

# QUICKSTART

---

How to design, code, deploy and execute a smart contract



# Instructor Introduction

- Joseph Holbrook
  - Consulting Blockchain Solutions Architect/Trainer/Speaker out of Jacksonville, FL
  - Certified Blockchain Solutions Architect (CBSA)
  - Certified Blockchain Expert (CBE)
  - Certified Bitcoin Professional (CBP)
  - Certified Blockchain Developer Hyperledger (CBDH)
  - Certified Corda Developer
  - Certified Google Cloud Platform Cloud Architect
  - Certified AWS Solutions Architect
  - Brocade Distinguished Architect (BDA) 2013
  - EMC Proven Professional – Expert – Cloud (EMCCE)
  - Published Course Author on Pearson Safari, Udemy, LinkedIn Learning
  - Author “Architecting Enterprise Blockchain Solutions” – Wiley July 2019
  - Prior US Navy Veteran



# Webinar Objectives

---

By the end of this webinar you should be able to understand

- The concepts, use cases and basics of smart contracts
- How Blockchain and smart contracts work and developer success
- How smart contracts work on both the Ethereum and Hyperledger platforms from a practical level
- The constructs of smart contract, common coding requirements and demos
- What are the most in demand Blockchain Certifications?
- How do these certification meet the needs of todays Enterprises?
- What about Blockchain Career Demand?

# How to design, code, deploy and execute a smart contract

---

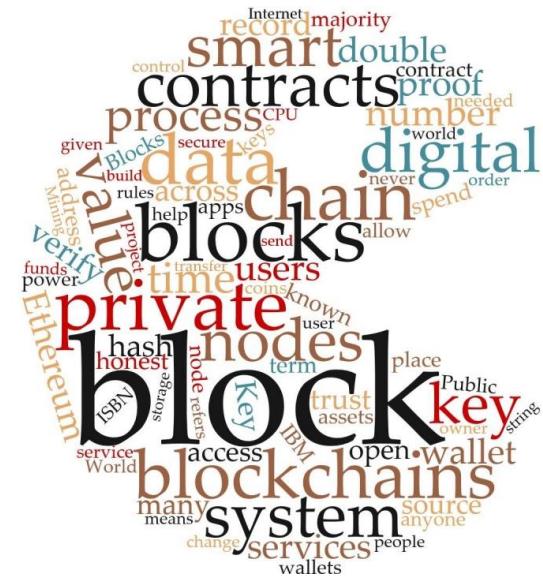
## **The concepts, use cases and basics of smart contracts**

# The concepts, use cases and basics of smart contracts

# Smart Contracts

- The concepts, use cases and basics of smart contracts are generally the same for all blockchain platforms.

Before we get into details lets discuss what a smart contract is.



# The concepts, use cases and basics of smart contracts

---

## 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

# The concepts, use cases and basics of smart contracts

---

## Smart Contracts

- Smart contracts are becoming a central capability of blockchain platforms, where rules can be embedded into the blockchain through code.

From a context of Blockchain, smart contracts are:

- Computer program, run autonomously
- Stored and replicated on a distributed storage platform ,
- Executed/run by a network of computers
- Can result in ledger updates (e.g. transaction data)

# The concepts, use cases and basics of smart contracts

## Smart Contracts



- Contract made
- Individuals involved are anonymous (Permissionless)
- Contract is deployed on the blockchain

- When triggered (event)/ received, contract executes
- Contract performs agreed action

- Contract code is publicly available
- Compliance

# The concepts, use cases and basics of smart contracts

## Smart Contracts Provide

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



Dilbert Cartoon - Scott Adams

# The concepts, use cases and basics of smart contracts

## Comparing Smart Contracts

<i>Traditional contracts</i>	<i>Smart contracts</i>
 1-3 Days	 Minutes
 Manual remittance	 Automatic remittance
 Escrow necessary	 Escrow may not be necessary
 Expensive	 Fraction of the cost
 Physical presence (wet signature)	 Virtual presence (digital signature)
 Lawyers necessary	 Lawyers may not be necessary

# The concepts, use cases and basics of smart contracts

## Smart Contracts Define?

- 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



# The concepts, use cases and basics of 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.

# The concepts, use cases and basics of smart contracts

---

## DaPPS Define?

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

# The concepts, use cases and basics of smart contracts

---

## Transactions

- 'Transaction' in Ethereum - Signed data package that contains a message to be sent from an externally owned account to another account on the Ethereum blockchain
- The state of all accounts is the state of the Ethereum network
- State of the network is updated with every block

# How to design, code, deploy and execute a smart contract

---

## Developer Success

# Developer Success

## Common Development Languages

- Python
- Golang
- Javascript
- Solidity

When trying to find blockchain developer roles. Consider these as a starting point in common blockchain development roles:



## Examples

Ethereum – C++,  
Solidity

Ripple – Python

Hyperledger Fabric  
– Node JS, Java,  
and Chaincoding

# Developer Success

---

Common Development roles:

- Enterprise Skills (Networking, Security)
- Agile, DevOps, Angular, Spring Development Skills
- Compliance, Legal, and Regulatory experience
- Open Source Interopability
- APIs, Node.Js, Solidity, C++, Go, Python, Java, etc.

