«KLAR IST DER AETHER UND DOCH VON UNERGRÜNDLICHER TIEFE» – SMART CONTRACTS ALS INTERDISZIPLINÄRES PROBLEM

Bettina Mielke / Christian Wolff

Dr. iur., Vorsitzende Richterin am Landgericht Regensburg, Lehrbeauftragte an der Universität Regensburg
Kumpfmühler Straße 4, 93047 Regensburg, DE
bettina.mielke@lg-r.bayern.de

Professor, Institut für Information und Medien, Sprache und Kultur, Lehrstuhl für Medieninformatik
Universität Regensburg, 93040 Regensburg, DE
christian.wolff@ur.de, http://mi.ur.de

Schlagworte: ***Smart Contracts, Blockchain, Ethereum, Kryptographie, Legal Tech, Vertragsrecht***

Abstract: *Smart Contracts sind ein bereits vor 20 Jahren publiziertes Konzept, das derzeit durch die Entwicklung Blockchain-basierter kryptographischer Infrastrukturen viel Beachtung erfährt und einen wichtigen Teil der Diskussion um Legal Tech ausmacht. Smart Contracts erlauben es, ausführbare Programme auf der Basis von Krypto-Infrastrukturen für die dynamische Vertragsabwicklung zu nutzen. Wir geben zunächst einen Überblick zum Stand der Technik und der Bandbreite der Anwendungsgebiete und arbeiten nachfolgend die interdisziplinären Fragestellungen zwischen Rechtswissenschaft und Informatik heraus.*

1. Einführung[1]

Das Konzept der Smart Contracts wurde von Nick Szabo in den 1990er Jahre erstmals beschrieben (Szabo 1996, 1997) und erfährt nunmehr – 20 Jahre später – durch die Entwicklung Blockchain-basierter kryptographischer Infrastrukturen im Rahmen der Diskussion um Legal Tech große Beachtung (Mielke/Wolff 2017). Im nachfolgenden Beitrag wollen wir herausarbeiten, dass gerade dieses Thema interessante interdisziplinäre Fragestellungen zwischen Rechtswissenschaft und Informatik aufwirft. Kölvart et al. 2016, 133 führen hierzu aus: «There has been little discussion about smart contracts in relation to contract law. The concept of smart contracting has remained incomprehensible to most lawyers, and programmers tend to perceive it as a solution that replaces traditional contracts and contract law.»

2. Smart Contracts

Der Begriff des Smart Contract steht für die Ausweitung der Anwendungsmöglichkeiten Blockchain-basierter Krypto-Infrastrukturen. Während anfangs vor allem digitale Währungen wie Bitcoin im Mittelpunkt des Interesses standen, werden mittlerweile vielfältige Anwendungen auf der Basis einer Blockchain diskutiert: «A second generation of blockchains are more general-purpose: transactions can record data about any kind of application domain, and can deploy and execute user-defined scripts («smart contracts»). This greatly expands the potential uses for blockchain technology» (Weber et al. 2017, 64). Smart Contracts lassen sich als «zweite Ebene» der Krypto-Infrastrukturen betrachten, die aufbauend auf den Basisdiensten der ersten Ebene (verteilte Speicherung, Verschlüsselung) dazu dient, Daten, (Programm-)Logik und Verhalten für die gemeinsame Nutzung bzw. Zusammenarbeit zu spezifizieren (vgl. Hull 2017, 1).

[1] «Klar ist der Aether und doch von unergründlicher Tiefe» stammt aus Schillers Gedicht *Genialität*, enthalten in den Votivtafeln Goethes und Schillers im Musen-Almanach für das Jahr 1797: Schiller, Friedrich (Hrsg.), Musen-Almanach für das Jahr 1797. Tübingen: J. G. Cottaische Buchhandlung, 1797, S. 173. Online: https://de.wikisource.org/wiki/Tabulae_votivae.

2.1. Übersicht zur Forschungslage

Zur Aufbereitung der aktuellen Forschungslage wurden in einem ersten Schritt unterschiedliche Fachbibliografien bzw. -datenbanken untersucht: Die *ACM Digital Library* (ACM full text collection) für die Informatik, das *Web of Science* als fachübergreifende Bibliografie, *Google Scholar* als freizugängliches Recherchesystem für wissenschaftliche Literatur sowie die deutschsprachige Datenbank *juris* für die Rechtswissenschaft. Die nachfolgende Übersicht gibt eine erste Vorstellung von der derzeitigen Forschungslage bzw. Literatursituation:

Datenbank	**Bezug**	**#Dok «Smart Contracts»**	**#Dok «Blockchain»**	**#Dok «Legal Tech» / «LegalTech»**
ACM Digital Library	Informatik	40 (vollst. Durchsicht)	137	0[2]
juris	Rechtswissenschaft	81 (partielle Durchsicht)	191	187 / 10
Web of Science	Fachübergreifend	21 (vollst. Durchsicht)	108	3 (aus den Jahren 1992 und 2001!)
Google Scholar	Fachübergreifend	ca. 3.300, Durchsicht der ersten 60 Dokumente	21.300	2.310 / 231
Google	Allgemeine Suchmaschine	ca. 468.000 (keine systematische Durchsicht)	ca. 50.700.000	492.000 / 633.000

