LP-III Blockchain Technology (2024-25)
Assignment: -01
MetaMask Installation

Students Name:	Roll No:
Batch:	BE Computer Div:

Aim: Installation of MetaMask and study spending Ether per transaction.

Theory: How to Install and Use Metamask

In order to use the blockchain, you need to install a mobile distributed application (dApp) browser. One of the most popular dApp browsers is known as Metamask. Follow the first set of instructions to install it on the Chrome browser, and then read on to start unlocking the power of the blockchain! Installing Metamask

Step 1. Go to the Metamask website.

Step 2. Click "Get Chrome Extension"/Firefox Addons to install Metamask.

Step 3. Click "Add to Chrome"/ "Add to Firefox in the upper right.

Step 4. Click "Add Extension" to complete the installation.

You will know Metamask has been installed when you see the fox logo on the upper right hand corner of your browser.

Installing Metamask App on Android/ios

Step 1. Go to the Metamask website/Playstore/app store

Step 2. Search and Click on install Metamask.

Step 3. Open Metamsk app

.Using Metamask

One of the first tasks you will want to do on Metamask is to install a wallet (or "vault" as Metamask calls it) to hold your cryptocurrencies. To do this, follow the instructions below.

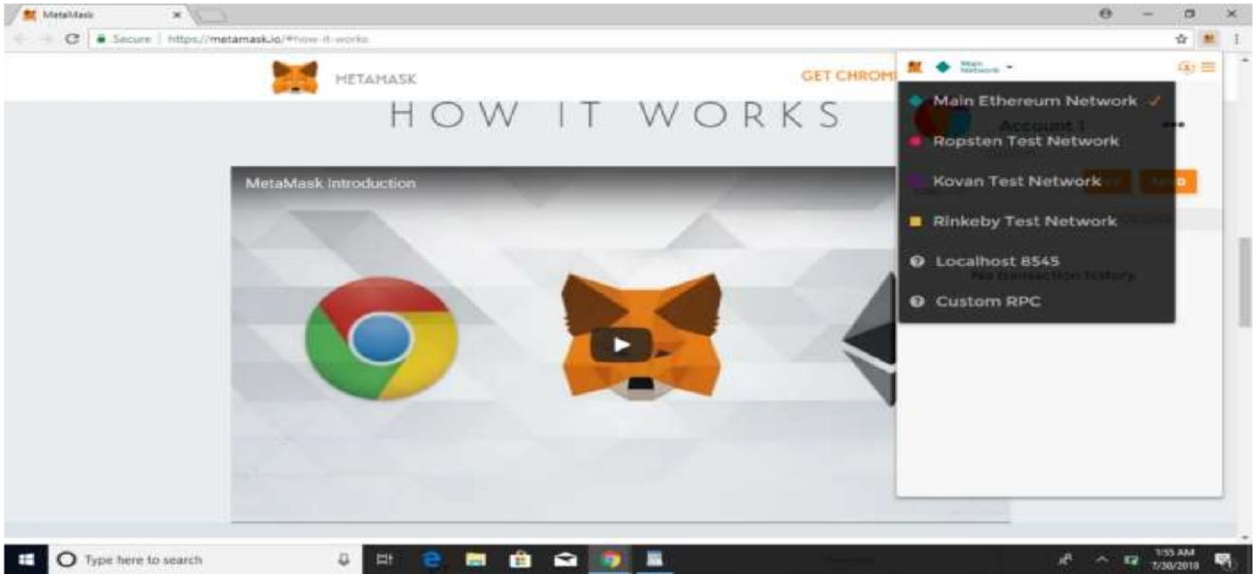
Step 1. Click on the Metamask logo in the upper right hand corner of your Google chrome browser. Step 2. Read and agree to the terms and conditions. You may have to agree to 2 to 3 pages worth of terms.

Step 3. Enter a password and click "Create" to create your wallet.

Step 4. You will see a set of 12 "seed words" for your vault. Click "Save Seed Words as File" and copy the "MetaMask Seed Words" file that is downloaded to a safe place. You will need it to access your vault.

Step 5. Click “I’ve Copied It Somewhere Safe” once your seed words file has been secured. You’ll be taken into your Metamask wallet!

You are now in the Ethereum mainnet network. To start experimenting with Metamask, you can switch to one of the testnet networks by clicking “Main Network” in the left hand corner of the wallet pop up screen, and selecting one of the testnets such as Ropsten Test Network or Kovan Test Network.



From these Test Networks, you can safely “buy” and “send” test Ether from a faucet and begin experimenting with the blockchain. Click around within the test network and start understanding how the Ethereum blockchain works!

MetaMask Wallet Transfer fees

MetaMask Wallet transfer fees are levied on several types of transactions. For the most part, it is charged on connections or transactions taking place via blockchain.

These transfer fees are based on the type of chain you use and often differ. Ethereum chain has a different fee structure whereas if you use the Binance Smart chain the transfer fee or network fee is different.

The transfer is a collective term which involves account fees, spot fees, fees given to miners and validators and tiered transaction fees.

Predominantly people question why the MetaMask transfer fee is so high? It is because it’s only one fee you are being charged. It includes other factors too.

Sending Transactions

Transactions are a formal action on a blockchain. They are always initiated in MetaMask with a call to the eth_sendTransaction method. They can involve a simple sending of ether, may result in sending tokens, creating a new smart contract, or changing state on the blockchain in any number of ways. They are always initiated by a signature from an external account, or a simple key pair.

