

一、历史，金融，经济背景

1. What is Earlier Ledgers & Miners

早期账本与“矿工”的历史演变： The historical evolution of early ledgers and 'miners':

公元前 7 世纪：吕底亚人与希腊人发明标准化硬币。 7th century B.C.: Lydians and Greeks created standardized coinage.

14 世纪：如美第奇家族等商人银行兴起，涉足多国金融与制造业。 14th century: Merchant banks (e.g., the Medicis) expanded into international finance and industry.

17 世纪：银行开始贷出存款，提高生产力但也带来经济风险，中央银行应运而生。

17th century: Lending out deposits increased productivity but introduced risks; central banks emerged.

18 世纪：金本位开始形成，以储备贵金属支持货币流通。 18th century: The gold standard evolved to manage money via metal reserves.

20 世纪：金本位被巴塞尔协议替代，允许用易变现资产取代黄金储备。 20th century: The Basel Accords replaced gold with liquid asset reserves.

其他账本形式包括：族谱、土地登记、所有权证书；验证依赖公证人（Notaries）。

Other forms of ledgers included genealogical trees, land registries, and title deeds—verified by notaries.

2. What are the Characteristics of Paper Money

纸币的特征： Characteristics of paper money:

没有内在价值 No intrinsic value

更低成本的交易媒介 A less costly instrument of commerce

更便于流通与生产 Easier to circulate and produce

不易被第三方伪造 Harder to replicate by third parties

是否印制、印多少仍具争议 Still subject to debate: when and how often to print

示例：11 世纪中国纸币、1666 年斯德哥尔摩银行发行纸币、英镑、苏联钞票等。

Examples include Chinese paper money (11th century), Stockholm Banco (1666), British pound (1805), and Soviet chevronets.

3. What are the Earlier Anti-counterfeiting Measures

早期防伪措施包括： Early anti-counterfeiting measures include:

伪造行为古已有之，甚至曾被判死刑。 Counterfeiting is ancient and was often punishable by death.

金属币伪造方式包括： Metallic counterfeiting techniques include:

削边 (Clipping) Clipping

摩擦取屑 (Sweating) Sweating

掏心换料 (Plugging) Plugging

纸币防伪技术包括: Paper money anti-counterfeiting techniques include:

特种纸张 (带有声学特征) Special paper (with acoustic features)

金属安全线 Metallic security threads

全息图 Holograms

平板印刷 Intaglio printing

自然印 (如树叶纹理) Nature prints (e.g., leaf textures)

几何车纹印刷 Guilloche printing

聚合物基材 Polymer substrates

4. What is Protecting German Marks Notes

德国马克纸币的防伪机制: Anti-counterfeiting mechanism of German mark notes:

每张纸币都有唯一序列号, 包含字母与数字; Each German banknote has a unique alphanumerical serial number;

防伪校验码 (check digit) 基于数学算法生成, 用于检测伪造; Check digit is calculated using a secret algorithm for error detection;

使用一系列嵌套的置换函数 $\text{perm}()$, 加上一个数学乘法表 (用“.”表示) 作为防伪校验逻辑; Uses permutations $\text{perm}()$ and an associative operation “.” to verify validity;

示例: 序列号如 DA7843100Z7 会被拆解并套用多轮计算, 若最终结果为 0 则为真币。 Example: Serial number DA7843100Z7 is processed through transformations—if result is 0, it's valid.

5. What is the Role of Money

货币的作用包括: Money serves the following roles:

交换媒介 (Medium of exchange) Medium of exchange

支付手段 (Means of payment) Means of payment

价值储藏 (Store of value) Store of value

记账单位 (Unit of account) Unit of account

满足商品与服务的持续需求 Perpetual need for goods & services

历史上, 货币常常是用于缴纳税款的任何东西。 Historically, money often becomes anything used to discharge taxes.

6. What is the Difference Between Money and Barter

货币需要有可代表价值的代币。 Money must be tokenized.

货币需要被广泛接受, 可以终结债权债务关系。 Money must be accepted as final settlement.

非现金交易中, 货币不应赋予付款方铸币税的特权, 因此需引入银行作为第三方。 In

non-cash payments, money should avoid seigniorage for payers, thus requiring banks as third parties.

7. What are the Requirements for Electronic Money

可在线支付与离线支付 Available for online and offline payments

不可复制性 Non-reproducibility

匿名或伪匿名 Anonymity or pseudonymity

可转移性 Transferability

可分割性 Divisibility

8. What is the Origin of Money

源起包括以下几种理论： Origins include several theories:

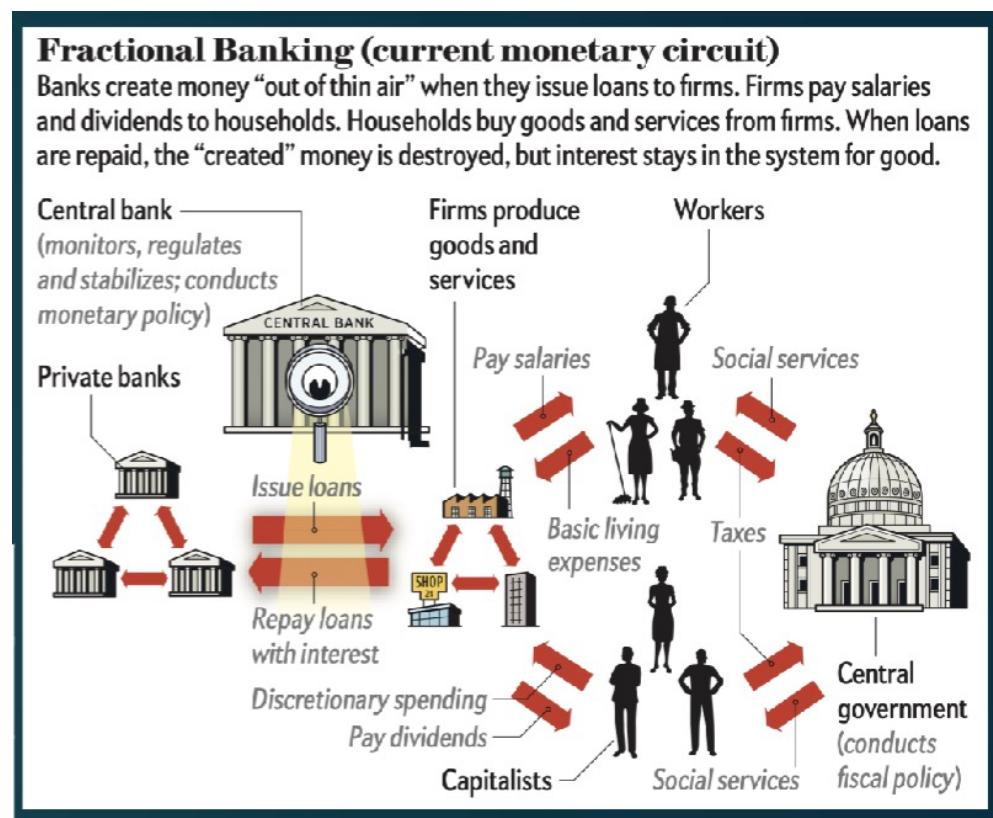
信贷创造理论 (Credit creation theory) Credit creation theory

部分准备金理论 (Fractional reserve theory) Fractional reserve theory

金融中介理论 (Financial intermediation theory) Financial intermediation theory

货币循环理论 (现代) (Monetary circuit theory) Monetary circuit theory

9. How Does the Current Financial System (Monetary System) Work



	Central Bank	Government	Private Banks	Firms	Households
Central Bank		loans + interests			
Government			interests	consumables	social services
Private Banks		loans + taxes		loans + interests	Interests + dividends
Firms		taxes	interests		wages
Households		taxes		consumables	

现金是中央银行的负债；银行存款是商业银行的负债。 Cash is the liability of the central bank; deposits are liabilities of commercial banks.

银行需保持资本充足率与流动性以避免违约。 Banks must maintain capital adequacy and liquidity to avoid defaults.

存款保险制度（如 FDIC）降低小额存款的违约风险。 Deposit insurance (e.g., FDIC) reduces default risk on small deposits.

银行之间存在互助与竞争，形成银行间网络。 Banks are both competitors and collaborators, forming interbank networks.

银行需执行 KYC、AML 等监管职责。 Banks must comply with KYC and AML regulations.

信用卡交易成本高，流程复杂。 Credit card transactions are costly and complex.

使用 SWIFT 网络与 Forex 系统进行跨境资金转移。 Cross-border transfers are conducted through SWIFT and Forex networks.

还存在传统汇款方式如 Hawala。 Traditional remittance systems like Hawala still exist.

10. What Are the Challenges for the Current Financial System

负利率 Negative interest rates

货币数量增加但使用效率下降 Currency supply increases but velocity declines

储备增加但贷款减少 More reserves but fewer loans

中介机构过多 Too many intermediaries

信息不对称导致财富集中 Information asymmetry causes wealth inequality

11. What is Financial System Architecture

货币 = 商业银行的债务 Money = liability of commercial banks

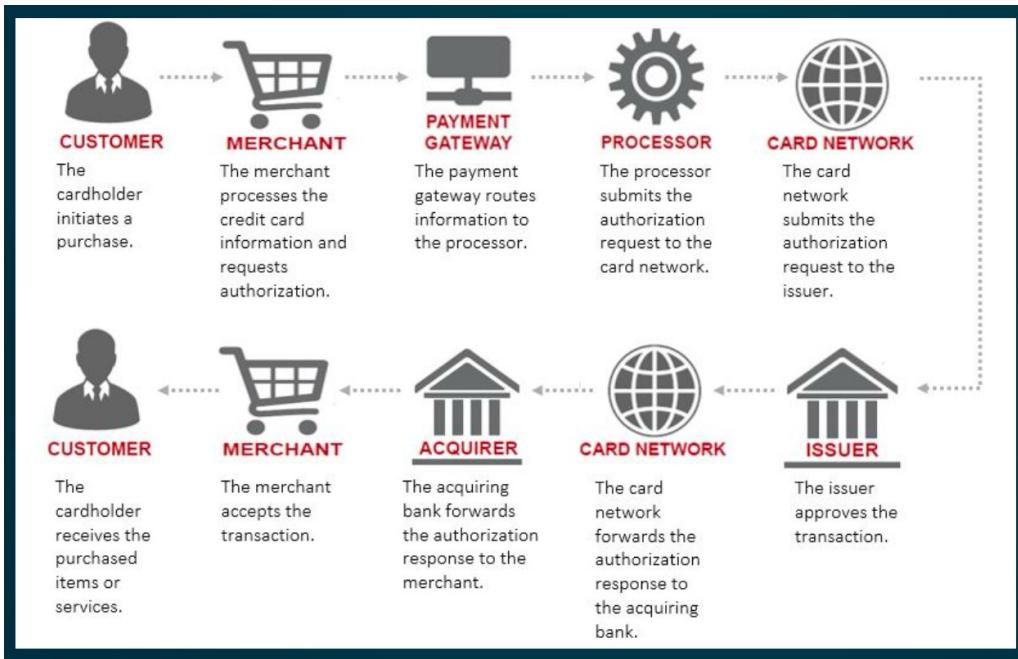
现金 = 中央银行的债务 Cash = liability of the central bank

银行根据资本、流动性、抵押品提供贷款 Banks issue loans based on capital, liquidity, and collateral.

银行通过 KYC/AML 协助中央银行监管 Banks assist the central bank with supervision via KYC/AML.

包括复杂的信贷、汇款、跨境交易网络 The system includes complex credit, remittance, and cross-border networks.

12. What is the Credit Card Transaction Lifecycle



用户使用信用卡支付的流程包括： Credit card transaction lifecycle includes:

商户请求支付 → 银行网络中转 → 发卡银行授权/拒绝 → 清算与结算 → 商户收到资金 Merchant initiates request → bank network → issuer authorization/decline → clearing and settlement → merchant receives funds

13. What is a Ledger

非银行企业账本: 资产 = 负债 + 资本 ($A = L + K$) Non-bank ledger: Assets = Liabilities + Capital ($A = L + K$)

银行账本: Bank ledger:

$$A + A' + C = L + L' + K$$

A: 资产 A: assets

A': 银行间资产 A': interbank assets

C: 中央银行现金 C: central bank cash

L: 负债 L: liabilities

L': 中央银行负债 L': central bank liabilities

K: 资本 K: capital

14. How to Analyze the Money Creation Process / Example

示例：两个银行、两个借款人、有违约和无违约情境 Example: two banks, two

borrowers, scenarios with and without default

银行 1 向客户放贷 (假设不受资本与流动性约束) Bank 1 lends to customer (assuming no capital/liquidity constraint)

客户使用资金, 银行 1 向银行 2 借钱补充流动性 Customer spends money, Bank 1 borrows from Bank 2 for liquidity

借款人还款 → 银行偿还银行 2 Borrower repays → Bank repays Bank 2

或借款人违约 → 银行资本减少 Or borrower defaults → bank loses capital

结论: 银行“凭空创造货币”, 但需保持风险控制与清算能力。 Conclusion: banks create money 'out of thin air' but must manage risk and settlement.