Tabelle 1: Datenbank-Trefferzahlen für «Smart Contracts» und verwandte Konzepte, Aufruf: 4. Januar 2018

Bei der Durchsicht der Trefferlisten in den verschiedenen Datenbanken lag ein Fokus auf den Aspekten Interdisziplinarität und Anwendungsbezug. Insbesondere das Spektrum der in der Literatur vorgeschlagenen Anwendungen für Smart Contracts sollte möglichst breit erfasst werden, rein technische Artikel ohne Anwendungsbezug wurden nicht berücksichtigt. Nicht systematisch, sondern nur fallweise über Zitierungen in der Aufsatzliteratur wurden Blog-Beiträge, *white paper* und andere Formen der grauen Literatur im offenen Web erfasst.

2.2. Definition von Smart Contracts

Festzuhalten ist, dass für das Konzept der Smart Contracts keine allgemein akzeptierte Definition existiert.[3] Der Begriff des Smart Contract steht letztlich für beliebige ausführbare Programme im Kontext einer Krypto-Infrastruktur wie *Ethereum*. Juristisch relevante Aspekte im Sinne des Vertragsrechts können mit derartigen Programmen und ihrer Ausführung verbunden sein, müssen aber nicht. Warum der *Ethereum*-Erfinder Vitalik Buterin 2014 den Vertragsbegriff als Schlüsselkonzept für *Ethereum* gewählt hat, lässt sich nicht abschließend klären (Buterin 2014, Marino 2016). Eine im juristischen Sinne untechnische Verwendung des Vertragsbegriffs ist in der Informatik allerdings nichts Neues: Bertrand Meyer hat schon in den frühen 1990er Jahren *design by contract* als Grundkonzept des Software Engineering ausgearbeitet und zur Grundlage innovativer Programmiersprachen gemacht (Meyer 1992).

Nick Szabo charakterisiert Smart Contracts wie folgt: «The basic idea of smart contracts is that many kinds of contractual clauses (such as liens, bonding, delineation of property rights, etc.) can be embedded in the hardware and software we deal with, in such a way as to make breach of contract expensive (if desired, sometimes

[2] Dies macht deutlich, dass «LegalTech» / «Legal Tech» offenkundig ein Germanismus ist und im englischen Sprachraum kaum bekannt ist; dort wird eher, aber ebenfalls selten, der Begriff «legal technology» verwendet.

[3] Soweit sie in der Literatur auch als technologische Weiterentwicklung herkömmlicher Formularhandbücher bezeichnet werden (Bormann 2017, 636; Buchholtz 2017, 956), ist dies zumindest missverständlich, da der Smart Contract als in der Blockchain verankertes, unveränderliches Programm gerade nicht den Charakter einer anpassbaren Mustervorlage aufweist und zumindest nach gängiger Anschauung entscheidend ist, dass sich Smart Contracts selbst vollziehen.

prohibitively so) for the breacher» (Szabo 1996, 1). Smart Contracts werden insofern als computergestützte Transaktionsprotokolle verstanden, die vertragliche Regelungen ausführen.[4] Christidis/Devetsikiotis 2016, 2292 betonen die technische Seite der Realisierung: «Smart contracts (self-executing scripts that reside on the blockchain) integrate these concepts and allow for proper, distributed, heavily automated workflows.» Ähnlich sehen dies Kölvart et al. 2016, 134 : «A smart contract is an intelligent agent. In other words, it is a computer program capable of making decisions when certain preconditions are met. The intelligence of an agent depends on the complexity of a transaction it is programmed to perform.» Das Ausführungsmodell von Smart Contracts – Programmcode, der bei Eintreten eines Ereignisses ausgeführt wird – erläutert Paech 2017, 1082: «The term «smart contract» refers to computer code that is designed automatically to execute contractual duties upon the occurrence of a trigger event. The simple example of a vending machine has been cited to explain the concept: upon insertion of a specific type of coin, the computer programme instructs the mechanism of the machine to release the good.»

In der (deutschsprachigen) juristischen Literatur zu Smart Contracts stehen erwartungsgemäß eher rechtliche Aspekte des Konzeptes im Vordergrund. Wagner 2018, 6 führt aus: «Bei ihnen findet sich der «Vertrag» nicht mehr (nur) in menschlicher Sprache, sondern ist als computerisiertes Transaktionsprotokoll geschrieben und kann sich dadurch ganz oder teilweise selbst vollziehen, also etwa eine in ihm vereinbarte Zahlung selbst auslösen». Auch Fries 2017, 2862 betont den Aspekt, dass bei Smart Contracts der Rechtsvollzug bereits im Vertrag selbst angelegt ist (vgl. auch Djazayeri 2016, 2). Kaulartz/Heckmann 2016, 618 machen folgende wesentliche Merkmale von Smart Contracts aus: Ein digital prüfbares Ereignis, ein Programmcode, welcher das Ereignis verarbeitet und eine rechtlich relevante Handlung, welche auf Grundlage des Ereignisses ausgeführt wird.

