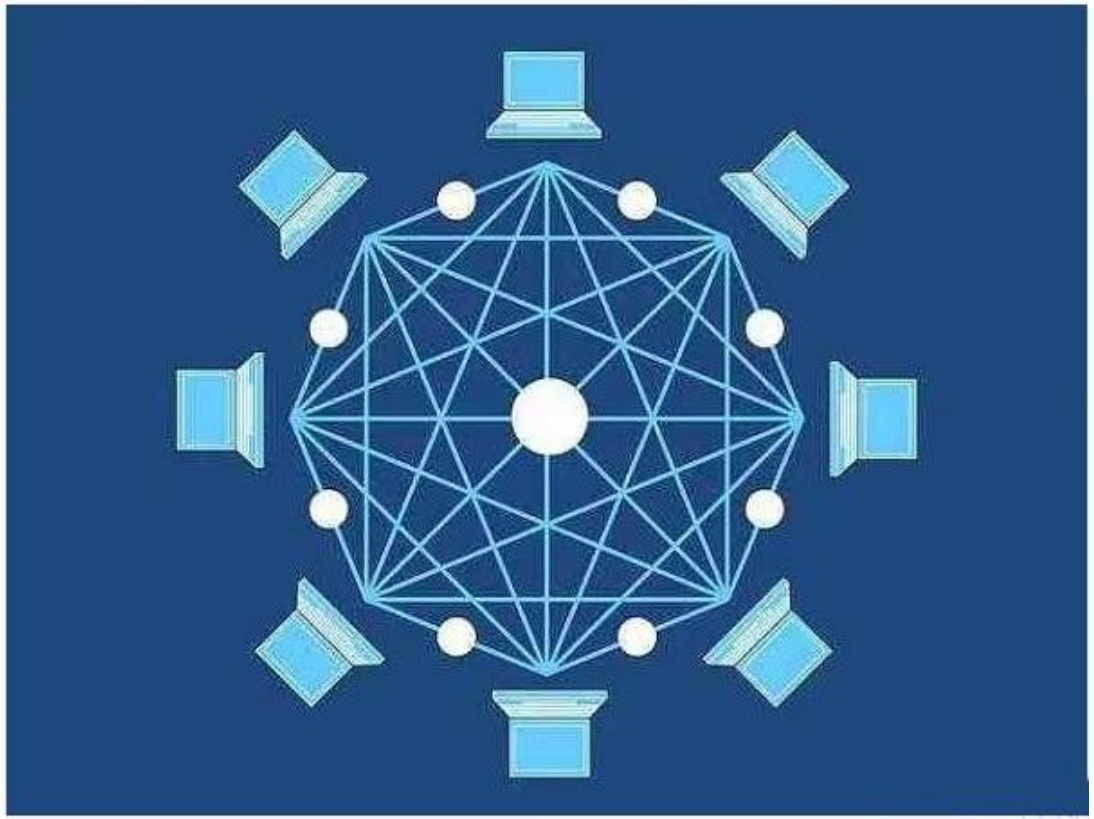
私钥：信息的安全性

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

密码学原理



分布式存储



- ✓ 区块链每个节点都按照块链式结构存储完整的数据
- ✓ 区块链每个节点存储都是独立的、地位等同的，依靠共识机制保证存储的一致性

智能合约

智能合约的三个特性：

1.数据透明

2.不可篡改

3.永久运行

智能合约

✓ 工作原理

开发人员为智能合约撰写代码

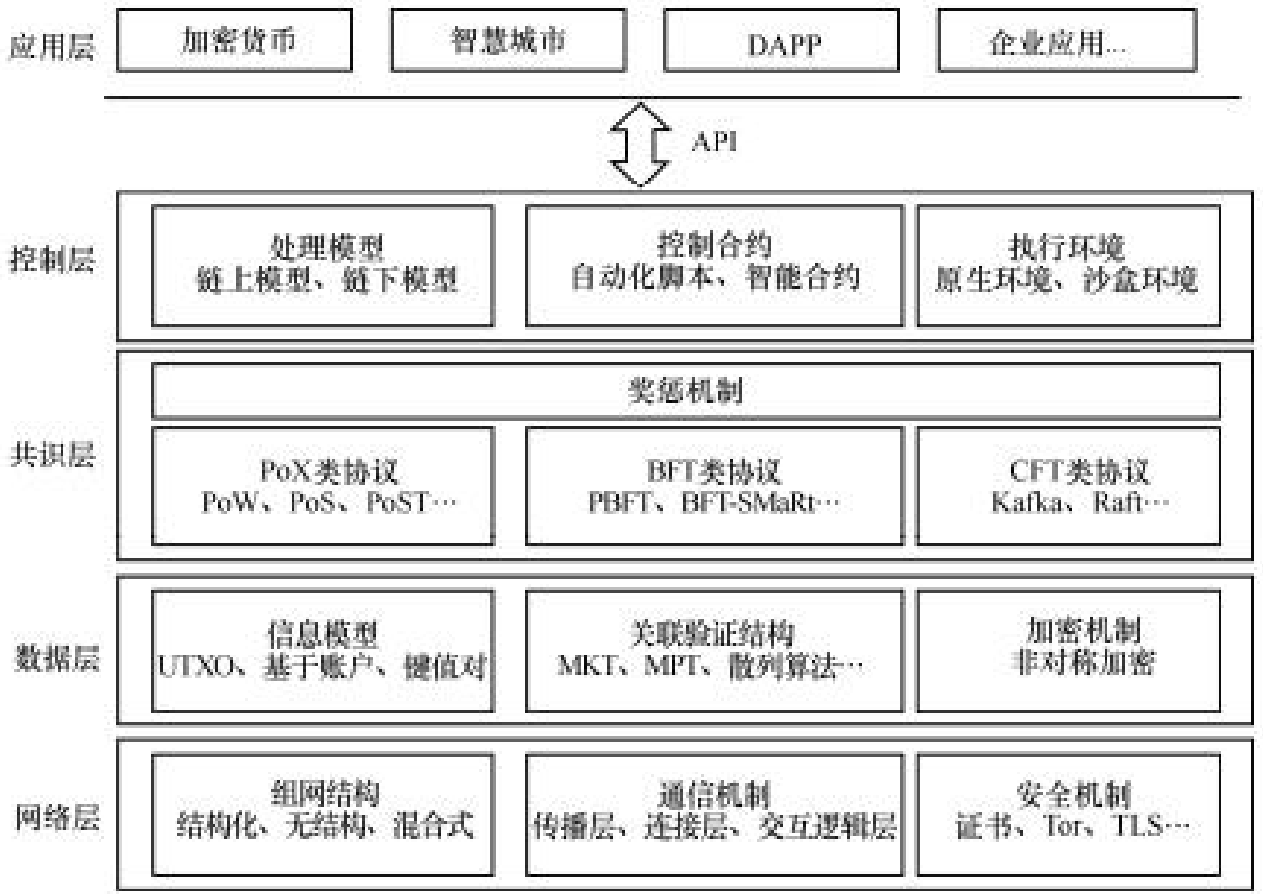
智能合约被上传到区块链网络上

用户与执行程序代码的结果达成协议

更新数据库并监督合约的条款

✓ 应用

区块链体系结构


应用层
加密货币
智慧城市
DAPP
企业应用…
API
控制层
处理模型
链上模型、链下模型
控制合约
自动化脚本、智能合约
执行环境
原生环境、沙盒环境
共识层
奖惩机制
PoX类协议
PoW、PoS、PoST…
BFT类协议
PBFT、BFT-SMaRt…
CFT类协议
Kafka、Raft…
数据层
信息模型
UTXO、基于账户、键值对
关联验证结构
MKT、MPT、散列算法…
加密机制
非对称加密
网络层
组网结构
结构化、无结构、混合式
通信机制
传播层、连接层、交互逻辑层
安全机制
证书、Tor、TLS…