二、 Blockchain Basic Concepts

1. What is Blockchain

区块链是一种共享的分布式账本, 设计用于记录交易与追踪资产的所有权变更。

Blockchain is a shared distributed ledger designed to record transactions and track changes in asset ownership.

资产可以是: Assets can be:

有形的 (如货币、股票、房地产) Tangible (e.g., money, stocks, real estate)

无形的 (如知识产权、商标、声誉等) Intangible (e.g., intellectual property, trademarks, reputation)

2. What Does It Mean to Have a Blockchain

拥有区块链意味着: Having a blockchain means:

账本由用户自身维护 The ledger is maintained by users

不需要预先授权 No pre-authorization required

每笔交易都有完整的审计记录 Every transaction has a full audit trail

全账目对所有人开放 The entire ledger is open to everyone

对攻击具有高度弹性 Highly resilient to attacks

无需单一信任节点 No single point of trust required

匿名下仍可激励合作、惩罚破坏行为 Cooperation can be incentivized and misbehavior punished even anonymously

3. Why Is Blockchain So Complex

区块链是技术-社会-经济系统的交叉产物, 它融合了: Blockchain is a socio-technical-economic intersection that combines:

经济学与博弈论 Economics and game theory

加密学与安全性 Cryptography and security

分布式系统理论 Distributed systems theory

4. What Are the Benefits of Blockchain

降低风险 Reduces risk

降低成本 (不再需要中介) Reduces cost (eliminates intermediaries)

重塑业务模式 (数字身份与网络化) Transforms business models (digital identity and networking)

增强交易可信度 Increases transaction trustworthiness

提高透明度与可靠性 Improves transparency and reliability

改善数据质量与准确性 Enhances data quality and accuracy

降低欺诈与网络犯罪风险 Mitigates fraud and cybercrime

5. What Challenges Does Blockchain Address

区块链解决了以下问题：Blockchain addresses the following challenges:

如何在匿名且动态的参与者中建立信任 How to build trust among anonymous, dynamic participants

如何无需银行开户 How to operate without bank accounts

如何在缺乏法律身份下证明账户所有权 How to prove ownership without legal identity

如何在无可信数据提供方下存储账本 How to store ledgers without trusted providers

如何在无中央银行的情况下调节货币政策 How to regulate currency without a central bank

6. What Is Centralized, Decentralized, and Distributed

集中式：单一控制点 Centralized: one control point

去中心化：多个控制点 Decentralized: multiple control points

分布式：点对点网络 Distributed: peer-to-peer network

举例： Examples:

国内银行：集中式 Domestic banking: centralized

国际汇款：去中心化 International remittance: decentralized

比特币：分布式 Bitcoin: distributed

7. What Are the Communication Models, Pros and Cons

Client-Server 模式 Client-Server model

优点：简单、高效 Pros: simple, efficient

缺点：存在单点故障，不可扩展 Cons: single point of failure, poor scalability

P2P 模式 P2P model

优点：弹性强、可扩展、自管理 Pros: resilient, scalable, self-managed

缺点：协议复杂、调试困难 Cons: complex protocols, hard to debug

8. What Is Distributed Identity: Identification

基于公钥加密的身份验证系统: Identity system based on public-key cryptography:

私钥用于签名 (仅用户自己拥有) Private key is used for signing (user-only possession)

公钥可公开验证签名是否属实 Public key can verify the signature's authenticity

9. What Is Distributed Identity: Wallets

钱包是用户的公钥/私钥对, 或者为每笔交易生成新的密钥对。 A wallet is a user's public/private key pair or generates a new pair for each transaction.

公钥 = 银行账户号 (接收转账) Public key = bank account number (receives funds)

私钥 = 授权转账 (通过签名) Private key = authorizes transfer (via signature)

10. What Are Network Nodes

网络节点的角色包括: The roles of network nodes include:

监控网络流量 Monitoring network traffic

验证交易与区块 Validating transactions and blocks

应用共识协议 Applying consensus protocol

协助拒绝无效交易 Helping to reject invalid transactions

11. What Is the Role and Lifecycle of Network Nodes

节点类型: Types of nodes:

Archive Node: 存储全链数据 Archive Node: stores entire blockchain

Seed Node: 提供入口连接 Seed Node: provides bootstrap connections

生命周期包括: Lifecycle includes:

引导阶段: 连接已知节点 Bootstrap: connect to known peers

发现阶段: 获取更多节点地址 Discovery: learn about more peers

同步阶段: 下载账本 Synchronization: download ledger

服务阶段: 参与转发与验证 Service: participate in relay and validation

12. What Is a Hash Function

哈希函数将任意长度数据映射为固定长度摘要, 具备特性: A hash function maps input of any length into a fixed-length digest, with these features:

确定性: 相同输入 → 相同输出 Determinism: same input → same output

单向性: 无法反推出输入 One-way: infeasible to reverse

抗碰撞: 不同输入几乎不可能输出相同结果 Collision resistance: different inputs unlikely to match

13. 区块链中加密哈希的作用是什么 In a blockchain, what is the role of cryptographic hashing

为区块创建唯一标识符 create unique identifiers for blocks.

13. What Is the Structure of Blockchain Data

区块链是一串按顺序链接的区块，每个区块包括： Blockchain is a sequential chain of blocks, each containing:

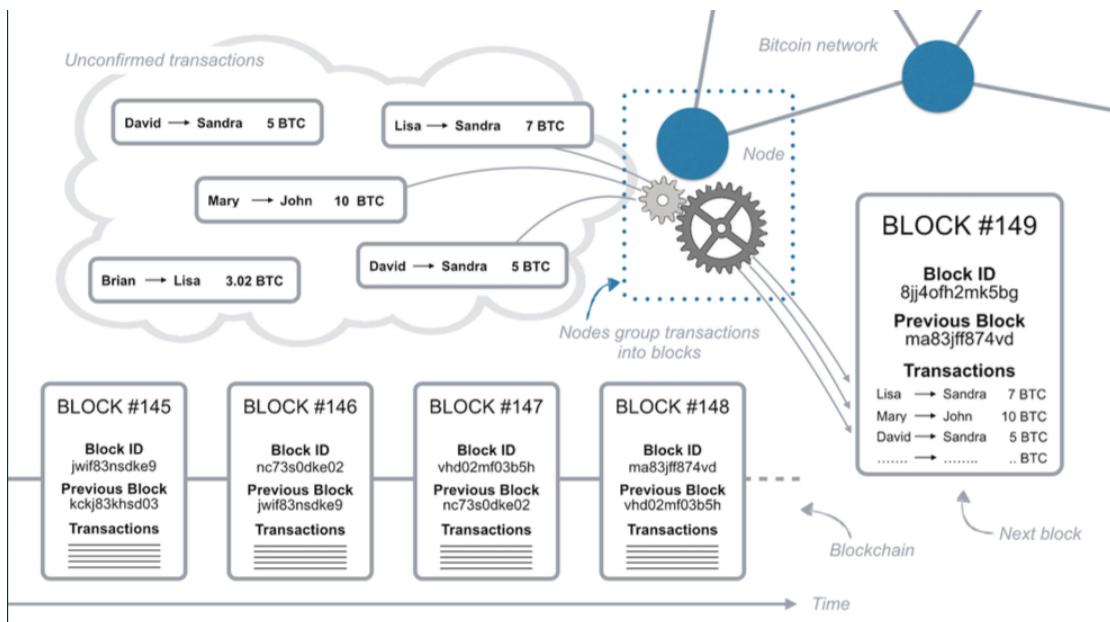
前一区块的哈希值 Hash of the previous block

当前区块的交易列表 Current block's transaction list

例如： $\text{block}_0: (\text{transactions}_0)$ Example: $\text{block}_0: (\text{transactions}_0)$

$\text{block}_i: (\text{hash}(\text{block}_{i-1}), \text{transactions}_i)$ $\text{block}_i: (\text{hash}(\text{block}_{i-1}), \text{transactions}_i)$

14. How Does Blockchain Work



篡改某一区块会导致后续所有区块的哈希无效 Tampering with one block invalidates all subsequent hashes

因此篡改成本极高，保证了数据不可更改性 Thus, tampering is expensive and ensures immutability

节点可自由决定加入哪些交易（按费用、大小、到达顺序） Nodes independently select transactions based on fees, size, and arrival order

15. What is Mining & Miners

通过计算复杂的哈希难题，来验证并将新的区块添加到区块链中，从而维持整个网络的共识与安全。

挖矿是将新区块添加到区块链的过程。 Mining is the process of adding blocks to the blockchain.

比特币每 10 分钟添加一个区块，以太坊约 14 秒。 Bitcoin: ~ every 10 mins,

Ethereum: ~ 14 sec.

挖矿通过哈希值满足网络难度目标。 Hashing must meet the network difficulty target.

需要暴力尝试生成正确的 nonce (一次性使用的数字)。 Requires brute-force search for a valid nonce.

提供正确 nonce 的区块即被视为有效。 Block with the correct nonce is considered valid (proof of work).

挖矿实现了什么：

- 验证交易合法性；
- 维护交易顺序；
- 防止双花；
- 添加新区块；
- 奖励矿工（新币 + 交易费）。

16. What is Proof of Work

生成有效证明非常困难，但验证容易。 Difficult to produce but easy to verify.

需要大量计算资源与电力。 Requires significant computational resources and energy.

矿工通过交易费与区块奖励获得回报。 Miners earn transaction and mining fees.

17. What are Limits of Proof of Work

高能耗且不可持续。 Consumes high energy, unsustainable.

交易吞吐量低，出块慢。 Low transaction throughput, slow block production.

容易出现算力集中。 Power concentration still possible.

没有交易终局性保证。 No finality guarantee.

竞争挖矿需高性能硬件和便宜电价。 Requires high-end hardware and cheap electricity.

扩展性差。 Poor scalability.

18. 什么是权益证明 What is Proof of Stake

Proof of Stake 是一种更节能的共识机制，使用者不靠“算力”，而是依靠持有的币的数量和时间来获得出块权。

节点不再比谁算得快，而是根据“你持有多少币”和“质押时间”来决定谁有权打包新区块；你质押越多币、质押时间越长，你越可能被选中；被选中者可以获得区块奖励（或手续费）；如果作弊，质押的币可能被惩罚性销毁（叫 slashing）。

特点：能耗低，效率高；安全性仍然较高（通过惩罚机制保障）；被以太坊 2.0、Cardano、Solana 等现代区块链广泛采用。

Proof of Stake is a more energy-efficient consensus mechanism. Users do not rely on "computing power" but on the number and time of coins held to obtain the right to produce blocks.

Nodes no longer compete to see who can calculate faster, but decide who has the right to pack new blocks based on "how many coins you hold" and "staking time"; the more coins you stake and the longer you stake, the more likely you are to be selected; those who are selected can get block rewards (or handling fees); if you cheat, the staked coins may be punitively destroyed (called slashing).

Features: low energy consumption, high efficiency; security is still high (guaranteed by a penalty mechanism); widely adopted by modern blockchains such as Ethereum 2.0, Cardano, and Solana.

19. What are Limits of Proof of Stake

不存在风险：无需成本就可支持多个分叉。 Nothing at stake problem: costless voting on forks.

极端事件下难以恢复。 Hard to recover from long-term forks.

需要加密货币才能参与。 Requires cryptocurrency to participate.

富者更富。 Rich-get-richer effect.

奖励较低。 Lower rewards than PoW.

