Bird & Bird & DLT: Can Code Be Law?

by Dr. Michael Jünemann and Dr. Udo Milkau.

July 27th, 2021

* The authors thank *Timo Förster* and *Julia Fröhder* for their support.

1. Introduction

The so-called "blockchain technology" (aka distributed ledger technology) is discussed currently as some kind of philosophers stone, to solve nearly all questions which have not been answered by traditional information technology. In addition, blockchain should change the whole world of economy as said in the subtitle of the book "Blockchain Revolution" of Tapscott[1]: *How the Technology Behind Bitcoin Is Changing Money, Business, and the World*. Especially, the extension of the (transactional) blockchain with so-called "smart contracts" (i.e. computer programs or "scripts" to be executed in a runtime environment of the blockchain) generated much interest as, in a summary of the authors derived from a variety of references, "self-executable" and "self-enforcing" code with a "truth outside the authority of court".

Nearly two decades ago, L. Lessig[2] wrote about a rather "dark" vision of a future, in which "code" will be a threat to liberty and in which "code is law":

"Every age has its potential regulator, its threat to liberty. [...] Ours is the age of cyberspace. It, too, has a regulator. This regulator, too, threatens liberty. But so obsessed are we with the idea that liberty means "freedom from government" that we don't even see the regulation in this new space. We therefore don't see the threat to liberty that this regulation presents. This regulator is code - the software and hardware that make cyberspace as it is. This code, or architecture, sets the terms on which life in cyberspace is experienced."

Recently Paech[3] articulated a controversial opinion and discussed a new sequence of authority in the context of "smart contracts":

"However, court decisions do not exert the same authority as in the traditional context of financial market transactions. [...], the hands of the court are tied to a large extent. First, the parties, using their contractual freedom, are likely to have agreed to the application of the internal rules [of a technical blockchain system] to their dealings, superseding the relevant private law rules. However, should the court hold that private law takes precedence [...] it will still be unable to order a rectification of the blockchain, as the blockchain cannot be changed subsequently, [...]

By adhering to the network, they have, implicitly or explicitly, agreed to operate in a technical, trustless environment, which only relies on maths and cryptography[4], and accepted that its internal rules may lead to outcomes different from those governed by private law rule.

*...court should decide whether the insolvent has acquired or lost the asset on the basis of private law, or should apply the internal rules of the blockchain network as an expression of party autonomy, or as a form of **lex mercatoria**?"*

Last but not least, Ortolani[5] argues rather focussed:

"... that Bitcoin must be regarded as an original and self-contained system of dispute resolution, whose characteristics can be used to theorise new models of self-enforcement."

The main difference between "code is law", Medieval sea laws and merchant laws (*lex mercatoria* – both transnational laws[6]) is the antagonism of *ex-ante* "hard coded" rules with a clockwork-like implementation versus a system of special custom and best practice, which could be enforced *ex-post* through dedicated merchant courts or arbitration. Furthermore, *leges mercatoriae* are mostly effective only inter partes while the concept of "code is law" appears to claim effectiveness inter omnes.

[1] Tapscott D, Tapscott A. *Blockchain Revolution: How the Technology Behind Bitcoin Is Changing Money, Business and the World* London: Portfolio, 2016.

[2] Lessig L, *Code is Law. On Liberty in Cyberspace.* Harvard Magazine, https://harvardmagazine.com/2000/01/code-is-law-html (accessed 16 January 2020).

[3] Paech P. The Governance of Blockchain Financial Networks. *Modern Law Review* 2017; 1073: 1098.

[4] Remark: The key component of the original Bitcoin blockchain is a game theory "Game Theory" approach (that rational players can benefit more than by cheating), but not cryptography as the applied cryptographic methods can be taken from any standard code library.

[5] Ortolani P. Self Enforcing Online Dispute Resolution: Lessons from Bitcoin. *Oxford Journal of Legal Studies* 2016;595:595.

[6] see e.g. Berger K P. Center for Transnational Law "CENTRAL", University of Cologne, Germany; www.trans-lex.org (accessed 16 January 2020).

In this paper, we will challenge the idea that "code is law" and that "smart contracts" on the blockchain can form a new legal system with pre-defined self-executing self-enforcement for contractual relationship in the age of digitalization.

The discussion will start with a comparison of smart contracts on the blockchain and traditional computer code via juxtaposition between ideal worlds of clockwork-like computing with the reality of an unknown future, and governance processes to end with an outlook on the social phenomenon of the wish for a "better" world without any uncertainty.

2. Smart Contracts on the blockchain and traditional computer code

Recently, Raskin[7] discussed the legality of smart contacts and defined smart contracts as:
"… agreements wherein execution is automated, usually by computers. Such contracts are designed to ensure performance without recourse to the courts. Automation ensures performance, for better or worse, by excising human discretion from contract execution."

However, there are three different levels of implementation of smart contracts:

- **Normative definition** - what a smart contract should be (independent from any technology: currently defined by some authors as "smart legal contracts"[8])
- **Positive perspective** - how smart contracts can be implemented on computer systems (including blockchain-based distributed ledger technology)
- **Actual situation** - how blockchain technology has been developing in the reality

The term "smart contracts" was probably coined by Szabo[9] with a definition in line with the one mentioned by Raskin[10]. The definition raises the question, how such an "automation" and "performance without recourse to the courts" can be implemented practically in the real world, in which any commercial relationship has a social context and contracts are embedded in a matrix of the world.

Already before "smart contracts", computer scientists developed "Intelligent Software Agents"[11] for application in electronic commerce with features to negotiate and confirm contractual relations autonomously. In addition, legal issues were discussed[12], but the complexity of mobile software agents lead to technical problems, and no actual application of agent technology for legal contracting was developed.

With the development of the blockchain (aka distributed ledger technology) starting with the first use case of Bitcoin, smart contracts have been revitalised based on the underlying technology of blockchain as a synchronised protocol for transactions based on a distributed database system. Therefore, smart contracts - or smart contract code - are computer code on top of a blockchain runtime environment on distributed computer systems. Smart contracts are computer code executed on-top of computer code running on computer code. Therefore, any discussion about smart contracts is strongly dependent on the definition and the technical environment they are implemented on. A discussion about benefits of smart contracts can only be done in comparison to "usual" computer software applications used for the same task.

Although "automation" and "self-execution" should be important features of smart contracts, traditional banking systems are already highly automated and typically process transactions without any human intervention. Examples are *inter alia* the nearly 100 percent straight-through processing for electronic SEPA payment transactions, automated initiation of credit transfers by a standing order (sic!), processing of interest payments for a bond at the contractual date, or calculation of variation margin or collateral from derivative contracts triggered by a pre-defined condition.

The original development of blockchain with the first use case of Bitcoin did not target automation but was focussed on the issue of "electronic cash" in an open peer-to-peer computer network without any (central) intermediaries. As it is not the scope of this paper to make a deep dive into distributed computer technology, the reader is referred to publications such as Milkau et al. (2016) for a discussion of details.[13] In a nutshell, the challenge results from the theoretical impossibility to synchronise an "open" network with a technical protocol (Fischer, Lynch, Paterson theorem on "Impossibility of distributed consensus with one faulty process" from 1985) and, in consequence, from (i) the Double Spending Problem and (ii) the Byzantine General Problem to agree on the right sequence of "valid" transactions.

The Bitcoin blockchain "solved" this "impossible" problem with a bypass to technology and the concept of game theory to reach distributed consensus, but (a) with assumption and (b) under limitations. The practical solutions came with a price to be paid. Bitcoin (and the underlying blockchain) is a closed system with a repeated game between rational players, i.e. it is comparable to a game of poker in a casino with agreed rules and with a proprietary "currency" in form of tokens. For such a proprietary game, of course, no courts are needed, as self-enforcement is achieved with *ex-ante* fixed rules by agreement of all "rational" players, who risk their stake and hope to win the jackpot. The "price" to be paid for this knack on the technical side is inefficiency, slowness, and limited capacity; and on the methodical side, it is the problem of (only) eventual consistency: Bitcoin has no commercial or technical finality of transactions, but a probabilistic approach to technical synchronisation.

