itworldcanada.com

Understanding Cybersecurity Management in DeFi (UCM-DeFi) – Introduction to Smart Contracts and DeFi (Article 2) - IT World Canada

Sepideh HajiHosseinKhani and Arash Habibi Lashkari

16-20 minutes

Smart contracts are digital versions of traditional legal agreements programmed to automate the execution of terms and conditions without intermediaries. They are a crucial part of many blockchains and distributed ledger technology applications. However, their immutability can make modifications challenging. Continue reading to learn more about smart contract fundamentals, decentralized finance, and their historical development.

Understanding Cybersecurity Management in DeFi (UCM-DeFi, a five article series, aims to discuss decentralized finance and explore a range of cybersecurity issues that impact DeFi and blockchain-based financial solutions. The articles in this series are based on the recent book titled *Understanding Cybersecurity Management for DeFi*, published by Springer this year. This second article discusses the fundamentals of Smart Contracts, essential operations and how can we use Smart Contracts along with main advantages and disadvantages. It also introduces

Decentralized Finance and compares it with Centralized Finance and discusses the importance of Oracles in this domain.

Contents

History of smart contracts

Smart contracts were first proposed by Nick Szabo in 1994 with the aim of enhancing POS terminal functionalities. The idea is to communicate transaction semantics between parties while ensuring observability, verifiability, privacy, and enforceability. The concept of smart contracts predates the advent of cryptocurrencies, but has since flourished with the development of Ethereum.

Fundamentals of smart contracts

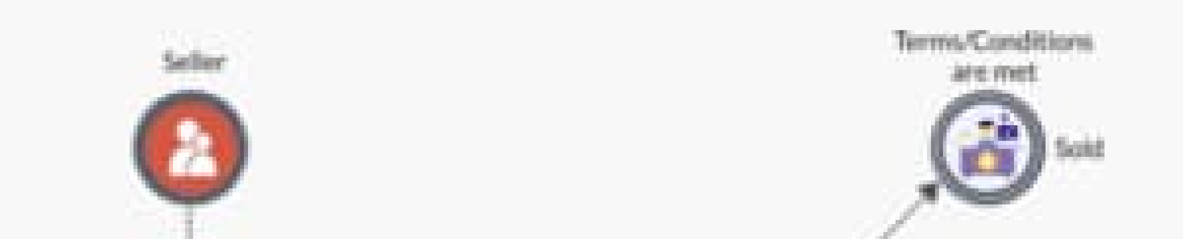
Smart contracts are sophisticated computer programs stored on blockchain-based platforms designed to facilitate, verify, and enforce the performance of contractual agreements between parties in a decentralized manner. As a digital alternative to traditional paper-based contracts, they automate the execution of agreements, eliminating the need for intermediaries and reducing costs associated with enforcement and execution.

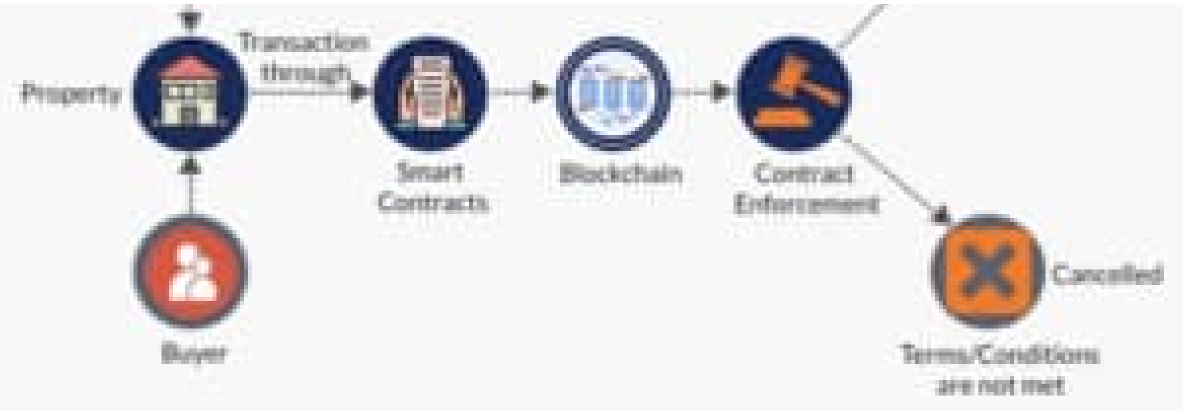
At their core, smart contracts possess several critical attributes:

- They are computer programs that automatically execute once predetermined conditions are met
- They do not require human intervention
- They are both immutable and deterministic

These characteristics contribute to heightened security, trust, and transparency in transactions, offering significant advantages over conventional contracts. Smart contracts can be designed to handle intricate transactions, manage digital assets, or even interact with external data sources via oracles, providing dynamic and adaptable solutions to modern challenges.

