

TECHNOLOGY

Beyond the Blockchain Technology Hype: When Does it Make Sense to Use Smart Contracts?

by Bert-Jan Butijn, Jens K. Roehrich, Kostas Selviaridis, and Wendy van der Valk



Image Credit | Pierre Borthiry - Peiobty

Businesses must know when to use smart contracts.

✔ **INSIGHT** | FRONTIER 20 Feb 2023

The Proliferation of Smart Contracts

Building trusting business-to-business relationships is a challenging task, yet vital for any business. In the absence of trust between businesses, relationships may be confronted with excessive monitoring costs, such as those associated with ensuring that contractual obligations have been met before making payments and enforcing contracts when suppliers do not uphold their end of the bargain or act opportunistically. Because of the extensive effort and cost involved, companies grapple with safeguarding business transactions from opportunistic behavior, and other issues such as late delivery and faulty products.

RELATED CMR ARTICLES

“The Boon and Bane of Blockchain: Getting the Governance Right” by Curtis Goldsby & Marvin Hanisch. (Vol. 64/3) 2022.

“Transformational Transparency in Supply Chains: Leveraging Technology to Drive Radical Change” by Cory Searcy, Pavel Castka, Jakki Mohr, & Sönke Fischer. (Vol. 65/1) 2022.

Blockchain-based smart contracts offer a solution for companies and supply chains (**McGrath et al., 2021; Goldsby and Hanisch, 2022**), having a direct impact on their business strategy and operations (**Searcy et al., 2022**). They help enforce contractual obligations by autonomously performing transactions using funds held in escrow that are only executed when predefined conditions are met (**Luu et al., 2016**), thereby essentially promoting contract-based trust (**Sako, 1992**).

Smart contracts are increasingly being adopted across sectors. For example: Maersk, the Danish shipping company, experimented with smart contracts to track their supply chains on **TradeLens**, while banks like **JPMorgan Chase** and **Goldman Sachs** now have invested

in their own initiatives. Recently, Google announced that it will invest in blockchain too (Novet, 2022). Gartner (Gartner, 2021) has predicted that smart contracts will become a critical component of the global digital infrastructure. Despite the technical vulnerabilities smart contracts and blockchains may have (see Madnick, 2019), a recent survey by Deloitte (2021) involving 1,488 senior executives indicated that they believe blockchain technology will be in their top five strategic priorities for the coming two years. Similarly, a report by the World Economic Forum (2021), based on interviews among 200 executives from leading financial firms, reported that smart contracts could greatly reduce transactions costs for inter-firm transactions.

Despite smart contracts' increasing adoption, there is limited actionable guidance for managers and companies on *when* to use blockchain-based smart contracts, i.e., which kinds of business transactions can be effectively served using smart contracts. Smart contracts are not a one-size-fits-all solution. Actually, our evidence obtained through numerous interviews with managers, lawyers, and coding experts; industry workshops and conferences; content analysis of over 50 industry and government reports; building models of over 30 smart contracts; and rich case evidence across various sectors including shipping and finance, suggests that blockchain-based smart contracts may not be appropriate for specific sets of contracting situations. Based on our rich insights, we developed a framework to aid managers and companies to make decisions regarding when to use smart contracts to govern business transactions.

Blockchain Technology and Smart Contracts

Formal contracts are legal instruments used by businesses to safeguard against the negative consequences of a business partner's inappropriate behavior. Here, provisions are aimed at contractual control, i.e., provisions that clarify each party's rights and responsibilities and help to diminish opportunistic behavior and hence relationship conflict (Roehrich et al., 2020). Similarly, smart contracts - in their simplest form - are explicit, usually written, and legally enforceable agreements in computer code, that specify the roles and obligations of contracting parties. Although the concept of smart contracts

was pioneered in the 1990s by Nick Szabo (**Szabo, 1997**), it did not gain traction initially due to the lack of supporting technology infrastructure. The advent of blockchain, however, reinvigorated the concept of smart contracts for businesses.