# How to design, code, deploy and execute a smart contract

---

## **How Blockchain and smart contracts work**

# How Blockchain and smart contracts work

---

## Smart Contracts Development

- Solidity is the language to code Ethereum smart contracts
- The syntax is very similar to JavaScript
- It's designed with the Ethereum Virtual Machine in mind
- An ever growing language in use
- Latest version is 0.4.25 as of 1st July 2018

# How Blockchain and smart contracts work

## Smart Contracts Development

Solidity contracts are much like classes in object-oriented programming languages such as Java

- Every contract contains declarations of elements such as
- Functions: Blocks of code that are executed when called
- Function modifiers
- State variables: State variables are values, permanently stored in the contract's storage
- Struct, enum and other data types,
- Events as needed for its purpose
- Contracts in Solidity can inherit elements from other contracts

# How Blockchain and smart contracts work

---

Solidity has four types of visibilities for both functions and variables:

- Public: Can be called internally or through messages
- Private: Only available to the current contract and not derived contracts
- Internal: Can only be accessed internally (current contract or derived)
- External: Can be called from other contracts and transactions. They cannot be called internally, except with "this.functionName()"

# How Blockchain and smart contracts work

## Ethereum Variables and Types

**Int** Integers, or number values including negative value

**Uint (unsigned)** Integers, or number values

**Address** A 20 bytes value, which is meant to store an Ethereum address

**Bytes1 - Bytes32** A fixed-size byte array

**Bytes** A dynamically-sized byte array

**String** A dynamically signed string (Strings are costworthy on the Ethereum blockchain, and when possible, you should actually use type of bytes)

**Enum** A custom types with finite value sets

E.g. enum contractState { Created, Locked, Inactive}

**Mapping** Hash tables with key types and value types (examples in following slides)

**Struct** Structs allow you to define new types

# How Blockchain and smart contracts work

## Smart Contracts Functions

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.

# How Blockchain and smart contracts work

## Ethereum Events

Events in Solidity are a convenience interface for EVM logging facilities.

This makes them excellent for keeping track of what is happening with a contract

### Event Example

```
event Student( // creating event
    string name,
    uint age
);
function setStudent(string _ fName, uint
    _age) public {
    fName = _ fName;
    age = _age;
    Student(_ fName, _age); // Trigger event
}
```

# How Blockchain and smart contracts work

---

## New Ethereum Language

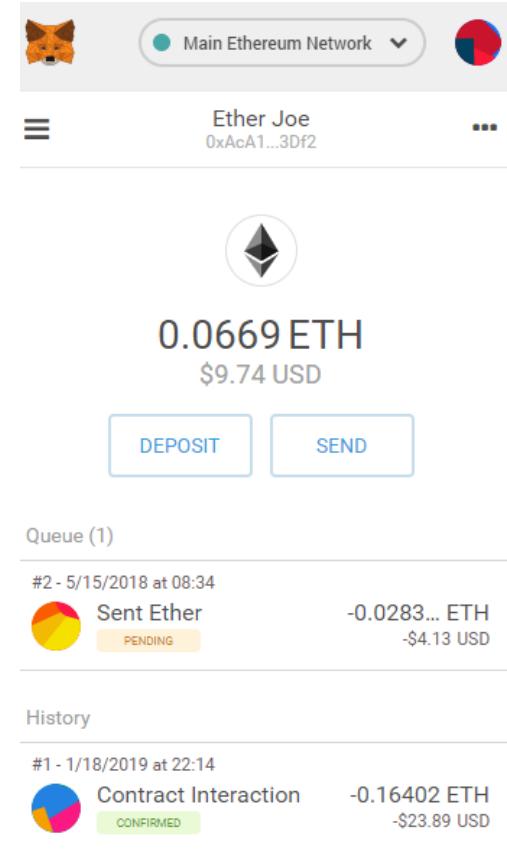
- Vyper is a general-purpose, experimental programming language that compiles down to EVM (Ethereum Virtual Machine) bytecode, as does Solidity
- Vyper is designed to massively simplify to create easier-to-understand Smart Contracts that are more transparent for all parties involved
- Vyper looks logically similar to Solidity, and syntactically similar to Python
- Vyper is still in v0.1.0-beta.1 as of June 2018
- <https://blockgeeks.com/guides/understanding-vyper/98>

# How Blockchain and smart contracts work

## Metamask

### Working with a web wallet – Metamask

1. Installing MetaMask – (Chrome Plug-in)
2. Creating new Wallet – Store mnemonic or seed safely
3. Switching between different accounts
4. Changing to different environments
5. Sending Ether – Explore about transaction confirmation
6. Funding your wallet – Test faucet
7. Learn about ERC20 token in Metamask



# How Blockchain and smart contracts work

View in Etherscan

→ ⌂ ⌂ https://etherscan.io/tx/0xa6253b26112a82ba88c75155f8f57096b4bf6cee43845722219b7b1fcf5611df

Etherscan

Eth: \$146.13 (-1.62%)

Search by Address / Txhash / Block / Token / ENS

Home Blockchain Tokens Resources More Sign In

Crypto Loan

### Transaction Details

Sponsored: 📺 Subscribe to The Week in Ethereum otherwise you're just pretending you know what's going on. [Subscribe here!](#)

Overview	
Transaction Hash:	0xa6253b26112a82ba88c75155f8f57096b4bf6cee43845722219b7b1fcf5611df
Status:	Pending
Block:	(Pending)
Time Last Seen:	0 days 00 hr 01 min 06 secs ago (Feb-20-2019 03:51:35 PM)
Estimated Confirmation Duration:	~ 1 min : 53 secs   <a href="#">?</a>
From:	0xaca1c75d20ca47d9b0dc2c9c7f545306609c3df2
To:	0x24be629886dd078d435a28f9af7d68a9ad9e356e
Value:	0.028383 Ether (\$4.15)
Max Txn Cost/Fee:	0.000168 Ether (\$0.02)

Click to see more ↓

Main Ethereum Network

Ether Joe  
0xAcA1...3Df2

0.0669 ETH  
\$9.74 USD

DEPOSIT SEND

Queue (1)

#2 - 5/15/2018 at 08:34  
 Sent Ether PENDING -0.0283... ETH -\$4.13 USD

History

#1 - 1/18/2019 at 22:14  
 Contract Interaction CONFIRMED -0.16402 ETH -\$23.89 USD

