Digital Collectibles

PROBLEM STATEMENT :

The upcoming slides examines the issue of major platforms such as Steam, Google Playstore, and iOS Store dominating the game asset market.

A proposed solution involves using Ethereum blockchain to make buying and selling assets fairer.

The project aims to address this problem and explain the workings of Ethereum blockchains in a simple manner

The Code

```
public String buyInGameAsset(String playerId, int quantity, int purchaseAmount) {
```

Player Initiates Purchase Request

```
    if (purchaseAmount >=
            quantity * Inventory.getsellingPriceFor_PCGParachutes_()
                && Inventory.getTotalSupply_PCGParachutes_() >= quantity) {

        boolean paymentSuccess =
            PaymentService.transferMoneyToGame(purchaseAmount);
        /**
         * Will Initiate transfer of money from player to game, using UPI, card payment
         */

        if (!paymentSuccess) {
            return "FALIURE";
        }
        Inventory.reduceTotalSupply(quantity);
        /** Reduce totalSupply of _PCGParachutes_ */

        Inventory.add_PCGParachutes_ForPlayer(playerId, quantity);
        /** Add "quantity" amount of _PCGParachutes_ to playerId */

        return "SUCCESS";
    } else {
        return "FALIURE";
    }
}
```

Completes the Card Payment

The Item is transferred to player’s account

Buying Digital Assets

PLAYER

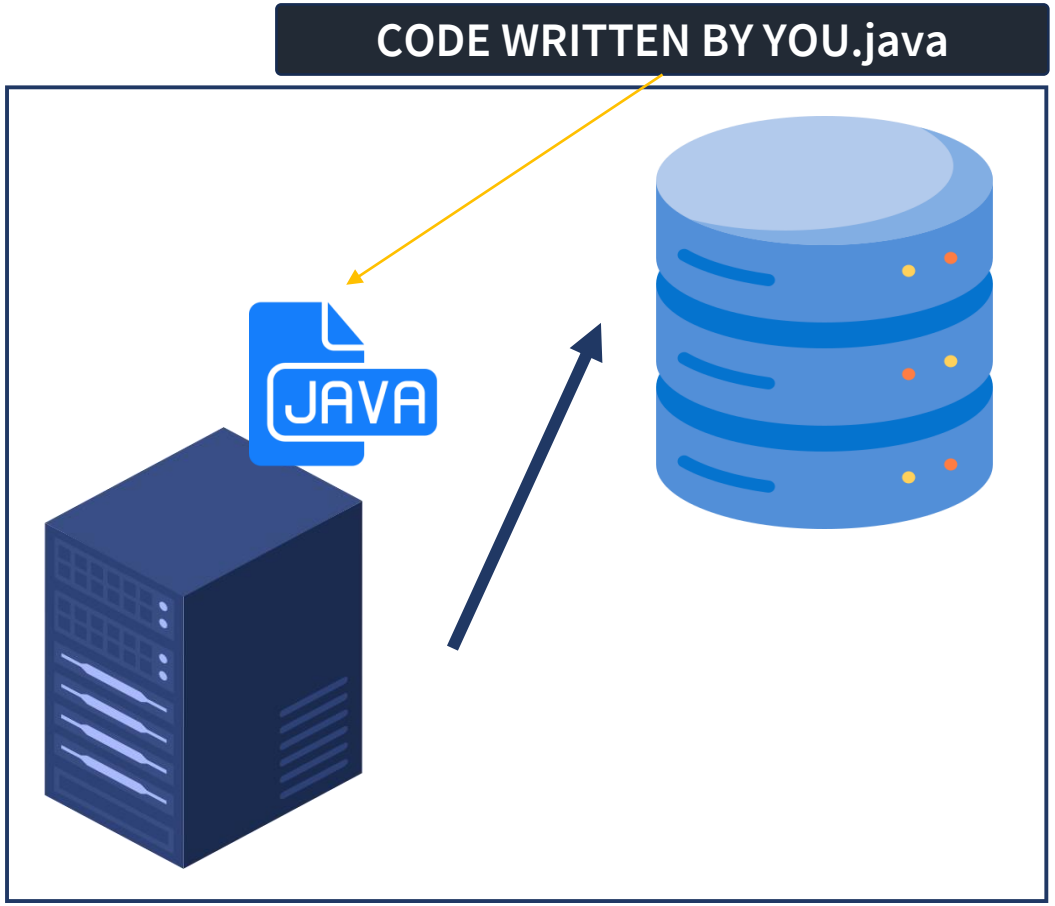