Business transactions executed through a smart contract are verified by the blockchain's rigorous consensus protocol, thus validating every single transaction. The execution of this protocol entails that all participants in the network (called nodes) verify the transaction. When declared valid by the majority of nodes, business transactions are stored in containers (called blocks), that are cryptographically chained to each other to render them tamperproof. These containers are vital to secure the integrity of the decentralized ledger blockchain technology. The tamperproof ledger then provides all contracting parties with a transparent record of bilateral facts and their evolution, allowing for monitoring without requiring costly replication between companies (**Felin and Lakhani, 2018**). Another attractive feature of smart contracts is that they can hold funds in escrow, and once all conditions are satisfied, the nodes in the blockchain network ensure that the transaction is automatically performed (e.g., parties are paid) (**Luu et al., 2016**). To guarantee that no one can tamper with the smart contract, it is stored on a blockchain's ledger to render it immutable once deployed.

Smart contracts thus provide enormous benefits when it comes to the operational stages of the purchasing process - ordering, expediting, payment, and evaluation. For example, a smart contract allows automatically checking deliveries in terms of quantity and quality against the corresponding order and associated contract terms, and sanctions payments accordingly. Smart contracts can also accommodate performance-based agreements, with payments (bonuses or penalties) paid/deducted reflecting suppliers' performance. Even more contingencies could be considered, such as sources of transaction uncertainty. For instance, smart contracts might preclude a penalty payment when poor performance was caused by force majeure (e.g., see COVID19).

However, in reality companies are involved in a wide range of business transactions, featuring diverse characteristics and requirements. Accordingly, contracts may serve purposes other than controlling trade partners. Specifically, governing business transactions characterized by complexity requires designing and invoking contractual provisions aimed at coordination and adaptation (**Schepker et al., 2014**). The coordination

function of contracts relies on provisions that help address relationship conflict through the reduction of task conflict, and the promotion of information exchange and joint problem solving, as well as by specifying actions and related milestones (**Schilke and Lumineau, 2018**). Contractual clauses aimed at coordination emphasize how the exchange process should be monitored and evaluated. As such, the focus of the coordination function is on the process itself rather than the outcomes thereof. Such contracts involve a certain level of flexibility, allowing contracts to be adjusted considering emerging developments. For example, in large and often long-term infrastructure projects such as building bridges, airports, or power stations, changes to a customer's requirements, and a myriad of other coordination requirements associated with additional technical expertise possibly being required, dynamics in the political environment may occur.

Adaptation contracts are most appropriate in the presence of uncertainty (**Leiblein, 2003**), i.e., when exogenous or endogenous changes create the need for provisions that define mutually agreed on tolerance zones or procedures for dealing with changes (Schepker et al., 2014). Examples of exogenous changes include changes to, or volatility in, demand or supply: consider contracts for raw materials or commodities (e.g., wheat), the prices of which may be highly fluctuating in general, and prone to unforeseen events as the ongoing conflict in Ukraine teaches us. Consequently, the contract may include provisions that cover the general price fluctuations of goods in terms of tolerance zones, and associated risk/gain sharing mechanisms, as well as provisions that detail how, and under what conditions, prices outside the tolerance zones are adjusted. Endogenous changes relate to reducing uncertainty or improving ways of dealing with uncertainty, for example as a result of learning. Here, the contract would need to specify what learnings will be incorporated and how, with some changes possibly even resulting in a revised contract. However, once deployed, smart contracts cannot be revised, but only removed altogether. A revision of a smart contract would require all of the data stored in the initial smart contract to be transferred to the revised smart contract. Needless to say, this is a complex technical problem which is difficult to overcome.

Aligning Transaction Characteristics with the Design of Smart Contracts

As previously discussed, coordination and adaptation contracts are substantially different from contracts aimed at control. Whether such functional characteristics can fruitfully be captured in smart contracts depends on two important prerequisites. First, it must be possible to capture the complexity of the contractual agreement. A contractual agreement can be considered complex when several contingencies need to be foreseen. However, a smart contract would need to know these contingencies in advance to enact upon their occurrence. In other words, bounded rationality cannot be a limiting factor, meaning that smart contracts may be most appropriate for business transactions involving relatively simple exchanges for which contractual safeguards can be enforced. For example: whether what is delivered actually meets, or does not meet, specifications is easy to foresee for many business transactions. Timeliness of payments is another example of a clause which can be quite easily transferred to a smart contract environment. In more complex and uncertain business transactions (high transaction complexity) however, some contractual elements may be incomplete, which by definition also makes the smart contract incomplete. For example, parties may not be able to identify certain contingencies or not acknowledge the need to specify them (**Bernheim and Whinston, 1998**). Also, actions may be observable for one party only, or impossible to write down in a way that can be legally enforced by a third party (i.e., a court; **Lyons, 1996**). Contracts may also be left open for purposes of freedom and flexibility, or simply to save costs. In general, complex contracts are less suitable to be captured in smart contracts.