# How Blockchain and smart contracts work

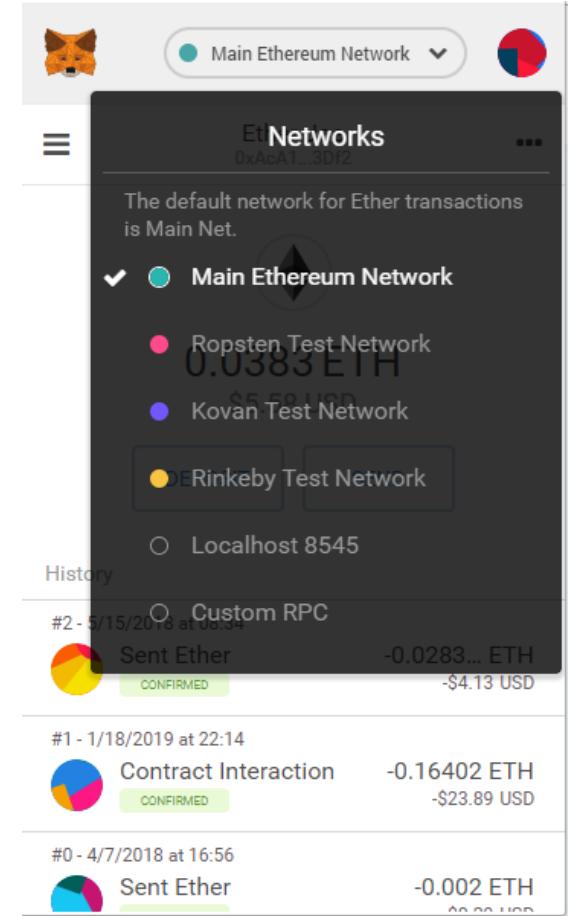
## Ethereum TestNets

To practice development Ethereum has some test networks.

These generally use the latest stable Ethereum code base.

- Rinkeby
- Kovan
- Ropsten

And you can use localhost or a custom RPC



# How Blockchain and smart contracts work

---

Remix is a browser-based IDE built by the Ethereum development team

- Open source tool to write Solidity contracts straight from the browser
- Written in Javascript, Remix supports both usage in the browser or locally
- Remix also supports testing and deploying of smart contracts
- Simply visit the online version at <http://remix.ethereum.org>
- Remix documentation: <https://remix.readthedocs.io/en/latest/>
- Remix alpha - version where new Remix features are release for testing

# How Blockchain and smart contracts work

---

## Smart Contracts Review

- 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.

# How to design, code, deploy and execute a smart contract

---

**How smart contracts work on both the Ethereum and Hyperledger platforms from a practical level**

# How smart contracts work on both the Ethereum and Hyperledger platforms from a practical level

---

## 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”

# How smart contracts work on both the Ethereum and Hyperledger platforms from a practical level

## Smart Contracts Legal Enforcement

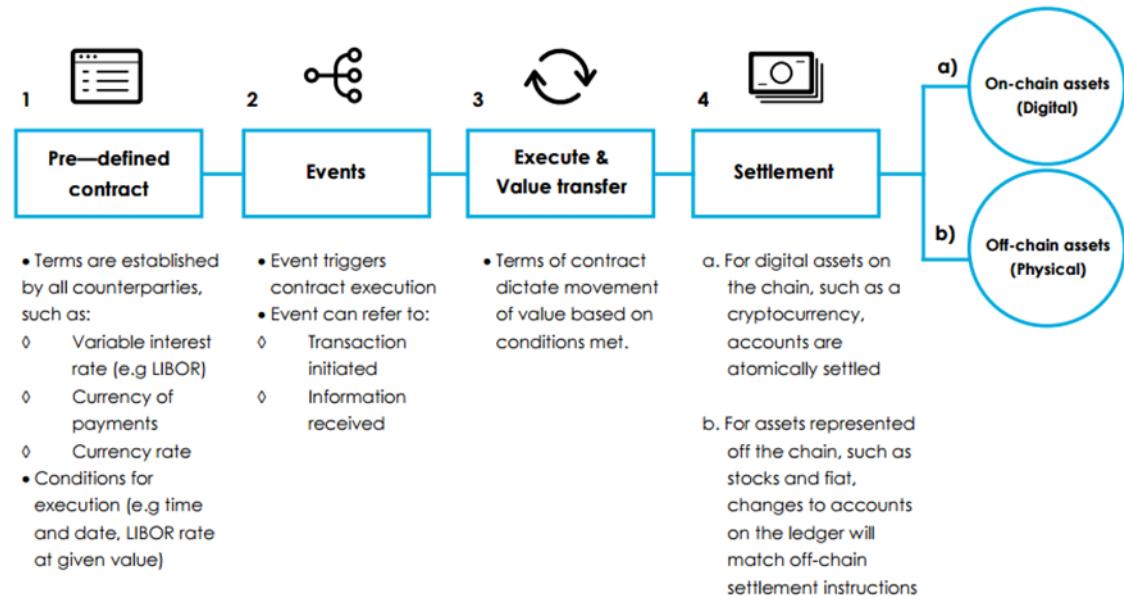
- Smart Contracts may not be legally enforceable. Especially across borders.
- Could be used as evidence
- Think of a vending machine where you put in the required funds to get the drinks or food...