1. Buy Asset Request

2. Card, Wallet, UPI Payment

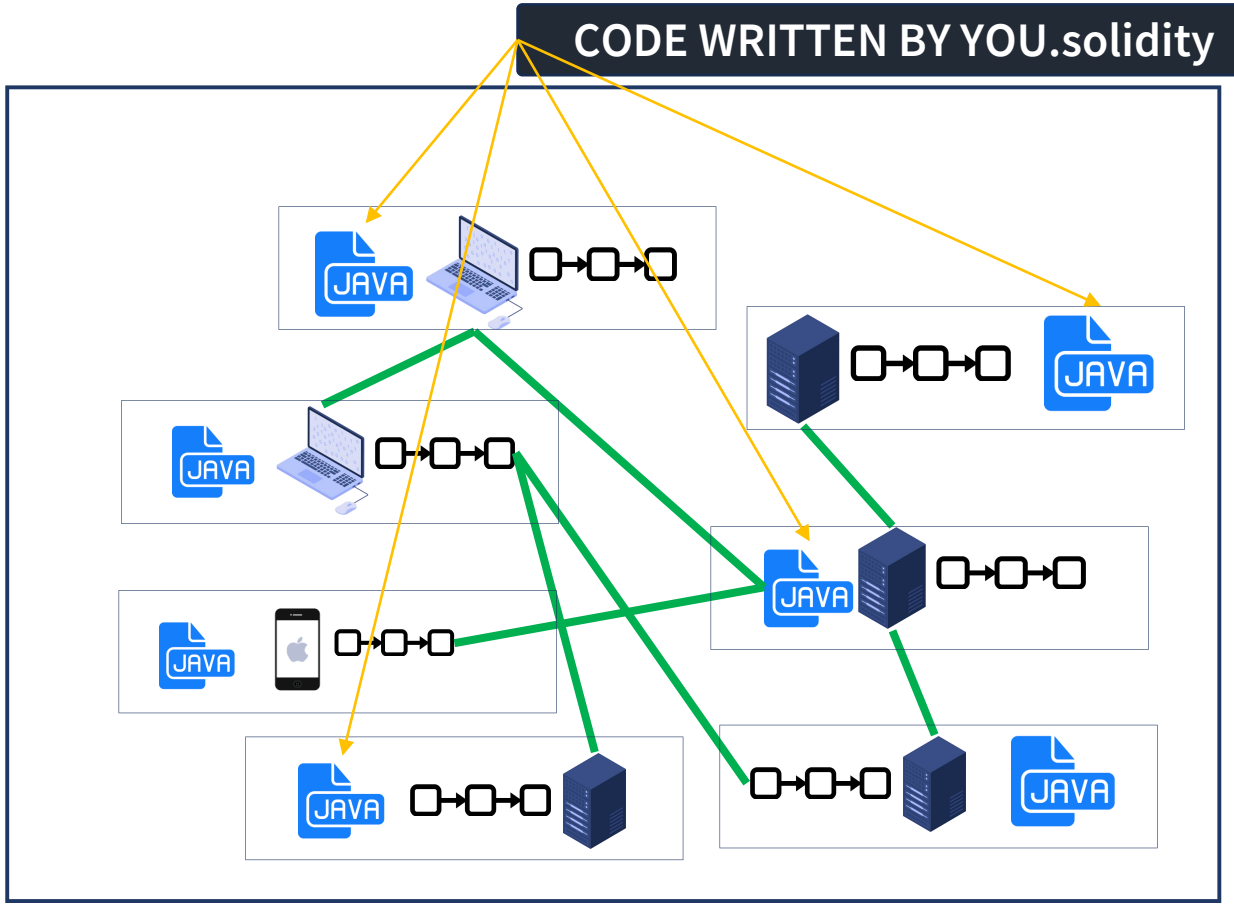
3. Buy Asset Response

How to deploy code on Ethereum Chains

Web2/Centralized Servers {Owned by an entity}. Code is deployed on finite servers owned by entity.

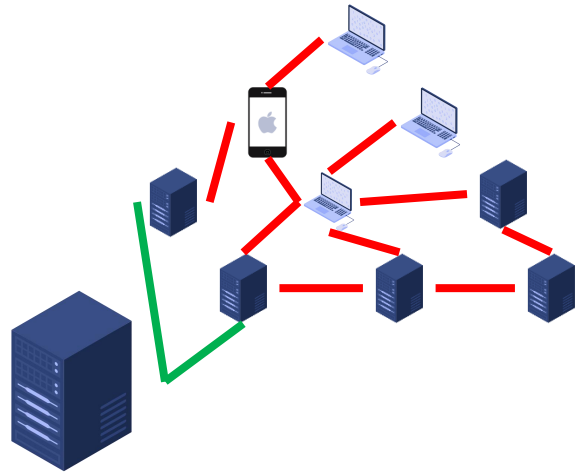


Web3/Decentralized "Owned by everyone"
code is deployed on thousands of servers owned by individuals or entities.
The deployed code is assigned a unique address uniform across all servers.



1

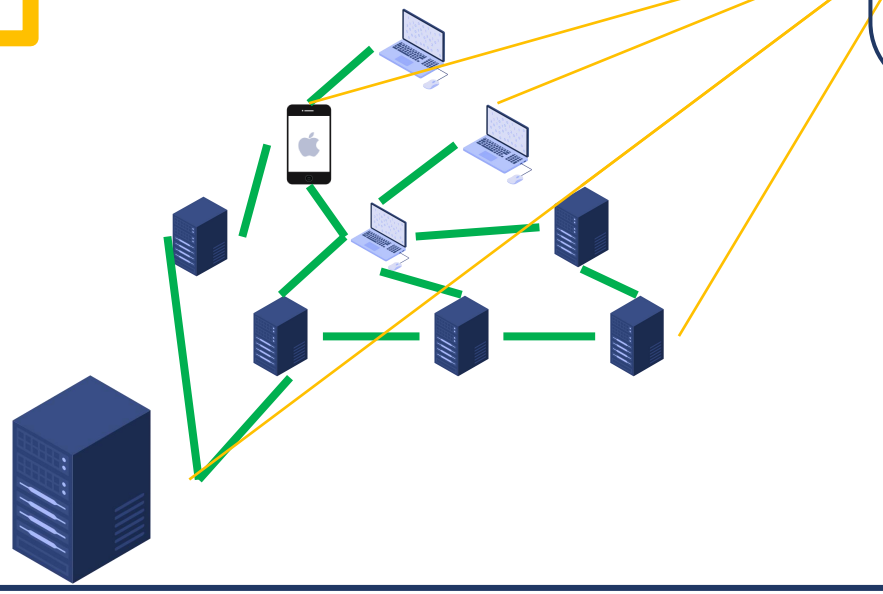
2



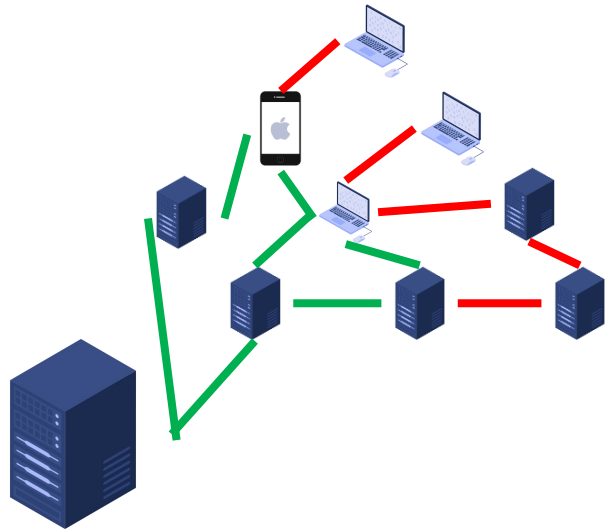
All nodes execute the same code

```
public String buyInGameAsset(String playerId, int quantity, int purchaseAmount) {

    if (purchaseAmount >=
            quantity * Inventory.getsellingPriceFor_PCGParachutes_()
            && Inventory.getTotalSupply_PCGParachutes_() >= quantity) {

        boolean paymentSuccess =
            PaymentService.transferMoneyToGame(purchaseAmount);
        /**
         * Will initiate transfer of money from player to game, using UPI, card payment
         */

        if (!paymentSuccess) {
            return "FAILURE";
        }
        Inventory.reduceTotalSupply(quantity);
        /** Reduce totalSupply of _PCGParachutes_ */

        Inventory.add_PCGParachutes_ForPlayer(playerId, quantity);
        /** Add "quantity" amount of _PCGParachutes_ to playerId */

        return "SUCCESS";
    } else {
        return "FAILURE";
    }
}
```