Gas Price [optional]

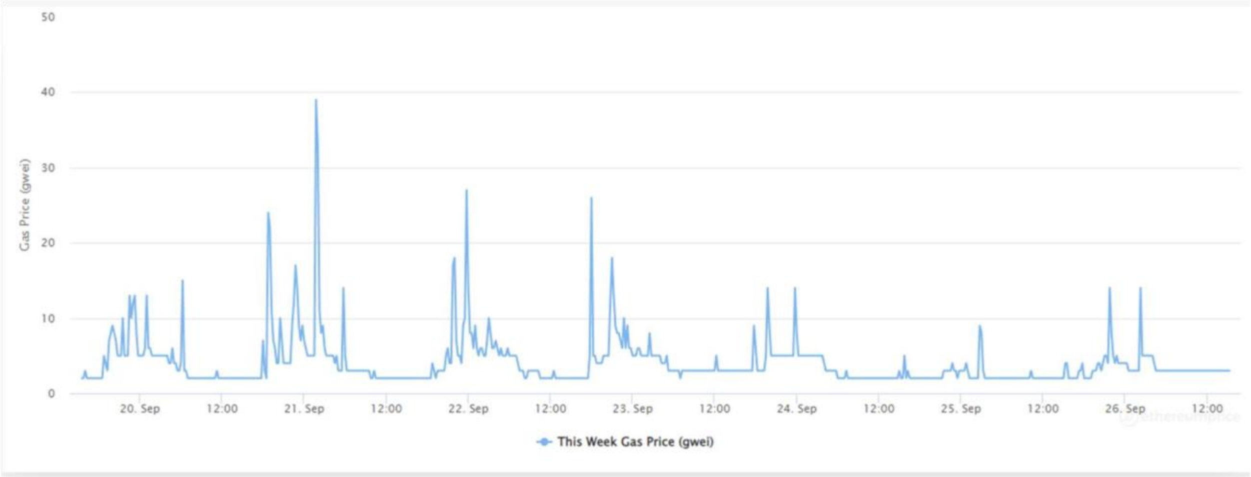
Optional parameter - best used on private blockchains.

In Ethereum, every transaction specifies a price for the gas that it will consume. To maximize their profit, block producers will pick pending transactions with higher gas prices first when creating the next block. This means that a high gas price will usually cause your transaction to be processed faster, at the cost of greater transaction fees. Note that this may not be true for e.g. Layer 2 networks, which may have a constant gas price or no gas price at all.

To [semi-optional]

A hex-encoded Ethereum address. Required for transactions with a recipient (all transactions except for contract creation).

Contract creation occurs when there is no to value but there is a data value.



https://etherscan.io/chart/gasprice

We estimate an empirical model based on queueing theory and analyse the factors determining the “gas price” (transaction cost per unit of service, “gas”). Using block- and transaction-level data from the Ethereum blockchain, changes in service demand significantly affect the gas price—when there is high block utilization, per-unit fees increase on average, with a strong nonlinear effect above 90% utilization. The transaction type is another important factor—a larger fraction of regular transactions (direct transfers between users) is associated with higher gas prices.

Conclusion: In this way we study Installation of MetaMask and study spending Ether per transaction. Also how to use test net or test blockchain for study different operations on blockchain. We study the economic determinants of transaction fees in the Ethereum blockchain.

QUESTIONS:-

1. What is MetaMask.
2. Compare MetaMask with other crypto wallet.
3. Explain Test net blockchain

4 study spending Ether per transaction.

LP-III Blockchain Technology (2024-25) Assignment: - 02 MetaMask crypto transactions	
Students Name:	Roll No:
Batch:	BE Computer Div

Aim : Create your own wallet using Metamask for crypto transactions.

Theory :

MetaMask is one of the most popular cryptocurrency wallets. Despite the ups and downs of the market, there are more and more users each year. MetaMask has about 21 million active users each month, 80 times more than in 2019 and significantly more than any noncustodial wallet. This article will explain why this wallet is so popular and its functions. Also, we will describe the development of a cryptocurrency wallet like MetaMask.

A Brief introduction to MetaMask

MetaMask is an open-source, straightforward, and easy-to-use cryptocurrency wallet. It functions as a web browser extension available for Chrome, Firefox, Brave, or a mobile application for iOS or Android. Initially, this wallet supported only Ether and ERC-20 tokens, and now it is compatible with ERC-721 and ERC-1155 token standards. Furthermore, MetaMask benefits include interaction with websites; hence, it can function as a connection node for various DApps on Ethereum.

Adrian Devis and Dan Finlay are the MetaMask developers. Their idea was revolutionary and straightforward; they intended to create a web browser extension that would allow managing cryptocurrency and using the browser for fast and secure access with DApps. ConsenSys Software Inc. — a development company, focusing on applications that use Ethereum's blockchain, implemented the idea in 2016.

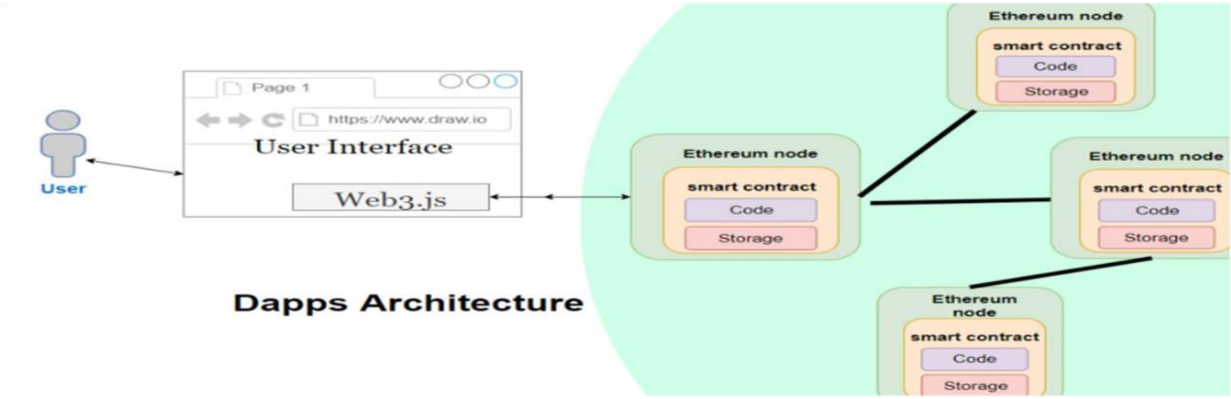
The solution used Ethereum's interface and a web API called web3.js. This Ethereum library is the fundament of MetaMask since it allows the browser to interact with the local or remote blockchain nodes via HTTP, IPC, and WebSocket; also, it gained the ability to record and read data from smart contracts, transfer tokens, etc. In another way, web3.js allowed the blockchain developers to create proxy and communication bridges between MetaMask, DApps, and the user.

