



**IJIRCCCE**

e-ISSN: 2320-9801 | p-ISSN: 2320-9798



# INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

Volume 10, Issue 7, July 2022

**ISSN** INTERNATIONAL  
STANDARD  
SERIAL  
NUMBER  
INDIA

**Impact Factor: 8.165**

 9940 572 462

 6381 907 438

 [ijircce@gmail.com](mailto:ijircce@gmail.com)

 [www.ijircce.com](http://www.ijircce.com)

# DAPP MART- A Blockchain Based Ecommerce Website

Aayushman Kharel, Abhijeet Verma, Dhananjai Nirmal, Vivek Gupta, Dr. J Girija,

B.E Students, Department of Computer Science and Engineering, Bangalore Institute of technology, Bangalore, India

Professor and Head, Department of Computer Science and Engineering, Bangalore Institute of Technology, Bangalore,

India

**ABSTRACT:** Blockchain is a new technology that may be used to create Dapp for a variety of applications. The blockchain technology's transactions are transparent to all nodes in the decentralized network. For the sake of security, the majority of the apps were designed and attempted to be implemented on the Ethereum platform in blockchain. In the blockchain technology, the nodes in a distributed network exchange all of their data with the other nodes without modifying it.

The nodes' transactions are verified using digital signatures. Using the blockchain Ethereum platform, we created a decentralized web application for e-commerce in this paper. The front-end and back-end processes are outlined, and our application is deployed utilizing smart contracts written in the Solidity language. All this information regarding e-commerce transactions is maintained in blockchain blocks. For the nodes in the decentralized network, the web application for online purchasing will be safe and trustworthy. Other online retail sites, such as Amazon, are attempting to include blockchain for data security and consumer convenience. Storage, scalability, and other challenges can be explored for future development.

**KEYWORDS:** Blockchain ·Ethereum ·Ethers ·E-commerce ·Smart Contracts

## I. INTRODUCTION

The most important aspect is that the data is kept on a public ledger that keeps track of everything in a safe and transparent manner and ensures that no one can tamper with it. Blockchain is the name given to a distributed database that keeps all records of transactions between nodes in a network. It serves as the foundation for the Bitcoin digital currency.

Bitcoin is one of the most well-known blockchain instances. Satoshi Nakamoto authored a paper in 2008 titled "Bitcoin: a peer-to-peer electronic currency system," which popularized the idea. The transactions in the blockchain are recorded in immutable digital ledgers distributed among the network's nodes. Land assets, supply chain management systems, and other use-cases in blockchain are examples. Bitcoin relies on cryptographic proofs to trust nodes to carry out transactions on the blockchain. The digital signatures were used to verify each node's transaction. There is no central control or mechanism to maintain track of other people's data here. The machines that are connected to the dispersed network are referred to as nodes. The data of one node is spread across all other nodes in the globe via a distributed peer-to-peer network in the blockchain. It is tamper-proof and visible to all nodes that validate the data. Miners are the ones that verify and store the transactions in the blocks. A miner's primary responsibility is to solve the Crypto Puzzle to verify transactions between two nodes in a distributed network, with miners receiving a reward for each transaction verified.

Ethereum Dapps, like traditional centralized programs, have a back-end powered by smart contracts (the logic) and a front-end powered by web3 (the visual).

## II. ETHEREUM, A PLATFORM FOR BLOCKCHAIN TECHNOLOGY

The blockchain enables developers to create a variety of Dapp's on a variety of platforms. We'll use the Ethereum platform to create a decentralized application. One of the most essential platforms for executing arbitrary code on Ethereum projects. The distributed infrastructure may be able to finish many sorts of projects on Ethereum with the aid of smart contracts. Ethereum enables us to create decentralized and fault-tolerant apps by eliminating intermediaries and ensuring that all network nodes are transparent. Ethereum, which may be used as a cryptocurrency, can be used to produce a new tradable token. Tokens a standard API are used by wallets that are compatible with the Ethereum blockchain. To create a blockchain-based organization, the nodes must be members of the network and the limits must

