2024

When to Use Smart Contracts Instead of Traditional Contracts – A Conceptual Analysis

Anna Hupe
University of Kassel, Germany, anna.hupe@uni-kassel.de

When to Use Smart Contracts Instead of Traditional Contracts – A Conceptual Analysis

Research Paper

Anna Hupe

University of Kassel, Information Systems, Kassel, Germany
anna.hupe@uni-kassel.de

Abstract. Smart contracts are code-based representations of business logic that encapsulate predefined rules that they verify and execute autonomously. As self-executing agreements, these contracts running on a blockchain facilitate interactions between untrusted parties, eliminating the need for intermediaries. They speed processes, increase transparency, and facilitate data exchange across national and organizational boundaries, providing a competitive advantage. However, there are significant costs associated with implementing smart contracts. In our study, we conduct a comparative analysis of these costs and compared them to traditional contracts. Using a combination of literature review and expert interviews, we dissect the cost structures of both approaches at an abstract level. We then construct a matrix to systematically compare traditional contracts with smart contracts. This framework allows us to model the conditions that favor the use of smart contracts over traditional ones, and to identify instances where each approach has a distinct advantage.

Keywords: Smart Contract, Transaction Costs, Conceptualization, Conventional contracts.

1 Introduction

Since Szabo (1996) defined smart contracts already back in the mid-1990s as a "*computer protocol that digitally facilitates, verifies, and enforces the contract's performance*", the term should be widely known and applied by organizations. Eggers et al. (2021) see smart contracts as the "*next revolutionary innovation toward extensive automation*". However, to date, smart contracts are rarely implemented by organizations. This is despite the fact that they offer several solutions, such as automating processes, reducing transaction costs, eliminating sources of error, and increasing trust between contracting parties. Smart contracts can be implemented wherever business logic can be defined as a clear if-then relationship. Current automation technologies such as enterprise resource planning systems, workflow management systems, and robotic process automation differ from smart contracts in one important way: the location of their storage. Smart contracts are stored on a blockchain, its content is stored in a decentralized manner. After they are programmed and stored on a blockchain, they serve three functions. They (1) store rules, (2) verify rules, and (3) self-execute them (Khan et al., 2021). Although there has been some research on smart contracts (Huang et al., 2021; Kannengieser et al., 2022; Khan et al., 2021; Lacity and van Hoek, 2021; Liu, 2020), no one has yet looked at how they compare to traditional contracts. However, comparing traditional contracts to smart contracts is an important first step in facilitating their implementation. It provides the basis for making informed decisions for or against a smart contract. With our study, we provide a basis for organizations and individuals to classify whether and in which cases a traditional contract should be used and when the use of a smart contract is appropriate. The following research question forms the basis of our study:

RQ: "*In what cases and under what conditions are smart contracts more appropriate than traditional contractual arrangements*?"

To answer the research question, we compare the costs of creating and executing a traditional contract with the costs of creating and executing a smart contract. In doing so, we focus on the most prominent blockchain on which smart contracts run: Ethereum. This paper is structured as follows: first, we explain the concept of smart contracts and use cases where smart contracts are already implemented. Second, we present other use cases that are particularly suitable for the application of smart contracts. In the third part, the methodological part, we explain the procedure of our literature review, the expert interviews and the framework on which our analysis is based. In the fourth part, we examine the cost structures of conventional and smart contract arrangements. We use the matrix developed in part 3 as the basis for this analysis. The underlying assumption is that smart contracts are perfectly programmed and free from corrupting influences. We then summarize and discuss our results (part 5) and present a conclusion and the contribution of our research to theory and practice (part 6).