Adrian Devis and Dan Finlay admit that their idea was great. Yet, the technical implementation was super complicated, especially in providing security for the users (web wallets are considered the most vulnerable to hacker attacks). Nonetheless, ConsenSys succeeded, and on the 14th of July in 2016, they offered the first version of MetaMask web browser cryptocurrency wallet for Chrome. Later, they presented a version for Firefox, Brave, and other popular browsers. In 2019 they also launched the mobile version of the MetaMask cryptocurrency wallet.

How does the MetaMask wallet function?

As we mentioned above, the MetaMask cryptocurrency wallet employs the web3.js library to function. This library is a part of the official Ethereum product. The library was developed focusing on the requirements of web applications that could interact with the Ethereum blockchain and take advantage of all blockchain's benefits and functions.

MetaMask is a cryptocurrency wallet for Ethereum and an instrument that helps to interact with DApps. MetaMask connects the extension to the DApp so that to fulfill both tasks. When the application identifies the MetaMask, it creates a connection, and the user can start using all the features of a specific application.



For instance, it can assets trading, access to resources or services, or any other task within the capability of a DApp. Each action has its cost (transaction fee) that must be paid in Ethereum or any specified token. MetaMask wallet has all instruments and protocols for this purpose.

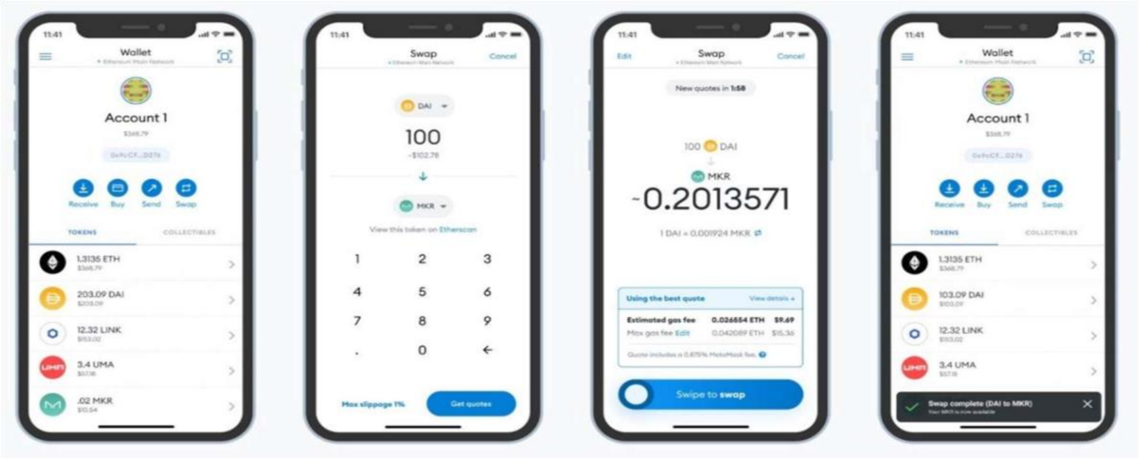
Hence, we can state that Metamusk also controls the interaction of the user and DApp, and processes the operations required for specific actions, besides the function of a wallet. Reliable and secure cryptography and safe internet connection are the environments for these operations.

Furthermore, MetaMask can generate asymmetric keys, store them on a local device, and manage access to the keys. To sum up, MetaMask is a super-safe extension.

MetaMask wallet key features

Easy to use. The first and most apparent benefit of the MetaMask wallet is the ease of use. The wallet offers an intuitive and straightforward user interface that makes the management of cryptocurrencies and interaction with DApps easier than ever before.

Furthermore, it allows the creation of several wallets. When users create a new wallet, they generate new public and private keys that MetaMask users have fast access.



Integration with various DApps. MetaMask’s users can connect with numerous decentralized applications with a single tap on the screen. It allows them to exchange their tokens swiftly on Uniswap or PancakeSwap exchangers, launch blockchain games like Gods Unchained or Decentraland, buy NFTs on OpenSea or Rarible. All these actions do not require complicated processes. The only movement is connecting the MetaMask cryptocurrency wallet.

Integration with other blockchains. The additional benefit of MetaMask is that you can join it to other blockchains, e.g., Avax, Polygon, Fantom, Binance Smart Chain (BSC).

Hardware wallets support. MetaMask service is compatible with hardware cryptocurrency wallets like Ledger, Nano, or Trezor. All you need is to tap the ‘connect a hardware wallet’ button. This is a significant benefit that helps to interact with applications that do not support Trezor and Ledger hardware wallets.

Swaps support. In 2021 MetaMask offered its users a relatively cheap and fast opportunity to exchange their tokens within the wallet with Swaps. This function supports several automatic marker makers (MM) that help find the best token exchange rate.

NFT support. MetaMask wallet also allows storing NFTs. When a user buys an NFT on a platform, the token is automatically displayed in the tab of collectible items in the wallet.

Main functions of a cryptocurrency wallet

To develop a cryptocurrency wallet like MetaMask, one should consider two roles of users: regular users and administrators. The average user apparently will use the wallet to store, trade, and exchange tokens. The administrators are the employees responsible for the proper functioning and management of the wallets. Often, they solve the issues of the regular users.

Here is a list of functions for regular users:

- Registration. It must be easy to create a new profile. The essentials are opening an internet browser, downloading the MetaMask extension from the official website, and coming up with a complicated password. Next, the user will need to put down the seed phrase, and they are ready to go. If one already has a cryptocurrency address, one can easily connect it to the cryptocurrency wallet.

- Applicable exchange rate. MetaMask displays only the amount of tokens on the account, while other cryptocurrency wallets show the exchange rate for tokens in the investment profile. If you plan to develop a MetaMask crypto wallet clone, it is a sound idea to add these features, which will enhance the user experience of your upcoming project.
- Operations with the cryptocurrency assets. The main functions of any wallet are transferring, receiving, and exchanging cryptocurrencies; hence, it is vital that these functions are straightforward so that any user is confident in using the wallet. Furthermore, it is an advantage if there is a chance to check the current balance and transaction history. If you have experience trading on an exchange, you will expect MetaMask to display the history of operations and the current balance; yet MetaMask does not offer it. Pay special attention to it if you intend to develop a wallet.
- Favorite addresses. This is another feature that MetaMask is not showing, while the other wallets do. It allows the users to make a list of addresses for fast token transfers. It is super convenient for minor regular operation fees.
- Employing other blockchains. After installing the MetaMask extension or application, the user can use only Ether and ERC-20 tokens. However, the users can connect to other blockchains. What is more, they will have a single address for all blockchains.
- NFT support. Recently, MetaMask wallet also started supporting non-fungible tokens.

