



# Using Ethereum Blockchain Technology to Deploy Using Remix IDE

Author : Chaiwitchit Konfoo

Advisor: Assistant Professor Dr. Ratsameetip Wita

## Abstract

Decentralized Applications (DApps) are software applications that operate on blockchain networks, eliminating the need for centralized servers. The Ethereum Blockchain is a prominent platform for DApp development through Smart Contracts written in Solidity. However, the deployment and management of Smart Contracts on Ethereum remain complex. This research explores the process of deploying a DApp on Ethereum using Remix, a platform that streamlines Smart Contract creation, testing, and deployment. The findings will enhance developers' understanding of DApp development, reduce costs, and improve system management efficiency through Smart Contracts. Furthermore, this research contributes to the study of Blockchain, Smart Contracts, and Web3 technologies.

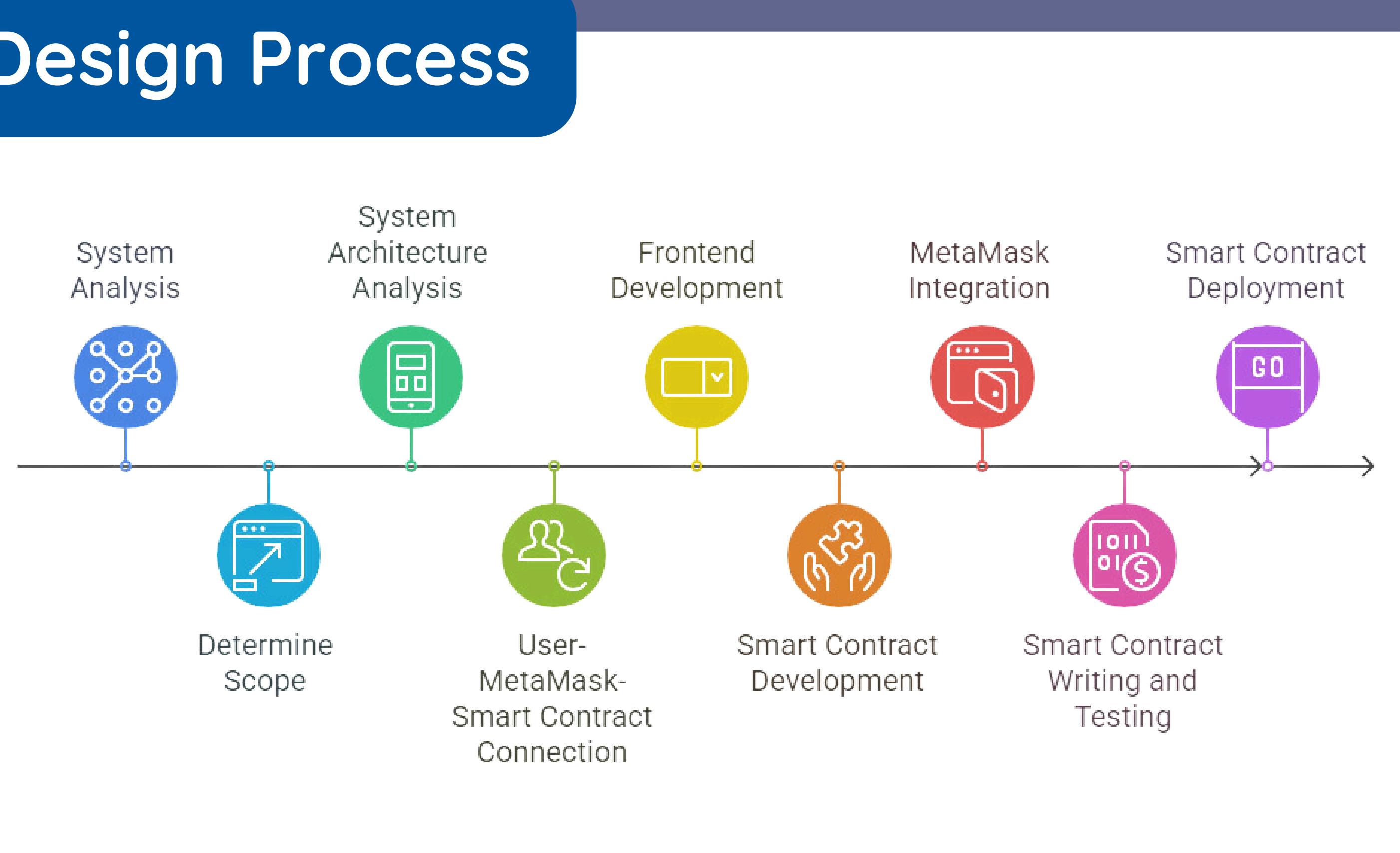
## Introduction

A Decentralized Application (DApp) operates on a blockchain network, distinguishing it from traditional applications that rely on centralized servers. DApps function through Smart Contracts, self-executing programs that run on the blockchain upon meeting predefined conditions.

