

# Ethereum Development Fundamentals



**101 Blockchains**

# THIS COURSE IS FOR YOU IF YOU WANT TO:



Understand what Ethereum is all about.



Understand how to get started developing with Ethereum



Improve your enterprise blockchain professional skills.



**101 Blockchains**



**FUNDAMENTALS**



**2 HOURS**



**FLEXIBLE LEARNING**

**Ethereum**

# Ethereum Development Fundamentals

- **Module 1: Blockchain Fundamentals**
- **Module 2: Ethereum Fundamentals**
- **Module 3: Development Fundamentals**
- **Module 4: Ethereum Development Tools**
- **Module 5: Additional Resources and Closeout**

# WHO SHOULD TAKE THIS COURSE?

- Systems Engineers
- Software Developers
- Software Architects
- Enterprise Architects
- Pre-Sales Engineers
- Post Sales Engineers

# Pre-Reqs

- Basic IT Networking Knowledge
- Basic Database Knowledge

# WHO YOU WILL LEARN FROM?

This course is led by  
industry experts who will  
share their knowledge and  
experience with you



## **Joseph Holbrook**

Joseph is a widely published technologist who has been in the IT field for over 25 years. He is the author of numerous course and books.

# Module 1

**Blockchain Fundamentals**



**101 Blockchains**



# Module Overview

- Lesson 1: What is a Blockchain?
- Lesson 2: Blockchain Components
- Lesson 3: Blockchain Consensus
- Lesson 4: Smart Contracts
- Lesson 5: Demo Anders Blockchain

# LESSON 1

What is a Blockchain?



**101 Blockchains**

# What is a Blockchain?

- A cryptographically secure, shared, distributed ledger.
- Immutable transactions are written on this distributed ledger on distributed nodes
- Transformational technology in which business and government invest in.
- It's a decentralized database which stores information in the form of transactions

# What is a Blockchain?

- **Book = Blockchain**
- **Page = Block**
- **Page Entry = Blockchain transaction**
- **Lets think of the blockchain as a book that can be written to but not erased.**
- **Blockchains can be private or public**
- **Blockchains are revolutionary way of implementing “trust” into a platform.**

# What is a Blockchain?

- A blockchain is a globally shared data structure, transactional backend database.(In Bitcoin it's generally called a ledger)
- This means that everyone can read entries in the database just by participating in the network.
- If you want to change something in the database, you have to create a so-called “transaction” which has to be accepted by all others.
- The word “transaction” implies that the change you want to make (assume you want to change two values at the same time) is either not done at all or completely applied.

# What is a Blockchain?

Lets compare blockchain to other technology

- Telcom network to a telephone
- Databases are centralized where the blockchain is decentralized.
- Blockchains are not built from a new technology. Built from a unique synching of three existing technologies.
- Private or Public

# What is a Blockchain?

**Built from these technologies.**

- **P2P Networks**
- **Private Key Encryption**
- **Programs**

# What is a Blockchain?

**Digital Identity is established**

- **Combining a public and private key creates a strong digital identity reference based on possession.**
- **Private Key**
- **Public Key**



# What is a Blockchain?

Blockchain is revolutionary in several ways

- Blockchain is not new technology but a synching of technologies that now make sense.
- Trust is at the center and essentially removes intermediaries. (efficiency)
- Tamperproof public ledger of value.
- Disruptive to the status quo. Legacy is out
- Platform with numerous use cases.

# What is a Blockchain?

- Blockchain is complex technology but is a simple concept really. Trust is at the center.
- Blockchains are ledgers shared among computers around the world
- The ledgers in a blockchain are immutable.
- Blockchain is not new technology but a synching of technologies

# LESSON 2

Blockchain Key Components



**101 Blockchains**

# Blockchain Technology

## Technology in Blockchain:

- Blockchain is a complex technology but quite a simple concept really. Trust is at the center.
- Blockchains are ledgers shared among computers around the world
- The ledgers in a blockchain are immutable.
- Blockchain is not a new technology but a synching of technologies

# Blockchain Key Components

**Key Blockchain Components are:**

- **Cryptography**
- **P2P Network**
- **Shared Digital ledger**
- **Consensus algorithm**
- **Validity Rules**
- **Virtual Machine**

# Blockchain Key Components

## Blockchain Key Components

- **Cryptography for transactions.**
- **Recorded, encrypted and secured between peers in blockchain.**
- **No need for a centralized authority.**

# Blockchain Key Components

## Blockchain Key Components

- P2P Network connect the blockchain nodes
- All computers share responsibility on the network
- Workloads are shared

# Blockchain Key Components

## Ledgers

- Shared Digital Ledger is a data structure managed inside the node application.
- Distributed Database held and updated independently by each participant (or node) in a large network.
- A ledger is an append-only record store, where records are immutable and may hold more general information than financial records.



# Blockchain Key Components

## Blockchain Key Components

- A distributed ledger is a database that is stored and updated independently by each or node in the blockchain.
- The decentralized and distributed nature is what makes it unique.
- In blockchain they are immutable
- Every single node on the network processes every transaction that occurs.

# Blockchain Key Components

## Consensus

- The Consensus algorithm is implemented as part of the node application for how the ecosystem comes to a single view of the ledger.
- Different ecosystems have different methods for attaining consensus

# Blockchain Key Components

## Consensus

- Consensus is when the distributed ledger has been updated and all nodes maintain their own identical copy of the ledger.
- This architecture allows for a new capacity as a system of recordkeeping that goes beyond being a simple database.
- Determines method for “World State” which is similar to a point in time picture.

# Blockchain Key Components

## Blockchain Key Components

- **Validity Rules (validation)** state how the user and the transactions will be validated.
- **Defines how transactions occur.**

# Blockchain Key Components

## Blockchain Key Components

- Virtual Machines are a representation of a server created by a computer program and operated with instructions embodied in a language.
- Ethereum and Bitcoin use VMS.
- The virtual machine lives in the Ethereum node applications for example

# LESSON 3

Blockchain Consensus



**101 Blockchains**

# Consensus Methods

## Consensus

- Proof of Work
- Proof of Stake
- Delegated Proof-of-Stake (DPoS)
- Byzantine Fault Tolerance (BFT)
- Directed Acyclic Graphs (DAGs)
- Many other algorithms ...

# Consensus Methods

## Consensus

- Proof of Work was the first blockchain consensus algorithm.
- Satoshi Nakamoto created for the Bitcoin blockchain
- PoW, miners solve hard problems to create blocks.
- PoW runs on a system of “the longest chain wins.”
- Expensive both cost and environmentally
- Bitcoin, Ethereum, Litecoin



# Consensus Methods

## Proof of Stake

- Proof of Stake the blocks aren't created by miners doing work
- Instead by minters staking their tokens to “bet” on which blocks are valid.
- In the case of a fork, minters spend their tokens voting on which fork to support.
- Attacks costly but more environmental
- Peercoin and Ethereum will go to.