Here is a list of necessary functions for platform administrators.

- Tools panel. The administrators can use this tab to trace the current state of the platform, current issues, the number of users, transaction volume, etc.
- User management. This tab helps the administrators track the users' status and manage them: add, delete, block, communicate, verify, etc.
- Managing the fees. It helps to add, delete, or change the transaction fees.
- Payment system management. MetaMask wallet does not offer to link the credit or debit card, PayPal, or another payment system. However, if the wallet offers these functions, the administrators require instruments to manage it.
- Content management. These are the user interface management tools (changing texts, icons, links, etc.).
- Connecting Facebook, Twitter, and Instagram accounts.
- Reports and analytics.
- API documentation

Extended functions set for MetaMask clone

To help your MetaMask wallet clone become famous, you should add some advantages that highlight it from the competitors and improve the user experience.

These can be the following:

- Linking an account. Your users will find it useful to be able to buy a cryptocurrency and exchange it for fiat within the wallet. This will be possible if you develop a wallet like MetaMask and add the feature of linking bank accounts, credit/debit cards, PayPal, or other online payment systems.
- eCommerce integrations. We mean integrating the wallet with exchanges, NFT marketplaces, decentralized applications, shops, and other services that the users might find useful.
- Multilingual interface. If you focus on a market where all people speak the same language, you might neglect this aspect. However, your intentions are global, and you should add as many languages as possible to increase the target audience.
- Push notifications. The notifications will inform the users of receiving payments, ending transactions, rapid exchange rate changes in the investment account, system updates, suspicious activity, etc.
- VIP support. Numerous cryptocurrency trading platforms offer support for an additional fee. This may include 24/7 support, communication with a personal specialist, etc.
- QR scanner. This is another useful feature that allows your users to make payments even faster. Moreover, it will decrease the number of transfers done by mistake.

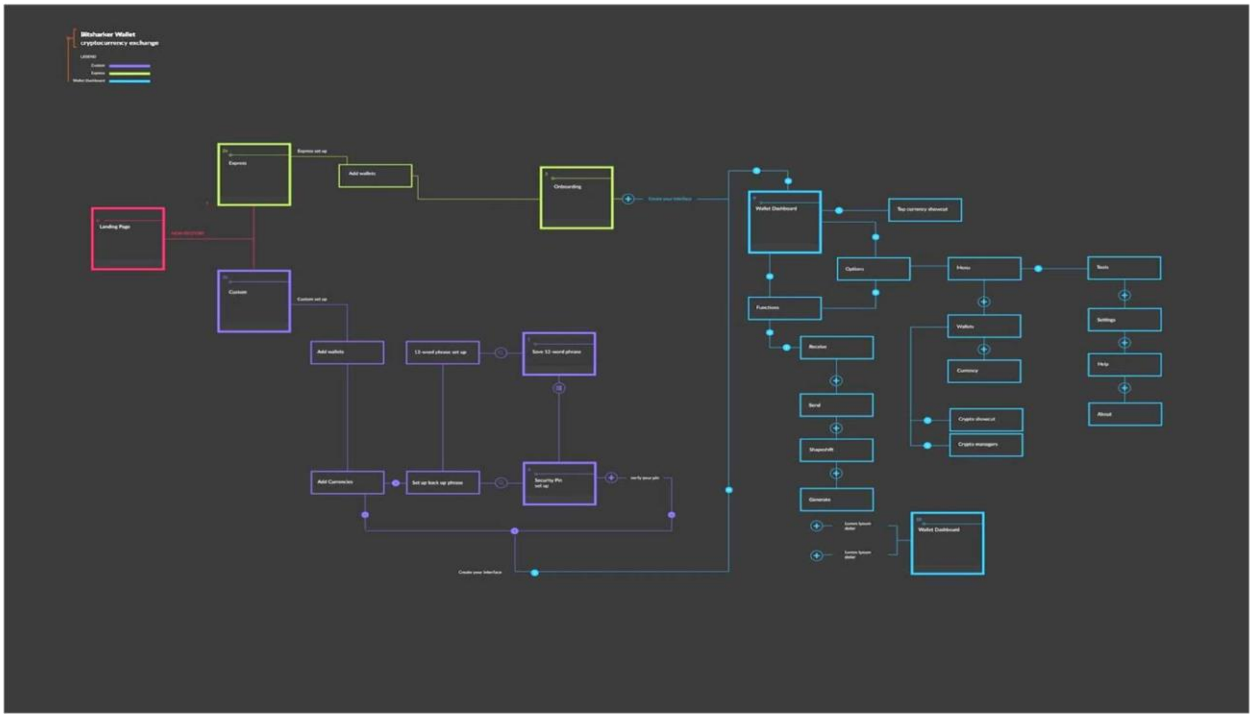
The process of wallet development

Step 1: Opening

It is necessary to develop the concept thoroughly before starting with the design and program code. If you intend to develop a MetaMask clone, you should develop ideas that will attract the users and make them use your extension instead of the MetaMask. Dedicated research and analyzing the market, competitors, and the target audience will help.

Step 2: Developing the architecture

According to the data collected in the first step, an information architecture of the upcoming product will be created. It is a scheme showing the functions, system components, and their interaction.