be agreed upon. Only when all of the limitations are met will smart contracts execute the contracts automatically. Before diving into Ethereum, it's important to understand the basics. Now we'll study a little more about the Blockchain - Ethereum platform and how it works. The client, server, and Ethereum nodes are all discussed in the architectural diagram. Where the customer must have an ether wallet with specific amounts of ethers in order to sell or acquire the required items. The front end is made up of JavaScript files that communicate with the network's other nodes. Next.js and React.js are the two frameworks. The truffle server hosts the truffle process, which serves as a smart contract development framework. The Ethereum nodes that make up the main network are being used as a public network for contract development. Ganache is used to store ether.

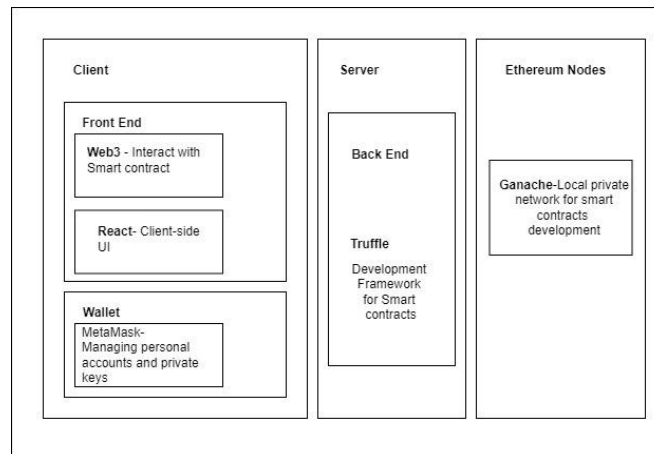


Figure 1 : Architecture

**2.1 Ethereum Virtual Machine (EVM):** Ethereum virtual machines are given by the Ethereum platform, which executes instructions utilizing the distributed network's nodes. The machine manages all of the node's information, including its location, balance, current gas price, and block information. It is written in a variety of computer languages, including C++, Java, JavaScript, Python, Solidity, Node.js [6], and others. Anyone on the network may use this to run code that is guaranteed and predictable.

**2.2. Gas:** Miners on the Ethereum blockchain process smart contracts, which results in the introduction of a new block to the Ethereum blockchain.

The miners are rewarded for their work through smart contracts on the Ethereum virtual computer, which demand certain payments known as GAS.

The quantity of gas that the miners would need was calculated in their smart contracts.

If the smart contract becomes more intricate, the gas fee will rise as well.

**2.3. Solidity language:** Ethereum uses the Solidity language to create smart contracts [10].

On top of the Ethereum virtual computer, a programming language has been built.

Solidity's major goal is to store all of the states of the tokens that are sent and received by the nodes. Programmers who are familiar with JavaScript, C++, and Python will have little trouble understanding the Solidity language.

If you're familiar with the Solidity programming language, writing a program for an Ethereum smart contract should be simple.

**2.4. Algorithm:** The newly formed blocks in the blockchain are appended to the preceding block.

It is included in the work proofing process.

The hash value of each block is created during the addition process, and that hash value becomes the value for the following block. For instance, the hash value of block 3 will be used in block 4, and the hash value of block 4 will be used in block 5, and so on. If one of the nodes tries to edit or modify any of the data in the block, the hash value for that node will be updated, and the data will be updated as well.

It couldn't be changed (ie. Data Immutability).

**Proof of Work (POW):** If we wish to add a new block or edit an existing block, we must validate it in the public blockchain; otherwise, we will be unable to add or modify data in the blockchain network.

When we execute any processing to add a block to the blockchain, we must use the POW concept [21], which states that it will take at least 10 minutes for any node to add the block to the blockchain network.

Modifying data on the blockchain network will take longer.

After the block has been validated, it is stored in several devices known as nodes, which are also known as distributed ledger databases [DLD].

"Data from the blockchain will be kept on numerous devices" [23]. It is not the main server; each node or machine in the blockchain network will have a single copy of the blockchain's data.