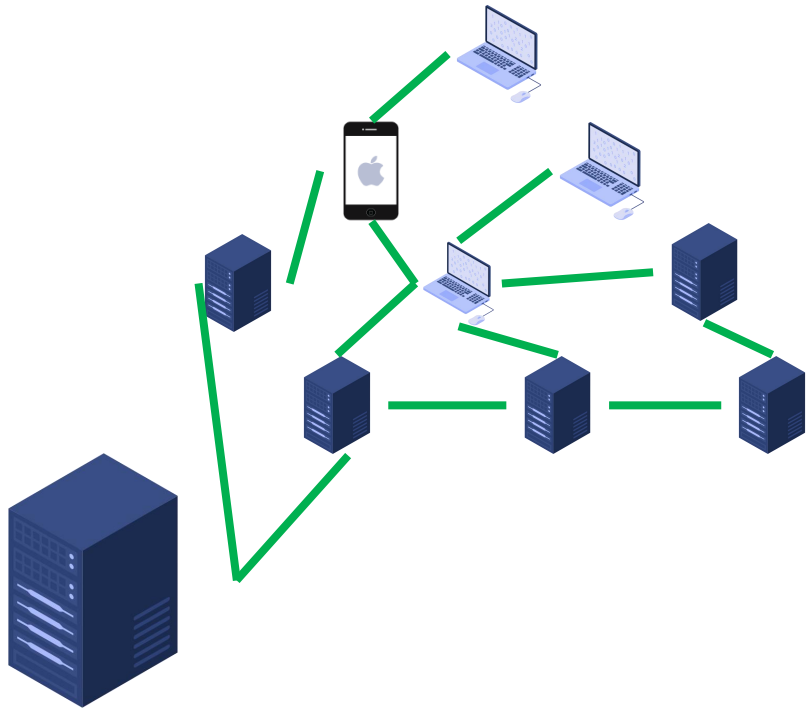
4



2

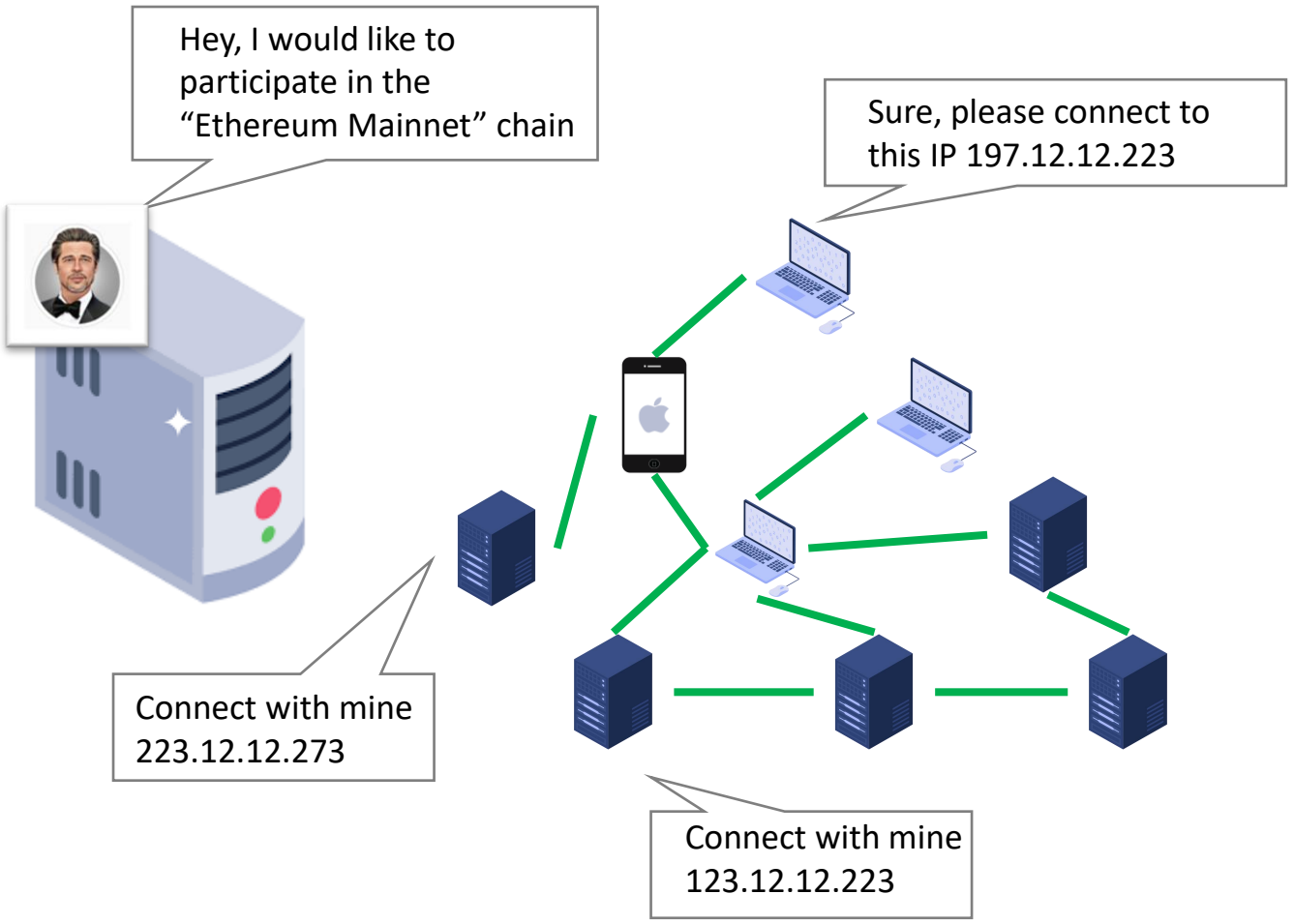


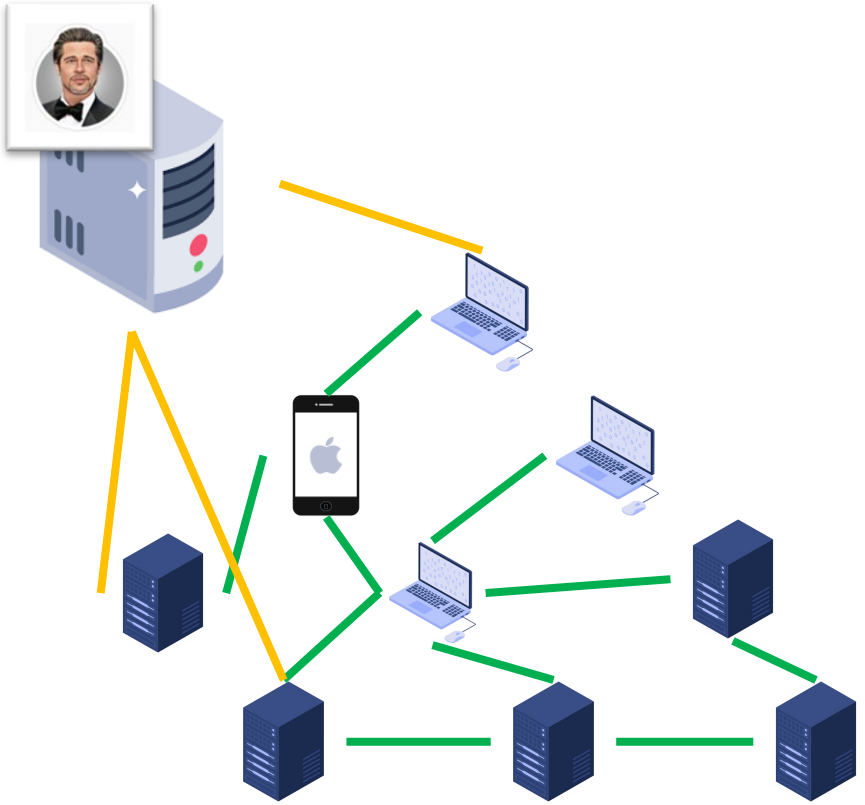
CONSENSYS



The transaction will complete only when all the initial states and final states of the code you have deployed.

How to become a part of the blockchain ?





GAS FEES AND MINING REWARD

1. What is the incentive for nodes/people to join the blockchain network ?
   Since running a server costs electricity and internet, there has to be some mechanism which compensates the nodes.
2. There are two stream of revenue for a node

GAS FEES

1. Whenever we make an Api call to the blockchain, what we are doing is essentially running a block of code in the server, in this case all the nodes on blockchain are running the same lines of code, this requires processing power (electricity/gas).
2. So, whenever we make an Api call to the blockchain we need to send some processing fee to the chain, the processing fee depends on two factors
   a. the complexity of code to be executed
   b. the traffic on the blockchain at the given point of time.

MINING REWARD:

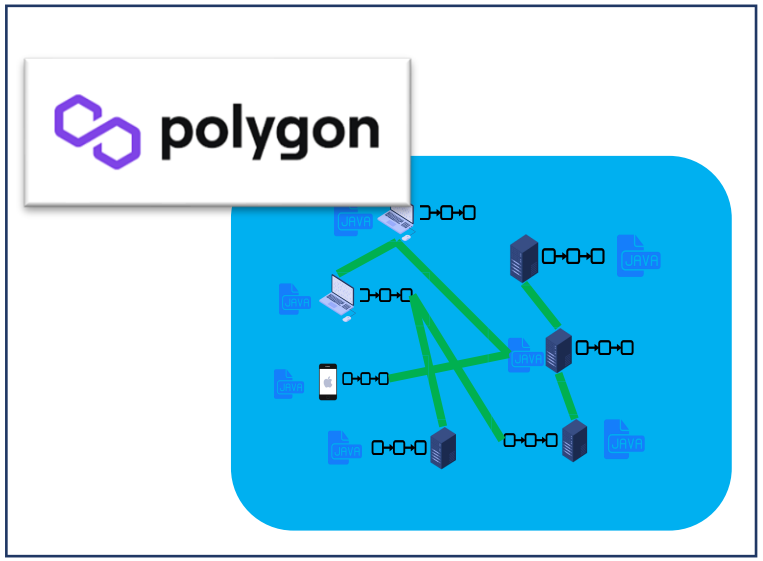
For some other day…..

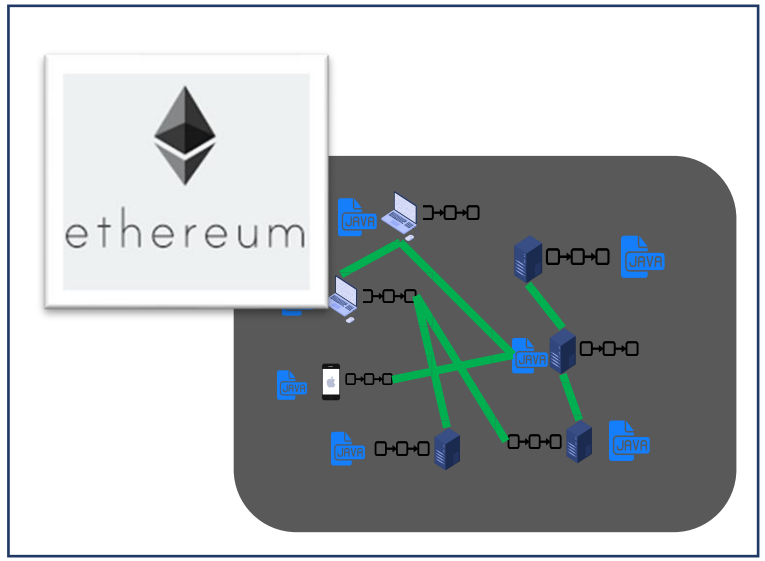
THE BIRTH OF ETHER CRYPTO CURRENCY

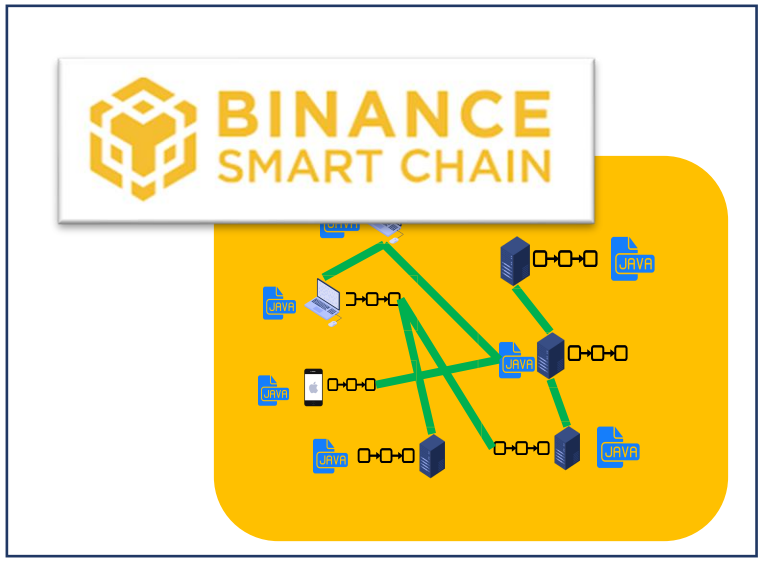
1. Lots of cryptocurrencies today are born out of the concepts as discussed previously
2. Here is the list of some of them you might own
    1. Matic (Polygon)
    2. Avalanche (AVAX)
    3. Tron (TRX)

Note that Bitcoin does not belon to this group nor do the concepts we are discussion applies to Bitcoin, it is simpler and more basic blockchain, read the story here

What are Chains ?







1. All the above chains can run the same Solidity code, with the same behavior.
2. Different chains have different currencies to pay gas fees and as MINING reward.
3. All these chains run the code on Ethereum Virtual Machine (Similar to Java Virtual Machine to run java code).
4. You Might own one of these currency in you Binance/CoinDCX exchanges. (Matic – Polygon, Ether – Ethereum Mainnet and Binance Smart Chain Girl – Binance Smart Chain)

Blockchain Economy

+ Mining reward (Ethers)

Exchange (Ethers) with (Dollars) and vice versa

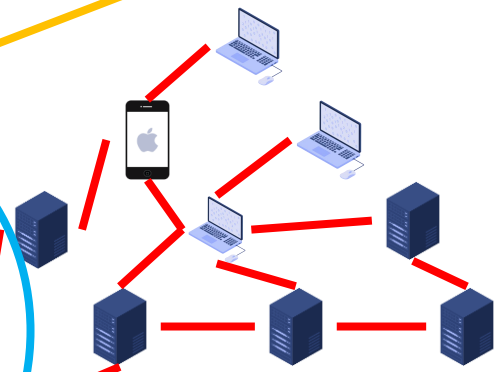
Exchange (Dollars) with (Ethers) and vice versa

BINANCE

CoinDCX
Digital Cryptocurrency Exchange

coinbase

0x0CBb3ab897C
03Cab897CBb6a
b897Cab897Cab
897Cab897Cb



+ Gas Fees(Ethers)

0x06775ABeFeA0A581CBb
3Cab8972d034690a0566b

Ethers as a Mode Of Payment

0x06775ABeFeA0A581CBb3C
ab8972d034690a0566b

Peer-to-peer
payment

0xsdsd4ABeFeA0A581CBb3C
ab8972d034690a05dsds

To buy in-
game assets

0x06775ABeFeA0A581CBb3C
ab8972d034690a0566b

PLAYERUNKNOWN'S
BATTLEGROUNDS

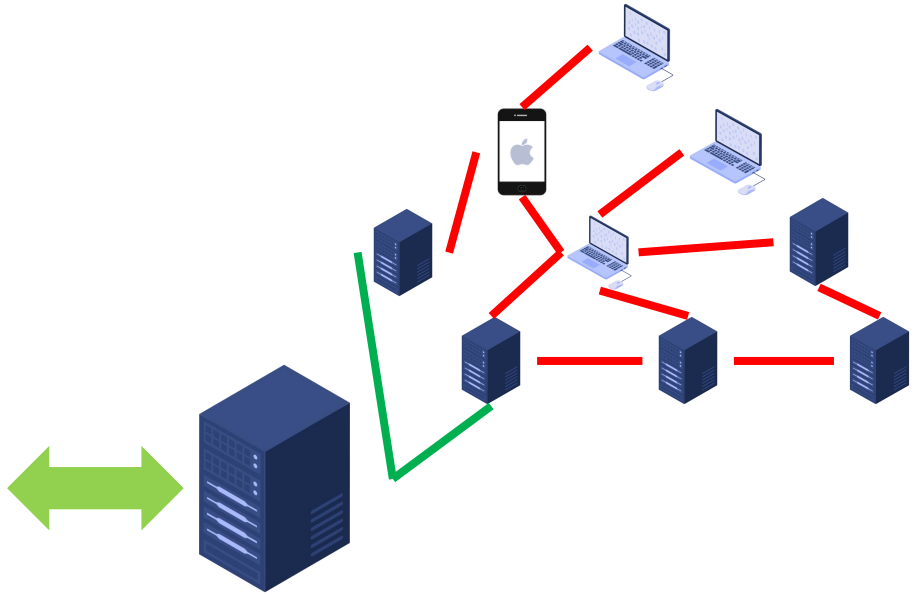
How Does the request to the Ethereum chain look like?

To : Address of the deployed code

THE MESSAGE IS SIGNED BY THE SENDER

Request Name

```
{
    "quantity" : "1"
}
```



The payload required to the request name

Gas fee : the request processing fee that needs to be paid to the nodes

The sender signs the complete payload, including the intent to transfer "purchaseAmount" and "Gas Fee," using his private key.

If any amount I want to send :
The “purchaseAmount ” to buy the item. (Funded by the account sending the request)

```
contract SimpleStorage is ERC1155, Ownable {

    uint256 private sellingPrice = 100;
    address private initialOwner = 0x1234567890123456789012345678901234567890;


     // Buyer can purchase a token by sending the required amount of Ether
    function purchase(uint256 assetId,uint256 quantity) external payable {
        require(msg.value >= sellingPrice*quantity, "Incorrect Ether amount sent");

        _mint(msg.sender, assetId, 1, "");
        /** Add "quantity" amount of tokens to "playerBlockchainAddress" */

        payable(initialOwner).transfer(msg.value);
        /** Transfer "purchaseAmount == msg.value" worth of
        "Ether" to the owner's Blockchain Address */

    }
    // Get the amount of tokens owned by an address
    @override
    function balanceOf(address account, uint256 id) external view returns (uint256) {
        return balanceOf(account, id);
    }
}
```