技术选型分析

- ✓比特币
- ✓以太坊

比特币

比特币是目前规模最大、影响范围最广的非许可链开源项目。

为了保持账本的稳定和数据的权威性，业务制定奖励机制，即账本为节点产生新的比特币或用户支付比特币，以此驱动节点共同维护账本。

比特币网络主要由2种节点构成：全节点和轻节点。全节点是功能完备的区块链节点，而轻节点不存储完整的账本数据，仅具备验证与转发功能。全节点也称为矿工节点。

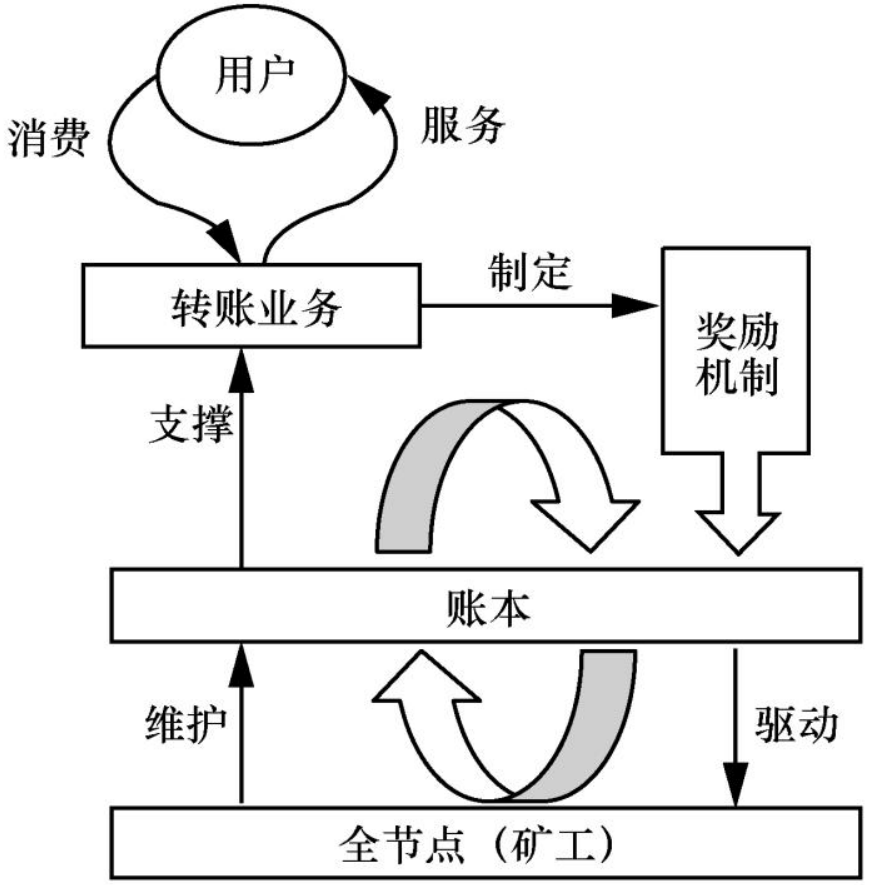
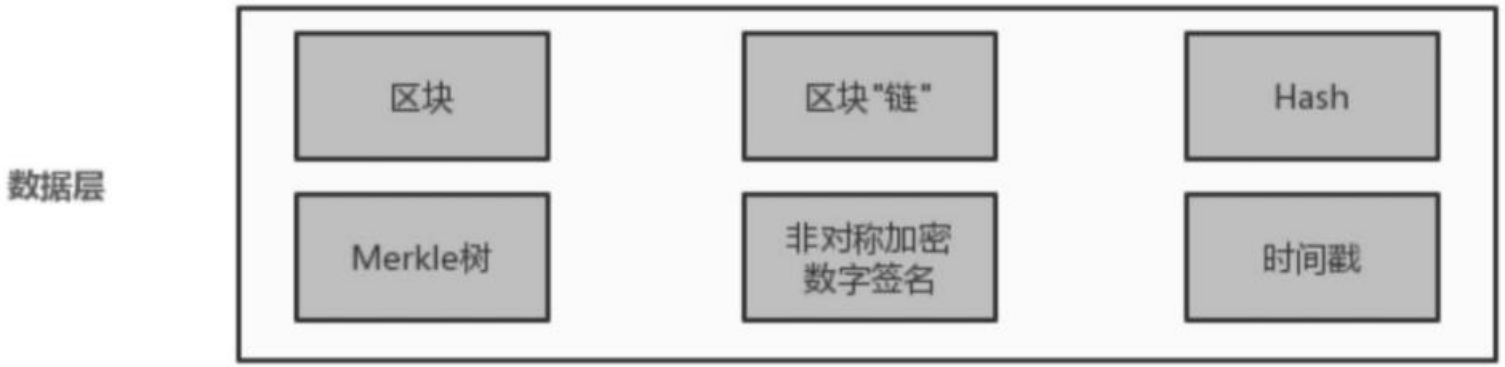