# Consensus Methods

## Byzantine Fault Tolerance (BFT)

- Classic problem is distributed computing and is explained with Byzantine generals
- Pre-selected generals and meant for a private blockchain
- PBFT runs incredibly efficiently.
- High transaction throughput BUT Centralized/permission based.
- Hyperledger, Ripple and Stellar

# Consensus Methods

## Delegated Proof-of-Stake (DPoS)

- In DPoS, miners can collaborate to make blocks instead of competing like in PoW and PoS.
- By partially centralizing the creation of blocks, DPoS is able to run orders of much faster than most other algorithms.
- Cheap transaction and faster. Centralized
- Steemit and EOS

# Consensus Methods

## Directed Acyclic Graphs (DAGs)

- DAGs are a form of consensus that doesn't use the blockchain data structure and handles transactions mostly asynchronously.
- The plus is theoretically infinite transactions per second
- IOTA, Hashgraph

# Consensus Methods

Consensus Method	Used In	Primary Pros	Primary Cons
Proof of Work (PoW)	BTC, ETH, LTC	Widely tested	Slow and resource intensive
Proof of Stake (PoS)	Peercoin, ETH Casper	Energy efficient	Nothing to stake
Proof of Elapsed Time (POET)	Hyperledger Sawtooth	Participation cost	Specialized hardware (Intel)
Delegated Proof of Stake (DPoS)	Steemit, EOS, LISK	Fast and efficient	Centralized
Delegated Byzantine Fault Tolerance (DBFT)	NEO	Fast and scalable	Root Chain control
Practical Byzantine Fault Tolerance (PBFT)	Hyperledger Fabric	Transaction throughput	Centralized
Federated Byzantine Fault Tolerance (FBFT)	Ripple, Steller	Low cost and high throughput transactions	Centralized

# LESSON 4

Smart Contracts



**101 Blockchains**

# Smart Contracts

- Smart Contract is a term to describe computer program code that is capable of facilitating, executing, and enforcing the negotiation or performance of an agreement using Blockchain technology. (Contract)
- The entire process is automated and can act as a complement or substitute for legal contracts.
- Terms of the smart contract are recorded in a computer language as a set of instructions

# Smart Contracts

- Smart contracts define the rules and penalties around an agreement in the same way that a traditional contract does
- Automatically enforce those obligations
- Several smart contracts can make up a dApp generally



# Smart Contracts

**The Smart Contracts provide:**

- **Autonomy**
- **Trust**
- **Backup**
- **Safety**
- **Speed**
- **Savings**
- **Accuracy**

# Smart Contracts

Types of functions which are required in a Smart Contract:

- **Constructor Function** – The function which is called only once, when you deploy the smart contract. For example, it can be used to receive the initial Ether sent to it, at the time of deployment.
- **Fallback Function** – The function without a name (literally no name, defined as function `() { code... }`) which is invoked when someone sends Ether to the address of your smart contract. In the lack of this function, Ether sent to the smart contract will be rejected.

# Smart Contracts

## Security in Smart Contracts

- Smart contracts provide security that is better than traditional contracts.
- Cut transactional costs associated with traditional contracting.
- Smart contracts on Ethereum network run on Ethereum Virtual Machine (EVM).
- The Decentralized Applications (dApps) running on the Ethereum network are basically complex Smart Contracts.

# Smart Contracts

## Smart Contracts Enforcement

- Basics of Ethereum states that all modifications to a contract's data must be performed by its code.
- Modification of a contract's data requires a blockchain user to send requests to its code. This process kickoff determines whether and how to fulfill those requests.
- A traditional database uses an “enforced stored procedure”.
- Think of this approach as “pre defined rules”.



# Smart Contracts

---

## Smart Contracts Legal Enforcement

- Smart Contracts may not be legally enforceable, especially across borders
- Could be used as evidence
- Think of a vending machine where you put in the required funds to get the drinks or food

# Smart Contracts

## Smart Contract Workflow

- Pre Define
- Trigger Events
- Execute/Transfer
- Settlement

# Smart Contracts

## What is a dApp?

- dApps are decentralized applications.
- dApps run on a P2P network of computers, instead of one computer.
- One or more Smart Contracts.

# Smart Contracts

## What is a dApp?

- Open-source
- Data & the records of operation of application to be cryptographically stored on ledger
- Uses cryptographic token
- Generate tokens
- Decentralized P2P Network
- Connect users directly



# LESSON 5

Demo – Hashing and Blockchain  
Transactions



**101 Blockchains**



# Hashing

- Demo Anders Blockchain Hashing

# Module 2

Ethereum Fundamentals



**101 Blockchains**

# Module Overview

- Lesson 1: What is Ethereum
- Lesson 2: Ethereum Architecture
- Lesson 3: Ethereum Network
- Lesson 4: Ethereum Nodes(EVM)
- Lesson 5: Whiteboard - EVM

# Module Overview

- Lesson 6: Ethereum Consensus
- Lesson 7: Transaction in Ethereum
- Lesson 8: Whiteboard Ethereum Transactions
- Lesson 9: Demo – Ethereum Explorer
- Lesson 10: Ethereum 2.0 Beacon Chain

# LESSON 1

What is Ethereum?



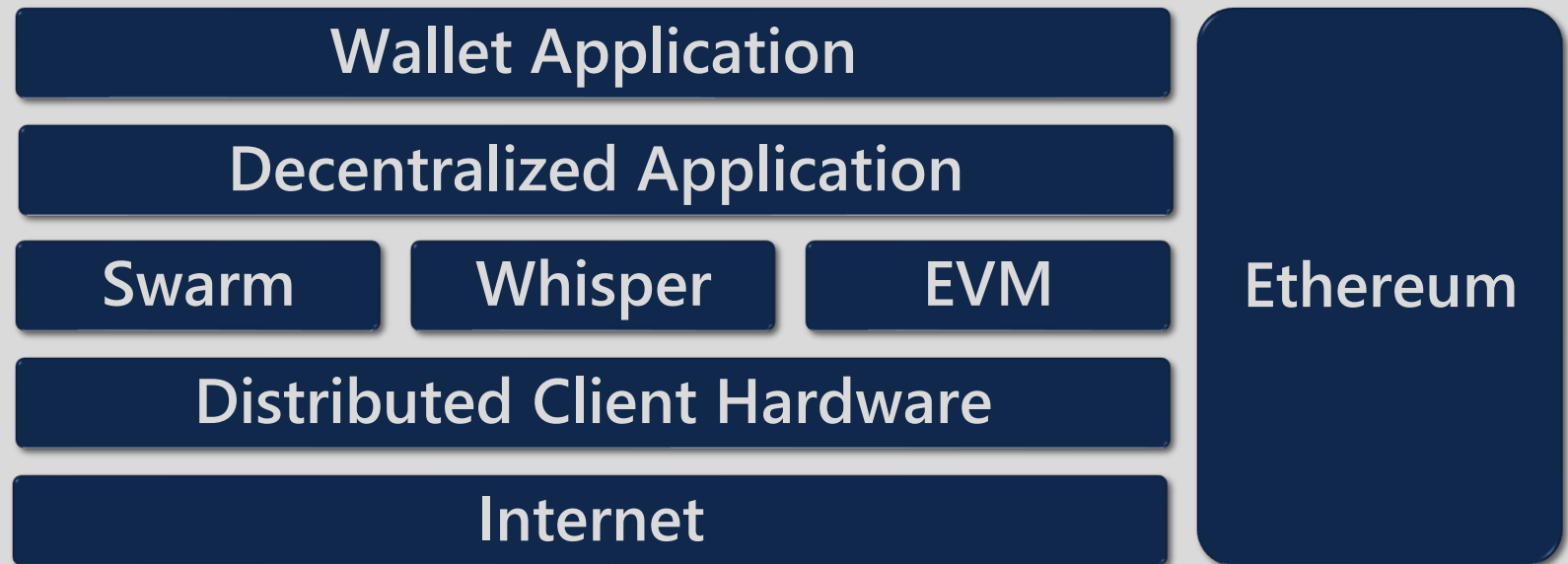
**101 Blockchains**

# Ethereum Overview

- Most widely used Open Source Blockchain-based distributed computing application platform
- Mainly used to building & implementing smart contracts functionality
- It offers a Decentralized Virtual Machine aka Ethereum Virtual Machine (EVM)
- Initiated by Vitalik Buterin in 2013
- Ethereum's live Blockchain was launched on 30 July 2015

# Ethereum Overview

- A client application will access an Ethereum application via the HTTP protocol through an API.
- The smart contract will then be invoked and will run on the EVM.





# Ethereum Overview

- Ether is the native token of the Ethereum blockchain which is used to pay for transaction fees, miner rewards and other services on the network.
- Ethereum is an open software platform based on blockchain technology that enables developers to write smart contracts and build and deploy decentralized applications.

