Smart contracts may become a risk for computer scientists

Thomas Hofmann; Tallinn Law School, Tallinn University of Technology; thomas.hoffmann@taltech.ee
Volker Skwarek; FTZ "Digitale Wirtschaftsprozesse", Arbeitsgruppe DLT³-Hamburg, Hochschule für Angewandte Wissenschaften Hamburg; volker.skwarek@haw-hamburg.de

Abstract: Smart contracts not only provide automation opportunities in blockchains, but can also depict legally binding contracts. This article discusses different facets of the application of smart contracts and the possible or meaningful scope of the legal will as well as the legal consequences of fulfilment and performance disruptions on the Blockchain. Also, a brief overview of the Legal Services Act shows where legislators limit software developers in the creation of smart contracts and how they can protect themselves from liability.

Smart Contracts and Automation of Law

Modern Blockchain and Distributed Ledger technologies (BC / DLT) are characterised by freely programmable automation mechanisms in the form of so-called Smart Contracts; according to Swan (2015, Chapter 2) [1, Ch. 2], they form "second-generation blockchains". Although the classification by Swan is arguably in comparison with other taxonomies [2], [3, p. 9], it is undisputed that Smart Contracts have become a wide-spread tool today [4]–[6], which significantly defines the application and acceptance areas of BC / DLT.

The term "contract" as an eponymous element may initially be misleading since smart contracts are primarily programs using programming languages that are based on current languages such as JavaScript or Python. A legal relationship in the sense of a contract was first taken into by Szabo Nick Szabo,[7] who described in his publication "The idea of Smart Contracts" [7] the possibility to automate contractual clauses in terms of legal transactions in order to avoid misinterpretations or execution delays. For the depiction of complete contracts or even the representation of entire legal systems, Szabo even in subsequent publications did not provide any no evidence. Even Buterin - the author of Ethereum – stated only that

"Another important area of inquiry is "smart contracts" - systems which automatically move assets according to arbitrary pre-specified rules. [...] The logical extension of this is the decentralised autonomous organisations (DAOs) - long-term smart contracts that contain the assets and encode the bylaws of an entire organisation."[8],

i.e. the automation of rules and organisation. Also, Buterin commented in an entry on social media pages to the somewhat unfortunate naming of these distributed programs:

"To be clear, at this point I'll regret adopting the term smart contracts. I should have something more boring and technical, perhaps something like persistent scripts." [9]

Nevertheless, there are efforts to use Smart Contracts also for documenting contract conclusions and their (partial) automation on a blockchain. For example, within the scope of international standardisation, both ISO TC 307 are concerned [10], just as well as a focus group of the ITU [11], with the legal consideration of smart contracts.

In the prevailing legal opinion, it is indisputable that documentation of the contract on a blockchain is legally unproblematic [12]. More controversial, however, is the automatic performance of contractual obligations: Is.e.g a court itself, through reviewers, capable of interpreting algorithmic correctness in terms of a party will? In case of doubt, can this transaction be reversed if it violates a legal prohibition (§134 BGB) and is therefore void? How do courts deal with the question of the fundamentally impossible repayability of transactions on blockchains without the participation of the opposing party?

This raises many unanswered questions in which the compatibility of BC / DLT with its decentralisation and anonymity or cryptographic principles of operation with existing legal principles is unclear, as the place of execution of a smart contract is unpredictable and immediate termination or reversal is not possible. No central instance can enforce the law with the help of the state monopoly on power. Further issues arise when the financing of business start-ups via ICOs (initial coin offerings) and autonomous organisations (distributed autonomous organisations, DAO) affects different areas of law and jurisdictions. For example, where and how are the profits taxed, which labour law applies to employment contracts, or which warranty right in case of disruption, and is even a prospectus obligation, because the impression of an investment or company participation is awakened.
After a brief introduction to basic dogmatic backgrounds of contracts, the article also deals with the challenge of translating legal content into software code and possibly making mistakes by its interpretation. Often also regulations on the provision of legal services are concerned, which many software developers are not aware of due to a lack of legal training, in Germany for example by the Legal Services Act (RDG). Programmers run the risk of violating this law if, in addition to pure programming, they conduct legal audits or - even if not audited - accept orders where a legal analysis would actually be required.

### Legal contract versus smart contracts

From a legal point of view, a smart contract is initially not necessarily a contract. [13] In most legal systems, a contract is composed of two generally form-free - eventually even only implicit - declarations of intent, namely offer and acceptance. The offer must contain sufficient information about essentialia negotii and be accepted without modification. In the case of purchase contracts, this would include, for example, information about contractual partners and objects of the contract (price and purchase item).