Smart contracts can facilitate property transactions in a real estate example (as seen below) involving a buyer and a seller. Both parties agree on terms and conditions, which are then incorporated into a smart contract. The contract executes the transaction once the conditions are met. These conditions are stored as blocks in a blockchain, ensuring a secure and transparent process.





The operation process of smart contracts

The operational process of smart contracts is built upon three fundamental components: blockchain technology, programming languages, and cryptographic proof of work. These elements work together to ensure secure, transparent, and automated transactions. Ethereum, a leading blockchain platform, employs a state transition function to execute smart contracts and manage transactions.

1. **Blockchain technology** – serves as the underlying infrastructure that stores transaction data in a decentralized and secure manner. It ensures the immutability and transparency of smart contracts.
2. **Programming languages** – such as Solidity for Ethereum, are used to code the logic and conditions of smart contracts, defining their behavior and execution.
3. **Cryptographic proof of work** – contributes to the security and integrity of the blockchain network by validating transactions and preventing double-spending or tampering.

In Ethereum, every transaction initiates with a start state and concludes with an end state. A valid state transition occurs when the start and end states maintain consistency throughout the transaction process. Transactions are grouped into blocks, which are securely linked together via cryptographic hashes, forming the blockchain.

The typical process for executing a transaction on the Ethereum network involves the following steps:

- Verify the transaction's structure and validity by ensuring it adheres to the network's consensus rules and protocol.
- Calculate the transaction fee, authenticate the sender, and deduct the required payment amount from the sender's account.
- Execute the transaction by updating the blockchain's state and transferring the deducted amount to the recipient's account.

The state transition function governs the changes that occur between the initial state and the new state during a transaction. If the transaction is successful, the function results in an updated state. Conversely, if the transaction fails, the function produces an error, and the state remains unchanged. This process is critical for ensuring the accurate and secure execution.

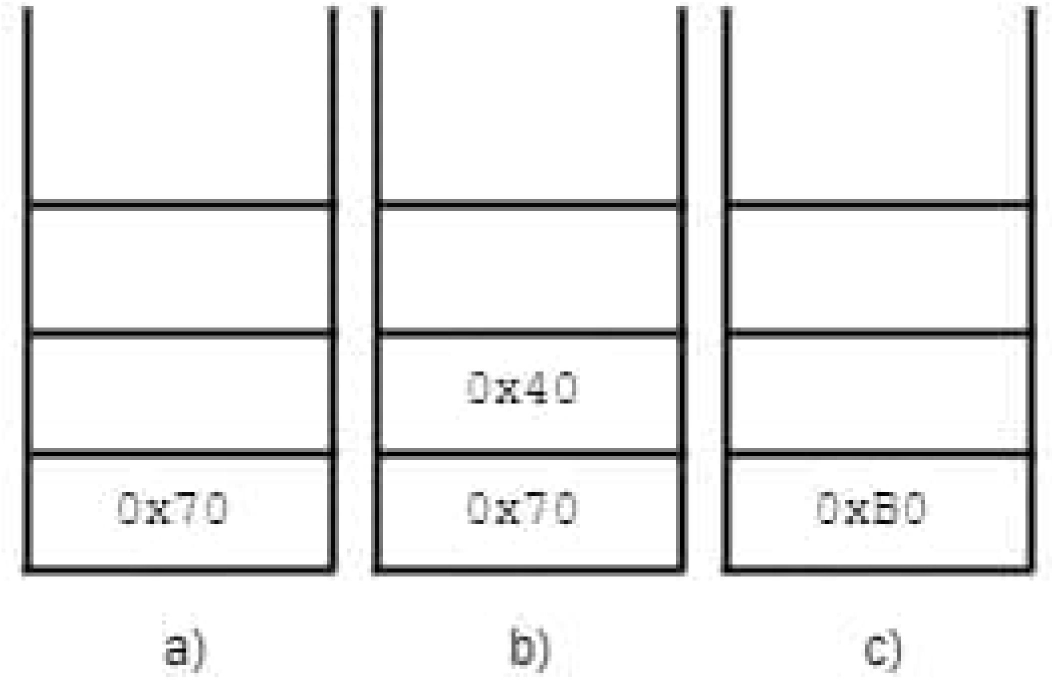
Technical operational process

The technical operational process of smart contracts in Ethereum uses the Ethereum Virtual Machine (EVM) to execute bytecode on a simple stack machine. Ethereum has its own virtual currency, Ether, and stores Turing-complete programs as EVM bytecode. Some important terms in the operational process include:

- **Address** – A unique 160-bit hexadecimal hash contract address identifying the bytecode.
- **Gas system** – Measures the computational effort required for

smart contracts and calculates the execution fee.