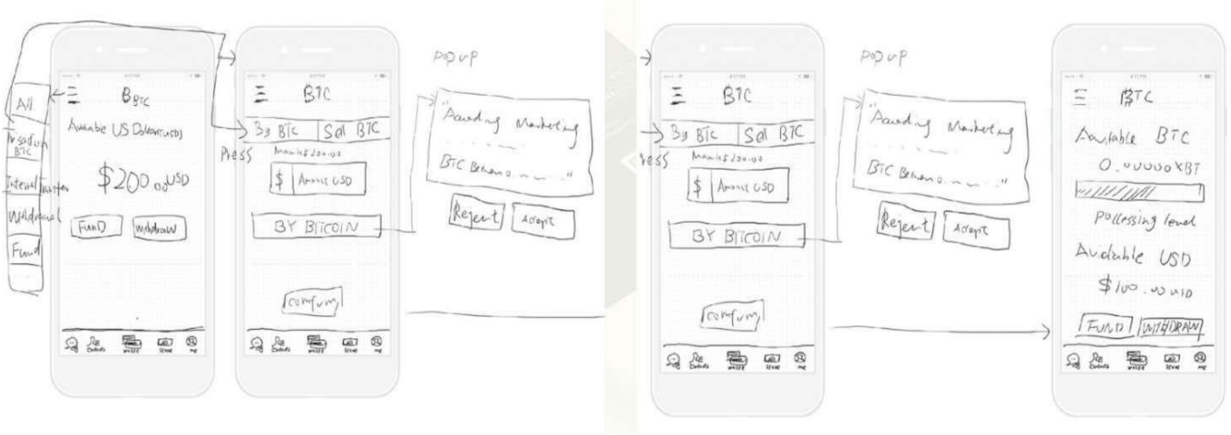


An example of the information architecture of a cryptocurrency wallet.

Step 3: UX/UI design development

The users will appreciate your MetaMask clone if it has a beautiful and intuitive user interface design. UX/UI designers are responsible for creating the designs. They study the project goals, business analytics, tasks, and user requirements. Based on that information, they develop the low-fidelity and high-fidelity wireframes and the prototypes of the upcoming cryptocurrency wallet.

Low-fidelity wireframes are basic sketches of the design that represent the tabs and their content. Here are a few examples:



Low-fidelity wireframes of a cryptocurrency wallet.

Next, the designers develop high-fidelity wireframes. This is how the users will see it. Here are several examples of high-fidelity wireframes for a cryptocurrency wallet. If the client and the target audience (focus groups) like the high-fidelity wireframes, they become the interface prototypes with all animation, transfers between screens, etc., included, yet there is no wallet functionality. The prototypes allow receiving more effective feedback to optimize the design.

When the design prototypes are ready, the coding team starts implementing the technical aspect of your MetaMask clone. Backend and frontend developers use various tools (technical stack) to make the development process faster, cheaper, and simpler. The developers analyze the task, the targets, knowledge, and experience, and then select the instruments.

Step 4: writing the code

Technical stack for backend development of a MetaMask clone:

- Smart contracts: Ethereum virtual machine or BSC.
- SQL databases: MySQL, PostgreSQL, MariaDB, MS SQL, Oracle.
- DevOps: GitLab CI, TeamCity, GoCD Jenkins, WS CodeBuild, Terraform.
- NoSQL databases: MongoDB, Cassandra, DynamoDB.
- Search engines: Apache Solr, Elasticsearch.
- Programming languages: Java, PHP, Python.
- Frameworks: Spring, Symphony, Flask.
- Cache: Redis, Memcached.

Tech stack for frontend development of a MetaMask clone:

- Programming languages for web: Angular.JS, React.JS, and Vue.JS.
- Programming languages for App: Java, Kotlin for Android, and Swift for iOS.
- Architecture: MVVM for Android and MVC, MVP, MVVM, VIPER for iOS.
- IDE: Android Studio and Xcode for iOS.
- SDK: Android SDK and iOS SDK.

Step 5: product testing

To develop a wallet, you should test the product’s functionality, usability, workload, smart contract functioning, meeting the specifications, etc. QA specialists are responsible for it. A good solution is that the QA team starts testing the product right after the development launch. It will help to avoid accumulating issues. Furthermore, it requires independent analysts like Certik or Techrate to test the smart contracts of your wallet.

Step 6: deployment, release

After your MetaMask cryptocurrency wallet clone is ready and tested, it requires deployment in the cloud (AWS, Azure) or on local servers of your company. If you have a mobile version of the wallet, you should release it on Google Play, App Store, Microsoft Store, or Amazon App Store.

Step 7: product support,

In the end, you need to establish a support service for your cryptocurrency wallet, which will help users solve problems and collect feedback for future product improvements.

Using Metamask

One of the first tasks you will want to do on Metamask is to install a wallet (or "vault" as Metamask calls it) to hold your cryptocurrencies. To do this, follow the instructions below.

Step 1. Click on the Metamask logo in the upper right hand corner of your Google chrome browser. Step 2. Read and agree to the terms and conditions. You may have to agree to 2 to 3 pages worth of terms.

Step 3. Enter a password and click "Create" to create your wallet.

Step 4. You will see a set of 12 "seed words" for your vault. Click "Save Seed Words as File" and copy the "MetaMask Seed Words" file that is downloaded to a safe place. You will need it to access your vault.

Step 5. Click "I've Copied It Somewhere Safe" once your seed words file has been secured. You'll be taken into your Metamask wallet!

You are now in the Ethereum mainnet network. To start experimenting with Metamask, you can switch to one of the testnet networks by clicking "Main Network" in the left hand corner of the wallet pop up screen, and selecting one of the testnets such as Ropsten Test Network or Kovan Test Network.

Questions :

1. What is Digital Wallet.
2. Explain block header.
3. Explain Ethereum and Ether.
4. Compare Bitcoin,Ether.

LP-III Blockchain Technology (2024-25) Assignment: - 03 **Write a smart contract on a test network, for Bank account of a customer**	
Students Name:	Roll No:
Batch:	BE Computer Div:

Aim : Write a smart contract on a test network, for Bank account of a customer for following operations: Deposit money, Withdraw Money
Show balance

Theory :

1. Writing a Banking Contract

This will demonstrate how to write a simple, but complete, smart contract in Solidity that acts like a bank that stores ether on behalf of its clients. The contract will allow deposits from account, and can be trusted to allow withdrawals by accounts that have sufficient funds to cover the requested withdrawal.

This post assumes that you are comfortable with the ether-handling concepts introduced in our post, Writing a Contract That Handles Ether.