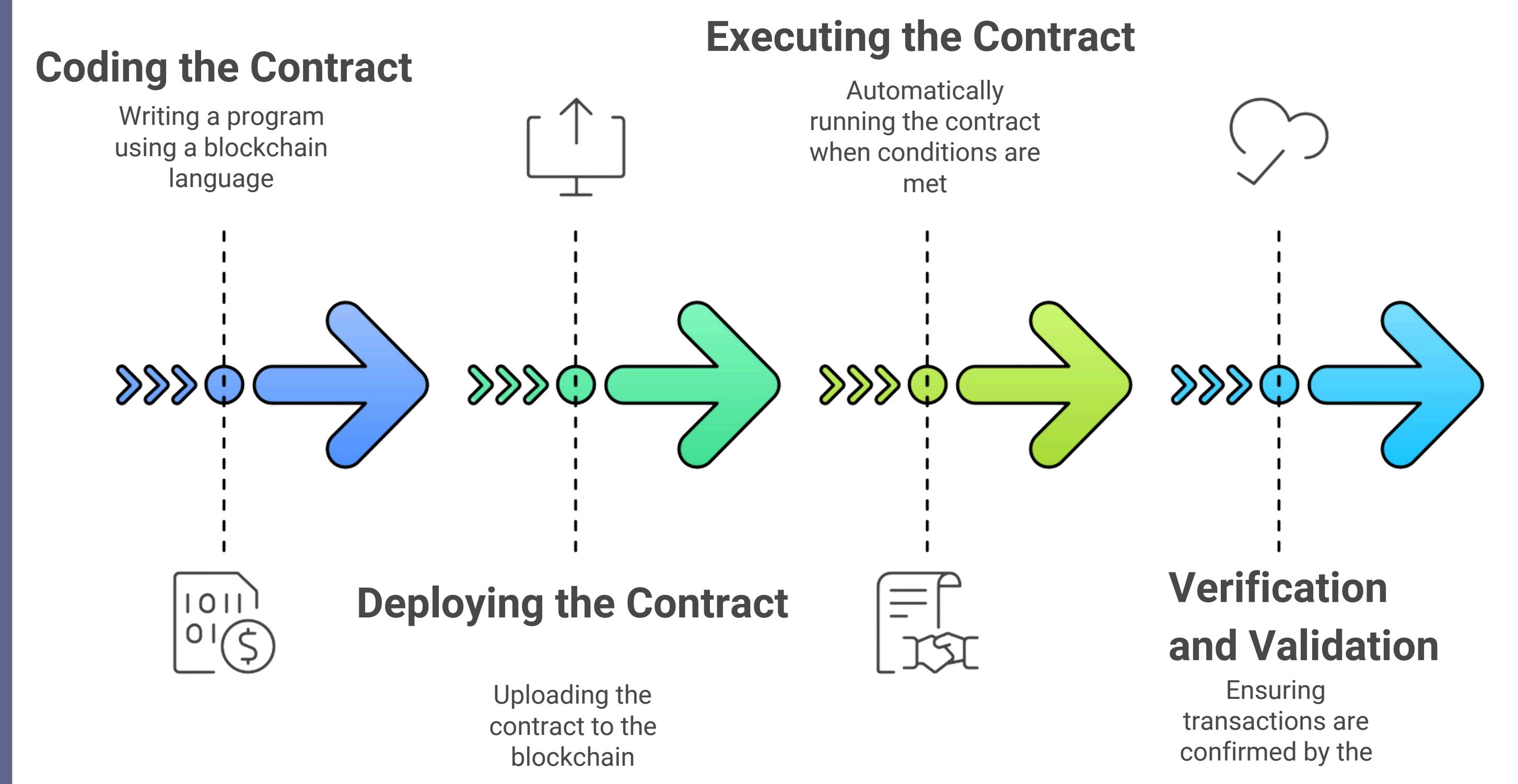
Ethereum is a leading platform for Smart Contract and DApp development, utilizing Solidity as the primary programming language and Remix IDE for testing and deployment. Remix IDE allows developers to efficiently simulate and optimize DApp functionality.

Despite Ethereum's growing popularity, the deployment and management of Smart Contracts remain complex. This research aims to explore the DApp development process on the Ethereum Blockchain and provide a comprehensive guide for deploying DApps using Remix IDE, facilitating understanding and implementation for developers and enthusiasts.

## Design Process



## Smart Contract Process



## Results

**Smart Contract Testing** After writing and testing a Smart Contract in Remix IDE, the results confirm that the contract functions correctly. Testing can be done using JavaScript VM, Injected Web3 (MetaMask), or Web3 Provider to verify the functionality of various contract functions.

**Deploying a Smart Contract on Ethereum Testnet/Mainnet** Upon deploying a Smart Contract to the Ethereum network (e.g., Rinkeby, Goerli for Testnet or Ethereum Mainnet) via Remix IDE, the deployment is successful, and a Transaction Hash is generated, which can be verified on Etherscan.

**Connecting to the Frontend (Web Interface)** By connecting the deployed Smart Contract to a DApp using Web3.js or Ethers.js, users can interact with the contract via the web interface, and data will be accurately saved and updated on the Blockchain.

## Reference

- [1] Remix IDE , <https://remix.ethereum.org/>
- [2] Metamask, <https://metamask.io/>
- [3] DApp, Techsauce,<https://techsauce.co/tech-and-biz/what-is-dapp-decentralized-applications-to-use>

## Technology



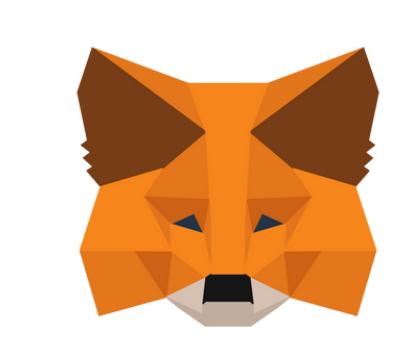
Ethereum Blockchain



Remix IDE



Solidity



Metamask

## Conclusion

Ethereum Blockchain and Remix IDE offer efficient development and deployment of DApps, enabling rapid Smart Contract creation and testing, particularly on the free Testnet. Integration with the frontend via Web3.js or Ethers.js facilitates seamless user interaction with Smart Contracts. However, Remix IDE has limitations in managing large projects and Mainnet testing, which may require additional time and resources. Future research could explore tools such as Truffle or Hardhat, which support the development of more complex DApps and facilitate more efficient Mainnet deployment.