Since the start of Bitcoin, there have been a lot of developments, which changed the balance of parametrisation of Bitcoin to

[7] Raskin M. The Law and Legality of Smart Contracts. 1 Georgetown Law Technology Review 2017;305:306.
[8] see e.g. EBRD and Clifford Chance *Smart Contracts: Legal Framework and Proposed Guidelines for Lawmakers*. https://www.ebrd.com/documents/legal-reform/pdf-smart-contracts-legal-framework-and-proposed-guidelines-for-lawmakers.pdf (accessed 16 January 2020).
[9] Szabo N. Formalizing and Securing Relationships on Public Networks. *FirstMonday* 1997 Vol. 2/9; Szabo N. Smart Contracts: Building Blocks for Digital Markets. *Extropy: Journal of Transhumanist Thought* 1996;18:18.
[10] see above.
[11] Brenner W, Zarnekow R, Wittig H. *Intelligente Softwareagenten*. Berlin: Springer, 1998; Guttman R H, Moukas A, Maes P. Agent mediated Electronic Commerce: A Survey. *The Knowledge Engineering Review* 1998;147:147.
[12] Wettig S, Zehender E. A legal analysis of human and electronic agents. *Artificial Intelligence and Law* 2014;111:111.
[13] Milkau U, Bott J. Towards a Framework for the Evaluation and Design of Distributed Ledger Technologies in Banking and Payments. *Journal of Payments Strategy & Systems* 1016;153:153.

improve some features, but take into account more assumptions or limitations:

- the Bitcoin ecosystem changed from a (theoretical) peer-to-peer system with equal "players" to a profit- and speculation-driven environment, in which the processing concentrated significantly on so-called mining pool plus a group of so-called "core developers",
- the idea of an automated synchronisation between participants triggered the concept of "private distributed ledgers", i.e. some "schemes" with identified/registered/on-boarded participants to run a closed network for synchronisation of transactions (e.g. R3/Corda, Hyperledger Fabric, or Ethereum Enterprise Alliance). With known participants and, consequently, a contractual relationship between "members", the problem to achieve a consensus in an open network of anonymous nodes is gone and the synchronisation between known nodes can be implemented by methods such as Byzantine Fault Tolerance (BFT; similar to the redundant autopilots in an airplane with three computers running in parallel to lead the majority to qualify the correct result),
- the technology of blockchain was extended with so-called "virtual machines" sitting on-top of the blockchain, which provide a runtime environment to execute computer-scripts called "smart contracts". Those smart contracts have to be "state machines", i.e. code with an always defined status, and all smart contracts be gathered in a global state machine representing an overall defined status of all contracts at a point in time. This global state machine will be computed at each node of the blockchain network in an asynchronic manner with one probabilistically selected "referee" deciding about the "true" outcome in case of an open network.

If one focusses on "smart contract code" run by registered participants (i.e. assuming some central intermediary to onboard participants) with an atomic protocol (i.e. a protocol, which can fail, but has always a defined status: true or false) to confirm contractual transaction, the story ends here. This is a conventional situation with participants of a scheme, who agree *ex-ante* on an applicable law, on rules and regulations and on obligations (e.g. SEPA, SWIFT, TARGET2-Securities or ISDA).

If one analyses the situation of "smart contracts" on a public blockchain, the following questions have to be discussed in detail:

- dependence on a very complex runtime environment of a software stack with natural errors of any non-trivial software and interdependency of the changing stack,
- certainty of executions, as open networks require some kind of probabilistic consensus mechanism without full finality,
- question of governance or, respectively, life-cycle management of the environment including obligations for quality of service,
- relevance in case of execution of program code in contradiction to the legal situation of the referenced participant (e.g. automatically executed payments in case of default of the payer or "contracts" made by unauthorised participants).

Those questions link the blockchain technology either with bounded rationality and our limitation to foresee the unknown future, or with the governance of contractual relationship (see fig. 1).

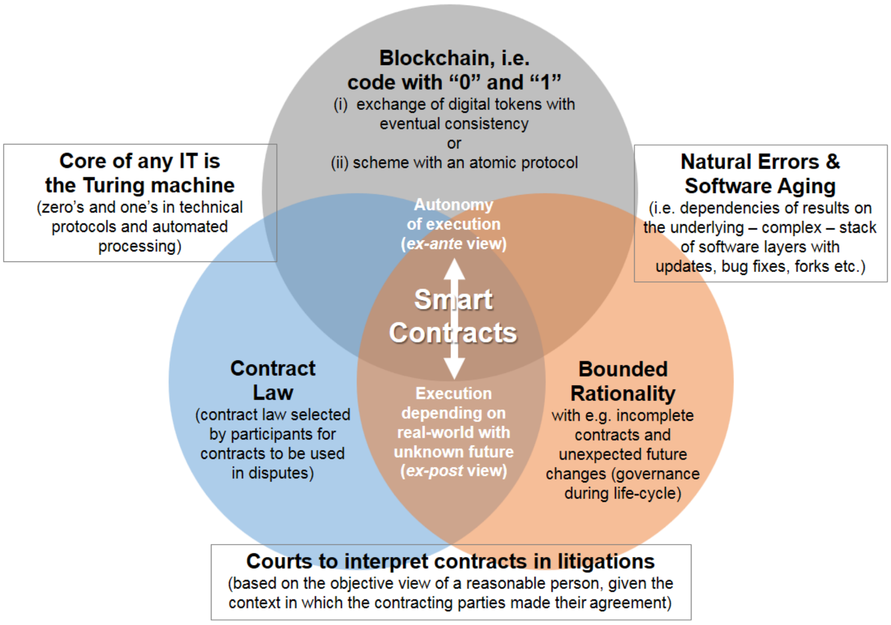
Blockchain, i.e. code with "0" and "1"
(i) exchange of digital tokens with eventual consistency
or
(ii) scheme with an atomic protocol
Core of any IT is the Turing machine
(zero's and one's in technical protocols and automated processing)
Natural Errors & Software Aging
(i.e. dependencies of results on the underlying – complex – stack of software layers with updates, bug fixes, forks etc.)
Autonomy of execution (ex-ante view)
Smart Contracts
Contract Law
(contract law selected by participants for contracts to be used in disputes)
Execution depending on real-world with unknown future (ex-post view)
Bounded Rationality
with e.g. incomplete contracts and unexpected future changes (governance during life-cycle)
Courts to interpret contracts in litigations
(based on the objective view of a reasonable person, given the context in which the contracting parties made their agreement)

Figure 1: The context of "smart contracts" between contracts law, bounded rationality and technical implementations such as the blockchain.

3. Bounded rationality and information technology

The concept of "bounded rationality" was developed by Simon[14] and extended by Gigerenzer and Selten[15]. As the future is uncertain, any decision made by individuals has to be made with limited rationality and based on subjective experiences[16]. In reality, not all information is available or, respectively, the time available to make decisions is not sufficient for a full calculation - whether made by men or computers. Consequently, no computer program (and no contract, however "smart" it would be) can include all possible situations to be managed later *ex-ante* (see Fig. 1).

Additionally, complex software systems represent a special situation of "uncertainty about the future", for which

[14] Simon H A. Bounded rationality and organizational learning. *Organization Science* 1991;125:125.

[15] Gigerenzer G, Selten R. *Bounded rationality. The Adaptive Toolbox*. Cambridge, MA: MIT Press 2002.

[16] See also the Popper-Adorno controversy of 1961 (or "Positivismusstreit" in German), in which both, Popper and Adorno at least agreed that all decisions are generally based on (individual) experiences, (personal) values and the (social) context.

Parnas[17] coined the term “software aging”. Smart contracts are an archetypal example for such software errors, which develop over time and result from the interaction of the different software layers of the software stack with:

- code of the “smart contract code” as a state machine,
- interaction with other smart contracts all gathered in a “block” on the chain,
- compiler and virtual machine (needed to execute the smart contract code),
- potentially supporting services such as storage and messaging,
- software of the blockchain itself (such as Ethereum or Neo et cetera),
- operating systems on the distributed computers to run the local replica of the blockchain,
- network protocol stack for the communication via the internet.

Over time, there will be changes to the elements of the stack with unknown interdependencies, i.e. the whole software can “get old” and will develop “unexpected” errors over time due to the complexity of the technology and the interaction of multiple layers with various parameters.

In the context of a complex software system with inevitable errors and software aging, the vision of “code is law” shows a fundamental flaw. While the concept of Bitcoin works for tokens of “electronic cash” with immutable records of atomic transactions, the extension to smart contracts with a longer life-cycle will end-up with unpredictable errors sooner or later, which contradicts the idea of an *ex-ante* description of smart contracts with a deterministic behaviour. Fig. 2 illustrates a hierarchy of legal components and technical components (shaded in grey) along the life cycle of a contractual agreement between two parties.

4. Incomplete contracts and governance of contractual relationship

The paradigm of “incomplete contracts” was introduced by Grossman and Hart[18], Hart and Moore[19], and Hart[20]. They argue that contracts in reality cannot specify all scenarios for every possible future contingency. In parallel to a contractual relationship, a governance model is required to solve future frictions and intermediaries can take on the role of advisors or mediators (Williamson[21]). The (normative) vision of a frictionless and *ex-ante* ultimately defined contractual relationship is replaced by the understanding of the actual (positive) reality of misunderstanding, errors and inconsistencies (➔ see also: interpretation of contracts according to Prenn v Simmonds, 1971, 1 W.L.R. 1381). In that sense, neither a contract nor any software based on zeros and ones and no blockchain technology will ever be a 100 percent “truth machine”.

If for a split-second, one assumed that a software could be free of any errors and could translate a legal contract into a code 1:1, without any problems in semantics and syntax, this code would reflect the static situation at the time of coding. Within a closed system, this may be applicable, such as in games people play with fixed rules. However, dynamic relationships between agents in a free market economy have to take the principle of human ignorance about the future into account. In general, man-made technology cannot overcome the limitation of bounded rationality. Mechanisms are required to solve the problem of “incompleteness” during the life-time of a contract.