To add a block, all nodes must reach a consensus. To do so, we have the notion of a consensus method for adding data to a blockchain, as well as a variety of consensus algorithms.

### Working mechanism of the POW consensus algorithm

- 50 percent of the nodes in the blockchain network
- require a supercomputer (i.e., computational power)
- difficulty in solving the mathematical crypto challenge

**2.5. Smart contracts:** Smart contracts are agreements that require us to obey certain rules and constraints when doing transactions. Our distributed application [3] initializes the Smart Contract [10], which serves as back-end storage and logic. (as seen in Fig. 2)

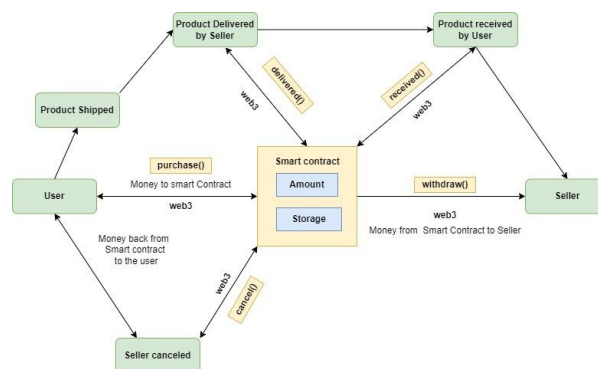


Figure 2: Interface diagram

### III. COMPILATION AND MIGRATION ON THE BACKEND

#### A. Truffle Compilation

The contracts are written and performed in the Solidity programming language, with a.sol file extension. Now we will use the truffle compile command to compile a truffle file. The Migrations. Sol file then aids in the deployment process. If we utilize Truffle Box, we will end up with several files

#### B. Truffle Migration to smart contract

The truffle has been successfully compiled in our smart contract. Now we'll look at migrating truffle to the Ethereum blockchain network. The Migrations files are used to install the smart contracts in Ethereum. The task deployment and assumptions throughout time are the responsibilities of doing this file. The new migration scripts on the blockchain network are being developed as part of our project. The command to perform the migration is truffle migrate, as we already compiled the truffle.

The data may be moved or replaced by fresh migrations.

#### C. Ganache

We utilized Ganache to deploy the contracts, construct the apps, and run the tests on the Ethereum blockchain in this study. We've previously downloaded Ganache [12] (which stores the default addresses for each ether in ten separate accounts) as well as the necessary files. It will create a local blockchain that will be accessible at <http://127.0.0.1:7545> (see Fig. 4). The block's condition in Ganache has been modified. The block number is altered after the calculation.

It has been updated to Block 4 on our end (see Fig. 5). The original account will be compensated with 100 ethers at first. Because of the transaction costs for migrations, it is now becoming less.



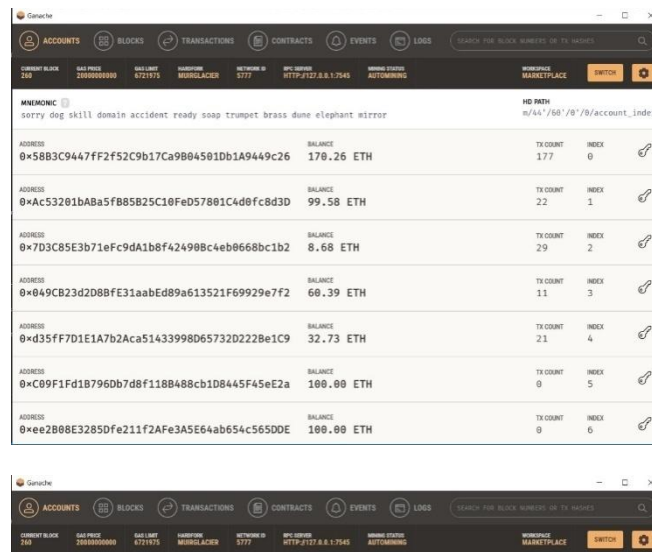


Figure 3: Ganache

#### D. Metamask ( Version 4.4 )

Metamask is the most straightforward way to communicate with a browser-based DApp. Both the chrome extension and firefox may be used to install it.