Der Begriff der Smart Contracts ist dabei insofern irreführend, da sie zumindest derzeit weder *smart* sind – bei den bisherigen Beispielen handelt es sich meist um triviale Mini-Programme, die aus wenigen Wenn-Dann-Regeln bestehen[5] –, noch, dass es sich bei ihnen um Verträge im rechtlichen Sinn handeln muss. Zu unterscheiden ist nämlich zwischen dem eigentlichen Vertragsschluss und der tatsächlichen Durchführung eines Smart Contract (Djazayen 2016, 4; Kaulartz/Heckmann 2016, 621). Inwieweit es nicht völlig außerhalb der Vorstellungskraft liegt, dass Computerprogramme selbst künftig einmal Willenserklärungen abgeben können (Paulus/Matzke 2017, 772), soll hier nicht beurteilt werden.

Neben der Vereinfachung im Rahmen der Vertragsvollziehung wird zudem der Aspekt der Desintermediation betont, also der durch Smart Contracts mögliche Wegfall von Intermediären (Banken, Versicherungen, Anwälten, Notaren etc., Kölvart et al. 2016, 134).

2.3. Ethereum als technisches Realisierungsbeispiel

Die bekannteste Realisierung einer für den Einsatz von Smart Contracts geeigneten Krypto-Infrastruktur wurde von dem *Ethereum*-Projekt entwickelt (Buterin 2014, Dannen 2017, Müller 2017 und https://www.ethereum.org/). Dabei handelt es sich um eine verteilte Krypto-Infrastruktur auf Blockchain-Basis, die anders als bei Bitcoin nicht nur als verteiltes Konto (*distributed ledger*) statisch Informationen speichern kann, sondern auch Zustände und damit für die Ausführung von Code geeignet ist. Zustände werden in *Ethereum* über Konten (*accounts*) realisiert, die jeweils durch vier Merkmale (Datenfelder) charakterisiert sind (Buterin 2014, 13):

1. Ein eindeutiger Zähler (*nonce – number used once*), mit dessen Hilfe jede Transaktion eindeutig identifiziert werden kann,

[4] Die Szabo in der Literatur vielfach zugeschriebene, aber nicht direkt verifizierbare Definition «a computerized transaction protocol that executes the terms of a contract» bringt dies zum Ausdruck, vgl. etwa Christidis/Devetsikiotis 2016, 2296.

[5] Vgl. dazu die Beispiele auf https://ethereum.org und die über https://www.stateofthedapps.com/ erreichbaren Dokumentationen von Beispielanwendungen.

2. der aktuelle Kontostand in *Ether* – der für Ethereum verfügbaren Basiswährung,
3. der ausführbare Code auf dem Konto, soweit vorhanden, und
4. der Speicherbereich des Kontos, der zunächst leer ist.

Man kann grundsätzlich beliebigen ausführbaren Code in *Ethereum* hinterlegen. Dieser wird dann von den beteiligten Netzwerkknoten ausgeführt. Das Ausführungsmodell sieht vor, dass eine Ethereum-Anwendung auf ein externes Ereignis (Event) reagiert, dann Code ausführt und damit konkrete Effekte erzielt. Dies stellt die technische Basis für Smart Contracts dar.

Um entsprechenden Code zu generieren, verfügt *Ethereum* über eine eigens entwickelte Programmiersprache, *Solidity*. Diese ist an die Syntax der Programmiersprache Java angelehnt und für sie gilt die Annahme der Turing-Vollständigkeit, was bedeutet, dass den Realisierungsmöglichkeiten für ausführbaren Code keine grundsätzlichen Grenzen gesetzt sind.[6] Die auf der Basis von *Ethereum* realisierten Anwendungen werden als *distributed apps* oder *Dapps* (Schreibweise mit dem altenglischen Buchstaben *Eth*: Ðapps) bezeichnet. Für sie gilt, dass sie genau so wie programmiert, nachvollziehbar und ohne externe Eingriffsmöglichkeit, sowie durch die Verteilung im Ethereum-Netz ausfallgeschützt ausgeführt werden (Müller 2017, 608 f.).

3. Anwendungsgebiete für Smart Contracts

Mittlerweile sind vielfältige denkbare Anwendungsszenarien für Smart Contracts benannt worden. Mit Getränke- und Geldautomaten bzw. Kassen (POS – *point of sale*) als Formen maschinengestützter Vertragsabwicklung zeigt bereits Szabo 1996, 1 f. einfache Beispiele: «A canonical real-life example, which we might consider to be the primitive ancestor of smart contracts, is the humble vending machine.» Sowohl in der Informatik-Literatur als auch in juristischen Fachaufsätzen finden sich zahlreiche Beispiele, von denen allerdings viele auf künftige Anwendungs*möglichkeiten* verweisen. Problematisch sind dabei sowohl der unklare und von verschiedenen Communities offensichtlich unterschiedlich interpretierte Begriff des Smart Contract als auch die nicht immer klare Abgrenzung zwischen dem, was bereits eine einfache Blockchain (ohne eingebettete Programme) zu leisten vermag, und den zusätzlichen Möglichkeiten, die durch Smart Contracts entstehen. Eine detaillierte Bewertung aller Anwendungsvorschläge nach technischen und rechtlichen Kriterien würde zu weit führen, wir beschränken uns daher nachfolgend darauf, das Anwendungsspektrum aufzuzeigen und geben abschließend einige Beispiele für bereits auf Ethereum realisierte verteilte Smart Contract-Anwendungen.