Second, it must be possible to codify the contractual agreement into smart contract code in an unambiguous and legally enforceable way. **Lumineau et al. (2021)** refer to this prerequisite as codifiability, that is whether it is possible to structure information or knowledge into code. Contracts designed to provide safeguards usually involve a large variety of detailed contractual clauses, e.g., to draw specific attention to how companies must act in case of a contract breach or conflict, and when and how to terminate the contract. Codifying some provisions like those describing simple fixed-price payments for goods is a relatively easy task. However, the codification of clauses related to amendments, or undoing, a contract for instance, are more difficult as they are at odds with the

immutability property of smart contracts ([Marino and Juels, 2016](#)). Even if these problems could potentially be remedied by the introduction of novel mechanisms, this would require the inclusion of additional code in the smart contract. Each execution step that the smart contract has to perform increases the cost of execution. Besides increased costs, additional code would have to be written and tested, hence adding to the overall development costs of the smart contract. Consequently, when programming increasingly elaborate contractual agreements, the cost and time consumed may start to outweigh the benefits of smart contracts. Consequently, firms might need to evaluate for each business transaction whether it is worthwhile to pursue smart contracts.

The challenges regarding contractual complexity and codifiability suggest that smart contracts may not be applicable for all types of business transactions. A more nuanced approach regarding their design and use might thus be timely to consider.

When to Use Smart Contracts?

In essence, transaction-related contingencies may be coded into a smart contract using several parameters that can be changed depending on the situation, but this requires that these contingencies can reasonably be foreseen, and associated scenarios (i.e., parameters that are affected and associated bandwidths; adjustments taking place under specific conditions) can unambiguously be described. Subsequently, capturing these details in computer code should be relatively easy and efficient; otherwise, the cost and time of programming may start to outweigh the benefits of having a smart contract to govern business transactions.

Accordingly, based on our in-depth research, we distinguish four types of contracting situations, depending on two key conditions: (1) ease of capturing transaction complexity; and (2) ease of converting contract clauses into code (Figure 1). Starting with the latter, a smart contract is not feasible when contingencies and associated scenarios cannot (efficiently) be captured in computer code. When functional characteristics are difficult to codify and programming becomes challenging, and hence time- and cost-consuming, companies may better stick to either traditional contracts for simpler transactions, or to trust-based exchange governance for more complex transactions.

Typically, codifiability decreases with transaction complexity; yet, even characteristics of relatively simple transactions may be difficult to capture in code. Typically, coordination contracts may contain provisions that cannot be verified, e.g., related to the amount of effort exerted in or to the capabilities of employees assigned to projects. Terms like “reasonable” or “fair” cannot simply be expressed in quantities, and are therefore difficult to code. For instance, it would be difficult to code what it means that a party has made “reasonable efforts” to prevent an accident from occurring. When contingencies can be captured in computer code, companies may opt for partial or fully smart contracts. Partially smart contracts are most appropriate when contingencies, and their potential impact in terms of size and direction, cannot reasonably be foreseen. In such cases, parties may for instance use a smart contract to execute payment clauses only and resort to traditional means of enforcement for the remainder of the clauses. Future contracts for instance, guarantee that a party can buy a specific asset at a fixed price in the future. The transactions that future contracts concern are already being executed automatically at stock exchanges, however when disputes between the parties arise they resort to non-automated clauses. In contrast, fully smart contracts should be pursued when contingencies and associated scenarios can be identified and unambiguously be described beforehand. Examples of such contracts are employment contracts and insurance contracts. The clauses in, for instance, an employment contract are clear; when an employee is employed a monthly salary will be provided. In most cases leave days are known in advance, and can be monitored using the smart contract.

Contracts for complex infrastructure projects, for example, are characterized by performance challenges that put pressure on time schedules and milestones, and contingency factors and tolerance zones for time and cost can probably be identified. However, in some cases, performance problems cannot readily be attributed to the supplier only, and discretionary decision-making regarding whether to execute penalties is required. For instance, when confronted with labor strikes that impede construction activities, the business exchange may be better served by engaging in dialogue and building trusting relationships rather than by executing a fully smart contract.

