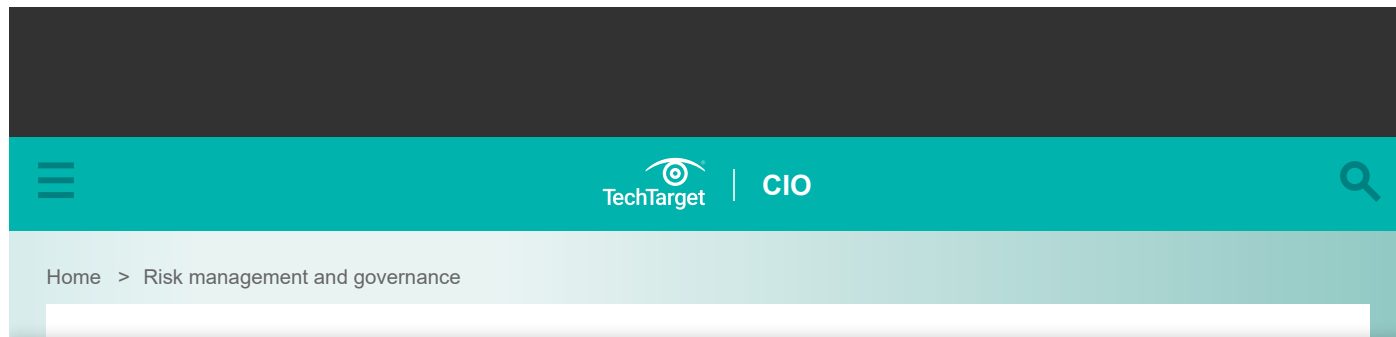


**Tech Accelerator****Blockchain for businesses: The ultimate enterprise guide**

DEFINITION

smart contract

By **Nick Barney**, Technology Writer | **George Lawton**

What is a smart contract?

A smart contract is a self-executing computer program that automatically executes the terms of a contract without the involvement of third parties. Smart contract execution can result in the exchange of money, delivery of services, unlocking of content protected by digital rights management or other types of data manipulation such as changing the name on a land title. Smart contracts can also be used to enforce privacy protection by, for example, facilitating the selective release of privacy-protected data to meet a specific request.

There are a variety of architectures for how the programs underpinning smart contracts are developed, distributed, managed and updated. They can be stored as part of a [blockchain](#) or other [distributed ledger](#) technology, and integrated into various payment mechanisms and digital exchanges that can include bitcoin and other cryptocurrencies.

Despite the name, smart contracts are not legally binding contracts. Their main function is to programmatically execute [business logic](#) that performs various tasks, processes or transactions that have been programmed into them to respond to a given set of conditions. Legal steps must be undertaken to link this execution to legally binding agreements between parties.

Blockchain and IoT for automated freight invoicing

Blockchain's distributed ledger technology allows multiple parties to create and update a single, trusted record. DLT Labs developed a SaaS blockchain platform for Walmart Canada to tender shipments with carriers over a private network. A smart contract is populated with standard terms and updated with details on the new shipment. IoT devices send real-time updates during transit and confirm delivery, and a preapproved invoice is automatically generated and paid. Reconciliation is unnecessary because each party works from the same information.

SHIPMENT DETAILS CONFIRMED	GOODS TRANSPORTED	CARRIER AGREES TO FINAL DETAILS	SHIPPER INVOICED	PRE-APPROVED PAYMENT
Tender contract: - Accounting - Contract terms - Legal terms Bill of lading: - Date/time - Delivery instructions - Destination - Distance - Fuel, mileage - Hazmat - Load details - Number of stops - Shipment details - Size, weight, type	Transportation data from IoT devices: - Temperature - GPS-related data: - Demurrage (delay charges) - Layover - Location - Shunts (short movements of containers) - Wait time 	Data from IoT devices: - Accessorial (additional) charges - Proof of delivery		

ILLUSTRATION © ALAMY/STOCK IMAGES

© 2019 TECHTARGET. ALL RIGHTS RESERVED. TechTarget



Walmart Canada deployed a blockchain application that combines smart contracts to automate freight invoicing.

