THADOMAL SHAHANI TSEC ENGINEERING COLLEGE

Blockchain

Experiment No-1

08/10/24

Aim : WAP to implement merkel tree.

Theory : When we use Hash functions it maps input to fixed output. For every input a unique hash is generate as output. Hence, it is possible to validate huge amount of data through this hash merkel tree is a data structure, also known as Binary tree. This data structure summaries & verifies the integrity of large sets of data. Each non-leaf node is a hash of its child node. All leaf nodes are at the same depth. Inputs are first placed at the leaves. Both the child nodes once hashed together to generate a value for the parent node the procedure is repeated till the single hash value which is merkel root is generated.

In the figure T1, T2, T3, T4 represent a typical transition these transaction are hashed seperately to get their corresponding hash value.

Eg :- T1 is hashed to get its corresponding hash values (T)

After each transaction have been seperately hashed to generate its corresponding hash value, the new resultant values are further combined with the adjacent pathen to be hashed again

THADOMAL SHAHANI TSEC ENGINEERING COLLEGE

#(Output 12, output 34)
output 1, 2, 3, 4

H'(output 1,2)
output 1, 2

H'(output 3, 4)
output 3, 4

H(T1) H(T2) H(T3) H(T4)
output 1 Output 2 output 3 Output 4

T1 T2 T3 T4

Merkel tree

the process is repeated until it the top of the tree is reached i.e Merkel Root (Final Hash value) In the diagram merkel Root is labelled as output 12.34. Merkel root is the part of the block header, which represents a summary of all transactions data to validate a ; in a tree, we can selectively compare) the hash with the selected nodes rather than find a Hash of the whole tree

THADOMAL SHAHANI TSEC ENGINEERING COLLEGE

Experiment No. 2

Aim:- Create smart contract using solidity & Remix IDE

Theory :-

A smart contract is an encoded agreement between two parties that execute and exchange automatically once goods are delivered or service completed. They eliminate the need of intermediaries & automatically carry out terms one requirements are met

Working of smart contract

1) Pre defined contract An option contract is written as code into a blockchain.
2) Events - If the events specified by the contract occurs it triggers the coded terms of the contract automatically.
3) Once executed the terms of the contract will automatically transfer the value to the relevant parties. This is known as execute & value transfer
4) settlement. Regulators can study the immutable transactions record to understand all activities that has taken place

The structure of smart Contract contains

1) State variables stored in contract storage
2) Functions:- Few lines of code that can perform a certain function
3) function modifiers they are like a subset

4] Struct types :- Struct is a custom defined data type that contains several variable declared in it.

5] Events :- These are functions that rise up after completion of function creation if event code is written there.

6] Enum :- used to create custom types & sets a set of values

* Characteristics of smart contract

Immutable :- No party will be able to change the contract once it is fixed & written to the public ledger

Customizable :- Smart contracts have the ability for modification or combination before being launched to do what user wants it to do

Smart contract :- built using solidity provide a way to automate agreements between parties Remix IDE support the development problem with tools for coding debugging & deploying contracts making it essential resource for the developers together. Ability of Remix IDE streamline the creation management of smart contracts using blockchain apps

08/10/24

Experiment No. 3

Aim :- Write a program in solidity preparing to create transaction using the remix IDE

Theory :-

In blockchain a transaction is a transfer of data or assets between participants it involves recording information on the blockchain ledger, ensuring it is secure, transparent & immutable once a transaction is validated by the network (through mechanisms like proof of work). It is grouped into a block added to a blockchain, making it a permanent part of the ledger. Transactions are designed to be tamper proof, to maintain trust. Each transaction typically includes details such as the sender, recipient, amount & a timestamp. After a transaction is initiated, it is broadcasted to the network where nodes verify its authenticity using cryptographic algorithms. Once verified, the transaction is included in a block and the block is appended to the existing chain. This process ensures that all participants in the network have a consistent & accurate record of transactions.

