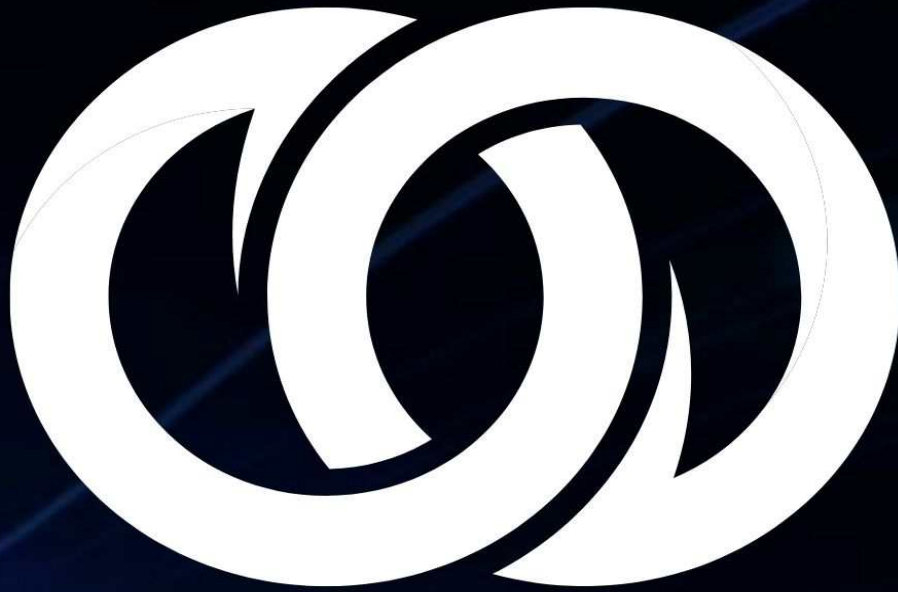


SMART CONTRACTS



COMPLETE GUIDE TO TECH AND CODE THAT
IS ABOUT TO TRANSFORM THE ECONOMY

PATRICK EJEKE

Smart Contracts

What Is A Smart Contract? Complete Guide To Tech And Code That Is About To Transform The Economy-Blockchain, Web3.0, DApps, DAOs, DEFI, Crypto, IoTs, FinTech, Digital Assets Trading

By

Patrick Ejeke

You also might be interested in my BEST SELLER, "[METAVERSE](#)" Click the links below...

[Grab A Free PDF Version](#) OR [Get it on Amazon](#)

© Text Copyright Patrick Ejeke 2022 - All Rights Reserved

All rights reserved. No part of this guide may be reproduced in any form without permission in writing from the publisher except in the case of brief quotations embodied in critical articles or reviews.

Legal & Disclaimer

The information contained in this book and its contents is not designed to replace or take the place of any form of medical, financial, or professional advice; and is not meant to replace the need for independent medical, financial, legal, or other professional advice or services, as may be required. The content and information in this book have been provided for educational and entertainment purposes only.

The content and information contained in this book have been compiled from sources deemed reliable, and it is accurate to the best of the Author's knowledge, information, and belief. However, the Author cannot guarantee its accuracy and validity and cannot be held liable for any errors and/or omissions. Further, changes are periodically made to this book as and when needed. Where appropriate and/or necessary, you must consult a professional (including but not limited to your doctor, attorney, financial advisor, or such other professional advisor) before using any of the suggested remedies, techniques, or information in this book.

Upon using the contents and information contained in this book, you agree to hold harmless the Author from and against any damages, costs, and expenses, including any legal fees potentially resulting from the application of any of the information provided by this book. You agree that the author is not liable to any loss, injury, or damage sustained in any way. This disclaimer applies to any loss, damages, or injury caused by the use and application, whether directly or indirectly, of any advice or information presented, whether for breach of contract, tort, negligence, personal injury, criminal intent, or under any other cause of action.

You agree to accept all risks of using the information presented inside this book.

You agree that by continuing to read this book, where appropriate and/or necessary, you shall consult a professional (including but not limited to your doctor, attorney, or financial advisor, or such other advisor as needed) before using any of the suggested remedies, techniques, or information in this book.

Table Of Contents

Introduction

[The Problem With Contracts](#)

[The Smart Solution](#)

[Distinctive Properties](#)

[What You Need to Know](#)

What Is A Smart Contract?

[Blockchain and Smart Contracts](#)

[Vitalik Buterin On Smart Contracts](#)

[Digital and Real-World Applications](#)

How Smart Contracts Work

[Smart Contracts' Historical Background](#)

[A definition of Smart Contracts](#)

[The promise](#)

[What Do All Smart Contracts Have in Common?](#)

[Elements Of Smart Contracts](#)

[Characteristics of Smart Contracts](#)

[Capabilities of Smart Contracts](#)

Life Cycle Of A Smart Contract

[Why Are Smart Contracts Important?](#)

How Do Smart Contracts Work?

[What Does Smart Contract Code Look Like In Practice?](#)

[The Structure of a Smart Contract](#)

[Interaction with Traditional Text Agreements](#)

Are Smart Contracts Enforceable?

[Challenges With the Widespread Adoption of Smart Contracts](#)

[Non-Technical Parties: How Can They Negotiate, Draft, and Adjudicate Smart Contracts?](#)

Smart Contracts and the Reliance on “Off-chain” Resources

[What is the "Final" Agreement Reached by the Parties?](#)

[The Automated Nature of Smart Contracts](#)

Are Smart Contracts Reversible?

[Smart Contract Modification and Termination](#)

[The Difficulties of Integrating Specified Ambiguity Into Smart Contracts](#)

[Do Smart Contracts Really Guarantee Payment?](#)

[Allocation of Risk for Attacks and Failures](#)

[Governing Law and Location](#)

[Best Practices for Smart Contracts](#)

[Types Of Smart Contracts](#)

[A Technical Example of a Smart Contract](#)

[Smart Contract Use-Cases](#)

[Smart Contracts in Action](#)

[Smart Contracts and Blockchains In the Automobile Industry](#)

[Smart Contracts and Blockchains in Finance](#)

[Smart Contracts and Blockchains In Governments](#)

[Smart Contracts And Blockchains In Business Management](#)

[Smart Contracts and Blockchains in Initial Coin Offerings \(ICOs\)](#)

[Smart Contracts and Blockchains In Rights Management \(Tokens\)](#)

[Smart Contracts And Blockchains In NFTs - Gaming Technology](#)

[Smart Contracts and Blockchains in the Legal Industry](#)

[Smart contracts and Blockchains in Real Estate](#)

[Smart Contracts and Blockchains in Corporate Structures - Building DAOs](#)

[Smart Contracts and Blockchains in Emerging Technology](#)

[Smart Contracts and Blockchains In Insurance Companies](#)

[Smart Contracts and Blockchains in Finance](#)

[Smart Contracts And Blockchains In Powering DEFI](#)

[Smart Contracts and Blockchains In Healthcare](#)

[Smart Contracts and Blockchains In Other Industries](#)

[What Smart Contracts Can Give You](#)

[How Are Smart Contracts Created?](#)

[Make Your Very Own Smart Contract!](#)

[Are Smart Contracts Secure?](#)

[Why Should You Have Faith In Smart Contracts?](#)

[Blockchain Networks Using Smart Contracts](#)

[What Are Ethereum Contracts And How Do They Work?](#)

[Other Platforms And Initiatives](#)

[What Is Hybrid Smart Contracts?](#)

[How Oracles Are Extending Blockchain Collaboration](#)

[The Structure of Hybrid Smart Contracts](#)

[How Do Hybrid Smart Contracts Combine On- and Off-Chain Computation?](#)

[Chainlink Decentralized Services That Power Hybrid Smart Contracts](#)

[What Hybrid Smart Contracts Mean for Global Industries](#)

[Interoperability and Connectivity: Unlocking Smart Contracts 3.0](#)

[A DLT Stack That Isn't Connected](#)

[Creating TCP/IP and HTTP Equivalents for DLT](#)

[Chainlink: A General Purpose Communication Standard](#)

[The Beginnings of Connected Consensus](#)

[Future of Smart Contracts](#)

[Smart Contract Advantages](#)

[Smart Contracts Are Not Perfect](#)

[Wrapping Up](#)

[Smart Contracts FAQs](#)

Introduction

What exactly is a smart contract? What distinguishes it from a standard contract? What are its applications? Let's find answers to these questions shall we?

Right now, if you want to get anything important done, you'll almost certainly need a contract.

This is a legally enforceable agreement that can cover almost anything, including property transfers, ordinary employment contracts, non-disclosure agreements, licensing contracts, and more.

Most of us are already accustomed to dealing with contracts; most of our energy bills are contracts, and we make contracts with our banks and employers, as well as many of the services we use—because all of those 'terms of use' agreements we sign are legally enforceable contracts.

They're dense, tedious to read, and boring, yet they're a crucial element of the financial world, and we can't avoid them if we want to interact securely.

But, owing to the advent of a new sort of self-executing digital contract known simply as a 'smart contract,' the way we construct and deal with contracts is starting to change.

The Problem With Contracts

Contracts are often written documents that outline who performs what, when, and under what terms. In the event of a disagreement, the contract provisions are commonly applied to resolve it via an arbitrator or mediator, but in rare situations, a court of law may be required to intervene to settle disputes.

Even though contracts are so common in today's culture, they virtually all have the same set of problems: they're slow to establish, difficult to enforce, and often subjective—some are even created to be purposefully ambiguous or deceptive... how frustrating!

Disputes

If there is a problem with a contract, resolving it might be a nightmare.

It might take weeks, months, or even years to know the outcome of a disagreement, which is not only inconvenient but also costly.

Given that resolving a disagreement in this manner might cost thousands of dollars, it is cost-prohibitive to enforce contracts for low-value agreements.

What good is a contract if it can't be easily enforced?

Third-parties

Contracts are often enforceable only with the participation of other parties.

After all, if you have a contract to purchase a property at an agreed-upon price and the seller breaches the agreement, you'll have to go through the right legal processes to have it resolved—and you'll most likely have to pay middlemen in the process.

Contract issues are seldom resolved "mano a mano," as the saying goes. That just will not do.

Forgery

Contracts may be faked, altered, or destroyed—all of which are very impractical or difficult to establish.

When there are no copies, the contract becomes null and void, and there have been instances when individuals have falsified contracts and gotten away with it—at least temporarily.

Overall, they have a slew of restrictions, but change is on the way!

The Smart Solution

If contracts are supposed to make our lives easier, then, why are they so hard to enforce? Surely there must be a better way...

It turns out that there simply could be. Smart contracts are affecting not just the financial sector, but almost every business under the sun.

Smart contracts, in their most basic form, are snippets of code that are used to automatically execute an agreed-upon set of conditions.

A smart contract, like any other contract, is used to guarantee that everyone participating in an agreement understands what is expected of them and that all parties meet their responsibilities.

They're constructed using blockchain, the same technology that powers Bitcoin and Ethereum, and they're changing the way we think about contracts and how they're enforced.

The concept of smart contracts comes up often while discussing [Web3](#), blockchain technology, non-fungible tokens (NFTs), or bitcoin.

Smart contracts are believed to be self-executing, less costly, dependable, transparent, secure, and borderless, and to rely less on attorneys and legal teams than conventional contracts.

Traditional Contracts



Smart Contracts



This book will explain what a smart contract is and why brands should be aware of them.

Many platforms and apps are created using blockchain or distributed ledger technology including "smart contracts." I'll review the history and purposes of smart contracts, examine whether they may be called enforceable legal agreements under US contract law, and emphasize important legal and practical issues that must be addressed before they can be widely employed in commercial situations.

Distinctive Properties

Smart contracts have several unique qualities that make them especially intriguing since they are developed on top of very secure decentralized networks like Ethereum.

For starters, they are tamper-proof. This implies that once a smart contract has been formed, it is very hard to change it without leaving a visible trace.

They aren't also subject to interpretation. There's no technical language to dig through, no hidden conditions or addendums to deal with. Code is law in the case of smart contracts. Whatever is mentioned in the contract will be followed—no ifs, and, or buts!

Away With The Middleman

These contracts, as the name implies, are 'smart,' which means they don't need any middlemen to carry out their tasks.

This is a very cost-effective and time-efficient method of getting things done. After all, middlemen and other intermediaries are sometimes expensive, time-consuming, and may cause significant delays in settling issues.

All of this is no longer an issue with smart contracts since there is nothing to interpret or dispute.

Data Protection

At present, as of the time of writing this book, if you want to engage in a contract with someone, you'll almost certainly need to provide some private information, such as your identification and address data (and perhaps more).

This isn't usually necessary with smart contracts. It is perfectly feasible to engage in contracts anonymously while maintaining the same degree of security. This is because they are trustless—you don't have to trust the other person since the smart contract will do it for you.

Overall, smart contracts provide an altogether new level of privacy while also assuring that your personal information is never sold, exploited, or mismanaged! More on these later in the coming sections.

Smart Contracts for CTOs, CMOs, and Other Leaders

[Understanding Web3](#) — and, with it, smart contracts — entails being familiar with:

- [Cryptocurrency](#)
- Digital currency wallets (Crypto wallets)
- [Decentralized applications \(dApps\)](#)

- [The metaverse](#)
- Virtual reality (VR)
- Augmented reality (AR)

Nobody expects CTOs, CMOs, or other leaders to be programmers, web developers, network engineers, software developers, or crypto experts — but they should understand these technologies and how people use them because smart contracts can have an impact on customers, employees, and supply chain vendors.

Many people may wonder how brands can utilize smart contracts in ways other than Web3 initiatives. Forward-thinking executives envisage smart contracts being used by businesses as part of loyalty rewards programs, for selling NFTs (such as Taco Bell, Louis Vuitton, Nike, and Tony Hawk, to name a few), and to ease interactions between content providers, goods, and customers.

"Marketers can undoubtedly utilize smart contracts to enhance brand visibility, and the possibilities are exciting," said Tal Lifshitz, a partner, and co-chair of Miami-based [Kozyak Tropin & Throckmorton's](#) cryptocurrency, digital asset, and blockchain division.

"Imagine, for example, a smart contract that might cause an ad to play only if specific circumstances were satisfied, like as the number of viewers on a stream or the number of clicks on a website, so maximizing views." "I believe all marketers are interested in it."

What You Need to Know

- Smart contracts are self-executing lines of code that automatically verify and execute the conditions of a buyer-seller agreement over a computer network.
- Nick Szabo, an American computer scientist who designed the virtual currency "Bit Gold" in 1998, described smart contracts as computerized transaction protocols that carry out contract conditions. More on Nick later.
- When smart contracts are put on blockchains, transactions become traceable, transparent, and permanent.

What Is A Smart Contract?

Technology keeps improving and changing fast every year, and even those who aren't aware of blockchain have probably heard of Ethereum. Ethereum is the second-largest cryptocurrency, with a market capitalization of more than \$44 billion. To clearly understand Ethereum, one must first understand what a smart contract is.

Ethereum is more than simply a [cryptocurrency](#); it is also a platform on which other blockchain applications may be developed. To pay for transactions on the Ethereum network, a currency called Ether is used.

The Ethereum blockchain operates similarly to the Bitcoin blockchain in that a network of computers (or nodes) runs software that verifies network transactions.

Ether functions more like fuel than a traditional coin. Ether is required to operate the smart contracts and apps on the Ethereum blockchain in the same way that fuel or diesel is required for your automobile.

Because of the rise in popularity of Ethereum (which can be observed by visiting any crypto exchange site), the question 'what is a smart contract?' has recently been one of the most often asked in the crypto industry.

A smartphone, as we all know, is a mobile phone with extra capabilities such as game applications, GPS, online surfing, chat features, and so on. But, given that a contract is only words on paper or in digital form, what exactly is a "smart contract?"

A smart contract is a self-executing digital agreement that allows two or more parties to trade money, property, shares, or anything else of value in a transparent, conflict-free manner without the need of a third party.

To put it simply, smart contracts may be compared to a vending machine for complicated transactions.

Normally, if you had a complicated transaction involving large sums of money, you would go to a lawyer or a notary, have them set up an escrow account, pay them, and wait while they execute the job and guarantee the contract requirements are met. Only once your lawyer has completed the necessary steps to guarantee that everything is carried out appropriately will you be given the document/goods/money, etc.

When you use smart contracts, you just put a bitcoin into the vending machine (i.e. ledger), and your escrow, deed, contract, products, driver's license, or whatever the contract is for is automatically deposited into your account.

The smart contract undertakes all of the work to assess if the order's requirements were met.

Smart contracts set the rules and punishments that govern an agreement in the same way that conventional contracts do, and they also automatically enforce those rules and penalties.

So, where can you find smart contracts? Where are they executed? And how do they function?

They are computer software that operates on a blockchain network that verifies, executes, and enforces smart contracts. When both parties to the smart contract agree on its conditions, the program will run automatically. As the contract is confirmed and enforced by the blockchain network, there is no need for a third party.

Smart contracts, since they are implemented by code rather than humans, eliminate the potential for a human mistake and may automate numerous operations that would otherwise need human contact.

One of the most advantageous aspects of the blockchain is that, since it is a decentralized system that exists between all authorized participants, there is no need to pay intermediaries (middlemen), saving you time and conflict.

Smart contracts, like ordinary contracts, include the conditions of a deal. The former's terms, on the other hand, are established and implemented as code running on a blockchain. Smart contracts may also be transmitted and received without the requirement for a "trusted intermediary," such as a bank, government organization, company, or person, allowing practically any kind of agreement to be safely automated and decentralized. A blockchain ensures the dependability, security, and transnational accessibility of smart contracts.

That's because, in the case of a smart contract, the code specifies the transaction's processes and serves as the ultimate arbitrator of the terms. As a result, smart contracts have become the foundation of a whole ecosystem of [decentralized apps \(dApps\)](#) and a significant focus of blockchain development in general.

A single smart contract can only be used for one sort of transaction: If something occurs, something else happens. Most dApps, on the other hand, operate by combining smart contracts to provide a comprehensive, synergistic set of features. Thousands of dApps exist across multiple blockchain networks, including banking, gaming, exchanges, and media – and they may all use smart contracts in different ways and for different purposes.

Financial applications such as trading, investing, lending, and borrowing are examples of smart contract applications. They may be utilized for gaming, healthcare, and real estate applications, and they can even be used to construct whole business organizations. In this post, we'll look at some real-world instances of smart contracts and dApps, as well as their potential to change the future.

Ryan Boder, CMO and key contributor at [API3](#), a first-party Oracle solutions provider that allows Web3 apps to use any web API straight from a smart contract, is in an excellent position to discuss smart contracts.

A smart contract, according to Boder, is "an agreement between many parties that is automatically and reliably implemented such that the parties do not have to trust each other or any third-party intermediary." It results in the exchange of value, which might be anything from the exchange of digital tokens to the automation of process or a one-of-a-kind user experience. Smart contracts are the basis for everything else that is feasible in the [Web3 universe](#)."

Blockchain and Smart Contracts

The idea of smart contracts is mostly based on blockchain technology.

The computer software that operates on a blockchain network verifies, executes, and enforces smart contracts. When both or all parties to the smart contract agree on its conditions, the program will run automatically. As the contract is confirmed and enforced by the blockchain network, there is no need for a third party.

A blockchain is a decentralized network consisting of a growing list of records (blocks) connected by encryption. A blockchain network, unlike a traditional database, does not have a single central point. The data saved in the blockchain is shared by all of the computers that make up the network. As a result, the network is less vulnerable to potential outages or assaults.