# Ethereum Overview

## Ethereum need to know

- The total supply of Ether and its rate of issuance was decided by the donations gathered on the 2014 presale of the currency.
- 60 million ether created to contributors of the presale
- 12 Million (20% of the above) were created to the development fund (early contributors and developers) and the remaining to the Ethereum Foundation
- 5 ethers are created every block (about 15 seconds) to the miner of the block
- 2-3 ethers may be sent to another miner (Uncle)

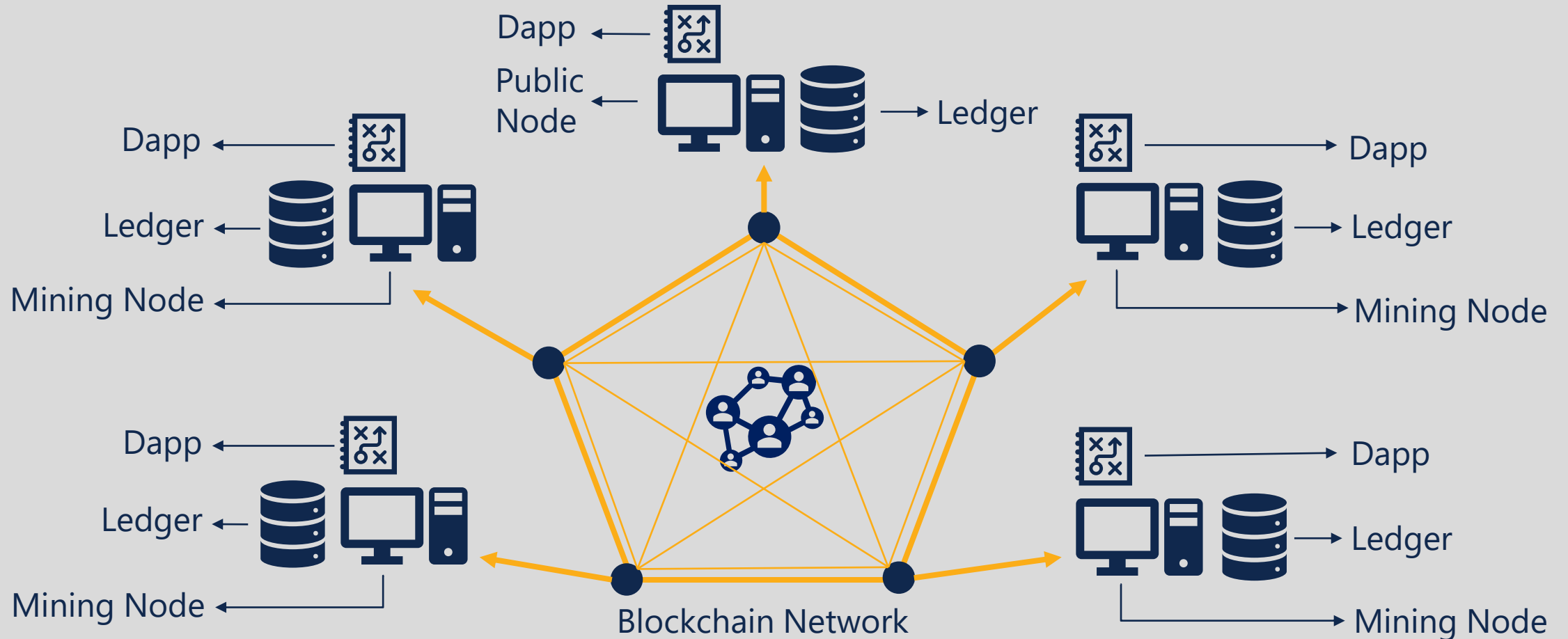
# Ethereum Overview

Ethereum has four main components.

- Network Nodes
- Ethereum Virtual Machines
- Smart Contracts
- dApps

# Ethereum Overview

## Ethereum Public Network



# Ethereum Overview

Ethereum is a transaction based blockchain

- The Ethereum blockchain is a transaction-based state machine.
- In computer science, a state machine refers to something that will read a series of inputs and, based on those inputs, will transition to a new state.
- Genesis State is the beginning state of the blockchain.

# LESSON 2

Whiteboard –Ethereum Overview



**101 Blockchains**

# Ethereum Overview

## Ethereum Network



# LESSON 3

Ethereum Network



**101 Blockchains**



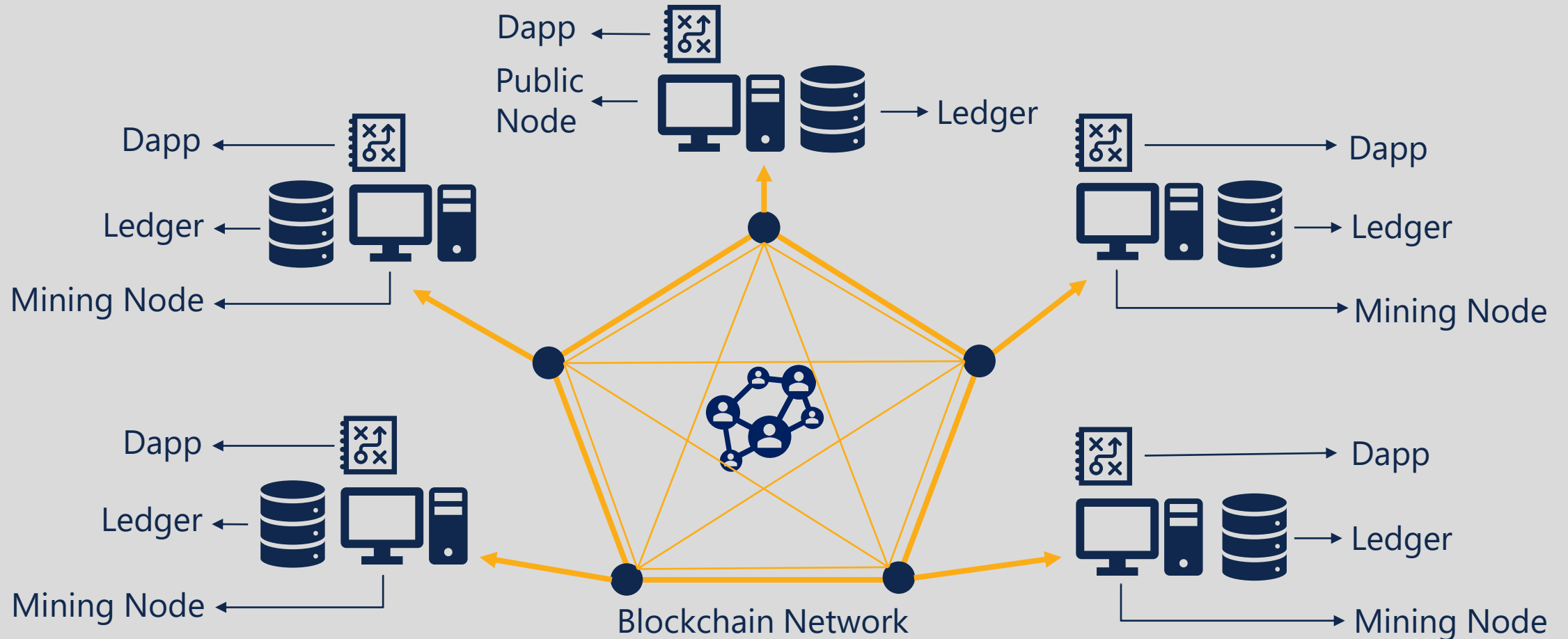
# Ethereum Overview

## Ethereum Mainnet

- MainNet is data on the blockchain including account balances and transactions—are public, and anyone can create a node and begin verifying transactions.
- Ether on this network has a market value and can be exchanged for other cryptocurrency or fiat currencies

# Ethereum Overview

## Ethereum Public Network



# Ethereum Overview

## Three Main types of Extended Ethereum Networks

- Public Test Networks
- Enterprise/Private Networks
- Local Test Networks

# Ethereum Overview

## Public Test Networks

There are several test networks with Ethereum

- Ropsten
- Kovan
- Rinkeby

▲ May be there is a list of network ids? If not, let's create one.

65 Good idea.

✓

- 0 : Olympic, Ethereum public pre-release PoW testnet
- 1 : Frontier, Homestead, Metropolis, the Ethereum public PoW main network
- 1 : Classic, the (un)forked public Ethereum Classic PoW main network, *chain ID* 61
- 1 : Expanse, an alternative Ethereum implementation, *chain ID* 2
- 2 : Morden Classic, the public Ethereum Classic PoW testnet
- 3 : Ropsten, the public cross-client Ethereum PoW testnet
- 4 : Rinkeby, the public Geth-only PoA testnet
- 5 : Goerli, the public cross-client PoA testnet
- 6 : Kotti Classic, the public cross-client PoA testnet for Classic
- 8 : Ubiq, the public Gubi main network with flux difficulty *chain ID* 8
- 42 : Kovan, the public Parity-only PoA testnet
- 77 : Sokol, the public POA Network testnet
- 99 : Core, the public POA Network main network
- 100 : xDai, the public MakerDAO/POA Network main network
- 401697 : Tobalaba, the public Energy Web Foundation testnet
- 7762959 : Musicoin, the music blockchain
- 61717561 : Aquachain, ASIC resistant chain
- [other] : Could indicate that your connected to a local development test network.

share improve this answer edited Mar 19 at 8:30 community wiki 12 revs, 7 users 62% soc1c

add a comment

# Ethereum Overview

## Local Test Network

- The Ethereum blockchain can be simulated locally for development.
- Local test networks process transactions instantly and Ether can be distributed as desired.
- Ethereum simulator
- Ganache

# Ethereum Overview

## Ethereum Network Status Monitor

Check the health of the Ethereum network

<https://ethstats.net/>

- Must add your node to the network to be seen
- Does not represent the full state of the network.
- Shows Key Performance Indicators

# LESSON 4

Ethereum EVM



**101 Blockchains**

# Ethereum Overview

## Ethereum Nodes (Clients)

- A node is a device, program, virtual machine that communicates with the Ethereum network.
- Ethereum Nodes run the Ethereum blockchain which also entails the EVM as well.
- Ethereum Nodes maintain the ledger strictly via a Proof of Work Consensus.
- Ethereum Nodes mine the blockchain for Ether



# Ethereum Overview

When joining the Ethereum network you have some options about the type of node.

The options currently are:

- Light node
- Full node
- Archive node

# Ethereum Overview

## Ethereum Node Types defined

- Full nodes verify the block is broadcasted onto the network
- Light nodes do not verify every block or transaction and do not have a copy of the current blockchain state
- Archive nodes are full nodes that preserve the entire history of transactions

# Ethereum Overview

## Ethereum EVM

- Ethereum Virtual Machine (EVM) is built into the software running on the Ethereum protocol. It executes smart contracts - Ethereum programs written in the Solidity language.
- The EVM is contained in the full nodes of the Ethereum network, inside of which it executes these Ethereum-user-written programs

# Ethereum Overview

## Ethereum EVM

- EVM code is the programming language in which accounts on the Ethereum blockchain can contain code.
- The EVM code associated with an account is executed every time a message is sent to that account, and has the ability to read/write storage and itself send messages.

# LESSON 5

Whiteboard EVM



**101 Blockchains**

**EVM**

**EVM Structure**



# LESSON 6

Ethereum Consensus and Ledger



**101 Blockchains**

# Ethereum Consensus

- Consensus is when the distributed ledger has been updated and all nodes maintain their own identical copy of the ledger.
- This distributed architecture allows for a new capacity as a system of recordkeeping that goes beyond being a simple database.
- Consensus in Ethereum currently uses a consensus protocol called Proof-of-work (PoW).
- Algorithm in Ethereum is called "Ethash".



# Ethereum Consensus

- Consensus will be changing from a Proof of Work (PoW) to a Proof of Stake (PoS)
- Miners are still needed and will be until full changeover in 2022
- Mining will go to Staking.
- More on this in the Ethereum 2.0 Lesson

# Ethereum Ledger

In Ethereum, the ledger is used for blockchain transactions.

- To access ledger, you need the keys.
- Ethereum uses public key cryptography for authentication.
- Elliptic Curve Digital Signature Algorithm(ECDSA) secp256k1's curve is used.
- The private key is random 256 data.

# LESSON 7

## Transactions in Ethereum

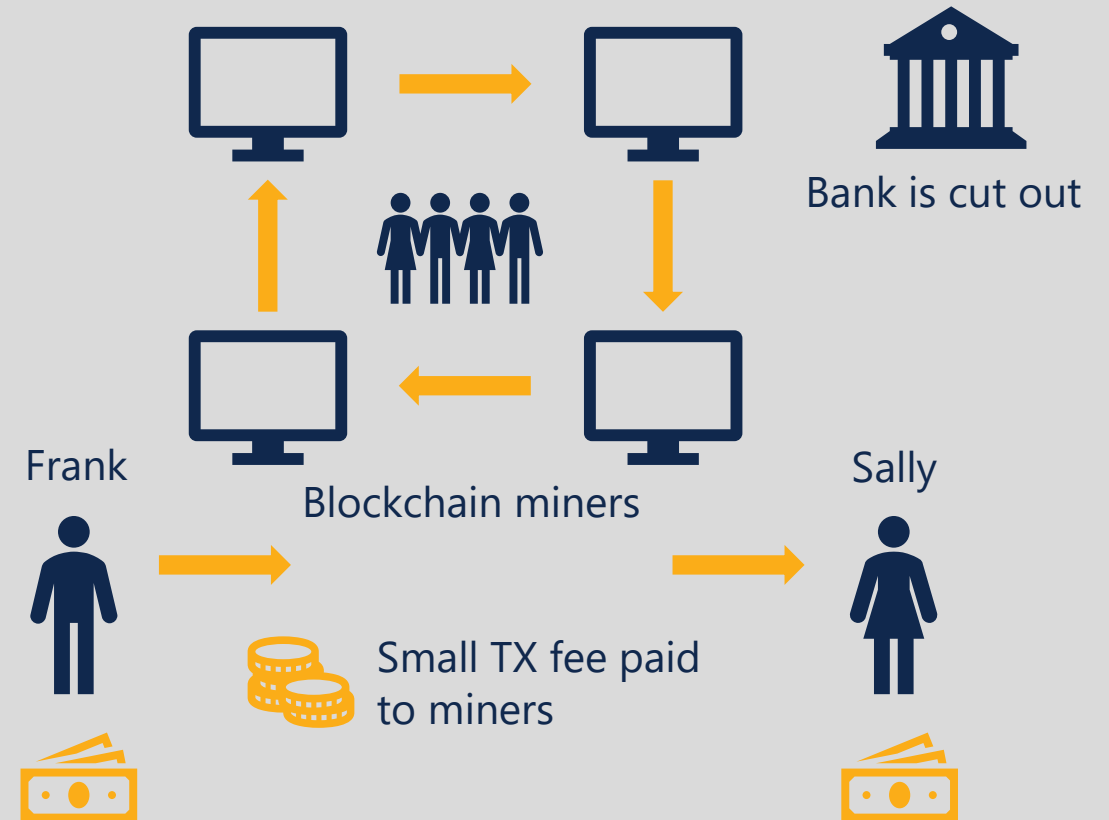


**101 Blockchains**

# Ethereum Overview

## Ethereum Transactions

- For a transaction to be considered valid it will be processed by a validation process known as blockchain mining.
- Mining is when a group of nodes use their computing resources to create a block of valid transactions.