# How smart contracts work on both the Ethereum and Hyperledger platforms from a practical level

## Smart Contracts Functions

- Constructor Function



## Smart Contract

Ethereum Account Type (Just like User Account)



Address  
Balance  
Code  
State

```
0x16E0022b17B...
0 Ether
contract Counter {
    uint counter;
    function Counter() public {
        counter = 0;
    }
    function count() public {
        counter = counter + 1;
    }
}
```

# How smart contracts work on both the Ethereum and Hyperledger platforms from a practical level

---

## Hyperledger Smart Contracts

- Smart Contracts = Chaincode in Hyperledger - A smart contract is code – invoked by a client application external to the blockchain network – that manages access and modifications to a set of key-value pairs in the World State
- Chaincode services are secured and lightweight.
- The environment is a “locked down” and is a secured container with a set of signed base images which contains secure OS and Chaincode language, runtime and SDK images for Golang, Java, and Node.js

# How smart contracts work on both the Ethereum and Hyperledger platforms from a practical level

---

## Hyperledger Smart Contracts

- Every Chaincode implements the Chaincode interface in particular, Init and Invoke functions.
- Init - is called during Instantiate transaction after the chaincode container has been established for the first time, allowing the chaincode to initialize its internal data.
- Invoke - is called to update or query the ledger in a proposal transaction. Updated state variables are not committed to the ledger until the transaction is committed

# How to design, code, deploy and execute a smart contract

---

**The constructs of smart contract, common coding requirements and demos**

# The constructs of smart contract, common coding requirements and demos

## Chaincode in Hyperledger

- Getting started- Writing chaincode you will want to make sure that you have the Go programming language installed and correctly configured.
- Make sure that a directory is created for your chaincode application as a child

```
// Extract the function and args from the transaction proposal
fn, args := stub.GetFunctionAndParameters()
var result string
var err error
if fn == "set" {
    result, err = set(stub, args)
} else {
    result, err = get(stub, args)
}
if err != nil {
    return shim.Error(err.Error())
}
// Return the result as success payload
return shim.Success([]byte(result))
}
```

# The constructs of smart contract, common coding requirements and demos

## Chaincode in Hyperledger

- We can then implement the init function. Init is called during chaincode instantiation, and it will initialize any data.

Chaincode:

```
// Init is called during chaincode instantiation to initialize any data.  
func (t *SimpleAsset) Init(stub  
shim.ChaincodeStubInterface)  
peer.Response  
{  
}
```

# The constructs of smart contract, common coding requirements and demos

## Chaincode in Hyperledger

- Now our example chaincode application implements two functions that can be invoked via the invoke function.

```
// Set stores the asset (both key and value)  
on the ledger. If the key exists,  
// it will override the value with the new one  
func set(stub shim.ChaincodeStubInterface,  
args []string) (string, error) {  
if len(args) != 2 {  
return "", fmt.Errorf("Incorrect arguments.  
Expecting a key and a value")  
}  
err := stub.PutState(args[0], []byte(args[1]))  
if err != nil {  
return "", fmt.Errorf("Failed to set asset: %s",  
args[0])  
}  
return args[1], nil  
}
```

# How to design, code, deploy and execute a smart contract

---

**What are the most in demand Blockchain Certifications?**

# What are the most in demand Blockchain Certifications?

## Certified Blockchain Solutions Architect (CBSA)

- The Certified Blockchain Solution Architect (CBSA) exam is an elite way to demonstrate your knowledge and skills in the Blockchain arena.
- You will become a member of a community of Blockchain leaders.
- <https://blockchaintrainingalliance.com/products/cbsa>



# What are the most in demand Blockchain Certifications?

## Certified Blockchain Solutions Architect (CBSA)

- Discount Code – Save 30% - JH30QS

The screenshot shows a checkout process for the Certified Blockchain Solution Architect (CBSA) certification. The left side displays payment options (PayPal, Google Pay) and contact information fields (Email, newsletter checkbox). The right side shows the product details, a discount code input, and the final total after applying the discount.

Blockchain Training Alliance

Cart > Customer information > Payment method

Express checkout

PayPal Google Pay

Contact information

Email

Keep me up to date on news and exclusive offers

OR

Discount code  Apply

1 Certified Blockchain Solution Architect (CBSA)  
Certified Blockchain Solution Architect (CBSA) - \$300 USD

Subtotal \$300.00

Discount JH30QS - \$90.00

Total USD \$210.00

# What are the most in demand Blockchain Certifications?

## Certified Blockchain Professional (CBCP)

- The Certified Blockchain Professional (CBCP) certification is designed for professionals currently working in blockchain, Bitcoin and cryptocurrency roles, or those who wish to understand and work with these technologies, level up their professional skills and certify their level of competency and expertise.
- <https://blockchaininstituteoftechnology.com/certifications/certified-blockchain-professional/>



# What are the most in demand Blockchain Certifications?

## IBM Blockchain Essentials

- This badge earner has developed an understanding of Blockchain principles and practices and how they can be applied within a business environment. They have an understanding of Blockchain and distributed ledger systems, the important concepts and key use cases of Blockchain and how assets can be transferred in a Blockchain network. <https://www-03.ibm.com/services/learning/ites.wss/zen?pageType=badges&id=00aa7abf-f310-47c3-96a3-26eedb323b56>



# How to design, code, deploy and execute a smart contract

---

**How do these certification meet the needs of todays Enterprises?**

# How do these certification meet the needs of todays Enterprises?

Enterprises are constantly challenged for blockchain skills. Enterprises require employees that have at least a base knowledge in blockchain.

- Certifications from Blockchain Training Alliance(BTA), Blockchain Institute of Technology(BIT) and IBM provide most current content and certifications.
- Developed by experts in the field.
- Enterprise focused Certifications.



# How to design, code, deploy and execute a smart contract

---

## **What about Blockchain Career Demand?**

# Blockchain Careers

---

## Blockchain Careers

- Blockchain Developer
- Blockchain Architect
- Software Developers (PHP, Solidity, Python, etc.)
- Solutions Architect
- Sales Engineer
- Blockchain Analyst

# Blockchain Careers

---

## Blockchain Careers

Demand in Blockchain expertise:

- Demand is growing for developers and architects
- Pre-sales and Post-sales experience desired
- Multiple stages of Blockchain adoption drive demand
- Industries such as finance, retail, supply chain, and healthcare

# Blockchain Careers

---

## Blockchain Careers

- LinkedIn has over 4100 roles listed in the US and over 1406 in the United Kingdom
- Cryptojobslist has over 60 roles listed worldwide.
- Demand is growing and now is a great time to learn more about the blockchain.