Furthermore, with a blockchain, a record on one computer cannot be changed until the same record is changed on other devices in the network. Transactions on a blockchain are organized into blocks that are connected in a chain. When the preceding block is finished, a new block is formed. The blocks are ordered chronologically, and each one carries a cryptographic hash of the preceding one.

One of the most advantageous aspects of the blockchain is that, since it is a decentralized system that exists between all authorized participants, there is no need to pay intermediaries (middlemen), saving you time and conflict.

Blockchains are not without flaws, but they are unquestionably quicker, cheaper, and more secure than older systems. This is why we're seeing more smart contracts implemented on various blockchain networks including Ethereum, Solana, Tezos, Hyperledger, and others.

Vitalik Buterin On Smart Contracts

"In a smart contract approach, an asset or currency is transferred into a program and the program runs this code and at some point, it automatically validates a condition and it automatically determines whether the asset should go to one person or back to the other person, or whether it should be immediately refunded to the person who sent it, or some combination thereof," explained Vitalik Buterin, the 28-year-old founder of Ethereum, at a DC Blockchain Summit.

Meanwhile, the decentralized ledger stores and copies the document, providing it with security and immutability.

Digital and Real-World Applications

Smart contracts are so-called because they outline the parameters of a transaction or agreement. When such conditions are satisfied (i.e., the receipt of a trigger or event), they automatically execute and complete their assigned responsibilities.

A buy-here, pay-here automobile dealership is a real-world example. When a car dealership agrees with a buyer, they promise that the buyer will get the vehicle's title after the automobile is paid for in full. When the final payment is received through a smart contract, the title is immediately transmitted to the buyer without the need for human interaction.

Smart contracts open up new avenues for individuals to borrow, buy, sell, and trade. Unlike typical financial apps, [decentralized finance \(DeFi\)](#) [dApps](#) are not restricted to certain hours, since "market hours" are available 24 hours a day, seven days a week. Furthermore, dApp users may engage without the need for centralized custody or intermediary costs.

Smart contracts are also used by companies in the gaming industry. NFTs are one-of-a-kind digital assets that are used to represent in-game content like typical game applications, which require players to pay to get access to in-game assets or gameplay customizations, NFTs are owned wholly and allow players to keep in-game purchases, sell them to other game players, or move them to other compatible games. More on these later in the coming sections.

Smart contracts can provide the same functions as regular contracts, however at a significantly cheaper cost in most circumstances. Smart contracts, like digital and physical contracts, may serve as legally enforceable contracts. Some jurisdictions, notably Arizona and California, currently permit the use of smart contracts as legally binding transactions, including marriage licenses.

Mitchell Amador, CEO, and creator of Immunefi, a bug bounty, and security services platform, talked with CMSWire about smart contracts and how he sees them having a significant economic effect.

"Smart contracts represent a significant advancement in business process and automation." "Amador said. "We fully anticipate that they will alter the economy by making it easier for enterprises to connect their key activities in a peer-to-peer manner. Smart contracts may be used for trading, investing, and borrowing, but they can also be applied for voting, health care, and real estate."

Example Of A Simple Smart Contract: Escrow For Transactions

Assuming you are interested in renting an apartment from an agent.

You may achieve this using the blockchain and [cryptocurrencies](#). You get a receipt that is stored in the agent's virtual contract; He provides you with

the digital entrance key, which is sent to you by a given date. If the key does not arrive on time, the blockchain will provide a refund.

If he sends the key before the rental date, the function retains it and releases both the fee and the key to you and him when the date comes due.

The system is based on the If-Then logic and is observed by hundreds of people, so you can anticipate flawless delivery.

The agent is confident he'll get compensated if he gives you the key. You will obtain the key if you submit a specified number of bitcoins. Because all participants are simultaneously notified, the document is automatically terminated after the allowed period, and the code cannot be tampered with by any one of you without the other knowing.

Smart contracts may be used in a variety of contexts, including:

- financial derivatives
- insurance premiums
- breach of contract
- property law
- credit monitoring
- financial services

- legal procedures

More about this in the next sections.

How Smart Contracts Work

Smart contracts are computer programs that automatically execute all or portions of an agreement and are kept on a blockchain-based platform. As mentioned further below, the code may either be the only embodiment of the parties' agreement or it can supplement a standard text-based contract by carrying out certain clauses, such as transferring assets from Party A to Party B.

Because the code is duplicated through various nodes of a blockchain, it benefits from the security, permanence, and immutability that a blockchain provides. That replication also implies that the code gets performed when each new block is added to the network.

If the parties have signaled that certain parameters have been satisfied by commencing a transaction, the code will perform the step triggered by those parameters. The code will not take any action if no such transaction has been started. The majority of smart contracts are written in one of the programming languages that are specifically designed for such computer applications, such as Solidity.

Currently, the input parameters and execution stages for a smart contract must be precise and objective. In other words, if "x" happens, then step "y" is executed. As a result, the actual functions performed by smart contracts are quite simplistic, such as automatically sending a particular amount of bitcoin from one party's wallet to another when certain conditions are met.

Smart contracts will grow more complicated and capable of managing sophisticated transactions as blockchain use increases and more assets are tokenized or come "on-chain." Indeed, developers are already connecting numerous transaction phases to create more complicated smart contracts.

Nonetheless, we are several years away from the code being able to evaluate more subjective legal criteria, such as whether a party met a commercially reasonable efforts level or if an indemnifications provision should be activated and the indemnity paid.

Before a compiled smart contract may be performed on certain blockchains, an extra step is necessary, which is the payment of a transaction fee for the contract to be added to the chain and acted upon. In the case of the Ethereum blockchain, smart contracts are performed on the Ethereum Virtual Machine (EVM), and this payment is known as "gas," and it is paid using the ether cryptocurrency.

The more complicated the smart contract (based on the transaction stages to be done), the more gas that must be paid to execute the smart contract. As a result, gas presently serves as a critical gatekeeper to prevent extremely complicated or many smart contracts from overloading the EVM.

Smart contracts are now best adapted to autonomously executing two sorts of "transactions" featured in many contracts:

- (1) securing the payment of money upon specified triggering events, and
- (2) imposing financial penalties if certain objective requirements are not met. Human participation, whether via a trusted escrow holder or even the legal system, is not necessary in any of these cases after the smart contract has been implemented and is functioning, decreasing the contracting process's execution and enforcement costs.

Smart contracts, for example, might reduce so-called procure-to-pay gaps. When a product arrives and is scanned at a warehouse, a smart contract can quickly seek the necessary permissions and, if received, send payments from the buyer to the seller.

Sellers would get paid quicker and would no longer need to participate in dunning, while purchasers would save money on accounts payable. This might affect working capital needs as well as simplify finance operations for both parties.

In terms of enforcement, a smart contract may be coded to disable access to an internet-connected item if payment is not received. For example, if payment is not received, access to particular material may be automatically restricted.

Currently, Ethereum is the most popular smart contract platform, although it may be operated on a variety of cryptocurrency blockchains, including EOS, Neo, Tezos, Tron, Polkadot, and Algorand. Anyone may design and deploy a smart contract to a blockchain. Because its code is open and publicly verifiable, any interested party can see precisely what logic a smart contract follows when it receives digital assets.

- Smart contracts may be developed in several different programming languages (including Solidity, Web Assembly, and Michelson). Each smart contract's code is kept on the Ethereum network's blockchain, enabling any interested participant to view the contract's code and current state to verify its operation.
- Along with the blockchain and transaction data, each computer on the network (or "node") holds a copy of all existing smart contracts and their current state.

- When a smart contract gets cash from a user, all nodes in the network run its code to establish a consensus on the conclusion and resultant flow of value. This is what enables smart contracts to execute safely without a central authority, even when users conduct sophisticated financial transactions with unrecognized entities.
- To execute a smart contract on the Ethereum network, you will often be required to pay a price known as "gas" (so named because these fees keep the blockchain running).
- Smart contracts, once put on a blockchain, are largely unchangeable, even by their author. (This rule is not without exceptions.) This makes it more difficult for them to be restricted or shut down.

Smart Contracts' Historical Background

As a doctoral student at the University of Washington, computer scientist and [cryptographer](#) Nick Szabo coined the phrase "smart contract." He was also credited with inventing a virtual currency dubbed "Bit Gold" in 1998, more than a decade before bitcoin. Nick was one of the early "cypherpunks" who gathered in Santa Cruz in 1992 and was instrumental in the conception and creation of alternative and privatized currencies. His research focused on leveraging computer science to safeguard property and privacy in cyberspace.

Szabo is widely suspected to be the genuine Satoshi Nakamoto, the mysterious creator of bitcoin, which he has denied.

In 1994, Nick created the term "Smart Contracts," proposing that a smart contract infrastructure be built by replicated asset registries and contract execution through [cryptographic hash chains](#) and a [Byzantine fault tolerance method](#). He also created BitGold, an early forerunner to Bitcoin. I

came upon a video of Nick in one of his computer science lectures at a US institution in the early 1990s, in which he discusses computer science as the future of law, noting the potential of smart contracts. This was long before the internet or the international web became popular.

Szabo claims that:

The digital revolution has enabled the creation of new institutions as well as new means to codify the ties that comprise these organizations. These new contracts are dubbed "smart" because they are significantly more useful than their inert paper-based forefathers. There is no mention of artificial intelligence. A smart contract is a collection of digitally stated promises, as well as the protocols through which the parties fulfill these promises.

Szabo's placement of quotations around the term "smart" when comparing smart contracts to paper-based contracts, and his rejection of artificial intelligence are noteworthy. Smart contracts are "smarter" than paper contracts because they can perform certain pre-programmed stages automatically, but they should not be seen as sentient instruments capable of parsing a contract's more subjective needs.

Indeed, Szabo's vending machine is a famous example of a smart contract. When a customer fulfills the terms of the "contract" (i.e., inserts money into the machine), the machine automatically follows the terms of the unwritten agreement and distributes the snack.

Simply put, smart contracts are automated transaction protocols that carry out the provisions of a contract, according to Szabo. He wished to digitalize the capabilities of electronic transaction systems such as POS (point of sale).

Szabo also advocated in his article the execution of a contract for synthetic assets such as derivatives and bonds. Szabo stated: "These new securities are created in several ways by mixing securities (such as bonds) and derivatives (such as options and futures). Due to automated analysis of these complicated term structures, very complex payment term structures may now be constructed into standardized contracts and exchanged with minimal transaction costs."

In layman's terms, he was talking about the selling and purchase of complicated derivatives.

Many of Szabo's predictions in the paper came true before blockchain technology. Derivatives trading, for example, is now primarily done through computer networks employing complicated term structures.

Isn't it a great idea? For many years, Szabo worked on this concept and even authored a book called "Smart Contracts: Building Blocks for Digital Free Markets." The issue was that blockchain technology did not exist in 1994.

But it now does!

Bitcoin pioneered the use of blockchain technology in 2009. Ethereum was developed in 2015 by an astute young guy called [Vitalik Buterin](#), and it pioneered the use of functional smart contracts.

Ricardian Contracts, a notion introduced in 1996 by Ian Grigg and Gary Howland as part of their work on the Ricardo payment system to transfer assets, are also the genesis of smart contracts today. Grigg envisioned Ricardian Contracts as a bridge between text contracts and code, with the following parameters: a single document that "is

- a) a contract offered by an issuer to holders,
- b) for a valuable right held by holders and managed by the issuer,
- c) easily readable by people (like a contract on paper),
- d) readable by programs (parsable like a database),
- e) digitally signed,
- f) carries the keys and server information,
- g) allied with a unique and secure identifier

A definition of Smart Contracts

“Smart contracts are self-executable and automated computer programs that can carry out the terms of a contract or a business agreement between two or more parties. Smart contracts also have an additional feature of a “value exchange” that is contingent on some pre-agreed terms, in addition to the features of a traditional contract. What distinguishes a smart contract from a normal contract is that firstly it is automated, and secondly, a smart contract can transfer economic value automatically and simultaneously while satisfying the terms of the traditional contract, due to a unique and exclusive feature of a blockchain, without needing a trusted intermediary like a bank, notary, auditor or accountant from the old traditional system.”
– [Arifa Khan](#)

Nick's reading of Ayn Rand and his knowledge that overlaps many academic domains of contract law, computer science, cryptography, game theory, economics, and studies of arcane and little known subjects such as the origins of money inspired him to create these utopian concepts such as smart contracts (this is as far as one can distill from his astute research and mind-blowing insights from his writings, and there could be more)!

Each of his books is a condensed or hashed encyclopedia for a more efficient contemporary digital economy, and it has the potential to spark a

new Nobel Prize-winning theory. Those who were fortunate enough to engage with Nick in person would uncover even more remarkable revelations, such as his self-taught knowledge of financial services and capital markets, which he happily shared with others.

Many of Nick's significant contributions to cryptography originate from his conviction that "trusted third parties are security flaws." That is, centralization is the primary weakness in monetary systems, as well as everything wrong with the contemporary economy.

The notion that the evils of conflicts of interest, self-dealing, and moral hazards that define contemporary central institutions everywhere can be cured decisively with decentralized networks is as radical as flat-earthers discovering that the earth was round.

This concept serves as the basis for a new global [cryptocurrency-based](#) architecture that promises a more democratic, equitable economy that is less susceptible to corruption and more robust to man-made shocks and black swan occurrences like the 2008 economic crisis.

Nick is a firm believer in Bitcoin and the future of cryptocurrencies –

“Running non-stop for eight years, with almost no financial loss on the chain itself, [Bitcoin] is now in important ways the most reliable and secure financial network in the world,”

Users must be able to exchange and trade with one another with trust, despite not knowing each other beforehand. This establishes Blockchain’s core principles, where the power of its users, and not a central entity, determines its success.

The promise

When democratic institutions throughout the globe are at risk of falling to the gripping vices and blatant avarice of a few, what background checks of elected leaders and constitutionally appointed guardians of democracy cannot do may now be accomplished with the power of technology. That is the fundamental truth of blockchain.

The promise of crypto protocols for the ordinary man has democratized forms of governance for new types of organizations, not a quick buck from purchasing one.

The promise of cryptocurrency-based paradigms is a totally new architecture for counterparties to come together and transact value on smart contracts and blockchains — without middlemen, or at the very least a more competitive atmosphere for intermediaries. There are no faster, cheaper, more efficient, or more lucrative methods for established incumbents to operate.

The initial coin offerings (ICOs) that became popular in 2017 are an example of smart contracts on steroids, which were made possible by Ethereum — the world's first enterprise-grade generic smart contract generation platform. A smart contract on Ethereum may be created and executed by anybody, from an individual to a gigantic company like a bank.

The price of ether, the cryptocurrency that powers this technology platform, skyrocketed from less than \$10 at the end of 2016 to \$1000 by the end of 2017, after \$5.5 billion in ICOs were launched on Ethereum's platform. Ethereum shook under the weight of its rush, leading it to gain prominence. However, it is a welcome change from the days when technology had to wait decades before becoming ubiquitous.

The venture capital business paradigm has been thoroughly and irreparably undermined by ICOs (a smart contract for crowd-funding digital firms). Smart contracts (ICOs) put a harsh mirror up on venture capitalists for the

value they provided to entrepreneurs, just as they will now to middlemen in every other sector.

What began as the promise of unstoppable currency (bitcoin in 2008) and unstoppable computers (Ethereum in 2014) will be known as unstoppable smart contracts (Himalaya Capital Exchange in 2018) - with the promise of reshaping the global financial services infrastructure. Smart contracts have the potential to push the Wolf of Wall Street to adapt to the new environment.

What Do All Smart Contracts Have in Common?

Smart contracts are made up of three main parts:

- The first are the signatories or the parties to the smart contract. Digital signatures are used in smart contracts to accept or reject contractual obligations. The agreement is "signed" using a person's crypto wallet.
- The second is the topic of an agreement or contract. A person, for example, may agree to buy a certain quantity of Bitcoin.
- The third: The exact conditions involved, such as the current Bitcoin price.

Furthermore, all smart contracts have certain characteristics. Because they are a part of a blockchain, they have a state that is shared by the whole network. Each node running the blockchain has a copy of the smart contract's state, and when a transaction involving it happens, its state changes and is updated on each node.

The conditions of a smart contract that has been added to the blockchain cannot be modified since there is no mechanism to amend them without informing the network. The only thing that has changed is its status.

"Smart contracts are programmed to do precisely what they are meant to do – no matter what." They cannot be changed once they have begun. "The blockchain guarantees it," added Boder.

Furthermore, the logic of a smart contract, like that of a lawyer's contract, cannot be twisted since it is a binding agreement between two parties that is self-verifying and self-enforcing.

"Smart contracts that can be used without the users having to trust anyone, such as the developer or company that created them, are especially useful for applications or deals involving a transfer of value, such as exchanging money, finance, banking, legal contracts, escrow, deeds/property rights, and insurance," Boder explained.

"The beauty of smart contracts is that they are self-executing and do not need confidence that your counterpart will fulfill their responsibilities," Lifshitz noted. "When the particular criteria written within the smart contract are met, the contract will execute automatically." That is, smart contracts may boost efficiency dramatically in any firm."

Elements Of Smart Contracts

Conditions

Every smart contract has programmed logic — an "if, when, and then" statement that serves as the foundation of any software. Developers and business teams collaborate to define the conditions that must be satisfied for the contract to proceed.

These terms and conditions often contain permitted payment or a shipping receipt. However, depending on the logic of the program, more intricate phrases may also be encoded.

Placement

After creating and evaluating the logic for correctness, the contract's security is evaluated. Once approved, the contract may be added to an existing blockchain where it cannot be changed and will get updates from a secure data source known as an oracle.

Execution

The oracle basically outlines the external events occurring between the two parties (payments, shipment receipts, etc.). If all of the preset requirements are satisfied, the smart contract may be finalized and stored on the blockchain.

When compared to its manual cousin, smart contracts provide a plethora of benefits — as well as a few drawbacks — to crypto aficionados as well as other emerging businesses. Continue reading to understand more about what distinguishes smart contracts from traditional contracts.

Characteristics of Smart Contracts

Smart contracts have the following features:

- they are self-verifying due to automated capabilities;

- they are self-enforcing when all rules are followed;
- They are tamper-proof since no one can modify what has been encoded.
- The data intake and security are the next two most critical properties of SCs.
- Data from external sources, notably data feeds and APIs, is fed into blockchains and used for smart contract execution. These real-time data feeds for blockchains are known as "oracles," and they serve as the intermediary between the data and the contract. Oracles may be hardware or software-based. Consider an insurance smart contract that pays out if particular places are affected by winds of more than 100 mph. The SC would get external data (in this example, from an anemometer) and pay based on the input (mph). Because there are just a few (or possibly one) data sources at times, the decentralization element suffers, resulting in a single point of failure in the SC.
- Bugs and errors in the code - While it is advised that a smart contract be a combination of computer code and a physical contract (worded), bugs and errors in code might lead to disputes and procedural issues in identifying errors and the parties liable for them. For example, in June 2016, a hacker exploited a weakness in the code of the Decentralized Autonomous Organization (DAO), a smart contract based on Ethereum, and stole 50 million Ether.

Capabilities of Smart Contracts

Smart contracts are capable of:

- automating formerly manual operations;

- guaranteeing safety;
- cutting ties with trustworthy middlemen;
- allowing multi-signature accounts to release payments after all parties involved have confirmed the arrangement;
- handling user contracts; giving usefulness to other contracts (much as a software library);
- storing app-specific information (domain registration information, membership records, etc.).