图 9　比特币运行模式

比特币-存储层

存储层主要用于存储比特币系统运行中的日志数据及区块链元数据，存储技术主要使用文件系统和LevelDB。

比特币-数据层

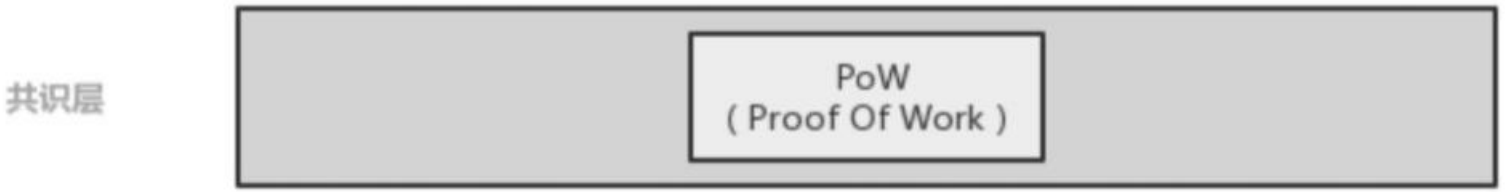


数据层主要用于处理比特币交易中的各类数据，如将数据打包成区块，将区块维护成链式结构，区块中内容的加密与哈希计算，区块内容的数字签名及增加时间戳印记，将交易数据构建成Merkle树，并计算Merkle树根节点的哈希值等。

比特币-网络层

网络层用于构建比特币底层的P2P网络，支持多节点动态加入和离开，对网络连接进行有效管理，为比特币数据传输和共识达成提供基础网络支持服务。

比特币-共识层



共识层主要采用了PoW(ProofOfWork)共识算法。在比特币系统中，每个节点都不断地计算一个随机数(Nonce)，直到找到符合要求的随机数为止。在一定的时间段内，第一个找到符合条件的随机数将得到打包区块的权利，这构建了一个工作量证明机制。

比特币-RPC层

RPC层实现了远程过程调用服务，并提供JSONAPI供客户端访问区块链底层服务。

比特币-应用层

应用层主要承载各种比特币的应用，如比特币开源代码中提供了比特币客户端。该层主要是作为RPC客户端，通过JSONAPI与比特币底层交互。除此之外，比特币钱包及衍生应用都架设在应用层上