2 Background

2.1 Smart Contracts

According to the first existing definition, given by Szabo (1996), "*a smart contract is a set of promises, specified in digital form, including protocols within which the parties perform on these promises*". Richard et al. (2020) state that smart contracts "*enforce autonomous verification in any transaction in the network under some predefined conditions*". Thus, smart contracts are promises in digital form, with accompanying protocols that fulfill those promises. Programs that execute a protocol under certain circumstances and are stored on a blockchain can be categorized as a smart contract (Zheng et al., 2020). Smart contracts map business logic and execute agreements and transactions, even complex ones, between blockchain participants automatically and without intermediaries (Zheng et al., 2020). They are computer protocols that map, verify, or provide technical support for the execution of contracts (Taherdoost, 2023). A smart contract is a program code written to be deployed, verified, and executed on a blockchain platform (Simić et al., 2021). It is immutable, decentralized, open source, self-executing, accurate, and fast. Therefore, the smart contract technology is an attractive decentralized application that enables computations on top of a blockchain (Wan et al., 2021). In order to run and execute a smart contract, it is necessary to have a money exchange via cryptocurrency to incentivize validators to solve cryptocraphic puzzles. Therefore, in the following analysis, we assume that the appropriate and necessary wallet is set up and available. Since the analyzed case focuses on the Ethereum blockchain, the gas fees are paid in Ether (Li, 2021). The fee for the operation of smart contracts is called "*gas fee*" (Li, 2021). The exact cost of the gas fee depends on supply and demand, similar to the price of electricity. When all this is set up, smart contracts allow users to create self-executing algorithms with immutable rules on blockchain systems (Richard et al., 2020). The computer protocols can manage large sums of money, execute asset transactions, and govern the transfer of digital rights between multiple parties (Liu, 2020). There is a consensus in the scientific community that the application domain for smart contracts is much broader than the currently widely known use cases in the financial sector (Cuccuru, 2017; Huang et al., 2021; Wan et al., 2021). Wherever business logic can be described as an if-then relationship by exchanging events and value, smart contracts (and their sub smart contracts) can be implemented (Huang et al., 2021). The principle of "*code is law*" applies (Wan et al., 2021). The three main tasks that smart contracts are overtaking are (1) storage of rules, (2) verification of rules, and (3) self-execution of the terms of an agreement between untrusted parties (Khan et al., 2021; Mohanta et al., 2018). Storage on the blockchain ensures that a smart contract is immutable and that transactions initiated by the contract are autonomously and truthfully executed. A smart contract is subsequently unalterable, making it all the more important for the parties involved to agree on all contract contents and contingencies.

2.2 Common Use Cases for Smart Contracts

As mentioned earlier, smart contracts have great applications when intermediaries need to be trusted and when contracting parties agree on a contractual object that can be described by simple or nested if-then relationships (Huang et al., 2021). By using smart contracts, processes can be managed with greater speed, efficiency, and accuracy. Trust and transparency can be created because there is no central third party involved and the encrypted records of transactions are shared with network participants (Khan et al., 2021). This eliminates the question of whether information has been altered for personal gain. Because of the blockchain approach, smart contracts create great security and enable savings by eliminating intermediaries and the associated time delays and fees. In the financial sector, smart contracts are used to make international trade faster and more efficient (Wang and Xu, 2022). Standardized rules agreed upon by the parties involved enable an ecosystem of trust. Banks as trusted parties become redundant and credit checks can be automated with smart contracts (Wang and Xu, 2022). Furthermore, smart contracts facilitate data exchange and thus serve as a driver for industrial business models (Chong et al., 2019). In the case of Walmart's pilot project, smart contracts enable automatic penalties for late deliveries (Lacity and van Hoek, 2021). The food companies Nestlé, Dole, Unilever, and Tyson Foods are part of the pilot project. By tracking deliveries, transfers are made automatically as soon as the goods are delivered. These smart contracts are processed via a private blockchain on Hyperledger Fabric. If deliveries do not happen within the agreed timeframe specified in the smart contract, a penalty payment is due (Lacity and van Hoek, 2021). According to Walmart, the pilot project has resulted in significant savings due to a reduction in transaction costs (Lacity and van Hoek, 2021). The exact amount of savings generated by the implementation of smart contracts has not yet been disclosed. In addition to their use in supply chains, smart contracts also have great potential applications in the public sector. They can be used to automate the transfer of ownership of real estate and the associated land registry entries (Shinde et al., 2019). This saves the parties involved expensive notary fees and time-consuming procedures. The smart contract would then formulate in a way that the land registry entry changes automatically (Shinde et al., 2019). This could be, for example, the receipt of the transfer of the real estate price from the buyer to the seller. Once the smart contract detects such a transfer ("*if function*"), the land registry entry is changed ("*then function*").