# Blockchain Courses on Quickstart

---

**Blockchain Course on Quickstart**

# Blockchain Courses on Quickstart

## Blockchain Courses on Quickstart

- Enterprise Blockchain Bootcamp
- <https://www.quickstart.com/enterprise-blockchain-bootcamp-for-solutions-engineers.html>

The screenshot shows the Quickstart website with the URL https://www.quickstart.com/enterprise-blockchain-bootcamp-for-solutions-engineers.html. The page title is "Enterprise Blockchain Bootcamp for Solutions Engineers". It features a "Self-Paced Learning" section with "Beginner Difficulty" and "7 Hours Course Duration". There are two main purchase options: "BUY INDIVIDUALLY" at \$30.00/Each and "BUY AS SUBSCRIPTION" at \$99.00. Below these are sections for "About this course" and "Course Information". On the right side, there's a sidebar for "Get A Team Quote or Got Questions?" with fields for First Name, Last Name, Phone Number, Email Address, and a comment area.

# Blockchain Course on Quickstart

---

Coming Soon to the Quickstart Platform

- Intro to Blockchain Technology – March 2019
- Blockchain Architecture Fundamentals – March 2019
- Hyperledger Fabric Crash Course - March 2019
- Certified Blockchain Solutions Architect March 2019
- R3 Corda Blockchain Basics – April 2019
- Smart Contract Programming – April 2019

# Thank you

---

- Questions
- Thank you





## **BONUS MATERIALS**

### **What is a Blockchain**

# Blockchain Fundamentals

---



**What exactly is Blockchain technology?**

# What Exactly is Blockchain Technology

---

## 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 Exactly is Blockchain Technology

A Blockchain essentially is a record of transactions similar to a traditional ledger. These transactions can be recorded for anything and not just financial

- Blockchain is NOT Bitcoin
- Blockchain is NOT just about Cryptocurrency
- Blockchain is to Bitcoin, what the internet is to email



# What Exactly is Blockchain Technology

A distributed ledger is a database that is stored and updated independently by each 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.

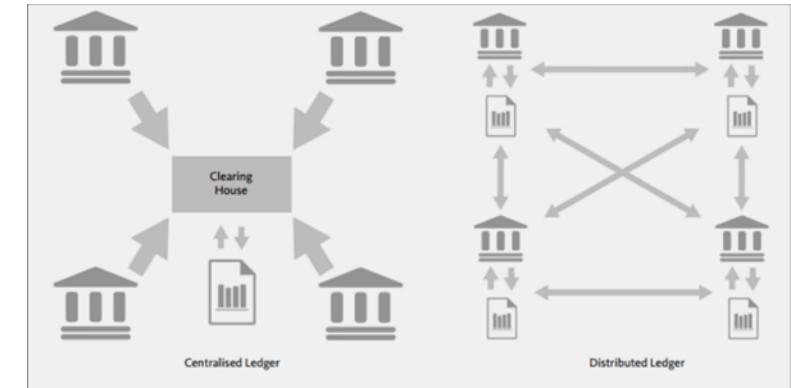


Image – Medium.com

# What Exactly is Blockchain Technology

## What is a Cryptocurrency?

- Digital currencies are secured using cryptography and combining that with their role as a currency.
- Are mined. Not printed.
- Considered digital gold, silver..
- Bitcoin is the most widely known and has the largest market cap.



# What Exactly is Blockchain Technology

## Lets Discuss Bitcoin Briefly

- Bitcoin(BTC) was created by a discrete creator named Satoshi Nakamoto in 2009. To this day no one knows who he or she is.
- Bitcoin now has a fork called Bitcoin Cash.
- Bitcoin is known as “crypto gold”



# What Exactly is Blockchain Technology

---

Blockchain is evolving

- Bitcoin was released in Jan 2009 by Satoshi Nakamoto
- Blockchain technology then enabled cryptocurrencies, inspired by distributed ledgers, asymmetric encryption and Merkle trees technologies
- Altcoins for specific use cases = Dogecoin, Ethereum, etc (2011-2017)
- Enterprises explore potential benefits of the technology (Hyperledger, Corda and Quorum)

# What Exactly is Blockchain Technology

Built from these technologies.