Life Cycle Of A Smart Contract

Smart contracts are usually constructed in a high-level language like Solidity. However, for them to function, they must be compiled to the low-level bytecode that runs in the EVM. Once constructed, they are deployed on the Ethereum network using a specific contract creation transaction, which is recognized by being submitted to the unique contract creation address, 0x0 (see [[contract reg](#)]).

Each contract is recognized by an Ethereum address, which is calculated from the contract formation transaction based on the originating account and nonce. A contract's Ethereum address may be used as the receiver of a transaction, transferring payments to the contract or calling one of the contract's functions.

It is worth noting that, unlike EOAs, no keys are connected with an account established for a new smart contract. You do not have any protocol-level privileges as the contract creator (although you can explicitly code them into the smart contract). You do not obtain the private key for the contract account, which does not exist—smart contract accounts are self-sufficient.

Contracts, it is important to note, only execute when they are invoked by a transaction. All smart contracts in Ethereum are eventually executed as a result of a transaction started by an EOA. A contract may call another contract, which in turn can call another contract, and so on, but the initial contract in such a chain of execution is always called by an EOA transaction.

Contracts are never operated "in the background" or "on their own." Contracts are basically inert until triggered by a transaction, either directly or indirectly as part of a sequence of contract calls. It is also important to note that smart contracts are not processed "in parallel" in any way—the Ethereum global computer is a single-threaded machine.

Transactions are atomic in the sense that they are either successfully completed or reversed. In various contexts, a successful transaction termination signifies different things:

(1) If a transaction is sent from one EOA to another, any changes to the global state (e.g., account balances) caused by the transaction are recorded;

(2) If a transaction is sent from one EOA to a contract that does not invoke any other contracts, any changes to the global state are recorded (e.g. account balances, state variables of the contracts)

(3) If a transaction is sent from an EOA to a contract that only invokes other contracts in a way that propagates errors, any changes to the global state are recorded (e.g. account balances, state variables of the contracts); and

(4) If a transaction is sent from an EOA to a contract that only invokes other contracts in a way that does not propagate errors, some changes to the global state may be recorded (e.g. account balances, state variables of the contracts) (e.g. state variables of the erroring contracts). If a transaction is reversed, all of its consequences (state changes) are "rolled back" as if the transaction never occurred. A failed transaction is still recorded as attempted, and the ether spent on gas for the execution is deducted from the originating account, but it has no additional impact on the contract or account state.

As previously said, it is critical to note that the code of a contract cannot be modified. A contract, on the other hand, maybe "deleted," which removes the code and its internal state (storage) from its address, leaving a blank account. Because there is no longer any code to run, any transactions submitted to that account address after the contract has been removed result in no code execution.

To remove a contract, use the EVM opcode SELFDESTRUCT (previously called SUICIDE). That process costs "negative gas," or a gas refund, motivating the release of network client resources due to the elimination of the stored state. Because the blockchain is immutable, deleting a contract in this manner does not destroy the contract's transaction history (past). It is also worth noting that the SELFDESTRUCT feature will be accessible only if the contract author built the smart contract to support it. The smart contract cannot be erased if the contract's code lacks a SELFDESTRUCT opcode or is unavailable.

Why Are Smart Contracts Important?

Developers may use smart contracts to create a broad range of decentralized applications and coins. They're utilized in anything from new financial tools to logistics and gaming experiences, and they're kept on a blockchain just

like any other cryptocurrency transaction. Once a smart-contract program is put to the blockchain, it cannot be modified or reversed (although there are some exceptions).

Smart-contract-powered apps are also known as "decentralized applications" or "dapps," and they include decentralized finance (or DeFi) technology, which aspires to revolutionize the financial sector. [DeFi applications](#) enable bitcoin users to conduct complicated financial activities – saving, lending, and insurance — without the involvement of a bank or other financial institution, and from anywhere around the globe. Among the most popular contemporary smart-contract-powered apps are:

- **Uniswap:** A decentralized exchange that enables users to trade various types of cryptocurrency using smart contracts without the need for a central authority to establish the exchange rates.
- **Compound:** A platform that leverages smart contracts to allow investors to earn interest and borrowers to acquire a loan instantaneously, eliminating the need for a bank in the middle.
- **USDC:** A cryptocurrency that is linked to the US dollar through a smart contract, making one USDC equal to one US dollar. USDC is a stablecoin, which is a newer kind of digital currency.

So, how would you put these smart contract-enabled instruments to use? Assume you have some Ethereum that you want to swap for USDC. You can put some Ethereum into Uniswap, which will automatically discover the best exchange rate, conduct the deal, and give you your USDC via smart contract. You could then invest part of your USDC in Compound to lend to others and earn an algorithmically calculated interest rate — all without utilizing a bank or other financial institution.

Swapping currencies is a costly and time-consuming process in conventional finance. It is both difficult and risky for people to lend their liquid assets to strangers on the other side of the planet. Smart contracts, on the other hand, enable both of these possibilities, as well as a plethora of others.

How Do Smart Contracts Work?

A smart contract is a computer program that runs on a blockchain. It consists of a collection of rules that form an agreement between two or more parties. When these conditions are satisfied, the digital contract will carry out the transaction. It functions similarly to a standard application in that it implements certain business rules, but it makes use of a blockchain as a database.

Identify Agreement

- multiple parties identify a cooperative opportunity and desired outcomes
- agreements potentially in scope could include business processes, asset swaps, transfer of rights and more

Set Conditions

- smart contracts could be initiated by the parties themselves or by satisfaction of certain conditions like financial market indices, natural disasters or event via GPS location
- temporal conditions could initiate smart contracts on holidays, birthdays and religious events

Code the Business Logic

- a computer program is written in a way that the arrangement will automatically perform when the conditional parameters are met

Encryption & Blockchain Technology

- encryption provides secure authentication and verification of messaging between the parties relating to the smart contract

Execution & Processing

- in a blockchain iteration, when consensus is reached on authentication and verification, the smart contract is written to a block
- the code is executed, and the outcomes are memorialized for compliance and verified

Network Updates

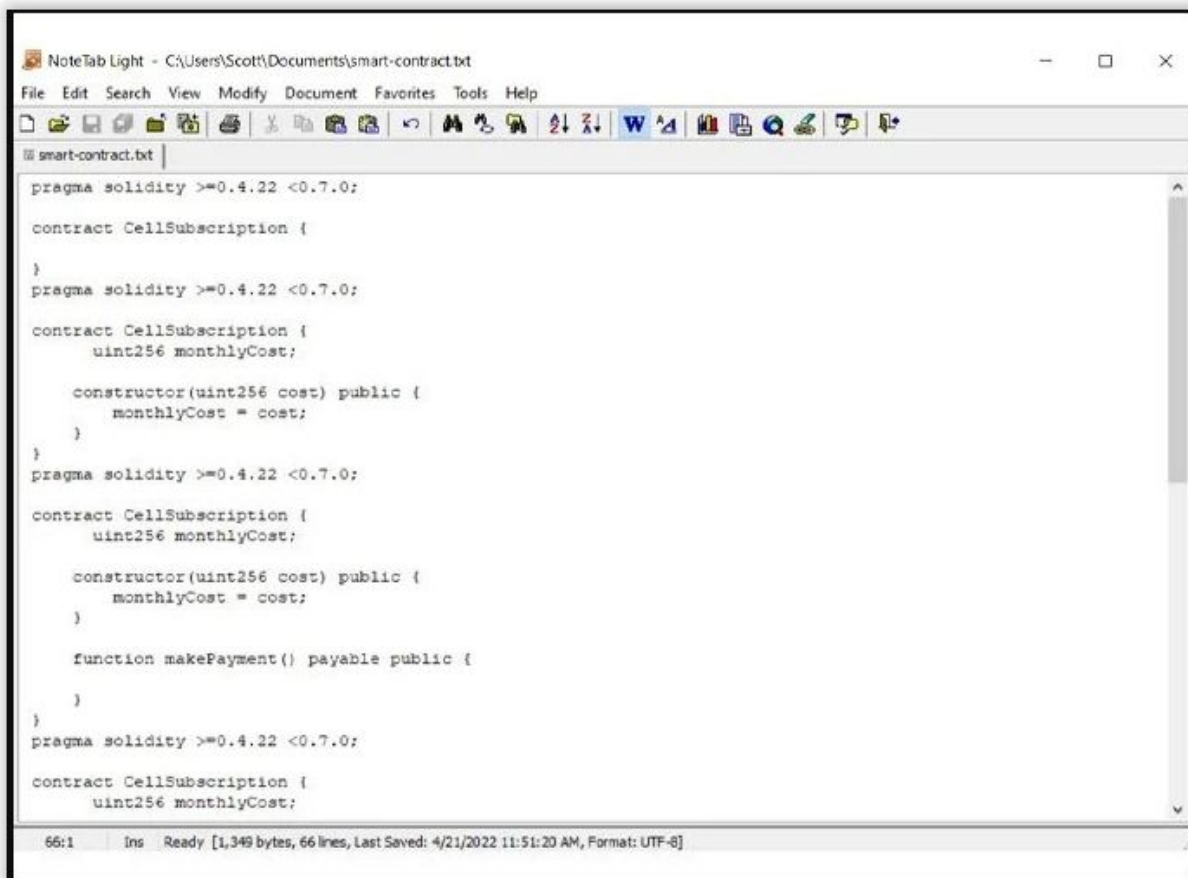
- after performance of the smart contract, all computers in the network update their ledgers to reflect the new state
- once the record is verified and posted to the blockchain, it cannot be altered, it is append only

SOURCE: THE-BLOCKCHAIN.COM

What Does Smart Contract Code Look Like In Practice?

Smart contracts make use of code that is comparable to that found in other programming languages. They effectively state, "if this occurs, do this; otherwise, do nothing." smart contracts may collaborate with other smart contracts to provide more complicated functionality.

Here's an [example of code](#) that might be used to construct a smart contract. It was created using [solidity](#), a smart contract-specific object-oriented, high-level programming language.



```
pragma solidity >=0.4.22 <0.7.0;

contract CellSubscription {
}
pragma solidity >=0.4.22 <0.7.0;

contract CellSubscription {
    uint256 monthlyCost;

    constructor(uint256 cost) public {
        monthlyCost = cost;
    }
}
pragma solidity >=0.4.22 <0.7.0;

contract CellSubscription {
    uint256 monthlyCost;

    constructor(uint256 cost) public {
        monthlyCost = cost;
    }

    function makePayment() payable public {
    }
}
pragma solidity >=0.4.22 <0.7.0;

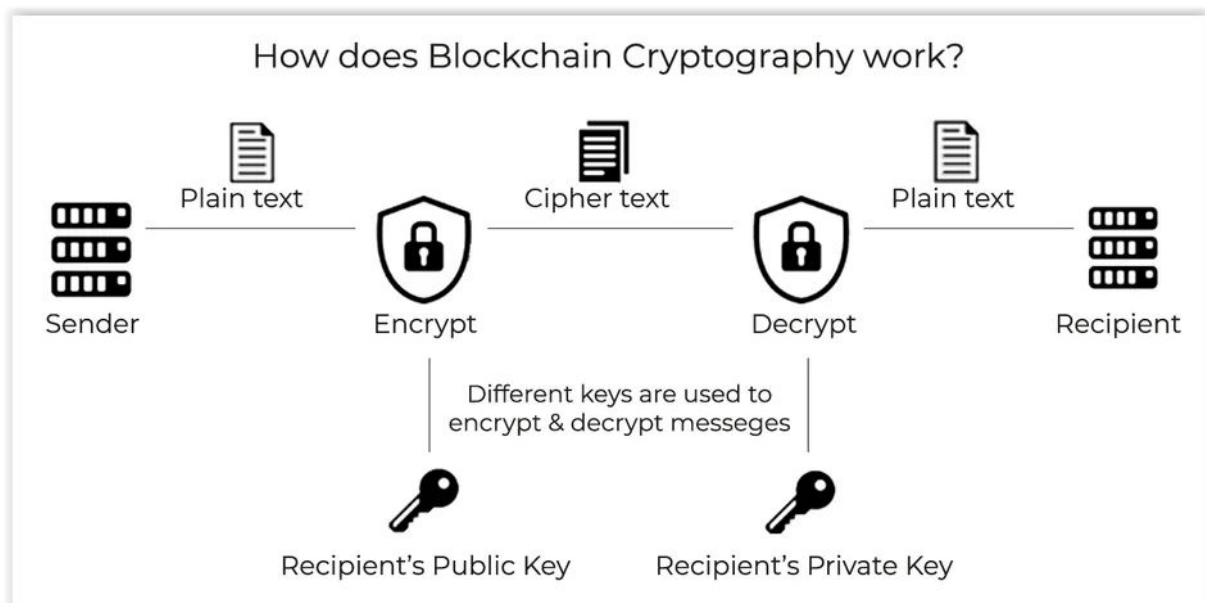
contract CellSubscription {
    uint256 monthlyCost;
```

The screenshot shows a Notepad++ window titled "NoteTab Light - C:\Users\Scott\Documents\smart-contract.txt". The window contains Solidity code for a smart contract named "CellSubscription". The code is repeated four times, each time starting with a "pragma solidity" statement. The first instance is an empty contract. The second instance adds a "uint256 monthlyCost" variable and a constructor. The third instance adds a "makePayment" function. The fourth instance is partially visible. The status bar at the bottom indicates "66:1 Ins Ready [1,349 bytes, 66 lines, Last Saved: 4/21/2022 11:51:20 AM, Format: UTF-8]".

Once created, this code is built using the Solidity compiler and evaluated using the [Ethereum Remix IDE](#), an open-source browser-based IDE for Ethereum smart contracts. Once the smart contract is complete, it may be deployed on the Ethereum network, which costs between \$400 and \$2,000. Other blockchain networks may charge less, and some, like [Infura](#), provide three smart contracts for free.

The Structure of a Smart Contract

Nobody can change the software once it has begun to execute. This is another benefit. Because smart contracts depend on blockchain encryption, anyone may inspect the source code to understand what the software does and know that it cannot be manipulated by hackers or viruses.



SOURCE: PUBLIC DOMAIN

Interaction with Traditional Text Agreements

One of the challenges in addressing smart contracts is that the word is used to describe two quite distinct concepts. The first concerns smart contracts that are generated and implemented without the support of an enforceable text-based contract. For example, two parties may form an oral agreement on the business relationship they wish to capture and then immediately convert that agreement into executable code. These are referred to as "code-only smart contracts" farther down. The second paradigm includes using smart contracts as vehicles to carry out some terms of a conventional text-based contract, with the text referencing the usage of the smart contract to carry out specific provisions. These are referred to as "ancillary smart contracts."

Are Smart Contracts Enforceable?

As a disclaimer, I am not a lawyer and I do not claim to be one. For legal matters, seek the advice of a qualified and competent lawyer. This section is just for entertainment purposes only.

In the united states, there is no federal contract law; rather, the enforcement and meaning of contracts are established at the state level. As a result, although many essential concepts apply similarly across state boundaries, and the national conference of commissioners on uniform state laws has been working to unify state laws, any conclusions about smart contracts must be tempered by the fact that states may take various positions.

A debate on the enforceability of smart contracts must begin with defining the difference between an agreement and a "contract." although two parties may engage in a variety of "agreements," states typically understand that a contract indicates that the agreement is legally binding and enforceable in a court of law. State courts have typically looked at whether the common law conditions of offer, acceptance, and consideration are met to assess enforceability.

These fundamental needs may very certainly be met with supplementary smart contracts. For example, an insurer may create a flight insurance plan

that automatically pays out if a flight is delayed by more than two hours.

The important conditions, such as how the delay is computed, may be specified in a text-based contract, with the actual creation of the contract (payment of the premium) and execution (automatic payout after a verified delay) handled by an auxiliary smart contract. In this case, the insurer has issued a firm offer for a flight insurance policy, which the insured accepts in exchange for payment of the premium.

Although some contracts must be in writing today, and extra formalities such as those required by the uniform commercial code (ucc) and state laws of frauds may be necessary, agreements do not always need to be in writing to be deemed enforceable. As a result, many code-only smart contracts will be enforceable under state contract laws. In this light, Szabo's vending machine example is illuminating.

While the customer has several implicit rights in such a situation, a contract was created with no relevant written conditions other than a price display for each item. Thus, outside of the hurdles imposed by the ucc and laws of fraud, the fact that an agreement is represented solely in code, as is the case with code-only smart contracts, offers no specific impediment to contract formation. Indeed, the importance of information technology in contract creation has long been studied by several laws and legal structures.

For example, the uniform electronic transactions act (ueta), which was enacted in 1999 and serves as the foundation for state law in 47 states, states that electronic records, which include records created by computer programs, and electronic signatures (i.e., digital signatures using public-key encryption technology), have the same legal effect as their written counterparts.

Ueta even acknowledges the legitimacy of "electronic agents," which it defines as "a computer program or an electronic or other automated means used independently to initiate an action or respond to electronic records or performances in whole or in part, without review or action by an individual." an electronic agent is "capable, within the constraints of its programming, of initiating, reacting, or engaging with other parties or their electronic agents once triggered by a party, without further attention from that party," according to ueta. Possibly a foresighted recognition of smart contracts.

Similarly, the federal electronic signatures recording act (e-sign act) not only recognizes the validity of electronic signatures and electronic records in interstate commerce, but also states that a contract or other record relating to a transaction "shall not be denied legal effect, validity, or enforceability solely because its formation, creation, or delivery involved the action of one or more electronic agents so long as the action of any such electronic agent is lethal." an "electronic agent" is defined as "a computer program or other automated mechanism employed independently to start an action or react to electronic records or performances in whole or in part without inspection or action by a human at the time of the action or response."

Though comprehending the existing legal environment is necessary for assessing the enforceability of smart contracts today, users utilizing smart contracts in the future may not need to depend on laws that predate the creation of blockchain technology. Arizona and Nevada have already revised their state versions of ueta to specifically include blockchains and smart contracts.

The fact that these governments have chosen significantly diverse meanings of those important words shows that if additional jurisdictions follow their example, there may be increased pressure to adopt unified definitions to reflect blockchain and smart contract advances.

Challenges With the Widespread Adoption of Smart Contracts

Given the current legal foundations for recognizing electronic contracts, a court today is quite likely to accept the legality of code that executes smart contract provisions—what we call auxiliary smart contracts.

There is additional precedence to imply that a smart contract written in code may be afforded comparable legal protection. The barrier to broad smart contract adoption may therefore be less about legal constraints and more about possible incompatibilities between how smart contract code runs and how parties conduct business. Some of these problems are outlined below:

Non-Technical Parties: How Can They Negotiate, Draft, and Adjudicate Smart Contracts?

A significant barrier to the broad adoption of smart contracts is that parties will need to depend on a trusted, technical expert to either record the parties' agreement in code or check the accuracy of code created by a third party.

While some compare this to paying a lawyer to explain "the legalese" of a typical text-based contract, this is an incorrect comparison. Non-lawyers can often grasp basic short-form agreements as well as many parts of larger agreements, particularly those establishing commercial conditions. A non-programmer, on the other hand, would be at a loss to grasp even the most basic smart contract and would be substantially more reliant on an expert to explain what the contract "means."

To some degree, contractual parties' inability to grasp the smart contract code will not prevent them from engaging in auxiliary code agreements. This is because text templates may be developed and used to define what parameters must be submitted and how those parameters will be performed for many fundamental operations.

Consider a basic smart contract function that deducts a late charge from a counterparty's wallet if a specified payment is not received by a set date. The text template might request the parties to input the projected payment amount, due date, and late charge amount. However, a party may wish to validate that the underlying code will truly execute the functions indicated in the text and that there are no extra conditions or parameters—particularly if the template disclaims any obligation deriving from the correctness of the underlying code. This assessment will need the involvement of a reputable third party with programming knowledge.