Die nachfolgend angeführten Anwendungsgebiete gehören zu den in der Literatur am häufigsten genannten. Teilweise sind auch mehrere Zuordnungen denkbar (per Smart Contract gesteuertes Wohnungsschloss als Beispiel für *Internet of Things*; gleichzeitig Anwendung eines *Legal Tech*-Dienstes, der die passenden Vertragsdokumente erzeugt).

- *Legal Tech* – juristische Arbeitsfelder können dort von Smart Contracts profitieren, wo mit einer vertraglichen Regelung eine überprüfbare und automatisierbare Interaktion mit digitalen Gütern erfolgt. Als möglicher Anwendungskontext wird das Gesellschaftsrecht genannt (Zahlungspflicht der Gesellschafter für die Stammeinlage bei GmbH-Gründung, Eintragung im Handelsregister erst nach Einzahlungsbestätigung der Bank, vgl. Bormann 2017, 635).
- *FinTech / InsurTech / RegTech* – zahlreiche Anwendungen von Smart Contracts sind in den Bereichen Finanzdienstleistungen (z.B. Börsenhandel, Wagner 2018, 21), Versicherungen (fallweiser Kauf von Versicherungsleistungen, Vo et al. 2017) oder staatliche Regulierung (auch Abgaben und Zölle, Magazzeni et al. 2017, 52) benannt worden.

[6] Etwas technischer ausgedrückt: Es lassen sich mit dieser Sprache alle Programme realisieren, die auch von einer universellen Turing-Maschine ausgeführt werden könnten, unter der (praktisch nicht gegebenen) Annahme nicht beschränkten Speicherplatzes.

- *Internet of Things* – hier finden sich zahlreiche anschauliche Beispiele (Überblick: Christidis/ Devetsikiotis 2016) wie «intelligente» Wohnungsschlösser, die sich nach bezahlter Miete öffnen lassen, oder Smart Contracts als Grundlage des Datenhandels im Internet of Things (Missier et al. 2017). Nissen et al. 2017 untersuchen, inwieweit Smart Contracts für Alltagsgegenstände genutzt werden können.
- *Wissenschaft und Forschung* – Bell et al. 2017 schlagen vor, entscheidende Merkmale von Blockchain-Infrastrukturen für die Absicherung wissenschaftlicher Forschung zu nutzen. Sie wollen Smart Contracts realisieren, die, in die Blockchain eingebettet, unter bestimmten Voraussetzungen (zum Beispiel der Abschluss eines Experiments) den Zugriff auf durch die Blockchain verifizierbare Daten erlauben. (Bell et al. 2017, 14)
- *Datenschutz, Urheberrecht*, intellectual property rights (*IPR*) – Neisse et al. 2017 diskutieren die Nutzbarkeit von Smart Contracts unter dem Gesichtspunkt des neuen europäischen Datenschutzrechts: Sie stellen prototypische Anwendungen vor, die Smart Contracts nutzen, um die Herkunft von Daten nachzuverfolgen (*data provenance tracking*). Meitinger 2017 und Morabito 2017, Kap. 6 betonen die Nutzbarkeit von Smart Contracts im Bereich des Urheberrechts und der Lizenzierung urheberrechtlich geschützter Inhalte, was an Technologien aus dem Bereich des *digital rights management* erinnert.
- *Entwicklungspolitik* – Smart Contracts werden auch in Zusammenhang mit den rechtlichen und ökonomischen Problemen der südlichen Hemisphäre diskutiert (Kshetri 2017). Potenzial wird in dezentralen und ohne Vertrauensinstanz funktionierenden Krypto-Infrastrukturen dort gesehen, wo Rechtssysteme nicht hinreichend ausgebaut oder nicht durchsetzbar erscheinen (Kshetri 2017, 1712).
- *Kriminelle Smart Contracts* (criminal smart contracts) – Die Schattenseite der Nutzungsmöglichkeiten von Smart Contracts nehmen Juels et al. 2016 in den Blick und diskutieren, welche Straftaten sich auf der Basis von Krypto-Infrastrukturen mit Smart Contracts realisieren lassen (u. a. illegale Marktplätze, Erpressung mit digitalen Medien, Geldwäsche).