- P2P Networks
- Private Key Encryption
- Programs

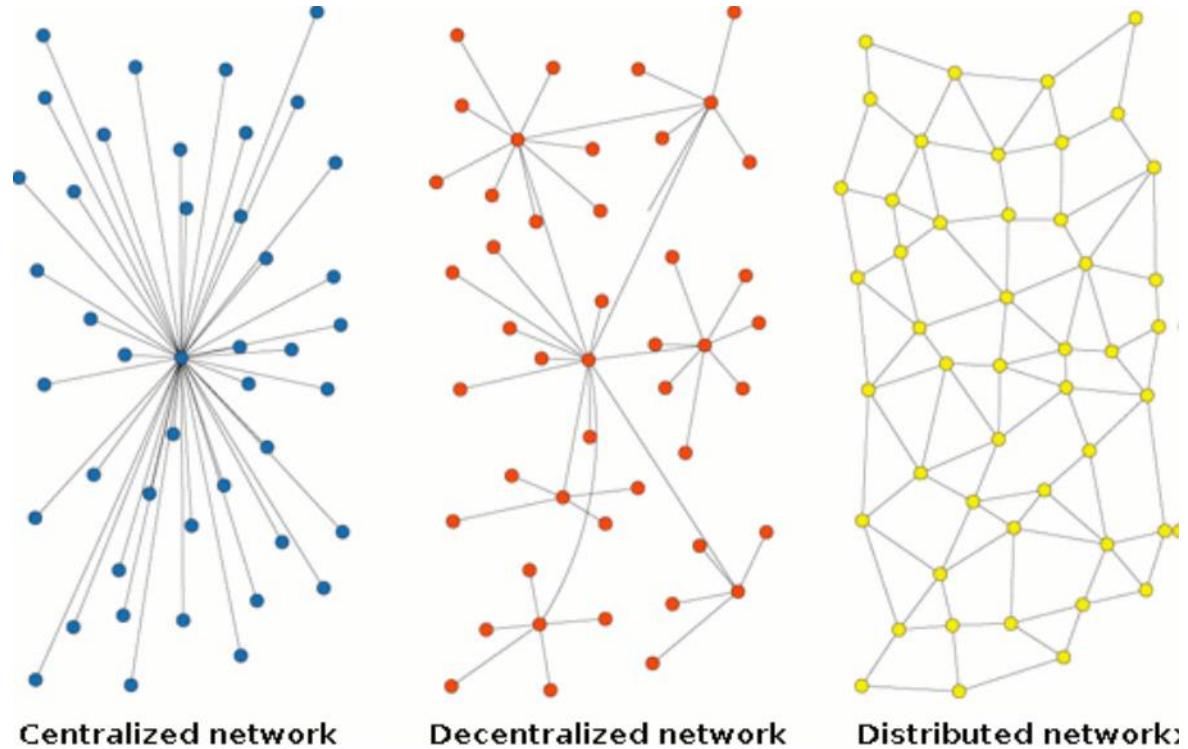


Diagram - Wallstreet Technologist

# What Exactly is Blockchain Technology

Digital Identity is established

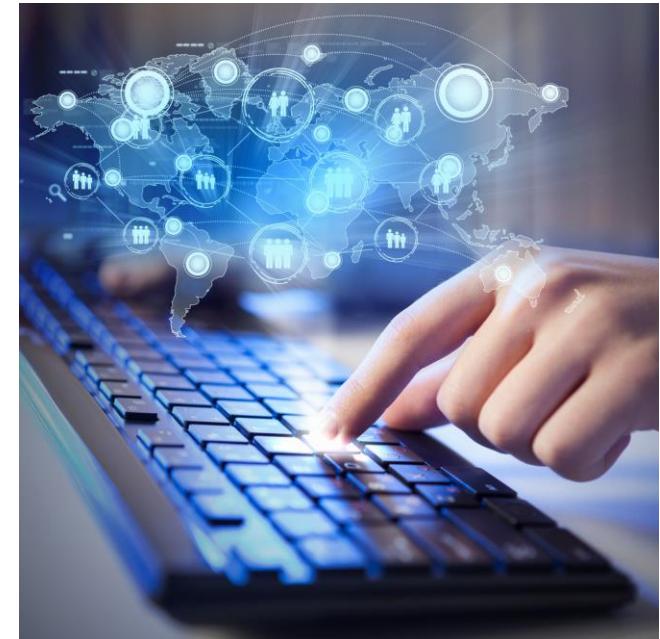
- Combining a public and private key creates a strong digital identity reference based on possession.
- Private Key
- Public Key



# What Exactly is Blockchain Technology

Blockchain is revolutionary in several ways

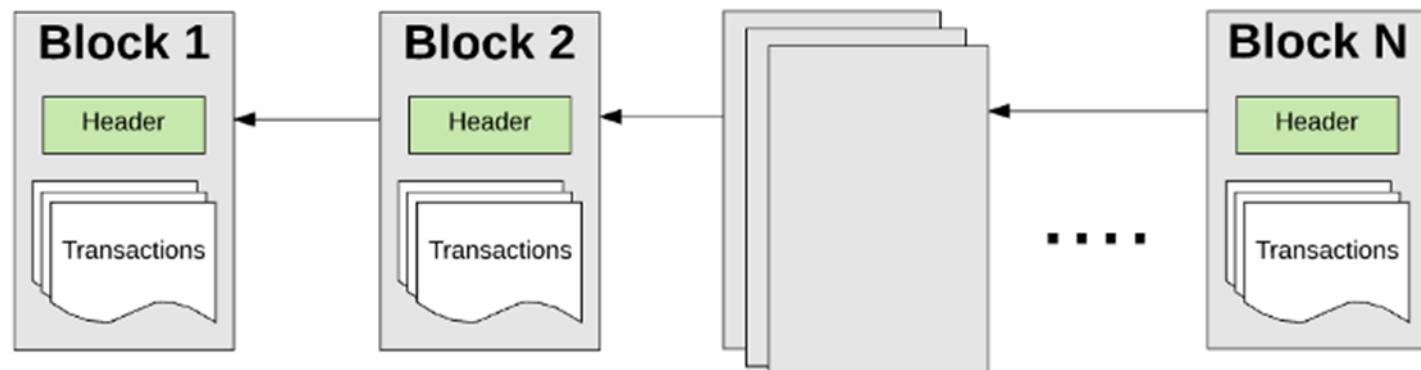
- Blockchain is not new technology but a synching of technologies that now make sense.
- Trust is at the center and essentially removes intermediaries. (efficiency)
- Tamperproof public ledger of value.
- Disruptive to the status quo. Legacy is out
- Platform with numerous use cases



# What Exactly is Blockchain Technology

## Transactions in Blockchain

- In a Blockchain 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.