making fraud & double spending difficult
the list of transactions is a component of
header. It forms a tree like structure called
as merkel tree. The merkle
the merkel root is used to construct the
block hash. If you change a transaction
you need to change all the subsequent
block hash

THADOMAL SHAHANI TSEC ENGINEERING COLLEGE

Experiment No. 4

B

D8102

Aim :- Write a program in solidity to implement transaction by ending of wallet

Theory :-

Blockchain is a decentralized digital ledger that records transactions across many computers so that the record cannot be altered retroactively. It operates in a peer to peer network where each participant maints a copy of the ledger this ledger ensures & transparent ensures that once recorded in the decentralized ecosystem, smart contracts use self executing agreements coded directly into the blockchain automating & enforcing terms without intermediaries

Smart contract are self-executing contracts with terms written directly into code they run on blockchain platforms like Ethereum they contracts automatically enforce & execute terms of an agreement based on predefined rules & conditions Ethereum is a blockchain platform that enables developers to deploy & run smart contracts & decentralized applications it has its own cryptocurrency ether (ETH) which is used to pay for transactions fees & computational services on the network Events are used to log activities occ

deposits, withdrawals & transfers enabling real time updates for off chain applications mapping these user balances, allowing efficient data retrieval functions define the contracts logic including methods for depositing funds with drawing funds & transferring either between addresses with built in checks to ensure correction & security. To interact with blockchain a wallet refuses to an ethereum wallet address that can hold & manage Ether or other tokens

transaction in ethereum involve sending ether or tokens from one address to another. Each transaction is signed by the senders private key & is validated by the network

Assignment no. 1

THADOMAL SHAHANI TSEC ENGINEERING COLLEGE

Experiment No. 5

Aim :- Write implementation of the blockchain platform Ethereum

Theory :

Ethereum is a decentralized platform that enables the creation & execution of smart contracts & decentralized apps (dApps)

Geth is one of the three original implementations of the ethereum protocol written in go language

Steps to build your private blockchain.

Step 1 :- Install Geth
Geth is a UI with some resources to connect you to Ethereum network.
You should have access to Geth
Binary before moving ahead

Step 2 : Create genesis.json
Every blockchain starts with a genesis (first block) the genesis block is configured using genesis.json file for a private network

Step 3 : Initialize the private blockchain we have the configuration ready for the genesis block. we need to run first geth command to initialize the private blockchain

geth --datadir ./nodes init genesis.json

Step 4: Add the first node to the private blockchain

port : Our node is running on this port

http.port : this is for HTTP protocol access

Step 5: Add more nodes

Step 6 : Connect node 2 with node 1 as peer

Step 7 : Mining blocks & creating transactions

Mining is the process of validating transactions & adding them to blockchain

Group no. 1

THADOMAL SHAHANI TSEC ENGINEERING COLLEGE

08/02/24 (A)

Experiment No. 6

Aim: Show implementation of blockchain platform Ganache

Theory:

Ganache is a personal blockchain for Ethereum development that you can use to deploy smart contracts, develop applications & run tests

Steps for implimentation

Step 1: Download Ganache from the truffle suite. Ganache is a part of the truffle suit, a popular platform

Step 2: Install: Launch Ganache after downloading, select "Quickstart Ethereum" from Ganache GUI.

Step 3: Explore accounts & addresses Ganache automatically generates 10 accounts each loaded with 100 Eth for testing purposes

Step 4: Write simple smart contract using remix IDE, write simple smart contract for adding two numbers

```
pragma solidity 0.8.0;
contract additem {

function add (int a, int b)
    {
      public pure returns (int){
        return a+b;
    }
```

Step 5: Connect Remix IDE with Ganache until know. Local blockchain running on Ganache for deploy your smart contract

Step 6: Deploy the contract.
using Remix IDE, deploy the smart contract to the local blockchain

Step 7: Observe Ethereum transactions monitor the effects of contract deployment on account balances

Group no. 1
Roll no. 17

A

08/10/24

Case Study: Hyperledger - An Open Source Blockchain Framework for Enterprise Applications

Introduction

Hyperledger is an open-source collaborative effort hosted by the Linux Foundation, established in 2015 to advance cross-industry blockchain technologies. Unlike public blockchain systems like Bitcoin or Ethereum, Hyperledger focuses on permissioned blockchains that are designed specifically for enterprise use cases. It allows businesses to build and deploy customized blockchain applications, tools, and systems with a strong emphasis on privacy, scalability, and security.