In the absence of such templates, when new code must be generated, the parties must express the purpose of their agreement to a programmer. Giving the programmer a copy of the legal agreement would be wasteful since it would necessitate the programmer to attempt to understand a legal document. Parties depending on auxiliary smart contracts may need to provide a separate "term sheet" describing the functionality that the smart contract should deliver to the programmer.

The parties may also want formal assurances from the programmer that the code works as intended. As a consequence, for unique arrangements that do not depend on an existing template, the parties may need to enter into a formal agreement with the smart contract programmer, similar to what parties may do today with a supplier of services for Electronic Data Interchange (EDI) transactions.

Insurance firms might potentially provide insurance to safeguard contractual parties against the risk that a smart contract code does not

execute the tasks indicated in an agreement's language. Although the parties would want to check (or have third parties review) the code, insurance may offer extra protection since the parties may overlook problems when evaluating the code. The parties would also be relieved to know that the insurance firm most likely completed its own code audit before agreeing to insure the code.

Code-only smart contracts used for business-to-consumer transactions may raise new challenges that must be addressed. Courts are suspicious of enforcing agreements in which the consumer did not get proper notice of the terms of the agreement, and may be unwilling to enforce a smart contract in which the customer was not also furnished with an underlying text agreement that covered the whole contents.

Finally, when the validity or performance of smart contracts is increasingly assessed, courts may need a system of court-appointed experts to assist them in deciphering the code's meaning and purpose. When technological concerns are at the heart of a disagreement, parties now commonly utilize their own experts. While both federal and many state courts have the right to select their own experts, they seldom do so. If the number of standard contract conflicts centered on reading smart contract code rises, this method may need to change.

Smart Contracts and the Reliance on “Off-chain” Resources

Many suggested use-cases for smart contracts presume that the smart contract would get information or parameters from resources that are not on the blockchain itself—so-called off-chain resources. Assume a crop insurance smart contract is set up to transfer value to an insured party if the temperature goes below 32 degrees Celsius at any time.

The temperature data must be received by the smart contract from an agreed-upon source. This raises two concerns. To begin, smart contracts cannot extract data from off-chain resources; rather, that information must be "pushed" to the smart contract.

Second, if the data is constantly changing, and the code is duplicated over various nodes throughout the network, different nodes may get different information, even if they are just a few seconds apart. In our example, Node-1 may get information indicating that the temperature is 31.9 degrees Celsius, whilst Node-2 may receive information indicating that the temperature is really 32 degrees Celsius. Given that unanimity among nodes is essential for a transaction to be verified, such variations might result in the criterion being declared "not fulfilled."

Contracting parties will be able to settle this quandary by consulting an "oracle." Oracles are trustworthy third parties that obtain off-chain data and then push it to the blockchain at predefined intervals. In the above scenario, the oracle would monitor the daily temperature, decide that a freezing event had happened, and then notify the smart contract.

Although oracles provide an easy mechanism for accessing off-chain resources, this procedure adds another entity with whom the parties must engage to carry out a smart contract, slightly weakening the decentralized advantages of smart contracts. It also raises the possibility of a "point of failure." For example, an oracle may encounter a system defect and be unable to send out the required information, supply incorrect data, or just go out of business. Before smart contracts may become more widely used, they must accommodate these possibilities.

What is the "Final" Agreement Reached by the Parties?

Courts will evaluate the final, written document to which the parties have agreed when examining conventional text-based contracts to determine whether the parties are in compliance or breach. Courts have always stressed that the ultimate agreement embodies the parties' common intent—the "meeting of the minds."

In the case of code-only smart contracts, the code that is executed—and the result it produces—represents the sole objective proof of the parties' agreed-upon conditions. In these instances, email conversations or oral talks between the parties about what functions the smart contract "should" perform would most likely provide the definitive code lines as the determinative embodiment of the parties' purpose.

In the case of auxiliary smart contracts, a court would most likely consider the text and code to be a cohesive single agreement. When the usually written agreement and the code do not concur, the situation gets difficult. Assume that the wording of an agreement says that an insurance payout will be paid if the temperature falls below 32 degrees and that the smart contract code triggers the payment if the temperature is equal to or below 32 degrees.

Assuming that the text agreement does not state whether the text or the code controls in the event of a conflict, courts will have to decide whether the code should be treated as a mutually agreed amendment to the written agreement or whether the text of the agreement should prevail—possibly on a case-by-case basis. In some ways, the approach should be similar to when the terms of a primary agreement diverge from what is contained in an attached schedule or exhibit. The fact that the disagreement here would be between text and computer code rather than two text documents should not be decisive, but courts may take a different perspective.

One option is for parties to employ a text-based contract in which the parameters that trigger smart contract execution are not only visible in the

text but also fill the smart contract. In our case, "less than 32 degrees" would appear not only in the text but would also be created as a parameter in the smart contract itself, reducing the possibility of discrepancy.

The Automated Nature of Smart Contracts

One of the most important characteristics of smart contracts is their capacity to conduct transactions automatically and indefinitely without the need for human involvement. However, this automation, as well as the fact that smart contracts cannot be readily altered or canceled unless the parties add such capabilities during the smart contract's construction, provide some of the most significant barriers to the broad adoption of smart contracts.

In typical text contracts, for example, a party may readily explain a violation merely by not enforcing the relevant sanctions. If a valued client is one month late with a payment, the vendor may make a real-time judgment that maintaining the long-term business relationship is more essential than any applicable termination right or late charge.

However, if this relationship had been reduced to a smart contract, the ability to not enforce the agreement on an ad hoc basis would most likely not have existed. If the smart contract was configured to do so, a late charge will be automatically deducted from the customer's account or access to a software program or an internet-connected gadget would be suspended. As a result, the automatic execution offered by smart contracts may conflict with the way many organizations function in the real world.

Similarly, in a text-based contractual relationship, a party may be ready to accept partial performance as complete performance on an ad hoc basis. This might be because of a desire to maintain a long-term relationship or because one side thinks that partial performance is better than none at all. Again, the impartiality necessary for smart contract programming may not accurately represent the facts of how contractual parties interact.

Are Smart Contracts Reversible?

Smart or blockchain contracts are distinct in that they guarantee two-party compliance. One of the most notable characteristics of a self-executing contract is its immutability. It implies that once the codes, rules, and transactions are put in the blockchain, they cannot be reversed, altered, or tampered with.

Because they are computer programs, the automatic implementation of the agreement occurs in accordance with the codes (contract terms) and regulations stated. However, if it is still necessary to update these codes and conditions, there are a few indirect methods that may be useful:

- Create an alternate contract containing transaction data from the previous contract, such as the address. As a result, any transaction initiated via the intermediate contract will be forwarded to the active one.
- Create a new contract version and copy over all of the previous contract's codes, conditions, and transaction data.
- Save the existing contract's logic code in a library and use it to get the current agreement's terms, regulations, and transaction data.

Smart Contract Modification and Termination

At the moment, there is no straightforward way to alter a smart contract, which presents some difficulties for contractual parties. In a classic text-based contract, for example, if the parties have mutually agreed to modify the boundaries of their commercial arrangement, or if the law changes, the

parties may swiftly prepare an addendum to accommodate that change, or simply adjust their course of activity.

Smart contracts do not presently provide this level of flexibility. Given the immutability of blockchains, updating a smart contract is significantly more difficult than altering normal software code that does not live on a blockchain. As a consequence, modifying a smart contract may result in greater transaction costs than changing a text-based contract, and it raises the risk that the parties may not precisely represent the changes they intend to make.

Similar difficulties occur when it comes to canceling a smart contract. Assume a party finds inaccuracy in an agreement that provides the counterparty greater rights than intended, or that completing its stated responsibilities would be significantly more expensive than anticipated.

In a text-based contract, a party may participate in, or threaten to engage in, so-called "efficient breach," which is deliberately violating a contract and paying the ensuing damages if the cost of performing is more than the damages it would owe. Furthermore, by suspending or threatening to cease performance, a party may bring the counterparty back to the table to seek an equitable conclusion. Smart contracts do not yet provide comparable self-help solutions.

There are now projects ongoing to develop smart contracts that may be terminated at any moment and are more readily changed. While this contradicts the immutable and automated nature of smart contracts in some aspects, it highlights the fact that smart contracts can only acquire commercial adoption if they mirror the business realities of how contracting parties operate.

The Difficulties of Integrating Specified Ambiguity Into Smart Contracts

The impartiality and automation needed by smart contracts may be incompatible with how business parties make agreements. During negotiations, parties participate in an implicit cost-benefit analysis, recognizing that there are decreasing rewards in attempting to conceive of and handle, every imaginable possibility.

These parties may no longer want to spend management time or legal fees on the negotiations, or they may feel that starting revenue-generating activities under a signed contract outweighs resolving outstanding difficulties. Instead, they can decide that if an unexpected incident happens, they will find a solution at that time. Similarly, parties may choose to intentionally keep a term in an agreement slightly vague to provide themselves the flexibility to argue that the clause should be read in their favor.

This technique of contracting is complicated by smart contracts, which need an exactitude not available in the negotiation of text-based contracts. A smart contract cannot include unclear phrases, nor can it leave certain hypothetical circumstances unanswered. As a consequence, smart contract parties may discover that the transaction costs of negotiating sophisticated smart contracts surpass the transaction costs of conventional text-based contracts.

It will take some time for those implementing smart contracts in a certain sector to establish whether requirements are objective enough to lend themselves to smart contract execution. As previously stated, most smart contracts now execute very basic operations where the parameters of the "if/then" expressions are obvious. As smart contracts get more complicated, parties may differ on whether a certain contractual condition can be represented via the objectivity required by a smart contract.

Do Smart Contracts Really Guarantee Payment?

One of the most often cited advantages of smart contracts is their ability to automate payment without the need for dunning notifications or other collection expenditures, as well as the necessity to go to court to acquire a judgment ordering payment. While this is correct for basic use cases, it may be less so in sophisticated business interactions. In reality, parties frequently move cash within their organizations and do not "park" whole sums owed on a long-term contract with the anticipation of future payment obligations. Similarly, a borrower is unlikely to maintain the whole loan amount in a wallet connected to the smart contract. Instead, the borrower will put those funds to use, financing the required repayments on an as-needed basis.

If the party owing sums under the smart contract fails to fund the wallet on time, a smart contract attempting to move money from that wallet in response to a trigger event may discover that the necessary funds are not accessible. Adding another layer to the process, such as having the smart contract seek cash from other wallets or having the wallet "fund itself" from other sources, would not fix the issue if those wallets or sources of funds lacked the necessary payment quantities as well.

The parties may try to address this problem by requiring that a wallet connected to the smart contract always contain a minimum amount, but that solution would only provide the side a better legal position if the disagreement was resolved. It would not completely automate the smart contract's payment function. As a result, although smart contracts will make payments significantly more efficient, they may not remove the requirement for payment disputes to be adjudicated.

Allocation of Risk for Attacks and Failures

Smart contracts present an extra risk that most text-based contractual agreements do not: the danger that the contract will be hacked or that the code or protocol has an unexpected programming mistake. Given the relative security of blockchains, both notions are closely related; specifically, most "hacks" linked with blockchain technology are really exploits of an unintentional code flaw. These mistakes, like many defects in computer code, are not noticeable at first and only become apparent after they have been exploited.

In 2017, for example, an attacker was able to drain \$31 million in ether from numerous multi-signature wallets provided by Parity. Because they need more than one private key to access the wallet, multi-signature wallets give an extra degree of protection.

In the Parity assault, however, the attacker was able to exploit a hole in the Parity technology by reinitializing the smart contract and becoming the sole owner of the multi-signature wallets. Parties to a smart contract must examine how risk and responsibility for unintentional coding mistakes and resultant exploitations are divided among the parties, as well as possible with any third-party developers or insurers of the smart contract.

Governing Law and Location

The establishment of powerful, decentralized, and worldwide platforms is one of the primary promises of blockchain technology and, by extension, smart contracts. However, due to worldwide acceptance, parties may use a smart contract in much more countries than in the case of text-based contracts. As a result, the party proposing conditions under a smart contract would be best served by defining the controlling law and venue for that smart contract.

A governing law provision defines which substantive law will be used for the smart contract's interpretation, while a venue clause states which

jurisdiction's courts will hear the dispute. Given the vast variety of countries in which a smart contract may be employed, a plaintiff may be somewhat unconstrained in deciding where to bring a claim or arguing which substantive law should apply in circumstances where controlling law or venue is not mentioned. Given that many early conflicts involving smart contracts would be of first impression, contractual parties will desire some assurance as to where such disputes will be resolved.

Best Practices for Smart Contracts

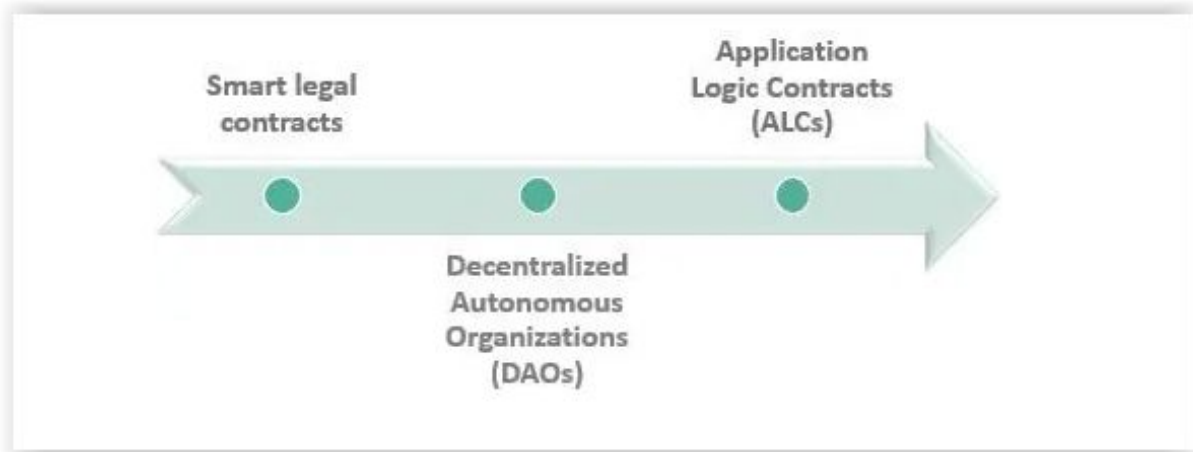
Given that smart contract, usage is still in its early stages, best practices for creating such code are continually improving. The checklist below, on the other hand, should assist developers in creating successful smart contracts and advise businesses that want to employ them.

- For the time being, parties entering into any form of contractual agreement would benefit from a hybrid approach that blends text and code. As previously stated, there are compelling reasons why code-only smart contracts should be enforceable, at least under US state contract law. However, until further clarification on their validity and enforceability is provided, code-only smart contracts should be utilized solely for smaller transactions. Text-versions of agreements will continue to be desired by parties to read the agreed-upon terms, commemorate terms that smart contracts are unable to address, and have a document that a court will enforce.
- In a hybrid contract that combines text and code, the text should clearly specify the smart contract code with which it is associated, and the parties should have full visibility into the variables being passed to the smart contract, how they are defined, and the transaction events that will trigger code execution.

- When depending on oracles for off-chain data, parties should consider what would happen if the oracle is unable to push out the required data, gives incorrect data, or simply goes out of business.
- In the case of a coding mistake, the parties should consider risk allocation.
- In the case of a controversy, the text agreement accompanying the code should define the controlling legislation and venue, as well as the order of precedence between text and code.
- Each party shall state in the text agreement that they have evaluated the smart contract code and that it follows the provisions of the text agreement. Although such a representation cannot compel a party to study the code, it may assist the counterparty in defending against a claim that the code was not inspected. Parties may also opt to protect against the possibility of faults in the code. As previously stated, parties may need to enlist the help of third-party specialists to evaluate the code.

Types Of Smart Contracts

There are three types of self-executing contracts based on their applications:



Source: [Smart Contracts](#)

#1 – Smart Legal Contracts

These contracts are legally binding and obligate the parties to carry out their contractual commitments. Failure to do so may result in severe legal consequences.

#2 – Decentralized Autonomous Organizations

These are blockchain communities bound by particular rules inscribed into blockchain contracts and governed by governance systems. As a result, each action done by members of the community is replaced by a self-enforcing code.

#3 – Application Logic Contracts

These contracts include application-based logic that keeps them in line with other blockchain contracts. It permits communication between diverse devices, such as the Internet of Things and blockchain technology.

A Technical Example of a Smart Contract

You must be intrigued about the potential connected with smart contracts after reading all of this. Let's have a look at an Ethereum-powered smart contract to get a better understanding of what smart contracts can do and accomplish.

On their [GitHub website](#), [Azure](#) provides an excellent variety of examples. Let's look at their asset-transfer scenario since we addressed asset transfer or ownership transfer before.

Aside from the buyer and seller, two other actors must be involved to guarantee appropriate management of high-quality assets: the appraiser and the inspector. The inspector is in charge of examining the assets before the buyer purchases them.

The appraiser is on the seller's side of the transaction. He makes assets valuable to purchasers. He also helps the seller with the sale.

Let's look at the graphic below to gain a better idea.



Source: Azure GitHub Sample Page

It also has several states that specify the smart contract condition. Currently, the smart contract that we will explain has ten states, which comprises.

Active
Offer placed
Pending inspection
Inspected
Appraised
Notional acceptance
Seller accepted
Accepted
Terminated

The process is complicated, which is why we won't go into it here. You may, however, read the `readme.md` file for the asset-transfer repository.

Smart Contract Use-Cases

Still, have questions about what smart contracts look like in the wild? There are several smart contract applications all over the place. But first, how about we start with your ordinary day-to-day life?

Smart Contracts in Action

Your Saturday begins with a fender bender. We never said it was going to be a glamorous ride, did we? Your bumper has a dent. You contact your insurance company and photograph the damage. You weren't the perpetrator, but you do have their contact information. Your details and crash data are logged into the insurance agent's blockchain-based system, triggering a provision in your contract. You get a notification. It has assessed your repair costs and provided you with a recommended service provider. The cost will be taken care of if you go there.

So you take your automobile to the mechanic. A new bumper has just arrived at the store, and ownership was transferred to the business using – you guessed it – a smart contract. Unfortunately, their supply of new air filters was delayed, but don't worry. The funds will not be sent to their supplier until the store has received them. Smart contracts and the supply chain are in motion.

I could go on, but I believe you get the idea.

In the early days of the cryptocurrency sector, smart contracts were largely utilized for games of chance like Roulette since they are (relatively) simple to implement and enable users to gamble their bitcoins without having to believe that the casino is necessarily fair (see: [provably fair](#)).

Developers have employed smart contracts for a wide range of beneficial, innovative, and even bizarre reasons since then, yielding some fantastic instances of how this basic technology may be used to disrupt today's businesses.

Smart contracts are being employed in almost every business, including safeguarding [land ownership records in Africa](#), increasing the [efficiency and openness of supply chains](#), blockchain-based voting, and distributing royalty awards.

There's even a smart contract that gives you access to your [cryptocurrency tokens](#) for a certain period, preventing you from selling or using them too soon and missing out on any possible price growth.

Smart Contracts and Blockchains In the Automobile Industry

There's no denying that we've progressed from sluggish pre-human vertebrates to super-smart robots. Consider a world in which everything is

mechanized. Google is making strides in this direction with smartphones, smart eyewear, and even smart automobiles. This is when smart contracts come in handy.

One example is self-driving or self-parking cars, where smart contracts may activate a type of 'oracle' that could determine who was at blame in an accident; the sensor or the driver, among a plethora of other factors. Using smart contracts, a car insurance business might charge various premiums depending on where and under what circumstances clients drive their automobiles.

Smart Contracts and Blockchains in Finance

Decentralized finance (DeFi) dApps are a strong alternative to conventional financial services, and their popularity is rising as a result of blockchain and smart contract technology's trustless, immutable, and transparent qualities. DeFi dApps offer complementary services to the banking and financial services industries, **such as lending**, borrowing, trading, and a variety of other financial services, as well as entirely new types of products and decentralized business models that can provide significant benefit and utility to users. With smart contracts' improved transparency (combined with 24/7 functioning and cheaper prices), dApps have the potential to decrease the barriers to entry into the financial services industry for individuals all over the globe.

DeFi ventures have already garnered billions of dollars in value and are expected to continue this trend as more individuals get acquainted with the sector's unique value propositions. Users can make use of this new generation of financial services without the requirement for centralized custody or intermediary costs. Although the DeFi sector is still in its early stages, the implications of smart-contract-powered dApps on the financial industry are already being felt, given the number of creative dApps that are currently offering value and usefulness to users.

Here are the stats to back it up:

DeFi's yearly transaction value increased 14x in 2020 and more than quadrupled to \$112.07 billion in 2021, with more than USD 20 billion in total value locked up in DeFi smart contracts alone. As of the time of writing, the total value locked is \$196.6B.



Data recorded on May 1, 2022, indicates the current total value locked in defi protocols is \$196.6 billion, according to defillama.com statistics.

DeFi smart contracts facilitate the exchange of commodities, services, data, finances, and so on. Users of centralized financial institutions, such as banks and credit unions, depending on third-party intermediaries to complete a transaction. DApps, on the other hand, use smart contracts to guarantee that each activity is real, transparent, and free of human mistakes.

Smart Contracts and Blockchains In Governments

Smart contracts enable government entities to carry out transactions, therefore promoting justice and democracy. Smart contracts based on the blockchain, for example, may make voting systems fully trustless and significantly more safe for governments.

Votes would be stored in a distributed register in this situation, and deciphering them would need remarkable computational power. Because there are now no computers capable of managing it, it is impossible to hack this system.

Tennessee has joined several other states in making blockchain technology legal. Tennessee's General Assembly [passed legislation](#) that makes blockchain signatures and smart contracts legally binding. On March 8, 2018, electronic signatures and digital contracts were granted the same legal status as conventional contracts.

Insiders confirm that our voting method is incredibly difficult to manipulate, but smart contracts would alleviate these worries by offering a far more secure system. Ledger-protected votes would have to be decrypted, which would take a lot of computational power. Because no one has so much processing power, God would be required to hack the system!

Second, smart contracts have the potential to increase low voter participation. Much of the inertia stems from a clumsy procedure that entails queuing up, displaying your identification, and filling out documents. Volunteers may transmit voting online via smart contracts, and millennials will show out in droves to vote for their President.

Smart contracts and blockchain technology are used by applications such as [FollowMyVote](#) to secure votes from fraud. The vote transaction cannot be modified after it is recorded on the blockchain. When the voting is finished, the smart contract will deliver a token to the address of the winner of the vote.

This ensures that voting is always fair and that the winner is always right.

Case Study

- For the 2020 US Presidential Elections, Utah County successfully collected votes from absentee voters using Voatz, a blockchain-powered smartphone application. The voter's identity is concealed behind a unique signature/hash-value, and the essential data is likewise kept private and free of external threats inside the blockchain.

Smart Contracts And Blockchains In Business Management

Smart contracts may be very beneficial to businesses. Smart contracts may be used instead of paying employees to administer payrolls.

Because of its precision, transparency, and automated system, the blockchain not only offers a single ledger as a source of confidence but also eliminates any snags in communication and workflow.

Normally, company activities must undergo a back-and-forth while awaiting approvals and resolving internal or external concerns. This is made easier by using a blockchain ledger. It also eliminates inconsistencies that are common with separate processing and may lead to expensive litigation and settlement delays.

Businesses may simply create a smart contract that states WHEN the date is March 28, 2023, the business transfers John 2 ETH. This implies that John will never be underpaid and will always be paid on time. The company

benefits since everything is automated, saving them a lot of time and money!

Case Study

- The Depository Trust & Clearing Corp. (DTCC) employed a blockchain ledger in 2015 to handle more than \$1.5 quadrillion in securities, totaling 345 million transactions.

Smart Contracts and Blockchains in Initial Coin Offerings (ICOs)

If you wish to launch your own blockchain project, as we saw before, you may do it on the Ethereum blockchain. You will, however, need funds!

How are you going to receive the funds you require? Welcome to the world of initial coin offerings (ICOs).

An ICO ([Initial Coin Offering](#)) is a kind of crowdfunding system for new blockchain-based apps. You construct a smart contract as well as a token for it. Assume you've named your token ABC.

You want to fund \$10,000,000 to begin your project and construct your application - let's say \$10,000,000 equals 10,000 Ether. You elect to deposit 100,000 ABC tokens into the smart contract, with each ABC token worth 0.1 Ether.

As a result, if you sell all 100,000 ABC tokens, you will acquire the necessary 10,000 Ether since $100,000 \times 0.1 = 10,000$.

Now, under the smart contract, write something like this: IF 0.1 ETH is submitted to the smart contract, THEN the smart contract will transfer 1

ABC to the address that supplied the 0.1 ETH. As a result, everyone who contributes to the ICO receives the correct quantity of ABC coins.

Remember! It is critical to store your cryptocurrency in safe wallets. The most popular choices are the Ledger Nano S, Coinbase, and Trezor.

Why would anybody want to purchase the ABC token?

The two most prevalent motivations for purchasing tokens from ICOs are:

1. Once the application is developed, the token may be utilized.
2. As the initiative grows in popularity, the token's price may rise.

Consider ICOs to be the blockchain equivalent of [Kickstarter](#). The significant distinction is that it securely automates the whole crowd-sale process.

Smart Contracts and Blockchains In Rights Management (Tokens)

Token smart contracts are used on blockchain networks to generate, monitor, and grant ownership rights to particular digital tokens. The token contract embeds functionality into the tokens it issues, giving holders access to features such as utility/insurance in a dApp (utility token), voting weight in a protocol (governance token), equity in a company (security token), ownership claim to a unique real-world or digital asset (non-fungible token), and more. The FIL token, for example, is used to pay for Filecoin's decentralized storage services, while the COMP token enables users to participate in Compound protocol governance.

Blockchains and Smart Contracts Increasing the level of trust in retailer-supplier interactions

Smart contracts operate on the If-Then basis, therefore, in the words of [Jeff Garzik](#),

"UPS may execute contracts that say, 'If I collect cash on delivery at this place in a growing, emerging market, then this other [product] will trigger a supplier generating a new item since the previous item was just delivered in that developing market.'"

All too frequently, supply chains are impeded by paper-based systems, in which paperwork must be approved via many channels, increasing the risk of loss and fraud. The blockchain eliminates this by giving all stakeholders on the chain a secure, accessible digital version and automating chores and payment.

Case Study

- When a change of ownership occurs, Barclays Corporate Bank utilizes smart contracts to track it and automatically transmit payments to other financial institutions.
- The Home Depot employs blockchain smart contracts to efficiently settle vendor issues. They are strengthening connections with suppliers via real-time communication and better insight into the supply chain, resulting in more time for vital work and innovation.

Making international trading more efficient and quicker

- Businesses are building a trusting environment for global commerce by joining we.trade, the trade finance network formed by IBM Blockchain. We.trade, a blockchain-based platform, provides uniform rules and streamlined trading choices to decrease friction and risk while simplifying the trading process and boosting trade prospects for participating enterprises and institutions.

Smart Contracts And Blockchains In NFTs - Gaming Technology

The worldwide gaming business is a multibillion-dollar ecosystem that is rapidly expanding, yet the way wealth is produced and dispersed within the sector may be inequitable.

NFTs have surged in the market in only a few years, with the market value closing up on a staggering \$40.9 billion in 2021 as they proved to be the most successful use-case of smart contracts.

Developers create and distribute games, and players pay to play and interact with them. This creates a one-way flow of value in which players pay to get access to in-game materials and gameplay options. In contrast, blockchain technology in gaming may help gamers more efficiently capture the usefulness and value of in-game transactions and asset acquisitions.

Non-fungible tokens (NFTs) – unique digital assets that reflect in-game content — are often used to drive blockchain technology in gaming. Smart contracts are used in NFTs. These tokens are one-of-a-kind, scarce, and indivisible, while the blockchain networks that support NFTs provide player ownership, proved scarcity, interoperability, and immutability. Together, these blockchains in gaming qualities have the potential to promote widespread adoption and a more equal value model.

Smart contracts enable the implementation of a sale agreement between the owner of the NFT and the buyer. The smart contract includes information about the NFT, such as the author of the work, other parties who are entitled to royalties when the NFT is sold, and the work's ownership history.

The bulk of NFTs is not recorded on the blockchain since it is both expensive and energy-consuming to store that much data on the blockchain. As a consequence, smart contracts usually contain a link to the work they represent, which is exclusively accessible to the owner.

Smart contracts are used in blockchain-based games to provide tamper-proof execution of in-game operations. PoolTogether, a no-loss savings game in which players stake their cash in a communal pool, which is subsequently channeled into a money market where it generates interest, is one example.

The game concludes after a certain period, and a winner is chosen at random to get all of the collected interest, while everyone else may withdraw their initial payment. Similarly, limited-edition NFTs may employ fair distribution techniques, and RPGs can use randomization to provide unexpected loot drops, ensuring that all players have a fair chance of obtaining rare digital goods. Many applications utilize Chainlink Verifiable Random Function (VRF) to access randomness—a random number generator (RNG) that employs cryptography to ensure its per-proof, which means the RNG process is publicly auditable.

You may preserve in-game purchases, sell them to other players, or transfer them to other supported games thanks to the deployment of blockchain technology in the gaming business. Meanwhile, the scarcity of in-game NFT purchases may be proven using the immutable records included in an NFT's underlying blockchain network, as can its ownership history.

Blockchain-built games and dApps can extend gaming economies, introduce new gaming categories, and stimulate the creation of new games since NFTs are unique and can be designed to maintain value beyond the game in which they originated. Ethereum, TRON, EOSIO, and NEO are among the blockchains that have seen substantial game development.

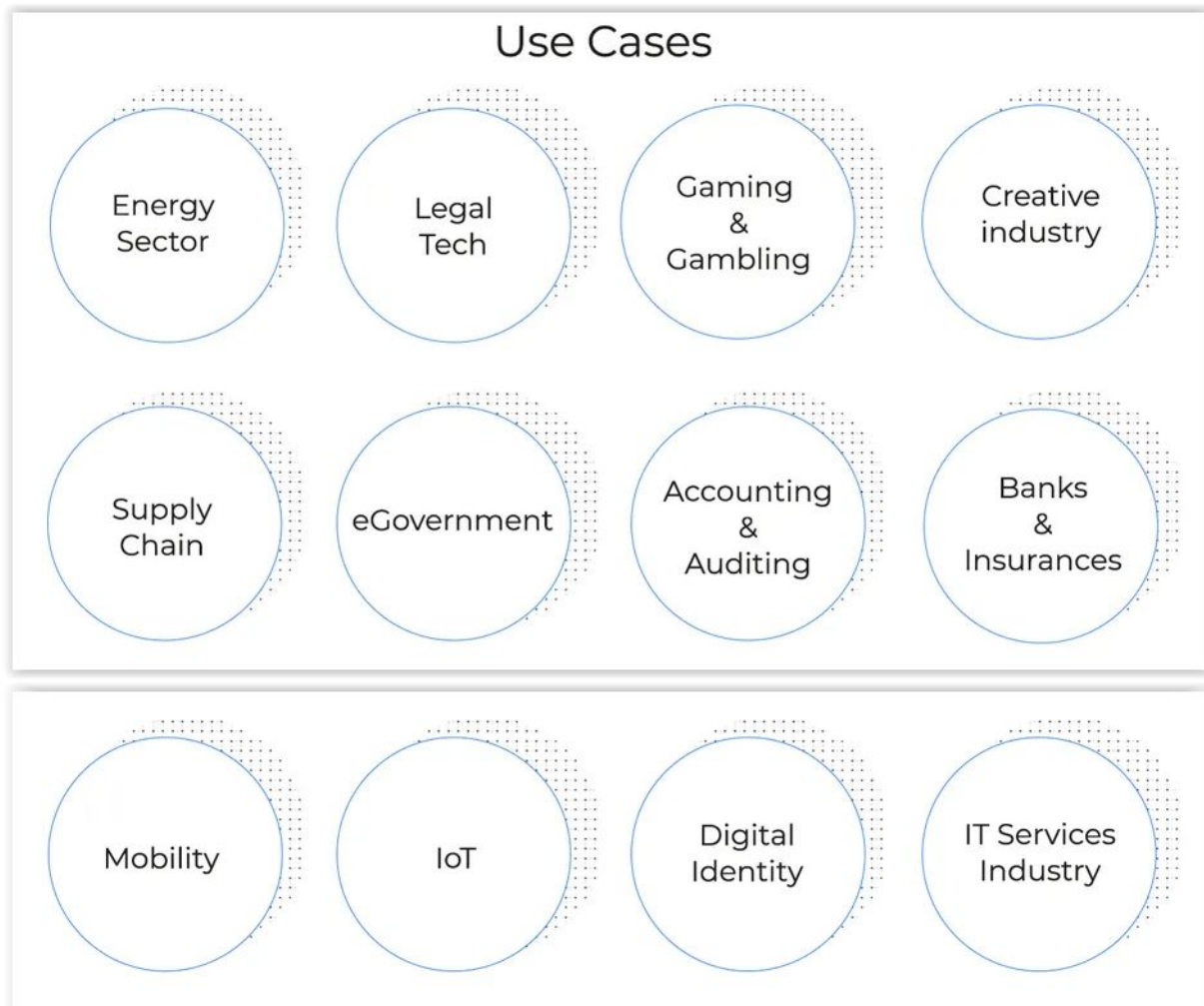
Smart Contracts and Blockchains in the Legal Industry

One of the most intriguing real-world smart contract use cases is their ability to serve as legally enforceable contracts — the kind that governs the majority of today's corporate transactions. Technology has driven innovation in the legal business, most recently with the introduction of e-signatures for legally enforceable agreements. Another new development in this sector is smart contracts, which may soon be an alternative for parties to legal agreements, possibly cutting the expenses paid by engaging attorneys and other middlemen.

The widespread adoption of customized smart contracts for a broad range of activities, may reduce costs and speed up transactions, may be closer than you think. Some jurisdictions in the United States have started to allow the use of smart contracts and blockchain in the legal business under limited circumstances.

Smart contracts may lead the way in the legal sector when document attestation is necessary. They can decrease the requirement for notarization while also providing a cost-effective and impartial alternative.

In Arizona, for example, smart contracts may be used to make legally binding agreements, while in California, marriage licenses can be produced using blockchain technology.



[SOURCE: BLOCKCHAINHUB.NET](http://BLOCKCHAINHUB.NET)

Smart contracts and Blockchains in Real Estate

Smart contracts may be used in the real estate market to sign agreements between parties interested in purchasing, selling, or renting real estate.

If you wanted to rent out your flat, you'd have to pay an intermediary, such as Craigslist or a newspaper, to promote it, and then you'd have to pay someone to confirm that the individual paid rent and followed through.

A decentralized solution may help you save money. All you have to do is pay with cryptocurrencies and have your contract encoded on a smart contract. Everyone notices and you achieve automatic satisfaction. Brokers, real estate brokers, hard money lenders, and everyone else involved in the real estate business stand to benefit.

By combining blockchain with real estate transactions, smart contracts are promoting fractional ownership of properties and, as a result, decreasing the barrier to entry for many investors. There have been several successful ventures in tokenizing real estate assets, including systems like RealT and SolidBlock that combine blockchain with real estate.

By adding blockchain to real estate deals, smart contract technology may also remodel the paperwork and transaction procedures. Since 2016, the Republic of Georgia (in the Caucasus area) has been creating a blockchain-based land title register, and similar initiatives are ongoing in other jurisdictions such as the United Arab Emirates (UAE).

Anyone who has bought a house or other property is probably aware of the possibility of hidden charges associated with closing fees, title transfers, and broker fees. These are expenses that might be lowered, if not eliminated, by automatically executing smart contracts that operate without the need of middlemen.

When a piece of property is tokenized, smart contracts may handle most of the needed record-keeping, saving the parties time and money. According to some experts, smart contracts may benefit parties by simplifying rental agreements, complicated credit or mortgage arrangements, warranties, and insurance. The use of smart contracts and blockchain in real estate reduces the need for legal advice or other advisory services, possibly lowering overall expenses.

Smart contracts are gradually supplanting conventional contracts as the only means of an agreement between seller and customer. It automatically performs the requirements as soon as certain contract conditions are satisfied.

By building trust, smart contracts provide trust via a single version of the truth. Smart contracts allow all parties, including the bank, the agent, and the mortgage lender, to sign an agreement. Because transactions are stored on a blockchain, the parties involved may inspect the process at any time and from any location.

[Propy](#), for example, was one of the first real estate businesses to embrace smart contracts. Their first transaction occurred in September 2017, when a \$60,000 flat in Ukraine was acquired. Owners and brokers may advertise homes on this real estate marketplace. Buyers can seek and bargain. Both buyers and sellers engage in the smart contract, and both parties take certain actions throughout the process to guarantee fair and lawful play.

Smart Contracts and Blockchains in Corporate Structures - Building DAOs

Delaware approved Senate Bill 69 in 2017, which permits enterprises to be formed and operated using blockchain technology. This measure paved the way for the growth of decentralized [autonomous organizations \(DAOs\)](#), which operate as companies with ownership and remuneration embedded into smart contracts.

DAOs may provide complicated, automatically enforced incentive systems inside a corporate framework by encoding corporate structures with smart contracts. DAOs may also save money on administrative expenditures like as office space, recruiting, and payroll by using incentive systems that do not need formal employment contracts.

Smart Contracts and Blockchains in Emerging Technology

The capacity to assist difficult computational activities such as those involved in machine learning and artificial intelligence is one of the most promising uses of blockchain technology and accompanying smart contract technologies (AI). AI-powered smart contracts have the potential to be created by combining the data-intensive processing of AI with the decentralized security and immutability of blockchain technology.

Smart contract apps will need to grow more complicated as they are applied across multiple sectors to fit their expanded duties. While simple smart contract use cases may be created manually, AI-enabled smart contracts may enable the creation of extremely sophisticated, more responsive, enterprise-grade smart contracts and dApps with the potential to greatly increase the technology's capabilities.

Many experts believe that the domains of AI and blockchain may benefit from each other's distinguishing features. Smart contracts may benefit from AI technology's superior computing capabilities and adaptive systems, while AI implementations can use smart contract technology for autonomous rule execution and to create a safe environment for sensitive and important machine learning data to reside.

With its unique smart contract programming language, Scilla, and powerful parallel processing structure enabled by sharding, Zilliqa is one of several blockchain platforms exploring extraordinary computing capabilities.