# Smart Contracts

---

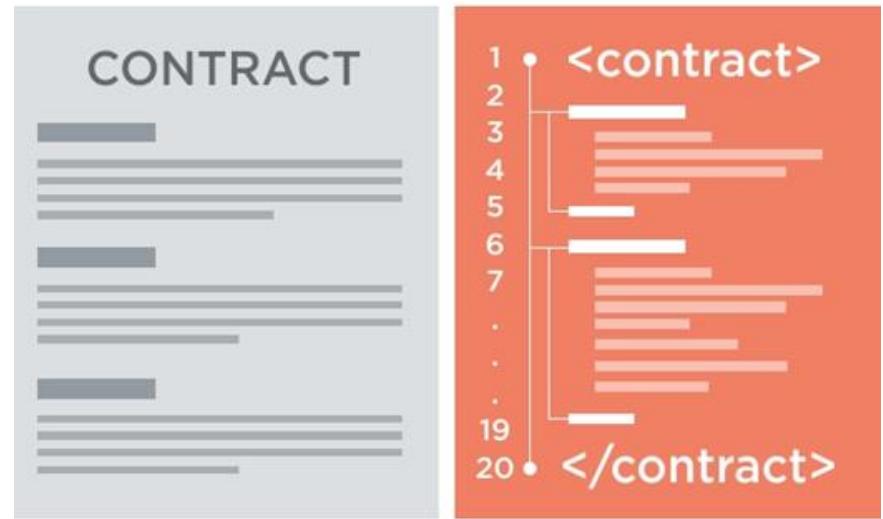
Smart Contract is a term to describe computer program code that is capable of facilitating, executing, and enforcing the negotiation or performance of an agreement using Blockchain technology. (Contract)

- 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

- 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



# Blockchain Fundamentals

---



**Why are companies are embracing Blockchain technologies?**

# Why are companies are embracing Blockchain technologies?

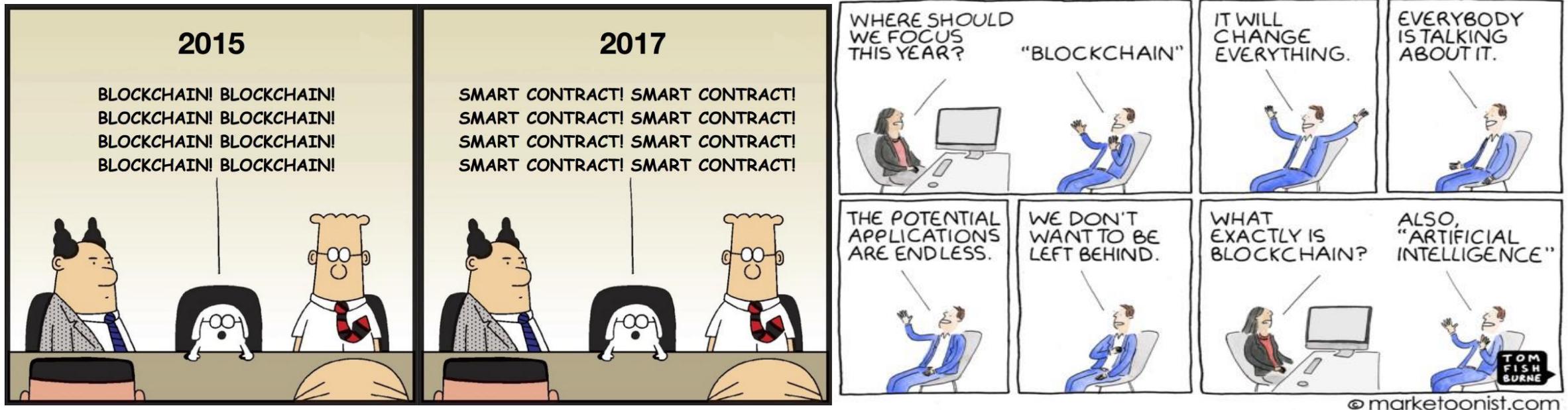
## Blockchain and Change

- Enterprises require innovation
- Enterprises require efficiencies
- Enterprises must adapt or fail
- Provides new market opportunities
- Adapt to ever changing business needs
- Meet Compliance demands



# Why are companies are embracing Blockchain technologies?

- Some Blockchain Humor



# Why are companies are embracing Blockchain technologies?

## Blockchain and Change

- In PwC's 2018 survey of 600 executives from 15 territories, 84% say their organizations have at least some involvement with blockchain technology.



**pwc**

<https://www.pwc.com/blockchainsurvey>

# Why are companies are embracing Blockchain technologies?

---

## Blockchain and Change

- The business value-add of blockchain will grow to slightly more than \$176 billion by 2025, and then it will exceed \$3.1 trillion by 2030.
- <https://www.gartner.com/doc/3627117/forecast-blockchain-business-value-worldwide>

**Gartner®**

# Blockchain Fundamentals

---



## **Overview of major Enterprise Blockchains**

# Enterprise Blockchains

Lets Discuss the most common smart contract enterprise blockchains



# Enterprise Blockchains

## Ethereum

- Most widely used Open Source Blockchain-based distributed computing application platform
- Mainly used for 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



# Enterprise Blockchains

## Ethereum

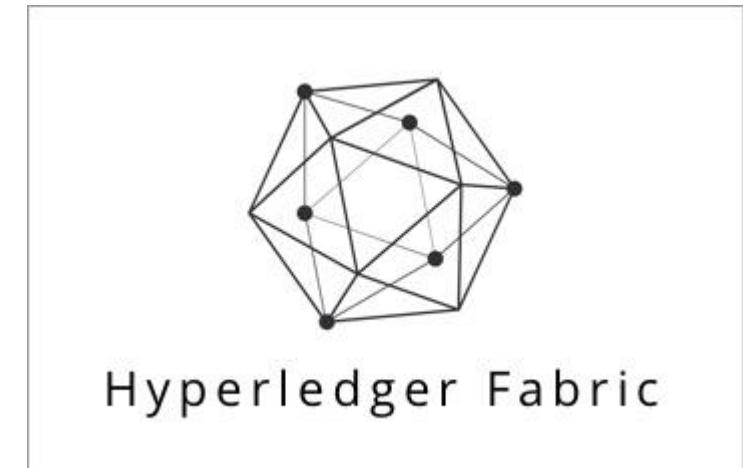
- Ether is the native token of the Ethereum blockchain which is used to pay for transaction fees, miner rewards, and other services on the network.
- Ethereum is an open software platform based on blockchain technology that enables developers to write smart contracts and build and deploy decentralized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



# Enterprise Blockchains

## Hyperledger

- Hyperledger is an open source project that came out of the Linux Foundation and was created in order to help advance cross-industry blockchain technologies.
- It is essentially a global open source collaboration involving leaders from numerous industries.