How do smart contracts work?

A smart contract is a special kind of program that encodes business logic that runs on a special-purpose [virtual machine](#) baked into a blockchain or other type of distributed ledger. Smart contracts function on the principle of "if/when ... then ..." statements. These statements are written into the code and when certain predetermined conditions are met and verified by the network of computers, or [nodes](#), the actions specified in the contract are executed.

THIS ARTICLE IS PART OF



Blockchain for businesses: The ultimate enterprise guide

Which also includes:

[Top 10 benefits of blockchain for business](#)

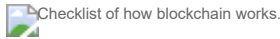
[What are the 4 different types of blockchain technology?](#)

[7 must-have blockchain developer skills](#)

The process of creating a smart contract starts with business teams working with developers to describe their requirements for the desired behavior of the smart contract in response to various events or circumstances. Simple events could be conditions such as payment authorized, shipment received or a utility meter reading threshold. More sophisticated logic might encode more complex events such as calculating the value of a derivative financial instrument and processing a trade of the derivative, or automatically releasing an insurance payment in the event of a person's death or a natural disaster.

The developers then work in a smart contract-writing platform to develop the logic and test it to ensure that it works as intended. After the application is written, it is handed off to another team for a security review. This could be an internal expert or a firm that specializes in vetting smart contract security. Once the contract has been approved, it is deployed on an [existing blockchain or other distributed ledger infrastructure](#).

Once the smart contract is deployed, it is configured to listen to event updates from an "oracle," which is essentially a cryptographically secured streaming data source. The smart contract executes once it receives the appropriate mix of events from one or more oracles.



The five steps in executing and verifying transactions in data in a blockchain include creation of blocks.

Smart contract programming languages

There are several [popular programming languages used for smart contract development](#). Some of the most popular include the WebAssembly (WASM) language and the Digital Asset Modeling Language (DAML).

WASM allows developers to create smart contracts that can run in a web browser and be integrated into blockchains and other distributed ledgers using various programming languages such as C, [JavaScript](#), TypeScript and Rust.

DAML is an enterprise-focused language that is designed to model various business use cases, and which also helps to enforce privacy safeguards.

Smart contract applications and blockchain

Smart contracts are used in various industries for [supply chain management](#) and intellectual property rights, among other purposes. Smart contracts are most often deployed using blockchain technology, a decentralized record-keeping technology that keeps immutable and secure records of transactions. Though smart contracts do not by definition require blockchain, the technology is ideal for storing smart contracts because of its [security](#) and [immutability](#). Smart contract data is encrypted on a shared ledger, making it virtually impossible to lose the [information stored in the blocks](#).

Flexibility is another advantage of blockchain technology being incorporated into smart contracts. Developers can store almost any type of data in a blockchain, and they have a wide variety of transaction options to choose from. Thus, blockchain-based smart contracts are helping make transactions and other business processes more secure, efficient and cost-effective, thereby reducing transaction costs and benefiting a variety of industries.

Some applications of smart contract include the following:

Cryptocurrency. One of blockchain's most popular applications is [cryptocurrency](#), a form of digital currency created by solving complex mathematical algorithms and securing blockchain networks. The most popular smart contract platform is [Ethereum](#), which is also a widely used cryptocurrency platform. The Ethereum community has developed the Solidity language for writing smart contract applications that are designed to run on the Ethereum Virtual Machine execution environment.

Insurance contracts. Smart contracts have the potential to transform the insurance industry by automating policy issuance, claims processing and premium payments. By using smart contracts, insurers can streamline operations, reduce paperwork and provide faster and more transparent services to policy holders.

Intellectual property rights. Smart contracts are used to manage and enforce intellectual property rights such as copyrights and patents. By creating digital assets such as non-fungible tokens ([NFTs](#)) that represent these rights and using smart contracts to govern their ownership and licensing, creators can protect their [intellectual property](#) and ensure fair compensation for their work.