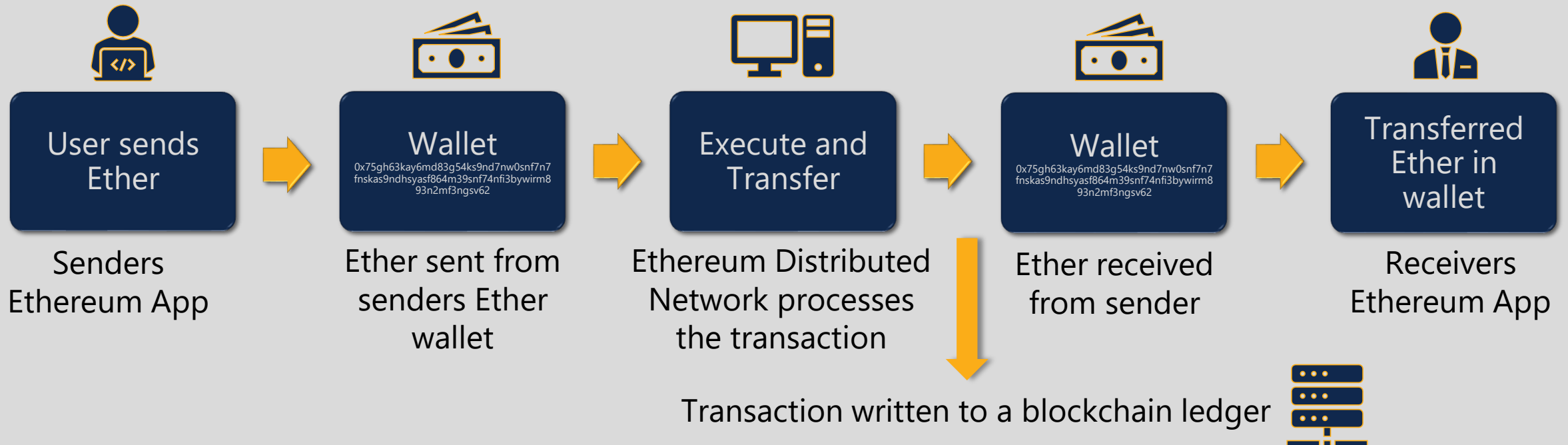


Blockchain model – Frank wants to send funds to Sally and can use a P2P “cryptocurrency” to bypass a bank or transfer services along with high fees and slow service.

# Ethereum Overview

## Ethereum Transactions

- Example of how Ethereum blockchain transactions occur between two parties.



# Ethereum Overview

## Ethereum Transactions

- Smart contract is a term used to describe computer program code that is capable of facilitating, executing, and enforcing the negotiation or performance of an agreement using Blockchain technology. (Contract)
- The entire process is automated can act as a complement, or substitute, for legal contracts.
- Terms of the smart contract are recorded in a computer language as a set of instructions

# LESSON 8

**Whiteboard - Transactions in  
Ethereum**



**101 Blockchains**

# Transactions

## Transactions in Ethereum





# LESSON 9

Demo – Ethereum Explorer



**101 Blockchains**



# Ethereum Metamask

- Demo Metamask Transactions

# LESSON 10

Ethereum 2.0 Beacon Chain



**101 Blockchains**

# Ethereum 2.0

- Ethereum is currently being designed to use a different consensus method in the future
- Ethereum 2.0 - term that describes all the updates from Serenity, which will make Ethereum more scalable, faster and a better blockchain
- Changes to Consensus, EVM and Mining (2021 or later)
- New "Chain" will be formed. (Forking)

# Ethereum Consensus

## Etheruem EVM Changes important for consensus

- The new EVM called eWASM stands for Ethereum WebAssembly.
- WebAssembly is an open standard defining a portable binary code format for executable programs.

# Ethereum Consensus

## Beacon Chain

- This new chain will have a Proof of Stake (PoS) consensus algorithm, and it will run in parallel to the main PoW Ethereum blockchain.
- Beacon Chain - coordination mechanism of the new network, responsible for creating new blocks and making sure those new blocks are valid. (Rewards validators with ETH)
- The PoS algorithm is called Casper.

# Ethereum Consensus

Potential changes coming in later 2022 to Ethereum

- Official End to Proof of Work Mining
- Cross-shard transactions
- Lightweight clients
- Super-square charting
- Closer ties with developers

# Module 3

Ethereum Development  
Fundamentals



**101 Blockchains**



# Module Overview

- Lesson 1: Smart Contracts
- Lesson 2: Dapps
- Lesson 3: Smart Contract Workflow
- Lesson 4: What is Ether
- Lesson 5: What is Gas

# Module Overview

- Lesson 6: Demo Gas Estimate
- Lesson 7: Development Tools
- Lesson 8: Ethereum Development Overview
- Lesson 9: Solidity Basics
- Lesson 10: APIs

# LESSON 1

Smart Contracts



**101 Blockchains**

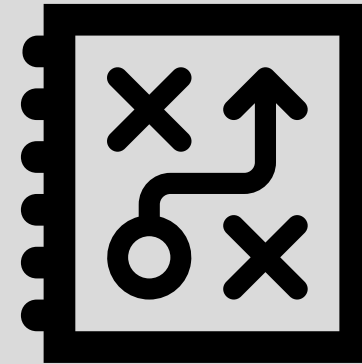
# Smart Contracts

Smart contract is a term used to describe computer program code that is capable of facilitating, executing, and enforcing the negotiation or performance of an agreement using Blockchain technology. (Contract)

- The entire process is automated can act as a complement, or substitute, for legal contracts.
- Terms of the smart contract are recorded in a computer language as a set of instructions

# Smart Contracts

- Smart contracts define the rules and penalties around an agreement in the same way that a traditional contract does
- Automatically enforce those obligations.
- Several smart contracts can make up a dapp generally



# Smart Contracts

- Smart contracts provide security that is better to traditional contracts.
- Cut transactional costs associated with traditional contracting.
- Smart contracts on Ethereum network run on something called Ethereum Virtual Machine (EVM)
- The Decentralized Applications (DApps) running on the Ethereum network are basically complex Smart Contracts.

# Smart Contracts

Ethereum Smart Contracts provide:

- Autonomy
- Trust
- Backup
- Safety
- Speed
- Savings
- Accuracy

# Smart Contracts

Ethereum Smart Contracts functions which are required in a smart contract:

- **Constructor Function** - This function is called only once when a smart contract is deployed.
- **Fallback Function** - The function without a name (literally no name, defined as `function () { code... }`) which is invoked when someone sends Ether to the address of your smart contract. In the lack of this function, Ether sent to the smart contract will be rejected.



# Smart Contracts

Ethereum Smart Contracts functions which are required in a smart contract:

- Smart contracts provide security that is better to traditional contracts.
- Cut transactional costs associated with traditional contracting.
- Smart contracts on Ethereum network run on something called Ethereum Virtual Machine (EVM)
- The Decentralized Applications (DApps) running on the Ethereum network are basically complex Smart Contracts.

# Smart Contracts

## Ethereum Smart Contracts Enforcement

- Basics of Ethereum states that all modifications to a contract's data must be performed by its code.
- Modification of a contract's data requires a blockchain user to send requests to its code. This process kickoff determines whether and how to fulfill those requests.
- A traditional database uses an “enforced stored procedure”.
- Think of this approach as “pre defined rules”