In contrast to this, a smart contract is merely a software-based protocol, which, if a previously defined condition occurs, results in other agreed protocol steps. So, formally, it initially only represents a documentation of the contract. [14]

This applies even more for "non-smart", other "closed" contracts concluded over the Blockchain - which is finally, as already explained, only a protocol on the alleged exchange of two declarations of intent, which may have a certain amount of evidence. However, this is not decisive in the presence of further stronger evidence against the conclusion of a contract - for example, that one of the parties did not even initiate such a protocol. The protocol part of a smart contract is therefore only one indication among many that a judge will consider when deciding on the formation of a contract.

The second, more characteristic part of a smart contract - the automatic start of another process - can have a legal meaning. This is particularly the case if the purpose of the transaction is to perform obligations arising out of the recorded contract without further action by the parties as soon as a specific condition arises, such as the performance of the contractual obligation of one of the parties. A classic example is the execution of an action on receipt of payment. From a legal point of view, however, this is not a new concept; vending machines were already known in antiquity, such as the issue of purified water against a coin in ancient Greece.

However, in the classical legal understanding, it is unusual that documentation or a document in the form of a Smart Contract triggers an independent action. Compared to the coin entry just mentioned, this would mean that courier sends the coin, then the coin is thrown into the vending machine and then - perhaps - at some time in another place in the world the desired product is ejected from the machine. In other words, in the classical understanding of the law, actions are to be assigned to "persons" in the legal sense, who have the power over the action. Under German law, such persons are either people (natural persons) or a clearly defined number of legal entities (generally companies). Specht / Herold, for example, consider that at least an analogous application of the rules on agency for the consequences of contracts concluded by autonomous systems can be discussed.[15] However, the main question remains who is accountable for actions triggered via a smart contract: The owner whose property is deposited via his cryptographic owner-address? The transaction itself, which may trigger the smart contract from outside as Oracle? The miner who runs the process on the blockchain? Against this background, a smart contract is more than just documentation, and its decentralisation and asynchronicity of creation and execution can raise significant questions of responsibility.

### Legally relevant smart contracts in the system of contract law

#### A conclusion of a contract

The examination of a contractual claim consists of three steps:

1) First, the conclusion of the contract is examined, which gives rise to contractual rights and obligations in case of a positive examination result. The existence of these can, as stated if required also be protocolled via software-based systems, but this has an only declaratory effect.

2) Subsequently, it is examined whether these contractual obligations have been expired - as a rule, the performance of those duties effects this. The elements of this performance depend entirely on the content of the specific contract. In the case of purchase contracts, for example, these claims are generally expired by the performance itself if the corresponding dispositions - the payment of the purchase price, on the one hand, the transfer of ownership of the purchased item on the other hand - have been made.
Even smart contracts can set such dispositions in motion and affect the corresponding legal consequence - the performance of the respective contractual claim. If, for example, a smart contract is programmed in such a way that the system transfers ownership of a moveable item as soon as the system registers the receipt of payment by bank transfer, the buyer's right to performance expires. In the German legal system, it is necessary to be aware of a very abstract distinction between the claim for transfer of ownership that resulted from the conclusion of the purchase contract and the actual transfer, which constitutes a further legal transactions. The obligations under the purchase contract itself are performed by the payment of the price and the transfer of ownership.

However, the transactions based on a smart contract do not have to be exclusively dispositions in the legal sense. A disposition changes the legal distribution of a thing to a subject, that is, one not only has the right to something but for something. If, for example, the Smart Contract causes the possession of a thing to pass - the typical example is that a hotel door opens after an Oracle announces the payment of the rental price to the Smart Contract - the legal distribution of the right in the room (i.e. property) will not be affected because the Landlord continues to own property. Nevertheless, the claim to perform the duties of the previously concluded hotel room rental agreement has expired, as the landlord has fulfilled his contractual obligation, namely has provided possession in the room for one night according to the agreement.

The expiration of the contractual claim by performance is the rule. However, there are some other, much more complicated causes within the catalogue of remedies in the law of non-performance which make a claim expire. This may be the actual or legal impossibility of performance, a successfully declared rescission of one party due to a material breach of the other party's contractual obligation, offset, or another accepted performance instead of fulfillment.

For a truly autonomous contract execution, smart contracts would need to map these pathological cases of non-performance as well accurately. Otherwise, actions by natural persons claiming rights arising out of breach of these contractual obligations would still be required. For a smart contract, for example, this would mean depicting the much more complex declarations of intent in non-performance. If, for instance, a rented wooden hut is burned down, the duty to perform a rental contract for that expires due to impossibility, even if the obligation was recorded in a smart contract in such way that access to the cottage is granted on receipt of payment. If the hut is to be rented in winter, but cannot be heated for technical reasons, the expiration of the right to provide access to the hut occurs only if the tenant declared a corresponding rescission - and if a corresponding rescission was also admissible. This declaration of rescission would then theoretically have to be modelled in the Smart Contract in such a way the corresponding declaration of rescission was already made upon conclusion of the contract on the condition that a material breach of contract occurs. The remaining conditions of rescission are still present, such as the unsuccessful expiry of a reasonable period for supplementary performance.