MOVING FUNDS (ETHERS) AND ASSETS IN THE SAME CODE

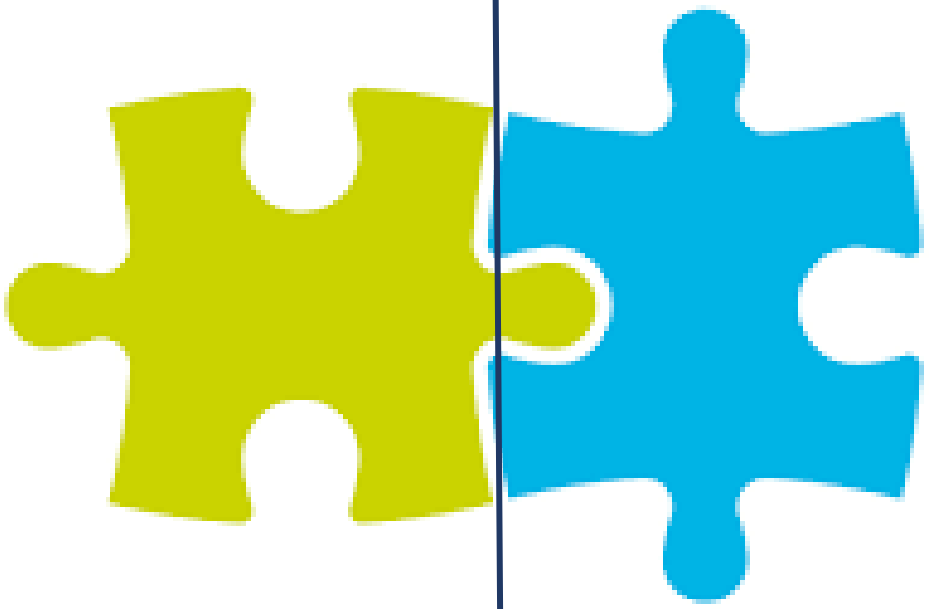
Moves the token to buyer's address ("msg.sender")

When the funds sent by the buyer gets transferred to owner address. {It's like payment method embedded in the Ethereum infrastructure}

NFT MARKETPLACES

NIKE NFT

Solidity Code to issue the tokens



OpenSea

[Code Deployment Address Link]

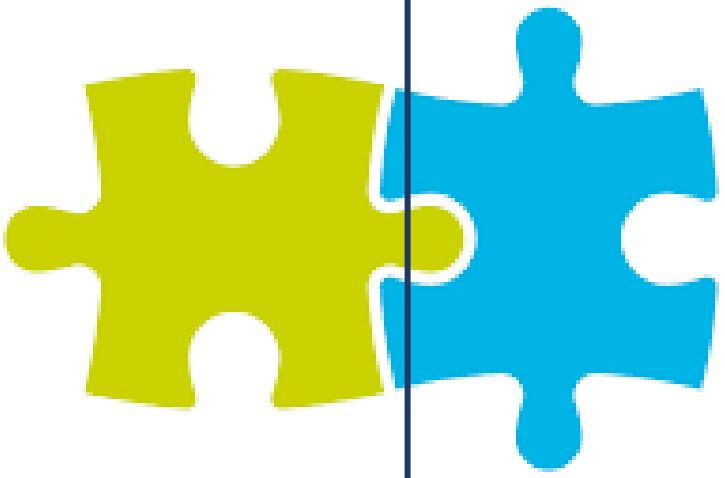
[Buy Nike NFTs Here]

NFT MARKETPLACES

Ok, so now the game has deployed its code that keeps track of ownership of different assets, and also sells it to the buyers

But still the costumers needs a User Interface and a trusted Marketplace to buy the tokens

Deployed
ECR1155
Compliant Code



OpenSea

RARIBLE
NFT Marketplace

mintable.app

SuperRare

Blockers to start a new Marketplace	How Ethereum Can Solve It?
Make a website, write a backend code.	Deploy smart-contract code, and make an interface to interact with the smart contract, {significantly less development effort}
Acquire infrastructure resources such as databases and servers, ensuring both their security and continuous uptime.	No infrastructure required **Blockchain Is the Infrastructure.**
Comply with rules and regulations of different countries.	No Country/Entity Owns it, no regulations can be put in place. Only the design/code is the regulator.
Handling different currencies , payment methods, and ensuring the security of financial information.	Making payments on chain is easy if you own the crypto-currency.
Depositing and withdrawing funds from the marketplaces.	The crypto-currency are the funds, 10 crore Indians own crypto-currencies, so getting crypto funds from crypto-exchanges should not be a blocker.

What are the Blockers For The Game.

The game must go through a lot of blockers:

1. Make a website, write a backend code.
2. Acquire infrastructure resources such as databases and servers, ensuring both their security and continuous uptime.
3. Comply with rules and regulations of different countries.
4. Handling different currencies , payment methods, and ensuring the security of financial information.
5. Depositing and withdrawing funds from the marketplaces.

https://www.youtube.com/watch?v=Q-0H7QU9hs8

https://www.youtube.com/watch?v=k_1Z3WEzLj4

NOT JUST ANOTHER PAYMENT OPTION:

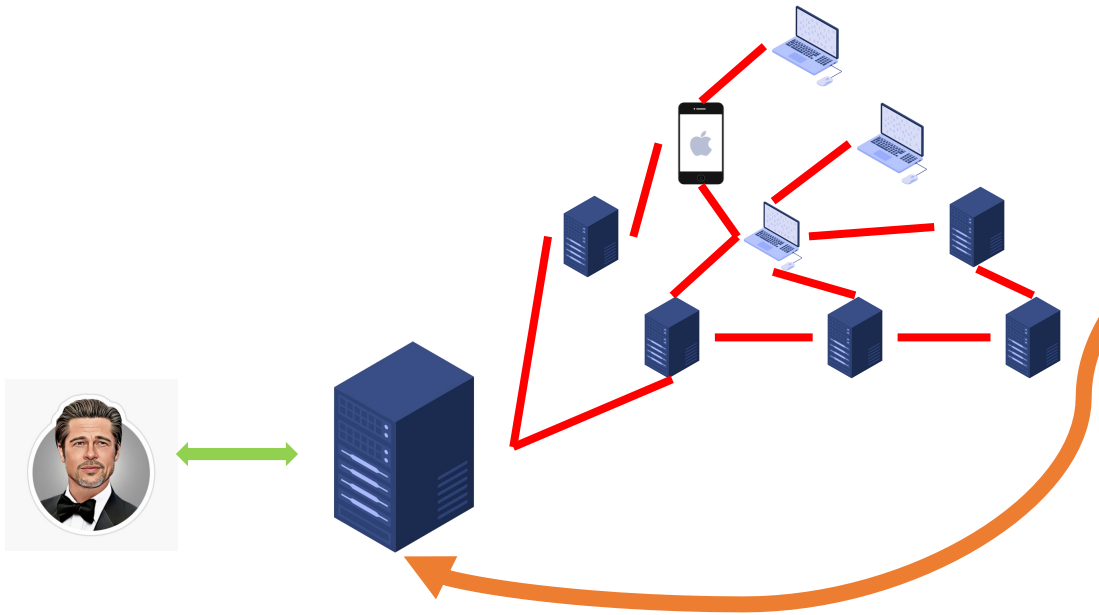
Ethereum Is not just a mode of payment, It is a network of server + payment method bundled together.
It is a network of servers that runs some code and accepts crypto currency as a mode of payment.