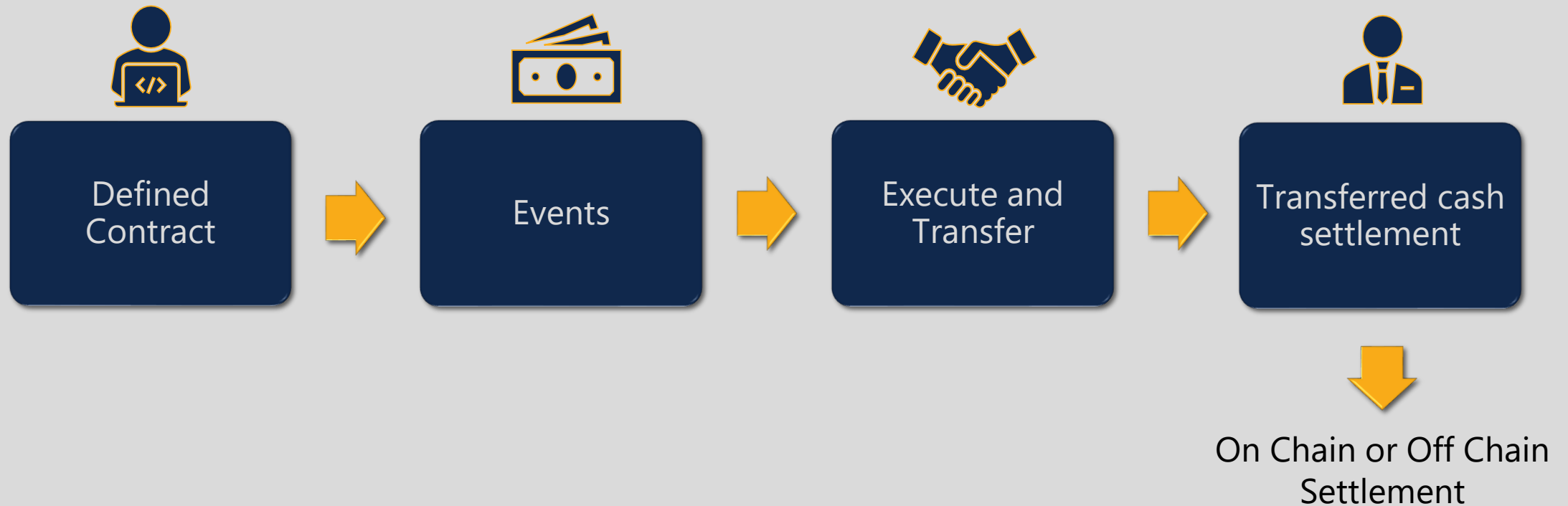
# Smart Contracts

## Smart Contract Workflow

- Pre Define
- Trigger Events
- Execute/Transfer
- Settlement

# Smart Contracts

## Ethereum Smart Contracts Workflow 4 Steps



# LESSON 2

dApps



**101 Blockchains**

# dApps

## Smart Contracts (dApps)

- dApps are “decentralized applications” - These applications run on a P2P network of computers, instead of a one computer.
- One or more Smart contracts
- DappRadar has a concise list

The screenshot shows the DappRadar website interface. At the top, there's a navigation bar with links like Rankings, Portfolio, Reports, Industry Overview, Blog, OpenData, and Developers. Below the navigation bar, a header section reads "Track, analyze and discover decentralized applications (dapps)". Underneath, there's a "Hot Topics" section with buttons for DeFi, NFT Marketplaces, Games, Gambling, and Exchanges. The main content area displays three columns of application lists. The first column is titled "DeFi" and lists applications like BAEX, PancakeSwap, RICETRON, JulSwap, and Synthetix. The second column is titled "Gambling" and lists SportBet, Serious Dice, EarnBet, WINK, and RocketGame. The third column is titled "Exchanges" and lists BAEX, Uniswap, 1inch.exchange, SushiSwap, and JustSwap. Each application entry includes its logo, name, the blockchain it runs on, and its current volume and user count.

Category	Application	Blockchain	Users	Volume
DeFi	BAEX	ETH	-	-
	PancakeSwap	BSC	15,745	-8.25%
	RICETRON	TRON	2,463	-0.89%
	JulSwap	BSC	1,907	-5.27%
	Synthetix	ETH	1,837	-2.91%
Gambling	SportBet	EOS	-	-
	Serious Dice	ThunderCore	23,109	+195.21%
	EarnBet	WAX	2,157	+0.24%
	WINK	TRON	1,005	-3.94%
	RocketGame	TRON	973	-12.42%
Exchanges	BAEX	ETH	-	-
	Uniswap	ETH	45,887	+2.55%
	1inch.exchange	ETH	3,255	-1.49%
	SushiSwap	ETH	2,585	-9.84%
	JustSwap	TRON	2,385	+17.17%

# dApps

- Open-source
- Data & the records of operation of application to be cryptographically stored on ledger
- Uses cryptographic token
- Generate tokens
- Decentralized P2P Network

# LESSON 3

**Whiteboard - Smart Contract  
Workflow**



**101 Blockchains**



# Smart Contracts

How Smart Contracts  
should flow in Ethereum



# LESSON 4

What is Ether



**101 Blockchains**

# What is Ether?

## Tokenization

- Ether is the native token of the Ethereum blockchain which is used to pay for transaction fees, miner rewards and other services on the network.
- Using the Ethereum network requires you to pay for the privilege with "Gas"
- Two types of tokens - Usage tokens and work tokens

# What is Ether?

## ERC20 Tokens

- ERC20 defines a common list of protocols that an Ethereum token has to implement.
- Allows developers the ability to program behavior of new tokens within the Ethereum ecosystem.
- Common with crowdfunding companies via Initial Coin Offerings (ICO).

# Ethereum Overview

## Ethereum Token types

- Usage Tokens
- Work Tokens

# Ethereum Overview

## Usage Tokens

- These are tokens in specific blockchain that are similar to their own native currency in their DAPPS.
- Example is Golem

# Ethereum Overview

## Work Tokens

- These are the tokens that identify you as a sort of shareholder in the DAPP.
- As a shareholder you generally have some voting rights.
- Dash is a good example

# LESSON 5

What is Gas



**101 Blockchains**



# What is Gas?

## Ethereum Ether vs Gas

- Ether is the native token of the Ethereum blockchain which is used to pay for transaction fees, miner rewards and other services on the network.
- Using the Ethereum network requires you to pay for the privilege with "Gas"

# What is Gas?

## Ethereum Gas

- Gas is a measurement roughly equivalent to computational steps for Ethereum. Gas is used to maintain stability due to price fluctuations with Ether.
- Every transaction is required to include a gas limit and a fee that it is willing to pay per gas.
- Miners have the choice of including the transaction and collecting the fee or not.
- Every operation has a gas expenditure on an EVM.

# What is Gas?

- Gas price is the amount of ether you are willing to spend on every unit of gas. You need to determine the price of gas you want to pay for as well as maximum amount of gas you want to spend on performing a transaction on the Ethereum network.
- The more important the processing of the transaction the price should be higher..

# What is Gas?

- Use Metamask for confirming your transaction
- Balance
- Gas Limit
- Gas Price
- Can proceed to buy more Ether at Coinbase or Shapeshift

# What is Gas?

- Gwei = Unit of Ethereum coin used to calculate transaction fees
- One Ethereum coin is worth 1 billion (1,000,000,000) gwei.

<https://www.myetherwallet.com>

# What is Gas?

## Ethereum Gas

- Ether Units
- One Ethereum coin is worth 1 billion (1,000,000,000) Gwei

Unit	Wei Value	Wei
wei	1 wei	1
Kwei (babbage)	1e3 wei	1,000
Mwei (lovelace)	1e6 wei	1,000,000
Gwei (shannon)	1e9 wei	1,000,000,000
microether (szabo)	1e12 wei	1,000,000,000,000
milliether (finney)	1e15 wei	1,000,000,000,000,000
ether	1e18 wei	1,000,000,000,000,000,000

<http://ethdocs.org/en/latest/ether.html>

# What is Gas?

➤ ETH Gas Station aims to increase the transparency of gas prices, transaction confirmation times, and miner policies on the Ethereum network.

➤ <https://ethgasstation.info/>

# LESSON 6

Demo - Gas



**101 Blockchains**





# Ethereum Gas

- Demo Gas Estimate

# LESSON 7

Development Tools



**101 Blockchains**

# Ethereum Tools

## Tools with Ethereum

- Languages – Solidity, Serpent, Mutan
- IDE – Solidity Browser, Ethereum Studio
- Clients – geth, eth, parity, Ethereum Wallet
- Storage – IPFS, Swarm and Storj
- Dapp Browsers – Metamask or Mist
- Testing – Testnet, TestRPC

# Ethereum Tools

## Frontend Tools used with Ethereum:

- HTML
- CSS
- JavaScript

# Ethereum Tools

## Backend Tools used with Ethereum:

- Solidity
- Serpent
  
- Download Solidity or compile with your browser.