Real estate transactions. Smart contracts can be used to simplify real estate transactions, which usually involve intermediaries and high transaction costs. Smart contracts can automate tasks such as property transfers, escrow services and title registrations. This reduces the need for intermediaries, speeds up transactions and ensures transparency in the real estate market. For example, in 2016, Cook County, Ill., used blockchain to create a database to transfer and track property titles. When these transactions occur, in addition to the traditional paper deed, the buyer receives a digital token that can be used as proof of ownership.

Supply chain management. Supply chain management involves multiple stakeholders and complex transactions. Smart contracts can automate and streamline processes such as order fulfillment, [inventory management](#) and payment settlements. By using smart contracts, supply chain participants can ensure transparency, traceability and efficiency in the movement of goods and services.

A [variety of other industries could benefit from using blockchain-based smart contracts](#). Automating healthcare payments using smart contracts can reduce overbilling and prevent fraud. The music industry could record the ownership of music in the blockchain and then deploy a smart contract to ensure royalties are paid when the music is used for commercial purposes. Smart contracts and blockchain could benefit the automobile industry by storing readily available information about vehicle maintenance and accident and ownership history.

What is a Smart Contract and What is it Used For?



Types of smart contracts

Smart contracts can be classified into different types based on their functionality and purpose. There are generally four types of smart contracts.

Payment contracts. Payment contracts facilitate the transfer of funds between parties based on predefined conditions. These contracts can automate payment processes, ensuring funds are released only when specific criteria are met. For example, a payment contract could be set up to release funds to a seller once the buyer has received the goods or services.

Escrow contracts. Escrow contracts act as intermediaries in transactions, holding funds or assets until certain conditions are fulfilled. They provide an additional layer of security and trust in peer-to-peer transactions, ensuring both parties meet their obligations before funds or assets are released.

Governance contracts. Governance contracts enable decentralized decision-making within a blockchain network or organization. These contracts allow token holders to vote on proposals or changes to the network's rules, ensuring democratic and [transparent](#) governance.

Multi-signature contracts. Multi-signature contracts require multiple parties to provide their approval or signatures before a transaction can be executed. These contracts add an extra layer of security and can be used in situations where consensus or agreement from multiple parties is necessary.

Smart contract advantages

There are several potential [business and security advantages](#) from using smart contracts.

Cost efficiency. Smart contracts promise to automate business processes that span organizational boundaries. This can eliminate many operational expenses and save resources, including the personnel needed to monitor the progress of a complex process that executes in response to conditions that span companies.

Processing speed. Smart contracts can improve the processing speed of business processes that run across multiple enterprises.

Autonomy. Smart contracts are performed automatically by the network and reduce the need for a third party to manage transactions between businesses.

Reliability. Smart contracts can also take advantage of blockchain ledgers and other distributed ledger technologies to maintain a verifiable record of all activity related to execution of complex processes and that cannot be changed after the fact. It also supports automated transactions that remove the potential for human error and ensure accuracy in executing the contracts.

Common issues and challenges with smart contracts

There are several challenges when planning a smart contract rollout, including the following:

Security. Smart contracts secure certain key elements in a [business process that involves multiple parties](#). However, the technology is new, and hackers continue to identify new attack surfaces that allow them to compromise the intent of the businesses that specified the rules. In the early days of Ethereum, smart contract hackers managed to steal \$50 million in cryptocurrency. The IEEE has also documented concerns about inconsistencies in the tools used to [detect different vulnerabilities in smart contract security](#).

Integrity. One oracle (one of the streaming data sources that sends event updates) needs to protect against hackers faking events that trigger smart contracts into executing when they should not. It must be programmed to accurately generate events, which can be challenging for complex scenarios.

Alignment. Smart contracts can speed the execution of processes that span multiple parties regardless of whether they are in alignment with all parties' intention and understanding. But this capability can also magnify the impact of the damage that can occur when events spiral out of control, particularly when there is no way to stop or unwind unintended behavior. The Gartner research firm has noted that this issue poses challenges in smart contract [scalability](#) and manageability that have yet to be fully addressed.