In the given context, the figure below illustrates the state of the stack after each step in a smart contract transaction, such as the real estate example mentioned earlier. This visual representation can help users better understand how the smart contract operates by showcasing the progression of the transaction and the changes in the stack state as the contract moves through each condition or step.



When a function in the source code is called, the corresponding bytecode is generated upon compiling. The bytecode is stored in every node in the blockchain. EVM splits bytecodes into opcodes to execute tasks. In summary, for smart contracts to work, parties need to apply their signature technologies in blockchain applications, and the exact conditions must be coded in the smart contracts for automation.

How can we use smart contracts

Smart contracts offer versatile applications across various industries, automating processes, enhancing transparency, and reducing overhead costs. Some key areas where smart contracts have demonstrated their potential include:

- **Healthcare** – Securely store patient records, manage hospital billing, and track supply chain data, facilitating efficient data sharing among healthcare institutions.
- **Supply chain management** – Securely store patient records, manage hospital billing, and track supply chain data, facilitating efficient data sharing among healthcare institutions.
- **Financial services and insurance** – Expedite insurance claims processing, payment transfers, and optimize mortgage and loan procedures, fostering greater efficiency.
- **Voting systems** – Implement secure and tamper-resistant voting mechanisms to protect the integrity of electoral processes.
- **Digital identity** – Safeguard individuals against digital fraud and streamline Know Your Customer (KYC) processes for frictionless identity verification.
- **Financial data recording** – Boost accuracy and transparency in financial record-keeping while reducing auditing costs and streamlining compliance.
- **Government** – Automate processes, property management, and auditing procedures to minimize costs and enhance efficiency in the public sector.
- **Clinical trials** – Improve cross-institutional visibility and automate data sharing in clinical research, fostering collaboration and innovation.

- **Real estate** – Establish a centralized database ledger for property information, reducing legal costs and enhancing transparency in property transactions.
- **Smart grids and critical infrastructures** – Integrate blockchain technology into energy trading, power generation, and critical infrastructure protection, preventing cyber-attacks and optimizing resource management.
- **Gaming** – Develop and manage games of chance, skill-based games, and hybrid gaming experiences through the use of smart contracts, ensuring fairness and transparency.

Benefits and problems of smart contracts

Smart contracts offer several benefits, including accuracy, transparency, speed and efficiency, security, reduced cost, decentralized validation, trust, and data redundancy:

- **Accuracy** – Smart contracts minimize human errors by automating the execution of agreements based on predefined conditions.
- **Transparency** – By storing transactions on a public ledger, smart contracts promote transparency and foster trust among parties.
- **Speed and Efficiency** – Automated execution of smart contracts accelerates transaction processes and enhances overall efficiency.
- **Security** – Blockchain technology and cryptographic mechanisms provide robust security, protecting smart contracts from unauthorized alterations or tampering.
- **Reduced Cost** – By eliminating intermediaries, smart contracts can significantly reduce the costs associated with contract

enforcement and execution.

- **Decentralized Validation** – Blockchain networks validate smart contracts in a decentralized manner, eliminating the need for centralized authorities.
- **Trust** – The immutability and transparency of smart contracts foster trust among parties involved in transactions.
- **Data Redundancy** – Blockchain networks store smart contract data across multiple nodes, ensuring data durability and redundancy.

However, they also have some limitations:

- **Immutability** – Once set up, smart contracts are difficult to modify, leading to practical issues related to changing terms and conditions.
- **Contractual Secrecy** – Although participants remain anonymous, the transactions are visible on the public ledger, potentially compromising privacy.
- **Legal Enforceability** – The enforceability of smart contracts is still uncertain, and there is potential for improper translation from traditional legal contracts.
- **Understandability** – Smart contracts require a shift from traditional legal language to code, necessitating specialist knowledge for an accurate representation of terms and conditions.
- **Signature Verification** – Digital signatures consume significant computing power and energy, making them a challenge in blockchain technology.