# LESSON 8

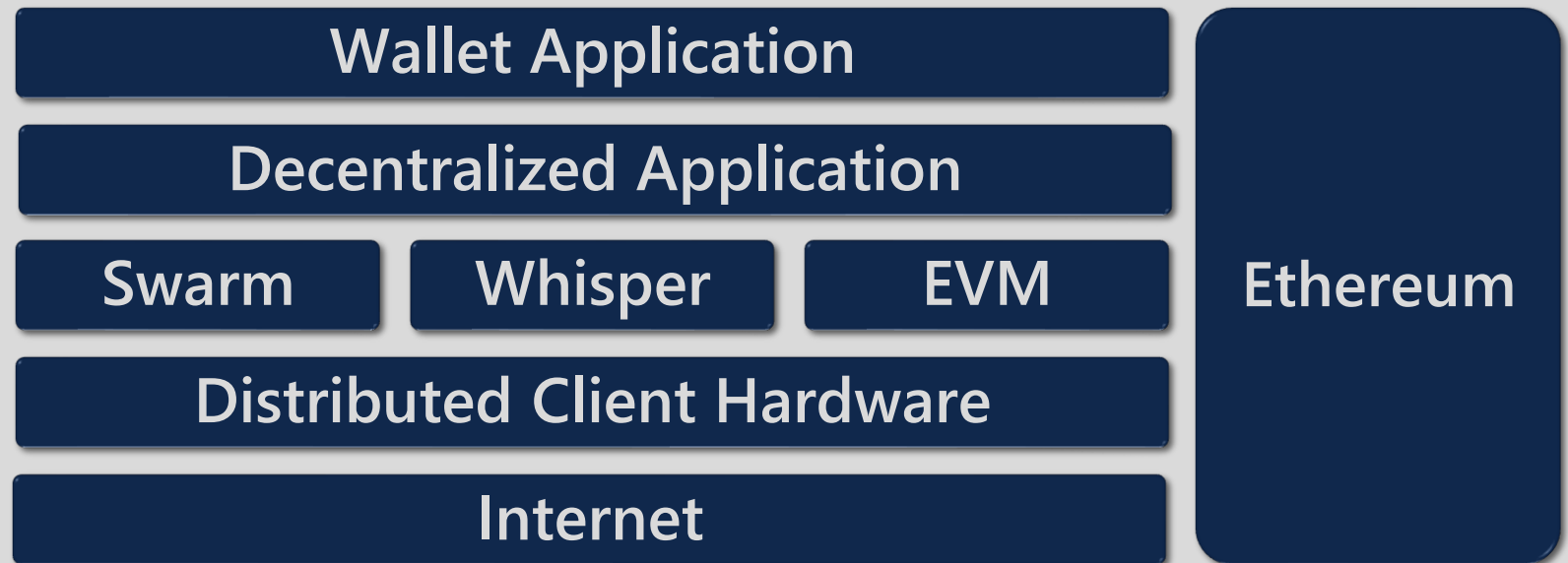
Ethereum Development



**101 Blockchains**

# Ethereum Overview

- A client application will access an Ethereum application via the HTTP protocol through an API.
- The smart contract will then be invoked and will run on the EVM.
- Ethereum Development includes both the Front End and Backend for developing applications



# LESSON 9

Solidity



**101 Blockchains**



# Ethereum Solidity

- Solidity, the language behind Ethereum, is specifically designed to utilize the Ethereum Virtual Machine or EVM.
- Ethereum-based applications and Smart Contracts are written in Solidity.
- Solidity was proposed in August 2014 by Dr. Gavin Wood for Ethereum
- It has similarities to JavaScript and C.

# Ethereum Solidity

- Solidity uses a whole new framework but it is very similar to the popular JavaScript.

## Solidity vs JavaScript

- Solidity is kind of an Object Oriented language just like C++ and C# whereas JavaScript is based on HTML.
- Solidity is designed specifically for Ethereum applications and it runs only on the Ethereum blockchain.
- JavaScript is a universal language for the web and is being used in a large number of applications.

# LESSON 10

APIs



**101 Blockchains**

# Ethereum APIs

## APIs

- JSON is a lightweight data-interchange format. It can represent numbers, strings, ordered sequences of values, and collections of name/value pairs.
- To talk to an Ethereum node from inside a JavaScript application, use the web3.js library, which gives an convenient interface for the RPC methods.

# Module 4

Ethereum Development Tools



**101 Blockchains**

# Module Overview

- Lesson 1: Metamask
- Lesson 2: Demo - Metamask
- Lesson 3: Demo - Parity
- Lesson 4: Demo - Geth
- Lesson 5: Demo – Ethereum Wallets

# Module Overview

- Lesson 6: Demo Test Ether
- Lesson 7: Ethereum Testing
- Lesson 8: Demo – Truffle
- Lesson 9: Demo – Ganache
- Lesson 10: Demo - Remix

# LESSON 1

Metamask

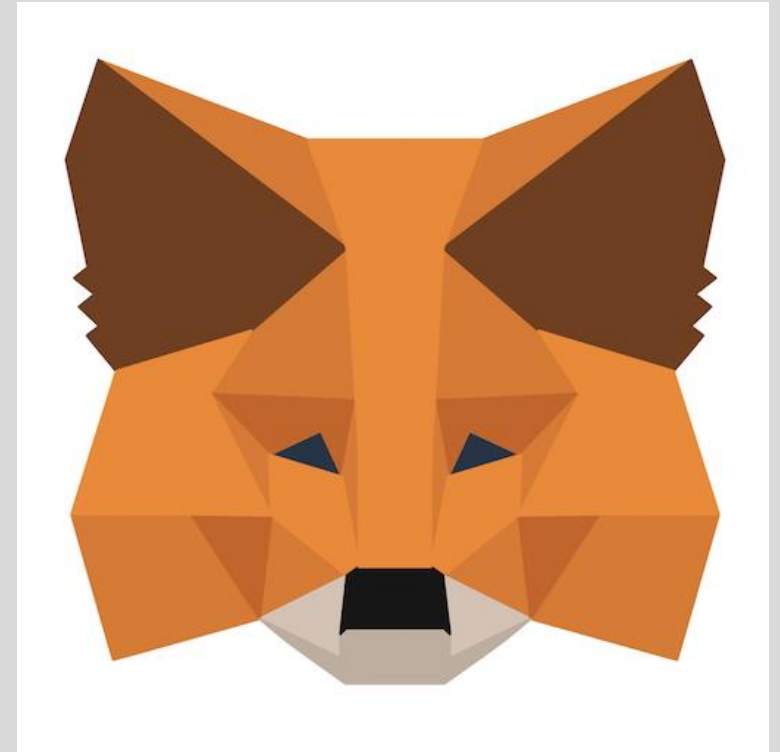


**101 Blockchains**



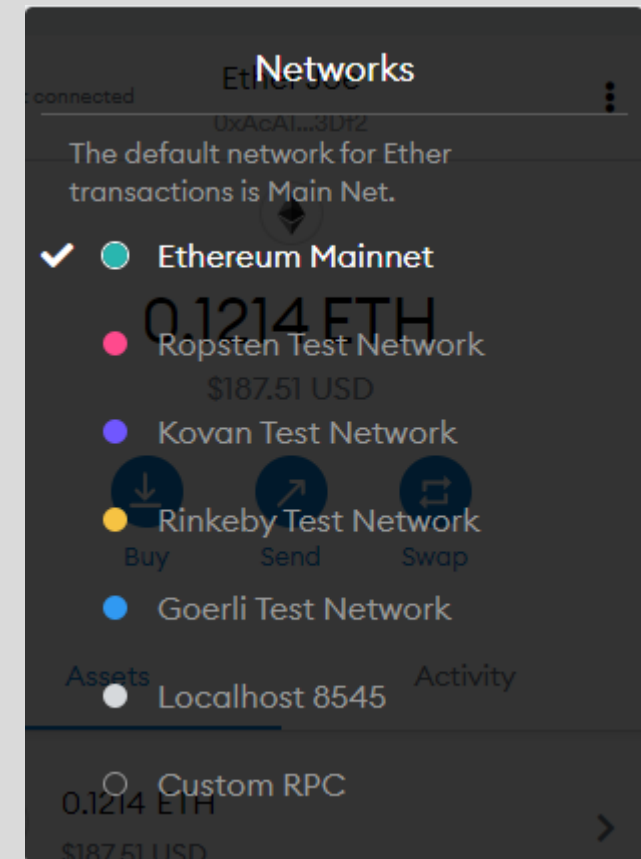
# Ethereum MetaMask

- MetaMask is a bridge that allows you to visit the distributed web of tomorrow in your browser today.
- It allows you to run Ethereum dApps right in your browser without running a full Ethereum node.
- MetaMask injects a javascript library called web3.js.