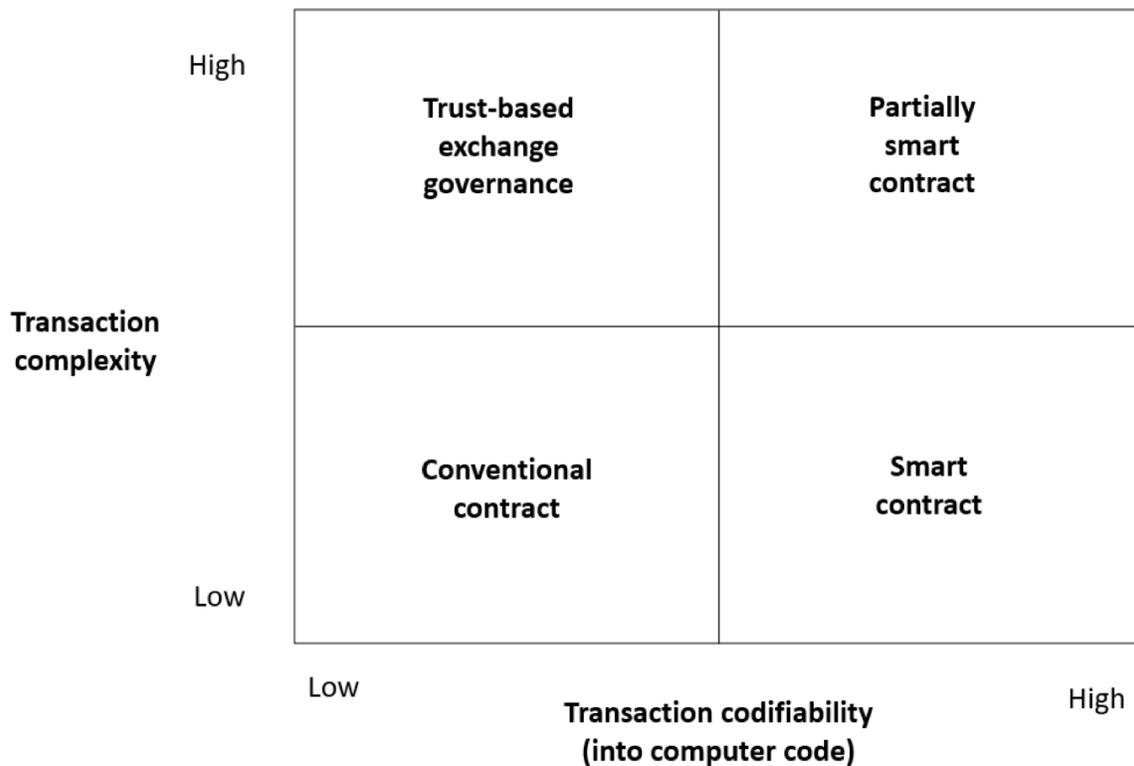


Figure 1 *When should your business use smart contracts?*

Conclusion

Some business exchanges would suit the sole use of a smart contract to substitute a traditional legal contract when the contract is self-executing (i.e., when contractual provisions are automatically enforced). We argue, however, that smart contracts are less suited for more complex exchanges characterized by complexity and uncertainty, evolving requirements, and associated adaptations. In such situations, parties may benefit from complementing the use of smart contracts (or partially smart contracts) with trusting relationships.



Bert-Jan Butijn [Follow](#)

Dr Bert-Jan Butijn has conducted his Ph.D. research at the Jheronimus Academy of Data Science, the Netherlands, in the field of blockchain technology. He is currently employed as Data Scientist and Information Analyst at EY. IT has always been of great interest to him.



Jens K. Roehrich [Follow](#)

Professor Jens Roehrich is Chair in Supply Chain Innovation at the University of Bath, School of Management, United Kingdom. His research, consultancy, media engagement, and teaching activities focus on long-term inter-organizational relationship management across public and private organizations, and the role of technology and governance.



Kostas Selviaridis [Follow](#)

Dr Kostas Selviaridis is Associate Professor of Operations Management at Lancaster University Management School, United Kingdom. His research explores how contracting promotes innovation, resilience, and sustainability goals in supply chains. His work influences public policy, for example with respect supporting technology-based small businesses and securing supply of medicines.



Wendy van der Valk [Follow](#)

Professor Wendy van der Valk is the Nevi Chair in Purchasing and Supply Chain Management at Tilburg University, the Netherlands. Her research focuses on the effective governance of interorganizational relationships, specifically the contracting of business-critical services, in relation to contemporary transitions in business and society, i.e., digital transformation and circularity.