That post demonstrated how to restrict ether withdrawals to an "owner's" account. It did this by persistently storing the owner account's address, and then comparing it to the msg.sender value for any withdrawal attempt. Here's a slightly simplified version of that smart contract, which allows anybody to deposit money, but only allows the owner to make withdrawals:

MetaMask: The crypto wallet for Defi, Web3 Dapps and NFTs
https://metamask.io

MetaMask provides an essential utility for blockchain newcomers, token traders, crypto gamers, and developers

Remix Project

Remix Project is a platform for development tools that use a plugin architecture. It encompasses sub-projects including Remix Plugin Engine, Remix Libraries, and of course Remix IDE.

Remix IDE is an open source web and desktop application. It fosters a fast development cycle and has a rich set of plugins with intuitive GUIs. Remix is used for the entire journey of contract development with Solidity language as well as a playground for learning and teaching Ethereum.

Start developing using Remix on browser, visit: https://remix.ethereum.org

For desktop version, see releases: https://github.com/ethereum/remix-desktop/releases

Remix libraries work as a core of native plugins of Remix IDE

Step 1: Open Remix IDE on any of your browsers, select on the New File and click on Solidity to choose the environment.

Step 2: Write the Smart contract in the code section, File Save with .sol (solidity) Extension and

Step 3 :Select compiler and click the Compile button under the Compiler window to compile the contract. In remix solidity compiler window have option Auto compiler and hide warnings

Step 4: To execute the code, click on the Deploy button under Deploy and Run Transactions window.

Step 4: After deploying the code click on the method calls under the drop-down of deployed contracts to run the program, and for output, check to click on the drop-down on the console There are option for deployment like Remix VM, Hardhat provider, External Http Provider and Wallet connect to connect metamask & other types of supporting Wallets.

Step 5: For debugging click on the Debug button corresponding to the method call in the console. Here you can check each function call and variable assignments.

////Solidity Bank Deposit, Withdraw, and view balance program

```
pragma solidity 0.4.25;

contract Bank
{
    int bal;

    constructor() public
    {
        bal=1;
    }
    function getBalance() view public returns(int)
    {
        return bal;
    }
    function withdraw(int amt) public
    {
       if(bal < amt)
       {
           bal

       }

       else{
           bal = bal -amt;
       } }

    function Deposit(int amt) public
    {
        bal =bal + amt;
    }}
```

Conclusion

We have study how to use remix.ethereum ide and create,compile and deploy program/project to Remix VM

Questions:

1. Explain smart contract.
2. Explain remix.ethereum ide
3. Write note on Solidity programming language.
4. Explain Gas Value.
5. Explain types of ethereum accounts.

LP-III Blockchain Technology (2024-25) Assignment: - 04 **Write a program in solidity to create Student data**	
Students Name:	Roll No:
Batch:	BE Computer Div:

Aim - Write a program in solidity to create Student data. Use the following constructs:

·Structures
·Arrays
·Fallback

Deploy this as smart contract on Ethereum and observe the transaction fee and Gas values.

Theory :

Solidity – Deploy a Smart Contract for Marks Management System

Solidity is a high-level language. The structure of smart contracts in solidity is very similar to the structure of classes in object-oriented languages. The solidity file has an extension .sol.

What are Smart Contracts?

Solidity's code is encapsulated in contracts which means a contract in Solidity is a collection of code (its functions) and data (its state) that resides at a specific address on the Ethereum blockchain. A contract is a fundamental block of building an application on Ethereum.

Example: In the below example, the aim is to deploy a Smart Contract for Marks Management System by using Solidity. In this contract, the details of every student like student ID, Name, Marks, etc can be added and if one wants to give some bonus marks to students then they can also be added. After building the contract all the details of every student can be retrieved.

Approach:

The first step is to deploy the smart contract using the Remix IDE. After writing the code compile the code. When it is successfully compiled then deploy it. After deploying the contract a deployed Contract is obtained and then add the student details one by one.

If bonus marks need to be added then add in the bonusMarks section after that click on stdCount and fetch the student details to call the stdRecords.

Add one or more new student details in this Smart Contract by the increment of stdCount.

Implementation

Step 1: Open Remix IDE.

Step 2: Click on File Explorers and select Solidity in the environment and create a new file StudentMarksMangmtSys.sol by clicking on New File section.