Smart Contracts and Blockchains In Insurance Companies

A metric insurance policy is one in which the payoff is directly linked to a preset event. Smart contracts offer a secure foundation for constructing parametric insurance contracts that are triggered depending on data inputs.

Crop insurance, for example, may be constructed using smart contracts, in which a customer obtains a policy based on precise meteorological data, such as seasonal rainfall in a geographic region. The smart contract will automatically provide payment at the conclusion of the policy if the quantity of rainfall in the given place exceeds the initial declared amount.

Not only can end-users obtain timely reimbursements with lower overhead, but the supply side of insurance may also be made available to the public through smart contracts. The smart contract enables users to deposit monies into a pool and subsequently distributes collected premiums to pool members depending on the proportion of their pool participation.

In 2017, two insurance firms, Atlas Insurance in Malta and Axa in France, experimented with using smart contracts. They had prototypes that rewarded airline passengers who experienced flight delays.

Here's an example:

John is set to board a flight from New York to Los Angeles. He pays \$5 in cryptocurrencies to the Axa Insurance smart contract and gives them his flight number. Axa delivers a payment of \$95 to the smart contract. So the smart contract contains \$100.

If John's flight arrives on schedule, the smart contract pays Axa \$100. However, if the aircraft is late, the smart contract sends \$100 to John. Everything runs on autopilot.

This saves a significant amount of time and money. It also means that John does not have to trust AXA to pay him the agreed-upon amount if his flight is delayed – if it is delayed, the smart contract will immediately give him his compensation (\$100).

Smart Contracts and Blockchains in Finance

In finance, these contracts may aid in the simplification and acceleration of many financial services. Insurance firms, for example, may use them to generate legal agreements and pay disputes. Similarly, stock exchanges may establish [securities trading](#) rules in these contracts to issue bonds that are compatible with regulatory requirements. Similarly, banks may use these contracts to handle syndicated loans more quickly and lower operational risks.

Smart Contracts And Blockchains In Powering DEFI

Smart contracts are powering a completely new variety of businesses that were just not viable with traditional contracts, in addition to resolving some of the major frictions of current sectors.

One of these businesses is referred to as 'decentralized finance,' or 'DeFi,' and it is effectively a full ecosystem of decentralized financial services like trading platforms, interest-bearing accounts, [stablecoins](#), and insurance protocols.

[Decentralized finance \(DeFi\)](#) applications use smart contracts to recreate traditional financial products and services such as money markets, options, stablecoins, exchanges, and asset management, as well as combine multiple services to create new financial primitives through permissionless composability. The smart contract may escrow user cash and distribute them to users depending on predetermined circumstances. BarnBridge, for example, employs smart contracts to automate trades for customers who desire fixed asset exposure to a price pair (e.g., 45 percent token A, 55 percent token B), while Aave uses smart contracts to simplify lending and borrowing in a permissionless and decentralized way.

Have you ever heard of a [stablecoin](#), a decentralized exchange, or a yield farm? Yes, all of them are instances of DeFi!

Most of these DeFi platforms are accessible through a [decentralized application \(dApp\)](#), which makes engaging with the underlying smart contracts easier – similar to how your favorite food delivery app allows you to order Pizza without having to interface with any of the underlying code.

[Check out our brief introduction of DeFi here to learn more about how it is altering finance.](#)



Smart Contracts and Blockchains In Healthcare

Another area that has started to use blockchain technology for safe, trustless, and transparent data exchange in healthcare. The integration of smart contracts and whole [dApps](#) intended to tackle critical healthcare pain points such as interoperability, identity, and authentication difficulties may help strengthen the interaction between healthcare and blockchain technology.

Smart contracts in healthcare assist to optimize insurance trial procedures, provide access to cross-institutional data, and provide patients with assurance that their sensitive data is protected.

Personal health records might be encoded and kept on the blockchain using a private key that only particular persons would have access to. A similar

technique might be used to guarantee that research is done in accordance with HIPAA regulations (securely and confidentially).

Surgery receipts might be maintained on a blockchain and automatically delivered to insurance companies as evidence of delivery. The ledger might also be leveraged for general healthcare management, such as drug supervision, regulatory compliance, testing findings, and healthcare supply management.

Consider the following example:

Smart contracts are already being utilized in the medical business by companies such as [EncrypGen](#). This is an application that utilizes smart contracts to securely send patient data, preventing third-party access.

Patients are in charge of their own data in this manner. Patients' data must be paid for if researchers wish to use it. Not only that, but the patient must decide whether or not to sell it to them.

Another example is the [Robomed Network](#), a decentralized medical network that creates its own coins to facilitate smart contracts between healthcare providers and patients. Robomed Network is currently under beta testing and employs the Robomed EHR (Electronic Health Record). Based on Ethereum blockchain smart contracts, this solution enables users to connect and govern their collaboration. The digital contract tracks all contacts with patients, efficacy indicators, and patient perceptions of exchanges with care providers.

Smart Contracts and Blockchains In Other Industries

This is by no means an entire list of real-world smart contract uses, and smart contract technology has the potential to benefit many additional

sectors long into the future. Many researchers and developers are keen to take advantage of smart contract technology to meet the demands of the expanding Internet of Things (IoT). While blockchain technology, in general, is currently being utilized to provide security and transparency to IoT devices, the benefits of smart contracts may accelerate this integration.

Smart contracts and dApps are primed to continue transforming the world of digital agreements, with all of these documented use cases and the continuous discovery and creation of many more.

What Smart Contracts Can Give You

Here's what smart contracts give you:

Autonomy

You are the one who makes the agreement; there is no need to depend on a broker, lawyer, or other intermediaries to validate it. In addition, since execution is controlled automatically by the network rather than by one or more potentially biased humans who may blunder, the risk of manipulation by a third party is eliminated.

Trust

Your papers are encrypted and stored on a distributed ledger. Someone cannot claim to have lost it.

Backup

Consider what would happen if your bank lost your savings account. On the blockchain, every single one of your buddies has your back. Your papers are replicated several times.

Safety

Your papers are protected thanks to [cryptography](#), which encrypts websites. There is no tampering with it. In reality, cracking the system and infiltrating would need an exceptionally gifted hacker.

Smart contracts ensure the safety of a blockchain network. Although the technology itself is capable of providing enough security to avoid any hacks, developing a smart contract also accounts for assuring its safety.

A simple flaw in a smart contract, similar to what occurred with The DAO attack in 2016, resulted in the crypto market's greatest robbery. It could have been avoided if that loophole had been corrected sooner. But here's the catch: since every transaction on a blockchain can be tracked, the moment the stolen ether/ETH enters circulation, those behind the robbery will be revealed. As a result, all of the stolen cryptocurrency is worthless.

A minor human mistake in the creation of a smart contract jeopardized its security. To avoid this, you'll need the appropriate engineers to create a foolproof smart contract.

Speed

Manually processing papers would normally take a significant amount of time and paperwork. Smart contracts employ software code to automate functions, reducing the time it takes to complete a variety of business procedures.

Savings

Smart contracts save you money because they eliminate the need for a middleman. You may have to pay a notary to witness your transaction, for example.

Accuracy

Contracts that are automated are not only quicker and less expensive, but they also eliminate the mistakes that might occur when filling out a plethora of forms by hand.

How Are Smart Contracts Created?

Smart contracts may be developed on a variety of blockchain systems, such as Ethereum and NEO. Because Ethereum is the most popular platform for developers, I'll go through Ethereum's smart contracts.

[Solidity](#), Ethereum's original coding language, is used to create smart contracts.

If you want to study Solidity, check out any online interactive Solidity lessons. You will likely find It's a nice and enjoyable method to learn Solidity. You will construct your own Solidity game by following the procedures in the courses.

Make Your Very Own Smart Contract!

To construct a smart contract, you must first grasp blockchain technology and how smart contracts function. You will also need to be familiar with the programming language used to create smart contracts, which is often Solidity.

Once you've mastered this information, you'll be able to design your own smart contracts. Many blockchain networks enable the creation and deployment of smart contracts.

Ethereum is one of the most popular platforms.

Ethereum is a public blockchain platform that enables the creation and execution of smart contracts.

The procedure of building a smart contract on Ethereum is rather straightforward. To begin, you must establish a new account on the Ethereum network. After you've registered an account, you may start creating smart contracts. The following is the procedure for constructing a smart contract:

- Make a new contract file.
- In Solidity, write the smart contract code.
- Distribute the smart contract on the Ethereum network.

When you deploy your smart contract, it is stored on the Ethereum blockchain and executed by the Ethereum network.

Are Smart Contracts Secure?

To summarize all of the facts, knowledge, and examples we've shared regarding smart contracts, are incredibly safe.

Aside from the benefits of automated formulation and activation when the contract criteria are satisfied, protecting the anonymity of both parties, the fact that Smart Contracts are maintained using blockchain technology makes them incredibly safe. It is also beneficial due to its security and decentralized nature. Our data is secure using this technique since we do not share it with other parties.

Why Should You Have Faith In Smart Contracts?

Because smart contracts operate inside blockchains, they are immutable and distributable, much like the blockchain itself.

A digital contract is immutable if it cannot be modified, tampered with, or broken.

The term "distribution" refers to the need that a contract to be approved by all parties in the current network. The agreement is protected by distribution, and no adversary may release cash.

Blockchain Networks Using Smart Contracts

Smart contracts are used in several blockchain networks and initiatives. Ethereum is one of the most well-known.

What Are Ethereum Contracts And How Do They Work?

[Ethereum](#) is a decentralized platform for smart contracts. It is secure, and no one and nothing can obstruct its operation. The Ethereum blockchain database holds transactions between users, smart contract operations, and their source code.

The Solidity programming language is used to create smart contracts that use ETH. This language facilitates the creation of self-enforcing smart contracts that operate on the Ethereum Virtual Machine.

Within the decentralized database, smart contracts exist as bytecode. This is the source of Ethereum's innovation and revolutionary potential.

Interesting Research

Smart contracts may help to speed up research. NASA, for example, has funded a research study that uses the Ethereum smart contract system to automate space technology moving to avoid space debris. Dr. Jin Wei Kocis, Associate Professor in the Department of Electronics and Computer Engineering at the University of Akron, is leading the project, titled Resilient Networking and Computing Paradigm.

NASA provided Kocis's team with \$330,000 funding to carry out the pilot experiment. The team intends to create a cognitive model based on smart contracts over the next three years. This model does not rely on data from Earth, but rather on data from space. Ethereum-based smart contracts will train space shuttles on how to avoid littering the planet.

“The Ethereum technology will be used to create a decentralized and safe network. The technology will also be used to create a calculative infrastructure to explore deep

space. Additionally, we plan to study consensus protocols of blockchain to increase the stability of the infrastructure.”

– Dr. Kocis

NASA officials believe that this project can make use of the potential of **decentralized ledgers/registers** to develop next-generation space networks.

Other Platforms And Initiatives

[NEO](#) is a non-commercial blockchain initiative that was established in China in 2014. It facilitates the growth of a decentralized "smart economy." To meet stated requirements on the NEO network, smart contracts employ virtual machines (VMs) that automatically optimize the digital contract code before launching it and arrange it so that it runs as efficiently as possible. In the long term, this strategy will be more efficient, albeit code restructuring takes longer to run and execute than in Ethereum.

[Nxt](#) is a decentralized open-source platform for the development of secure DApps such as electronic payment systems, instant messengers, and trading platforms. The platform went live in November of 2013. Its purpose was to create its own coins on the Nxt network that could be used indefinitely. Furthermore, the Nxt platform provides a restricted collection of digital contract templates, but users cannot execute their own smart contracts.

[Ubiq](#) is a decentralized platform with open source code for automatically establishing and deploying smart contracts and DApps. The platform went live in September of 2014. It replaced the Jumbucks network with the UBIQ blockchain, which is based on Ethereum, in January 2017. The Ubiq project is focused on offering companies automated smart contracts with high bandwidth, while developers pitch the platform as a supercomputer to work with the blockchain.

[Exonum](#) is a blockchain project framework. Exonum enables organizations or governments to construct a Blockchain system that addresses many difficulties safely and simply. Its encryption is based on the Bitcoin Blockchain. It also provides the most appealing characteristics of the technology, such as transparency and smart contracts. Exonum is built on Rust, which is one of the most secure programming languages available today.

What Is Hybrid Smart Contracts?

Hybrid smart contracts combine code running on the blockchain (on-chain) with data and computation supplied by Decentralized Oracle Networks (off-chain). Hybrid smart contracts allow sophisticated forms of economic and social coordination by using secure off-chain oracle services to achieve new features like scalability, secrecy, order fairness, and connectedness to any real-world data source or system.

In this part, I'll outline the importance of hybrid smart contracts in evolving blockchain-based trust models and highlight the many decentralized services that Chainlink oracles provide to increase their capabilities. You can then see how this finally opens the door to a new generation of hybrid blockchain-based apps with the necessary real-world qualities to enhance how society works across practically every major sector in the future.

How Oracles Are Extending Blockchain Collaboration

Blockchains, at their core, are computer infrastructure intended to allow a single important function: extremely trustworthy cooperation. Trust is what provides members a deep conviction in the collaboration's dependability, honesty, competence, or strength.

A contract, which outlines each participant's legal and commercial duties as well as the penalties/rewards of their behavior, is the most typical technique to develop trust in a collaborative process. Unfortunately, today's contract enforcement mechanism is far from perfect, especially when one participant has an asymmetrical advantage, such as unfair influence over the enforcement infrastructure, a better understanding of the fine print, or the time and capital to prolong the arbitration process.

As a consequence, believing in a counterparty's brand has become important to assess their trustworthiness in a contract system.

Blockchains are a collaboration-enabling technology that replaces brand-based trust with math-based trust by moving a contract's hosting, execution, enforcement, and custody procedures to software logic operated over a decentralized network that no one participant can undermine.

Blockchains are very trustworthy because they are enclosed networks, purposefully confined to allowing a very narrow, established range of cooperation kinds that are straightforward to enforce, such as moving tokens between addresses on a self-contained ledger, similar to a computer without Internet.

While this isolation and restricted spectrum of capability [yield](#) the tamper-proof and deterministic assurances that make blockchains useful, it also precludes support for any form of cooperation that involves data, computation, or features that are not local to the given blockchain.

The drive to broaden the sorts of cooperation possible on blockchains resulted in the development of oracles and, later, the creation of hybrid smart contracts. Oracles offer blockchains with safe portals to the outside world, allowing smart contract applications to validate external events, activate actions on other systems, and exploit calculations that would be impossible or impractical to do on-chain.

The off-chain services provided by Decentralized Oracle Networks (DONs), as detailed in the [Chainlink 2.0 Whitepaper](#), considerably broaden the sorts of on-chain partnerships that smart contracts may allow. This is already evident in the rapid rise of [Decentralized Finance \(DeFi\)](#), which accelerated once Chainlink's decentralized oracle networks made external financial market data available on-chain, enabling hybrid smart contract protocols such as Aave's money markets, Synthetix's derivatives platform, dYdX's leveraged trading markets, Ampleforth's algorithmic stablecoin, and many more.

The Structure of Hybrid Smart Contracts

A hybrid smart contract is an application made up of two parts:

- 1) smart contract—code that only runs on the blockchain; and
- 2) decentralized oracle network(s)—secure off-chain services that help the smart contract.

The two components work together smoothly and securely to produce a single hybrid smart contract application. As a consequence, on-chain code is supplemented in several unique and significant ways, enabling numerous new use cases that would not be conceivable with on-chain code alone owing to technical, legal, or budgetary restrictions.

Hybrid smart contracts synchronize two completely different computing environments to produce a greater application that neither a blockchain nor an Oracle network could accomplish on their own, especially when one environment specializes in delivering characteristics that the other does not. On-chain code executes in an exceptionally secure and limited-functionality blockchain environment with a small attack surface area, providing users with a high level of execution and storage determinism—the code will

execute precisely as intended, and results will be permanently and immutably saved. DONs, on the other hand, operate off-chain and so provide significantly greater functionality flexibility and data accessibility.

It's crucial to note that DONs still offer a very high degree of tamper-resistant and dependability to meet the smart contract commitments, but they do so in an isolated off-chain environment utilizing a variety of different security mechanisms.

Each DON delivers a specialized decentralized service to a single application, which means that the performance of other smart contracts on the same blockchain is neither linked to that DON's performance nor is the underlying blockchain consensus process that protects all smart contracts jeopardized.

DONs, as separate services, are not only beneficial in terms of security, but they also provide the freedom required to verify and compute in an immensely more complicated and open-ended off-chain environment.

For example, one smart contract may only use a DON for its specific external data needs if it is highly decentralized and backed by a significant amount of crypto-economic security, whereas another smart contract may prefer a DON with a more specific set of highly reputable nodes that use advanced cryptographic techniques to perform private verifiable computation.

Thousands to millions of DONs can run in parallel without cross-dependence to provide purpose-built decentralized services to specific applications on such heterogeneous network architecture, though some users may share the costs of the same DON service (e.g., numerous [DeFi protocols](#) currently use and fund the [Chainlink ETH/USD Price Feed Oracle](#)).

This framework is critical for meeting the needs of all blockchains and applications at the same time, such as applications running on a high-speed blockchain that require external data and privacy, as well as applications running on a highly decentralized blockchain that require scalable computation.

How Do Hybrid Smart Contracts Combine On- and Off-Chain Computation?

To further appreciate the distinction between on-chain and off-chain components, consider the following:

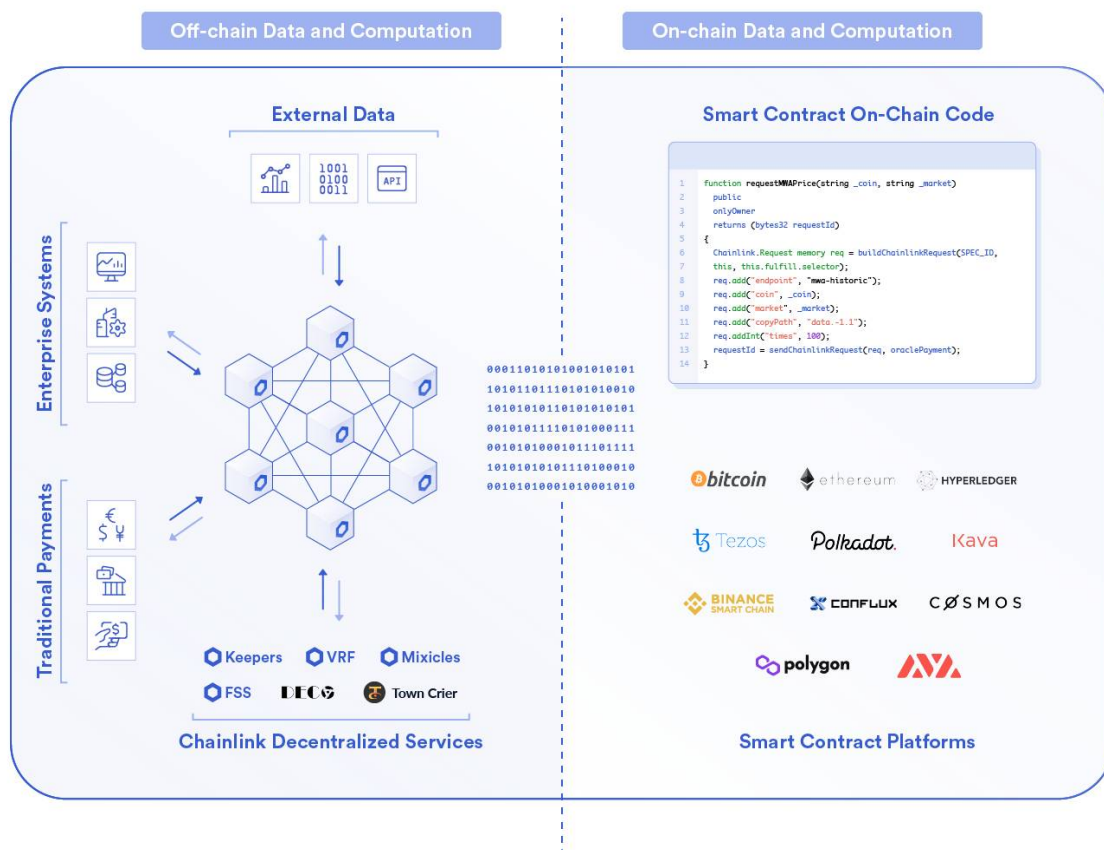
On-Chain: Blockchain

- Keep a permanent ledger that has authoritative custody of user assets and interacts with private keys.
- Carry out an ultimate settlement by performing irreversible transactions that transfer value between users.
- Provide dispute resolution and safeguards to ensure the correct operation of a DON's off-chain services.