比特币

比特币系统的脚本语言存在一些严重的限制

- ✓缺少图灵完备性
- ✓价值盲
- ✓缺少状态
- ✓区块链盲

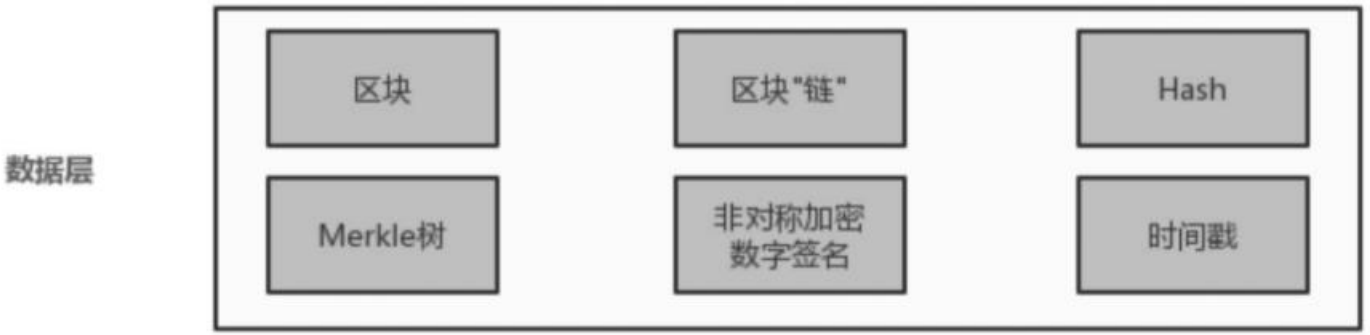
以太坊

以太坊的目的是基于脚本、竞争币和链上元协议概念进行整合和提高，使得开发者能够创建任意的基于共识的、可扩展的、标准化的、特性完备的、易于开发的和协同的应用。以太坊通过建立终极的抽象的基础层-内置有图灵完备编程语言的区块链-使得任何人都能够创建合约和去中心化应用，并在其中设立他们自由定义的所有权规则、交易方式和状态转换函数。域名币的主体框架只需要两行代码就可以实现。

以太坊-存储层

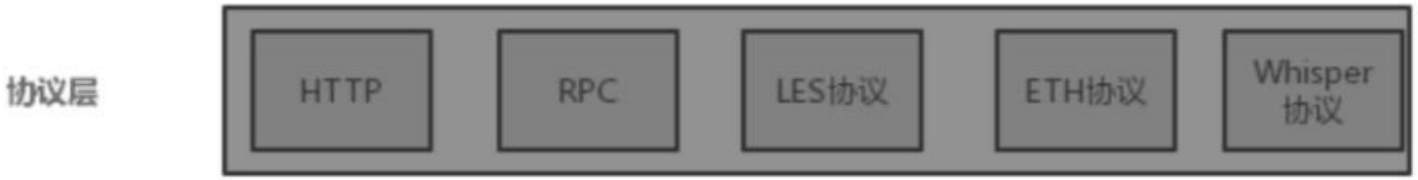
存储层主要用于存储以太坊系统运行中的日志数据及区块链元数据，存储技术主要使用文件系统和LevelDB。

以太坊-数据层



数据层主要用于处理以太坊交易中的各类数据，如将数据打包成区块，将区块维护成链式结构，区块中内容的加密与哈希计算，区块内容的数字签名及增加时间戳印记，将交易数据构建成Merkle树，并计算Merkle树根节点的hash值等。