Eine eigene Plattform (https://www.stateofthedapps.com/) dokumentiert den derzeitigen Entwicklungsstand verteilter Anwendungen (Đapps) auf der *Ethereum*-Plattform. Dort fanden sich am 7. Januar 2018 930 Beispielanwendungen. Insgesamt zeigt sich ein sehr weit gefasstes Spektrum (u.a. auch Spiele / Gewinnspiele / Lotterien, schneeballsystemartige Strukturen (*Ponzi Schemes*), Social Media Plattformen, Blockchain-basierte Messenger). Tabelle 2 führt Beispiele auf, bei denen vertragsrechtliche Aspekte offenkundig bzw. vertragliche Regelungen erforderlich sind (Auswahlkriterien: Status *live* auf *StateoftheDapps*, Anwendungswebsite erreichbar, expliziter Bezug zu Smart Contracts):

Name	**Beschreibung**	**URL**
Ethereum Alarm Clock	Zeitgesteuerte Ausführung von (Smart) Contracts auf *Ethereum*	http://www.ethereum-alarm-clock.com/
Giveth. Bulding the Future of Giving	Spenden-Plattform, in der mit Hilfe des *LiquidPledging Smart Contracts* Spendenmittel für Empfängerkampagnen bereitgestellt werden	https://giveth.io/#communities
HelloSugoi. Create and purchase tickets for events	Ticketing-Anwendung für *live events* (Konzertkartenverkauf etc.)	https://www.stateofthedapps.com/dapps/hellosugoi
LegalThings One – A fair legal system for everyone	Weiterentwicklung von Smart Contracts zu Live Contracts (mit mehr Felxibiliutät)	http://livecontracts.io/
SplitCoin	Aufteilung der Ethereum-Währung *Ether* für mehrere Beteiligte	http://app.splitcoin.io/
WeiCards	Verkauf digitaler Karten, Smart Contracts können eingesehen werden.	https://willdn.github.io/weicards/#/

Tabelle 2: Ethereum-Đapps mit Smart Contracts

4. Probleme im Umfeld von Smart Contracts

Smart Contracts bringen vielfältige technische und rechtliche Probleme und Herausforderungen mit sich, die sich teilweise nur schwer trennen lassen, wie etwa die Frage nach der Abbildung zwischen rechtlicher (Vertrags-)Modellierung und der Umsetzung im ausführbaren Smart Contract.

4.1. Technische und konzeptuelle Herausforderungen

Hull 2017 stellt Querbezüge zwischen Smart Contracts und einschlägigen Themen der Wirtschaftsinformatik her (betriebliche Informationssysteme, Prozessmodellierung, event-based systems) und benennt eine Reihe von Entwicklungsproblemen in seinem Forschungsüberblick:

- Änderbarkeit und Dynamik von Smart Contracts (*across domains, across time*, Hull 2017, 3)
- Bedarf an geeigneten Modellierungssprachen und Modularität
- Verifikation und formale Korrektheit von Smart Contracts (Hull 2017, 3)

Seijas et al. 2016 geben einen Überblick zu den für die Kodierung von Smart Contracts verfügbaren Skriptsprachen, wobei die im Kontext von *Ethereum* entwickelte Python-nahe Sprache *Serpent* sowie *Solidity* zu den bekanntesten Beispielen zählen (Dannen 2017, 72).

Das Verteilungskonzept der derzeit genutzten Krypto-Infrastrukturen sieht vor, dass Daten auf allen beteiligten Knoten repliziert bzw. Programme tatsächlich auf allen beteiligten Knoten ausgeführt werden (*Ethereum*). Dies setzt der Skalierbarkeit solcher Infrastrukturen klare Grenzen. Kürzlich (Anfang Januar 2018) hat daher das *Ethereum*-Projekt ein eigenes Forschungs- und Stipendienprogramm zur Verbesserung der Skalierbarkeit ausgerufen (Buterin 2018). Eine weitere Einschränkung ist das Konzept des *proof-of-work,* wie es bei Bitcoin und *Ethereum* verwendet wird: Danach kommt derjenige für bestimmte Aktionen in einer Krypto-Infrastruktur zum Zug, der entsprechend viel Rechenzeit investiert hat (Konsens durch investierte Arbeit/Energie). Da seit Beginn der Ausgabe von Bitcoin die aufzuwendende Rechenzeit pro Währungseinheit dramatisch zugenommen hat, dürfte allein das Problem der gewachsenen Energiekosten[7] die Weiterführung dieses Prinzips der Krypto-Infrastrukturen der ersten Generation relativ bald ad absurdum führen. Man wird zukünftig Lösungen finden müssen, die weiterhin eine Nachvollziehbarkeit, unter anderem durch Informationsverteilung, realisieren, ohne dabei entsprechende (Energie-)Kosten zu beanspruchen.

4.2. Rechtsfragen

Auf das Grundproblem des unglücklich gewählten Begriffs Smart Contract weisen Schrey/Thalhofer 2017, 1431 hin: «Der Begriff «Smart Contract» legt fälschlicherweise nahe, dass es sich bei diesen immer um einen Vertrag im Sinne des Zivilrechts handelt, was im Blockchain-Umfeld gerade nicht stets der Fall ist. Nach deutschem Recht kommen Verträge durch die Inhalte von Antrag (§ 145 BGB) und Annahme (§ 147 BGB) zustande.» Wie bereits ausgeführt, ist die menschliche Mitwirkung bei Smart Contracts nicht obsolet, da der automatisierten Vertragsvollziehung eine «generelle Einwilligung desjenigen, in dessen Namen gehandelt wird, vorgeschaltet» ist (Börding et Al. 2017, 138) bzw. «zwischen dem eigentlichen Vertragsschluss und der tatsächlichen Durchführung eines Smart Contract unterschieden werden» sollte (Djazayeri 2016, 4).

Folgt man dem Ausspruch «*code is law*» («Der Code ist das Gesetz», Lessig 2001, 24) Lawrence Lessigs in dem Sinn, dass der exakte und verifizierbare formale Code eines Smart Contract die Vertragsbedingungen präzise beschreibt, so kann man zwar davon ausgehen, dass solche auf dem Rechner implementierten Verträge ein hohes Maß an Rechtssicherheit bieten (Börding et al., 138) – sie müssen weder ausgelegt noch interpretiert werden. Die Kehrseite davon dürfte aber bis auf weiteres die Beschränkung auf vergleichsweise einfache

[7] Bitcoin-Erzeugung läuft heute weitestgehend in spezialisierten Rechnerfarmen ab, die in Ländern mit niedrigen Energiekosten wie der Inneren Mongolei (bzw. China) betrieben werden, mit stetig wachsendem Energiebedarf, vgl. Peck 2017; siehe auch Bormann 2017, 638.

Sachverhalte darstellen (alles andere als Smart gewissermaßen), da nur wenige Verträge ohne interpretationsbedürftige Begrifflichkeiten auskommen (Müller 2017, 610).

Im Kontext von Smart Contracts ergibt sich insgesamt eine Vielzahl rechtlich ungelöster Fragen:

- Neben der Frage einer notwendigen Interpretation oder Auslegung von Smart Contracts besteht die Notwendigkeit zur Einhaltung der Grundsätze des Vertragsrechts; der *code* ist damit nicht das einzige *law*, sondern im Zusammenhang mit geltendem Recht zu lesen (Kaulartz/Heckmann 2016, 623).
- Programmiersprachen sind zwar als Vertragssprachen frei wählbar (Grundsatz der freien Sprachenwahl, der aus dem Prinzip der Vertragsfreiheit folgt, Djazayeri 2016, 4, Kaulartz/Heckmann 2017, 621), nach § 184 GVG ist aber die Gerichtssprache Deutsch (Börding et al. 2017, 139), so dass spätestens im Falle einer gerichtlichen Überprüfung das Problem einer notwendigen Versprachlichung auftritt.
- Da davon auszugehen ist, dass Smart Contracts selten individuell ausgehandelt werden, ist das Recht der Allgemeinen Geschäftsbedingungen zu beachten (Kaulartz/Heckmann 2017, 624). Da beispielsweise aus § 305 Abs. 2 Nr. 2 BGB folgt, dass die Vertragspartei in zumutbarer Weise vom Vertrag Kenntnis nehmen kann, erscheint jedenfalls bei Verträgen mit Verbrauchern der Verweis auf den Quellcode einer Programmiersprache insoweit problematisch. Zudem sind einzelne Klauseln einer Inhaltskontrolle zu unterziehen (Kaulartz/Heckmann 2017, 622).
- Der Umstand, dass diese Art Vertrag, einmal auf der Blockchain festgeschrieben, nicht ohne Weiteres geändert werden kann und seine Ausführung auch nicht aufgehalten werden kann (Erbguth 2016, 157), zur Korrektur bei Programmierfehlern oder etwa im Fall der Anfechtung (z.B. wegen Unkenntnis der Programmiersprache), erscheint mit Grundprinzipien des Vertragsrechts nicht vereinbar (Kilian 2017, 3050; Schrey/Thalhofer 2017, 1435).
- Es bestehen vielfältige und komplexe Haftungsfragen, insbesondere stellt sich die Frage der Zurechnung bei hohem Automatisierungsgrad (Börding et al. 2016, 139 f.).

Kaulartz/Heckmann 2017, 624 nennen zahlreiche weitere rechtliche Problemfelder in Zusammenhang mit Smart Contracts, von der Notwendigkeit der Regulierung bis zu Fragen der Rechtsdurchsetzung. Ganz allgemein ist zu fragen, ob die Nutzung von Smart Contracts juristische Expertise gerade *nicht* überflüssig macht, sondern neue Formen juristischen Handelns dadurch erst entstehen – mehr Arbeit für Juristen durch Smart Contracts statt weniger, neue Beratungsdienstleistungen statt Disintermediation (Voigt 2001)?[8]

4.3. Abbildung zwischen rechtlicher und technischer Ebene

Für den Erfolg nicht-trivialer Smart Contracts dürfte die Abbildung rechtlicher Zusammenhänge auf die Ebene der Programmierung / Kodierung entscheidend sein – ungeklärt ist derzeit, wie eine solche Abbildung erfolgen kann und wer für die Validität und Korrektheit dieser Abbildung (nicht der Kodierung, diese wird durch die kryptographischen Eigenschaften der Blockchain gesichert) eintritt und wie sichergestellt sein soll, dass die verteilte Infrastruktur, die derartige kryptographische Dienste bereithält, selbst zuverlässig und vertrauenswürdig ist. Erste spektakuläre Straftaten bei der Nutzung der Krypto-Infrastrukturen (Luu et al. 2017, Wieczener 2017) machen deutlich, dass hier nicht nur weitere konzeptuelle Grundlagenforschung zu betreiben ist, sondern dass auch die tatsächlich entwickelten Infrastrukturen noch einen längeren Reifeprozess vor sich haben dürften. Die ungeklärte Abbildung zwischen rechtlicher und technischer Ebene rührt an ein altes Problem der Rechtsinformatik: Die Frage nach standardisierten Beschreibungssprachen für rechtliche Zusammenhän-

[8] So sieht es auch Wagner 2018, 23: «Smart Contracts haben hinsichtlich ihres Veränderungspotenzials eine besondere Stellung. Sie werden in erster Linie Gegenstand der internen oder externen anwaltlichen Beratung sein. Dennoch berühren sie die Art und Weise der juristischen Tätigkeit als solche, nämlich dort, wo es um die Vertragsgestaltung geht. Denn zur Vertragsgestaltung in menschlicher Sprache, welche im Grundsatz immer den Ausgangspunkt bildet, tritt als zusätzlicher Schritt deren Abbildung in Maschinensprache (Software) hinzu. Insoweit wird der übergeordnete Arbeitsablauf um zusätzliche Arbeitsschritte erweitert.»

ge. Im vorliegenden Fall könnte das Problem sich als noch komplexer erweisen, da nicht nur die rechtliche Beschreibungsebene, sondern auch die technische Dimension weiter standardisiert werden müsste.

Ein bereits seit den 1990er Jahren bekanntes Beispiel für die Formalisierung von Kommunikation stellt die Agent Communication Language dar (FIPA ACL), die eine Menge an Kommunikationsprimitiven definiert und formalisiert. Bisherige Ansätze zur Formalisierung komplexer juristischer Zusammenhänge sind über das Stadium der Grundlagenforschung nicht hinausgekommen (Mielke/Wolff 2017, Rz. 44 und Fn. 110 ff.).

Die Frage der Verständlichkeit und Nachvollziehbarkeit von Smart Contracts stellt sich nicht nur für den Juristen, der den Bezug zur rechtlichen Ausgestaltung herstellen muss, sondern auch für den Endanwender, der typischerweise juristischer (und technischer) Laie ist: Mit welchen Mechanismen kann verständlich kommuniziert werden, was in einem konkreten Smart Contract codiert ist und welche Aktionen unter welchen Bedingungen ausgeführt werden sollen? Es geht unter anderem um die Ausgestaltung angemessener Benutzerschnittstellen und Informationsflüsse für den Anwender (Eskandari et al. 2015).

In der Literatur finden sich einfache Beispiele dafür, wie sich ausführbare Zusammenhänge als Smart Contract formalisierten / kodieren / programmieren lassen: Diese sind entweder in einer Art neutralem Pseudocode oder in mehr oder weniger beliebigen Programmiersprachen verfasst.[9] Tatsächlich ist aber aus Sicht der Informatik zu fragen, welche Programmiersprachen eingesetzt werden können, um ein größtmögliches Maß an Verifizierbarkeit des Codierten sicherstellen zu können. Erste Studien machen dazu Vorschläge (O'Connor 2017).

5. Ausblick

Werbach/Cornell 2017 machen deutlich, wie divergent derzeit noch die allgemeinen Einschätzungen hinsichtlich der Auswirkungen der Einführung von Smart Contracts sind: Diese reichen tatsächlich einerseits von der Annahme, dass autonome Smart Contracts das Rechtssystem, insbesondere das Vertragsrecht überflüssig machen, bis zur Unterstellung, dass es sich bei Blockchain-basierten Krypto-Infrastrukturen letztlich um nichts anderes als ein betrügerisches Schneeballsystem handele (Werbach/Cornell, 316 f.). Sie machen deutlich, dass eine Differenzierung der Aufgaben zwischen Vertragsrecht und den Möglichkeiten von Smart Contracts zwingend erforderlich ist und betonen, dass auch Krypto-Infrastrukturen nicht ohne einen rechtlichen Rahmen funktionieren können.

Nicht zu Unrecht arbeitet Marino 2016, 1 die universelle Anwendbarkeit von Smart Contracts heraus: «Should the Ethereum community stop using the words «contract» and «smart contract» to refer to, well, *everything*?»

Ungeachtet der vielfältigen technischen und rechtlichen Herausforderungen steht zu erwarten, dass für verifizierbar auf der Blockchain ausgeführte Programme nicht nur in Zusammenhang mit digitalen Gütern ein großes Potential besteht, wenn auch nicht im Sinne von «intelligenten» oder «autonomen», die juristische Vertragsgestaltung überflüssig machenden, Verträgen und wohl ebenfalls nicht auf der Basis der heutigen teuren, langsamen und unflexiblen Krypto-Infrastrukturen.

6. Literatur

Bell, Jonathan/LaToza, Thomas D./Baldmitsi, Foteini/Stavrou, Angelos (2017), Advancing Open Science with Version Control and Blockchains. In: Proceedings of the 12th International Workshop on Software Engineering for Science, IEEE Press, Piscataway, NJ, S. 13–14.