Estonia recently launched an "*e-residency*" program based on smart contracts that allows anyone in the world to apply for a "*transnational digital identity and authentication to access secure services, as well as to digitally encrypt, verify, and sign documents*" (Triana Casallas et al., 2020). Further applications in the public sector currently tested are registration of property titles (prototyped inter alia in Sweden, Georgia, Honduras and Ghana), tax administration and electronic invoice insurance (initiated in China), digital identity (initiated in Italy), managing identification of state residents (planned in the United States), transfer of ownership (planned in the United States) and online votes (tested inter alia in Denmark, France and The Netherlands) (Triana Casallas et al., 2020). According to the World Bank, close to 2% of the Global Gross Domestic Product ends up in the hands of corrupt government agents (Triana Casallas et al., 2020). As implementing smart contracts in the public sector creates transparency, the World Bank identifies them as a key technology to overcome corruption (Triana Casallas et al., 2020). Once the required infrastructure for the relevant use case is in place, there are significantly more applications for smart contracts. For example, self-driving taxis can independently bill users for their services (Li et al., 2023). In that case, the smart contract is programmed in a way that it opens the door if the car as soon as a solvent passenger is interested in a ride. The location of the car is tracked by an oracle as soon as the passenger is getting into the car. The oracle records the trip and then feds this information into the smart contract. The computer protocol calculates the amount to be paid, which is automatically debited from the deposited account when the passenger completes the trip and leaves the vehicle. Following the same principle, it is possible to automate the process of renting vacation homes, bicycles, etc. In the manufacturing industry, smart contracts enable networked machines to automatically request services. For example, repair slots are automatically booked when the machine needs to be repaired or serviced (Balcerzak et al., 2022). Smart contracts also provide use cases for facilitating and automatizing administration within organizations. In human resources, for example, smart contracts can be used to automatically manage bonuses and expense reimbursements. Besides, they can verify the originality of certificates and skills which assists the process of (intraorganizational) staffing.

3 Methodology

In order to compare traditional and smart contracts and their associated costs in a structured manner, we created a framework to guide our analysis. Specifically, we developed a matrix that allows us to capture and compare the factors relevant to setting up and operating both traditional and smart contracts and their associated costs. In the first column, we distinguish the number of contracting parties. We distinguish between cases where (1) two parties (1:1) or (2) more than two parties (1) are involved in the contract. For the contract duration in the second column, we analyze (1) non-recurring/one-time use contracts and (2) longer contract durations (see figure 1). For the different contract durations, we then distinguish the costs associated with the contract; we differentiate between one-time and recurring costs. The resulting matrix allows us to evaluate which type of contract is most appropriate in each case, focusing on the predictable costs incurred over the lifetime of the contract. In our model, we assume that the subject matter of the contract is fulfilled. Costs that may be incurred as a result of non-compliance are not included in the analysis to ensure comparability of cases.

Contractual Relationship	**Contract Period: Non-recurring**		**Contract Period: Recurring**	
	One Time costs*	**Recurring costs***	**One Time costs***	**Recurring costs***
1:1 contracting parties				
1:n contracting parties				
* Costs for contract creation, contract verification, and execution of contract, *assuming contract compliance*				

Figure 1. Analysis Framework for Comparing Contract Related Costs

As a basis for our analysis, we conducted a systematic literature review following vom Brocke et al. (2015) and using Cooper's related taxonomy (vom Brocke et al. 2015). We defined existing research on smart contracts and directly related concepts and technologies (phases 1 and 2) to provide a framework for our literature analysis. In addition to information systems databases AIS eLibrary and Information Systems Research Journals, we selected the computer science databases ACM Digital Library and IEEE (phase 3). By selecting highly ranked databases and including papers published in them, we ensure the quality of the papers included and thus the validity of our analysis. As suggested by Cooper in his taxonomy, it is also necessary to define keywords that form the basis of the literature analysis (vom Brocke et al., 2015). Our search strings (“smart contracts” AND “transaction costs”) and (“smart contracts” AND “gas fee”) and (“transaction costs” AND “business contract”) yielded a total of 1,886 potentially relevant documents as of February 10, 2024, with a starting year of 2008, as 2008 is considered the beginning of blockchain technology with the publication of the white paper around bitcoin . After reviewing the titles and abstracts, we divided the papers into the categories “highly relevant” and “relevant”, where the title and abstract indicated that the paper detailed transaction costs for smart or conventional contracts. If neither the title nor the abstract indicated that the paper detailed transaction costs in conventional or smart contracts, the paper was considered “not relevant”. This process resulted in 100 highly relevant and relevant papers. After reading the articles in their entirety, we were able to index 17 papers that we included in the analysis based on their content on transaction costs in smart contracts or conventional contract design (phases 3 and 4). Based on these highly relevant articles, we developed theoretical overview matrices of the cost structures of traditional and smart contracts (phase 5), which we present in the following sections of the paper. To validate and challenge our findings, we conducted a qualitative study. We interviewed twelve smart contract practitioners (table 1). Six of them are currently deploying smart contracts in their organizations, three are programming smart contracts, and three are lawyers who review smart contracts before they are translated into code.