Off-Chain: Decentralized Oracle Network

- Retrieve, verify, protect, and distribute data from external APIs to smart contracts operating on blockchains and Layer-2 technologies.

- Execute a variety of calculations for smart contracts operating on blockchains and Layer-2 solutions.
- Forward smart contract code outputs to other blockchains or other platforms.



Hybrid smart contracts combine on-chain code with off-chain decentralized oracle networks to enable more advanced blockchain-based applications. Source:Chain.Link

Chainlink Decentralized Services That Power Hybrid Smart Contracts

Now that we've defined hybrid smart contracts, let's look at the various decentralized services offered via Chainlink DONs that may dramatically improve a smart contract. Off-chain data and off-chain computing will be the two primary kinds of decentralized services.

DONs for Off-Chain Data may be used to connect many sorts of external data to and from blockchains, allowing hybrid smart contracts to be developed around those particular pieces of data. Among the first data kinds made available were:

- Price Feeds – asset price data compiled by hundreds of exchanges, weighted by volume, and free of outliers and wash trading.
- Proof of Reserve – current data on the reserve balances that back tokenized assets, such as the BTC reserves that back WBTC or the USD bank account that backs TUSD.
- Any API – premium data from password-protected APIs ranging from weather predictions and sports match outcomes to business backend and IoT network statistics.
- Blockchain Middleware – a layer of abstraction for an off-chain system that reads and writes data to and from smart contracts on any blockchain network.

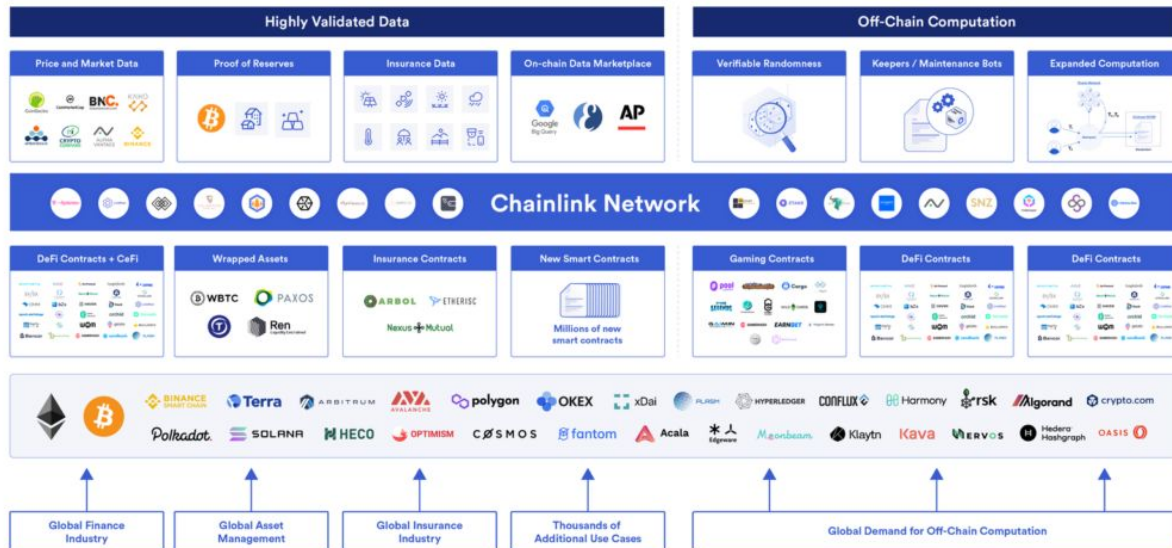
Off-Chain Computation

DONs may conduct a range of off-chain calculations on behalf of the smart contract to assist it in achieving specified inputs or generating characteristics that are not achievable on the smart contract's individual

blockchains, such as privacy, scalability, and order fairness. Some of the present and prospective off-chain calculations enabled by DONs are as follows:

- Keeper Network – automated bots that execute routine maintenance activities on the smart contract, waking it up when crucial on-chain services are required.
- Off-Chain Reporting (OCR) – the scalable aggregate of Oracle node replies in a DON, which is subsequently provided on-chain in a single transaction to decrease on-chain expenses.
- Scalable Computation – high-throughput, low-cost contract execution for independent smart contracts that sync on-chain regularly using current layer-2 technology.
- Verifiable Randomness Function (VRF) – safe and verifiable random number generation supported by cryptographic evidence of process integrity.
- Information and computation Privacy — Oracle computing that makes sensitive data securely accessible to smart contracts utilizing zero-knowledge proofs (DECO), trusted hardware (Town Crier), secure multi-party computation, and/or select DON committees.
- Fair Sequencing Services (FSS) – decentralized transaction ordering based on a specified idea of fairness, prohibiting frontrunning and extractable value from miners (MEV).
- On-Chain Contract Privacy – transaction privacy for a smart contract by a decorrelation between contract logic and settlement

output, with the DON acting as a relay between the two portions, like with Mixicles.



What Hybrid Smart Contracts Mean for Global Industries

DONs offer a sophisticated hybrid smart contract architecture that enables smooth, secure, and universal automation between any and all independent entities working across diverse systems and blockchains.

Chainlink empowers developers to leverage the deterministic execution guarantees of blockchain technology while securely outsourcing key functions like external connectivity, privacy, scalability, and order-fairness to DONs, allowing them to overcome the current technical limitations of smart contracts. Not only do hybrid smart contracts allow for more trustworthy and efficient cooperation among network users, but they also provide a mechanism to link current infrastructure to blockchain networks with no backend changes.

DONs enable a wide range of smart contract applications that demand either privacy or scalability, including the majority of business use cases as well as numerous gaming and financial applications that require high-throughput and real-time decision-making. Hybrid smart contracts also enable hitherto unseen use cases, such as those that leverage verifiable randomization and decentralized transaction ordering to establish a new precedent for math-based economic justice and transparency inside social systems.

Some of the major industries that have already been or will soon be impacted by hybrid smart contracts are as follows:

- **Identity** – information about one's identity that can be confirmed in an automated and privacy-preserving way. Smart contracts may specify the personal information necessary and the steps to be performed once it is received, while DONs can execute calculations that validate a user's personal information without exposing it publicly, disclosing it to the counterparty, or keeping it in an external system.
- **Finance** – create censorship-resistant, globally accessible, and transparent financial markets. DONs can price products and settle markets using external data, while smart contracts can define the rules of engagement for both buyers and sellers. DONs can also perform computations for optional features such as transaction concealment, KYC verification, fair transaction ordering, and high-speed off-chain processing.
- **Supply Chain** – multi-party trade agreements that use verifiable data to operate on a common ledger, digitize product lines, and/or automate processes across various platforms. Smart contracts can define various obligations, payment terms, and penalties, whereas

DONs can track shipments, monitor quality control, verify customer identities, and trigger settlement payments by combining privacy-preserving computations and external data feeds from IoT networks, and web servers, other blockchains, and enterprise backends.

- **Insurance** – a kind of parametric insurance supported by two-sided prediction markets based on pre-specified events. Smart contracts can establish premiums and claims procedures, while DONs can link the contract to external data streams for quoting and resolving claims. DONs can also do risk assessment computations, get sophisticated risk assessment results (for example, from a cloud platform), and securely validate IDs.

- **Gaming** – gaming systems that automate reward distribution, grant users with entire ownership of in-game assets through NFTs, and provide indisputable evidence that all players have an equal chance of winning. Smart contracts may describe models for gaming and reward distribution, while DONs can offer tamper-proof randomness to provide provably unbiased games and fair prize distributions. DONs enable gaming dApps to link real-world data streams such as IoT sensor readings for augmented reality and conduct some game operations off-chain for improved performance.

- **Marketing** – marketing programs that automatically deliver prizes in real-time depending on data-driven performance targets. Smart contracts may provide a tiered reward model with specified milestones, while DONs can confirm that performance parameters were satisfied and give secret calculations on consumer data and broader market trends for advanced campaign evaluations.

- **Governance** – dispersed communities that administer shared systems and pooled assets securely and equitably. Smart contracts may specify the complete governance structure, while DONs can

offer external data and calculations to trigger profit sharing, deduct shared fees, verify membership promises, and even automate decision-making.

Finally, DONs will be able to deliver all of the services that blockchains do not provide by default, as well as kickstart off-chain services by extending cryptographic security assurances to existing data and systems.

A hybrid smart contract architecture contributes to the realization of a broader vision of collaboration based on decentralized systems by enabling blockchains and non-blockchain infrastructure to interact in a secure, reliable, scalable, confidential, customizable, and/or universally connected manner. Even though cryptocurrency is a multi-trillion dollar asset class and DeFi is approaching a \$100 billion economy, the wide-ranging applicability of hybrid smart contracts and Chainlink Decentralized Oracle Networks is a clear indication that the blockchain ecosystem has only scratched the surface of what's to come.

Interoperability and Connectivity: Unlocking Smart Contracts 3.0

Today's blockchain ecosystems are reminiscent of the early days of the Internet when networks were separated into self-contained Intranets. Over time, the Internet evolved into a universal communication bridge connecting systems, resulting in the exponential expansion of individual networks and the generation of massive volumes of valuable data.

This similar shift in network architecture is taking place inside the Distributed Ledger Technology (DLT) ecosystem, where the original emphasis was on chain-maximalism — the assumption that a single

blockchain may include all of the features required for a decentralized application to operate end-to-end.

Surprisingly, the dominant logic has changed to more creative design patterns that mix various blockchain protocols, each designed for a unique purpose. This ideological shift derives from the understanding that there is just too much data, too many distinct developer requirements, and too many hazards involved with a single platform that can handle everything.

This creates a new issue: none of these disparate DLT protocols can communicate with one another, nor can they link to non-DLT systems. Chainlink – often in conjunction with other interoperability protocols – is becoming the standard method for facilitating safe and trustless communication across all different systems to tackle this basic challenge of interoperability and connection.

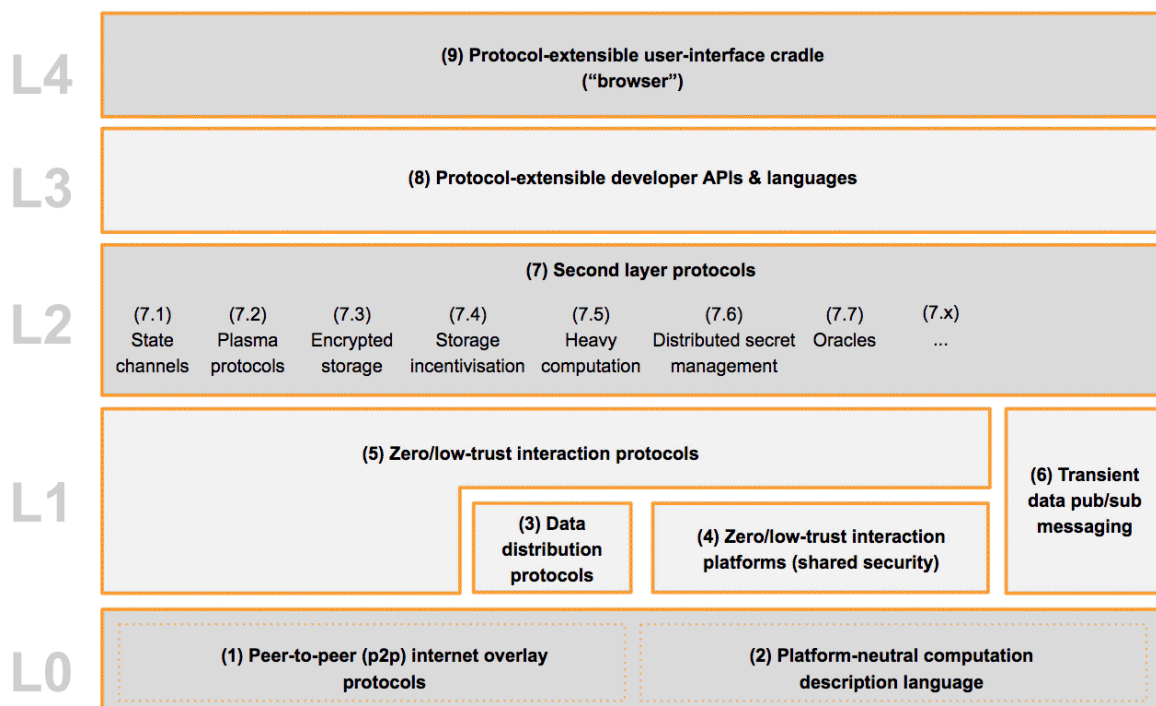
A DLT Stack That Isn't Connected

The present DLT architecture consists of several kinds of distributed ledgers, each of which makes distinct tradeoffs to specialize in certain functionalities. For example, Bitcoin, generally regarded as the gold standard for holding value, trades more functionality and slower transaction speeds for extremely trustworthy, totally decentralized transactions. Ethereum's functionality was expanded by enabling developers to associate conditional expressions (if/then) with state changes.

However, to accomplish this increased capability, it sacrificed simplicity in favor of a more sophisticated programming language, making it more difficult to learn and more prone to coding errors. Bitcoin and Ethereum have both achieved success as a result of their specialization in completing a single purpose.

This theory that chain specialization is preferable to chain maximalism is based on the same ideas as the existing Internet protocol stack. The Internet is not a single, all-encompassing protocol; rather, it is a layer of protocols that each specialize in a specific function. Protocol stacks provide optimization via layer specialization and risk mitigation by reducing each layer's attack surface.

Web3 Tech Stack



The Web3 tech stack proposed by the Web3 Foundation

There are now several useful and unique DLT protocols that comprise layers of the new distributed protocol stack Web 3.0 - a decentralized Internet where individuals own their data, identity, and assets. These protocols are improving their essential functionality, but they cannot connect with one another inside Web3 and outside the network.

The Internet stack is built on the Transmission Control Protocol and the Internet Protocol (TCP/IP), which package and route data to and from various computers/servers. It is analogous to the postal service, which routes and delivers mail. While blockchains allow secure data transmission inside their own ecosystems, the larger DLT ecosystem is still looking for common protocols that Dapps may use for cross-network communication and off-chain data interaction.

The HyperText Transfer Protocol (HTTP) and its encrypted extension, HTTPS, are two more significant protocols that gave birth to the World Wide Web (WWW). HTTP is a protocol that enables a web browser client to submit a request to a web server, which, if approved, allows the user to visit a web page.

It specifies how messages are prepared and transported, as well as what actions the browser should do when particular directives are received. In the same way that blockchains need a TCP/IP equivalent to transfer information between them at the base layer, they also require a protocol to control how that information flows. HTTP expands user capability by allowing them to visit websites, which builds on the current TCP/IP stack. It has resulted in the development of simple user interfaces and the widespread dissemination of information.

Developers require the capacity to transport data across systems as well as a common protocol to coordinate information flow to create usable decentralized apps that can be implemented in real-world circumstances. To gain broad DLT acceptance, these protocols must make creating Dapps as easy as dragging and dropping Apps.

Developers should be able to select a blockchain (Public vs. Private, Speed vs. Decentralization, Application-specific), integrate with off-chain components (Oracles, Off-chain computation, State channels), connect to a wide range of inputs (Data, IoT, Web APIs), and reliably settle using a wide

range of outputs (Payment systems, The Cloud, Other blockchains). A Dapp that uses an off-chain IoT device to trigger an Ethereum smart contract that pays out to two parties, one in Bitcoin and one in fiat through PayPal, is an example.

Creating TCP/IP and HTTP Equivalents for DLT

When considering how to enable and guide data flow, there are two major functions to consider:

Decentralized Asset Exchange

Allowing protocols to share native assets with one another is one of the more visible characteristics of interoperability. Allowing someone to pay for Ethereum Dapps using Bitcoin, or swapping Bitcoin for Litecoin, for example. Because tokens constitute the blockchain's data, facilitating asset exchange is most directly related to the TCP/IP protocol.

Decentralized Message Exchange

The second important, albeit less visible, the emphasis on interoperability is message transit across protocols. Allowing IoT data from an IOTA device, for example, to activate an on-chain smart contract on Ethereum, which in turn triggers a settlement payment on the Bitcoin blockchain.

Messaging may take place between two distributed ledgers or between an on-chain smart contract and a system that is not on the blockchain. For example, having an on-chain derivatives smart contract triggered by Reuters off-chain market data, which then triggers an off-chain fiat payment in the form of a SWIFT payment message. The HTTP protocol is most closely related to controlling how messages from one system activate data transfers on an underlying blockchain or backend system.

There are several methods for facilitating the decentralized exchange of assets and communications. Before we get into how Chainlink works across all models, let's have a look at [several other approaches](#) to the issue.

Blockchain Interoperability

One solution for interoperability is to employ a different blockchain as a bridge to enable cross-communication. This is essentially a third blockchain that sits in the midst of the two and keeps a cryptographically protected time-stamped log of the transactional and communications activity between the two.

Spoke and Hub

A hub and spoke architecture is a prominent concept in which a parent blockchain operates as a central hub to additional blockchains (spokes), sometimes known as sidechains. This technique, known as a Metachain, is being embraced by Polkadot, Cosmos, and Ethereum through multiple sidechain proposals (Plasma, Matic, Loom).

While it is feasible to establish a decentralized asset exchange as a sidechain(s) such as Plasma, most Metachains are mainly concerned with relaying cross-chain communications across all of the network's sidechains. Special-purpose bridges may also be constructed to link with distinct state machines outside their environment, such as Polkadot communicating with Ethereum.

Decentralized Finance

One of the other methods of interoperability focuses on creating a decentralized exchange of assets. Projects such as Wanchain and Icon have constructed blockchains that can be connected to other blockchains to support the decentralized exchange of assets. These solutions are very similar to decentralized banks that facilitate network trade. There are additional blockchain-based protocols, such as OX and Kyber Network, that enable the decentralized trading of native tokens and offer on-chain liquidity.

General Purpose Bridges

Another way offered is to use a blockchain to create general-purpose bridges. AION is a project that is creating a market for one-way bridges that provide timestamped blockchain consensus for validating, recording, and storing cross-bridge communication in their ledger. This is beneficial for blockchains that are not part of a hub and spoke arrangement or that need a customized bridge for a unique reason.

Non-Blockchain Interoperability

Off-chain or middleware systems are another technique to improve system compatibility. Because dealing with off-chain data cannot be confirmed in a deterministic system, entities in this category must be carefully examined. The appropriate technique, on the other hand, may provide very efficient, practical, and similarly secure solutions to their users.