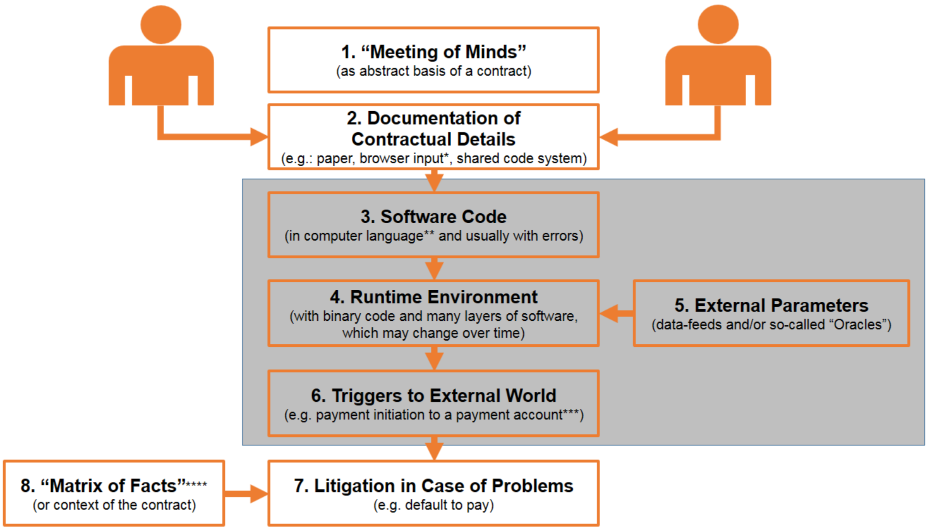
1. “Meeting of Minds”
(as abstract basis of a contract)
2. Documentation of Contractual Details
(e.g.: paper, browser input*, shared code system)
3. Software Code
(in computer language** and usually with errors)
4. Runtime Environment
(with binary code and many layers of software, which may change over time)
5. External Parameters
(data-feeds and/or so-called “Oracles”)
6. Triggers to External World
(e.g. payment initiation to a payment account***)
8. “Matrix of Facts”****
(or context of the contract)
7. Litigation in Case of Problems
(e.g. default to pay)

*Figure 2: Schematic illustration of the different layers of perspective of (smart) contracts in concrete technical implementations with the grey part of technical support functions without relevance for compliance to an agreed contract; *) see e.g. BGH 16.10.2012 X ZR 37/12: Intention of Parties to a contract defined by a communication tool (portal), not parametrisation of or implementation in software (at runtime); **) computer code as used in blockchain systems is dependent on many underlying concepts (object oriented programming) and software layers, so that this code is not easy to understand even for experts (see e.g. “The DAO” hack”); ***) a payment initiation trigger to pay from a (bank) payment account would have to be compliant to the Payment Services Regulation-2 in the EU; ****) according to UK law “Prenn v Simmonds [1971] 1 W.L.R. 1381 (HL)” with the general possibility of a misunderstanding at the time of negotiations and the parties’ actual intent: “The time has long passed when agreements, even those under seal, were isolated from the matrix of facts in which they were set and interpreted purely on internal linguistic considerations.”*

[17] Parnas D L. Software aging. *ICSE '94 Proceedings of the 16th International Conference on Software Engineering* 2014;279:279.

[18] Grossman S J, Hart O. The costs and benefits of ownership. A theory of vertical and lateral integration. *Journal of Political Economy* 1986; 691.

[19] Hart O, Moore J. Property rights and the nature of the firm. *Journal of Political Economy* 1990; 1119.

[20] Hart S. A natural-resource-based view of the firm. *The Academy of Management Review* 1995; 986.

[21] Williamson O E. Transaction-cost economics. The governance of contractual relations. *Journal of Political Economy* 1979;233.

When a predefined system – an agreed contract or written software – is embedded into the context of the world, the problem of the interaction of the internal fixed rules of the "game" and the external dynamic world become obvious. This can be illustrated by three examples:

- computer data are only zeros and ones; with the need to have a suitable "reader" and interface to the outside world, in which legal contracts are relevant,
- a token may be a representation of some internal "value" in a proprietary system such as a jetton in a casino, but for an exchange into a flat currency, some contractual relationship with an external party willing to exchange is necessary.
- the virtual tokens on a blockchain may be used for an ICO (initial coin offering), which can be seen as some kind of online gambling. However, if such tokens represent shares in an enterprise, voting rights in a company, or equity provided to a management team, such exchange of tokens for real money is subject to applicable legislation (see SEC opinion on ICO and DAO, October 2017[22]).

Considering "smart contracts" as scripts running in a computer environment with a distributed database, Greenspan (2016) stated:

"… smart contracts cannot do anything else, and they certainly cannot escape the boundaries of the database in which they reside."[23]

5. Why Contracts

Traditional contracts establish rights and duties to each party of the agreement based on their will. Duties must be exercised by the relevant party. This formal agreement effectively means that private "promises" can be enforced with governmental support. For this to occur, the contract must be compliant with the relevant jurisdiction's legislation.

It is first necessary to understand the creation of 'legal effects' as one of the constitutive elements of a smart contract, the expression of one person's will within codes, as well as a brief overview of the function of contracts in general (in particular: that an obligation arises from the contract and is enforceable by traditional law).

"Rights" and "duties" are core elements of contracts and are inseparably linked to each other. For example, if one party to the contract has a right to receive something from the other party, the other party must have the duty to convey the asset (or whatever that *something* is in their contract). The term "obligation" refers to "rights" as well as "duties" and hence to "the whole relationship" [24].

6. Set-up of a smart contract

6.1 Fundamental Aspects

As the original term "smart contract" was misleading, authors[25] recently started distinguishing between "smart legal contracts" and "smart contract code". The legal part comprises the transfer of a classic contract into a smart contract (step 1 and 2 in Fig. 2), and the potential legal enforcement (step 7 and 8 in Fig. 2). The technical part includes the coding process, the deployment into a runtime environment, and potentially a trigger to external (technical) systems (step 3 to 6 in Fig. 2).

Starting with determination of the declaration of intention, the communication could be performed by selecting a portal and entering the parameters (e.g. by clicking them). The given parameters are displayed and sent to the recipient for confirmation (in the sense of a term sheet). After the confirmation process, a computer script ("smart contract code") is activated by the two-sided acceptance and the smart contract "lives" on the blockchain including the data stored within the script.

Besides the legal contract and its parameters, it is necessary to agree on the applicable law, jurisdiction, format and language. In this way, the contract receives its "rulebook", determining how something should be performed by each party to the contract. Although a "smart contract" is a script with an automated execution, there is no guarantee that the rules will be determined automatically as well.

Smart contracts are often compared to vending machines, an invention probably as old as Roman law: the first vending machine was documented in 62 A.D.[26] A vending machine dispenses small articles such as soft drinks or candy when a button is pressed and a coin, bill or token is inserted. The same applies to the vending machine "smart contract" that could open the door of a rental car if a payment token such as Bitcoin is dropped in.

However, a properly recorded smart contract may in fact be void without the parties being aware of it. Under Common Law principles, a contract is voidable for mistakes and it can therefore be considered ineffective from the moment it was made (i.e. as if it had never taken place). This principle contradicts the principle of blockchain (being immutable).[27] Questions arise, in particular, with regard to the requirements of the conclusion of smart contracts: is it possible to write a legal contract only in code? Referring to the principle of the so-called freedom of contract, a contract does not have to be in any particular form unless a specific form requirement is stipulated by statutory law. Furthermore, under the principle of freedom of choice regarding the contractual language, the parties are free to select code as the language of the contract.[28]

However, smart contracts that are written in code raise concerns with regards to consumer protection: terms and conditions (*Allgemeine Geschäftsbedingungen - AGB*) must be formulated in such a way that a consumer can read them effortlessly.[29] But an average

[22] SEC opinion on ICO and DAO, October 2017

[23] Greenspan G. Why Many Smart Contract Use Cases Are Simply Impossible. https://www.coindesk.com/three-smart-contract-misconceptions (accessed 16 January 2020).

[24] Savelyev A. Contract Law 2.0: <<smart>> contract as the beginning of the end of classic contract law. *Higher School of Economics Research Paper* No. WP BRP 71/LAW/2016:17.

[25] see e.g. EBRD and Clifford Chance *Smart Contracts: Legal Framework and Proposed Guidelines for Lawmakers*. https://www.ebrd.com/documents/legal-reform/pdf-smart-contracts-legal-framework-and-proposed-guidelines-for-lawmakers.pdf (accessed 16 January 2020).

[26] [26] Savelyev A. Contract Law 2.0: <<smart>> contract as the beginning of the end of classic contract law. *Higher School of Economics Research Paper* No. WP BRP 71/LAW/2016:8.

[27] Heckelmann M. Zulässigkleit und Handhabung von Smart Contracts. *NJW* 2018;504:507.

[28] Jünemann M, Kast A. Rechtsfragen beim Einsatz der Blockchain. *ZfgK* 2017;531:534.

[29] Schlosser P. Commentary on § 305 BGB Rn 140. In Staudinger J (eds) *BGB Kommentar* Selier-de Gruyter 2018..

consumer cannot be expected to be able to read code. Apart from that, caution should be exercised with respect to formal requirements. Formal requirements have a protective function that cannot be met by a smart contract written in code. The code of a smart contract is not directly displayed and even if it was, it would not be understood by the majority of the contractual parties. Therefore a translation of the code into English, French, German, etc. could be necessary.[30] By contrast, it can be assumed that it is legally permissible to choose code as the contractual language in B2B commerce. The party that is unable to read code has to bear the language risk that comes with entering into a smart contract written in code.