Table 1. Overview of Interview Partners

Interviewee	Reason for including in data collection process
1 – 6	Smart contract users in banking, insurance and retail sectors
7, 8 , 9	Smart contract developers with ≥ 3 years of experience
10 , 11, 12	Legal professionals with ≥ 10 years of experience and an additional smart contract qualification for ≥ 3 years

In these semi-structured interviews, which lasted between 30 and 60 minutes, we presented our findings from the literature review and the matrix (figure 1). We then asked for their feedback and suggestions for changes. After transcribing the recorded interviews, we coded them using MAXQDA. We included all aspects and feedback of the analysis framework that were mentioned by three or more of the experts during the interviews. We included them in our model when three or more experts recommended the same changes. An overview of the methodological approach is shown in figure 2.

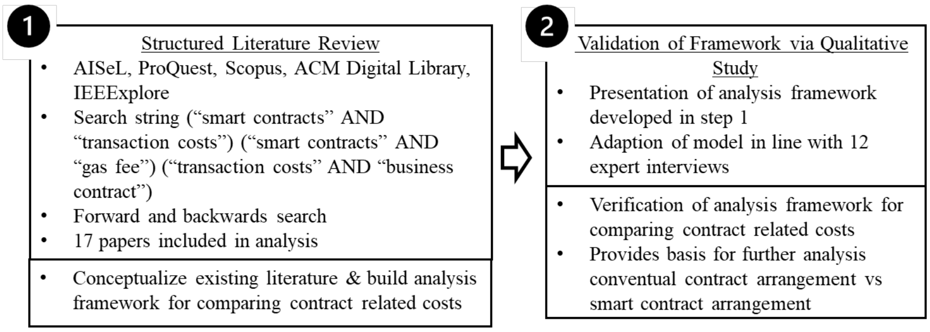

Figure 2. Methodological Approach

4 Analysis

4.1 Conventual Contract Arrangement

Following the analytical framework derived above (figure 1), we will first analyze the costs incurred in the case of *two contractors* (1:1) and a *single contract term*. We will then compare these costs to the case where *more than two contractors* (1:n) are involved. After analyzing the two one-time contract cases, we will consider the case of *recurring* contracts. Again, we start with the case where *two parties* are involved and then compare it to the case where *more than two parties* are part of the contract.

Many contracts are concluded by *two parties*. This applies for the majority of B2B supply contracts as well as for rental, employment, and sales contracts. In all cases, it is important that the subject matter and terms of the contract, as well as any related contingencies, are clarified and stated. The more complex the subject matter of the contract, the more likely it is that the parties will involve legal counsel in drafting the contract, also in the case of a *non-recurring contract* agreement. This ensures that the contract is legally binding and free of errors. The total one-time cost of agreeing on the content of the contract (including legal experts) comes at a cost; the more complex it is and the longer it takes to agree on its content, the more expensive the one-time cost of drafting it. Since in the first scenario the contract is *non-recurring*, there are no recurring costs.

In case *more than two parties* (1:n) are involved in the contract, the effort required to draft the contract is greater than in a two-party setting. This is because the interests of several parties must be coordinated when drafting the contract. Also, the effort to define the subject matter of the contract is obviously more complex than the process when only two parties are concerned. The more contractors involved and the more complex its content, the higher the coordination costs. Thus, the one-time costs in the 1:n setting are greater than those in the 1:1 setting. Since the duration of the contract is limited to a single event, no recurring costs arise.

In case of a *recurring contract* between *two parties* (1:1), the one-time costs are just as high as in the case of a non-recurring contract (see above). This is because in both cases, all contingencies must be addressed. In the case of a recurring contract, recurring costs are incurred because both parties must verify compliance with the agreed-upon subject matter. Depending on the application, this can also be done by third parties (service providers). The total recurring cost is therefore the sum of the verification costs of both parties.

In case *more than two parties* (1:n) are involved in a *recurring* contract, the one-time costs are just as high as in the case of a 1:n non-recurring contract. This is because in both cases, all possible contingencies must be addressed. The overall recurring costs (verification costs) in a 1:n setting are higher than in a 1:1 setting. Similar to the 1:1 setting, the verification costs comprise all expenses for checking the compliance of contractual objects, possibly completed by service providers. The more complex the contract setting (contracting parties and contract complexity) the higher the verification costs since the compliance with the contract must be checked by all contractors individually.