Management. Smart contracts are complicated to implement and manage. They are often configured in ways that make them difficult or impossible to change. Although this

could be considered a security advantage, the parties cannot make any changes to the smart contract agreement or incorporate new details without developing a new contract.

Attacks. Some of the most common and dangerous [cyber attacks](#) used on smart contracts include the following:

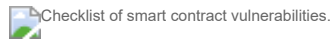
- Reentrancy attacks, which allow malicious contracts to withdraw resources repeatedly without waiting for verification.
- Transaction order dependence attacks, or frontrunning, in which miners front run profitable contracts by finding contracts with high fees and duplicating them with even higher fees.
- Force-feeding attacks, which sends Ether -- Ethereum's native currency -- to smart contracts and therefore affecting logic reliant on balances, such as paying out a reward if a balance increases.
- Denial of service ([DoS](#)), which can block contracts from executing or cause unintended reversions by overloading services.

Gas griefing. Ethereum users must pay gas fees -- Ether paid to verify the addition of content or additional transactions -- to execute transactions on the Ethereum blockchain. Gas griefing occurs when a user sends enough gas (Ether fees) for the target contract but not for subcalls, or calls the contract makes to other contracts. This can prevent subcalls from executing and negatively affect the application's logic.

Timestamp dependence. [Timestamps](#) are markers of time created by nodes executing smart contracts. Because it is difficult to perfectly synchronize every node, timestamps can be manipulated to generate logic attacks against contracts executing time-critical provisions.

Integer underflows and overflows. Arithmetic operations, which are used to verify transactions, usually operate within a fixed range of values in smart contract execution. When these operations fall outside value ranges, it can lead to unexpected changes and invalid operations in contracts.

Information and function exposure. Blockchains are public and accessible to anyone. So confidential information saved to a blockchain without being encrypted can leave it



open to nefarious abuse.

Examples of smart contracts

Smart contracts are used in various industries, most popularly decentralized finance ([DeFi](#)) and NFTs. Popular smart contract examples in those fields include the following:

DeFi

DeFi is one of the most significant use cases for smart contracts. DeFi applications use smart contracts to provide financial services without the need for intermediaries such



While smart contracts offer numerous benefits, they also come with vulnerabilities including several cyber attack possibilities.

as banks. These applications facilitate lending, borrowing, trading and yield farming.

One of the most popular decentralized exchanges using smart contracts is Uniswap. Using the Ethereum blockchain, Uniswap allows users to trade cryptocurrencies without banks or other central authorities.

NFTs

NFTs are one of the most popular use cases of smart contracts. They allow users to represent immutable ownership of unique digital assets and enable their secure creation, ownership and transfer. CryptoKitties was one of the first games that used NFTs. Created in 2017 on the Ethereum network, it allows users to buy and trade digital cats. The most expensive digital cat, Dragon, [was sold](#) in 2018 for approximately \$170,000.

4 Things to Know About NFTs (Non-Fungible Tokens)



The future of smart contracts

Smart contracts are complex, and [their potential goes beyond the simple transfer of assets](#). They can execute transactions in a wide range of fields, from legal processes to insurance premiums to [crowdfunding](#) agreements to financial derivatives. Smart contracts have the potential to disintermediate the legal and financial fields by simplifying and automating routine and repetitive processes for which people currently pay banks and lawyers sizable fees.

While Ethereum is the most popular blockchain platform for smart contracts, rivals like Cardano have emerged as new ecosystems for developers to build innovative and secure [decentralized applications](#). The integration of artificial intelligence ([AI](#)) is also emerging as an innovative development for smart contracts. AI algorithms can analyze data and execute actions based on predefined rules embedded in smart contracts. This combination of AI and smart contracts can enable automation and intelligent systems that can adapt and respond to real-world events in real time.

