Intro

1) How a blockchain works (plumbing, not poetry)

Think of a blockchain as a **global append-only database** that many strangers keep in sync without a central admin. Here's the life of a transaction and the mechanisms that keep it honest.

A. What a transaction is

- A transaction is a small message: "Alice authorizes 1 coin to Bob."
- Alice proves it's really her by attaching a **digital signature** (made with her private key). Anyone can verify it with her **public key**.
- On different chains the accounting model differs:
  - **UTXO model** (Bitcoin): you spend specific "coins/outputs" you received earlier.
  - **Account model** (Ethereum): each account has a balance; debits/credits change it.

B. How transactions enter the system

- Alice's wallet broadcasts the signed transaction to the peer-to-peer network.
- Nodes do **cheap checks**:
  - Signature valid?
  - Inputs unspent (UTXO) / balance sufficient (account model)?
  - Nonce correct (prevents replay), fees present?
- Valid ones sit in a public waiting room called the **mempool**.

C. Blocks: batching and ordering

- A **block** = header (metadata + hash of previous block + Merkle root of txs) + list of transactions.
- The block links to the previous block via a **hash** → that's the "chain."

D. Who gets to add the next block? (Consensus)

Different blockchains make writing to the ledger **costly** (to stop spam/cheating) but keep reading/verification **cheap**:

- **Proof-of-Work (PoW, e.g., Bitcoin):**
  Many miners race to find a lucky **nonce** so the block header's hash is below a target. Finding it costs **electricity** (millions/billions of hash trials). Verifying the winner is **trivial** (one hash and compare). This "asymmetric cost" is why it's open yet robust.
- **Proof-of-Stake (PoS, e.g., Ethereum today):**
  Validators lock up (**stake**) the native coin. One proposes a block; others **attest**. If they misbehave, their stake can be **slashed**. Cost here is **economic collateral**, not electricity. Verification is still cheap (check signatures/attestations).
- **BFT-style consensus (e.g., Ripple/XRP-style or some permissioned chains):**
  A known set of validators vote and need a **supermajority** to finalize a block. Fast finality, but validator membership is curated.

E. Forks, reorgs, and "finality"

- Sometimes **two blocks** at the same height appear (e.g., two miners win near-simultaneously). The chain "forks" briefly.
- Nodes follow the rule "**longest/most-work (or most-weight) chain wins**." As soon as one branch gets another block, the network converges on it.
- If your transaction was in the loser branch, it gets **reorganized** out (a **reorg**). It should be included again later, but that brief window is why we wait for **confirmations**.
- **Probabilistic finality (Bitcoin):** Every block after yours drops the chance of reorg exponentially. The common heuristic "**6 confirmations**" ≈ ~1 hour makes it practically irreversible.
- **Deterministic finality (many PoS/BFT systems):** Once enough validators sign a checkpoint (supermajority), it's **final** in seconds; no reorgs beyond that point unless the validator set itself is slashed/rolled back.

F. "Immutability," precisely

- Each block's header contains the **hash of the previous block**. Change *anything* in an old block → its hash changes → every later block's hash changes → the chain breaks.
- To tamper with history, an attacker must **rewrite that block and all after it** and outpace honest participants (i.e., >50% hashpower in PoW or control a supermajority in PoS/BFT). That **economic infeasibility** is what we mean by "immutable."

Applications

1) fast payments across countries (remittances)

the problem with today (banks):