4.2 Smart Contract Arrangement

After having structured the costs of conventional contract creation (see figure 3), we will now do the same for smart contracts (see figure 4). In the first case, where a contract is *non-recurring* and concluded between *two contracting parties*, the onetime costs are higher than those for traditional contract creation. This is because additional costs are incurred because not only do all parties need to agree on the subject matter of the contract and clarify contingencies, but the agreement also needs to be translated into a smart contract and programmed accordingly. The process leading up to the actual programming of the smart contract is also more expensive than creating a traditional contract agreement. This is because lawyers with special training in smart contract law are needed to validate its content. Lawyers without this training are not permitted to review the content of a contract that is to be translated into a smart contract and declare it legally binding ("*Many legal professionals have not yet fully grasped what smart contracts are or how they function.*", expert 10). Lawyers with the additional training to declare a smart contract legally binding are rare these days, so they have high hourly rate ("*Many do not see the potential and therefore do not perceive the necessity to undergo the appropriate further education.*", expert 11). If the programmer has questions about the content of the contract during the coding process, both parties to the contract and the aforementioned legal experts will need to come to the table to resolve the ambiguity. The experts in our qualitative study agree that the cost of creating a smart contract is three to ten times higher than the cost of creating a traditional contract. Even though the contract is set up for a one-time event, there are still recurring costs. This is because the smart contract is responsible for verifying and enforcing its terms. It increases the cost of verification. These costs come from the price of gas and the amount of gas used.
If *more than two* (1:n) contracting parties enter a smart contract for a *one-time* contractual relationship, the one-time costs for agreeing on the content of the contract are greater than in the 1:1 case. Because more parties with different interests are involved in agreeing on the subject matter of the contract, the associated setup costs are higher than when creating a smart contract. Since the contract terms are more complicated than in the 1:1 contract constellation, the cost of translating and programming a smart contract based on the agreement(s) is also higher. Since the contract is agreed for a longer period of time, the smart contract continuously monitors the occurrence of the deposited conditions. For the continuous monitoring of the smart contract, a gas fee is permanently incurred (recurring costs). These costs are higher compared to 1:1 contractual relationship because the associated smart contract is more complex in 1:n

Contractual Relationship	**Contract Period: Non-recurring**		**Contract Period: Recurring**	
	One Time costs*	**Recurring costs***	**One Time costs***	**Recurring costs***
1:1 contracting parties	- Agreement and review of contract contents by **A and B** - Legal verification by **legal advisor** - Cost for verifying the occurrence of contractual objects by A and B (if necessary, by service providers) Costs < one time costs for 1:n contracting parties for conventual contracts	No recurring costs as contractual object is non recurring	- Agreement and review of contract contents by **A and B** - Legal verification by **legal advisor** - Cost for verifying the occurrence of contractual objects by A and B (if necessary, by service providers) Costs = one time costs for 1:1 contracting parties for conventual contracts	- Costs for verifying the occurrence and compliance of contractual objects by **contracting parties A and B** (if necessary, by service providers) - Overall verification costs = x * 2 Costs < recurring costs for 1:n contracting parties for conventual contracts
1:n contracting parties	- Agreement and review of contract content by **contracting parties (n)** - Legal verification by **legal advisor** - Costs for verifying the occurrence and compliance of contractual objects by **contracting parties (n)** (if necessary, by service providers) Costs > one-time costs for 1:1 contracting parties for conventual contracts	No recurring costs as contractual object is non recurring	- Agreement and review of contract contents by **contracting parties (n)** - Legal verification by **legal advisor(s)** Costs = one-time costs for 1:n contracting parties for conventual contracts	- Costs for verifying the occurrence and compliance of contractual objects by **contracting parties (n)** (if necessary, by service providers) - Overall verification costs = x * n Costs > recurring costs for 1:1 contracting parties for conventual contracts
* Costs for contract creation, contract verification, and execution of contract, *assuming contract compliance*				

