How smart contracts work

Smart contracts work by following simple “if/when…then…” statements that are written into code on a blockchain. A network of computers executes the actions when predetermined conditions have been met and verified. These actions could include releasing funds to the appropriate parties, registering a vehicle, sending notifications, or issuing a ticket. The blockchain is then updated when the transaction is completed. That means the transaction cannot be changed, and only parties who have been granted permission can see the results.

Within a smart contract, there can be as many stipulations as needed to satisfy the participants that the task will be completed satisfactorily. To establish the terms, participants must determine how transactions and their data are represented on the blockchain, agree on the “if/when...then…” rules that govern those transactions, explore all possible exceptions, and define a framework for resolving disputes

## Benefits of smart contracts

#### Speed, efficiency and accuracy

Once a condition is met, the contract is executed immediately. Because smart contracts are digital and automated, there’s no paperwork to process and no time spent reconciling errors that often result from manually filling in documents.

#### Trust and transparency

Because there’s no third party involved, and because encrypted records of transactions are shared across participants, there’s no need to question whether information has been altered for personal benefit.

#### Security

Blockchain transaction records are encrypted, which makes them very hard to hack. Moreover, because each record is connected to the previous and subsequent records on a distributed ledger, hackers would have to alter the entire chain to change a single record.

#### Savings

Smart contracts remove the need for intermediaries to handle transactions and, by extension, their associated time delays and fees.

## Applications of smart contracts

Explore how businesses benefit from smart contracts in active blockchain solutions

### **Safeguarding the efficacy of medications**

Sonoco and IBM are working to reduce issues in the transport of lifesaving medications by increasing supply chain transparency. Powered by IBM Blockchain Transparent Supply, Pharma Portal is a blockchain-based platform that tracks temperature-controlled pharmaceuticals through the supply chain to provide trusted, reliable and accurate data across multiple parties.



Sonoco + IBM: Safeguarding the efficacy of lifesaving medications​ with blockchain (3:11)

[Explore IBM Blockchain Transparent Supply](https://www.ibm.com/case-studies/the-home-depot/)

### **Increasing trust in retailer-supplier relationships**

The Home Depot uses smart contracts on blockchain to quickly resolve disputes with vendors. Through real-time communication and increased visibility into the supply chain, they are building stronger relationships with suppliers, resulting in more time for critical work and innovation.

[Learn how The Home Depot builds trust](https://www.ibm.com/case-studies/the-home-depot/)

### **Making international trade faster and more efficient**

By joining we.trade, the trade finance network convened by IBM Blockchain, businesses are creating an ecosystem of trust for global trade. As a blockchain-based platform, we.trade uses standardized rules and simplified trading options to reduce friction and risk while easing the trading process and expanding trade opportunities for participating companies and banks.



IBM Blockchain: Transforming trade finance – and trade (3:38)

[Explore blockchain trade finance solutions](https://www.ibm.com/blockchain/trade-finance)

Explore blockchain for supply chain solutions

Explore blockchain trade finance solutions

## Related Solutions

### **Supply chain transparency**

Create a blockchain ecosystem to share data with your supply chain partners, with transactions that are more efficient and built on trust.

* [Learn about IBM Blockchain Transparent Supply](https://www.ibm.com/blockchain/container-logistics)

### **Blockchain for trade finance**

Pull down invisible barriers to growth and reinvent trade and trade finance with our network-convening expertise and the industry’s leading platform.

* [Learn about blockchain for trade finance](https://www.ibm.com/blockchain/trade-finance)

### **Blockchain for food supply**

Growers, distributors and retailers can build trust and make our food safer, by enhancing visibility and accountability in every step of the food supply.

* [Learn about IBM Food Trust™](https://www.ibm.com/blockchain/solutions/food-trust)

## Resources

[What is blockchain?](https://www.ibm.com/topics/what-is-blockchain" \t "_self)

[Learn from the ground up what blockchain is all about and how it can benefit your organization. At no cost to you, download a copy of IBM’s Blockchain for Dummies guide.](https://www.ibm.com/topics/what-is-blockchain" \t "_self)

[Blockchain industry applications](https://www.ibm.com/blockchain/industries" \t "_self)

[Remove friction, build trust and unlock new value with IBM Blockchain. See how blockchain is solving problems for businesses and industries and get inspired.](https://www.ibm.com/blockchain/industries" \t "_self)

[Blockchain solutions](https://www.ibm.com/blockchain/solutions" \t "_self)

[You can join existing blockchain networks that are transforming industries by bringing revolutionary trust and transparency to supply chains, global trade, international payments, our food supply, and much more.](https://www.ibm.com/blockchain/solutions" \t "_self)

[IBM Blockchain Services](https://www.ibm.com/blockchain/services" \t "_self)

[Co-create with the industry’s leading blockchain services provider. We can help you build a scalable business network underpinned by the best technology.](https://www.ibm.com/blockchain/services" \t "_self)

[Writing smart contracts](https://cloud.ibm.com/docs/blockchain-sw?topic=blockchain-sw-develop-smart-contracts" \t "_self)

