

Deposit to earn rewards

Sign up and deposit to receive up to **10,055 USDT** in bonuses.
Exclusive for new users only.

Get it now

[PDF Database Document] - BTCC Cryptocurrency Exchange

Original:

<https://www.btcc.com/en-US/academy/crypto-basics/a-deep-dive-into-smart-contracts>

A Deep Dive into Smart Contracts

What Is Smart Contract?

Smart contracts (also known as “distributed applications”) are getting extremely popular today. But, what exactly are smart contracts? And what problems can they be used to solve?

The term “smart contract” was first coined by Nick Szabo in 1997, long before the birth of [Bitcoin](#). He is a computer scientist, jurist and cryptographer. So here we’ll explain this concept to you in a much simpler way instead of quoting his original words. In short, Nick Szabo wanted to use distributed ledger technology to store contracts.

Smart contracts today are just like contracts in the real world. The only difference is that smart contracts are completely digital. In fact, a smart contract is actually a small computer program stored in the [blockchain](#). Let’s take an example to see how it works.

How Does A Smart Contract Work?

You’ve probably heard of Kickstarter, one of the largest fundraising platform in the United States. Product teams can create their projects on Kickstarter and set a funding goal to raise money from those who believe in and support their ideas.

Kickstarter is essentially a third-party platform that sits between product teams and their financial backers. This means that both of them have to trust Kickstarter to handle their funds properly as they want. If the project is successfully funded, the project team expects to get the funds raised from Kickstarter. Likewise, backers ask that their investments to be used by the project they support or to be returned to them if it hasn’t reached its funding goals. In this case, both the product team and the funder have to trust Kickstarter.

But using a smart contract, we can also build a similar system that doesn't need a third-party platform like Kickstarter. Now let's create a smart contract for this!

We can program the smart contract to hold all the received funds until the set funding goal is reached. Those who support the project can transfer their money to the smart contract now.

If the project can meet the goal, the contract will automatically transfer the money to the project creator. And if the project fails to raise the targeted funds, the money will automatically go back to the supporters.

Pretty awesome, right? Also, since smart contracts are stored on the blockchain, it means that all information is completely distributed. With this technology, no one can control the money.

Why Use Smart Contracts?

But, wait a minute! Why should we trust this smart contract?

That is precisely because smart contracts are stored in the blockchain, they naturally have some interesting properties: They are immutable and distributed.

Immutability means that once a smart contract is set up, it can never be changed. Therefore, no one can tamper with the code of your smart contract behind your back. And distributed, means that the output of your contract needs to be confirmed by everyone in the network.

So it's impossible for a single person to force the contract to release the funds because other people on the blockchain network will definitely find out what you're trying to do and mark it as invalid. Therefore, it is almost impossible to tamper with smart contracts.

Other Use Cases of Smart Contracts

Of course, smart contracts can not only be applied to crowdfunding, but also to many other different areas. For example, banks can issue loans or provide automatic payments with smart contracts. Insurance companies can use it to handle certain claims. Postal companies can use it for delivery settlements, etc.

Well, now you may be wondering where and how you can use a smart contract. There are already several blockchain platforms that support smart contracts, and one of the largest is [Ethereum](#). It was specifically established and designed to support smart contracts. They can be programmed using a special programming language, Solidity. This programming language was created specifically for Ethereum, and its syntax is similar to JavaScript. Notably, [Bitcoin](#) also supports smart contracts, although there are more limitations compared to Ethereum.

By now, you should have an idea of what smart contracts are and what problems they can solve.

As always: thank you for watching and hope this video can help you!