Create a user account and acquire the 12 seed phrase when executing the basic steps once MetaMask has been installed.

1. Select "New Network" from the drop-down menu and type in <https://127.0.0.1:7545>.
2. Each account in Ganache is granted 100 ethers by default.

Check that the amount of ether is lowered when we deploy our contract and run the test, because we utilized it.

We must now use Ganache to link Metamask to the blockchain. By selecting Custom RPC from the menu, we can see that the network is connected to the "Main Network"

### IV. BLOCKCHAIN BASED DECENTRALIZED APPLICATIONS

We designed an E-commerce application for users in the peer-to-peer network utilizing the Ethereum platform in this part. For the smart contract deployment, we employed a handful of the functions.

Here's what it is:

- The Ethereum smart contract is written in JavaScript using the Solidity programming language.
- Using Web3.JS and React.JS, the smart contract is published on a blockchain

The smart contracts are deployed using the npm instructions, and a few of the processes to take are outlined and discussed below:

#### 4.1 Installing the dependencies

To run our smart contract, we'll need to install a few dependencies, which are detailed in the sections below.

##### 4.1.1 Ganache

Ganache is a private blockchain that may be used to simulate the behavior of a public blockchain. It enables us to deploy the smart contract, build the apps, and test the networks.

4.1.2 Node.JS — We'll need the node package manager, or npm, which comes with node.js [6] to deploy the smart contract. The required files for our project will be included in the migration file, source file, and test file (truffle config.js, package lock.json, package.json) by running the npm instructions.

4.1.3 Truffle Framework — it's a set of tools for creating Ethereum smart contracts with the Solidity programming language.

4.1.4 Metamask Ethereum wallet is a browser plugin that connects users to the Ethereum blockchain. Our web browser becomes a blockchain browser as a result.

4.1.5 Web3.js — Web3.js allows us to communicate with local or remote Ethereum nodes via HTTP or IPC connections. It can also access the user's account, as well as all transaction information

## V. FRONTEND OF OUR APPLICATION

After a full node to connect with Ethereum main network, we want to pay some gas fee or transaction fee. Whenever we do transaction miner will get re-wards to mine the block, before doing any transaction on the Ethereum network, we need ether (ETH). We have to buy the ethers. In this case, we are working on testing network called rinkeby network. We have 18.75 ETH in our account to deploy transaction in rinkeby network and also, we allow to do transaction on Ethereum main network with 100 ETH. We have in Ethereum software test networks are [Ropsten, Kovan, Rinkeby]. We have the browser, which works under the centralized system to make our browser to dapp for that we already inject the metamask extension to our browser [get chrome ADDON], we built the app and deploy it with metamask after we have connected to main network our account address linked with truffle ganache, metamask which provides unique address. We can view the account details in metamask like QR code of address, we can also view etherscan.io which is used to track all our activity with Ethereum blockchain

1. When we are trying to interact with the distributed application, the popup message from the metamask will ask the confirmation request that is connected to our metamask wallet. If not, we won't be able to connect with the DApp.
2. After confirmation, click on the Adopt button to select the pets.
3. Then, we will be directed to the metamask account to approve the transaction
4. We will now see the success button after purchasing (see After successfully submitted in metamask, in MetaMask we'll see the transaction list