Introduction to DeFi

Decentralized finance (DeFi) has reinvented financial systems by allowing investments, payments, and other transactions without relying on traditional financial institutions. DeFi operates in a decentralized manner, using digital wallets for users to store and exchange cryptocurrency. This approach eliminates the need for paperwork and increases transparency in transactions.

DeFi is built on Ethereum blockchain technology and smart contracts, which automate transaction execution and validation without intermediaries. This innovative approach offers peer-to-peer financial services through a decentralized network. DeFi has a significant market capitalization and has reduced transaction costs and the risk of monopoly in financial systems.

DeFi characteristics

DeFi leverages smart contracts to ensure automation, transparency, and immutability in transactions. It is decentralized, with blockchain data replicated across multiple nodes, ensuring data availability, and eliminating single points of failure. DeFi's security is enhanced by data encryption, making it autonomous and resistant to manipulation.

Participants in DeFi do not need to identify themselves beforehand, and there is no central authority governing regulations. Transactions are authenticated through digital signatures. DeFi is permissionless, using open-source technology that allows users to contribute to its development and innovation.

Examples of decentralized financial applications and platforms include Ethereum, Bitcoin, and Libra. DeFi's foundation on public blockchains and open standards increases interoperability and enhances the value of the internet.

DeFi vs CeFi

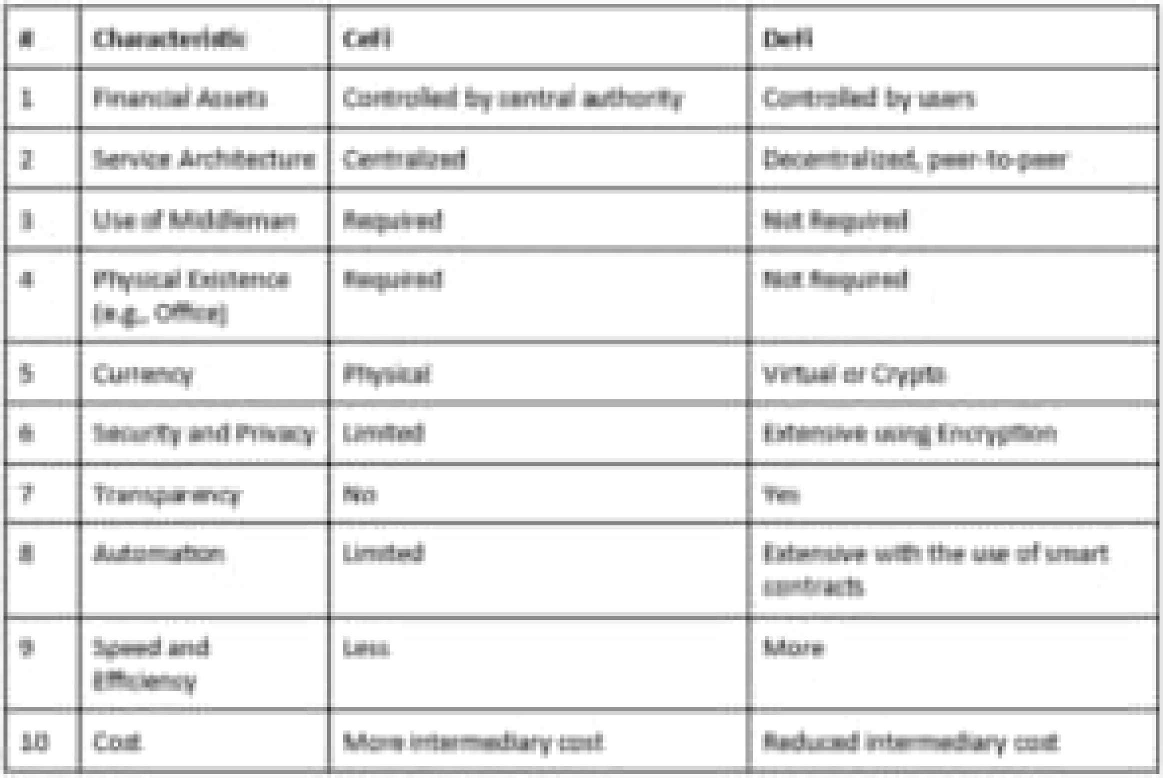
Centralized finance (CeFi) originated in ancient Mesopotamia and has evolved into the traditional financial systems involving banks, stock exchanges, and brokers that are the underpinnings of our financial systems today. CeFi is characterized by a monopoly controlled by a central governing body, with intermediaries necessary for transactions.

In contrast, decentralized finance (DeFi) eliminates intermediaries and enables peer-to-peer transactions. DeFi offers improved speed, efficiency, and reduced costs. Key differences between CeFi and DeFi include:

- **Financial assets** – Controlled by a central authority in CeFi, while users control them in DeFi.
- **Service architecture** – Centralized in CeFi, decentralized and peer-to-peer in DeFi.
- **Middleman use** – Required in CeFi, not required in DeFi.
- **Physical existence** – Required in CeFi, not required in DeFi.
- **Currency** – Physical in CeFi, virtual or crypto in DeFi.
- **Security and privacy** – Limited in CeFi, extensive with encryption in DeFi.
- **Transparency** – Absent in CeFi, present in DeFi.
- **Automation** – Limited in CeFi, extensive with smart contracts in DeFi.
- **Speed and efficiency** – Less in CeFi, more in DeFi.

- **Cost** – Higher intermediary costs in CeFi, reduced intermediary costs in DeFi.

Here’s a table that summarizes these differences in a user-friendly format:



#	Characteristic	CeFi	DeFi
1	Financial Assets	Controlled by central authority	Controlled by users
2	Service Architecture	Centralized	Decentralized, peer-to-peer
3	Use of Middleman	Required	Not Required
4	Physical Existence (e.g., Office)	Required	Not Required
5	Currency	Physical	Virtual or Crypto
6	Security and Privacy	Limited	Extensive using Encryption
7	Transparency	No	Yes
8	Automation	Limited	Extensive with the use of smart contracts
9	Speed and Efficiency	Less	More
10	Cost	More intermediary cost	Reduced intermediary cost

DeFi applications

DeFi, or Decentralized Finance, is a blockchain-based financial ecosystem that emphasizes transparency and operates on open protocols and decentralized applications (DApps). DeFi leverages smart contracts to facilitate secure, verifiable transactions across various financial applications. Some prominent DeFi use cases include:

DeFi exchanges

Decentralized crypto exchanges, such as Uniswap and SushiSwap, enhance transparency and trust in trading by utilizing smart contracts and blockchain-based settlement, bypassing traditional intermediaries.

Lending pools

Platforms like Maker, Compound, and Aave enable users to lend and borrow cryptocurrencies through smart contracts, providing transparency, liquidity, and additional benefits to both lenders and borrowers.

Derivatives

Crypto derivative platforms, including Synthetix and UMA, aim to mitigate risks associated with crypto asset exposure. They employ smart contracts to create secure tokens that help users hedge against future price fluctuations.

Insurance

DeFi insurance projects, like Nexus Mutual and Etherisc, leverage blockchain technology and smart contracts to streamline the settlement of claims, enhancing transparency and reducing the likelihood of errors or disputes.

Gaming

Blockchain-based gaming, also known as GameFi, grants players complete control over virtual assets and incorporates cryptocurrency networks for an enriched gaming experience. Examples of GameFi platforms include Superplayer World and Axie Infinity.

NFT

NFTs represent unique digital assets, such as art or virtual creations, with ownership recorded in smart contracts on a blockchain. Prominent NFT platforms include CryptoKitties, CryptoPunks, and Rarible.

Importance of Oracles in the rise of DeFi

Blockchain Oracles are third-party services that provide off-chain data to smart contracts, acting as a bridge between blockchains and the outside world. They widen the scope of smart contracts by enabling access to external information. Oracles can be classified based on source, the direction of information, and trust, and can be software, hardware, or human.

Oracles enhance the performance, interoperability, and functionality of smart contracts by bringing trust and transparency to various industries. They provide blockchains with real-time information from external sources such as stock markets, political events, and weather updates. Oracles are used in different DeFi applications like lending pools, automated market makers, and stablecoins.

However, Oracles introduce a single point of failure in decentralized environments, known as “the Oracle problem.” Oracle providers like Chainlink and Oraclize are working to address this issue. Despite the problem and trustworthiness concerns, Oracles have enabled the interfacing of blockchain data with off-chain resources.

What’s next

This article introduced Smart Contracts and Decentralized Finance. It digs into the main idea behind the Smart Contracts and focuses on the related critical attributes. It also presented the fundamental components of Smart Contracts operation, and how to use them. Finally, it focused on DeFi, the difference between DeFi and CeFi, DeFi applications, and importance on blockchain Oracles. The next article of the series, entitled “Understanding Cybersecurity Management in DeFi: DeFi Platforms,” will introduce the popular blockchains that support DeFi, the security and safety of the platforms, and the security evaluation methods.