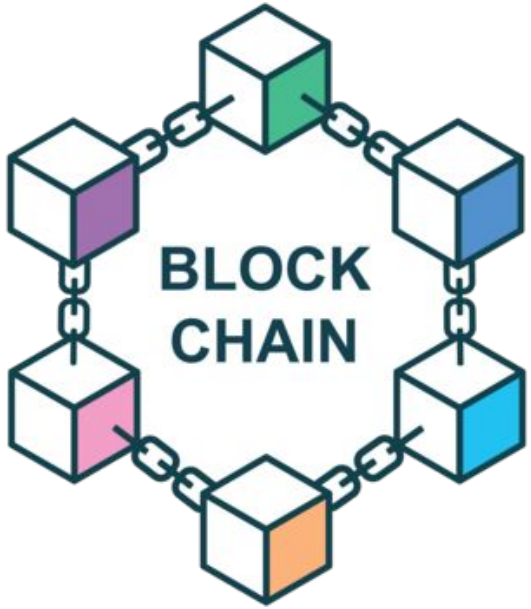


Blockchain Security and Performance Introduction

DAY 1

Name: Divesh Jadhwani

Goal : Overview of blockchain and why we need to secure it and overview of syllabus

Pre-Lecture Example: Introducing Blockchain ,

What Happens Without Blockchain? (The Problem)

Imagine you and your friends **eat lunch** Instead and someone pays , and only **one person** keeps track of who paid for lunch.

This means **you must trust** that person to be honest.

But what if they **lie** and change the records to benefit themselves?

Or what if the **notebook gets lost or destroyed**?

You have no way to prove what really happened!

Problem: Centralized control means **trust issues, fraud risk, and lack of transparency**.

Pre-Lecture Example: Introducing Blockchain ,

What Happens With Blockchain? (Problem Solved)

Instead of **one person** keeping the record, **everyone** has a copy of the notebook.

Whenever someone pays, **everyone updates their copy** at the same time.

If one person tries to **change the records unfairly**, the others will **reject the false change**.

The notebook **cannot be erased or altered** because copies exist everywhere.

Result: Blockchain makes the system **transparent, tamper-proof, and trustworthy**!

Pre-Lecture Example: Introducing Blockchain , (Another Problem)

Even though everyone has a copy of the notebook, what if **someone finds a way to trick the system**?

Imagine a friend **somehow convinces half of the group** to accept a fake record.

Or what if someone **finds a way to copy another person's payment entry** and use it again?

How do we make sure **no one can cheat the system**?

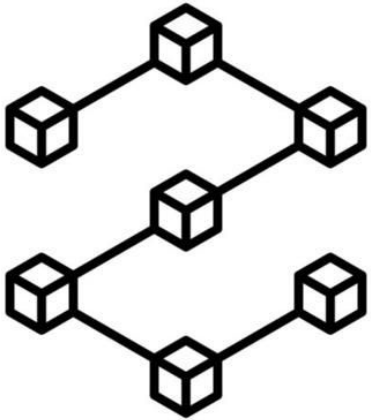
🔒 **Solution**: Blockchain security ensures that records **cannot be manipulated**, transactions are **verified**, and attackers **cannot trick the system**.

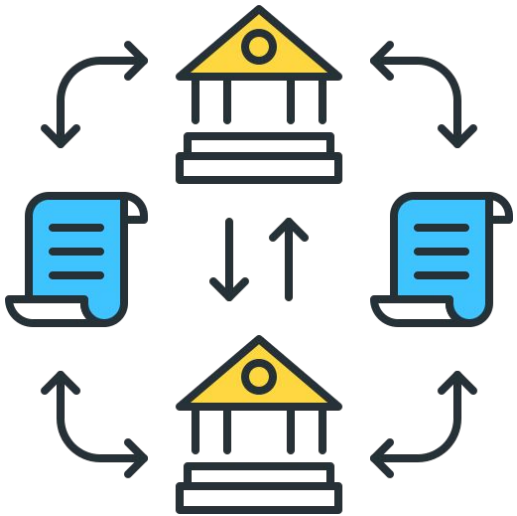
Agenda

1. What is blockchain? (Simple Explanation)
2. Why do we need security in blockchain?
3. What affects blockchain performance?
4. Overview of our course syllabus
5. Key programming languages & technologies for blockchain

Understanding Blockchain , What is Blockchain?

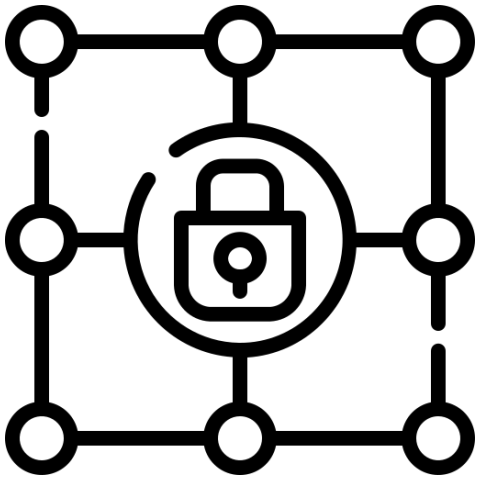
- A **blockchain** is like a digital notebook that everyone can see but **no one can change** unfairly.
- It stores information in **blocks**, and each block is connected (chained) to the previous one.
- Once a block is added, it **cannot be changed**, making it **secure and trustworthy**.
- Example: A Google Sheet where multiple people can add data, but once recorded, it cannot be edited!