What is Proof of Authority

Proof of Authority (PoA) 是一种共识机制，特别适用于私有链或联盟链，它牺牲了部分去中心化以换取更高的效率和吞吐量。

Proof of Authority (PoA) 是一种共识算法，其中预先授权的一组验证者 (Authorities) 负责依次打包区块，无需竞争或大量计算，依赖其身份信誉来维持安全性。

特征	说明
无竞争出块	每个区块由某个预定的验证节点依次打包
节点基于身份信任	验证者通常是组织、公司、政府等具备可信身份的实体
高性能	几乎没有计算资源消耗，TPS 高，延迟低
适合许可链	更适用于企业级、联盟链环境，如供应链、金融等场景
去中心化程度低	权力集中在少数节点，不适合公有链

20. What are Blockchain Full Nodes

全节点是指一个运行了完整区块链客户端程序的节点，拥有并验证整个区块链历史数据（从创世区块到当前最新区块）。

三个作用：

- 1) 存储数据：保存区块链上的每一个区块、交易、状态数据（包括地址余额等）；是最完整、最可靠的数据源。
- 2) 验证交易和区块：对接收到的每一笔交易、每一个新区块进行独立验证，确保没有作弊、双花攻击等问题；不相信别人，所有事情自己算一遍。
- 3) 传播数据：将验证后的交易和区块广播给其他节点；形成区块链网络的“信息中继站”。

完整节点维护区块链完整性。 Full nodes ensure blockchain integrity.

可以是挖矿节点，但不是所有完整节点都挖矿。 Can be mining nodes, but not all full nodes mine.

拥有完整区块与交易数据。 Contain full copy of blockchain and all transactions.

仅挖矿节点可获得奖励。 Only mining full nodes receive rewards.

21. What are Blockchain Light Nodes

轻节点是资源占用少、适合终端设备使用的区块链节点，主要依靠全节点来验证交易和数据。

- 1) 只下载“区块头（Block Header）”：区块头包含区块哈希、时间戳、Merkle root 等关键信息；不保存每一笔交易的完整内容。
- 2) 依赖全节点验证交易：当需要验证某笔交易时，轻节点会通过全节点请求“默克尔证明（Merkle Proof）”；轻节点不自己验证每笔交易，而是验证这笔交易是否在一个合法区块中。
- 3) 同步速度快，占用资源低：不需要几百 GB 的存储空间；适合在浏览器、手机、IoT 设备上运行。

全节点像一本完整账本，每一页、每一笔都记下来了；轻节点像是账本的目录页，知道哪些账存在，但需要别人（全节点）提供原始记录才能查明细。

轻节点仅存储区块头。 Light nodes store only block headers.

使用简化支付验证（SPV）验证交易。 Use Simplified Payment Verification (SPV) to verify transactions.

无需下载整个区块链，常用于钱包。 Do not download full blockchain; used in wallets.

22. 区块链类型

区块链类型	访问权限	控制权	适用场景
Permissionless	完全公开	无控制权	公有网络（如比特币，以太坊）
Permissioned	私有	中心化控制	单一机构或企业内部系统
Consortium	半私有	多方协作控制	多个组织间的协作（如银行联盟）
Hybrid	公私结合	灵活控制	需要公开与私有混合的系统

22.什么是权限管理 What is Permissioning

Permissioning 就是指网络中的访问与参与权限：哪些人能加入网络、读取账本、发布交易、参与共识等。

1. Permissionless ledger (无许可账本 / 公有链)

- 特点：任何人都能加入、下载账本、参与网络，不需要事先授权；
- 代表：Bitcoin、Ethereum；
- 完全去中心化、开放透明。

2. Permissioned ledger (有许可账本 / 私有链)

- 特点：只有获得许可的节点才能读取、写入或参与共识；
- 代表：R3 Corda、Hyperledger Fabric；
- 多用于企业或联盟链，有访问控制、治理机制。

3. Hybrid ledger (混合账本)

- 特点：结合了公有链和私有链的优点，可配置公开或私密功能；
- 代表：Dragonchain；
- 灵活适配不同场景，如部分数据对外开放、部分内部保密。

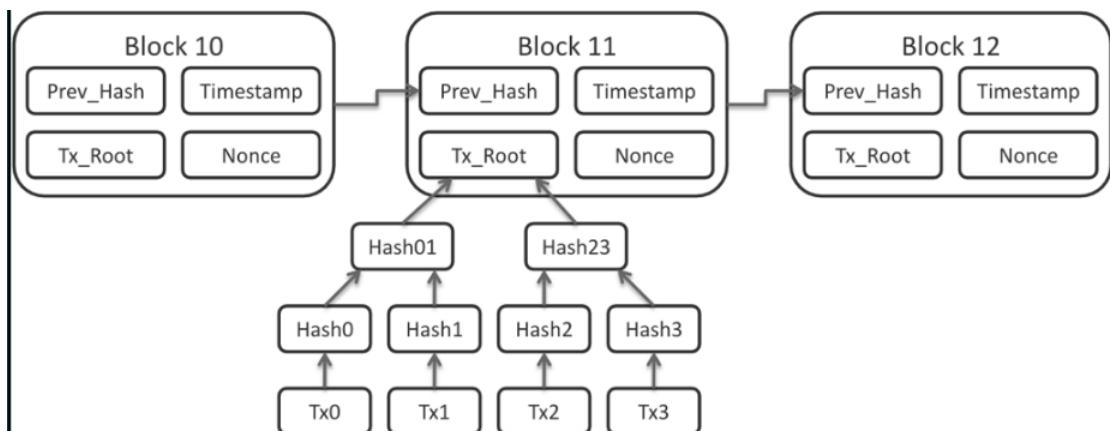
4. Sidechain (侧链)

- 特点：是连接在主链上的辅助链，可用于测试新功能、扩展性能；
- 优点：提高事务速度、减少主链负担、降低成本；
- 代表：Rootstock (比特币侧链)、Ardor。

比较维度	Permissionless (无许可)	Permissioned (有许可)
概述	任何人都可参与，完全去中心化	仅授权方可参与，部分去中心化
别称	公有链，去信任系统	私有链，权限控制沙盒
主要特点	开源、无中心、匿名、抗审查、高透明度	有中心控制、可定制隐私、安全策略明确
优势	更去中心化、更抗审查、公开透明、安全强	更高效率、更好隐私、更可定制、更易扩展
劣势	效率低、能耗高、扩展性差、隐私少	中心化风险、信息透明度低、权限控制难平衡
应用场景	B2C、P2P、政府对公民服务	B2B、政企合作

23. What is Block Structure

区块头包含：上一区块哈希、时间戳、Merkle 树、nonce 和难度目标。 Block header includes: previous hash, timestamp, Merkle tree, nonce, and difficulty target.
 区块包含交易列表。 Block contains a list of transactions.



24. What is Merkle Tree

Merkle 树（也叫哈希树）在区块链中的主要作用是：通过树状结构高效地验证某个交易是否包含在一个区块中，而不必下载整个区块的数据。

它的结构如下：

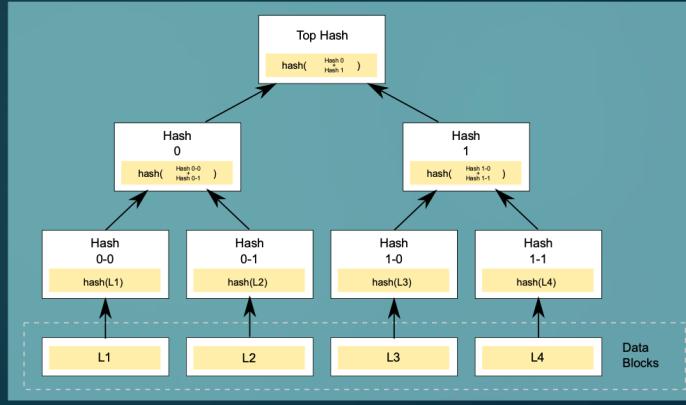
- 每个叶子节点是一笔交易的哈希；
- 上层节点是其子节点哈希拼接后的再次哈希；
- 最终得出一个根节点（Merkle root），写入区块头中。

Merkle 根记录在区块头中。 Merkle root stored in block header.

SPV 可使用 $\log 2n$ 个哈希验证交易。 SPV uses $\log 2n$ hashes to verify transactions.

Merkle Tree

- ▶ **Leaves:** transactions
- ▶ **Non-leaf nodes:** hashes
- ▶ **Merkle root:** stored in block header
- ▶ **Block:** contains transactions as a Merkle tree
- ▶ **SPV use:** match of Merkle roots calculated with $\log_2 n$ local & remote hashes to verify transactions of a light node, where n is the # of transactions



25. Why Do Users Mine Despite the Costs

用户通过挖矿获得奖励与交易费。 Users earn block rewards and transaction fees.

比特币区块奖励每四年减半，最终上限为 2100 万个。 Bitcoin reward halves every 4 years until 21 million cap.

交易费由交易发起者支付，用于优先打包。 Transaction fees are optional incentives for miners.

26. What is Incentives & Difficulty

Difficulty (挖矿难度)

endogenously defined & time-varying so that block production rate is stable

挖矿难度是由系统内部动态设定且随时间变化的，用来保持区块生成速率稳定。

Duration (区块时间间隔要求) :

large enough such that two valid blocks are not submitted at the same time

间隔要足够长，避免两个合法区块被同时提交，以防止链分叉。

short enough for high processing throughput of transactions

间隔要足够短，确保交易处理吞吐量高，提高网络效率。

Equilibria (挖矿收益的均衡点)

expected mining gains balance running costs

矿工的预期收益应与其挖矿成本达到平衡。

影响均衡的因素：

Price of electricity – 电力价格 (决定挖矿成本)

Computational efficiency of hardware – 计算硬件的效率（影响单位成本）

Exchange rate – 加密货币与法币的兑换率（影响挖矿收益）

Exchange Rate vs. Difficulty (币价与难度的联动关系) :

Higher Exchange Rate (币价上涨) :

Higher total value in the blockchain

区块链中的总资产价值上升

More gains for miners

矿工的收益更多

More miners joining

更多矿工加入挖矿

Higher difficulty

系统自动提高挖矿难度

Lower Exchange Rate (币价下跌) :

Lower total value in the blockchain

区块链中的总资产价值下降

Lower gains for miners / bankruptcy

矿工收益减少甚至可能破产

Less miners' activity or leaving

矿工减少或退出网络

Lower difficulty

系统自动降低挖矿难度

27. What are Hard Forks

硬分叉是指区块链网络中的节点分裂为两个群体，各自维护一条规则不兼容（not backward compatible）的区块链。

就像软件版本升级，但老版本无法兼容新版本；结果是两条链继续并存，各自为政；所有原始持币用户，在两条链上都会“复制”一份资产（即双链资产）。

为什么会发生分叉

几乎同时解出计算难题；两个矿工几乎在同一时间出块；网络暂时存在多个合法区块
→ 临时性分叉；系统通过“最长链规则”解决。

安全攻击或故意篡改账本，有人想撤销或修改某个区块中的交易记录；比如以太坊 DAO 攻击导致的“以太坊/以太坊经典”分裂。

如何解决分叉

在 Proof of Work (PoW) 区块链中，如果两个区块被几乎同时挖出，网络就会暂时形成两个分叉的链 (forks)，即两个不同的“当前最长链”。网络中不同的节点可能暂时接收到不同的那个新区块；

接下来，当后续某个节点在其中一条链上挖出新的区块，导致该分支变得更长；

网络会根据**“最长链原则” (longest chain rule) **继续接受这条链为主链，较短的分支会被丢弃（称为 orphan block）。

28. What are Smart Contracts

智能合约

一种计算机程序或交易协议，用于根据合约或协议的条款自动执行、控制或记录事件和操作。

优势：

- 无需可信中介
- 降低仲裁成本
- 减少欺诈损失
- 降低恶意和意外异常的风险
- 减少双方之间的摩擦
- 降低道德风险
- 无需法律协议，但智能法律合约应运而生
- 图灵完备的 Solidity 语言，用于智能合约，例如以太坊

工作原理：

- 智能合约代码包含在交易中
- 执行：交易验证通过后 并添加到区块链
- 防篡改/不变性：通过拜占庭容错共识算法
- 客户端通过交易与智能合约交互
- 智能合约的交易可以调用其他交易
- 确定性流程，确保拜占庭容错

Smart Contracts

A computer program or transaction protocol to automatically execute, control or document events & actions according to the terms of a contract or agreement.

