Professional Certificate in Blockchain and AI Governance

Blockchain Fundamentals and Distributed Ledger Technology

Address – a string of characters that identifies a destination on a blockchain where assets can be sent or received. Related terms: Public key, wallet, transaction
An address is typically derived from a public key using a cryptographic hash function, producing a shorter, user-friendly identifier (e.G., “0XAB12…”). In Bitcoin, addresses start with “1” or “bc1”, while Ethereum uses the “0x” prefix. **Practical application:** Users share their address to receive cryptocurrency payments, to participate in token sales, or to receive rewards from decentralized applications. **Challenges:** Human error when copying long strings, phishing attacks that substitute look-alike characters, and the need for privacy-preserving techniques such as stealth addresses to hide transaction links.

Block – a collection of transactions bundled together, sealed with a cryptographic hash, and added to the blockchain. Related terms: Chain, block header, merkle root
Each block contains a header (including the previous block’s hash, timestamp, and nonce) and a body that lists validated transactions. The block header’s hash links the block to its predecessor, forming an immutable chain. **Example:** In Bitcoin, a block typically holds up to 1 MB of transaction data, which translates to roughly 2,000 transactions per block. **Practical application:** Blocks enable batch processing of transactions, reducing network overhead and providing a chronological record for audit trails. **Challenges:** Block size limits can cause congestion, leading to higher fees and slower confirmations; scaling solutions must balance throughput with security.

Blockchain – a decentralized, append-only ledger composed of linked blocks that record transactions or other data in a tamper-evident manner. Related terms: Distributed ledger, consensus, immutability
The structure ensures that once a block is confirmed, altering its contents would require re-computing the hash of that block and all subsequent blocks, which is computationally prohibitive in proof-of-work systems. **Example:** Bitcoin’s public blockchain maintains a global record of all Bitcoin transfers since 2009. **Practical application:** Beyond cryptocurrencies, blockchains support supply-chain traceability, identity verification, and decentralized finance (DeFi) platforms. **Challenges:** Energy consumption (especially in proof-of-work), governance of protocol upgrades, and interoperability between disparate blockchain networks.

Consensus Mechanism – the algorithmic process by which a distributed network of nodes agrees on the validity and order of new blocks. Related terms: Proof-of-work, proof-of-stake, Byzantine fault tolerance

Common mechanisms include proof-of-work (PoW), where miners solve computational puzzles; proof-of-stake (PoS), where validators lock up tokens as collateral; and practical Byzantine fault tolerance (PBFT), which relies on message exchange among a limited set of nodes. **Practical application:** Consensus ensures that no single actor can unilaterally rewrite history, enabling trustless peer-to-peer transactions. **Challenges:** Trade-offs between security, decentralization, and scalability (the “trilemma”), susceptibility to 51 % attacks in PoW, and “nothing-at-stake” issues in PoS.

Cryptographic Hash – a deterministic function that maps input data of arbitrary size to a fixed-size output, exhibiting pre-image resistance, collision resistance, and avalanche effect. Related terms: SHA-256, keccak-256, merkle tree
Hashes are used to uniquely identify data, secure block headers, and build merkle trees that summarize large transaction sets. **Example:** Bitcoin uses SHA-256 twice (double-hash) to produce a 256-bit block identifier. **Practical application:** Hashes enable efficient verification of data integrity without exposing the original content, essential for light clients and off-chain storage. **Challenges:** Emerging quantum-computing capabilities could weaken current hash algorithms; careful selection of hash functions is required for future-proof designs.

Distributed Ledger – a database replicated across multiple nodes, where each participant holds a synchronized copy of the ledger. Related terms: Blockchain, permissioned ledger, consensus
Unlike centralized databases, a distributed ledger eliminates a single point of failure and enables transparent auditability. **Example:** Hyperledger Fabric implements a permissioned distributed ledger where only authorized participants can submit transactions. **Practical application:** Enterprises use distributed ledgers for inter-bank settlement, trade finance, and provenance tracking of goods. **Challenges:** Ensuring data privacy while maintaining transparency, achieving high throughput under regulatory constraints, and managing node onboarding/offboarding.

Ethereum – a programmable, permissionless blockchain platform that supports the execution of smart contracts written in languages such as Solidity. Related terms: EVM, gas, ERC-20
Ethereum introduced the concept of a Turing-complete virtual machine (EVM), allowing developers to deploy decentralized applications (dApps) that run autonomously. **Example:** The decentralized exchange Uniswap operates entirely on Ethereum smart contracts, enabling peer-to-peer token swaps without intermediaries. **Practical application:** Tokenization of assets, decentralized autonomous organizations (DAOs), and non-fungible tokens (NFTs). **Challenges:** Network congestion leading to high gas fees, security vulnerabilities in contract code, and the ongoing transition from PoW to PoS (Ethereum 2.0) Which requires careful coordination.