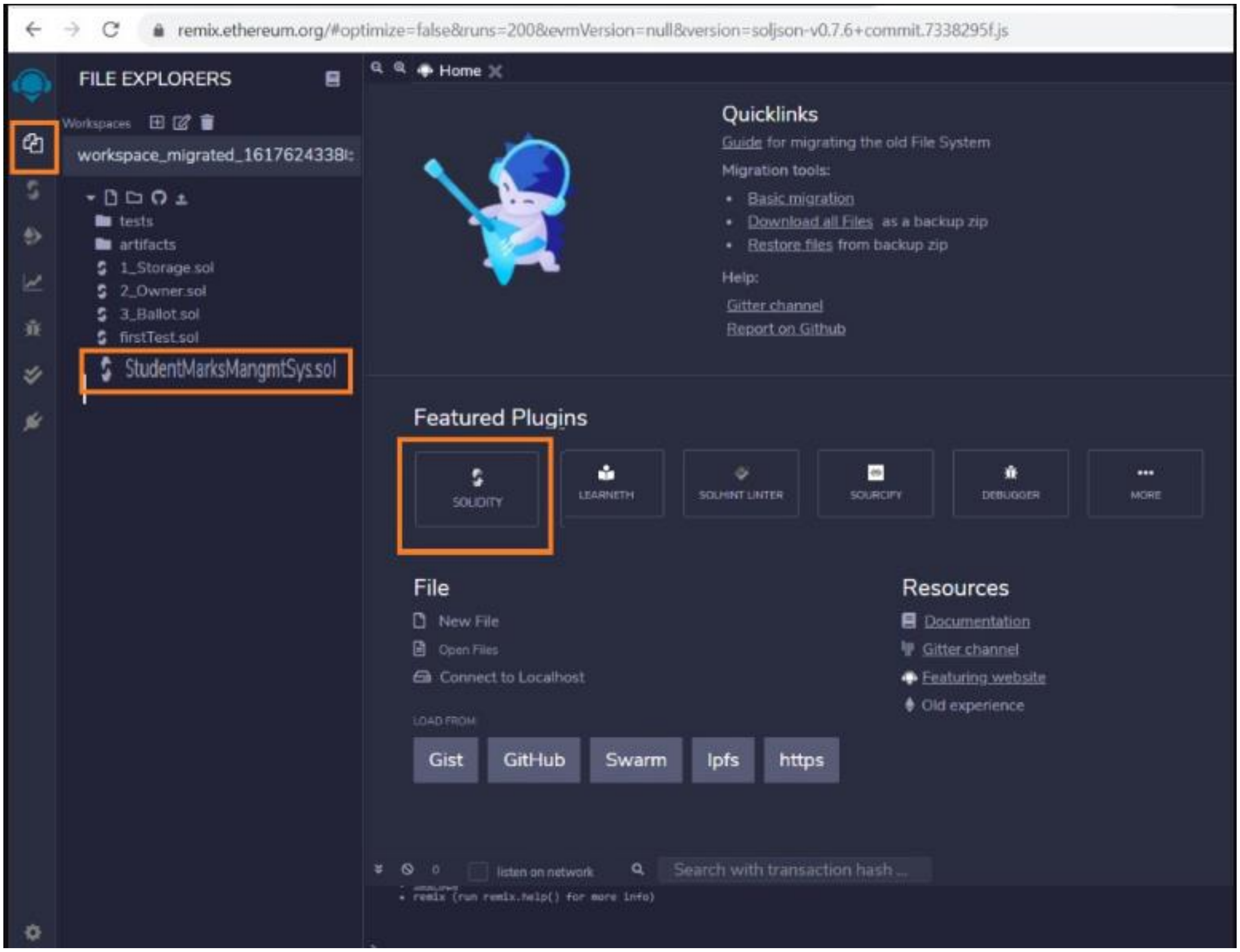


Fig. Remix IDE

Step 3: Build a smart contract that contains all the details of the student with the help of Remix IDE by clicking on the file name.

```
// Solidity program to implement
// the above approach
pragma solidity >= 0.7.0<0.8.0;

// Build the Contract
contract MarksManagmtSys
{
   // Create a structure for
   // student details
   struct Student
   {
      int ID;
      string fName;
      string lName;
      int marks;
   }
```

```
    address owner;
    int public stdCount = 0;
    mapping(int => Student) public stdRecords;

    modifier onlyOwner
    {
        require(owner == msg.sender);
        _;
    }
    constructor()
    {
        owner=msg.sender;
    }

    // Create a function to add
    // the new records
    function addNewRecords(int _ID,
                         string memory _fName,
                         string memory _lName,
                         int _marks) public onlyOwner
    {
        // Increase the count by 1
        stdCount = stdCount + 1;

        // Fetch the student details
        // with the help of stdCount
        stdRecords[stdCount] = Student(_ID, _fName,
                                   _lName, _marks);
    }

    // Create a function to add bonus marks
    function bonusMarks(int _bonus) public onlyOwner
    {
         stdRecords[stdCount].marks =
                 stdRecords[stdCount].marks + _bonus;
    }
}
```

Step 4: After building the contract compile it. Select the compiler version before clicking on Compile button.

Step 5: After successful compilation, to deploy the contract, select the Environment JavaScript VM (Berlin) before clicking on the Deploy button.

Step 6: If the contract is successfully deployed then deployed contract is obtained. Open the deployed contract and add the student details and transact it.

Step 7: Add the bonus marks if you want to give them to the student and transact it after that click on the stdCount. One can see the student details after calling the stdRecords by entering the stdCount.

Conclusion: In this way we have study Write a program in solidity to create Student data.

Questions:

1. What are Smart Contracts?
2. What is Ethereum blockchain ?
3. How to Deploy smart contact on Ethereum
4. Observe and Write down transaction fee and Gas values analysis

LP-III Blockchain Technology (2024-25) Assignment: - 05 **Survey report on types of Blockchains**	
Students Name:	Roll No:
Batch:	BE Computer Div:

Aim : **Write a survey report on types of Blockchains and its real time use cases.**

Theory :

What is Blockchains :-

Blockchain is a revolutionary technology that is making a great impact on modern society due to its transparency, decentralization, and security properties. Blockchain gained considerable attention due to its very first application of Cryptocurrencies e.g., Bitcoin. In the near future, Blockchain technology is determined to transform the way we live, interact, and perform businesses. Recently, academics, industrialists, and researchers are aggressively investigating different aspects of Blockchain as an emerging technology. Unlike other Blockchain surveys focusing on either its applications, challenges, characteristics, or security, we present a comprehensive survey of Blockchain technology's evolution, architecture, development frameworks, and security issues. We also present a comparative analysis of frameworks, classification of consensus algorithms, and analysis of security risks & cryptographic primitives that have been used in the Blockchain so far. Finally, this paper elaborates on key future directions, novel use cases and open research challenges, which could be explored by researchers to make further advances in this fieldConsider the problem having weights and profits are:

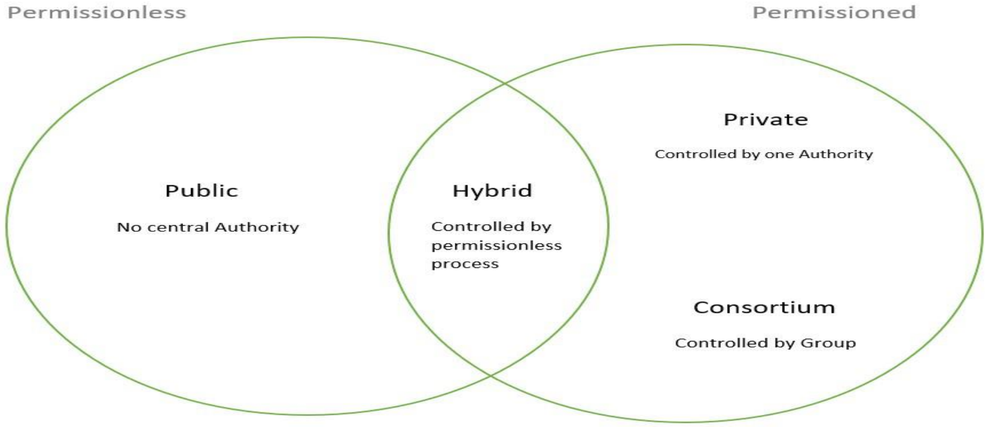


Fig. Type of BlockChain

1. Public Blockchain

These blockchains are completely open to following the idea of decentralization. They don't have any restrictions, anyone having a computer and internet can participate in the network.

