Designing a Secure Banking Transaction System Using Blockchain

1. System Design

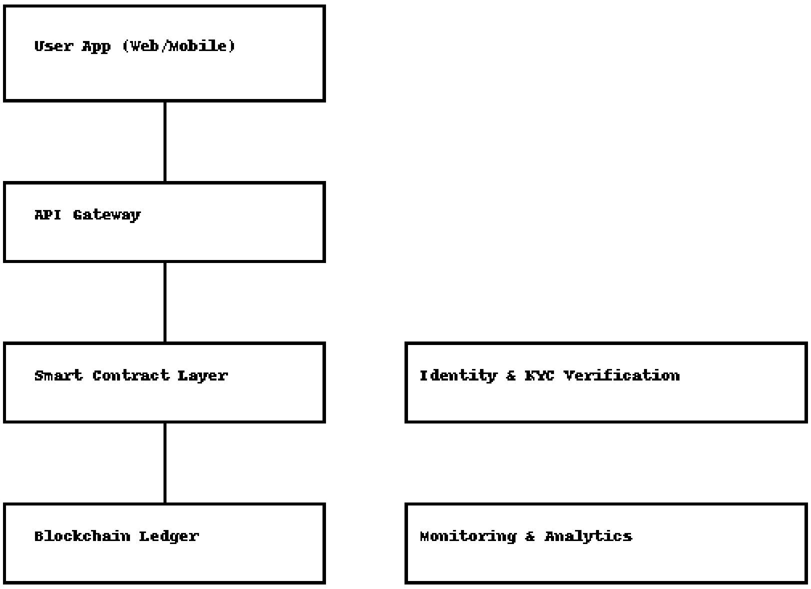
System Design Overview:

The secure banking transaction system using blockchain is based on a permissioned blockchain network. It includes several modules: identity management, transaction processing via smart contracts, consensus mechanism, and real-time monitoring.

Key Components:

1. Identity Management: Customers are onboarded with KYC verification and assigned a blockchain wallet with public/private key pairs.
2. Transaction Interface: Web/mobile app allows users to initiate secure transactions.
3. Smart Contracts: Enforce transaction rules (e.g., balance check, limits, AML rules).
4. Blockchain Ledger: Immutable, distributed ledger that logs transactions securely.
5. Monitoring & Analytics: Real-time monitoring of the ledger using AI/ML tools to detect fraud.

System Architecture Diagram



2. Smart Contract Design

Smart Contract Design:

Smart contracts act as rule engines that verify and process transactions before committing them to the blockchain. Example rules include:

- Checking sender's balance

- Enforcing transaction limits

- Running AML/KYC compliance checks

Sample Pseudocode:

```
contract SecureBankTransaction {
    mapping(address => uint) balances;

    function transfer(address receiver, uint amount) public {
        require(balances[msg.sender] >= amount, "Insufficient balance");
        require(amount < 10000, "Transfer limit exceeded");

        balances[msg.sender] -= amount;
        balances[receiver] += amount;
        emit TransferCompleted(msg.sender, receiver, amount);
    }
}
```

Smart contracts must be audited and tested rigorously to prevent security vulnerabilities like reentrancy, overflows, or logic flaws.