In conclusion, the non-autonomous, non-local, anonymous execution of contractual obligations, which is independent of the actions of natural persons, involves legal risks which, in the classical legal practice, would be prevented by acting persons or public authorities. In a fully automatic smart contract, however, all these decision trees would have to be mapped into its software design.

3) The final step in legal analysis is the question of the enforceability of the claim, that is, whether the claim for material and non-defunct claims would be enforceable by courts in practice. In principle, the enforceability can be barred by some exceptions, such as by the limitations of the claim or the non-performance of the consideration.

As a matter of principle, claims are enforced - if the debtor does not act voluntarily - upon obtaining a corresponding title and on the basis thereof by state bodies. In the context of smart contracts, one should thus not speak of “self-enforcing contracts”, since a contract concluded between private individuals and recorded in the form of a smart contract cannot impose state sovereignty-based enforcement measures. It is possible, however, to embed “compulsory means”, which ensure the performance of the contractual obligations of the counterparty. A classic example is the purchase of a PC on instalments, where specific functions are via Smart Contract automatically blocked when the payment of instalments is delayed. However, still, also this measure has long time been standard in long-term supply contracts (e.g. water and electricity supply). Again, in the context of complete automation without human interaction and review of items 2) and 3) above, caution should still be exercised, as delaying instalment payment by the customer may be lawful, while automatic vendor blocking mechanisms may be not.

Legally, such a blocking measure is not enforcement, but only the exercise of a - possibly previously contractually agreed - security tool, which is novel only to the extent that goods were so far after delivery of the thing outside the sphere of influence of the seller.
Unstoppable Contracts - Impermanence and Finality

One of the key features of smart contracts that run on blockchain mechanisms is the unstoppability of once-programmed events after a specific condition has been triggered.

Legally, this is just a technical peculiarity of a particular contract – for example comparable to a chewing gum vending machine, which cannot be stopped either after a coin has been inserted from eject the chewing gum, even if the vending machine operator and the customer agree after a coin has been inserted that the transaction is ineffective or should be reversed. In this case, the transaction would be reversed after ejection of the chewing gum between the buyer and the vending machine operators by concluding a second contract of that very same content and returning ownership to the seller.

The same applies to smart contracts, and especially to the second feature, "finality". The ejection of chewing gum, in its legal effect - the transfer of ownership of chewing gum from the transferor to the transferee - contains at this moment even a much stronger "finality" than a smart contract initiated transaction. Notwithstanding any measures taken later - the acquirer became an owner at that moment, and the initiation of a hard-fork-comparable measure is not available to the physical world. Of course, this property may later be returned if the parties so request, or if one of the parties successfully sued the other for the submission of a transfer of ownership and ownership. However, this also applies to all Smart Contract-initiated operations.

Also, the argument that specific assignments are legally ex tunc, that is, from the point of origin, ineffective because a non-ratified and thus ineffective contract was concluded by a minor, does not apply. The same is the case with orders initiated via smart contract: If a minor sells a thing over Smart Contracts and the Smart Contract makes the order, the property remains with the minor who may later request the release of the item, even if the contract recorded in the Smart Contract was active.

Obviously, most legal relationships in which the use of smart contracts makes sense are more complex than chewing gum automat. Moreover, indeed there are legal grey areas, especially about liability for the execution of a smart contract. Those may lead to effects no longer intended by all parties: If for example in the logistics industry the global delivery of a particular raw material was executed, delivered, and processed via a smart contract upon receipt of a particular payment, serious questions arise when a community decision results in a fork. In other words, to correct erroneous transactions, the blockchain is reconstructed to the point of time when neither the supplier nor the recipient of the already processed raw material possessed the money. The question as to who can claim which claims for which declaration of intent against whom and who was responsible for which non-performance can hardly be answered even with free access to all transaction data with the traditional non-performance law.

Also, the asynchronicity of transactions can lead to severe problems: After all, a contract should not be performed "sometime", but immediately after the occurrence of certain conditions. However, if a closed contract as a smart contract was only processed with a considerable delay due to a capacity bottleneck – for example, Easter eggs are delivered only after Easter - the limits of the blockchain and the fully automatic or even autonomous processing become apparent.

The Provision of Smart Contracts by Computer Scientists as Illegal Legal Services?

Smart contracts created on behalf of customers not only have a legal dimension conceptually, but they may also be the result of an activity that is regulated as a legal service and, as such, may not be offered by anyone. Programmers may be held liable in Germany if their performance includes at least in part legal advice. This applies not only to programmers, but also to any other IT-expert who touches legal issues while providing his or her services (fundamentally Fries [16] ) by LegalTech (on the market development [17] ), be it in the form of law enforcement (for details see Römermann / Günther [18] ), but also specifically in the form of automated contract design (Wettlaufer [19] ).