Benefits:

- ▶ No need for trusted intermediaries
- ▶ Reduction of arbitration costs
- ▶ Reduction of fraud losses
- ▶ Lower risks for malicious & accidental exceptions
- ▶ Reduced friction between parties
- ▶ Reduction of moral hazards
- ▶ No legal agreement, but smart legal contracts emerge
- ▶ Turing-complete Solidity language for smart contracts on, e.g. Ethereum



How they work:

- ▶ Smart contract code is included in transactions
- ▶ Execution: once the transaction is verified & added to the blockchain
- ▶ Anti-Tampering/Immutability: via Byzantine fault-tolerant consensus algorithms
- ▶ Clients interact with smart contract via transactions
- ▶ Transactions with a smart contract can invoke other ones
- ▶ Deterministic processes to ensure Byzantine fault-tolerance



29. What is Altcoins

比特币的继任者，尝试新算法和通胀模型。 Successors of Bitcoin with new algorithms or inflation rules.

包含模因币等，无实际创新。 Includes meme coins with little innovation.

市值曾达 4 亿美元。 Had a market cap of \$400M.

30. What is Initial Coin Offering (ICO)

ICO 是通过加密货币众筹，为初创公司融资。 ICO is crowdfunding via cryptocurrency for startups.

基于白皮书发行代币，无需传统中介。 Tokens issued based on whitepapers without intermediaries.

常用于诈骗或高风险投资。 Often used for scams or risky ventures.

2018 年市值达 8310 亿美元，随后崩盘。 Market cap hit \$831B in Jan 2018, then collapsed >85%.

31. The unit of gas is

Gwei.

32. 比特币的什么属性控制出块时间 Which two factors controls the block creation time

比特币网络设计的目标是每 10 分钟产生一个新区块。实际区块生成时间由以下两个主要因素控制：

1. Mining difficulty (挖矿难度)

- 会根据网络总算力每 2016 个区块动态调整，以保持 10 分钟的出块时间。

2. Hash rate (哈希率)

- Hash rate 是衡量一个区块链网络（比如比特币）中所有矿工计算能力的指

标，表示每秒钟可以进行多少次哈希运算，指的是整个网络的计算能力。如果哈希率提高，系统会通过提高难度来保持出块时间恒定。

三、 Design Blockchain

1. What is Conceptual Architecture

行动：由现实世界中的人/机器执行，最终在数字世界中产生一项主张

共识：矿工/铸币者就主张达成一致，最终记录在分布式账本上

分布式账本：主张被合并成区块并写入账本（交易）

代币：因行动或参与共识而获得的奖励，由链上或链下的价值来源支持

链上：存在于分布式账本上——交易和智能合约

链下：存在于共识网络或现实世界中

Conceptual Architecture

Action: performed by a human/machine in real-world resulting in a claim in the digital world

Consensus: the agreement of miners/minters on claims that result in writing on the distributed ledger

Distributed ledger: claims combined into blocks & written to the ledger
(transactions)

Token: rewards given for an action or participating in the consensus, backed up by an on-chain or off-chain source of value

On-chain: existence on the distributed ledger - transactions & smart contracts

Off-chain: existence on the consensus network or real-world

2. What is Taxonomy

Goal (目标)：捕捉区块链系统中的关键设计选择 (key design choices)；

Relies on Four Components (依赖四大组成部分)：

Distributed Ledger (分布式账本)

Consensus (共识机制)

Action (行为/使用权限)

Cryptoeconomic Design (加密经济设计)

3. What is Taxonomy-Distributed Ledger

分布式账本是用于记录行为的去中心化数据结构。

属性	选项	示例
Data structure 数据结构	blockchain, DAG, other	Bitcoin, Ethereum, IOTA, Ripple
Origin 来源	native, external, hybrid	Bitcoin (native), Aragon (external)
Address traceability 地址可追踪性	obfuscatable, linkable	Zcash (混淆), Bitcoin (可追踪)
Turing completeness 图灵完备性	yes, no	Ethereum (是), Bitcoin (否)
Storage 是否可存储额外数据	yes, no	Bitcoin (yes), IOTA (no)

4. What is Taxonomy-Consensus

共识机制是确保网络中所有参与者就记录内容达成一致的过程。

属性	选项	示例
Finality 确定性	deterministic, probabilistic	Ripple (确定), Bitcoin (概率)
Proof 共识算法	PoW, PoS, hybrid, other	Bitcoin (PoW), Ardor (PoS)
Write permission 写权限	public, restricted	Bitcoin (public), PBFT (restricted)
Validation permission 验证权限	public, restricted	IOTA (public), 某联盟链 (restricted)
Fee 是否需要手续费	yes, no	Bitcoin (yes), IOTA (no)

5. What is Taxonomy-Action

行为是指现实生活中的活动在区块链上的数字映射，比如交易、投票、发布内容等。

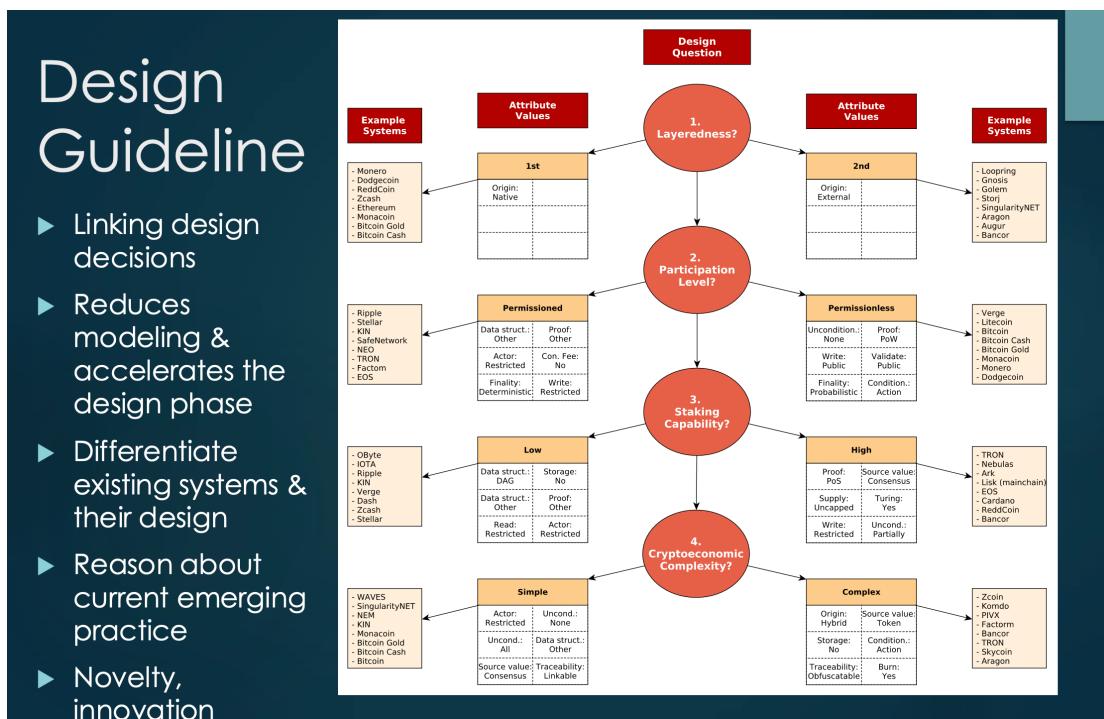
属性	选项	示例
Actor permission 行为者权限	public, restricted	Ripple (restricted), Bitcoin (public)
Read permission 读取权限	public, restricted	Zcash (restricted), Bitcoin (public)
Action fee 是否收取操作费	yes, no	Augur (yes, 服务方), Ripple (yes, 销毁手续费), Bitcoin (yes)

6. What is Taxonomy-Cryptoeconomic

Token: 在区块链系统中发行的价值单位，可作为交换媒介或计价单位。是激励机制的核心组成，影响系统的安全性、稳定性与可持续性。

设计属性	可选值	说明与示例
Supply (供应量)	capped (封顶) / uncapped (无限)	Bitcoin (2100万)、Dogecoin (无上限) 👉 封顶供应带来通缩、币值升值、更稳定
Burn (销毁机制)	yes / no	Ripple、Bitcoin: 通过销毁减少流通量 👉 有助于在需求不变时提高币值与汇率
Transferability (可转让性)	transferable / non-transferable	Bitcoin (可转让)、Akasha (不可转让声誉积分)
Creation condition (创建条件)	consensus / action / both / none	Bitcoin (通过共识奖励)、Steemit (写博客) 👉 代币是否与行为或共识直接挂钩
Unconditional creation (无条件创建)	all / partial / none	Ripple (创世时一次性分配)、Augur、Bitcoin
Source of value (价值来源)	token / ledger / consensus / action / none	Ethereum (智能合约)、Golem (计算)、Siacoin (存储) 👉 表明代币背后的经济支撑基础

7. Blockchain Design Guide



四、Consensus

1. What is Crash and Byzantine Faults

Crash 故障指进程突然停止且不会恢复。 Crash faults mean a process stops abruptly and does not resume.

Byzantine 故障包括恶意行为和错误信息传播，更具破坏性。 Byzantine faults involve malicious behavior and inconsistent information, making them more disruptive.

容错能力是指系统可容忍一定数量的故障进程仍能达成共识。 Fault tolerance refers to the system's ability to reach consensus despite some faulty processes.

2. What Are Properties of Consensus

终止性：每个正确的进程最终会决定一个值。 Termination: Every correct process eventually decides a value.

完整性：如果所有正确进程决定某个值，那么它必须是由正确进程提出的。 Integrity: Decided value must originate from a correct process.

一致性：所有正确进程必须决定相同的值。 Agreement: All correct processes must agree on the same value.

t-容错协议：在最多 t 个进程失败时，n 个进程可达成共识。 t-resilient protocol: consensus among n processes where at most t fail.

3. How to Evaluate Performance of Consensus

时间复杂度：达到共识所需的通信轮数。 Time complexity: number of communication rounds to reach consensus.

消息复杂度：传输的消息总数。 Message complexity: total number of messages.

内存使用与消息大小。 Memory usage and message size.

4. What Is Synchronous and Asynchronous Communication Model

异步模型：每个进程有自己的时钟，信息传递无确定时限。 Asynchronous: each process has its own clock; message delivery is uncertain.

同步模型：通信按轮进行，所有信息在当前轮结束前送达。 Synchronous: messages are exchanged in rounds and received within the same round.

5. What is FLP Impossibility

在一个异步系统中，如果可能存在节点崩溃（crash fault），那么不可能构造出一个既保证一致性（consistency）又保证终止性（termination）的确定性共识算法。

FLP 定理的影响

意味着：在现实的网络中（例如可能出现延迟或节点宕机），想要同时保证“不会出错”与“肯定完成”的共识算法是不可能存在的；这是分布式系统中不可能三角的一种体现；因此，区块链系统（如比特币、以太坊、Tendermint 等）都必须在一致性、终止性、容错性之间做出权衡。

比特币中的 PoW 算法保证了“最终一致性”，即随着新区块的产生，旧的区块被逐步确认，但可能暂时存在分叉；

PBFT (Practical Byzantine Fault Tolerance) 系统通过加入同步性假设和投票机制，绕过 FLP 的限制。

FLP 不可能性告诉我们，在没有时间保证的网络中，不能既保证所有节点最终都达成

决定，又保证不发生冲突。

所以所有实际系统（如区块链）都必须在一致性、可用性和容错性之间做出选择。

6. What is FLP Impossibility Triangle

在异步系统中不能同时满足以下三个条件： In asynchronous systems, the following three can't all be achieved:

一致性 (Consistency) Consistency

可用性 (Availability) Availability

分区容忍性 (Partition Tolerance) Partition Tolerance

7. How to Overcome FLP Impossibility in Practice

引入随机性，例如 Las Vegas 算法。 Introduce randomness, e.g., Las Vegas algorithms.

部分同步模型，例如 Casanova。 Use partial synchrony, e.g., Casanova.

放宽终止性，例如 Raft 选举超时机制。 Relax termination using leader election with random timeouts.

