



# BLOCKCHAIN Smart Contracts & DApps

Rishi Cherukuri

## Welcome to the Journey of Deep Dive on BlockChain

- Solidity basics for writing Smart Contracts
- The Lottery DApp overall architecture
- Writing Smart Contracts for the Lottery App
- Understanding about Web3 JS and MetaMask better
- Understanding Faucets to fund fake Ethers on Rinkeby network
- Using require to notate minimum payment associated
- Using payable methods to accept Ether payments
- Configuring Infura for the application
- Deploy to Infura based Rinkeby network
- Building the User Interface that can interact with the Solidity Smart Contract
- DEMO & Hands-on

<https://www.stateofthedapps.com/dapps/index>

Pre Requisites:

<https://code.visualstudio.com/>

<https://www.npmjs.com/package/node-windows>

<http://truffleframework.com/ganache/>

<http://truffleframework.com/tutorials/how-to-install-truffle-and-testrpc-on-windows-for-blockchain-development>

<https://github.com/ethereum/go-ethereum/wiki/Installing-Geth>

Signup Infura network - <https://infura.io/signup>

Fund Rinkeby Wallet using Faucet - <https://faucet.rinkeby.io/>

## Recap / Quiz

- What is a Smart Contract?
- What is MetaMask and why to use it?
- What are Solidity and Web3 libraries?
- Where to get the ABI and Bytecode?
- Whats the difference between Ganache & TestRPC?
- What are the various TestNets available for Ethereum?

<https://www.coindesk.com/information/what-is-a-decentralized-application-dapp/>  
<https://medium.com/@robertvermeulen/learn-solidity-the-ethereum-smart-contract-programming-language-7f106fc26d6>  
<https://www.tutorialspoint.com/reactjs/index.htm>  
<https://webapplog.com/tdd/>

## Inbox Smart Contract(Recap)

```
pragma solidity ^0.4.23;

contract Inbox {
    string public message;

    function Inbox(string initialMessage) public {
        message = initialMessage;
    }

    function setMessage(string newMessage) public {
        message = newMessage;
    }

    function getMessage() public view returns (string){
        return message;
    }
}
```

<https://remix.ethereum.org/>

## Ethereum Terminology (Recap)

Language / Stack	Purpose
Ether	Crypto currency used for paying for smart contracts to run
Solidity (sol)	Smart Contract programming language
Geth, eth, pyethapp	The main Ethereum software written in different languages
Ethereum Virtual Machine	Decentralized computation
Swarm & Whisper	File Storage and communication protocols
Truffle	Smart Contract Framework offering templates for building smart contracts
TestRPC/Ganache	Local Ethereum Blockchain network, where we can use fake Ethers
Kovan, Rinkeby, Ropsten	Test Ethereum Blockchain network, where we can use fake Ethers
Web3 JS	Ethereum JavaScript API

Solidity Docs - <https://solidity.readthedocs.io/>

Prerequisite Setup Instructions:

<https://code.visualstudio.com/>

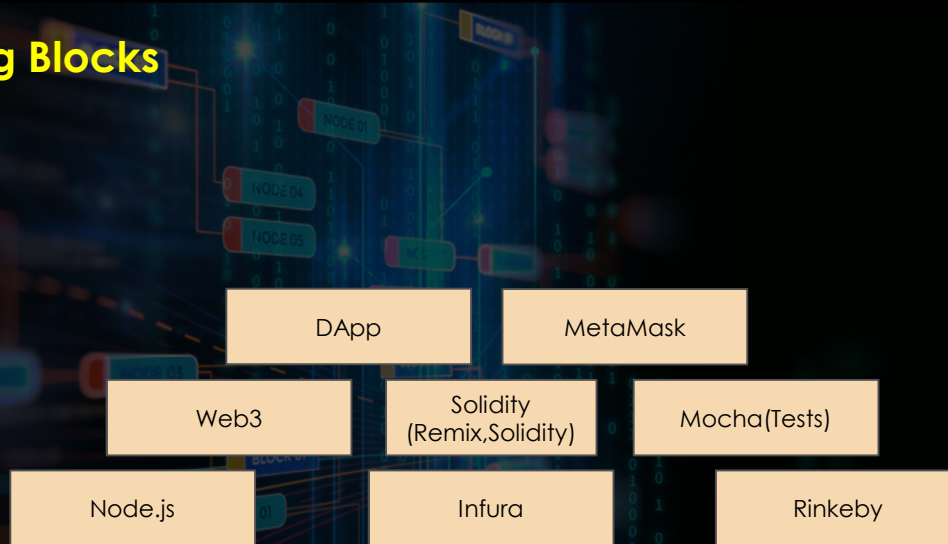
<https://www.npmjs.com/package/node-windows>