# Enterprise Blockchains

## Hyperledger Blockchains

The Framework Module includes

- Hyperledger Indy
- Hyperledger Fabric
- Hyperledger Iroha
- Hyperledger Sawtooth
- Hyperledger Burrow



# Enterprise Blockchains

## R3 Corda

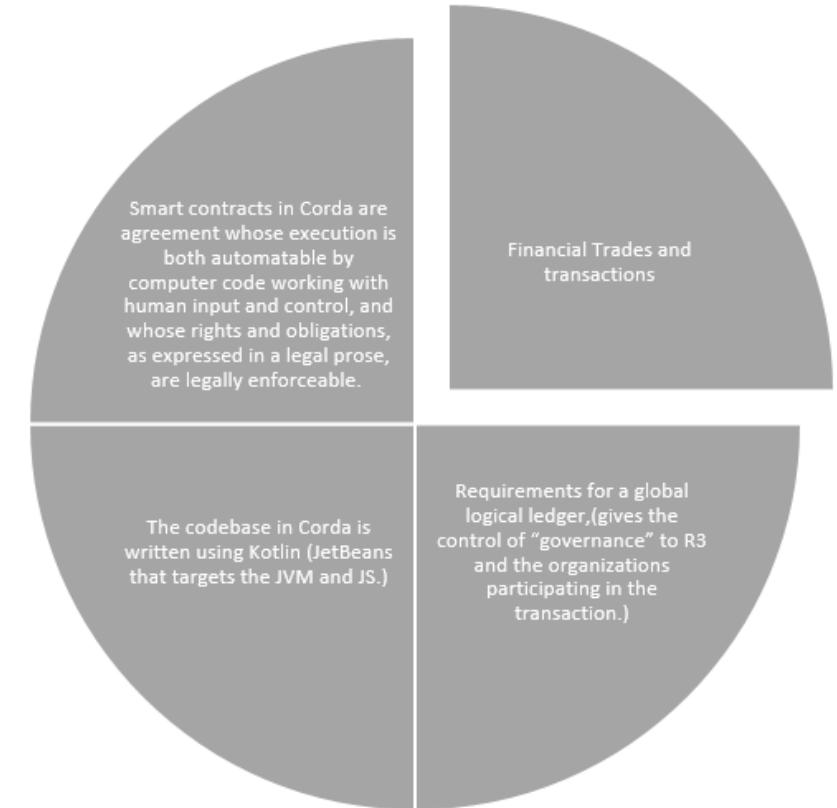
- R3 is an enterprise blockchain software firm working with a broad ecosystem of more than 200 members and partners across multiple industries from both the private and public sectors to develop on Corda, our open-source blockchain platform, and Corda Enterprise, a commercial version for enterprise usage.
- Corda removes costly friction in business transactions by enabling institutions to transact directly using smart contracts, while ensuring the highest levels of privacy and security.



# Enterprise Blockchains

## R3 Corda

- R3 Corda aka “Corda” and was developed in 2016
- Corda was first built to record, manage and automate financial agreements.
- Corda was developed specifically for banks with some limited use cases outside of that area.



# Enterprise Blockchains

## Quorum

- Quorum is ideal for any application requiring high speed and high throughput processing of private transactions within a permissioned group of known participants.
- Quorum addresses specific challenges to blockchain technology adoption within the financial industry, and beyond.
- Quorum was built on the Ethereum Platform and sponsored by JPMorgan Chase



# Enterprise Blockchains

## Quorum

- Quorum is an Ethereum side chain
- Quorum has benefits over Ethereum such as Privacy and Consensus Choices.
- Quorum supports two distinct Consensus Algos RAFT and IBFT



Quorum two distinct advantages.



**Privacy** — Quorum supports private transactions and private contracts through public/private state separation and utilizing Constellation (P2P Messaging)



**Alternative Consensus Mechanisms** — Quorum allows for multiple consensus mechanisms that are more appropriate for consortium chains.

# Enterprise Blockchains

## Transactions Per Second (TPS)

	VISA	BTC	Ripple	Paypal	Hyperledger
TPS	24,000	4	1500	193	3500**
Control	Centralized	Decentralized	Centralized	Centralized	Centralized
Notes	Push	P2P	Hybrid	Hybrid	P2P

\*\*Hyperledger Fabric. As Chris Ferris [noted](#) “We haven’t published performance figures for Fabric because there isn’t a standard benchmark.”.

# Enterprise Blockchains

Lets Compare

	Ethereum	R3 Corda	Quorum	Hyperledger	Ripple
Industry	Cross-Industry	Financials	Cross- Industry	Cross Industry	Financial
Ledger	Permission-less	Permissioned	Permissioned	Permissioned	Permissioned
Consensus	Proof of Work (PoW)	Pluggable	Majority Voting	Pluggable	Probabilistic Voting
Smart Contract Support	Yes	Yes	Yes	Yes	No
Cryptocurrency	Ether (Eth)	N/A	N/A	N/A	Ripple (XRP)