- As the name is public this blockchain is open to the public, which means it is not owned by anyone.

- Anyone having internet and a computer with good hardware can participate in this public blockchain.
- All the computer in the network hold the copy of other nodes or block present in the network
- In this public blockchain, we can also perform verification of transactions or records

Advantages:

- **Trustable:** There are algorithms to detect no fraud. Participants need not worry about the other nodes in the network
- **Secure:** This blockchain is large in size as it is open to the public. In a large size, there is greater distribution of records
- **Anonymous Nature:** It is a secure platform to make your transaction properly at the same time, you are not required to reveal your name and identity in order to participate.
- **Decentralized:** There is no single platform that maintains the network, instead every user has a copy of the ledger.

Disadvantages:

- **Processing:** The rate of the transaction process is very slow, due to its large size. Verification of each node is a very time-consuming process.
- **Energy Consumption:** Proof of work is high energy-consuming. It requires good computer hardware to participate in the network
- **Acceptance:** No central authority is there so governments are facing the issue to implement the technology faster.

Use Cases: Public Blockchain is secured with proof of work or proof of stake they can be used to displace traditional financial systems. The more advanced side of this blockchain is the smart contract that enabled this blockchain to support decentralization. Examples of public blockchain are Bitcoin, Ethereum.

2. Private Blockchain

These blockchains are not as decentralized as the public blockchain only selected nodes can participate in the process, making it more secure than the others.

- These are not as open as a public blockchain.
- They are open to some authorized users only.
- These blockchains are operated in a closed network.
- In this few people are allowed to participate in a network within a company/organization.

Advantages:

- **Speed:** The rate of the transaction is high, due to its small size. Verification of each node is less time-consuming.
- **Scalability:** We can modify the scalability. The size of the network can be decided manually.
- **Privacy:** It has increased the level of privacy for confidentiality reasons as the businesses required.
- **Balanced:** It is more balanced as only some user has the access to the transaction which improves the performance of the network.

Disadvantages:

- Security- The number of nodes in this type is limited so chances of manipulation are there. These blockchains are more vulnerable.
- Centralized- Trust building is one of the main disadvantages due to its central nature. Organizations can use this for malpractices.
- Count- Since there are few nodes if nodes go offline the entire system of blockchain can be endangered.

Use Cases: With proper security and maintenance, this blockchain is a great asset to secure information without exposing it to the public eye. Therefore companies use them for internal auditing, voting, and asset management. An example of private blockchains is Hyperledger, Corda.

3. Hybrid Blockchain

It is the mixed content of the private and public blockchain, where some part is controlled by some organization and other makes are made visible as a public blockchain.

- It is a combination of both public and private blockchain.
- Permission-based and permissionless systems are used.
- User access information via smart contracts
- Even a primary entity owns a hybrid blockchain it cannot alter the transaction

Advantages:

- **Ecosystem:** Most advantageous thing about this blockchain is its hybrid nature. It cannot be hacked as 51% of users don't have access to the network
- **Cost:** Transactions are cheap as only a few nodes verify the transaction. All the nodes don't carry the verification hence less computational cost.
- **Architecture:** It is highly customizable and still maintains integrity, security, and transparency.
- **Operations:** It can choose the participants in the blockchain and decide which transaction can be made public.

Disadvantages:

- **Efficiency:** Not everyone is in the position to implement a hybrid Blockchain. The organization also faces some difficulty in terms of efficiency in maintenance.
- **Transparency:** There is a possibility that someone can hide information from the user. If someone wants to get access through a hybrid blockchain it depends on the organization whether they will give or not.
- **Ecosystem:** Due to its closed ecosystem this blockchain lacks the incentives for network participation.
-

Use Case: It provides a greater solution to the health care industry, government, real estate, and financial companies. It provides a remedy where data is to be accessed publicly but needs to be shielded privately. Examples of Hybrid Blockchain are Ripple network and XRP token.

4. Consortium Blockchain

It is a creative approach that solves the needs of the organization. This blockchain validates the transaction and also initiates or receives transactions.

- Also known as Federated Blockchain.
- This is an innovative method to solve the organization's needs.
- Some part is public and some part is private.
- In this type, more than one organization manages the blockchain.

Advantages:

- **Speed:** A limited number of users make verification fast. The high speed makes this more usable for organizations.

- **Authority:** Multiple organizations can take part and make it decentralized at every level. Decentralized authority, makes it more secure.
- **Privacy:** The information of the checked blocks is unknown to the public view. but any member belonging to the blockchain can access it.
- **Flexible:** There is much divergence in the flexibility of the blockchain. Since it is not a very large decision can be taken faster.

Disadvantages:

- **Approval:** All the members approve the protocol making it less flexible. Since one or more organizations are involved there can be differences in the vision of interest.
- **Transparency:** It can be hacked if the organization becomes corrupt. Organizations may hide information from the users.
- **Vulnerability:** If few nodes are getting compromised there is a greater chance of vulnerability in this blockchain

Use Cases: It has high potential in businesses, banks, and other payment processors. Food tracking of the organizations frequently collaborates with their sectors making it a federated solution ideal for their use. Examples of consortium Blockchain are Tendermint and Multichain.

Use Cases	Representative Countries	Focus
Medical and healthcare	China, United States, Switzerland, Phillippines, Japan, Brazil, etc.	Supply chain, Internet-of-Things, etc.
Financial applications	(Almost) All	Cryptocurrencies, asset management, etc.
Critical infrastructures	South Korea	Asset management, optimization, etc.
Blockchain city	Malaysia	Cryptocurrency, data management
Asset management	Georgia, Sweden, Switzerland	Land registry, property transactions, etc.
Education	Japan, Malta	Certificate management
Data management	Phillipines, Australia	Cloud data management

Conclusion:

We can see that the different types of Blockchains, it advantages-disadvantages and its real time use cases.

Questions:

1. Explain types of blockchain
2. What is use cases
3. Write Blockchain case study
4. Write a survey report on types of Blockchains and its real time use cases