Figure 3. Overview of Costs in a Conventional Contract Arrangement

Contractual Relationship	Contract Period: Non-recurring		Contract Period: Recurring	
	One Time costs*	**Recurring costs***	**One Time costs***	**Recurring costs***
1:1 contracting parties	- Agreement and review of contract contents by **A and B** - Legal verification by **special legal advisor(s)** - The contract's content is programmed as a smart contract by an **external party** and stored on Ethereum. Costs < one time costs for 1:n contracting parties for smart contracts > one time costs for 1:1 contracting parties for conventional contracts	- Costs for verifying the occurrence and compliance of contractual objects by **smart contract** - Overall verification costs = total gas used * gas price (gas price is fluctuating) Costs < recurring costs for 1:n contracting parties for smart contracts	- Agreement and review of contract contents by **A and B** - Legal verification by **special legal advisor** - The contract's content is programmed as a smart contract by an **external party** and stored on Ethereum. Costs = one-time costs for 1:1 contracting parties for smart contracts	- Costs for verifying the occurrence and compliance of contractual objects by **smart contract** - Overall verification costs = total gas used * gas price (gas price is fluctuating) Costs < recurring costs for 1:n contracting parties for conventual contracts
1:n contracting parties	- Agreement and review of contract content **by contracting parties (n)** - Legal verification by **special legal advisor(s)** - The contract's content is programmed as a smart contract by an **external party** and stored on Ethereum. Costs > one-time costs for 1:1 contracting parties for smart contracts	- Costs for verifying the occurrence and compliance of contractual objects by **smart contract** - Overall verification costs = total gas used * gas price (gas price is fluctuating) Costs > one-time costs for 1:1 contracting parties for smart contracts	- Agreement and review of contract contents by **contracting parties (n)** - Legal verification by **special legal advisor(s)** - The contract's content is programmed as a smart contract by an **external party** and stored on Ethereum. Costs = one-time costs for 1:n contracting parties for smart contracts	- Costs for verifying the occurrence and compliance of contractual objects by **smart contract** - Overall verification costs = total gas used * gas price (gas price is fluctuating) Costs < recurring costs for 1:n contracting parties for conventual contracts

* Costs for contract creation, contract verification, and execution of contract, *assuming contract compliance*

Figure 4. Overview of Costs in a Smart Contract Arrangement

contractual relationships.

In case a contract is *recurring* and concluded between *two contracting parties*, the onetime costs are equal to the onetime costs that arise when two parties are entering a smart contract that is non-recurring. This is because all contingencies and cases need to be agreed, whether the contracts are recurring or non-recurring. Similar to the previous cases, lawyers with special training in smart contract validation are required. The recurring costs in a 1:1 recurring smart contract setting are higher than in a onetime smart contract setting, because gas fees are due throughout the term of the contract.

In the last scenario in which a contract is *recurring* and concluded between *more than two contracting parties* (1:n), the onetime costs for setting up a smart contract are equal to the costs that arise when a 1:n contract is set up for a non-recurring period. Due to the recurring contract period, the recurring costs of a 1:n smart contract relationship are higher than in case of a 1:n non-recurring smart contract relationship (see figure 3).

5 Summary of Results and Discussion

Our analysis shows that the total one-time cost of setting up smart contracts is much higher than the cost of setting up conventional contracts (experts 1 - 12). This is because in addition to the cost of agreeing the terms of the contract, there are legal fees and programming costs to translate the content of the contract into smart contracts. The total recurring cost of smart contracts depends on the complexity and duration of the contract. In the case of a 1:n contractual relationship, the recurring cost of smart contracts to automatically verify and execute contract elements is less than the sum of the costs incurred when all parties manually verify and execute contracts. Depending on the complexity of the contract between two contracting parties, the recurring costs are lower in a traditional contracting environment. This is because if the contract is entered into once, the one-time review of the contract terms is low and easily verifiable. However, if the contract terms between two contractors are highly complex, for example, involving various external databases, it may be more advantageous to implement smart contracts. In general, our analysis has shown that the more complex a contract (determined by the contract objects and the parties involved), the higher the one-time costs. In these cases, the use of smart contracts is preferable. Despite the high setup costs, smart contracts offer the advantage that checking and executing rules is less expensive than doing it manually. The less complex the contracts are, and only from a cost perspective, the traditional contract can still be used. However, if security is the primary concern and transaction costs are secondary, smart contracts should be used. This is also in line with the original idea and goal of all blockchain applications: the primary decision for blockchain and smart contract solutions should not be based on cost aspects, but on security aspects.

6 Conclusion and Contribution to Theory and Practice

Although there have been studies on the cost structure of smart contracts, such as the composition of gas fees (Vatiero, 2018) and energy consumption to run smart contracts (Zhang and Chan, 2020), we are the first to directly compare the costs associated with setting up traditional and smart contracts. This comparison provides the basis for further research around the computing protocols analyzed. Based on our findings, use cases such as Walmart can be quantitatively analyzed once the relative savings generated by using smart contracts are published. The matrix we developed provides a starting point for analyzing other key factors that influence a decision for or against smart contracts. Kannengieser et al. (2022) identify technical challenges associated with smart contracts. They provide initial solutions to the identified challenges. Based on this, our matrix offers another important step to sensitize organizations for the use of smart contracts. In addition, we create a first relevant tool for practitioners and academics to determine whether to choose conventional or smart contracts. In addition to the existing 10-step decision path for blockchain applications (Pedersen et al., 2019), we introduce a criteria catalog through our morphological box. This catalog aims to determine when it is appropriate to use a smart contract, and when a conventional contract should be preferred.