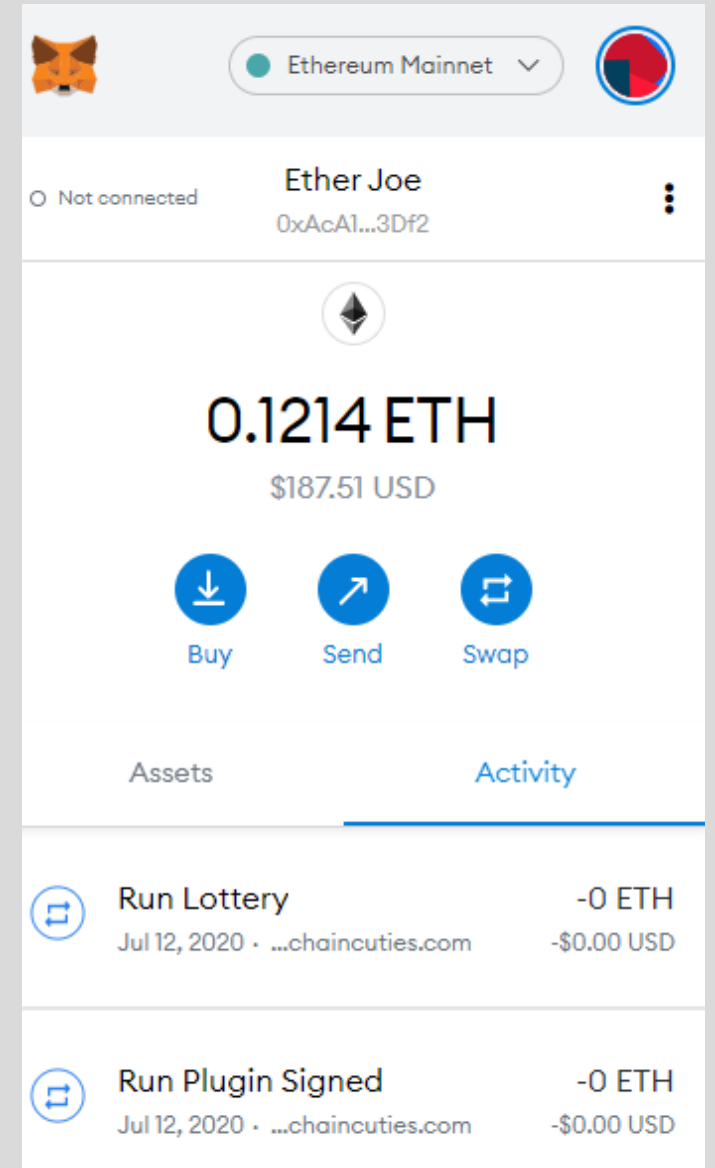
# Ethereum MetaMask

- MetaMask includes a secure identity vault, providing a user interface to manage your identities on different sites and sign blockchain transaction.
- MetaMask could also be considered a "Zero Client"
- Zero Clients run in the browser in JavaScript



# Ethereum MetaMask

- Metamask is the best ERC20 compatible wallet , it provides you extra layer of security from phishing sites.
- Download via Browser extension in Chrome.



# LESSON 2

Demo - Metamask



**101 Blockchains**



# Ethereum Metamask

- Demo Metamask

# LESSON 3

Parity



**101 Blockchains**



# Ethereum Parity Lightweight Browser

- Demo Parity

# LESSON 4

Geth



**101 Blockchains**





# Ethereum Geth

- Demo Geth

# LESSON 5

Demo – Ethereum Wallet



**101 Blockchains**



# Ethereum Wallets

- Demo Wallets

# LESSON 6

Demo – Test Ether



**101 Blockchains**



# Ether

- Obtain Test Ether

# LESSON 7

Ethereum Testing



**101 Blockchains**



# Ethereum Testing

- Ethereum Testing Options

# LESSON 8

Demo Truffle



**101 Blockchains**





# Development Tools

- Truffle

# LESSON 9

Demo Ganache



**101 Blockchains**



# Development Tools

- Ganache

# LESSON 10

Demo Remix



**101 Blockchains**



# Development Tools

- **Remix**

# Module 5

**Additional Resources and  
Closeout**



**101 Blockchains**

# Module Overview

- Lesson 1: Summary Review
- Lesson 2: Blockchain Roles
- Lesson 3: Resources
- Lesson 4: Closeout

# LESSON 1

## Summary Review



**101 Blockchains**



# Course Review

---

- Enterprise blockchains are generally considered private permissioned.
- Proof of Work was the first blockchain consensus algorithm.
- dApps are “decentralized applications” - These applications run on a P2P network of computers, instead of a one computer.
- Terms of the smart contract are recorded in a computer language as a set of instructions
- Smart contracts on Ethereum network run on Ethereum Virtual Machine (EVM).

# Course Review

---

- Ether is the native token of the Ethereum blockchain which is used to pay for transaction fees, miner rewards and other services on the network.
- Using the Ethereum network requires you to pay for the privilege with “Gas”
- MetaMask allows you to run Ethereum dApps right in your browser without running a full Ethereum node.
- Solidity is the development language of Ethereum

# LESSON 2

Blockchain Roles and Demand



**101 Blockchains**

# Blockchain Roles

Roles that dedicated to blockchain is somewhat limited. Here are the most common roles.

- **Blockchain Architect**
- **Blockchain Developer**
- **Blockchain Trainer**
- **Blockchain Operator**
- **Blockchain Manager**

# Blockchain Roles

Other roles are commonly required to have blockchain knowledge such as.

- Software Engineer
- Software Developer
- Software Architect
- Systems Engineer
- Enterprise Architect
- Cloud Architect

# Blockchain Roles

Other roles are commonly required to have blockchain knowledge such as.

- CIO
- IT Manager
- Corporate Counsel
- IT Sales Executive
- Pre-Sales Engineer

# Blockchain Demand

- The blockchain market is clearly expanding and this market demand for expertise is clearly evident in cities such as NYC, Boston, Austin, San Francisco and the Bay Area, Toronto, Singapore, London and even Chicago.
- The demand for this expertise is not just in major cities, I am seeing lots of startups in smaller cities as well. The great part is that you don't need to be in a "tech hub" to get a blockchain role now.
- LinkedIn Learning named "blockchain" as the number 1 hard skill for 2020

# LESSON 3

**Additional Resources**



**101 Blockchains**



# Additional Resources

## Resources to learn more about Ethereum

### ➤ Ethereum Website

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

# Additional Resources

## Resources to learn more about Blockchain

- **Certified Enterprise Blockchain Architect (CEBA) course from 101 Blockchains.**

- <https://academy.101blockchains.com/courses/certified-enterprise-blockchain-architect>

- **Certified Enterprise Blockchain Professional (CEBP) course from 101 Blockchains.**

- <https://academy.101blockchains.com/courses/blockchain-expert-certification>

# LESSON 4

Closeout



**101 Blockchains**

# Thank you for attending!

 [101blockchains.com](http://101blockchains.com)

 [contact@101blockchains.com](mailto:contact@101blockchains.com)

 [linkedin.com/company/101blockchains](https://linkedin.com/company/101blockchains)



## 101 Blockchains

© 2021 101 Blockchains. All rights reserved. This document may not be distributed, transmitted or reproduced in any form or by any means without 101 Blockchains' prior written permission. While the information contained in this document has been obtained from sources believed to be reliable, 101 Blockchains disclaims all warranties as to the completeness or accuracy. Although 101 Blockchains research may address business, financial, investment and legal issues, 101 Blockchains does not provide any business, financial, legal or investment advice and this document should not be construed or used as such. 101 Blockchains shall not be responsible for any loss sustained by any person who relies on this publication.