Atomic Swapping

Atomic swaps are a decentralized method of exchanging two assets, such as Bitcoin for Litecoin, without using a centralized exchange. An atomic swap is similar to a real-world barter in which participants agree on the parameters of the transaction and then transfer assets after they agree.

While the technology is still in its early stages and is largely restricted to like-kind protocols, projects like Komodo have pioneered work on atomic swaps for non-like-kind protocols such as swapping BTC and ETH.

Oracles

Oracles may be used as general-purpose bridges across blockchains, which is a lesser-known feature. Oracles in this function may prepare messages for communication with any disparate corporate system, not only blockchains. For example, one blockchain's transactional activity may be utilized as an input to initiate the execution of a smart contract on another blockchain. Another example is using cloud data to trigger an on-chain smart contract, which subsequently settles off-chain on a pre-existing payment system. Oracles provide a broad variety of cross-communication capabilities that other models do not.

State Channels

Creating an off-chain state channel to move assets between parties is another option for trading assets on the same blockchain. State channels enable separate parties to trade and record asset ownership without requiring any on-chain transactions until final settlement. It is advantageous for scalability and lowering on-chain transaction costs. With the next Ethereum Plasma update focusing on UTXO asset swaps across sidechains, state channels may be used to exchange messages regarding state changes before agreeing on an on-chain state change.

Chainlink: A General Purpose Communication Standard

Chainlink is the first decentralized Oracle network that operates as an all-purpose HTTP protocol-like counterpart (or HTTPS using a TEE) for communications at the protocol and application levels, both on and off-chain. Chainlink nodes can convert communications and data from public APIs into a smart contract-readable manner.

Chainlink nodes may connect to any API, including blockchains, business systems, Web APIs, and IoT devices. External adapters may be simply constructed to expand the functionality of the node for certain activities that are not supported by the Chainlink core node, such as those that need private credentials. Chainlink is an ideal solution for enabling smart contracts to control data flow on and between various platforms with varying degrees of decentralization and security.

Off-the-Shelf Resources

Chainlink is the industry's premier solution for transferring off-chain data to on-chain smart contracts. This stand-alone capability is critical for many blockchains that do not need to interface with other blockchain protocols but need access to external inputs and outputs. These resources are largely focused on off-chain data that is used to trigger smart contracts and settlement outputs such as existing payment systems and cloud backends. It is expected that more than 80% of smart contracts need some type of off-chain resource.

Hub and Spoke

While hub and spoke interoperability protocols have their cross-chain communications, they will need off-chain data to drive on-chain movements within the ecosystem. Chainlink has previously announced collaborations with Polkadot and Ethereum to supply off-chain data to their respective networks.

Decentralized Finance

Most [decentralized finance \(DeFi\)](#) systems also support cross-chain communication across protocols through bridges, however, to shift assets, these blockchains often need market data to trigger trades. Wanchain is collaborating with Chainlink to provide off-chain data to its on-chain smart contracts.

General Purpose Bridges

Using external adaptors, Chainlink may allow messaging across protocols. They may be used to trigger on-chain activities depending on occurrences in other blockchains and systems. External adaptors for AION, IOTA, and Zilliqa are already available for developers to use in cross-chain applications. Chainlink may also leverage external data to influence asset movements on one-way blockchain bridges like the AION network.

Enterprise Blockchains

While still in development, Chainlink oracles might operate in Trusted Execution Environments (TEEs) - secure environments similar to a black box capable of doing trusted computing. TEE-based oracles are perfect for securely linking public blockchains to enterprise blockchains, integrating corporate blockchains, and transmitting sensitive data across all systems. Interoperability on public blockchains is not appropriate for sensitive data; however, deploying Town Crier, a TEE-based oracle purchased by Chainlink in 2018, enables privacy-preserving message relays between several chains.

Private Keys and Credentials

Chainlink might also be utilized to manage credentials, which is crucial for decentralized exchange, by using TEEs. Chainlink oracles operating in a TEE might log into someone's account to confirm they have an asset or the finances to make a transaction. The data may be safely communicated to the smart contract, which might then activate or refuse an exchange.

Chainlink oracles might potentially be used to manage a private key to initiate transactions on public blockchains. Because the majority of accounts participating in exchanges will be from previously funded accounts, a TEE is required to handle sensitive information such as a private key without fear of money being stolen or account information being disclosed.

State Channels

State channels are excellent for cost reduction and off-chain scalability. However, impartial triggers are required to guarantee that state channels are respected and resolved on-chain in the same form that they took off-chain. If certain circumstances are satisfied, Chainlink oracles may enable data to trigger state channel transactions as well as utilize data to trigger on-chain settlement.

Atomic Swapping

Because an atomic swap is peer-to-peer, it is difficult for the typical user to determine if the script allowing the atomic swap is authentic. Decentralized oracles may be used to validate the payload of the atomic swap to ensure it works as expected. It may also employ collateral to motivate nodes to give correct swap information.

The Beginnings of Connected Consensus

Interoperability between blockchains and diverse systems will bring in a new wave of smart contract capability, similar to the progression of Intranets to the Internet. New use cases, previously unthinkable in Web 2.0, will alter the way consumers and organizations interact.

TCP/IP was the cross-communication protocol that allowed networks all over the globe to send information to one another, while HTTP allowed web browsers to instantly access data. These protocols facilitated the development of the world wide web as we know it today. Interoperability protocols enable multi-chain applications to obtain any off-chain resource to trigger state changes, bringing a comparable shift to the DLT space. They not only link blockchain silos but also the new DLT ecosystem to the world's present non-DLT infrastructure.

At its heart, Chainlink is a general-purpose communication standard that may be used to securely transmit input/output data messages across any system. Smart contracts 3.0 emerge as fully integrated end-to-end solutions capable of uprooting many present business models when all protocols in the DLT stack can interoperate effortlessly.

Future of Smart Contracts

Smart contracts are not a new notion; in fact, the concept of a self-executing digital contract has existed for more than two decades. As I previously said, it was initially described by a guy widely regarded as the "Father of Smart Contracts," famous computer scientist Nick Szabo.

However, the technology for really competent and secure smart contracts has only lately become available—blockchain technology. There are already a plethora of blockchains that enable smart contracts.

They're still in their early stages of development and acceptance, and they're also suffering from certain growing pains that keep them relatively specialized, such as potentially high costs and user experience limits.

As technology permeates every aspect of our lives, so will smart signing, with its ability to alter the way we do business. Smart contracts can conduct transactional agreements securely and safely. They are irrevocable and guarantee that legally enforceable regulations be followed. Smart contract technology will continue to be used by industries. The blockchain is well-suited to converting complicated legal agreements into software-capable step-by-step operations.

Today, only a tiny percentage of the public interacts with smart contracts daily, the majority of whom are bitcoin users.

But that might be about to change. Smart contracts are becoming more common as more efficient blockchains are built, and large corporations are constructing their smart contract-capable blockchains, such as IBM's Hyperledger Fabric and R3's Corda platform.

With the rapid use of enterprise-friendly smart contract platforms, it may not be long until smart contracts transition from a technology of the future to a technology of today.

Smart contracts are now a model illustration of "Amara's Law," the notion described by Stanford University computer scientist Roy Amara that humans tend to overestimate new technology in the short run and underestimate it in the long run.

Although smart contracts must improve before they can be extensively used in complicated business interactions, they have the potential to alter the reward and incentive structure that dictates how parties contract in the

future. For that reason, while considering smart contracts, it is critical not to focus just on how old ideas and structures may be carried over to this new technology. Rather, the actual smart contract revolution will come from wholly new paradigms that we have yet to imagine.

Entangling the challenges we described previously is part of the future of smart contracts.

Lawyers at [Cornell Tech](#), for example, who believe that smart contracts will become commonplace, have committed themselves to examine these risks.

There is more at stake today than ever before to overcome the problematic features of smart contracts, especially with the emergence of NFT Marketplaces.

Smart contracts may influence developments in specific sectors, particularly legal. This is because attorneys will transition from authoring conventional contracts to providing standardized smart contract templates, which will be comparable to the standardized traditional contracts available on LegalZoom.

Smart contracts may also be used by merchant acquirers, credit firms, and accountants for duties such as real-time audits and risk assessments.

As more DeFi Apps are deployed throughout the globe and the value of NFT marketplaces grows, it's clear that smart contracts are evolving into a combination of paper and digital content, with contracts validated by blockchain and supported by physical copy.

Smart contracts will only become better as technology advances, and they will become more trustworthy and efficient.

Smart Contract Advantages

These agreements already have a number of benefits over conventional arrangements. This figure is expected to rise in the future as technology advances.

For the time being, here are some advantages of employing them.

Autonomy

As previously stated, smart contracts are performed automatically by the network rather than by humans. This eliminates the need for third parties or middlemen and empowers people to control their transactions.

Safety

Smart contracts' security originates from their storage on the blockchain (where they cannot be modified) and data encryption.

When it comes to security problems like double-spending, smart contract encryption helps protect the same digital token against internet assaults that result in token duplication and false spending.

Speed

Decentralized finance, as opposed to banks and other financial organizations, expedites the bulk of transactions, from withdrawals to loans. Smart contracts, with their automated functions, continue this trend of efficiency.

Savings

Smart contracts' automated nature across sectors saves numerous hours of otherwise labor-intensive tasks conducted by pricey third parties. As a result, the related expenses are reduced.

Transparency

A lack of faith in current financial institutions is an often mentioned reason why individuals want to switch from fiat currency to cryptocurrencies. This is especially true in the aftermath of the Great Recession, concerns caused by lenders, and a lack of public openness.

The status of each transaction is recorded on the blockchain, where it is immutable and public knowledge, using smart contracts. This, together with a transparent system, contributes to a more trustworthy financial community.

Smart Contracts Are Not Perfect

Smart contracts are not without flaws. What if there are problems in the code? Or, more specifically, how should governments regulate such contracts? Alternatively, how would governments tax smart contract transactions?

The list of difficulties is endless. Experts are working to resolve them, yet these crucial challenges deter prospective users.

While human mistake is a fair criticism of the system, a smart contract will not necessarily sue you for it. Funds may not be issued, or an employer may be repaid automatically. Human mistakes will occur, whether on the blockchain or not.

Coding errors or human mistakes in security (giving away your secret keys) may also lead to hacking or theft. Because the coding is so sophisticated, contracts may occasionally become accessible to hackers.

For example, the ICO KICKICO lost [\\$8 million](#) in July due to a smart contract violation. However, the most prominent breach happened in June 2016 on the [DAO \(Decentralized Autonomous Organization\)](#), in which the hackers stole \$50 million. This resulted in a hard fork of Ethereum Classic (ETC) to Ethereum (ETH) in a bid to strengthen the platform.

Those seem to be big amounts – but are they? Consumers in the United States lost about \$17 billion in 2017 due to identity theft alone. (Of course, blockchain technology will eventually be able to safeguard your identity as well.)

Despite human fragility, blockchain and smart contracts are likely to provide the solutions to our existing system's troubles. Blockchain technology continues to provide more security. It's the difference between using a standard padlock and a Schlage deadbolt...or not using a lock at all.

"Fine, but aren't their constraints?" There's no way to keep critical data if everything is public."

That is correct – but only for a short time.

[Enigma](#) and [Wanchain](#) are two important efforts handling privacy and "hidden contracts." A secret contract is a smart contract that enables sensitive data to be safely kept while being confirmed using blockchain technology. Wanchain leverages ring signatures and one-time address creation for smart contract transactions to protect user privacy. This ensures that identities remain anonymous.

As challenges arise in the context of smart contracts, so do solutions. Szabo, Satoshi, and Buterin were all engaged in improving an inefficient financial system, whether they were discussing Bitcoin or smart contracts. Whether the answer is Ethereum smart contracts or another platform, blockchain technology is critical to the future of FinTech.

Here are a few key lessons to consider while dealing with Smart Contracts:

- **Immutability**

Smart contracts are not reversible, which means that if an issue arises, it may be difficult or impossible to resolve.

Some of the advantages of smart contracts are also aspects that generate limits. While the irreversible nature of these contracts increases safety and transparency, it might make it harder to repair mistakes.

After all, operational errors, human error, or unlawful conduct may all result in an erroneous blockchain recording that, once published, is not intended to be altered for security and transparency.

Written components of a typical contract may be readily amended if both parties agree. In the case of smart contracts, however, a new version of the contract must be deployed, and the procedure must be redone.

Smart contract dispute resolution is a system that aims to address many of these concerns, although the technology is still maturing alongside possible solutions to these faults.

- **Extraneous Information**

Another limitation that smart contracts now face is the inability to access external data that allows procedures to be conducted appropriately.

Today, Oracles serve as a solution to this issue by linking this external data to the internal network, but it introduces a third-party service that many customers want to avoid.

While these methods provide an overall trustworthy means to connect smart contracts with the events required to put the contract in action, they create potential problems that go beyond intermediaries.

In certain circumstances, bad actors may use these oracles to modify a smart contract with harmful code. Because the data given by Oracles is what lets each stage proceed, any inaccurate data impacts the conclusion of the contract.

- **Scalability**

Although smart contracts' autonomy helps to speed up the process, this is not always the case. Slow transaction delays that might span minutes to hours may be caused by the network itself.

Managing scalability in DeFi entails enabling an increasing number of transactions to be completed regularly without jeopardizing network

security or efficiency. This is not the case with smart contracts as they now exist.

When a network is highly congested, transaction speed suffers. As a consequence, consumers are subjected to a lower-quality experience. This is one of the barriers to widespread blockchain adoption.

- **Hackers**

Smart contracts, like everything else on the internet, are vulnerable to hacking. But what exactly does it mean for a bad actor to breach a contract?

The major issue with any cyber threat is that a hacker may exploit vulnerabilities or security flaws. Millions of dollars in cryptocurrencies might be stolen if a smart contract or a blockchain platform is hacked.

Because internet security is never completely secure against hackers, Vault safeguards your assets by transferring cash from your wallet to a centralized lending pool guaranteed for \$100 million in BitGo, a reputable digital asset trust, and security business.

- **Reliability Issues**

Because smart contracts are implemented on a blockchain network, they may not always be trustworthy. This implies that they may potentially be prone to downtime and outages; although Ethereum has shown to be quite dependable, [newer smart contract networks](#) such as Solana have seen a few outages since the technology is still in its early stages.

- **Expensive**

Smart contracts may be expensive to construct and need a high degree of technical knowledge.

- **Customization Issues**

Because smart contracts are not always configurable, they may not be appropriate for all firms or transactions.

In the Meantime

In reality, the most serious issue with smart contracts has nothing to do with smart contracts at all. The issue is that many governments do not comprehend them — and hence do not know how to regulate them. However, this is changing. This year, the US state of Tennessee approved legislation recognizing smart contracts as legally enforceable. Canada is also involved. They started testing smart government contracts in early 2018. As the advantages become more apparent, it won't be long until the rest of the globe joins in.

But why should we wait for them to catch up? You may begin utilizing smart contracts immediately – or you can learn to create your own.

Wrapping Up

Smart contracts are already starting to replace intermediaries, as seen by the examples I've provided. We also recognized the potential for future uses — remember a home sale example? You wouldn't need the services of an estate agent, a lawyer, or a bank, would you?

So, if smart contracts serve their objective, maybe we will one day live in a world without intermediaries.

What would be the result?

The best part about not using intermediaries is that we save a lot of money. Not only that, but we would no longer have to rely on anybody.

However, there is a possible downside: individuals may lose their employment. An intermediary, like you and me, is a real person. Why would someone pay an employee to execute a task that a smart contract could do for free? They wouldn't do it.

Of course, no one can predict the future. We may only estimate and forecast, but we must be prepared for any scenario.

As you can see, smart contracts may help to make the world a better place by eliminating the need for commissions. It has the potential to minimize fraud, delays, and total costs in a variety of situations. However, as technology advances, many vocations will become obsolete. After reading this book, you should be able to answer the enigmatic question, 'what is a smart contract?'

Smart Contracts FAQs

What exactly is a smart contract?

A smart contract is a computer protocol designed to digitally expedite, verify, or enforce contract negotiation or performance. Smart contracts enable the execution of credible transactions without the involvement of third parties.

What exactly is a blockchain smart contract?

A smart contract Blockchain is a platform that incorporates digital transaction protocols. It codifies the contract terms and defines a set of rules that govern the contract. The blockchain network ensures that transactions are visible, secure, and immutable.

Is a smart contract legally binding?

Because not all agreements must be in writing to be legally binding, smart or self-executing contracts would be appropriate and legitimate under contract laws in many nations. The agreement in Smart Legal Contracts states its parameters, and the parties involved are obligated to obey them or suffer legal ramifications.

What is the purpose of a smart contract?

Smart contracts allow you to trade money, property, shares, or anything else of value in a transparent, conflict-free manner without the need for an intermediary.

Who developed smart contracts?

In 1994, Nick Szabo, an American [cryptographer](#) and computer scientist coined the phrase "smart contract."

Can just Ethereum be used to generate smart contracts?

No. Smart contracts may be established using other cryptocurrencies - or, more precisely, their blockchains. That being said, Ethereum was the technology that began it all, and it is still regarded as the best choice for reaping the advantages of a smart contract.

What are the primary applications of a smart contract?

A smart contract's primary use would be the automation of certain operations that would otherwise need the employment of a middleman. Consider buying a home or collecting your salary: In the first case, you'd need attorneys and brokers, while in the latter, you'd need your employer to intervene. These requirements are eliminated by using an Ethereum smart contract.

How do you choose the best cryptocurrency exchange for yourself?

When selecting the top cryptocurrency exchange for yourself, you should always strive to strike a balance between the fundamental features that all top cryptocurrency exchanges should offer and those that are vital to you individually. For example, although all of the best exchanges should have top-tier security measures, if you're just interested in trading the [major cryptocurrencies](#), you probably don't care about the range of coins accessible on the exchange. It's all on a case-by-case basis!

Which bitcoin exchange is the most suitable for newcomers?

Reading through numerous top crypto exchange reviews online, you're certain to discover that one thing most of these exchanges have in common is that they're quite straightforward to use. While some exchanges are more basic and beginner-friendly than others, you should have no trouble utilizing any of the top-rated exchanges. Having said that, many users think that Coinbase is one of the most user-friendly exchanges available today.

What is the difference between a cryptocurrency exchange and a brokerage?

A [cryptocurrency exchange](#), in layman's terms, is a location where you may meet and trade cryptocurrencies with another person. The exchange platform (for example, Binance) functions as a go-between, connecting you (your offer or request) with that other individual (the seller or the buyer). However, there is no "other person" with a brokerage - you come and swap your crypto coins or fiat money with the platform in question, with no interference from a third party. However, when it comes to bitcoin exchange rankings, both of these kinds of firms (exchanges and brokerages) are frequently lumped together under the umbrella name - exchange. This is done to keep things simple.

Is it true that all of the biggest bitcoin exchanges are situated in the United States?

No, absolutely not! While some of the most popular cryptocurrency exchanges are headquartered in the United States (for example, Coinbase or Kraken), there are other well-known industry giants located all over the globe. Binance, for example, is situated in Tokyo, Japan, while Bittrex is based in Liechtenstein. While there are several reasons why an exchange might choose to be situated in one region over another, most of them are commercial reasons that have no bearing on the platform's user.



Your gateway to knowledge and culture. Accessible for everyone.



z-library.se

singlelogin.re

go-to-zlibrary.se

single-login.ru



[Official Telegram channel](#)



[Z-Access](#)



<https://wikipedia.org/wiki/Z-Library>