Even though the relevant regulations do not yet refer to smart contracts, and there is a lack of established case-law for programming smart contracts as a legal service, the criteria of the Legal Services Act still provide a clear picture of the extent to which smart contract programming is permitted and from when the programmer may be held liable.

Since 2008, legal services are regulated by the Legal Services Act (RDG). This replaced the Legal Advice Act and breaks with the principle of the lawyer’s monopoly. Therefore, legal services can since then be offered in principle also by non-state-examined lawyers, if either no "legal examination of a specific case required" (§ 2 para 1 RDG) or the provider the service provider is qualified for this activity for other reasons.

For computer scientists, this means first of all that their activity can also fall within the scope of the RDG if it has legal components, and that in this case they also may face respective consequences/sanctions due to the violation of the RDG. These range from claims for damages by the customer due to a contract that is void on
account of this breach of the RDG over competition law consequences, to fines due solely to this breach, which may be sanctioned as well as an administrative offence.

However, the programming of which smart contracts is now concretely addressed by the RDG as inadmissible? In principle, the RDG is also applicable to smart contracts, namely about "human service providers" and "activity" (§§ 10 I, 12 IV, 21 RDG). According to § 2, this activity must be required in the "examination" of a "concrete", actually existing and not a merely hypothetical individual case. This definition makes all those activities license-free, in which Smart Contracts are provided as a "sample", for example against payment for download since no legal analysis takes place, and such an analysis is due to the lack of a concrete individual case also not possible/necessary. In this context, it should be noted that, for example, the sale of contract form samples is covered by the RDG neither, which in this respect already have some parallels with Smart Contracts. In both cases, a general sample is written and then sold, albeit for smart contracts in the form of code rather than in human language, into which the customer then fills in data himself. Even the addition of a "completion assistance manual" for such forms is not yet a legal service (OLG Karlsruhe, NJW-RR 2011, 119, 120).

It becomes more controversial when a computer scientist creates a tailor-made smart contract for a particular customer, providing for a concrete individual case. However, here, too, a legal examination must be required, which must be done by the computer scientist. It is not sufficient for the customer to supply certain contract specifications and to have these translated into a code. For a "necessary legal examination" within the meaning of the RDG it requires a legal. Subsumtionsvorgang, that is, the comparison and the evaluation of concrete facts with a particular legal regulation, which is desired by the customer at least as part of the service. In the context of this examination, there is no requirement for any legal evaluation - even a purely logical deduction under the wording of the law is already seen as a legal examination/analysis. According to the Federal Supreme Court, § 2 (1) RDG "covers every concrete subsumption of a fact under the relevant legal provisions, which goes beyond a merely schematic application of legal norms without further legal examination. Whether it is a simple or difficult legal question is irrelevant "(BGH AnwBl 2016, 605). According to the protocol of the legislator's negotiations in the German Parliament 16/3655, p. 47, this only falls within the prohibition area of the RDG if the "client expects special legal assistance or education".

For the operation of this specific case - namely the customer, who has neither yet a clear legal framework nor an IT-based contract conclusion and fulfilment mechanism in order to solve a specific case, the RDG is indeed a considerable hurdle for programmers. However, any software developer who does not have to look into the BGB (the German Civil Code) or the ZPO (the German Civil Procedure Act) to design a smart contract designed for a specific customer should always be on the safe side. Especially, if he is aware of the fact that he is programming Smart Contracts performing themselves no legal examination, and that there is also no need for such analysis due to the lack of a concrete individual case, and communicated this in the form of an explicit and detailed disclaimer to the client. In order to prevent such a practice, there is no need for a broad interpretation of the RDG's permission reservation [19] or even a reform of the RDG - it is in this case merely no legal service. [20]

Conclusion

As a matter of principle, smart contracts do not open any new doors in the realm of law, primarily if they merely serve to document the conclusion of a contract or the uncomplicated automation of the performance of contractual obligations. Challenges arise, however, in non-performance law, as contract parameters are much more influenced by general legal concepts and subject to (human) valuations. Furthermore, system-inherent aspects such as decentralisation, asynchronicity, indelibility, irrevocability, or immutability can (but do not have to) lead to problems that are difficult to predict in their concrete legal consequences and are thus currently very difficult to handle, especially in highly automated processes with many subsequent actions.

Finally, the Legal Services Act can be a hurdle for computer scientists if they misdescribe their services and enrich them with a - often even not intended - legal value. Software publishers are here at risk if not only a simple programming service is provided, but additional performances are promised or performed, for which a legal examination of a specific case is required.

References