Key Features of Hyperledger

1. **Permissioned Network**: Hyperledger operates on a permissioned blockchain, meaning participants are known and authorized, ensuring greater control and trust in the system.
2. **Modular Architecture**: Hyperledger provides a modular framework that allows developers to integrate different consensus mechanisms, identity management systems, and smart contract logic based on their specific use case requirements.
3. **Smart Contracts (Chaincode)**: Hyperledger supports smart contracts, known as Chaincode, which automates transactions and processes within the blockchain network. This ensures reliability and transparency in contract execution.
4. **Consensus Algorithms**: Hyperledger supports various consensus mechanisms, such as Practical Byzantine Fault Tolerance (PBFT) and Raft, which offer faster transaction finality and scalability compared to traditional proof-of-work-based blockchains.

Components of Hyperledger

1. **Hyperledger Fabric**: A flexible, scalable, and modular DLT platform that supports plug-and-play components like consensus and membership services. It is suitable for applications requiring high levels of confidentiality.
2. **Hyperledger Sawtooth**: A blockchain platform designed for building, deploying, and running distributed ledgers. It is modular, allowing enterprises to choose consensus algorithms (such as Proof of Elapsed Time) without altering the core blockchain architecture.
3. **Hyperledger Indy**: A distributed ledger built for decentralized identity, enabling organizations to create and manage digital identities independent of any centralized authority.
4. **Hyperledger Iroha**: Aimed at mobile and IoT applications, Iroha focuses on simplicity and providing a simple architecture that is easy to integrate into existing applications.

Case Example: Walmart and Hyperledger Fabric

Walmart has successfully integrated Hyperledger Fabric to improve its supply chain management. Walmart uses blockchain technology to track the movement of food products from

farms to store shelves. Before blockchain, the tracking process took days; with Hyperledger, it now takes seconds. The solution ensures transparency, enhances food safety, and increases trust among suppliers, retailers, and customers.

Benefits of Hyperledger

1. **Privacy**: Since Hyperledger operates on permissioned networks, sensitive business data can be kept confidential while allowing authorized parties to access relevant information.
2. **Efficiency**: Hyperledger offers fast processing times for transactions and eliminates the need for intermediaries, reducing operational costs.
3. **Scalability**: Due to its modular architecture, Hyperledger is highly scalable, allowing enterprises to expand their blockchain networks as their business grows.

Conclusion

Hyperledger provides a robust, secure, and scalable solution for enterprises seeking to leverage blockchain technology. Its modular design, permissioned network, and focus on enterprise needs make it an ideal platform for businesses to deploy blockchain solutions that drive innovation and efficiency.

Case Study: Corda – A Blockchain Platform for Financial Institutions

Introduction

Corda is an open-source blockchain platform developed by R3, specifically designed for businesses, with a particular focus on financial institutions. Unlike typical blockchain systems that use a global ledger accessible to all participants, Corda operates with a unique approach that shares transaction data only between relevant parties. This ensures privacy, scalability, and regulatory compliance, making it highly suitable for financial services, supply chain management, and other enterprise applications requiring strict confidentiality.

Key Features of Corda

1. **Permissioned Network**: Corda is a permissioned blockchain where participants are pre-approved, ensuring trust and reducing the need for costly consensus mechanisms like Proof of Work (PoW).
2. **Privacy and Confidentiality**: One of Corda's standout features is its data privacy model. It ensures that transaction data is shared only with those who need to see it, unlike public blockchains where data is visible to all participants.
3. **Smart Contracts**: Corda uses smart contracts to automate the execution of legal agreements. These contracts are written in Java or Kotlin, making it easier for developers to integrate with existing enterprise systems.
4. **Notary Service**: To prevent double-spending, Corda uses notaries to validate transactions. These notaries can either be a single entity or a group of nodes working together, adding flexibility to the network's design.
5. **Interoperability**: Corda is designed to integrate easily with existing financial systems and other blockchain platforms, allowing businesses to leverage blockchain technology without overhauling their existing IT infrastructure.

