Câu 1. Định nghĩa công nghệ Blockchain và đặc điểm khác biệt

Blockchain là một cơ sở dữ liệu phân tán được thiết kế để ghi lại thông tin theo cách không thể thay đổi và không thể bị giả mạo. Mỗi khối trong chuỗi chứa dữ liệu, một mã băm (hash) của chính nó và mã băm của khối trước đó. Nhờ đặc tính liên kết này, Blockchain cung cấp một chuỗi lịch sử bất biến.

Đặc điểm chính:

- Phân tán (Distributed): Dữ liệu không lưu trữ tại một điểm trung tâm mà phân tán trên nhiều nút (nodes).
- Bất biến (Immutability): Một khi dữ liệu được ghi vào Blockchain, việc thay đổi là gần như không thể.
- Minh bạch (Transparency): Tất cả các giao dịch đều được công khai và có thể xác minh bởi tất cả các nút.
- Bảo mật cao: Nhờ sử dụng mã hóa mật mã học và các cơ chế đồng thuận (consensus).

So với cơ sở dữ liệu truyền thống (centralized database), Blockchain vượt trội về khả năng ngăn chặn giả mạo dữ liệu, đặc biệt trong các hệ thống yêu cầu độ tin cậy cao như tài chính, y tế và chuỗi cung ứng.

Câu 2. So sánh mô hình Client/Server và Peer-to-Peer (P2P) trong Blockchain

Tiêu chí Client/Server Peer-to-Peer (P2P) Cấu trúc Trung tâm (server) phục vụ nhiều client Mỗi nút vừa là client vừa là server Độ tin cậy Dễ bị tấn công nếu server bị lỗi Tăng độ bền vững nhờ phân tán Khả năng mở rộng Giới hạn do server trung tâm Mở rộng linh hoạt theo số lượng nút Ứng dụng Web, ứng dụng doanh nghiệp Blockchain, BitTorrent Blockchain hoạt động trên nền tảng P2P để đảm bảo phân quyền, chống kiểm duyệt, và khả năng phục hồi cao trước các sự cố hoặc tấn công mạng.

Câu 3. Cấu trúc của một khối trong Blockchain Một khối (block) trong Blockchain Bao gồm hai phần chính:

Block Header:

- Previous Hash: Mã băm của khối trước.
- Timestamp: Dấu thời gian tạo khối.
- Nonce: Số ngẫu nhiên dùng trong cơ chế đồng thuận (như PoW).
- Merkle Root: Mã băm tổng hợp từ toàn bộ giao dịch trong khối.

Phần dữ liệu (Body):

- Danh sách các giao dịch (transactions).

Vai trò:

- Header giúp liên kết các khối và hỗ trợ cơ chế đồng thuận.
- Dữ liệu giao dịch chứa thông tin thực tế được ghi nhận

Câu 4. Các giao thức đồng thuận: PoW, PoS và PBFT

Giao thức	Cách hoạt động	Ưu điểm	Nhược điểm
PoW (Proof of Work)	Giải bài toán băm để thêm khối mới	Bảo mật cao, phân quyền	Tốn điện năng, chậm
PoS (Proof of Stake)	Xác thực dựa trên số coin nắm giữ	Tiết kiệm năng lượng	Có nguy cơ tập trung hóa
PBFT (Practical Byzantine Fault Tolerance)	Đa số phiếu bầu từ các nút tin cậy	Hiệu suất cao, phù hợp hệ thống riêng tư	Khó mở rộng, yêu cầu danh tính các nút

Mỗi giao thức phù hợp với bối cảnh triển khai riêng: PoW cho công khai (Bitcoin), PoS cho các hệ sinh thái mới (Ethereum 2.0), PBFT cho Blockchain doanh nghiệp (Hyperledger).

Câu 5. Tính toàn vẹn dữ liệu và vai trò của SHA-256

Blockchain đảm bảo tính toàn vẹn nhờ:

- Mỗi khối chứa mã băm (hash) liên kết với khối trước.
- Mỗi giao dịch được mã hóa.

SHA-256 là một thuật toán băm thuộc họ SHA-2:

- Nhận đầu vào bất kỳ, cho ra chuỗi băm 256 bit.
- Không thể đảo ngược.
- Thay đổi nhỏ đầu vào → thay đổi hoàn toàn đầu ra.

SHA-256 giúp:

- Xác minh dữ liệu (qua Merkle Tree).
- Bảo vệ giao dịch khỏi giả mạo.
- Tạo chữ ký số cho các hợp đồng thông minh.

Câu 6. Smart Contract là gì? Ứng dụng thực tế

Smart Contract là các đoạn mã chạy trên Blockchain, thực thi tự động khi điều kiện thỏa mãn, không cần bên thứ ba.

Ứng dụng thực tế:

- Tài chính phi tập trung (DeFi): Giao dịch, cho vay (Uniswap, Aave).
- Logistics: Theo dõi hàng hóa.
- Y tế: Quản lý hồ sơ bệnh án.
- Chính phủ điện tử: Bỏ phiếu điện tử minh bạch.

Ví dụ: Một Smart Contract trong bảo hiểm sẽ tự động thanh toán khi điều kiện thời tiết (do IoT cung cấp) được xác nhận.

Câu 7. Các bước triển khai Smart Contract với Web3.py

1. Cài đặt Web3.py và thiết lập môi trường (Python, Ganache, MetaMask).
2. Viết Smart Contract bằng Solidity.
3. Biên dịch Contract (dùng solc hoặc Truffle).
4. Triển khai Contract lên mạng (ganache/testnet/mainnet).
5. Tương tác qua Web3.py:

- Kết nối tới provider (HTTP/WebSocket).
- Tạo và gửi giao dịch.
- Đọc dữ liệu từ Blockchain.

Câu 8. Tích hợp Blockchain trong xác minh nguồn gốc dữ liệu khoa học dữ liệu

Quy trình đề xuất:

1. Ghi nhận metadata (thời gian, nguồn, checksum) của dữ liệu lên Blockchain.
2. Gắn mã băm SHA-256 cho mỗi tập dữ liệu.
3. Xác minh dữ liệu đầu vào bằng cách so sánh mã băm và chuỗi ghi nhận.
4. Sử dụng Smart Contract để quy định quyền truy cập, ghi nhận sự kiện thay đổi.

Áp dụng Blockchain trong pipeline dữ liệu giúp tăng minh bạch, truy xuất nguồn gốc và chống giả mạo dữ liệu đầu vào, rất phù hợp với khoa học dữ liệu trong lĩnh vực nhạy cảm (tài chính, y tế).

Câu 9. Lỗi "invalid opcode" khi triển khai Smart Contract

Nguyên nhân khả dĩ:

- Sai cú pháp Solidity hoặc logic không hợp lệ.
- Tương thích ABI và contract không khớp.
- Gọi hàm không tồn tại hoặc truyền sai tham số.
- Thiếu gas khi gọi contract.

Các bước kiểm tra và khắc phục:

1. Kiểm tra bản Solidity có khớp với compiler không.
2. Dùng truffle debug hoặc remix IDE để mô phỏng contract.
3. Kiểm tra hợp đồng có bị lỗi logic (chia cho 0, stack overflow).
4. Đảm bảo đúng gasLimit khi gọi hàm.
5. Kiểm tra ABI khi sử dụng với Web3.py/JavaScript.