Overall, the paper provides a basis for evaluating the use of smart contracts in terms of effectiveness. For practitioners, our results help them to better assess when the use of smart contracts in their organization is worthwhile or not. We provide them with concrete evaluation dimensions. We also contribute to theory, as our work is the first to provide a basis for evaluating smart contracts in terms of effectiveness. So far, and since smart contracts are a rather new phenomenon, there has been quite a lot of research on this topic (Albizri and Appelbaum, 2021; Bai et al., 2018; Balcerzak et al., 2022; Carvalho and Karimi, 2021; Chen et al., 2018; Christidis and Devetsikiotis, 2016; Cuccuru, 2017; Egelund-Müller et al., 2017; Halaburda, 2020; Huang et al., 2021; Kannengieser et al., 2022; Khan et al., 2021; Shahab and Allam, 2020; Shailak Jani, 2020; Simić et al., 2021; Szabo, 1996; Taherdoost, 2023; Triana Casallas et al., 2020; Vatiero, 2018; Wan et al., 2021). The various application scenarios of smart contracts have been extensively analyzed and presented in the literature. It is common for new digital innovations to be surrounded by extensive research, and smart contracts are no exception. However, there is a lack of evaluation regarding the cost structures of smart contracts compared to conventional contracts. Our research addresses this gap by comparing the costs associated with deploying smart contracts versus traditional contracts. In doing so, we provide a foundation for further research to evaluate smart contracts across different aspects and on blockchains other than Ethereum, such as Hyperledger Fabric. These aspects may include employee adoption, organizational knowledge required for successful implementation, and exploration of additional applications and use cases for smart contracts.

References

Albizri A and Appelbaum D (2021) Trust but Verify: The Oracle Paradox of Blockchain Smart Contracts. *Journal of Information Systems* 35(2): 1–16.

Bai X, Cheng Z, Duan Z, et al. (2018) Formal Modeling and Verification of Smart Contracts. In: *Proceedings of the 2018 7th International Conference on Software and Computer Applications:* Kuantan Malaysia, 08 02 2018 10 02 2018, pp. 322–326. New York, NY, USA: ACM.

Balcerzak AP, Nica E, Rogalska E, et al. (2022) Blockchain Technology and Smart Contracts in Decentralized Governance Systems. *Administrative Sciences* 12(3): 96.

Carvalho A and Karimi M (2021) Aligning the interests of newsvendors and forecasters through blockchain-based smart contracts and proper scoring rules. *Decision Support Systems* 151: 113626.

Chen T, Li Z, Zhou H, et al. (2018) Towards saving money in using smart contracts. In: *Proceedings of the 40th International Conference on Software Engineering: New Ideas and Emerging Results: (*eds A Zisman and S Apel), Gothenburg Sweden, 27 05 2018 03 06 2018, pp. 81–84. New York, NY, USA: ACM.

Chong AYL, Lim ETK, Hua X, et al. (2019) Business on Chain: A Comparative Case Study of Five Blockchain-Inspired Business Models. *Journal of the Association for Information Systems*: 1308–1337.

Christidis K and Devetsikiotis M (2016) Blockchains and Smart Contracts for the Internet of Things. *IEEE Access* 4: 2292–2303.

Cuccuru P (2017) Beyond bitcoin: an early overview on smart contracts. *International Journal of Law and Information Technology* 25(3): 179–195.

Egelund-Müller B, Elsman M, Henglein F, et al. (2017) Automated Execution of Financial Contracts on Blockchains. *Business & Information Systems Engineering* 59(6): 457–467.

Eggers J, Hein A, Weking J, et al. (2021) Process Automation on the Blockchain: An Exploratory Case Study on Smart Contracts. In: *Proceedings of the 54th Hawaii International Conference on System Sciences: (*ed T Bui). Hawaii International Conference on System Sciences.

Halaburda H (2020) *Understanding Smart Contracts as a New Option in Transaction Cost Economics*. S.l.: SSRN.

Huang Y, Zhang P and Jin Z (2021) Improving business operation efficiency by using Smart Contract. In: *2021 2nd Asia Service Sciences and Software Engineering Conference:* Macau Macao, 24 02 2021 26 02 2021, pp. 34–40. New York, NY, USA: ACM.