以太坊-协议层

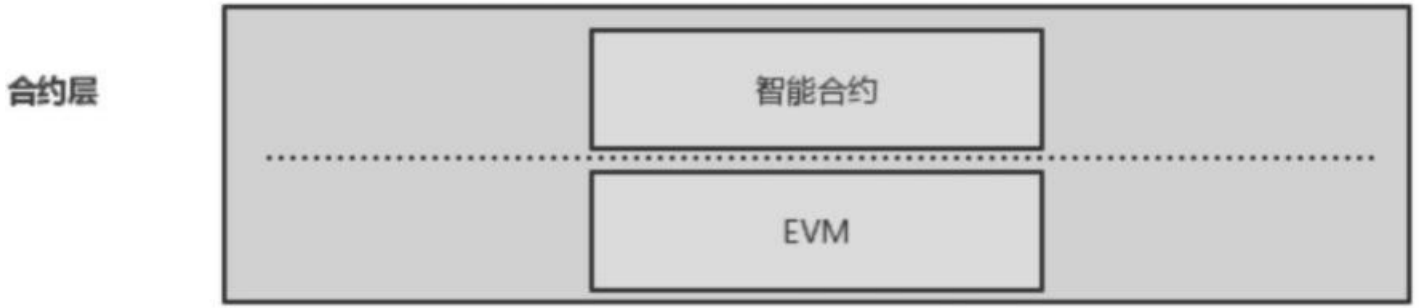


协议层是以太坊提供的供系统各模块相互调用的协议支持，主要有HTTP、RPC协议、LES、ETH协议、Whipser协议等。

以太坊-共识层

共识层在以太坊系统中有PoW(ProofofWork)和PoS(ProofofStake)两种共识算法。

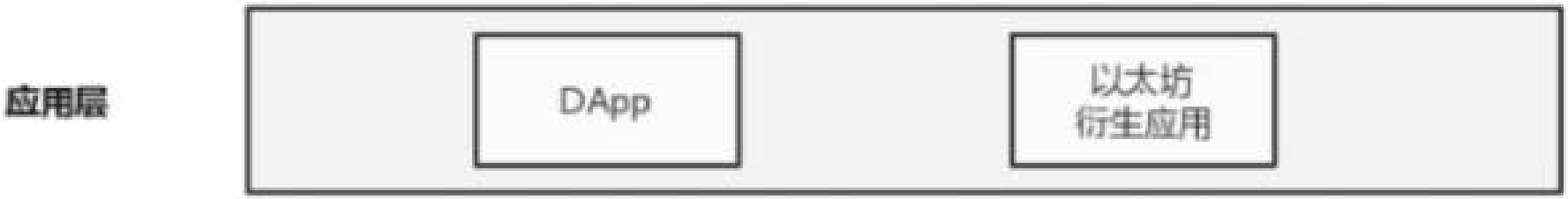
以太坊-合约层



合约层分为两层，底层是EVM(EthereumVirtualMachine，即以太坊虚拟机)，上层的智能合约运行在EVM中。智能合约是运行在以太坊上的代码的统称，一个智能合约往往包含数据和代码两部分。

智能合约系统将约定或合同代码化，由特定事件驱动触发执行。因此，在原理上适用于对安全性、信任性、长期性的约定或合同场景。

以太坊-应用层



应用层有DApp(DecentralizedApplication，分布式应用)、以太坊钱包等多种衍生应用，是目前开发者最活跃的一层。

以太坊账户

在以太坊系统中，状态是由被称为“账户”（每个账户由一个20字节的地址）的对象和在两个账户之间转移价值和信息的状态转换构成的。以太坊的账户包含四个部分：

- ✓随机数，用于确定每笔交易只能被处理一次的计数器
- ✓账户目前的以太币余额
- ✓账户的合约代码，如果有的话
- ✓账户的存储（默认为空）

以太币是以太坊内部的主要加密燃料，用于支付交易费用。

消息和交易

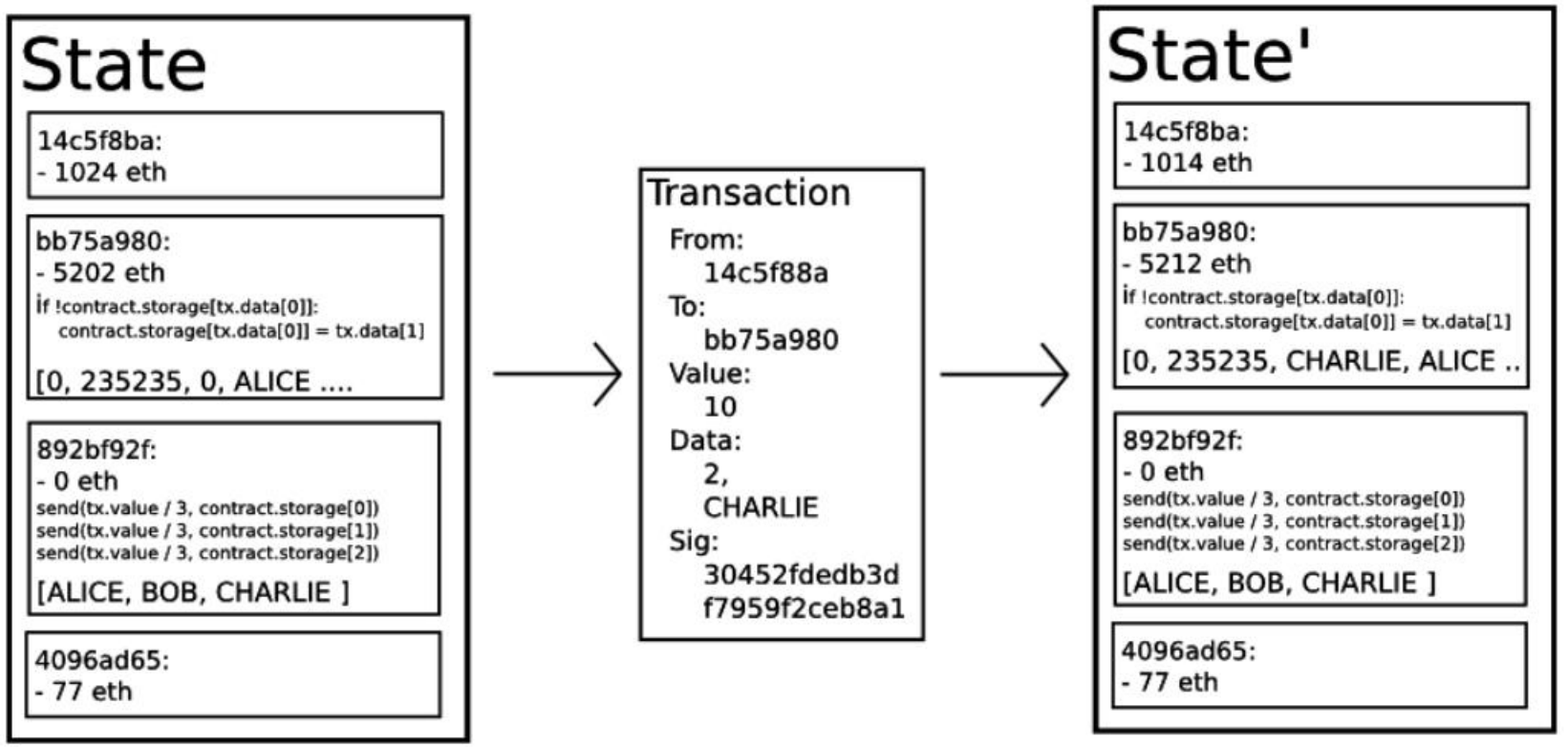
以太坊的**消息**在某种程度上类似于比特币的交易，但是两者之间存在三点重要的不同。

- ✓以太坊的消息可以由外部实体或者合约创建，然而比特币的交易只能从外部创建。
- ✓以太坊消息可以选择包含数据。
- ✓如果以太坊消息的接受者是合约账户，可以选择进行回应，这意味着以太坊消息也包含函数概念。

以太坊中“**交易**”是指存储从外部账户发出的消息的签名数据包。

- ✓交易包含消息的接收者、用于确认发送者的签名、以太币账户余额、要发送的数据和两个被称为STARTGAS和GASPRICE的数值。
- ✓为了防止代码的指数型爆炸和无限循环，每笔交易需要对执行代码所引发的计算步骤-包括初始消息和所有执行中引发的消息-做出限制。
- ✓STARTGAS就是限制，GASPRICE是每一计算步骤需要支付矿工的费用。

以太坊-状态转换函数


State
14c5f8ba:
- 1024 eth
bb75a980:
- 5202 eth
if !contract.storage[tx.data[0]]:
contract.storage[tx.data[0]] = tx.data[1]
[0, 235235, 0, ALICE ....
892bf92f:
- 0 eth
send(tx.value / 3, contract.storage[0])
send(tx.value / 3, contract.storage[1])
send(tx.value / 3, contract.storage[2])
[ALICE, BOB, CHARLIE]
4096ad65:
- 77 eth
Transaction
From:
14c5f88a
To:
bb75a980
Value:
10
Data:
2,
CHARLIE
Sig:
30452fdedb3d
f7959f2ceb8a1
State'
14c5f8ba:
- 1014 eth
bb75a980:
- 5212 eth
if !contract.storage[tx.data[0]]:
contract.storage[tx.data[0]] = tx.data[1]
[0, 235235, CHARLIE, ALICE ..
892bf92f:
- 0 eth
send(tx.value / 3, contract.storage[0])
send(tx.value / 3, contract.storage[1])
send(tx.value / 3, contract.storage[2])
[ALICE, BOB, CHARLIE]
4096ad65:
- 77 eth

以太坊-状态转换函数

以太坊的状态转换函数：APPLY(S,TX)->S’ ，可以定义如下：

①检查交易的格式是否正确（即有正确数值）、签名是否有效和随机数是否与发送者账户的随机数匹配。如否，返回错误。

②计算交易费用:fee=STARTGAS*GASPRICE，并从签名中确定发送者的地址。从发送者的账户中减去交易费用和增加发送者的随机数。如果账户余额不足，返回错误。

③设定初值GAS=STARTGAS，并根据交易中的字节数减去一定量的燃料值。

④从发送者的账户转移价值到接收者账户。如果接收账户还不存在，创建此账户。如果接收账户是一个合约，运行合约的代码，直到代码运行结束或者燃料用完。

⑤如果因为发送者账户没有足够的钱或者代码执行耗尽燃料导致价值转移失败，恢复原来的状态，但是还需要支付交易费用，交易费用加至矿工账户。

⑥否则，将所有剩余的燃料归还给发送者，消耗掉的燃料作为交易费用发送给矿工。

以太坊-区块确认算法

1.检查区块引用的上一个区块是否存在和有效。
2.检查区块的时间戳是否比引用的上一个区块大，而且小于15分钟。
3.检查区块序号、难度值、交易根，叔根和燃料限额（许多以太坊特有的底层概念）是否有效。
4.检查区块的工作量证明是否有效。
5.将S[0]赋值为上一个区块的STATE_ROOT。
6.将TX赋值为区块的交易列表，一共有n笔交易。对于属于0……n-1的i，进行状态转换
S[i+1]=APPLY(S[i],TX[i])。如果任何一个转换发生错误，或者程序执行到此处所花费的燃料（gas）超过了GASLIMIT，返回错误。
7.用S[n]给S_FINAL赋值,向矿工支付区块奖励。
8.检查S-FINAL是否与STATE_ROOT相同。如果相同，区块是有效的。否则，区块是无效的。

以太坊-应用

一般来讲，以太坊之上有三种应用。

- ✓第一类是金融应用，为用户提供更强大的用他们的钱管理和参与合约的方法。包括子货币，金融衍生品，对冲合约，储蓄钱包，遗嘱，甚至一些种类的全面的雇佣合约。
- ✓第二类是半金融应用，这里有钱的存在但也有很重的非金钱的方面，一个完美的例子是为解决计算问题而设的自我强制悬赏。
- ✓最后，还有在线投票和去中心化治理这样的完全的非金融应用。

感谢聆听！

華東師範大學
EAST CHINA NORMAL UNIVERSITY