Fork – a divergence in the blockchain protocol that creates two separate chains from a common ancestor

block. Related terms: Hard fork, soft fork, chain split
A fork may be intentional (protocol upgrade) or accidental (temporary network partition). When consensus is restored, one chain becomes the dominant canonical chain. **Example:** The Bitcoin Cash hard fork split from Bitcoin in 2017 to increase block size limits. **Practical application:** Forks enable the introduction of new features, such as privacy enhancements or governance changes, without requiring a centralized authority. **Challenges:** Community disagreement can lead to prolonged splits, confusing users and fragmenting liquidity; coordination of upgrades across diverse stakeholders is complex.

Gas – the unit of computational work required to execute operations on the Ethereum network, priced in ether (ETH). Related terms: Gas price, gas limit, transaction fee
Every operation in a smart contract consumes a predefined amount of gas; the total gas used multiplied by the gas price determines the transaction fee paid to validators. **Practical application:** Gas pricing incentivizes efficient code; developers must optimize contracts to reduce costs for end-users.
Challenges: Volatile gas prices can make dApp usage prohibitively expensive during network spikes; layer-2 solutions aim to abstract gas away from users but add complexity.

Hard Fork – a non-backward-compatible protocol change that forces all nodes to upgrade to the new software version, otherwise they will follow a divergent chain. Related terms: Fork, upgrade, consensus rule
Hard forks are used to implement major changes such as new transaction types, altered block size limits, or new consensus algorithms. **Example:** Ethereum's "London" upgrade introduced the EIP-1559 fee mechanism, altering how gas fees are calculated. **Practical application:** Enables rapid innovation and the adoption of novel features that cannot be accommodated by soft-fork constraints. **Challenges:** Requires coordinated migration; failure to upgrade can result in a split chain, as seen with Bitcoin Cash and Bitcoin SV.

Hash Function – a cryptographic primitive that converts input data into a fixed-length digest, ensuring data integrity and enabling efficient lookup structures. Related terms: Cryptographic hash, SHA-3, merkle root
Hash functions are deterministic, fast to compute, and resistant to pre-image and collision attacks.
Practical application: Building merkle trees to verify inclusion of a transaction without exposing the entire block; generating unique identifiers for digital assets. **Challenges:** Advances in cryptanalysis may render older hash functions insecure; selecting quantum-resistant alternatives is an emerging research area.

Immutability – the property that once data is recorded on a blockchain, it cannot be altered without detection. Related terms: Tamper-evidence, consensus, cryptographic hash
Immutability arises from the chaining of block hashes and the distributed nature of the ledger; any change would require re-computing hashes for the altered block and all subsequent blocks across a majority of nodes. **Practical application:** Provides reliable audit trails for financial reporting, regulatory compliance,

and provenance tracking of goods. **Challenges:** Errors or malicious entries become permanent; mechanisms such as “self-destruct” functions or off-chain correction procedures must be designed carefully to avoid undermining trust.

Initial Coin Offering (ICO) – a fundraising mechanism where new tokens are sold to early investors, often in exchange for established cryptocurrencies like Bitcoin or Ether. Related terms: Token sale, utility token, security token
ICOs gained popularity in 2017-2018 as a way for projects to raise capital without traditional venture funding. **Example:** The EOS token sale raised over $4 billion through an ICO model. **Practical application:** Allows startups to bootstrap development, create a community of token holders, and align incentives. **Challenges:** Regulatory scrutiny (many jurisdictions classify tokens as securities), high risk of fraud, and lack of investor protection.

Merkle Tree – a binary hash tree that aggregates transaction hashes into a single root hash, enabling efficient verification of data inclusion. Related terms: Merkle root, proof of inclusion, hash function
Each leaf node represents a transaction hash; parent nodes hash the concatenation of their children, culminating in the merkle root stored in the block header. **Practical application:** Light clients can verify a transaction’s inclusion by downloading only a small set of sibling hashes, reducing bandwidth requirements. **Challenges:** Designing scalable merkle structures for high-throughput systems; handling dynamic data sets where transactions are frequently added or removed.