Kannengieser N, Lins S, Sander C, et al. (2022) Challenges and Common Solutions in Smart Contract Development. *IEEE Transactions on Software Engineering* 48(11): 4291–4318.

Khan SN, Loukil F, Ghedira-Guegan C, et al. (2021) Blockchain smart contracts: Applications, challenges, and future trends. *Peer-to-peer networking and applications* 14(5): 2901–2925.

Lacity MC and van Hoek R (2021) How Walmart Canada Used Blockchain Technology to Reimagine Freight Invoice Processing. *MIS quarterly executive*: 219–233.

Li C (2021) Gas Estimation and Optimization for Smart Contracts on Ethereum. In: *2021 36th IEEE/ACM International Conference on Automated Software Engineering (ASE):* Melbourne, Australia, 15 - 19 November 2021, pp. 1082–1086. IEEE.

Li M, Chen Y, Lal C, et al. (2023) Nereus: Anonymous and Secure Ride-Hailing Service Based on Private Smart Contracts. *IEEE Transactions on Dependable and Secure Computing* 20(4): 2849–2866.

Liu H (2020) A Novel Supply Chain Model Based on Smart Contract. In: *Proceedings of the 2020 2nd International Electronics Communication Conference:* Singapore Singapore, 08 07 2020 10 07 2020, pp. 115–120. New York, NY, USA: ACM.

Mohanta BK, Panda SS and Jena D (2018) An Overview of Smart Contract and Use Cases in Blockchain Technology. In: *2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT):* Bangalore, 10 - 12 July 2018, pp. 1–4. IEEE.

Pedersen AB, Risius M and Beck R (2019) A Ten-Step Decision Path to Determine When to Use Blockchain Technologies. *MIS quarterly executive*: 99–115.

Richard R, Prabowo H, Trisetyarso A, et al. (2020) Smart Contract Development Model and the Future of Blockchain Technology. In: *2020 the 3rd International Conference on Blockchain Technology and Applications:* Xi'an China, 14 12 2020 16 12 2020, pp. 34–39. New York, NY, USA: ACM.

Shahab S and Allam Z (2020) Reducing transaction costs of tradable permit schemes using Blockchain smart contracts. *Growth and Change* 51(1): 302–308.

Shailak Jani (2020) Smart Contracts: Building Blocks for Digital Transformation, Unpublished.

Shinde D, Padekar S, Raut S, et al. (2019) Land Registry Using Blockchain - A Survey of existing systems and proposing a feasible solution. In: *2019 5th International Conference On Computing, Communication, Control And Automation (ICCUBEA):* Pune, India, 19 - 21 September 2019, pp. 1–6. IEEE.

Simić S, Marković M and Gostojić S (2021) Smart Contract and Blockchain Based Contract Management System. In: *7th Conference on the Engineering of Computer Based Systems:* Novi Sad Serbia, 26 05 2021 27 05 2021, pp. 1–8. New York, NY, USA: ACM.

Szabo N (1996) Smart Contracts: Building Blocks for Digital Markets. *EXTROPY: The Journal of Transhumanist Thought* 18(2).

Taherdoost H (2023) Smart Contracts in Blockchain Technology: A Critical Review. *Information* 14(2): 117.

Triana Casallas JA, Cueva-Lovelle JM and Rodríguez Molano JI (2020) Smart Contracts with Blockchain in the Public Sector. *International Journal of Interactive Multimedia and Artificial Intelligence* 6(3): 63.

Vatiero M (2018) Smart Contracts and Transaction Costs. *SSRN Electronic Journal*. DOI: 10.2139/ssrn.3259958.

vom Brocke J, Simons A, Riemer K, et al. (2015) Standing on the Shoulders of Giants: Challenges and Recommendations of Literature Search in Information Systems Research. *Communications of the Association for Information Systems* 37.

Wan Z, Xia X, Lo D, et al. (2021) Smart Contract Security: A Practitioners' Perspective. In: *2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE):* Madrid, ES, 22 - 30 May 2021, pp. 1410–1422. IEEE.

Wang X and Xu F (2022) The Value of Smart Contract in Trade Finance. *Manufacturing & Service Operations Management*. DOI: 10.1287/msom.2022.1126.

Zhang R and Chan WK (2020) Evaluation of Energy Consumption in Block-Chains with Proof of Work and Proof of Stake. *Journal of Physics: Conference Series* 1584(1): 12023.

Zheng Z, Xie S, Dai H-N, et al. (2020) An overview on smart contracts: Challenges, advances and platforms. *Future Generation Computer Systems* 105: 475–491.