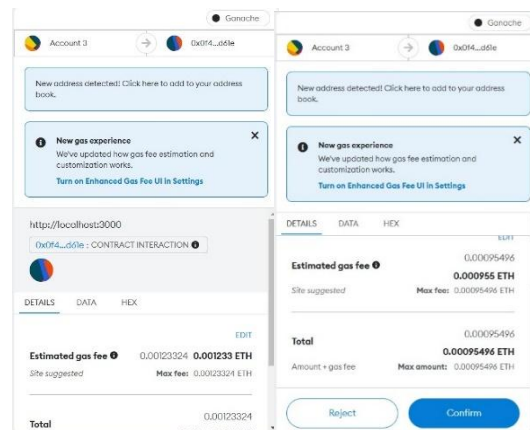


Figure 4: Metamask wallet

## VI. CONCLUSION AND FUTURE WORK

Our major objective in this article is to create a Blockchain-based decentralised application in our local host by combining the Web3 framework with ganache and metamask. Its purpose is to demonstrate how online purchasing will be secured and kept in the next years in light of customer privacy concerns. To understand the differences between each network in blockchain technology, additional networks like as rinkeby, ropsten, and others can be used.

The Ethereum main network and the local host <http://127.0.0.1:7544> RPC are used to interact with our smart contract. As a result, we believe that the dapp might be implemented with alternative test networks for efficient node operation, and the performance of the nodes with other test networks can be investigated in future study.

In addition, quantum computing [22] is now being researched under blockchain for high processing power and to easily mine the mathematical crypto problem in future development.

Supply chain can be handled by maintaining a smart contract at each point of shipments for ex shipment is from point A to point F then smart contract can be maintained to verify the product at point B, C, D and E.

Some more technologies can be involved to verify the physical product is the same that the user has ordered, some

processing of physical image can be done by taking a large amount of snaps of the physical product and go for digital image processing for each image and verify the product.

#### REFERENCES

1. E-Commerce Security Issues Randy C. Marchany & Joseph G. Tront 2002 IEEE
2. TRANSPARENCY STRATEGY IN INTERNET-BASED SELLING Liying Zhou, Weiquan Wang, Jingjun (David) Xu, Tao Liu, Jibao Gu 2018 INFMAN
3. A Comparison between Traditional SCM and e-SCM Fu-Hung Chiu 2002IEEE
4. e-Commerce, Business Methods and Evaluation of Payment Methods in Nigeria Michael Adeyeye 2008 Electronic Journal Information Systems Evaluation.
5. Scalability of Web-based electronic commerce systems Gregor v. Bochmann & Johnny W. Wong 2003 IEEE
6. Performance Benchmarking and Optimizing Hyperledger Fabric Blockchain Platform Parth Thakkar, Senthil Nathan N & Balaji Viswanathan 2018 IEEE
7. A Survey on Fraud Detection Techniques in Ecommerce Priya J Rana & Jwalant Baria 2015 International Journal of Computer Applications
8. Blockchain as E-Commerce Platform in Indonesia by L Ismanto et al 2019 J. Phys.: Conf. Ser
9. Blockchain Technology for E-commerce Industry by Hemantkumar P Bulsara & Pratiksinh S
10. On Blockchain Applications: Hyperledger Fabric And Ethereum by Sajana P & Sindhu M.
11. Centralization vs. Decentralization of Application Software by Robert St Louis
12. Data Security Sharing and Storage Based on a Consortium Blockchain in a Vehicular Ad-hoc Network Xiaohongzhang and xiaofengchen
13. The Supply Chain Has No Clothes: Technology Adoption of Blockchain for Supply Chain Transparency by Kristoffer Francisco & David Swanson
14. Blockchain-Enabled Smart Contracts: Architecture, Applications, and Future Trends Shuai Wang , Liwei Ouyang, Yong Yuan , Senior Member, IEEE, Xiaochun Ni, Xuan Han, and Fei-Yue Wang , Fellow, IEEE
15. How the Blockchain Revolution Will Reshape the Consumer Electronics Industry By Jong-Hyouk Lee and Marc Pilkington





INNO  SPACE  
SJIF Scientific Journal Impact Factor

Impact Factor: 8.165

 **doi**<sup>®</sup>  
**cross** **ref**

**ISSN** INTERNATIONAL  
STANDARD  
SERIAL  
NUMBER  
INDIA



# INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

 9940 572 462  6381 907 438  [ijircce@gmail.com](mailto:ijircce@gmail.com)



[www.ijircce.com](http://www.ijircce.com)

Scan to save the contact details