Node – a computer that participates in a blockchain network by maintaining a copy of the ledger, validating transactions, and possibly proposing new blocks. Related terms: Validator, miner, full node, light node
Nodes can be full (storing the entire blockchain) or light (storing only block headers). Validators in PoS networks also stake tokens to earn block-proposal rights. **Practical application:** Running a node contributes to network security, enables decentralized services, and can generate revenue through block rewards or transaction fees. **Challenges:** Resource consumption (storage, CPU, bandwidth), especially for high-throughput chains; ensuring software updates are applied uniformly to avoid forks.

Private Key – a secret cryptographic value that proves ownership of a blockchain address and enables signing of transactions. Related terms: Public key, wallet, digital signature
The private key must be kept confidential; loss or theft results in irreversible loss of assets. It is typically stored in a keystore file, hardware wallet, or seed phrase. **Practical application:** Users sign transfer orders, authorize smart-contract interactions, and manage multi-sig wallets. **Challenges:** Secure key management, protection against phishing, and recovery mechanisms for lost keys without compromising security.

Public Key – the cryptographic counterpart to a private key, used to derive a blockchain address and verify digital signatures. Related terms: Private key, address, asymmetric cryptography
Public keys are openly shared; they enable anyone to verify that a transaction was signed by the holder of the corresponding private key. **Practical application:** Enables transparent verification of ownership without exposing secret material; forms the basis of decentralized identity schemes. **Challenges:** Public-key infrastructure (PKI) integration, revocation handling, and ensuring that derived addresses are collision-free.

Smart Contract – a self-executing program stored on a blockchain that automatically enforces the terms of an agreement when predefined conditions are met. Related terms: Solidity, EVM, dApp, token
Smart contracts are immutable once deployed (unless upgrade patterns are used) and operate without intermediaries, providing deterministic outcomes. **Example:** A crowdfunding contract that releases funds to a project only if a funding goal is reached before a deadline. **Practical application:** Decentralized finance (lending, swapping), supply-chain automation, voting systems, and NFT minting. **Challenges:** Coding errors can lead to exploits (e.G., The DAO hack), gas inefficiency, and difficulty in updating contracts after deployment.

Token – a digital asset issued on top of an existing blockchain, representing utility, ownership, or access rights. Related terms: ERC-20, ERC-721, utility token, security token
Tokens inherit the security and consensus of the underlying chain, allowing rapid issuance without creating a new blockchain. **Example:** The Basic Attention Token (BAT) operates on Ethereum as an ERC-20 utility token for the Brave browser ecosystem. **Practical application:** Incentivizing user behavior, representing fractional ownership of real-world assets, and enabling governance voting. **Challenges:** Regulatory classification, token standards interoperability, and liquidity fragmentation across multiple platforms.

Transaction – a data structure that records the transfer of assets or the invocation of a smart contract, signed by the sender's private key. Related terms: UTXO, nonce, gas, block
Transactions contain inputs, outputs, amount, fee, and a digital signature. In account-based models (e.G., Ethereum), the transaction updates the sender's and receiver's balances. **Practical application:** Core operation of any blockchain—moving cryptocurrency, executing contract functions, or updating state. **Challenges:** Mempool congestion, fee estimation, replay attacks across forks, and ensuring privacy of transaction metadata.

Validator – a participant in a proof-of-stake network responsible for proposing and attesting to new blocks, earning rewards proportional to the amount staked. Related terms: Staking, slashing, consensus

Validators lock up tokens as collateral; misbehavior can lead to slashing (partial loss of stake). They replace miners in PoS systems, reducing energy consumption. **Practical application:** Securing networks like Cardano, Polkadot, and Ethereum 2.0 While earning staking yields. **Challenges:** Centralization risk if a few entities control large stake shares, the “nothing-at-stake” problem, and the need for reliable uptime to avoid penalties.

Zero-Knowledge Proof – a cryptographic protocol that allows one party to prove knowledge of a statement without revealing the underlying information. Related terms: Zk-SNARK, zk-STARK, privacy, succinct proof Zero-knowledge proofs enable privacy-preserving transactions and off-chain computation verification. Zk-SNARKs provide succinct, non-interactive proofs, while zk-STARKs avoid trusted setup and offer post-quantum security. **Example:** Zcash uses zk-SNARKs to shield transaction amounts and participants while still allowing the network to verify correctness. **Practical application:** Confidential payments, identity verification without data leakage, and scaling solutions that batch many transactions into a single proof. **Challenges:** High computational overhead for proof generation, trusted setup requirements for some schemes, and the need for specialized hardware to achieve practical performance.