Components of Corda

1. **Corda Nodes**: Each participant in a Corda network runs a node, which contains all the logic for conducting and verifying transactions. The nodes communicate peer-to-peer to minimize latency and maximize efficiency.
2. **Corda Ledger**: Instead of maintaining a global ledger accessible to all, Corda maintains individual ledgers for each transaction, shared only between the transacting parties. This ensures privacy while maintaining transparency where needed.
3. **Consensus Mechanism**: Corda uses a unique consensus mechanism focused on verifying the uniqueness of transactions (double-spend prevention) rather than reaching a global consensus. This allows for more efficient transaction processing, especially in financial applications.

Case Example: HSBC and Corda

HSBC, one of the world's largest financial institutions, adopted Corda to streamline its foreign exchange (FX) transactions. Previously, HSBC relied on outdated systems that were slow and prone to error, especially when managing complex trades across multiple jurisdictions. With Corda, HSBC can now track, execute, and settle FX transactions in real-time, significantly reducing settlement times and operational risks. Moreover, Corda's privacy features ensure that only relevant parties have access to transaction details, maintaining the confidentiality required in financial markets.

Benefits of Corda

1. **Privacy by Design**: Corda ensures that transaction data is only visible to the parties involved, which is crucial for industries like finance and healthcare that deal with sensitive information.
2. **Regulatory Compliance**: Corda is built with regulatory requirements in mind. Its architecture allows for easy auditing, and since only the relevant parties are privy to the transaction details, it complies with data privacy regulations such as GDPR.
3. **Cost Efficiency**: Corda eliminates the need for intermediaries in processes like clearing and settlement, reducing operational costs and increasing transaction speed.
4. **Scalability**: Corda's design ensures that it can handle high transaction volumes without compromising on speed or security. This makes it highly scalable for enterprise use cases.

Conclusion

Corda offers a tailored blockchain solution for enterprises, especially those in highly regulated sectors like finance. By focusing on privacy, efficiency, and interoperability, Corda provides businesses with the tools they need to adopt blockchain technology while maintaining regulatory compliance and data confidentiality. HSBC's successful use of Corda in FX trading highlights its potential to transform the financial industry, offering faster, more secure transactions and reduced operational costs. Corda's unique approach, especially its permissioned and privacy-centric design, sets it apart from other blockchain platforms in enterprise applications.

THADOMAL SHAHANI
TSEC
ENGINEERING COLLEGE

Assignment No. 1

Q What are the patterns problems with a centralized system?

⇒ Centralized system: In a centralized system, a single entity or a small group of entities controls the entire network. This centralized authority is responsible for validating transactions, maintaining the ledger & ensuring the security of the network traditional financial systems & many online services operate in a centralized manner.

Problem associated with centralized systems.

1) Single point of failure: In a centralized system all operations & data are managed by a single point of control. If this control server or authority fails or is compromised, the entire system can collapse.

eg :- If a bank's main server goes down due to technical issue or a cyber attack all the services dependent on the server will be disrupted potentially causing significant financial & operational damage.

2) Lack of transparency: Centralized systems often do not provide users with full visibility into the operations & data handling processes.

This opacity can lead to mistrust among the users & stakeholders.

Eg: In traditional banking, customers cannot see how their transaction are processed internally. They must trust the bank to hold their money correctly

3] Security Risk: Centralized system are attractive targets for hackers because compromising the central authority can grant access to the entire network.
Eg :- A successful hack on a centralized cryptocurrency exchange called result in the theft of all user funds stored on that platform

4] Censorship: the central authority has the power to control transactions it deems politically desirable, restricting citizens financial freedom.