<http://truffleframework.com/ganache/>

<http://truffleframework.com/tutorials/how-to-install-truffle-and-testrpc-on-windows-for-blockchain-development>

<https://github.com/ethereum/go-ethereum/wiki/Installing-Geth>

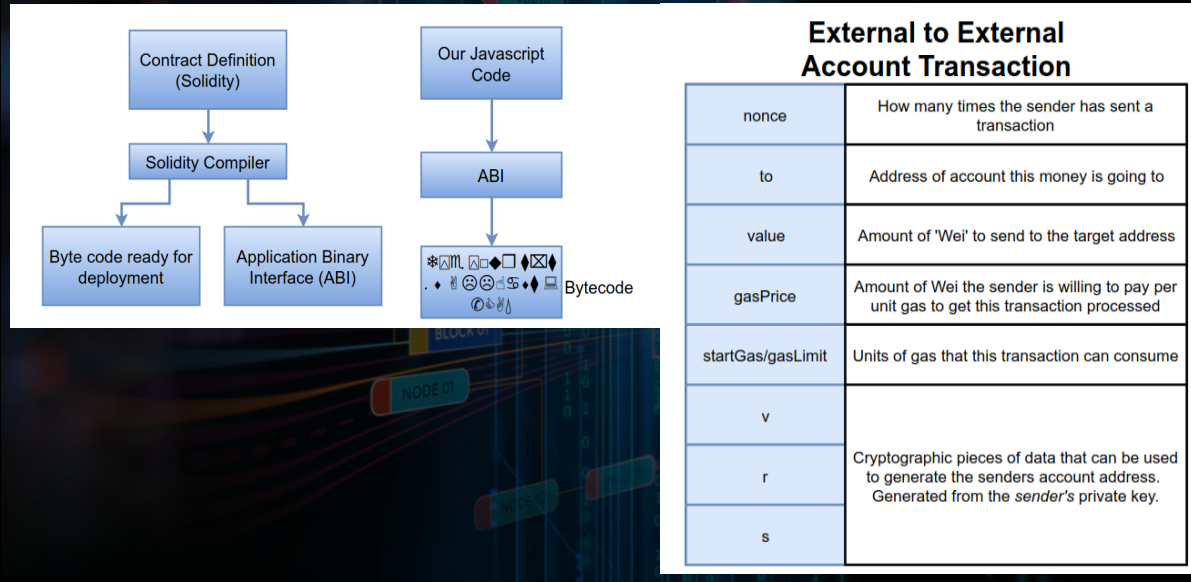
## Building Blocks



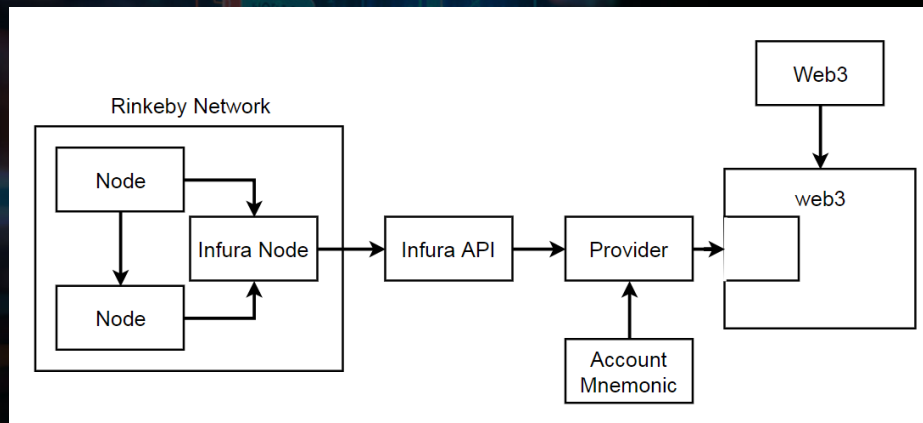
```
npm install --save solc mocha ganache-cli web3@1.0.0-beta.26
```

Node.JS  
Web3  
Rinkeby  
Infura  
VS Code  
Solidity Basics

## Some Block Diagrams to help you understand



## How Rinkeby , Infura and your local providers come together





## Advanced Smart Contracts?

### Hands-on

- Programming a Lottery Smart Contract with Remix IDE

<https://remix.ethereum.org/>

Some more examples

<https://github.com/BlockChain4All/EthereumCasts>



**BLOCKCHAIN**

**Thank you**

**Nagarro, CoderPlex & BlockChain4All community**

Questions?