

Smarter Contracts on the Blockchain: A Conversation with Bracewell's Jeonghoon Ha

January 10, 2023

By: [Seth D. DuCharme](#), [David A. Shargel](#) and [Jeonghoon Ha](#)

On this episode of Crypto Bits, New York associate [Jeonghoon Ha](#) joins hosts [Seth DuCharme](#) and [David Shargel](#) to discuss smart contracts, a central feature of the crypto and blockchain ecosystem.

What is a smart contract?

Smart contracts are computer programs, or transaction protocols on blockchain, that self-execute when certain conditions are met. Most listeners probably know about bitcoin blockchain created by Satoshi Nakamoto that basically introduced a decentralized digital payment system. That's why everyone's really excited about this technology of Vitalik Buterin, founder of Ethereum.

Buterin envisioned introducing more advanced applications like decentralized exchange, financial derivatives on blockchain identity and reputation systems. He wanted to do it on Bitcoin blockchain initially, but because of inherent limitations of Bitcoin blockchain, he really couldn't do it. So, he went ahead and developed his own blockchain called Ethereum.

This blockchain basically allows deployment creation, deployment and execution of smart contracts that we know now. Today when we talk about smart contract, we are in the Ethereum world.

How does a smart contract work? What makes a contract "smart"?

What you need to do first is to create a contract account on the Ethereum blockchain. This is different from a crypto wallet account that most of us think about when we talk about accounts on a blockchain system. So, the contract account says basically that crypto cannot be used to exchange cryptocurrency.

On Ethereum blockchain there will be ether, but you need to create a separate account called the contract account. After you do that, you need to use a solidity to write your own computer program. It's a particular type of programming language that Ethereum uses. You could hire a software engineer to help you convert certain terms in natural language into computer language, or you could learn how to do that yourself and create your own smart contract if you have time and ability to learn about solidity.

Will the creation and coding of smart contracts become easier in the coming years as smart contracts become more utilized and widespread?

Based on my conversation with some software engineers who work on smart contracts on a day-to-day basis, it sounds like the standards that Ethereum has on its own blockchain system is very simple for software engineers. Basically, it's as simple as one of the limitations of Ethereum blockchain and the smart contract technology is very simple.

What are some examples of real-world smart contracts?

One example is writing a program to run an auction as if you're going on eBay and doing auction to buy a certain item. You would first define the kind of digital assets that is being auctioned, and then you write various functions to run that auction. You would have a function to have participants make bids and then you would have a function to say when the highest bid comes, that person would be the highest bidder. You would also have a function saying that at certain time this auction will end. Lastly, the function saying that at that moment when the auction ends, the highest bidder will be the winner. When that function is triggered, that would be the automatic execution of the smart contract.

Have questions about smart contracts? Email [Seth DuCharme](#), [David Shargel](#) and [Jeonghoon Ha](#).

The opinions expressed in this podcast are those of the speakers and do not necessarily reflect the viewpoint of their institutions or clients.