共识机制	如何应对 FLP 不可能性	保证
PoW	弃用立即终止性，接受“最终一致性”	一致性优先
PoS	多数使用投票+部分同步假设	一致性优先，最终达成
BFT类	使用超时+view change	部分同步下的一致+终止

常见算法如何绕过 FLP 不可能性：

算法	如何“绕过”FLP 不可能性
Raft	用随机超时选主，活性不是强保证，但能最终推进
Paxos	不强求活性，只保证一致性；用提案编号+重试机制最终推进
PBFT	假设网络最终同步；引入 view-change 机制确保在恶意节点存在下仍能换主并推进

8.What is BFT Algorithm

BFT 是指系统在存在部分节点出现任意故障甚至恶意行为的情况下，仍能保证整个系统达成一致的能力。

BFT 共识的挑战在于：即使有一些节点在作恶，系统也必须保证大家不会被带歪。

以下是设计一个 BFT 系统时需考虑的核心问题：

1. 容错阈值 (Fault Tolerance)

在 n 个节点中，最多允许 f 个拜占庭节点；

理论上需满足： $n \geq 3f + 1$

例如：要容忍 1 个恶意节点，至少需要 4 个节点。

2. 投票与消息广播

BFT 系统通常分为多个“阶段”或“轮次” (rounds)：

提议 (Propose) → 投票 (Vote) → 承诺 (Commit)；

所有节点需收集超过 $2/3$ 的投票才能进入下一阶段；

使用 签名或消息认证 确保节点不能双重投票。

3. 超时与 View Change

如果某轮未能收集足够投票，说明 proposer 可能出问题；

进入下一轮 (View Change)，更换 proposer；

防止网络阻塞导致死锁。

4. 消息验证机制

使用数字签名（如 Ed25519）或聚合签名（如 BLS）验证身份；

防止节点伪造其他节点的消息。

9. What is Raft Algorithm

Raft 是一种为分布式系统设计的强一致性共识算法，用于在多个节点之间就一组命令（如日志）达成一致。

Raft 将一致性过程分解为三个关键子问题：

1. Leader 选举 (Leader Election)

2. 日志复制 (Log Replication)

3. 安全性保证 (Log Commitment & Safety)

Raft 依靠一个强领导者 (leader-based) 模型，所有客户端请求都由 Leader 处理，其他节点作为跟随者 (followers) 同步其日志。

Raft 算法的运行流程

1. 节点角色

Leader：唯一的写入者，负责日志复制；

Follower：被动响应 Leader；

Candidate：用于选举期间临时竞选 Leader 的角色。

2. 状态转换逻辑

所有节点初始为 Follower；

如果 Follower 超时未收到 Leader 的心跳，则转为 Candidate，发起新一轮选举；

候选者获得超过半数选票后成为 Leader；

选出的 Leader 负责接收客户端请求，将日志条目复制给大多数节点；

一旦日志被多数确认，就被“提交（commit）”。

3. 日志复制与提交

客户端发送请求 → Leader 添加到日志 → 广播给 Follower；

Follower 返回确认 → Leader 收到多数确认后将条目提交；

所有节点应用该条目到状态机。

10.What is Paxos Algorithm

Paxos 是一种 在不可靠网络环境中，多个节点就某个值达成一致的协议。

它能容忍一部分节点宕机或消息丢失，仍能保持一致性（Safety）和一定程度的活性（Liveness）。

角色：

Proposer（提议者） 提出要达成共识的值

Acceptor（接受者） 对提议进行投票，是共识的“投票人”

Learner（学习者） 获知被批准的值（通常是客户端）

流程：

第 1 阶段：Prepare / Promise

Proposer 选择一个唯一编号的提案 n ，发送 $\text{Prepare}(n)$ 给多数 Acceptor；

每个 Acceptor：如果 $n >$ 它已响应的最大编号，则回应 $\text{Promise}(n, v')$ ，承诺不再接受更小编号的提案；并返回它之前接受过的编号最高的提案值 v' （若有）。

第 2 阶段：Accept / Accepted

Proposer 收到大多数 Acceptor 的 Promise 后：如果有返回旧值 v' ，则提议该值；

否则提议自己原本的值；发送 $\text{Accept}(n, v)$ 请求。

每个 Acceptor：如果还没承诺更高编号，接受此值并回复 $\text{Accepted}(n, v)$ 。

一旦一个值被大多数 Acceptor Accepted，即认为该值被“决定”（Chosen）。

Paxos 是如何在 FLP 不可能性定理的限制下仍实现“进展”（liveness）的

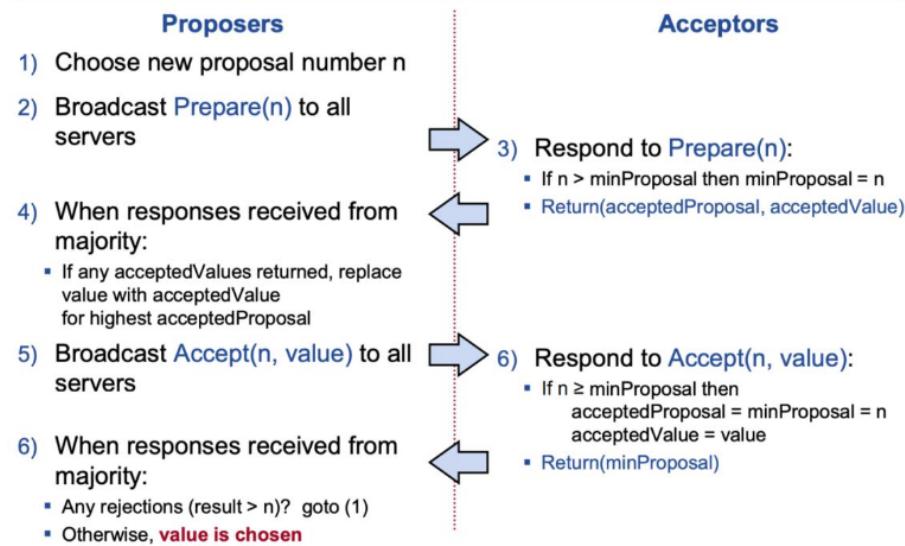
Paxos 的应对方式：

Paxos 保证一致性（safety）始终成立；

为了在异步系统中尽量实现进展（progress），Paxos 通过引入随机超时和竞选机制，使得：

- 多个 proposer 不会长期冲突；
- 最终某个 proposer 会独占提案权（以概率 1 达成 progress）；
- 这就是 B 选项所描述的策略。

Basic Paxos



Acceptors must record minProposal, acceptedProposal, and acceptedValue on stable storage (disk)

Paxos 的核心设计理念

设计点	考虑
编号系统 (n)	保证提案顺序性, 解决竞争冲突
多数派 (Quorum) 原则	任意两个多数派都有交集, 保证一致性
无 leader 模式	任何 Proposer 都可发起提案 (但这导致复杂性)
允许失败重试	网络分区、超时后可继续尝试新编号
容错能力	可容忍少数节点宕机 (通常是 $n \geq 2f + 1$)

11.What is PBFT Algorithm

PBFT (Practical Byzantine Fault Tolerance) 是一种实用的拜占庭容错共识算法，用于在存在恶意节点的前提下仍能达成一致。PBFT 是为小规模、许可型网络设计的高效共识算法，能容忍恶意节点并实现快速且最终确定的一致性，但扩展性较差。

在一个 PBFT 网络中，总共有 $n = 3f + 1$ 个节点，最多允许 f 个拜占庭错误节点。

角色：

Primary (主节点) 提出客户端请求排序方案

Backup (备份节点) 验证主节点提出的方案，并共同决定共识

Client (客户端) 提出请求，等待响应

流程：

PBFT 的共识流程（以单个请求为例）

PBFT 共识通常分为 三个阶段 + 一个提交阶段：

1. Pre-Prepare 阶段

主节点收到客户端请求后，打包为提案（请求编号 + 请求内容），广播 PRE-PREPARE 消息给备份节点。

2. Prepare 阶段

每个备份节点收到 PRE-PREPARE 后进行合法性校验；

若合法，广播 PREPARE 消息给其他所有节点。

3. Commit 阶段

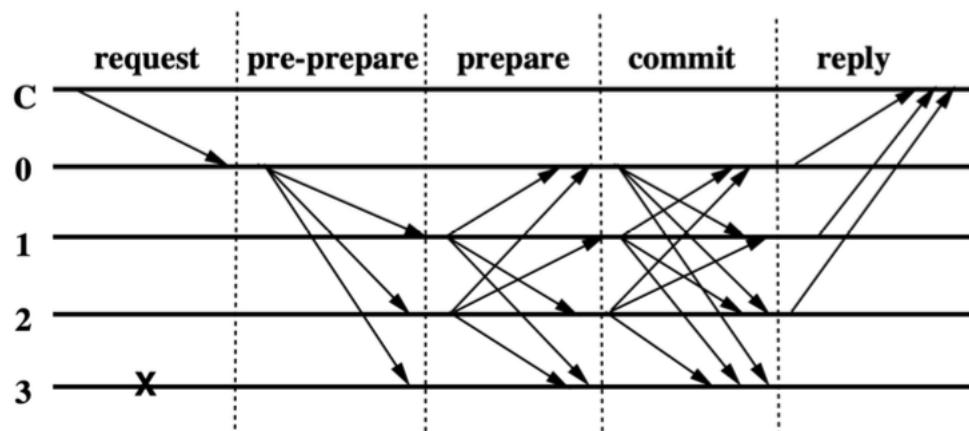
节点收到至少 $2f + 1$ 个 PREPARE 消息（包括自己），进入 COMMIT 阶段；

发送 COMMIT 消息给所有节点；一旦收到 $2f + 1$ 个 COMMIT，即认为该请求已达成共识。

4. Reply 阶段

节点执行请求，返回结果给客户端；

客户端等待来自 $f + 1$ 个不同节点的相同回复，以确认共识结果。



特点：

三阶段流程	降低拜占庭节点篡改提案的风险，逐步形成多数共识
$2f + 1$ 消息门槛	确保即便有 f 个节点恶意，正确节点仍可达成一致
确定性共识 (finality)	一旦达成共识即不可逆，不像 PoW 需“等待确认”
数字签名/消息认证码 (MAC)	防止消息伪造，验证消息来源真实性
View Change 机制	若主节点宕机或作恶，触发视图切换选出新主节点
客户端需多个节点回复确认	防止客户端被主节点单点欺骗，确保最终一致性

优点

拜占庭容错，适合联盟链/私有链

高吞吐、低延迟（适合小规模系统）

最终一致性 (Finality) 强

无需挖矿或质押，节能环保

缺点：

消息量为 $O(n^2)$ ，节点越多通信开销越大

不适用于成百上千节点的公链环境

实现较复杂，需处理主节点故障、切换等问题

主节点中心化程度高，安全性依赖切换机制

11. What Are the Types of Consensus

部分同步模型：容忍 $1/3$ 故障，有限网络延迟，可处理拜占庭故障。 Partially synchronous: $1/3$ fault tolerance, bounded delay, Byzantine support.

异步模型：不可容忍单点故障。 Asynchronous: no tolerance for even one fault.

同步模型：可容忍全部故障，但对节点行为限制大。 Synchronous: 100% fault tolerance but strict node behavior limits.

Agreement Problems, Solvability & Equivalency - Bilingual Summary

12. What are Agreement Problems

共识协议中存在以下三类关键一致性问题： There are three key types of agreement problems in consensus protocols:

1.1 可靠广播的终止问题 (Termination of Reliable Broadcast)

过程 0 向其他进程发送值 v ，需满足： Process 0 sends value v to others,

requiring:

如果过程 0 是正确的，每个正确进程都应收到 v; If process 0 is correct, every correct process must receive v;

所有正确进程收到的值必须一致。 All correct processes must receive the same value.

1.2 共识问题 (Consensus)

目标：所有正确进程必须就一个值达成一致。 Goal: All correct processes must agree on a single value.

一致性：所有正确进程的决定值必须相同； Agreement: All correct processes must decide the same value;

弱有效性：每个正确进程的输出必须是某个正确进程的输入。 Weak Validity: The decided value must be an input of some correct process.

1.3 弱交互一致性 (Weak Interactive Consistency)

每个进程有一个私有值，通过轮次通信在部分同步系统中达成共识： Each process has a private value and communicates in rounds to reach consensus in partially synchronous systems:

完整性：若正确进程发送了 v，所有进程要么接收到 v，要么什么也不接收；

Integrity: If a correct process sends v, processes either receive v or nothing;

一致性：所有正确进程在同一轮次内接收到的消息必须一致； Consistency: Correct processes must receive the same messages in the same round;

强有效性：若所有正确进程收到相同输入值，输出也必须一致； Strong Validity: If all correct inputs are the same, outputs must match;

终止性：所有进程最终必须决定一个输出值。 Termination: Every process must eventually decide an output.

12. What is Solvability & Equivalency

2.1 可解性 (Solvability)

某个问题是否可以在一定的失败数 t 与进程总数 n 下达成共识。 Solvability means whether consensus is possible under t failures among n processes.

共识问题： $t/n < 1/3$ 时有解； Consensus is solvable when $t/n < 1/3$;

若 $n < 3f$ (f 为拜占庭节点数量)，则无法达成共识。 If $n < 3f$ (where f is Byzantine faults), consensus is impossible.