6.2 German approach

At the moment there is no need for a special law within Germany to establish that a smart contract is legally effective. Rather, the existing principles are applied as usual and are largely transferable to smart contracts. Therefore, regulation is required in a few areas, but not considered to be necessary in principle.[31] Technically, current smart contract (code) environments require that one party writes and deploys the smart contract, and the other party agrees to the display of the (content of the) code in a front-end application, like a portal. This is rather near to the traditional paper-based procedure of offer and acceptance with declaration of will between the contractual parties. However, the accepting/confirming party has to relay on a browser-like portal to read the contract (code) and accept it (electronically). Non-German approaches (e.g. U.S.A.)

The situation in the U.S.A. is obviously different. In 2017, Arizona passed an amendment to its Electronic Transactions Act (so-called *Arizona Electronic Transaction Act - AETA*).[32] The legal Act has mainly supplemented its very short Article 5. The supplement is sometimes referred to as "[g]roundbreaking Blockchain and Smart Contract Law".[33] But is it really (that) ground-breaking? "A signature that is secured through blockchain technology is considered to be in an electronic form and to be an electronic signature."[34] This is of importance because "*AETA* stipulates that records or signatures in electronic form cannot be denied legal effect and enforceability based on the fact they are in electronic form."[35]

7. The legal principle of "code is law"

The legal concept of *pacta sunt servanda* means that agreements which are legally binding must be performed. At first sight, this seems comparable to the principle of "code is law", stipulating that the agreed or programmed aspects may not be changed ex-post. Having said this, the legal concept of *pacta sunt servanda* can be limited by the right of revocation of one party.[36] Moreover, part of the principle of freedom of contract is that contracts can be renegotiated and also modified by the parties through a further contract.[37] Hence, *pacta sunt servanda* means to understand more "what the parties finally agreed or are agreeing on that must be performed". This differs from the principle of "code is law", meaning "the agreed is *unchangeable* and must be performed".

7.1 Interpretation and Renegotiation of Smart Contracts

Having said this, the question arises if smart contracts can and need to be renegotiable. In case of ambiguities, contracts are interpreted and the will of both parties is carved out. Only what the parties really wanted to agree on shall be agreed.

By contrast, interpretations of code performed by machines are based on so-called Boolean logic, meaning something is either true or false. Therefore the maxim "*falsa demonstratio non nocet*", meaning "wrong designation does not harm", does not apply. A famous example among German law students is the so-called *Haakjöringsköd* case, dated 1916.[38] Both parties agreed on the purchase of whale meat. They labelled it *Haakjöringsköd* thinking that this was the correct Norwegian term for whale meat. Since neither party was able to speak Norwegian, they did not know that *Haakjöringsköd* actually meant shark meat. The court later ruled that the misunderstanding did not lead to the agreement being void because both parties wanted the same thing, but mislabelled it. Boolean logic would not be able to solve such a misunderstanding. By knowing only true or false, it would not be satisfied with the delivery of whale meat, even if this was what the parties wanted.[39] These interpretations do not rest upon simple true or false classifications and therefore cannot be used in relation to Boolean logic.

In 2012, the German Federal Court of Justice (*Bundesgerichtshof - BGH*) ruled that the way an automated system is expected to understand and process a declaration of intent, which was made using electronic means of communication and via an automated booking or ordering system, does not determine the content of the declaration. What matters is how the human addressee is allowed to understand the respective declaration in good faith and custom.[40] This means the displayed and confirmed content (see section 6.2) is binding but not the bits and bytes within a computer system (such as a blockchain).

7.2 (Smart) dispute resolutions and risk management

If a smart contract is set up and a dispute arises the (local) courts will not be experts in blockchain or coding and specialists will be needed. In 2017, Datarella, a Munich-based provider of blockchain

[30] Jünemann M, Kast A. Rechtsfragen beim Einsatz der Blockchain. *ZfgK* 2017;531:534.

[31] E.g. the Blockchain Verband considers the regulation of blockchain-technology regarding online business / finance by usage of token as helpful, Blockchain Bundesverband. Statement of the Federal Blockchain Associataion to the Committee on the Digital Agenda. Deutscher Bundestag, Ausschussdrucksache 19(23)028 page 32.

[32] AZ HB2417 especially the amendment of Article 5 "Blockchain Technology".

[33] Neuburger J. Arizona Passes Groundbraking Blockchain and Smart Contract Law – State Blockchain Laws on the Rise. https://newmedialaw.proskauer.com/2017/04/20/arizona-passes-ground-breaking-blockchain-and-smart-contract-law-state-blockchain-laws-on-the-rise/ (accessed 16 January 2020).

[34] Art. 5 A (section 44-7061), Title 44, Chapter 26, Arizona Revised Statutes.

[35] Neuburger J. Arizona Passes Groundbraking Blockchain and Smart Contract Law – State Blockchain Laws on the Rise. https://newmedialaw.proskauer.com/2017/04/20/arizona-passes-ground-breaking-blockchain-and-smart-contract-law-state-blockchain-laws-on-the-rise/ (accessed 16 January 2020).

[36] Palandt, 2017, b. section 145 footnote 4 lit. a.

[37] BeckOGK/Herresthal, 1.5.2018, BGB section 311 footnote 128.

[38] Reichsgericht, RGZ 99, 147.

[39] Lessig L, *Code is Law. On Liberty in Cyberspace.* Harvard Magazine, https://harvardmagazine.com/2000/01/code-is-law-html (accessed 16 January 2020).

[40] BGH judgement dated 16.10.2012 – file no. X ZR 37/12.

solutions, launched an arbitration proceeding based on blockchain technology: the Codelegit Certified Blockchain Arbitration Library (Datarella's legal library for smart contracts).[41] Codelegit took the opportunity to specialise in the implementation of arbitral proceedings within smart contracts. In case of a (detected) legal breach or a bug in the smart contract, the respective party triggers the arbitration process by calling the function "*pauseAndSendToArbitrator ()*". A grace period pausing the execution of the smart contract will commence. The arbitration service will then be performed. The parties can choose from a database of arbitrators, who may be legal experts and also technicians who understand blockchain technology and smart contracts.[42] Usually, the arbitrator will invite the parties to join a video conference, but the hearing can also be a meeting in person. It is not required that the parties are represented by a lawyer, but they are free to have one.[43] Afterwards, the smart contract will then be continued, modified by the appointed authority as foreseen in the arbitration library, or ended.[44] Depending on the settlement or award, the appointing authority calls function "*continueContract ( )*", modifies the smart contract, or calls function "*endContract ( )*". [45] Triggering the arbitration function has to be performed by one of the parties to the smart contract (e.g. by clicking on a link).[46] Furthermore, it seems likely that certain parameters might trigger an arbitration process by itself (e.g. receiving an amount less than what was set out in the smart contract; or receiving the right amount but at a later point in time than what was agreed).

By the choice of arbitration, the parties restrict their access to the state courts by contract. This is a reflection of the principle under which the parties determine the scope of the case and whether or not a court proceeding takes place at all (*Dispositionsgrundsatz*). Therefore, the model clause for arbitration in 2018 by the German Arbitration Institute (DIS) provides that all disputes arising out of or in connection with the relevant contract or its validity shall be finally settled in accordance with the Arbitration Rules of the DIS without recourse to the ordinary courts of the state.

New arbitration rules were developed for the CCP's 2017 proceedings - the "Blockchain Arbitration Rules". The advantage of the Blockchain Arbitration Rules over traditional Rules is that all parties involved have access to the documents which are made available by a blockchain that serves as a verification tool.[47] Several tech enthusiasts claim that blockchain arbitration will replace traditional arbitration.[48]

The existence of such arbitration clauses is highlighting that the notion of "code is law" does not mean that code is always right.

7.3 Freezing and Exit Scenarios

Ultimately, the question then arises as to whether smart contracts can be updated, patched or stopped. Smart contracts deployed on a blockchain cannot be modified since they are permanently written on the blockchain.[49] But given the fact that smart contracts cannot be changed, unless the possibility to "freeze" the execution of the smart contract is encoded, how could a government agency react if the review of the smart contract has shown that the aged smart contract is vulnerable to hacking and could lead to unwanted results?

If an "emergency exit" was coded into the respective smart contract then the smart contract could be stopped ("frozen") or ended ("killed"). The essential key data of the contract (e.g. contractual parties, object of purchase, purchase price) could be extracted from the aged contract and a new smart contract with the same content could be coded.[50] This would be very similar to an "update". Such a mechanism (automatic data readout) could, in principle, be written into the code of a "legal model smart contract". The address of the aged smart contract would have to be updated, and the users would see the address of the new smart contract.

Due to the immutability of smart contracts, as long as there is no "hard fork", the parties are only able to rescind or unravel the smart contract if such rights are programmed in the smart contract from the outset. It is, however, debatable whether other "emergency exits," other than rights of rescission, can be written into the code of a smart contract.[51] Indeed, transactions on the blockchain do not require the control or approval of a trusted third party. However, that such "emergency exits" have to be encoded means that the parties must rely (i) on the coder and (ii) on the correct execution at runtime / in the runtime environment to ensure fairness. But how can an IT specialist set up a complex legal agreement without a legal background or vice versa: how can a lawyer set up such a contract without the detailed technical knowledge including about the runtime environment and possible dependencies at runtime? Even if the coder of a smart contract was both a lawyer and an IT expert, the question would arise as to how she/he could foresee every possible scenario in an uncertain future. However, it would be necessary (but not feasible) to include all possible situations and solutions into the smart contract, or to abandon the fairness aspect of a contract as provided under traditional legal principles.

[41] https://datarella.com/ (accessed 16 January 2020); http://codelegit.com/blog (accessed 16 January 2020).
[42] CodeLegit White Paper on Blockchain Arbitration, https://docs.google.com/document/d/1v_AdWb-Muc2Ei70ghITC1mYX4_5VQsF_28O4PsLckNM4/edit# (accessed 16 January 2020).
[43] CodeLegit White Paper on Blockchain Arbitration, https://docs.google.com/document/d/1v_AdWb-Muc2Ei70ghITC1mYX4_5VQsF_28O4PsLckNM4/edit# (accessed 16 January 2020).
[44] A complex scheme is given in the Appendix – Arbitral Proceeding using CodeLegit Arbitration Library and Blockchain Arbitration Rules, https://docs.google.com/document/d/1v_AdWb-Muc2Ei70ghITC1mYX4_5VQsF_28O4PsLckNM4/edit (accessed 16 January 2020).
[45] Appendix – Arbitral Proceeding using CodeLegit Arbitration Library and Blockchain Arbitration Rules, https://docs.google.com/document/d/1v_AdWb-Muc2Ei70ghITC1mYX4_5VQsF_28O4PsLckNM4/edit(accessed 16 January 2020).
[46] The OpenLaw Team, OpenCourt: Legally Enforceable Blockchain-Based Arbitration. https://media.consensys.net/opencourt-legally-enforceable-blockchain-based-arbitration-3d7147dbb56f (accessed 16 January 2020).
[47] CodeLegit White Paper on Blockchain Arbitration, https://docs.google.com/document/d/1v_AdWb-Muc2Ei70ghITC1mYX4_5VQsF_28O4PsLckNM4/edit# (accessed 16 January 2020).
[48] Paulsson M R P. The Eve of the New York Convention's 60th Anniversary and the Birthday Party: How to Prepare with too Many Guests at the Table. "Il ne faut pas melangér les tables". Kluwer Arbitration Blog http://arbitrationblog.kluwerarbitration.com/2018/06/21/eve-new-york-conventions-60th-anniversary-birthday-party-prepare-many-guests-table-il-ne-faut-pas-melanger-les-tables/ (accessed 16 January 2020).
[49] Grincalaitis M. Can a Smart Contract be upgraded/modified? Is CPU mining even worth the Ether? https://medium.com/@merunasgrincalaitis/can-a-smart-contract-be-upgraded-modified-1393e9b507a (accessed 16 January 2020).
[50] Grincalaitis M. Can a Smart Contract be upgraded/modified? Is CPU mining even worth the Ether? https://medium.com/@merunasgrincalaitis/can-a-smart-contract-be-upgraded-modified-1393e9b507a (accessed 16 January 2020).
[51] Heckelmann M. Zulässigkleit und Handhabung von Smart Contracts. *NJW* 2018;504:507.

8. Challenges for "Code is law" and self enforceability of smart contracts

8.1 Financial Distress

8.1.1 Insolvency

The vast majority of insolvency proceedings in Germany follow the standard procedure. Under the insolvency standard procedure, as a general rule, the debtor's assets, which are part of the insolvency estate ("*Insolvenzmasse*"), will be liquidated and the proceeds will be distributed among the creditors, who have registered their receivables to the insolvency table. The term insolvency estate is defined as all assets owned by the debtor on the date when the proceedings are initiated as well as any assets acquired by him during the proceedings. By the time insolvency proceedings are initiated under the standard procedure, an insolvency administrator will be appointed by the competent court. The debtor's right to manage and dispose of the insolvency estate shall be vested in the insolvency administrator. In general, the core functions of the insolvency administrator are to seize the insolvency estate, to provisionally continue the operations of the company, to liquidate the assets, and to distribute the proceeds among the creditors.

However, even before insolvency proceedings are initiated (but once the debtor is already in financial distress), special legal and compliance requirements must be taken into account. According to Supreme Court rulings[52], the company's management may not discriminate against individual creditors during the period of financial distress prior to an insolvency proceeding. Nevertheless, paying supplier's invoices for certain goods and services that are necessary to perform the company's business could be in the scope of an exemption.[53] Depending on the severity of the financial distress, individual expenditures must be measured in economic terms. In addition, the total assets must always be kept in view and a precise distinction between the specific outstanding receivables must be made.[54] For example, social security contributions must be paid (regularly), whereas special agreements may need to be made between trading partners.

Implementation of a standard insolvency procedure or a standard financial distress procedure into a smart contract will not (at least not for the moment) meet the minimum legal requirements. By taking away every opportunity for the debtor to act in an economically advantageous manner, the smart contract could ultimately disadvantage the creditors as a whole. This is because sometimes the company could have been rescued through targeted investments so that all creditors could have been repaid (even if somewhat delayed). Since it is not only a matter of focusing on pure statistics, but also on economic, social (law), individually compatible and general corporate aspects, it does not seem possible (at least not currently) to map all this out on a blockchain.

The situation in relation to any other and/or future creditors is unfavourable as well. In order to ensure a functioning application of insolvency law, the position of each creditor in relation to the debtor would have to be accurately shown. This is the only way to ensure that the smart contract complies with the intended purpose under the Insolvency Code. But this goes hand in hand with the fact that every creditor knows exactly where he stands in the overall financial situation of the debtor. In the absence of smart contracts, the assumption is that all unsecured creditors rank *pari passu*. Self-executing smart contracts are incompatible with this assumption (unless self-execution is limited to preserve the *pari passu* ranking and this requires a "transparent debtor"). Smart contracts in a financial crisis will probably be negotiated between the creditors (among themselves) rather than with the "transparent debtor".

This in turn favours debtors who do not use smart contracts and do not (have to) disclose their overall financial situation.

8.1.2 Bonds and freedom of contract

Besides the *pacta sunt servanda* concept, the German legal system is characterised by the aforementioned principle of freedom of contract, which is based on the concept of private autonomy. Every individual has the right to shape and form their living conditions individually and freely by entering into contracts for any legal purpose they desire.[55] As a part of this, every individual is equally free to insist on the non-performance of an obligation.

If the smart contract prevents the possibility of insisting on non-performance this could lead to problems. As an example: Person A has purchased a bond from Person B with a term of several years and a fixed interest rate[56]. The interest is payable annually. B runs into payment difficulties within the first year (e.g. due to the sale of property and the absence of payments to him). The payment difficulties are (foreseeable) for only a short duration (there is still a large number of due invoices outstanding). If the smart contract now automatically cancels the bond and collects the total amount, this would possibly force B into insolvency. Even if B could demonstrate to A that future payments can be made as planned, and that even the delay in interest payments can be compensated, A could not stop the automatic settlement of the smart contract. This would ultimately, put A in a worse position as well, given that instead of receiving the full amount (with a slight delay), he would have to be satisfied with a smaller amount.

The aforementioned bond-example shows the problem between the principles of freedom of contract and "code is law". In case of a traditional contract, it is highly unlikely that the parties would perform adversely to their own understanding, but smart contracts are performed regardless. Lessig wrote almost two decades ago: "Every age has its potential regulator, its threat to liberty. [...] Ours is the age of cyberspace. It, too, has a regulator. This regulator, too, threatens liberty. [...] This regulator is code - the

[52] E.g. in relation to affiliate companies *BGH, NJW* 2008;2504.

[53] This exception is expressed e.g. in section 64 of the German Limited Liabilities Companies Act.

[54] The company's management has to file for insolvency proceedings at certain point in time, set out within the German Insolvency Act. If an insolvency application is not filed, there is a risk of severe penalties.

[55] Musielak H-J. Vertragsfreiheit und ihre Grenzen. *JuS* 2017;949:949.

[56] If would be even more challenging to take the example of a floating rare bond with a formula based on an interest rate index without including the possibility of negative interest rates.

software and hardware that make cyberspace as it is."[57] Was Lessig right after all? Can code be a threat to liberty?[58]

8.2 Aged or hacked Software

The code of a smart contract is software after all, and all software used over a long period of time is prone to error. Programming an everlasting perfect (and not trivial) smart contract is an impossible thing to do. Consequently, the original intention of the parties of a smart contract and the outcome might not be the same because aged software can develop unexpected errors. In contrast to other computer programs, smart contracts are unchangeable due to the immutability of the underlying blockchain. This means that vulnerabilities and software errors cannot be fixed. Recent reports indicate that over 34,000 published Ethereum smart contracts are vulnerable.[59] Little can be done about this because (as set out above in 0) smart contracts cannot be updated or patched.[60]

The code of smart contracts is also in the spotlight from hacker attacks: e.g. in June 2016 a hacker exploited a software weakness and transferred approximately 3.6 million ETH - 1/3 of the total ETH raised by the DAO offering - to his own wallet.[61] As a general rule, the older the code, the easier it is to hack. Both the aging of the code, and the code changed by hackers, leads to the fact that the contract will no longer align with the originally intended purpose of the agreement. Under German law, such an interference with the basis of the transaction leads to the contract's voidance (if an adjustment is not possible): "if circumstances which became the basis of a contract have significantly changed since the contract was entered into and if the parties would not have entered into the contract or would have entered into it with different contents if they had foreseen this change [...] one of the parties cannot reasonably be expected to uphold the contract without alteration."[62] Setting up long-term smart contracts without taking into consideration that they age (or that they become increasingly vulnerable to hacking) could lead to an interference with the basis of the transaction. "Code is law" therefore cannot apply from that point on, as the parties would have agreed otherwise. Assuming that we cannot adapt to the situation, the principle "code is law" will necessarily have to be broken here.

Additionally, blockchain based on a so-called "proof-of-work" consensus are by design vulnerable to "51 % attacks", as the creator of Litecoin Charlie Lee stated[63] in cointelegraph.com on Jan. 9, 2019:

"By definition, a decentralized cryptocurrency must be susceptible to 51% attacks whether by hashrate, stake, and/or other permissionlessly-acquirable resources. If a crypto can't be 51% attacked, it is permissioned and centralized."

This statement was made in the wake of recent news about "unusual mining activity" on the Ethereum Classic blockchain (see the same article on cointelegraph.com). Regardless of this particular event being qualified as a true "51 % attack", some successful 51 % attacks on minor blockchain/cyber tokens were reported in 2018:

- Double Spend Attack on Bitcoin Gold in May 2018: (Bitcoin Gold director of communications Edward Iskra warned[64] "that a malicious miner was using the exploit to steal funds from cryptocurrency exchanges. To execute the attack, the miner acquired at least 51 percent of the network's total hashpower, which provided them with temporary control of the blockchain").
- Block Withholding Attack on Monacoin also in May 2018: (According to CNN[65], the attack appears to have been a selfish mining attack, where one miner successfully mined a block on the blockchain but did not broadcast the new block to other miners. The miner, still unknown to this day, had enough computing power to take as much as 57% of the hashrate at one point in order to execute the attack).

It is important to remark that even with a "51 % attacks", an attacker cannot manipulate the stored data on a blockchain arbitrarily, but can – as in the two examples – conduct a double spending and/or redirect transaction. The consequences for smart contract performance are unclear, as it depends on the underlying blockchain (public vs. private and type of consensus method). However, those cases made it clear that – in principle – blockchains are vulnerable and the security of the runtime environment has to be carefully taken into account.

[57] Lessig L, *Code is Law. On Liberty in Cyberspace.* Harvard Magazine, https://harvardmagazine.com/2000/01/code-is-law-html (accessed 16 January 2020).

[58] Moreover, the principle of "code is law" involves difficulties with regards to debtor protection regulations. The self-enforceability of smart contracts is a mixed blessing: it would be possible to set up a lease agreement and determine that the rent shall be taken from the bank account of the tenant if it is not transferred until the end of the month. If the amount on the bank account was not sufficient to pay the full rate at the end of the month, the smart contract would automatically take the highest amount possible from the bank account. However, under German law, certain thresholds must be considered when taking a natural person's money. The code would ignore these thresholds unless they were programmed into the smart contract. Nevertheless even if they were part of the code, the risk of these provisions being circumvented would remain. This is demonstrated by the fact that these thresholds can differ from time to time. For this reason, the enforcement by writ has to be performed by an (official) attachment order, examining if certain thresholds apply in the first place. These specific regulations are based on German welfare state principle and can differ from country to country.

[59] Nikolić I. Finding The Greedy, Prodigal, and Suicidal Contracts at Scale. *ACSAC '18 Proceedings of the 34th Annual Computer Security Applications Conference* 2018;653:653.

[60] Nikolić I. Finding The Greedy, Prodigal, and Suicidal Contracts at Scale. *ACSAC '18 Proceedings of the 34th Annual Computer Security Applications Conference* 2018;653:653.

[61] US Securities and Exchange Commission. *Report of Investigation Pursuant to Section 21(a) of the Securities Exchange Act of 1934: The DAO.* Page 9. https://www.sec.gov/litigation/investreport/34-81207.pdf. (accessed 16 January 2020); Pastebin. An Open Letter. https://pastebin.com/CcGUBgDG (accessed 16 January 2020); Price R. Digital currency Ethereum is cratering because of a $50 million hack. http://www.businessinsider.de/dao-hacked-ethereum-crashing-in-value-tens-of-millions-allegedly-stolen-2016-6?r=UK&IR=T (accessed 16 January 2020).

[62] Section 313 German Civil Code.

[63] Alexandre A. Litecoin's Charlie Lee: Dezentralized Crypto 'Must Be Suspectible to 51% Attacks'. https://cointelegraph.com/news/litecoins-charlie-lee-decentralized-crypto-must-be-susceptible-to-51-attacks (accessed 16 January 2020).

[64] Wilmoth J. Bitcoin Gold Hit By Double Spend Attack, Exchanges Lose Millions. https://www.ccn.com/bitcoin-gold-hit-by-double-spend-attack-exchanges-lose-millions/ (accessed 16 January 2020).

[65] Gutteridge D. Japanese Cryptocurrency Monacoin Hit By Selfish Mining Attack. https://www.ccn.com/japanese-cryptocurrency-monacoin-hit-by-selfish-mining-attack/ (accessed 16 January 2020).

8.3 Suretyship upon First Demand

The principle "code is law" with its self-enforceability is challenged in relation to suretyships upon first demand and terms and conditions.

The problem is shown by the following example:

A and B are in a contractual relationship regarding the delivery of goods. To make it easier for B to do business, he does not have to pay for the goods immediately, but only collectively at a later point in time. As compensation for this concession, however, A requires a suretyship upon first demand. Hence, B agrees with its bank (C-Bank) to grant A suretyship upon first demand in favour of A. C-Bank demands an additional fee in case A demands the payment by C-Bank.

Other than a "standard" suretyship, the suretyship upon first demand gives the recipient (A) the right to demand the payment[66] by the bail (C-Bank)[67] without giving the bail any chance of intervention.[68] After the performed payment, the bail has the chance to draw back the payment. All in all, the only thing that matters is the formal aspects of the "demand", such as showing a document saying that the goods were delivered.[69] Any substantive objections, such as wrongfulness of the delivered goods or false goods, do not matter at all.[70] Only if a demand is obviously unlawful, will the mere demand not be sufficient.[71] If the (formally right, but in substantive aspects unjustified) demand is made too hastily, a complex situation arises (especially if a smart contract was coded):

When setting up the agreements and security using a smart contract, the demand will be performed if specific parameters are (or are not) given (e.g. no payment at agreed date). However, if an error in the delivery contract (e.g. delivery of the false goods) was the reason for the non-payment, the smart contract does not recognise this and automatically triggers the demand. This brings along a negative situation for every party of the agreement: B is now obliged to pay the additional fee to C-bank (as well as expenditures of C-Bank if a reclaim is performed, so-called *Aufwendungsersatz*); C-Bank now has to claim back their payment (regarding A by the so-called *Rückforderungsprozess*) and therefore has to bear B's and A's insolvency risks; and A must expect to be sued by both B and C-Bank. In addition to this, it is likely that B will change its supplier.[72]

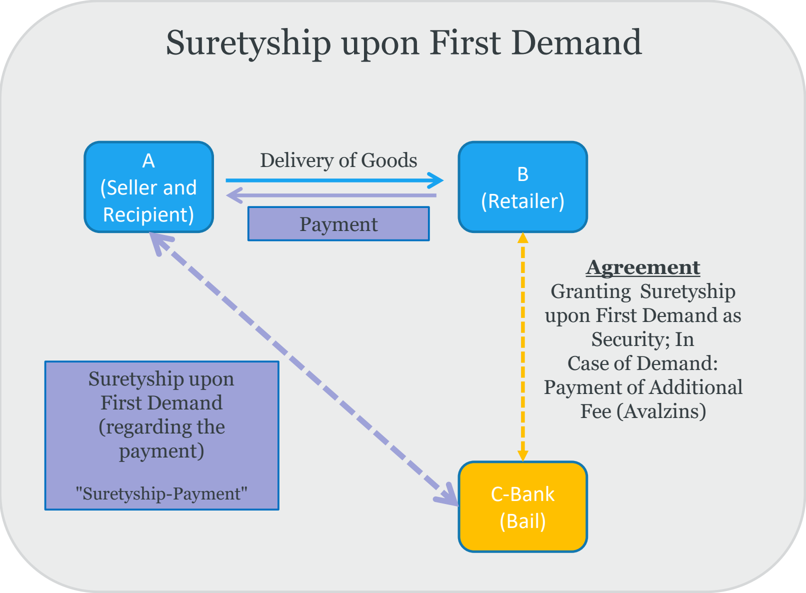
Suretyship upon First Demand
A (Seller and Recipient)
Delivery of Goods
Payment
B (Retailer)
Agreement
Granting Suretyship upon First Demand as Security; In Case of Demand: Payment of Additional Fee (Avalzins)
Suretyship upon First Demand (regarding the payment)
"Suretyship-Payment"
C-Bank (Bail)

Figure 3: Illustration of contractual agreement in the context of a survey ship.

In practice, however, the situation is even more complicated and thus prone to errors. The stipulations regarding the suretyship upon first demand are often to provide security embedded in general terms and conditions, particularly in the agreement between A and B.[73] However, if A obliges B to provide such security as per his general terms and conditions, it is very likely that this will unreasonably disadvantage B (contrary to the German principle of good faith).[74] Persons who do not conduct any banking business, international business transactions, and who are not familiar with suretyships on a professional basis are unable to assess the risk in the right way. According to the Federal Court of Justice, for them such a clause is a surprising and ambiguous section within the meaning of 305c German Civil Code.[75] This means that a clause to this extent will be rendered ineffective, so that the smart contract should not execute itself.

Of course, coding could be in place which accommodates this scenario. But the real challenge for the "code is law" principle is to determine when to apply the code for terms and conditions and when not to. Under German law terms and conditions "are all contract terms pre-formulated for more than two contracts which one party to the contract (the user) presents to the other party upon the entering into of the contract."[76] It is not necessary that the smart contract is actually used a third time. If the party to the smart contract already plans to use it more than twice at the first use, it is already classified as general terms and conditions within the meaning of German Civil Code.[77] The smart contract is not able to determine if the user plans to use the contract more than twice by using it the first time. Therefore, the principle of "code is law" is hard to maintain in this complex situation.

[66] These regularly involve cash payments. Although other forms of suretyships are also possible, these are not considered here.
[67] Affiliated companies are likely to grant a suretyship upon first demand as well.
[68] Regarding a standard suretyship, the bail has the chance to prevail the demand due to objection of demands. Such are that the surety recipient has to claim the debtor first or any objection of demands the debtor has against the suretyship recipient.
[69] Habersack in *MüKo BGB*, § 765 Rn. 102.
[70] Oepen K. Auf erstes Anfordern versprochene Bürgschaften und Garantien. *NJW* 2009;1110:1110.
[71] This is the case, for example, if the demand was performed regarding another agreement between A and B, that is not secured by C-Bank. Pioch C. Einstweilige Verfügung gegen die Inanspruchnahme einer Bürgschaft auf erstes Anfordern. *JuS* 2018;438: 439.
[72] Nobbe G. In Schimansky H, Bunte H-J, Lwowski H J. *BankR-HdB* § 91, Rn. 581.
[73] Typically, A obliges B in the agreement to provide a suretyship upon first demand.
[74] Pioch C. Einstweilige Verfügung gegen die Inanspruchnahme einer Bürgschaft auf erstes Anfordern. *JuS* 2018;438:439; w Nobbe G. In Schimansky H, Bunte H-J, Lwowski H J. *BankR-HdB* § 91, Rn. 558.
[75] Federal Court of Justice. *NJW* 2002;3627:3628.
[76] Section 305 paragraph 1 sentence 1 German Civil Code *(Bürgerliches Gesetzbuch – BGB)*.
[77] Basedow J. In MüKo BGB, § 305 Rn. 18.

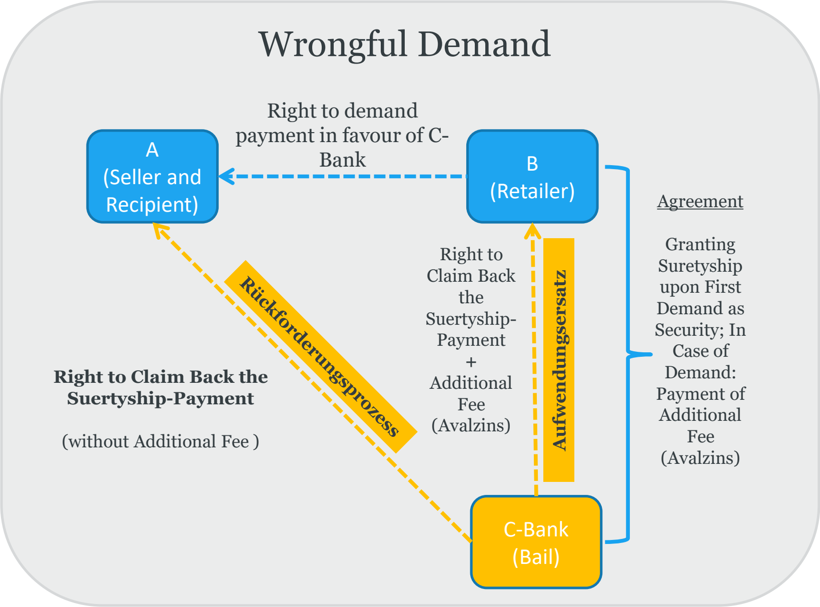
Wrongful Demand
A (Seller and Recipient)
Right to demand payment in favour of C-Bank
B (Retailer)
Agreement
Granting Suretyship upon First Demand as Security; In Case of Demand: Payment of Additional Fee (Avalzins)
Right to Claim Back the Suertyship-Payment + Additional Fee (Avalzins)
Aufwendungsersatz
Rückforderungsprozess
Right to Claim Back the Suertyship-Payment
(without Additional Fee)
C-Bank (Bail)

Figure 4: Illustration of the consequences of a wrongful demand under the suretyship. The bank can claw back the payment and it can demand reimbursement from A.

8.4 The State's Enforcement monopoly

Compulsory enforcement is seen as the necessary state institution of any developed legal system based on the prohibition of self-help.[78] The compulsory enforcement is part of the state's monopoly on legal protection.[79] The German legislator permits private proceedings (so-called arbitration proceedings) in addition to official proceedings. However, if the arbitral award is then to be enforced, it must be declared enforceable by a German Court first. The enforcement itself is performed in the usual way. This ensures that the power of enforcement remains exclusively with the state.[80]

This rigorous approach is based on the fact that the enforcement of a judgment is a massive intervention into the fundamental rights of the person concerned. In addition to the protection of property,[81] interventions into the fundamental right of personal freedom[82] are quite possible (depending on the manner of enforcement).[83] Such interventions find their limits in the fundamental right to human dignity[84] or in the state protection of the family.[85] This shows that important principles have to be weighed against each other, especially in enforcement. Some of these principles find their expression within the rules of the German Code of Civil Procedure. Nevertheless, these principles need to be kept in mind again and again as well as applied in case of uncertainties.

Regarding coded self-enforceability of smart contracts, such complex considerations cannot be reflected on the blockchain (at least not according to the current state of the art). In addition, the question arises as to whether the state's enforcement monopoly would be undermined if smart contracts were assumed to be used on a large scale. This would mean that the elaborately developed prime principles and considerations of the state would not be applied. Rather, a multitude of self-encoded enforcement systems would exist, each of them on an individual basis. Even where private enforcement requires self-help, such as under article 9 of the Uniform Commercial Code (UCC), or as allowed under the relevant contract, such self-help is typically subject to the without "breach of peace" standard. It is intended to safeguard the debtor from abuses that can occur during self-help repossession and it is typically underlined by the law but left to *ex post facto* determination by the courts.

As shown by this, the "without breach of peace" standard is not the result of an act of self-help, in the manner that self-help repossession is always executed without a "breach of peace". It's rather a condition for legitimate self-help.[86] In other words, self-help is only legitimate, when used without breaking the peace.

A "breach of peace" is for example determined by the courts in case of physical assault by the repossessor.[87] It may also be determined in other cases, such as cases where emotional distress is caused on the debtor, where law enforcement officers have provided assistance or where there has been an impact on third parties (e.g. children of the debtor). However, in these other cases, there is no clear line for the determination of a breach of peace.[88]

The exercise of self-help in violation of the "without breach of peace" standard can lead to serious consequences, such as criminal liability, compensatory damages, statutory and punitive damages as well as loss of the right to a deficiency claim.[89]

[78] Gaul H F, Schilken E, Becker-Eberhard, E. *ZwangsvollstrR*. Vol 12. § 1 Rn. 9. München: C. H. Beck, 2010.

[79] Gaul H F, Schilken E, Becker-Eberhard, E. *ZwangsvollstrR*. Vol 12. § 1 Rn. 12. München: C. H. Beck, 2010.

[80] Münch J. In *MüKoZPO* § 1060 Rn. 3ff.

[81] Granted by Article 14 of the German Constitution (*Basic law of the Federal Republic of Germany*). Gaul H F, Schilken E, Becker-Eberhard, E. *ZwangsvollstrR*. Vol 12. § 3 Rn. 2. München: C. H. Beck, 2010.

[82] Granted by Article 2 paragraph 1 of the German Constitution (*Basic law of the Federal Republic of Germany*). Gaul H F, Schilken E, Becker-Eberhard, E. *ZwangsvollstrR*. Vol 12. § 3 Rn. 2. München: C. H. Beck, 2010.

[83] For example, it is possible to force a coercive detention in order to force the information on quantities, section 802 g German Civil Code. Gaul H F, Schilken E, Becker-Eberhard, E. *ZwangsvollstrR*. Vol 12. § 3 Rn. 2. München: C. H. Beck, 2010.

[84] Granted by Article 1 paragraph 1 of the German Constitution (*Basic law of the Federal Republic of Germany*). Gaul H F, Schilken E, Becker-Eberhard, E. *ZwangsvollstrR*. Vol 12. § 3 Rn. 2. München: C. H. Beck, 2010.

[85] Granted by 6 of the German Constitution (*Basic law of the Federal Republic of Germany*). Gaul H F, Schilken E, Becker-Eberhard, E. *ZwangsvollstrR*. Vol 12. § 3 Rn. 2. München: C. H. Beck, 2010.

[86] § 9-609 of the UCC:
(a) After default, a secured party: (1) may take possession of the collateral; (…).
(b) A secured party may proceed under subsection (a): (1) pursuant to judicial process; or (2) without judicial process, if it proceeds without breach of the peace.

[87] Gikay A A, Stanescu C G. The Reluctance of Civil Law Systems in Adopting the UCC Article 9 "Without Breach of Peace" Standard- Evidence from National and International Legal Instruments Governing Secured Transactions. *J. Civ. L. Stud.* 2017;110:110.

[88] McRobert R. Defining „Breach of the Peace" in Self-Help Repossessions. Washington Law Review 2012; 570; Corkery M, Silver-Greenberg J. Miss a Payment? Good Luck Moving That Car. N.Y. TIMES, 24.9.2014, https://dealbook.nytimes.com/2014/09/24/miss-a-payment-good-luck-moving-that-car.

[89] Gikay A A, Stanescu C G. The Reluctance of Civil Law Systems in Adopting the UCC Article 9 "Without Breach of Peace" Standard- Evidence from National and International Legal Instruments Governing Secured Transactions. *J. Civ. L. Stud.* 2017;110:112.

8.5 Freedom of will in form of efficient breaches

Due to its self-enforceable nature, there is also no room for a so-called efficient breach in a smart contract. An efficient breach is a breach of contract based on the consideration that a breach is economically more efficient than performance under the contract.[90] Typically, the parties are not able to influence the smart contract. By using a strict understanding of "code is law", one could say that any established remedy for a breach of contract (such as damages or penalties) would not be available under a smart contract, unless it was explicitly included in its code.

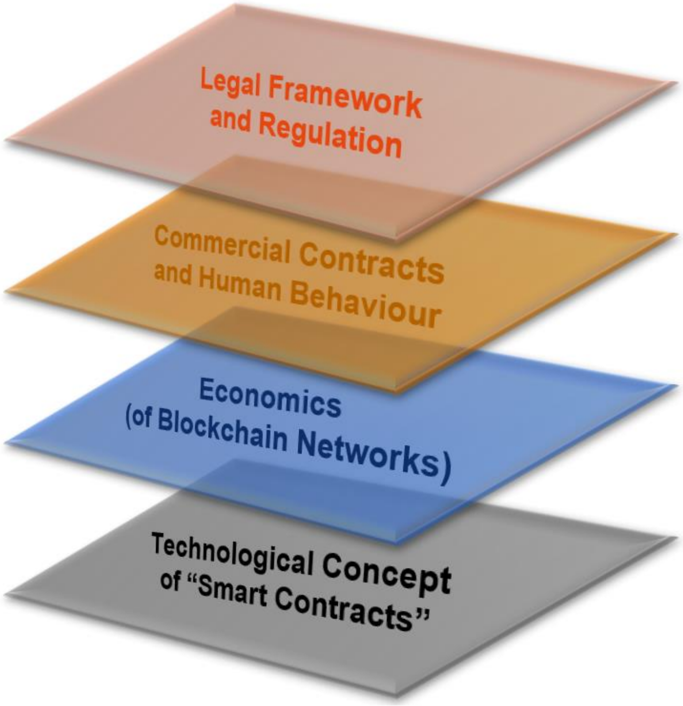

Figure 5: Technological concept of "smart contracts".

Of course, a smart contract can include if-then-else Statements and/or an "exit" to an external (i.e. new) data input via an so-called "Oracle" et cetera. However, all these possibilities are ex-ante programmed code by a programmer with their assumptions about an uncertain future and, consequently, cannot cover unexpected events, for which "efficient breaches" will be the only remedy from an economy perspective.

9. Conclusion: A look into the future

Up to now, the discussion around "smart contracts" usually started from a technical perspective. As Fig. 5 illustrates, technological concepts are an important, but not the only part of a whole stack of elements: from technology via the question about the economics[91] of (the operation of) blockchains, to the commercial contracts between human actor and, finally, the legal and regulatory framework.

Consequently, the vision of "Code is Law" (or "Code as Law") ignores the interaction of these elements and the context of the social, commercial and legal world. Vice versa, a programmer of a "smart contract" in the future has to know what he is (legally) allowed to code. He will most likely need basic legal knowledge if he does not want to cooperate with a law firm. The law firms themselves will have to hire coders if they want to offer smart contracts. No matter in which form, it will be unavoidable that each side knows at least a little about the other one. Universities are also recognising this trend and are increasingly establishing institutes for legal informatics.[92] Therefore, "Code" can be a part of "Law", but will not replace the law.

From an abstract point of view, one can distinguish three different Levels of Intention: (i) the original "Meeting of Wills" in the context of the world and customs; (ii) the best effort to put the intention(s) into a formalised documentation - or code - under uncertainty about the future; and (iii) the code or "smart contract" with all its possible if-then-else statements, but without an own will and only with the codified part of the intention of the programmers.

[90] Bigoni M, Bortolotti S, Parisi F, Porat A. Unbundling Efficient Breach. *Coase-Sandor Working Paper Series in Law and Economics* 2014.
[91] Budish E. The Economic Limits of Bitcoin and the Blockchain. https://faculty.chicagobooth.edu/eric.budish/research/Economic-Limits-Bitcoin-Blockchain.pdf (accessed 16 january 2020).
[92] Rechtsinformatik. https://www.rechtsinformatik.saarland/de/ (accessed 16 January 2020); Rechtsinformatik. https://www.uni-regensburg.de/sprache-literatur-kultur/medieninformatik/forschung/schwerpunkte/rechtsinformatik/index.html (accessed 16 January 2020); Institut für Rechtsinformatik. https://iri.uni-hannover.de/home.html (accessed 16 January 2020); Servicestelle für Rechtsinformatik. https://www.uni-marburg.de/de/fb01/fachbereich/it_support (accessed 16 January 2020); IT-Recht und Rechtsinformatik https://rewi-grundlagen.uni-graz.at/de/forschen/it-recht-und-rechtsinformatik/ (accessed 16 January 2020).

About the authors

Dr. Michael Jünemann
Partner,
Finance & Financial Regulation

Tel: +49 (0)69 74222 6230
michael.juenemann@twobirds.com

Michael co-heads the international Finance & Financial Regulation practice in of Bird & Bird and is a member of the international Financial sector group steering committee.

Michael mainly advises in the areas of national and international financial and capital markets law, as well as banking, payment services and insurance regulation. He has many years of experience of advising on the implementation and restructuring of transactions, both in Germany and internationally. Michael has corresponded and liaised with lawyers on transactions and restructurings in over 50 jurisdictions worldwide. He advises international and national companies, incumbents and start-ups in all fields of regulatory law and its interdependence with digitalisation. Michael has remarkable knowledge on new and disruptive technologies such as blockchain and distributed ledger, and the legal issues the respective business models face. The scope of support ranges from early steps in the lobbying process of new legislation to the comprehensive application of regulatory law to all sorts of fintech, insuretech, regtech and other related business models with the result of practical solutions for the client. For example, he advised Mastercard on the effective lobbying efforts in the course of new anti-money laundering laws in Germany and Europe. On a regular basis, he gives presentations at conferences regarding the future of banking and the financial services industry. Michael is frequently asked to provide expert opinions and interviews for the daily and specialised press. In this course, he spoke to the UK's Financial Conduct Authority about technology's impact on compliance and regulatory reporting (Regtech) or to German Handelsblatt regarding change on card payments and other payment methods due to PSD II implementation.

Dr. Udo Milkau
Digital Counsellor

udo.milkau@web.de

Udo Milkau is a 'digital dinosaur' with first experiences in digital technology in 1974, many innovation projects, including the first European securities online brokerage in 1995, and working as a Digital Counsellor now. For three decades, he held management positions with the automotive industry, professional services firms and transaction banking, served customers in Asia and Europe, the European banking industry and was Chief Digital Officer, Transaction Banking until 2020. After his academic education in physics, he worked as a research scientist in large collaborations at different European research centres, including CERN, CEA de Saclay and GSI. He was chairman of the European Association of Co-operative Banks (EACB) Digital and Data Working Group, member of the EACB Payment Services Working Group and member of the European Central Bank's Operation Managers Group (ECB OMG). Udo Milkau has published over 100 papers, including payments strategy, digitalisation of banking, risk management/risk culture, digital economies, blockchain and 'law & digitalisation'. He has lectured at Goethe University Frankfurt am Main, Frankfurt School of Finance and Management and WHU — Otto Beisheim School of Management (Vallendar) and currently lectures at Baden-Wuerttemberg Cooperative State University (DHBW in Mosbach, Germany).

twobirds.com

Abu Dhabi & Amsterdam & Beijing & Bratislava & Brussels & Budapest & Copenhagen & Dubai & Dusseldorf & Frankfurt & The Hague & Hamburg & Helsinki & Hong Kong & London & Luxembourg & Lyon & Madrid & Milan & Munich & Paris & Prague & Rome & San Francisco & Shanghai & Singapore & Stockholm & Sydney & Warsaw

Bird & Bird is an international legal practice comprising Bird & Bird LLP and its affiliated and associated businesses.
Bird & Bird LLP is a limited liability partnership, registered in England and Wales with registered number OC340318 and is authorised and regulated by the Solicitors Regulation Authority. Its registered office and principal place of business is at 12 New Fetter Lane, London EC4A 1JP. A list of members of Bird & Bird LLP and of any non-members who are designated as partners, and of their respective professional qualifications, is open to inspection at that address.