The role of lawyers could also shift in the future as smart contracts gain such capabilities as adjudications of traditional legal contracts and customizable smart contract templates. Additionally, smart contracts' ability not only to automate processes, but also to control behavior, as well as their potential for real-time [auditing](#) and risk assessments, can be beneficial for compliance.

Smart contracts also show promise in automating processes that run on [IoT](#) and edge computing devices. For example, a utility company might offer a service in which smart contracts execute in response to changes in power rates in coordination with devices built into power meters. For example, when prices reach a given threshold, a smart contract might automatically turn off or turn down power-hungry appliances such as air conditioners using a specially controlled IoT controller.

Another potential use case is integrating smart contracts into vending machines that could release goods in response to cryptocurrency payments.

In a supply chain scenario, smart contracts might unlock funds once a cargo container has arrived at its destination and IoT sensors indicate that it has remained unopened and the contents have been kept at the right temperature, appropriate humidity and not jostled too much on the journey.

Smart contract history and creation

The notion of smart contracts was first proposed by Nick Szabo in 1994. Szabo is a legal scholar and cryptographer known for laying the groundwork for digital currency. Back then, there was little interest or activity in smart contracts because there was no digital platform or distributed ledger technology that could support them.

In 2008, the bitcoin cryptocurrency was developed on a blockchain network with a distributed ledger that tracks monetary transactions. This technology enabled the development of smart contract code that is used to enter the terms of the contract into the blockchain.

Many platforms now allow for the use of smart contracts, including Ethereum, Hyperledger, Tezos and Corda. Today, with the growing adoption of bitcoin and the [support of blockchain technologies](#), smart contracts are growing in popularity.

Smart contracts are primarily used on blockchain technology. Familiarize yourself with [today's top blockchain use cases and industry applications](#).

This was last updated in September 2023

➤ Continue Reading About smart contract

- [Top 9 blockchain platforms to consider](#)
- [8 blockchain-as-a-service providers to have on your radar](#)
- [10 blockchain use cases in finance that show value](#)
- [5 tips to successfully implement blockchain for businesses](#)
- [How to become a smart contract developer](#)

Related Terms

chief financial officer (CFO)

A chief financial officer (CFO) is the corporate title for the person responsible for managing a company's financial operations ... [See complete definition](#) ⓘ

chief strategy officer (CSO)

A chief strategy officer (CSO) is a C-level executive charged with helping formulate, facilitate and communicate an ... [See complete definition](#) ⓘ

information governance

Information governance is a holistic approach to managing corporate information by implementing processes, roles, controls and ... [See complete definition](#) ⓘ

➤ Dig Deeper on Risk management and governance

Top 9 smart contract

Top 9 blockchain platforms to

platforms to consider in 2024

By: George Lawton

consider in 2024

By: George Lawton

decentralized application (DApp)

By: Nick Barney

blockchain

By: Nick Barney

Latest TechTarget resources

CLOUD COMPUTING



MOBILE COMPUTING

DATA CENTER

SUSTAINABILITY AND ESG

Cloud Computing



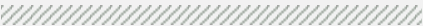
How to Create an AWS Lambda Function with CloudFormation

Do you want to enhance automation, consistency, scalability and cost-effectiveness? Follow this step-by-step tutorial on how to ...



IBM focuses on platform-based plan for hybrid cloud and AI

For decades, pundits have suggested that IBM has lost relevance. IBM wasn't in trouble then, and it's not in trouble now. See how...



[About Us](#)

[Editorial Ethics Policy](#)

[Meet The Editors](#)

[Contact Us](#)

[Advertisers](#)

[Partner with Us](#)

[Media Kit](#)

[Corporate Site](#)



[Contributors](#)

[Reprints](#)

[Answers](#)

[Definitions](#)

[E-Products](#)

[Events](#)

[Features](#)



[Guides](#)

[Opinions](#)

[Photo Stories](#)

[Quizzes](#)

[Tips](#)

[Tutorials](#)

[Videos](#)

All Rights Reserved, [Copyright 2007 - 2024](#), TechTarget

[Privacy Policy](#)

[Do Not Sell or Share My Personal Information](#)