口头消息模型： $t = f + 1$ 轮次才可达成共识。 In oral message model: $t = f + 1$ rounds are needed.

书面消息模型：容错能力更强。 Written message model offers stronger fault tolerance.

2.2 等价性 (Equivalency)

某些一致性问题之间可以通过转换实现。 Some agreement problems can be transformed into one another.

例如：一致性问题与弱拜占庭将军问题可以互相归约。 Example: Consensus and weak Byzantine generals problem are reducible to each other.

依据系统模型和假设条件的不同，这些问题具有等价性。 Depending on the system model and assumptions, these problems are equivalent.

五、 Cryptography

1. What is Cryptography

密码学是一种用于保护信息的技术手段。 Cryptography is a technological method for securing information.

传统密码学依赖对称密钥，即加解密使用同一密钥。 Traditional cryptography relied on symmetric keys, using the same key for encryption and decryption.

1970 年代诞生了非对称密码学，实现了使用公钥加密、私钥解密。 In the 1970s, asymmetric cryptography emerged, allowing encryption with public keys and decryption with private keys.

在区块链中，密码学保障资产归属与交易安全。 In blockchain, cryptography secures asset ownership and transaction integrity.

2. What is Symmetric Key Cryptography

对称密钥密码学使用相同的密钥加密与解密信息。 Symmetric key cryptography uses the same key to encrypt and decrypt data.

其安全性依赖密钥的保密性。 Its security depends on the secrecy of the shared key.

Kerckhoffs 原则：即使算法公开，只要密钥保密系统也应安全。 Kerckhoffs' principle: the system should be secure even if the algorithm is public, as long as the key is secret.

3. What is Caesar's Cipher, Affine Cipher, Vigenère Cipher

凯撒密码：将字母向右移动 k 位，例如 $CAT \rightarrow HFY$ 。 Caesar cipher: shifts letters right by k positions, e.g., $CAT \rightarrow HFY$.

仿射密码：加乘混合加密，使用两个密钥 I 与 k 。 Affine cipher: combines multiplication and addition using keys I and k .

维吉尼亚密码：基于多个 Caesar 密码，使用重复的密钥流。 Vigenère cipher: uses repeated key stream in a polyalphabetic substitution.

4. What Are Schemes of Symmetric Key Cryptography

Enigma 机：基于长置换序列的加密设备。 Enigma machine: long substitution sequence encryption device.

一次性密码本：密钥长度与消息相同，通过 XOR 加密。 One-time pad: uses XOR and a key of equal length to the message.

流密码：通过种子生成伪随机密钥流。 Stream ciphers: generate pseudo-random keystreams from a seed.

DES：使用异或和数据扰乱进行加密。 DES: encrypts via XOR and scrambling.

AES：替代 DES 的现代加密标准。 AES: modern standard replacing DES.

5. What is the Application of Asymmetric Key Cryptography

公钥加密：发送方用接收方公钥加密。 Public key encryption: sender uses recipient's public key.

数字签名：私钥签名，公钥验证。 Digital signature: signed with private key, verified with public key.

共享秘密：基于双方公钥和私钥生成共同密钥。 Shared secret: derived from both parties' public and private keys.

区块链应用：确保转账、认证和所有权。 Blockchain usage: ensures transfers, authentication, and ownership.

6. What is RSA Algorithm and the Calculation Process

RSA 是非对称密码系统，由 Rivest、Shamir 和 Adleman 于 1977 年提出。 RSA is an asymmetric cryptographic system proposed in 1977 by Rivest, Shamir, and Adleman.

它使用大素数计算模指数运算。 It uses large prime numbers for modular exponentiation.

6.1 RSA 密钥对生成流程：

步骤 1：选择两个大质数 p 和 q

- $p \neq q$, 随机选取两个足够大的质数（比如 512 或 1024 位）
- 这是 RSA 安全性的基础

步骤 2：计算 $n = p \times q$

- n 是模数 (modulus)，将用于 公钥和私钥
- n 的位数决定了加密强度 (如 2048-bit)

步骤 3：计算欧拉函数 $\phi(n) = (p - 1)(q - 1)$

- 欧拉函数表示与 n 互质的整数个数
- 在后续步骤中用于求解私钥

步骤 4：选择公钥指数 e

- e 是公钥的一部分，满足：
 - $1 < e < \phi(n)$
 - $\gcd(e, \phi(n)) = 1$ (与 $\phi(n)$ 互质)
- 通常选择常用值，如 $e = 65537$ (性能与安全的折中)

步骤 5：计算私钥指数 d

- d 是满足以下条件的整数： $(d \times e) \bmod \phi(n) = 1$, d 是 e 关于 $\phi(n)$ 的模反元素
- 可使用扩展欧几里得算法 (Extended Euclidean Algorithm) 计算

6.2 RSA 签名方案

1. 密钥准备（与加密相同）

Alice (签名者) 拥有：

- 私钥： (n,d)
- 公钥： (n,e)

其中：

- $n=p \times q$: 两个大质数的乘积
- d : 满足 $(d \times e) \bmod \phi(n) = 1$

2. 签名阶段（由 Alice 完成）

假设消息为 M , 流程如下：

步骤 1 对消息进行哈希

$$H=Hash(M)$$

使用标准哈希函数 (如 SHA-256) 生成定长摘要。

步骤 2 使用私钥对哈希签名

$$S=H^d \bmod n \quad (H \text{ 的 } d \text{ 次方模 } n)$$

这个 S 就是签名，发送给 Bob。

3. 验证阶段（由 Bob 完成）

Bob 收到 M 和签名 S , 用 Alice 的 公钥 (n,e) 验证：

步骤 3 对签名进行“解密”

$$H'=S^e \bmod n \quad (S \text{ 的 } e \text{ 次方模 } n)$$

步骤 4 重新哈希消息

$$H = Hash(M)$$

步骤 5 比较是否相等

如果 $H'=H \Rightarrow$ 签名合法

7. What is ElGamal Algorithm

ElGamal 算法由 Taher Elgamal 于 1985 年提出，是基于离散对数问题的加密方法。 The ElGamal algorithm was introduced in 1985 by Taher Elgamal and is based on the discrete logarithm problem.

它比基础 RSA 更安全，适用于短消息。 It is more secure than basic RSA and

works well for short messages.

它是将 Diffie-Hellman 密钥交换方法转换为加密算法的产物。 It transforms the Diffie-Hellman key exchange method into an encryption algorithm.

8. What is the Calculation Process of ElGamal

8.1 加密解密

1. 密钥生成 (Key Generation)

选择一个大的素数 p 和一个生成元 g ($g < p$ 且为 \mathbb{Z}_p^* 的原根) :

- 私钥 (Private key) : 选择一个随机整数 $x \in [1, p - 2]$
- 公钥 (Public key) : 计算 $y = g^x \pmod{p}$

所以:

- 公钥为: (p, g, y)
- 私钥为: x

2. 加密 (Encryption)

发送者 Alice 想加密消息 $m \in \mathbb{Z}_p^*$ 给 Bob (Bob 的公钥是 (p, g, y)):

1. 选取一个随机整数 $k \in [1, p - 2]$

2. 计算:

- $c_1 = g^k \pmod{p}$
- $c_2 = m \cdot y^k \pmod{p}$

3. 密文为: (c_1, c_2)

3. 解密 (Decryption)

Bob 使用自己的私钥 x 解密密文 (c_1, c_2) :

1. 计算:

- $s = c_1^x \pmod{p}$
- s^{-1} 是模 p 意义下 s 的乘法逆元

2. 还原明文:

$$m = c_2 \cdot s^{-1} \pmod{p}$$

设：

- $p = 17, g = 3$
- Bob 选私钥 $x = 15$, 则公钥 $y = 3^{15} \pmod{17} = 6$
- Alice 要加密 $m = 13$, 选随机 $k = 7$

加密：

- $c_1 = 3^7 \pmod{17} = 11$
- $c_2 = 13 \cdot 6^7 \pmod{17} = 13 \cdot 7 \pmod{17} = 6$
- 密文为 $(11, 6)$

解密：

- Bob 计算 $s = 11^{15} \pmod{17} = 7, s^{-1} = 5$ (因为 $7 \cdot 5 \pmod{17} = 1$)
- 明文 $m = 6 \cdot 5 \pmod{17} = 13$

还原成功！

8.2 数字签名

1. 密钥生成 (Key Generation)

见上面

2. 签名生成 (Signing)

签名者想对消息 M 签名 (通常对消息先求哈希: $h = H(M)$):

1. 选择一个随机数 $k \in \{1, \dots, p-2\}$, 使得 $\gcd(k, p-1) = 1$

2. 计算:

- $r = g^k \pmod{p}$
 - $s = k^{-1}(h - r) \pmod{p-1}$
- 这里 k^{-1} 是 k 在模 $p-1$ 下的乘法逆元

3. 签名是 (r, s)

3. 签名验证 (Verification)

验证者已知公钥 (p, g, y) 和签名 (r, s) , 以及消息哈希 $h = H(M)$:

验证等式是否成立：

$$y^r \cdot r^s \pmod{p} = g^h \pmod{p}$$

如果成立, 签名有效; 否则, 签名无效。

假设：

- $p = 467, g = 2, x = 127$ (私钥)
- 公钥 $y = g^x \pmod{p} = 2^{127} \pmod{467} = 323$
- 消息哈希 $h = 123$
- 随机 $k = 31$, 满足 $\gcd(31, 466) = 1$

签名者计算：

1. $r = g^k \pmod{p} = 2^{31} \pmod{467} = 5$
2. $k^{-1} \pmod{(p-1)} = 31^{-1} \pmod{466} = 15$ (假设)
3. $s = 15 \cdot (123 - 127 \cdot 5) \pmod{466} = \dots = 172$

签名为 $(r, s) = (5, 172)$

验证者验证：

$$y^r \cdot r^s \pmod{p} \stackrel{?}{=} g^h \pmod{p}$$

如果成立，签名通过。

9. What is Schnorr Algorithm

Schnorr 签名算法由 Claus Schnorr 于 1989 年提出，是 ElGamal 签名的扩展版本。The Schnorr signature algorithm was proposed by Claus Schnorr in 1989 as an extension of ElGamal. 它专利保护至 2008 年，具有更高的签名效率。It was patented until 2008 and is more efficient in signature generation. 该算法适用于数字签名。The algorithm is suitable for digital signatures.

数字签名流程：

1. 密钥生成 (Key Generation)

选择如下参数：

- 大素数 p, q (其中 $q \mid (p-1)$)
- 生成元 $g \in \mathbb{Z}_p^*$, 使得 $g^q \pmod{p} = 1$
- 私钥：随机选一个整数 $x \in \mathbb{Z}_q$
- 公钥： $y = g^x \pmod{p}$

输出：

- 私钥： x
- 公钥： (p, q, g, y)

2. 签名生成 (Signing)

对消息 M 签名，流程如下：

1. 选一个随机数 $k \in \mathbb{Z}_q$
2. 计算：
 - $r = g^k \pmod p$
 - $e = H(r \parallel M) \pmod q$
其中 H 是哈希函数，如 SHA256
3. 计算签名：
$$s = (k + x \cdot e) \pmod q$$
4. 输出签名为： (e, s)

3. 签名验证 (Verification)

验证者知道签名 (e, s) 、消息 M 、公钥 y ，执行：

1. 计算：
$$r' = g^s \cdot y^{-e} \pmod p$$
2. 计算：
$$e' = H(r' \parallel M) \pmod q$$
3. 若 $e' = e$ ，则签名有效。

$r' \parallel M$ 的含义：把 r 和 M 拼接起来 $\rightarrow "123456hello"$. 对 r 和消息 M 进行拼接，

然后取哈希，最后对 q 取模，得到签名中的 e

10. What is Diffie-Hellman Algorithm

Diffie-Hellman 算法由 Whitfield Diffie 和 Martin Hellman 于 1976 年提出。The Diffie-Hellman algorithm was proposed by Whitfield Diffie and Martin Hellman in 1976.

它是一种密钥交换协议，基于离散对数问题。It is a key exchange protocol based on the discrete logarithm problem.

其安全性要求使用 2048 位以上的素数，或采用椭圆曲线密码学。For security, it requires primes of at least 2048 bits or elliptic curve cryptography.

Diffie-Hellman 密钥交换 (Diffie-Hellman Key Exchange) 是一种经典的公开密钥交换协议，用于在不安全的信道中，让两个通信方安全地协商出一个共享的密钥，即使攻击者能够监听所有通信内容，也无法得知这个密钥。

Diffie-Hellman Key Exchange 的流程：

公共参数：

一个大质数 p ，一个生成元 g ($g < p$)。 p 和 g 是公开的，所有人都知道。

步骤	Alice (客户端)	Bob (服务端)
1	选择私钥 a (随机)	选择私钥 b (随机)
2	计算 $A = g^a \bmod p$	计算 $B = g^b \bmod p$
3	将 A 发送给 Bob	将 B 发送给 Alice
4	计算共享密钥 $s = B^a \bmod p$	计算共享密钥 $s = A^b \bmod p$

公共参数: $p = 23$, $g = 5$

Alice 选私钥 $a = 6 \rightarrow A = 5^6 \bmod 23 = 8$

Bob 选私钥 $b = 15 \rightarrow B = 5^{15} \bmod 23 = 2$

交换后:

Alice 得到 $B=2 \rightarrow 2^6 \bmod 23 = 18$

Bob 得到 $A=8 \rightarrow 8^{15} \bmod 23 = 18$

六、 Security and Privacy

1. What is double-spending

Double-spending (双重支付) 是指一个用户尝试用 同一笔加密货币 向 多个接收方 进行支付 —— 也就是说，试图“一币多花”。

例如：

Alice 给 Bob 发了 1 BTC

同时又向 Charlie 发了这 同样的 1 BTC

如果区块链系统没能正确识别并阻止，两个收款方都可能以为他们收到了钱 —— 这就是双重支付攻击

2. What is Sybil attack

Sybil 攻击 (Sybil Attack) 是一种分布式网络攻击方式，攻击者通过创建大量伪造身份 (虚假节点) 来获得不成比例的控制权，破坏系统的信任机制和共识过程。

如何防御 Sybil 攻击

方法	思路
Proof of Work (工作量证明)	创建身份需要计算资源，不易伪造
Proof of Stake (权益证明)	身份与经济利益挂钩，提高攻击成本
身份验证 (许可链)	控制加入节点的身份验证机制
信任图/声誉系统	根据行为建立信任权重

3. What is MetaMask wallet

MetaMask 是一种浏览器插件钱包，

以太坊钱包（如 Metamask）是去中心化的，你完全掌控自己的私钥。私钥一旦丢失，就无法再访问该钱包中的资产，也无法通过客服或节点恢复。

常见的安全威胁主要是来自用户终端设备（浏览器或操作系统）的恶意软件或钓鱼攻击。攻击者常通过以下方式获取用户私钥或助记词：假冒 MetaMask 网站或插件；浏览器中注入恶意脚本；利用木马或键盘记录器窃取密码或助记词。

七、 Applications

1. 资产通证化的好处 What is a primary benefit of tokenization of real-world assets on a blockchain

实现资产的“可分割所有权”（fractional ownership），以及通过智能合约进行更容易的交易和转让

1. What are Unmanned Aerial Vehicles (UAVs)

无人机（UAV）是执行搜索、农业、递送、监测和电信等任务的自主飞行设备。

Unmanned Aerial Vehicles (UAVs) are autonomous flying devices used in search, agriculture, delivery, monitoring, and telecommunications.

其任务元素包括：无人机、本地基站、传感数据、通信交互、飞行与环境感知。

Mission elements include UAVs, base stations, sensor data, communication, and flight/environment interaction.

2. Why Blockchain for UAVs

区块链为数据共享、充电和交付等提供安全交易。Blockchain enables secure transactions for data sharing, charging, and deliveries.

提供容错、可追踪性和交互式群体操作能力。It supports fault tolerance, traceability, and resilient swarm operations.

3. What is Classification of UAVs

按结构：多旋翼、固定翼、混合翼。By structure: Multi-rotor, Fixed wing, Hybrid wing.

按重量：微型 ($\leq 100\text{g}$)、非常小 ($\leq 2\text{kg}$)、小型 ($\leq 25\text{kg}$)、中型 ($\leq 150\text{kg}$)、大型 ($> 150\text{kg}$)。By weight: Micro ($\leq 100\text{g}$), Very Small ($\leq 2\text{kg}$), Small ($\leq 25\text{kg}$), Medium ($\leq 150\text{kg}$), Large ($> 150\text{kg}$).

4. What is UAV Structure

组成包括飞控器、FPV 摄像头、锂电池和其他传感器。Components include flight controller, FPV camera, LiPo battery, and other sensors.

5. What is UAV Battery Charging

飞行距离受电池容量限制，重量影响速度、高度和负载。Flight range is limited by battery; weight affects speed, altitude, and payload.

无线充电方式：电容、电感、磁共振、激光、微波等。 Wireless charging: capacitive, inductive, magnetic resonance, laser, microwave.

非无线方式：换电池、太阳能、气流滑翔。 Non-wireless: battery swap, solar power, gust soaring.

6. What are Blockchain UAV Applications

典型应用包括：供应链管理、分布式存储、协同服务、安全、边缘计算。

Applications include: supply-chain management, decentralized storage, coordinated services, security, and edge computing.

7. Supply-chain management

挑战：多人工库存管理成本高、易出错。 Challenge: Human-involved inventory is costly and error-prone.

解决方案：无人机自动扫描产品（如 RFID），记录至区块链。 Solution: UAVs scan products (e.g., RFID) and store data on blockchain.

区块链作用：实现数据透明、安全存储与验证，自动任务执行。 Blockchain: Enables transparent, secure storage and verification, automates task execution.

8. Coordinated UAVs

挑战：缺乏可靠共享知识、安全交互与资源分配。 Challenge: Lack of reliable shared knowledge and secure interactions.

解决方案：感知即服务、定位验证、安全负载分配与同步。 Solution: Sensing as a service, proof of location, secure load distribution, synchronization.

区块链作用：安全数据交换、链上坐标存储、行为验证。 Blockchain: Enables secure data exchange, coordinate storage, and behavior verification.

9. Decentralized Storage

挑战：无人机存储能力弱、效率低。 Challenge: UAVs are weak and inefficient for storage.

解决方案：使用地面站作为安全分布式存储。 Solution: Use ground stations for secure distributed storage.

区块链作用：提升安全性并激励节点交易。 Blockchain: Increases security and incentivizes drone-ground transactions.

10. Security

挑战：保障无人机自主性与高密度飞行安全。 Challenge: Maintain autonomy and secure high-density UAV operations.

解决方案：检测劫持、中毒数据、避碰、数据保护。 Solution: Detect hijacks, data poisoning, avoid collisions, secure data.

区块链作用：提供无需信任的共识机制和飞行调度智能合约。 Blockchain: Enables trustless consensus and flight scheduling via smart contracts.

11. Edge Computing

挑战：实现低延迟、隐私保护的数据处理。 Challenge: Achieve low-latency and privacy-preserving data processing.

解决方案：边缘缓存系统辅助路径优化。 Solution: Edge caching system supports optimal path planning.

区块链作用：确保可靠通信。 Blockchain: Ensures reliable communication.

12. What is UAVs Consensus Model

无人机需对感知频率等达成共识，采用工作量证明。 UAVs reach consensus on sensing frequencies using proof of work.

参与协议的无人机可获得奖励。 Compliant drones receive rewards.

13. What is Energy-aware Swarm Consensus

基于能量的群体共识模型考虑无人机的飞行距离、电池容量、功耗和哈希能力。

Energy-aware swarm consensus considers flight distance, battery capacity, power consumption, and hashing power.

无人机的能耗要小于电池容量的 20%。 Drone's consensus energy must not exceed 20% of battery capacity.

吞吐量由共识次数与飞行时间决定。 Throughput is determined by consensus frequency and total flight time.

该模型优化了能源分配与任务效率。 This model optimizes energy allocation and mission efficiency.

1. What is Digital Consensus in Physical World

在物理世界中达成数字共识尤为关键，尤其是与地理位置相关的共识。 Achieving digital consensus in the physical world is critical, especially concerning geolocation.

传统 GPS 存在不可靠、信号弱、能耗高和易欺骗的问题。 Traditional GPS is unreliable, has weak signals, high energy costs, and is vulnerable to spoofing.

2. What is Secure Localization & Verification

使用分布式信号信息进行定位，包括传输距离与角度等参数。 Localization uses distributed signal information like transmission distance and angles.

分类：基于节点/基础设施，基于距离/非距离。 Types: node-centric vs. infrastructure-centric; range-based vs. range-free.

抗攻击协议包括：挑战-响应协议和信号飞行时间测量。 Security includes challenge-response protocols and time-of-flight measurements.

3. What is Positioning Systems

定位技术包括三角测量、三边测量、多边测量。 Positioning techniques include triangulation, trilateration, and multilateration.

关键参数包括信号到达时间 (ToA)、飞行时间 (ToF)、到达时间差 (TDoA)、信号强度 (RSSI)。 Key parameters: Time of Arrival (ToA), Time of Flight (ToF), Time Difference of Arrival (TDoA), Received Signal Strength Indicator (RSSI).

高精度定位需同步时钟。 Accurate positioning requires synchronized clocks.

4. What is Clock Synchronization

本地时钟会漂移，需周期性重新同步。 Local clocks drift and need periodic resynchronization.

拜占庭容错同步方案可应对不同类型的攻击行为。 Byzantine fault-tolerant synchronization can address various adversarial behaviors.

5. What is Spatio-temporal Evidence

空间-时间证据结合定位机制、传感器融合、异常检测、社交见证等。 Spatio-temporal evidence integrates localization, sensor fusion, anomaly detection, and social witnessing.

区块链能提供去中心化信任、自我治理、隐私保护和激励机制。 Blockchain offers decentralized trust, self-governance, privacy mechanisms, and incentive models.

6. What is Localization Infrastructure

去中心化基础设施包括 LoRa WAN 信标和社区众包定位服务。 Decentralized infrastructure includes LoRa WAN beacons and crowd-sourced location services.

可通过加密货币质押注册验证节点（如 FOAM token）。 Token-curated registries (e.g., FOAM token) allow registration and validation of anchors.

7. What is Zone Formation

通过广播信标信号发现邻近节点并同步时钟，形成共识时区。 Nodes discover each other via beacon signals and synchronize clocks to form a consensus zone.

参与者可通过 FOAM token 获得奖励。 Participants are rewarded with FOAM tokens.

8. What is Verifying Presence Claims

位置客户提交位置信息，锚点通过信号距离验证。 Clients submit presence claims, anchors verify via signal distances.

本地链上存储共识结果，支持区域间协同验证。 Local blockchain stores consensus; cross-zone verification ensures synchronization.

9. What is Publishing Proofs of Location

验证后的位置被写入以太坊区块链并公开。 Verified location proofs are recorded on Ethereum blockchain and made public.

形成共识地图供应用访问。 A consensus map is formed and accessible by decentralized apps.

10. What is Proof of Witnessed Presence

见证存在需验证者具备距离要求、质押币数量和信誉。 Validators are judged by physical proximity, stake amount, and reputation.

验证需签署所有同步消息，无惩罚机制。 Validators sign all sync messages; no

slashing mechanism.

11. What is Incentivizing Witnessed Presence

使用代币奖励参与见证的人和基础设施。 Utility tokens reward participants and infrastructure for proving presence.

鼓励社交证明、资源使用、扩展覆盖范围和准确度。 Encourages social proof, resource use, and better coverage and accuracy.

抵抗女巫攻击，通过进入/退出/存在成本。 Resists Sybil attacks using entry, exit, and presence costs.

12. What is Witnessed Presence of Accident Risk

结合实地事故数据与目击者反馈生成风险热图。 Combines real-world accident data with witness feedback to generate risk maps.

观察到实证风险与目击者评估高度一致。 High alignment observed between empirical risk and witnessed presence assessments.

八、 Efficiency

1.What is the primary trade-off introduced by increasing the block size in a blockchain network

更高的吞吐量 (throughput)：每个区块可以容纳更多交易，提升每秒处理交易的能力 (TPS)。

增加中心化风险 (centralization risk)

其他影响：

扩大区块不会降低能耗，反而可能使传播变慢

增大区块不直接增加安全性，且可能削弱去中心化

增大区块不影响数据完整性 (integrity)

2. In a blockchain network with a transaction size of 250 bytes, a block size of 2MB, and a block time of 10 minutes, what is the theoretical transaction throughput of the network.

单笔交易大小：250 字节

区块大小： $2\text{MB} = 2 \times 1024 \times 1024 = 2,097,152$ 字节

区块生成时间：10 分钟 = 600 秒

Throughput = $2097152/250/600$

3. A blockchain network halves its mining reward every 210,000 blocks. If the initial block reward was 50 coins, what will be the mining reward after 635,000 blocks have been mined

区块链每 210,000 个区块减半一次，初始奖励为 50 coins，那么 635,000 个区块后的奖励是多少

6.25

4. An Ethereum transaction requires 30,000 gas to execute, and the gas price is set to 20 Gwei. What is the cost of this transaction in Ether

30,000 gas×20 Gwei=600,000 Gwei

1 ETH=1,000,000,000 Gwei=10⁹ Gwei

600,000/1,000,000,000 = 0.0006 ETH

5.

10. Bob wants to estimate the energy consumption of a fleet of UAVs (unmanned aerial vehicles) while using blockchain technology for data sharing and storage. The fleet consists of 20 UAVs, each with a flight time of 30 minutes for data collection and a power consumption of 300 watts per hour. Bob must charge UAVs for 30 minutes before each flight of data collection. In addition, the blockchain consensus mechanism used in this system requires 600,000 computations per transaction and has a total of 3,000 transactions during each flight.

- I.He can control each UAV to perform ____ flights per day.
- II.Assuming each computation consumes 1×10^{-5} Joules (1W = 1 J/s), how much energy does the fleet consume for blockchain computation per flight? ____ kWh.
- III.How much energy does the fleet consume per day? ____ kWh.

10. 24, 0.005, 72.12

- Total energy consumption of 3,000 transactions = 6 Joules × 3,000 = 18,000 Joules = 0.005 kWh
- Energy consumed per flight = 150 watt-hours × 20 + 0.005 kWh = 3.005 kWh
- Energy consumption per day = 3.005 kWh × 24 = 72.12 kWh]

$$1 \text{ kWh} = 3,600,000 \text{ J} = 3.6 \times 10^6 \text{ J}$$

九、 Solidity

1. 修饰符：

函数默认修饰符为 **public**

变量默认修饰符为 **internal**

修饰符	合约内部访问	合约外部访问	子合约访问
private	是	否	否
internal	是	否	是
public	是	是	是
external	否 (不能直接调用)	是	是 (外部调用)

2. Solidity data types

类型	说明
<code>bool</code>	布尔类型, 只能为 <code>true</code> 或 <code>false</code>
<code>uint / uint8 ~ uint256</code>	无符号整数, <code>uint</code> 等价于 <code>uint256</code> , 步长为 8 位
<code>int / int8 ~ int256</code>	有符号整数, <code>int</code> 等价于 <code>int256</code>
<code>address</code>	以太坊地址类型, 20 字节长
<code>address payable</code>	可接收 ETH 的地址类型 (如用于 <code>transfer()</code>)
<code>bytes1 ~ bytes32</code>	固定长度字节数组, 用于存储二进制数据
<code>enum</code>	枚举类型, 用户自定义的一组常量
<code>byte</code> (已废弃)	原始旧类型, 建议使用 <code>bytes1</code> 替代

类型	说明
<code>string</code>	字符串类型, UTF-8 编码的动态数组
<code>bytes</code>	动态字节数组, 用于存储任意大小的二进制数据
<code>array</code>	数组, 可分为静态数组和动态数组 (如: <code>uint[]</code> 、 <code>uint[10]</code>)
<code>mapping</code>	映射类型, 用于类似哈希表的结构 (如 <code>mapping(address => uint)</code>)
<code>struct</code>	用户自定义结构体类型
<code>function</code>	函数类型, 可用于函数指针调用 (如 <code>function(uint) external returns (bool)</code>)

类型	说明
<code>contract</code>	合约类型, 可用于定义接口或实例化其他合约
<code>msg , tx , block</code>	全局变量的上下文类型, 包含链上交易信息
<code>function , modifier</code>	函数类型、修饰符类型
<code>storage , memory , calldata</code>	数据位置修饰符, 决定变量存储位置和生命周期

3. What does the ‘revert’ keyword do in Solidity

Terminates the function execution and reverts any changes made during the call.

4. Array

在 Solidity 中, 如果你有一个 `array a = [1, 2, 3, 4];`, 并且想要删除最后一个元素 (例如 4), 并且希望数组长度减少, 正确的方法是:

```
a.pop();
```

5. Mapping

在 Solidity 中，mapping 默认会对所有键返回默认值（例如 uint 是 0，bool 是 false）。所以即使 msg.sender 从未存过钱，accounts[msg.sender] 也不会导致报错，而是返回一个默认的 Account 结构体。

6. Payable

在 Solidity 中，如果你希望一个函数能够接收 Ether，那你必须使用 payable 关键字进行标注。没有 payable 的函数，即使外部调用时附带了 Ether，也会拒绝转账。

withdraw 函数本身没有 payable 修饰符也依然可以调用 transfer() 向外部地址发送 Ether，这是完全合法的。

1. payable 的意义是 允许该函数“接收”Ether

即调用这个函数时，可以在交易中附带 msg.value。如果函数不打算接收 Ether，就不需要加 payable。

2. 向其他地址发送 Ether 不需要 payable 修饰符

调用 payable(address).transfer(amount) 是向别人发送 Ether。关键是目标地址必须是 payable 类型，而函数本身并不需要是 payable。

msg.sender.transfer(amount); 错误。编译器会报错，因为 msg.sender 默认是 address 类型，不具备 .transfer() 方法。你必须先强制转换为 payable 类型：

7. Require

```
require(amount <= accounts[msg.sender].balance, "Insufficient balance");
```

当请求提取的 amount 大于当前余额时，require 会触发错误并 revert 整个交易。

8. 获取合约地址

在使用 Truffle 框架部署合约后，获取已部署合约实例的标准方式是：

```
simpleStorage.deployed().then(function(ins) {  
    con = ins;  
});
```

十、 Questions

零知识证明是一种技术，其最初的设计目的是允许一方（证明者）向另一方（验证者）证明他们知道某些信息，而无需透露这些信息本身。现在，它也可以用于证明计算的正确性，而无需透露输入或中间步骤。证明者可以将计算结果连同简洁的证明一起提供给验证者，验证者可以在短时间内使用该证明来验证结果，而无需重复计算。

(a) 鉴于您对区块链技术的理解，您认为零知识证明可以应用于区块链吗如果可以，它可以解决区块链系统的哪些问题，以及如何实现这些解决方案

Zero-knowledge proof is a technique that is originally designed to allow one party (the prover) to prove to another party (the verifier) that they know some information without revealing the information itself. Now it can also be used to prove the

correctness of a computation without revealing the inputs or the intermediate steps. The prover can provide the computation result along with a sufficient proof to the verifier, who can then verify the result using the proof in a short time without redoing the computation.(a) Given your understanding of blockchain technologies, do you think zero-knowledge proof can be applied to blockchain If so, what problems can it solve for blockchain systems and how to implement the solutions

是的，零知识证明（Zero-Knowledge Proof, ZKP）可以有效应用于区块链技术，并能解决当前区块链系统中的多个关键问题：

ZKP 能为区块链系统解决哪些问题：

1. 隐私保护：

- 传统区块链（如 Bitcoin 或 Ethereum）上的交易记录是完全公开的。
- ZKP 允许用户在不泄露交易具体信息（如发送者、接收者、金额等）的前提下，证明交易是合法的。
- 例如：[Zcash](#) 使用 zk-SNARKs 来实现交易隐私。

2. 可扩展性：

- 验证一个零知识证明的计算量通常比直接验证原始计算要低。
- zk-Rollup 等技术可以将成千上万笔交易在链下处理，仅提交一个简短证明到链上，从而减少主链负担。
- 示例项目：StarkNet、zkSync。

3. 身份验证安全性：

- 用户可以在不泄露个人详细信息的前提下，证明自己满足某些条件（例如年满 18 岁、持有某个证书）。
- 这对构建去中心化身份（DID）系统和隐私保护的 KYC（实名认证）非常有帮助。

如何在区块链中实现零知识证明：

1. 使用已有的 ZK 工具库：

- ZoKrates：适用于 Solidity 智能合约的 zk-SNARK 工具。
- Circom + SnarkJS：适用于自定义电路和链下证明生成。

2. 在链上部署验证合约：

- 使用 zk 编译器生成的验证合约部署到区块链上，链下生成证明，链上快速验证即可。

3. 应用于实际场景：

- 隐私投票系统
- 隐私保护的 DeFi（去中心化金融）协议
- 零知识资产交换（ZK Swap）

供应链管理是对商品和服务流动的管理，涉及原材料、在制品库存和成品从原产地到消费地的运输和储存。有效的供应链管理可以帮助企业降低成本、减少浪费、提高质量并提升客户满意度。

(a) 最近的研究表明，区块链技术可以创新供应链管理。那么，在供应链管理中使用区块链有哪些潜在优势？这些优势是如何实现的？

Supply chain management is the management of the flow of goods and services involving the movement and storage of raw materials, of work-in-progress inventory, and of finished goods from point of origin to point of consumption. Effective supply chain management can help companies reduce costs and wastes, improve quality, and increase customer satisfaction. (a) Recent research suggests that blockchain technologies can innovate supply chain management, what are the potential benefits of using blockchain in supply chain management, and how are they achieved?

区块链技术通过引入透明性、可追溯性和数据安全性，为供应链管理带来了多项潜在创新。这些优势主要包括：

1. 提高透明度

区块链提供一个去中心化且不可篡改的账本，使供应链各方都能实时访问相同的数据，减少信息不对称，增强各参与者之间的信任。

实现方式：

将每一笔供应链交易记录在区块链上；

设定权限，允许相关方按需查看数据更新。

2. 增强可追溯性

区块链可记录产品从原材料到最终交付的全过程，帮助快速定位问题来源，支持质量、安全或道德标准合规。

实现方式：

将产品信息（如批次号、产地、时间戳等）写入区块链；

结合物联网传感器或二维码，实现实时跟踪。

3. 降低欺诈和假冒风险

由于区块链上的记录无法被随意篡改，恶意行为者难以伪造记录或引入假货，增强了系统的安全性。

实现方式：

使用加密签名验证每笔交易的合法性；

多方验证产品来源，确保数据可信。

4. 提升效率与自动化水平

通过智能合约，区块链可在满足特定条件时自动执行操作（如付款或发货），减少人为干预，降低流程成本。

实现方式：

在各方之间部署智能合约，例如根据温度监控或运输确认自动付款；

减少纸质审计与海关流程，实现自动化核查。

5. 增强消费者信任与可持续性追踪

终端消费者可通过扫描二维码等方式，验证产品的来源和可持续性信息（如是否为有机种植或公平贸易产品）。

实现方式：

将部分数据开放给消费者查阅；

将环保与道德认证信息上链，实现透明可信的展示。

Blockchain technology offers several potential benefits for supply chain management by introducing transparency, traceability, and security to the movement of goods and data across multiple parties. The main benefits include:

1. Improved Transparency

Blockchain provides a decentralized and immutable ledger where all participants can access the same data in real time. This reduces information asymmetry and increases trust among suppliers, manufacturers, distributors, and customers.

Achieved by:

- Recording each supply chain transaction on the blockchain.
- Enabling permissioned access to view updates along the chain.

2. Enhanced Traceability

Blockchain allows every step of the product journey—from raw materials to final delivery—to be recorded and verified. This helps identify the source of defects or delays and supports compliance with safety or ethical standards.

Achieved by:

- Attaching product data (e.g., batch number, origin, timestamp) to blockchain entries.
- Linking IoT sensors or QR codes to blockchain for real-time status.

3. Reduced Fraud and Counterfeiting

Since blockchain records cannot be tampered with, it is harder for malicious actors to manipulate records or introduce fake products into the system.

Achieved by:

- Using cryptographic signatures to validate each transaction.
- Ensuring provenance data is verified by multiple parties.

4. Improved Efficiency and Automation

Smart contracts on the blockchain can automatically execute actions (e.g., payments, shipping updates) when predefined conditions are met, reducing manual intervention and paperwork.

Achieved by:

- Deploying smart contracts between parties for conditions like delivery confirmation or temperature control.
- Streamlining audits and customs clearance with real-time verification.

5. Better Customer Trust and Sustainability Tracking

End consumers can verify product origin and sustainability practices (e.g., organic farming, fair trade) by scanning a code linked to blockchain records.

Achieved by:

- Making select data available to customers via blockchain-enabled interfaces.
- Recording environmental and ethical certifications immutably.