Why Do We Need Security in Blockchain? , Why Security Matters

- Just like a bank needs **security** to protect money, blockchain needs security to **protect data & transactions**.
- Without security, hackers can:
  a. Steal cryptocurrency
  b. Manipulate smart contracts
  c. Change transaction history
- **Example**: If someone **hacks** a bank and changes their account balance to $1 million, would that be fair?

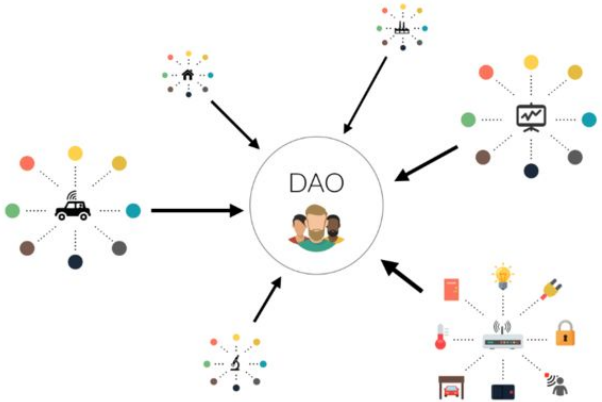


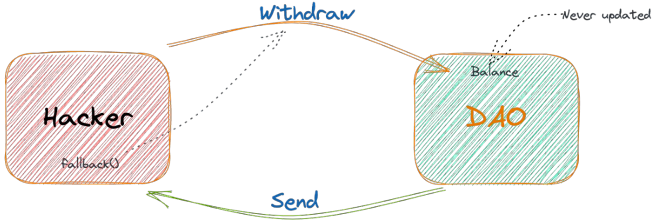
Real-Life Blockchain Security Issues , Blockchain Hacks

DAO Hack (2016) – $50 million stolen due to a smart contract bug.

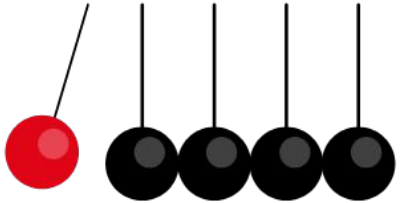
Ronin Bridge Hack (2022) – $600 million lost due to poor security.

Reentrancy Attack – A bug that allows hackers to repeatedly withdraw funds.
Key Takeaway: Even blockchain can have **weak points** if not secured properly!





What Affects Blockchain Performance? , Why is Blockchain Sometimes Slow or Expensive?



- **Transaction Speed** – More users mean slower transactions.
- **Transaction Fees** – Higher demand increases fees.
- **Network Size & Complexity** – More data means slower processing.
- **Interoperability Issues** – Different blockchains can’t always talk to each other.
- **Lack of Regulation** – Governments don’t always support blockchain systems.

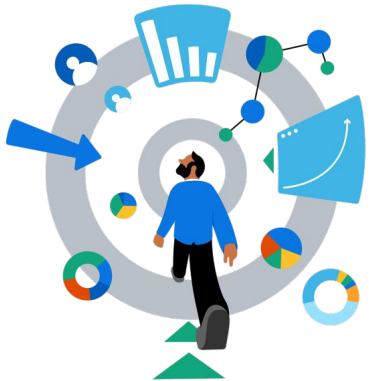
Example: Imagine a road with too many cars – traffic increases, making travel slower!

Overview of Our Course (What We Will Learn?)

Breakdown of the Syllabus

- **Unit 1**: Security Issues – Understanding common attacks and solutions.
- **Unit 2**: Security Tools – Learning how to analyze and secure smart contracts.
- **Unit 3**: Performance Issues – Understanding blockchain speed and efficiency problems.
- **Unit 4**: Performance Improvements – Exploring ways to make blockchain faster and better.

Outcome: By the end of the course, you will understand how to build and analyze secure and efficient blockchain applications!



Best Programming Languages & Technologies for Blockchain

- **Solidity** – Best for smart contracts (Ethereum).
- **Rust** – Used for secure and high-performance blockchains (Polkadot, Solana).
- **Go** – Used for blockchain infrastructure (Hyperledger Fabric).
- **Python & JavaScript** – Great for blockchain apps and testing.
- **SQL & NoSQL** – Used for storing blockchain data efficiently.
- **Important Tools**: MetaMask, Hardhat, Truffle, Ganache, Remix IDE.
  Example: Solidity is like C++ but designed for writing blockchain smart contracts.

Final Recap

✅ Blockchain is a **secure and decentralized** technology.
✅ Security is important because **hackers can exploit weak points**.
✅ Blockchain performance depends on **speed, fees, and network size**.
✅ We will study **security issues, tools, performance, and improvements** in this course.
✅ **Programming languages** like Solidity, Rust, and Go are essential for blockchain development.

Thank You