[The IBM Blockchain Platform supports smart contracts written in Go and Node.js. Read this tutorial to learn how to get started writing encoded business logic, terms and conditions for execution on blockchain.](https://cloud.ibm.com/docs/blockchain-sw?topic=blockchain-sw-develop-smart-contracts" \t "_self)

[Deploy a smart contract on your network](https://cloud.ibm.com/docs/blockchain-sw-213?topic=blockchain-sw-213-ibp-console-smart-contracts" \t "_self)

[If you are running an IBM Blockchain Platform network, this tutorial will guide you through writing, packaging and deploying a smart contract using Hyperledger Fabric.](https://cloud.ibm.com/docs/blockchain-sw-213?topic=blockchain-sw-213-ibp-console-smart-contracts" \t "_self)

# Introduction to smart contracts

Smart contracts are the fundamental building blocks of [Ethereum applications](https://ethereum.org/en/dapps/). They are computer programs stored on the blockchain that allows us to convert traditional contracts into digital parallels. Smart contracts are very logical - following an if this then that structure. This means they behave exactly as programmed and cannot be changed.

Nick Szabo coined the term "smart contract". In 1994, he wrote [an introduction to the concept](https://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/smart.contracts.html) and, in 1996, [an exploration of what smart contracts could do](https://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/smart_contracts_2.html).

Nick Szabo envisioned a digital marketplace built on these automatic, cryptographically secure processes. A place where transactions and business functions can happen trustlessly — without intermediaries. Smart contracts on Ethereum put this vision into practice.

## What are contracts?

You're probably thinking: "I'm not a lawyer! Why would I care about contracts?". For most people, contracts bring to mind needlessly long terms and conditions agreements or boring legal documents.

Contracts are just agreements. That is, any form of agreement can be encapsulated within the conditions of a contract. Verbal agreements or pen-and-paper contracts are acceptable for many things, but they aren't without flaws.

### Trust and contracts

One of the biggest problems with a traditional contract is the need for trusted individuals to follow through with the contract's outcomes.

Here is an example:

Alice and Bob are having a bicycle race. Let's say Alice bets Bob $10 that she will win the race. Bob is confident he'll be the winner and agrees to the bet. In the end, Alice finishes the race well ahead of Bob and is the clear winner. But Bob refuses to pay out on the bet, claiming Alice must have cheated.

This silly example illustrates the problem with any non-smart agreement. Even if the conditions of the agreement get met (i.e. you are the winner of the race), you must still trust another person to fulfill the agreement (i.e. payout on the bet).

## Smart contracts

Smart contracts digitize agreements by turning the terms of an agreement into computer code that automatically executes when the contract terms are met.

### A digital vending machine

A simple metaphor for a smart contract is a vending machine, which works somewhat similarly to a smart contract - specific inputs guarantee predetermined outputs.

* You select a product
* The vending machine returns the amount required to purchase the product
* You insert the correct amount
* The vending machine verifies you have inserted the correct amount
* The vending machine dispenses the product of choice

The vending machine will only dispense your desired product after all requirements are met. If you don't select a product or insert enough money, the vending machine won't give out your product.

### Automatic execution

One of the most significant benefits smart contracts have over regular contracts is that the outcome is automatically executed when the contract conditions are realized. There is no need to wait for a human to execute the result. In other words: smart contracts remove the need for trust.

For example, you could write a smart contract that holds funds in escrow for a child, allowing them to withdraw funds after a specific date. If they try to withdraw the funds before the specified date, the smart contract won't execute. Or, you could write a contract that automatically gives you a digital version of a car's title when you pay the dealer.

### Predictable outcomes

The human factor is one of the biggest points of failure with traditional contracts. For example, two individual judges may interpret a traditional contract in different ways. Their interpretations could lead to different decisions getting made and disparate outcomes. Smart contracts remove the possibility of different interpretations. Instead, smart contracts execute precisely based on the conditions written within the contract's code. This precision means that given the same circumstances, the smart contract will produce the same result.

### Public record

Smart contracts are also useful for audits and tracking. Since Ethereum smart contracts are on a public blockchain, anyone can instantly track asset transfers and other related information. You can check to see that someone sent money to your address, for example.

### Privacy protection

Smart contracts can also protect our privacy. Since Ethereum is a pseudonymous network (your transactions are tied publicly to a unique cryptographic address, not your identity), you can protect your privacy from observers.

### Visible terms

Finally, like contracts, you can check what's in a smart contract before you sign it (or otherwise interact with it). Better yet, public transparency of the terms in the contract means that anyone can scrutinize it.

## Smart contract uses cases

So, smart contracts are computer programs that live on the blockchain. They can execute automatically. You can track their transactions, predict how they act and even use them pseudonymously. That's cool. But what are they good for? Well, smart contracts can do essentially anything that other computer programs do.

They can perform computations, create currency, store data, mint NFTs, send communications and even generate graphics. Here are some popular, real-world examples: