S&P Global

Smart contracts could improve efficiency and transparency in financial transactions

September 2022

Table of Contents

Introduction

Authors:
Ashish Sinha, *Associate Director*, S&P Global Ratings,
Vanessa Purwin, *Analytical Manager*, S&P Global Ratings
Mohamed Damak, *Senior Director*, S&P Global Ratings
Erkan Ertuk, *Senior Director*, S&P Global Ratings

Key takeaways

- Smart contracts are a core building block of decentralized applications, facilitating the execution of specified tasks (such as payments) based on predetermined business rules.
- Using smart contracts in financial transactions can improve efficiency and reduce reliance on third parties like asset servicers and custodians, as well as make transaction resolutions faster--enhancing creditworthiness and the integrity of business dealings.
- Smart contracts have had slow and limited adoption in the financial markets owing to key risks including technology issues (such as incorrect coding) and legal and regulatory ambiguities that make accountability difficult.

Imagine your business was expecting delivery of 1,000 laptops. Instead, you receive only 400 and the order is closed. Checking the deal details on the wholesaler's web portal reveals--surprisingly--only 400 were in fact ordered. The rate card menu appears inflated as well.

Who changed the order on the web portal—and how?

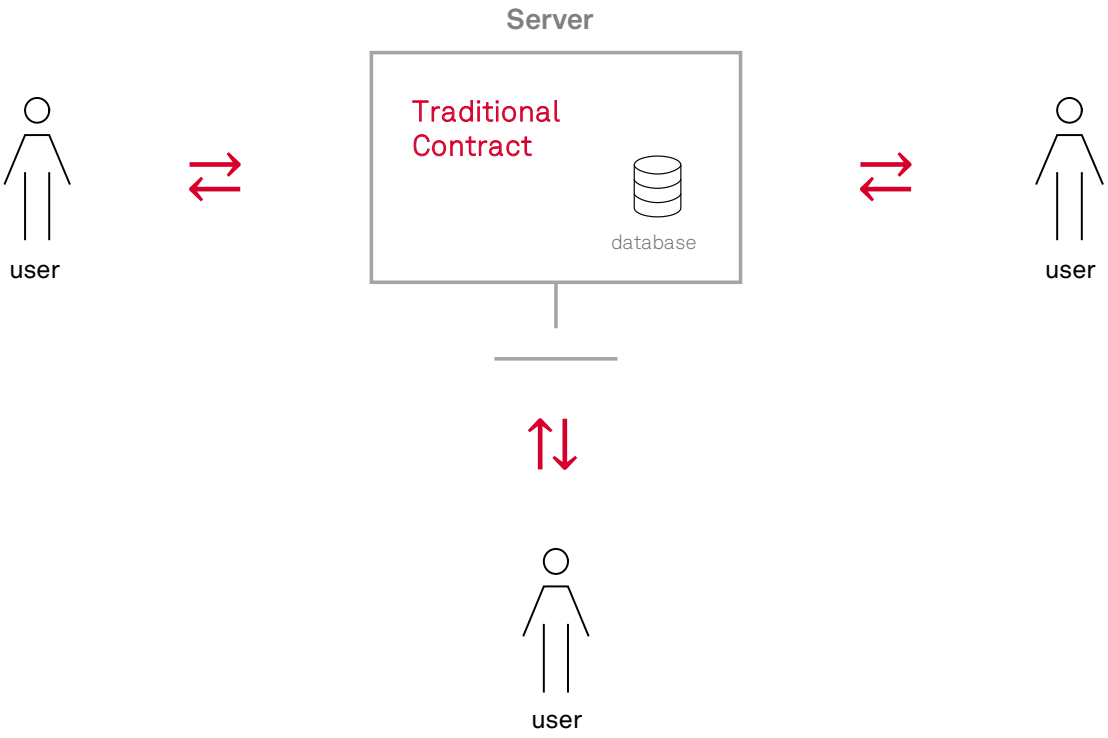
Suing the wholesaler would mean spending time, effort, and money. Regardless, you still have fewer laptops than you needed. Only the wholesaler and the web portal management company know of their under-the-table deal to directly and fraudulently update the rate card in the database, **completely bypassing** the business rules coded in the web portal application.

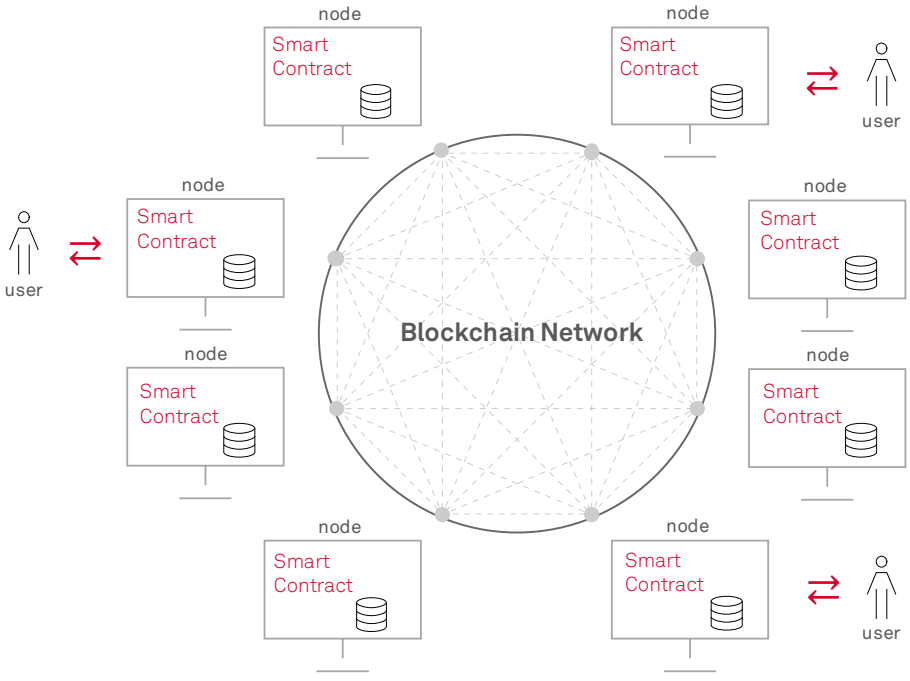
An alternative scenario: updates to the database are allowed only through consensus from all parties involved in the business (your company, the wholesaler, the logistics providers, and the financing firm). Only when all of these groups run the business rules of the application to ensure an update is genuine and provide their consent would a database update be made.

In blockchain networks, the application (containing the business rules) and the database are not hosted centrally with a single party. Instead, all parties in the network have their own copy of the database and their own copy of the application, called a smart contract (for more background on the blockchain technology, see "Digitalization Of Markets – Framing the Emerging Ecosystem", published on Sept. 16, 2021). **Smart contracts** are business rules (i.e. the clauses of a contract) written as software code and deployed at multiple parties in a blockchain network. This ensures that, for any single party, the only way to update the databases of all of the parties (which represents database of the network as a whole) is if they execute their own copies of the smart contract and **all** approve the update. Market participants typically qualify this execution as automatic, though there are nuances.

Back to our laptops example. The smart contract would say *"only if status of order is not confirmed, allow update of rate per piece."* If the wholesaler tried to update the rate per piece, since they all have a copy of the original smart contract the other participants would not approve the proposal, consensus on the transaction would not be achieved, the transaction would be rejected by the network as a whole, and the rate per piece would not be updated in each participant's database.

Traditional contracts vs. Smart contracts





How they work: consensus building

Contractual validity

This first phase of consensus building is contractual validity. When one party wants to update the network's database, they create a proposal containing the new value and the name of the relevant smart contract. The proposal travels through the network and all (or a designated set of) parties on the network view the proposal and verify its authenticity by executing a simulated run of the smart contract. If the simulation shows that the output is same as the proposed new value, the parties provide their consent through their digital signatures. However, if the simulation shows that the output is different from the proposed new value, it's evident that the proposer is trying to update the database without conforming to the business rules coded in the smart contract. All other parties would deny approval by not providing their digital signatures on the proposal.

Even with traditional contracts, every transaction should comply with the contractual obligations and be verified and approved by all parties. However, any fraudulent change made by a party that has privileged access to the centralized database can take time to be unearthed and involve lengthy and costly legal proceedings. In the world of blockchain and smart contracts, such fraudulent activity would be very difficult to achieve in the first place.

Timestamping and integrity

The second phase of consensus building involves timestamping and an integrity check. Every transaction is timestamped to ensure the correct sequence of transactions in the network using consensus-building algorithms like Proof of Work, Proof of Stake, and Raft. Correct timestamping ensures fairness and that no party in the network gets an unjustified business advantage over the others. Integrity checks and timestamping also prevent double-spending by ensuring that every transaction is cryptographically linked to the previous one. So, a transaction trying to double-spend will not be accepted as the facts it tries to present would be different from reality (e.g., a current balance of 2 BTC when it's actually 0 BTC).

How they're used: common applications

Advantages of decentralization make blockchain and smart contracts eligible for many applications. Any situation involving multiple parties conducting business transactions could use blockchain, and hence smart contracts. Traditional legal contracts can be expensive and time-consuming to draw up, and sometimes even incomplete. Blockchain and smart contracts can address concerns around this lack of transparency and greatly reduce, if not eliminate, the role of intermediaries. However, in some cases, intermediaries need to be retained and in fact form an important part of the network, such as regulatory intermediaries.

 |

Decentralized digital securities

Assuming issuers, investors, and regulators are all on a blockchain network, the issuance, re-sale, and redemption of securities would be done through smart contract business rules. There is no need for a centralized entity to host the application and facilitate the workflow. Smart contracts also allow developers to code how the security should function under different scenarios. This allows automatic execution of previously approved contractual agreements, rather than subjecting them to interpretations that could change over time.

Insurance

Insurance providers, customers, underwriters, and regulators are all on a blockchain network. Policy issuance, claim registration, underwriting, etc. can be done through smart contract business rules. For instance, a crop insurance smart contract could be programmed to ensure a given payment to the insured party if the temperature falls below a certain level at a given point or during a given duration. No centralized entity is needed to host the application and facilitate the workflow.

Supply chain logistics

Logistics providers can issue load tenders, carriers can submit regular manual, on-demand, and automated shipment statuses, suppliers can notify on shipments, and carriers can notify on invoices all on the blockchain with the help of smart contracts to facilitate real-time visibility and transparency among the participants. Smart contracts could also validate an invoice against the load tender to calculate any penalties.

KYC

For peer-to-peer Know-Your-Customer (KYC) sharing, customers' Personally Identifiable Information (PII) is obtained from current KYC records across multiple organizations and a hash of the PII is placed on the blockchain. Member organizations can submit requests for PII on the blockchain and receive them one-on-one off-chain. If the information available on the blockchain is not satisfactory, the organization can perform a traditional KYC check and update the hash as a contribution to the ecosystem. This is all done by smart contracts and no centralized entity is needed to host the application and facilitate the workflow.

Registration of immovable property

Ownership changes of real estate property are recorded on the blockchain with smart contracts replacing the traditional transfer of ownership via paper deeds that must be locally registered. This ensures a non-tamperable and publicly verifiable encumbrance record of the property, preventing forgery and fraud. Lantmäteriet, the land registry in Sweden, successfully piloted a program leveraging blockchain to record the transfer of real property, though legislative changes would be required for formal adoption.

Decentralized exchanges

Smart contracts in decentralized exchanges (DEXs) enable the sale and purchase of crypto assets in exchange for other crypto assets. They also ensure the exchange is free of counterparty risk by transferring the assets between the two parties in a single transaction. The transaction either completely goes through or completely fails.

Decentralized autonomous organizations

All of the organization's workflows are decentralized through blockchain and coded through smart contracts, including poll-based updates. For example, a smart contract may be used to tally votes and execute the outcome of the vote without any intermediary intervention, or a smart contract may be set up to facilitate distribution of funds in a predetermined manner if a particular event occurs.

Regulatory ambiguities

Smart contracts are an integral part of financial and nonfinancial applications and blockchain platforms, but many questions surrounding accountability and how regulation would be applied and where remain unanswered. Legal and regulatory ambiguities around enforcement can raise questions around the validity of results when smart contracts are executed, how contract changes are executed if needed, as well as around what party, if any, investors have recourse to if a smart contract fails or is programmed incorrectly (see "Regulating Crypto: The Bid To Frame, Tame, Or Game The Ecosystem," published July 13, 2022).

A key challenge is the variation in regulatory approaches to blockchain and cryptocurrency around the world. Since networks are decentralized, key jurisdictions will need to refine their approaches to create more consistent global regulation. One area ripe for this is stablecoins, which are meant to be tied to an external reference (and therefore less volatile than other cryptocurrencies). Stablecoins can replace the chain of intermediaries and service providers linking payers and payees, while underlying smart contracts automate back-end processes and simplify transactions on a commonly distributed digital ledger. These smart contracts would thus come under scrutiny as well to ensure they function as specified.

This also comes into play with smart contract-based decentralized lending. True decentralization makes it tough to identify persons or businesses that can be held accountable through regulations. While regulators could identify the physical location where the smart contract was deployed on the blockchain and apply the governing law in that jurisdiction, the pseudonymity of the developers who deploy smart contracts and the code's protection as a form of free speech in the U.S. make this difficult. For instance, leading decentralized finance (DeFi) lenders such as Aave and Compound operate autonomous and trustless smart contracts without a central authority. Because DeFi gives the financial responsibility--including asset custody and investments--back to users, greater regulatory clarity around smart contracts is needed.

Another example is DEXs. DEXs' underlying smart contracts operate autonomously to enable trading and are accessible by anyone. DEXs are not currently regulated, largely because they are not owned or operated by an entity or nor management-run, which would make natural persons or legal entities liable under traditional law. Also, it is currently very difficult to implement KYC and anti-money laundering/combating the financing of terrorism (AML/CFT) checks, as pseudonymous wallet owners can directly connect to the DEX platforms.

In general, the balancing act between protecting users while also encouraging innovation is one of the most difficult tasks policymakers and regulators face, for example, when it comes to regulating crypto tokens and underlying smart contracts. Crypto tokens are typically integrated into protocols that are programmed using smart contracts on public blockchains. Tokens are primarily used to create incentives that facilitate activity in an application and the surrounding network. Incentives are integral for blockchain projects to build scale, making crypto tokens an essential component of a project's entrepreneurial strategy.

How they could affect credit: benefits and risks

BENEFITS	RISKS
Intermediary displacement	Reliance on oracles
Reduced costs	Code defects, bugs
Increased efficiency	Programming limitations
Greater transparency	Legal/regulatory uncertainty
	Operational complexity

Key benefits

Smart contracts--as the core building block of decentralized applications--present multiple opportunities for financial instruments. Increased efficiency, greater transparency, and reduced dependency on transaction counterparties can improve companies' bottom lines and support overall enhanced creditworthiness. Using smart contracts in a financial transaction can reduce exposure to the traditional operational risks posed by reliance on asset servicers, physical custodians, and other conventional third parties. Smart contracts can also make transaction resolutions faster provided all scenarios have been coded in advance and all participants have common understanding of the outcomes. As such, they could help avoid costly legal proceedings. Instead of going to court to execute legal provisions in a transaction, in the digital world execution could be automated as long as the scenario is already coded in the contract. This in itself could enhance the creditworthiness and the integrity of transactions. If a margin call is executed, for example, and one of the parties does not uphold its contractual obligations, the transaction is liquidated automatically to avoid any additional losses that may be caused by market turbulence, saving significant resources. Smart contracts could also strengthen consumer protection as they could be coded to comply with specific laws, providing clarity on each party's exact obligations from inception.

Areas of concern

While smart contracts can confer significant benefits, they also have their own limitations and introduce unique risks that have contributed to their slow and limited adoption. For example, since smart contracts run on distributed infrastructure and are part of consensus building, they are deterministic and hence cannot access any external services/information. This makes them rely on oracles that work as trusted data feeds (i.e. bridging blockchain and the real world). Oracles can take several forms: hardware or software, and inbound or outbound. The most common are software oracles that pull third-party data from the internet such as real-time flight statuses or financial asset prices. Hardware oracles utilize sensors to extract data from physical objects such as radio-frequency identification (RFID) tags in supply chain logistics. As their respective names suggest, inbound oracles may be hardware or software oracles that bring data into the smart contract, while outbound oracles disseminate data from the smart contract to external parties. However, oracles are generally from centralized points of origin that typically require third-party permission, which dilutes some of the benefits of decentralization that smart contracts confer. Further, oracles may be vulnerable to the very security and veracity issues that smart contracts are purported to address, creating a crack in the smart contract armor.

Programming limitations also pose challenges for the broader use of smart contracts. In public blockchain networks like Ethereum, where the size and capability of member nodes cannot be determined and network abuse needs to be kept in check, simpler programming languages like Solidity are used instead of very advanced languages like Java and C#. Though easier to learn, Solidity has very few data structures, limited arithmetic precision, and overflow/underflow (values outside of a specified data range). In contrast, private permissioned networks do not face this challenge as they support smart contracts built with Java, Go, Kotlin, Javascript, etc., which are more flexible in adapting to the requirements of more complex smart contracts. However, the use of permissioned rather public blockchains deemphasizes the benefits of a decentralized network.

Smart contract language options

In private/permissioned blockchains, there are multiple choices of smart contract languages. For example, Hyperledger Fabric users can choose from Go, Javascript, and Java, while Corda lets users choose between Kotlin and Java. Most of the time, the following factors determine the choice:

- Performance factors
- Language stack of existing reference architecture to allow smoother integration with legacy applications
- Skill set availability
- License cost
- Any distinct features provided by the language that might be the need of the hour

In public blockchains, Ethereum, and other layer 2 solutions that work on Ethereum Virtual Machine (EVM), support Solidity, Vyper, Pyramid Scheme, Flint, LLLL, HAssembly-evm and Bamboo. Hedera supports Solidity. Most of the time, the following factors determine the choice:

- Developer friendliness/skill set availability
- Any distinct features provided by the language. For example, Bamboo is a language without loops but with explicit constructor invocation at the end of every call

As yet another software module built with code, smart contracts can have defects, bugs and change requirements, and thus require maintenance and enhancements like any other traditional application. Ensuring that the smart contract is coded correctly and that all the stakeholders have the same understanding of the different outcomes is paramount to the effectiveness of smart contract use. Otherwise, since public blockchains are built on open-source code, hackers could potentially exploit these weaknesses to their advantage. Auditing and attesting of the quality and validity of the smart contract could therefore be an important consideration to reassure investors.

A key consideration from a regulatory and legal point of view is that parties may need to rely on a technical expert to transfer the agreement into code or ascertain that third-party code is a reliable representation of the agreement. Unless they are programmers, most will not likely be able to decipher what is actually in the smart contract. Agreements between the business parties and these experts can help protect users and determine who might be liable if there's a coding error in automation.

Further, while smart contracts may disintermediate traditional transaction parties and the operational risks associated with their roles, they present technical complexities that pose their own distinct challenges to the smooth functioning of security issuances. Considerations such as backup provisions in the event of protocol failure or disruption, the depth of the market for replacement, and interoperability of systems if a transfer is needed could affect transaction performance and hence the credit quality of the securities issued. Given smart contract technology is relatively nascent, alternative providers remain limited and the resiliency of backup provisions largely untested, which could expose investors to greater disruptions than traditional financial contracts with more established redundancy procedures. The ultimate impact of smart contracts on creditworthiness will depend on how these concerns are addressed.

This report does not constitute a rating action.