PROBLEM STATEMENT :

1. Consider this, the player has bought some Game Item tokens on Blockchain, now he wants to unlock and use these items inside game.
2. Similarly, the player has won/unlocked an asset inside the game and he wants a token corresponding to the assets on his blockchain address, so that he can trade it in future

This requires a Bridge that connects between the Blockchain World and the Game.

NODE SERVICE PROVIDER



Visit Alchemy

To make any transaction or API calls, to the blockchain, you need to be connected to a node, there are 100's and 1000's of nodes available to be connected to, but there are a few nodes used widely.
One of them is Alchemy

BLOCKCHAIN WORLD

Account 1

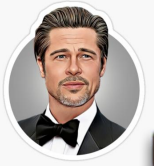
0x06775ABeFeA0A581CBb
3Cab8972d034690a0566b

Account 2

0x0CBb3ab897C03Cab897CBb6
ab897Cab897Cab897Cab897Cb

A player owns 3 units of “Sneakers”, 2 units of “Gun” , 1 unit of “Parachute” and “Kit” each on blockchain, on his two different addresses.

INSIDE GAME: PUBG

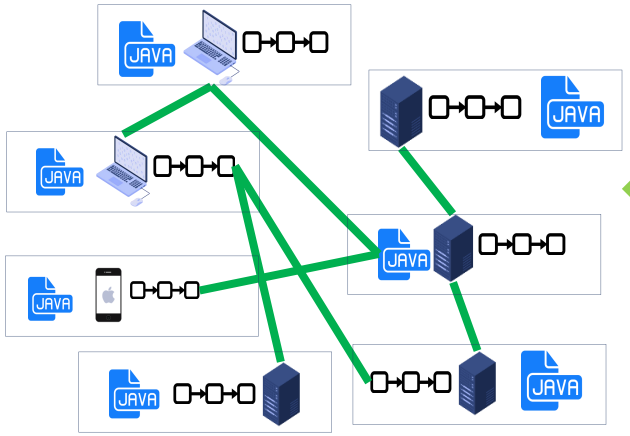


PLAYERUNKNOWN'S BATTLEGROUNDS

Player Id : Vaishnav012

The player plays “PUBG : Battlegrounds” as “Vaishnav012”.

MY PROJECT



PLAYERUNKNOWN'S BATTLEGROUNDS

Blockchain :

WEB3

The Bridge :

TokenMint

Game :

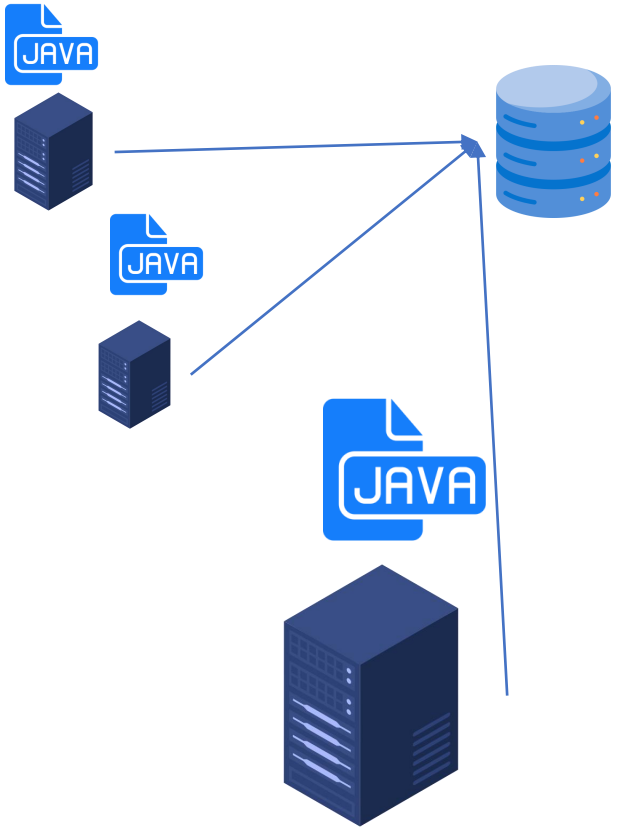
WEB2

```
class PCGParachutes {

    private Integer totalSupply = 9999;
    private Integer sellingPrice = 100;
    private HashMap<String, Integer> ownershipDetails = new HashMap<>();
    private String gameBlockchainAddress = "0x21xasa1ff12xgdfhay23273xv3hsdg3d7g3ydg38g3jwegdy2";

    public String buyInGameAsset(String buyerBlockChainAddress, int quantity, int purchaseAmount) {

        if (purchaseAmount >= quantity * sellingPrice && totalSupply >= quantity) {
            totalSupply = totalSupply - quantity;
            /** Reduce The Total supply of the tokens */

            ownershipDetails.put(buyerBlockChainAddress, quantity);
            /** Add "quantity" amout of tokens to "playerBlockchainAddress" */ // "
            transferMoney(gameBlockchainAddress, purchaseAmount);

            /**
             * Transfer "purchaseAmount" worth of "Ether" to the owner's Blockchain Address
             */

        } else {
            transferMoney(buyerBlockChainAddress, purchaseAmount);
            /** Return amount to "playerBlockChainAddress" */
        }
        return "Ok";
    }

    public Integer totalAssetsOwned(String blockchainAddress) {
        return ownershipDetails.get(blockchainAddress);
    }
```