5] High costs: Centralized systems often require significant investments in infrastructure security & maintenance to ensure reliability & performance.
Eg: Banks spend large amounts on secure data centers, cyber security measures & compliance with regulatory requirements which can drive up operational costs

THADOMAL SHAHANI TSEC ENGINEERING COLLEGE

1) Difference between centralized ; Decentralized & distributed system

Aspect	Centralized System	Decentralized System	Distributed System
1. Define	Single central server controls & manages all operations	Multiple nodes with Independent control. no central authority	Multiple interconnected nodes working together as a single system
2. Control	Centralized control with a single point of manager	Distributed control each node operate independently	Shared control nodes collaborate to achieve common goal
3. Single point of failure	High risk of the central server fails the whole system fails	Reduced risk failure of one node does not impact the entire system	Reduced risk designed for fault tolerance & redundancy
4) Scalability	limited	More scalable	Highly scalable
5) Management	Easier to manage centrally	More complex requires managing multiple nodes	Complex requires Coordination & management

THADOMAL SHAHANI TSEC ENGINEERING COLLEGE

6] Latency	Lower latency as operations are managed centrally	Can vary depends on the distance b/w the nodes	Potentially higher latency due to net work
7] Resource Utilization	Control central server resources are heavily utilized	Resources are spread across multiple nodes	Efficient resource sharing across nodes

Assignment 2

14/10/24

A

Q1)

⇒ Transaction in Blockchain:-

They represent the movement of cryptocurrency from one address to another address. A transaction includes the amount being sent, the senders digital signatures to prove ownership.

UTXO:-

Unspent transaction output is a concept used in certain Blockchain architecture like Bitcoin. It represents the change or balance of cryptocurrency owned by an address that hasn't been spent yet. This system increases privacy & efficiency

→ Double spending:

It is a potential issue in digital currencies where a user attempts to spend the same cryptocurrency more than once. It occurs when someone tries to send the same cryptocurrency to two different recipients simultaneously

Q2) ⇒ CORDA:-

An open source blockchain platform developed by R3 Primary focuses is on enabling private & secure transaction & contracts between different parties while ensuring regulatory compliance.

Ripple :- It is both a blockchain platform & a digital p
ment protocol. Its designed for real time cross payn
nts & settlements. Ripples focus is enabling fast
& cost effective international money transfer.

Quorum :- It is an enterprise focused, permissioned
blockchain platform developed by JPMC, Its built
on top of Ethereum but includes modifications fea-
ture tailored for private permissioned networks

Q3.

0, 1, 6 - confirmation refers to the no. of blocks
in blockchain.

0 confirmations means the transaction has been
broadcasted to the network but has not yet been
included in any block.

1 - confirmation : When a transaction receive 1
confirmation, it means that it has been included
in the latest block added to the blockchain

6 - confirmations : often considered the gold
standard for transaction security in crypto
when transaction has received 6 confirmation
it means that it has been included in 6 consecuti
blocks added to the block chain

Q4]

⇒ A smart contract is a self-executing contract with the terms of the agreement directly written into the code. It runs on blockchain, such as Ethereum, and automatically enforces & executes the terms of contract when predefined conditions are met. Smart contracts are transparent, immutable & decentralized.

Crowdfunding platforms can benifit from smart contracts in several ways

1. AFC Distribution Smart contracts can automatically collect funds from backers & distribute them to project creators when funding goals are met

2] Transparency & trust :- Since all transactions and contract terms are recorded on the blockchain, backers & project creators can trust that the funds will be handled as agreed

3] Conditional Payouts :- Smart contracts can release funds in stages based on project milestones. This ensures that project creators are incentivized to deliver on their promises & backers can see progress before more funds are released

4] Reduced fees :- Traditional crowdfunding is accessible to anyone with internet access, potentially attracting a global pool of backers

6] Global access: Blockchain-based crowdfunding platforms is accessible to anyone with internet access, potentially attracting a global pool of backers.