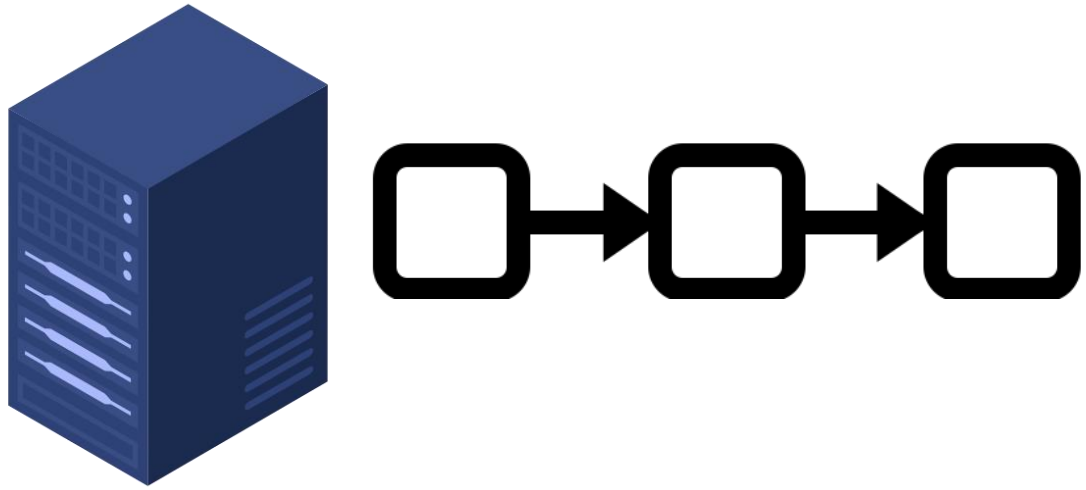
```java
public String buyInGameAsset(String playerId, int quantity, int purchaseAmount) {

    if (purchaseAmount >=
            quantity * Inventory.getsellingPriceFor_PCGParachutes_()
                && Inventory.getTotalSupply_PCGParachutes_() >= quantity) {

        boolean paymentSuccess =
            PaymentService.transferMoneyToGame(purchaseAmount);
        /**
         * Will Initiate transfer of money from player to game, using UPI, card payment
         */

        if (!paymentSuccess) {
            return "FALIURE";
        }
        Inventory.reduceTotalSupply(quantity);
        /** Reduce totalSupply of _PCGParachutes_ */

        Inventory.add_PCGParachutes_ForPlayer(playerId, quantity);
        /** Add "quantity" amount of _PCGParachutes_ to playerId */

        return "SUCCESS";
    } else {
        return "FALIURE";
    }
}
```

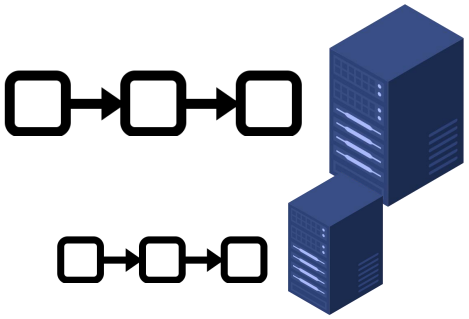
```java
public String buyInGameAsset(String playerId, int quantity, int purchaseAmount) {

    if (purchaseAmount >=
            quantity * Inventory.getsellingPriceFor_PCGParachutes_()
                && Inventory.getTotalSupply_PCGParachutes_() >= quantity) {

        boolean paymentSuccess =
            PaymentService.transferMoneyToGame(purchaseAmount);
        /**
         * Will Initiate transfer of money from player to game, using UPI, card payment
         */

        if (!paymentSuccess) {
            return "FALIURE";
        }
        Inventory.reduceTotalSupply(quantity);
        /** Reduce totalSupply of _PCGParachutes_ */

        Inventory.add_PCGParachutes_ForPlayer(playerId, quantity);
        /** Add "quantity" amount of _PCGParachutes_ to playerId */

        return "SUCCESS";
    } else {
        return "FALIURE";
    }
}
```

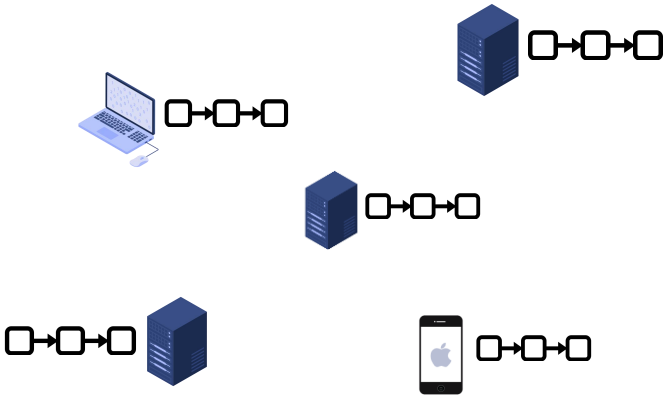

JAVA

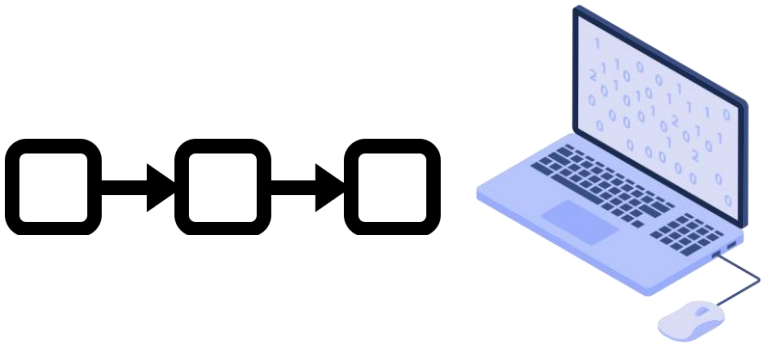

JAVA
JAVA
JAVA

https://www.youtube.com/watch?v=9N-GfdPm4Rc









https://www.theregister.com/2023/04/24/apple_antitrust_win_over_epic/

Various Marketplaces :
https://www.youtube.com/watch?v=s5_6pUHjW4c

"Project Hug,"

Read more at:
http://timesofindia.indiatimes.com/articleshow/105100600.cms?utm_source=contentofinterest&utm_medium=text&utm_campaign=cppst

https://www.ccl.nluo.ac.in/post/app-store-play-store-billing-policy-controversy-antitrust-violation-international-perspective

Various Marketplaces :
https://www.youtube.com/watch?v=s5_6pUHjW4c