Börding, Andreas/Jülicher, Tim/Röttgen, Charlotte/Schönfeld, Max von (2017), Neue Herausforderungen der Digitalisierung für das deutsche Zivilrecht, Computer und Recht 2017, S. 134–140.

Bormann, Jens (2017), Die digitalisierte GmbH, ZGR 2017, S. 621–647.

[9] Die zahlreichen Codebeispiele für Smart Contracts in der Literatur sind hinsichtlich ihrer technischen Realisierung arbiträr, d. h. sie verwenden beliebige Programmiersprachen. Ein Bezug zur konkreten rechtlichen Ausgestaltung fehlt üblicherweise völlig (vgl. dazu Kolvärt et al. 2016).

Buchholtz, Gabriele (2017), Legal Tech. Chancen und Risiken der digitalen Rechtsanwendung, JuS 2017, S. 955–960.

Buterin, Vitalik (2014), A Next-generation Smart Contract and Decentralized Application Platform. White Paper, Ethereum Project, https://www.weusecoins.com/assets/pdf/library/Ethereum_white_paper-a_next_generation_smart_contract_and_decentralized_application_platform-vitalik-buterin.pdf.

Buterin, Vitalik (2018), Ethereum Scalability Research and Development Subsidy Programs, Ethereum Blog, edited by Ethereum Project, 2. Januar 2018, https://blog.ethereum.org/2018/01/02/ethereum-scalability-research-development-subsidy-programs/.

Christidis, Konstantinos/Devetsikiotis, Michael (2016), Blockchains and Smart Contracts for the Internet of Things, IEEE Access, 4, S. 2292–2303.

Dannen, Chris (2017), Introducing Ethereum and Solidity: Foundations of Cryptocurrency and Blockchain Programming for Beginners, Apress, Berkeley, CA.

Djazayeri, Alexander (2016), Rechtliche Herausforderungen durch Smart Contracts, juris PraxisReport Bank- und Kapitalmarktrecht (jurisPR-BKR) 12/2016, S. 1–9.

Drescher, Daniel (2017), Blockchain Basics. A Non-technical Introduction in 25 Steps, Apress, Berkeley, CA.

Erbguth, Jörn (2016), Lösung Blockchain-basierter Konflikte. In: Schweighofer, Erich/Kummer, Franz/Hötzendorfer, Walter/Sorge, Christoph (Hrsg.), Trends und Communities der Rechtsinformatik, Tagungsband des 20. Internationalen Rechtsinformatik-Symposions IRIS 2017, Österreichische Computer Gesellschaft, Wien, S. 155–160.

Eskandari, Shayan/Clark, Jeremy/Barrera, David/Stobert, Elizabeth (2015), A First Look at the Usability of Bitcoin Key Management. In: Workshop on Usable Security (USEC 2015), San Diego, CA.

Fries, Martin (2016), PayPal Law und Legal Tech – Was macht die Digitalisierung mit dem Privatrecht?, NJW 2016, S. 2860–2865.

Ibba, Simona/Pinna, Andrea/Seu, Matteo/Eros Pani, Filippo (2017), CitySense: Blockchain-oriented Smart Cities. In: Proceedings of the XP2017 Scientific Workshops, New York, ACM, S. 1–5.

Hull, Richard (2017), Blockchain: Distributed Event-based Processing in a Data-Centric World: Extended Abstract. In: Proceedings of the 11th ACM International Conference on Distributed and Event-based Systems, New York, ACM, S. 1–5.

Juels, Ari/Kosba, Ahmed/Shi, Elaine (2016), The Ring of Gyges: Investigating the Future of Criminal Smart Contracts. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, New York, ACM, S. 283–295.

Kaulartz, Markus/Heckmann, Jörn (2016), Smart Contracts – Anwendungen der Blockchain-Technologie, CR 2016, S. 618–624.

Kilian, Matthias (2017), Die Zukunft der Juristen. Weniger, anders, weiblicher, spezialisierter, alternativer – und entbehrlicher?, NJW 2017, S. 3043–3050.

Kölvart, Merit/Poola, Margus/Rull, Addi (2016), Smart Contracts. In: Kerikmäe, Tanel/Rull, Addi (Hrsg.), The Future of Law and eTechnologies, Springer, Cham, CH, et al., S. 133–147.

Kshetri, Nir (2017), Will Blockchain Emerge as a Tool to Break the Poverty Chain in the Global South?, Third World Quarterly, 38, S. 1710–1732.

Lessig, Lawrence (2001), Code und andere Gesetze des Cyberspace, Berlin Verlag, Berlin, 2001.

Luu, Loi/Chu, Duc-Hiep/Olickel, Hrishi/Saxena, Prateek/Hobor, Aquinas (2016), Making Smart Contracts Smarter. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, ACM, New York, S. 254–269.

Magazzeni, Daniele/McBurney, Peter/Nash, William (2017), Validation and Verification of Smart Contracts: A Research Agenda, IEEE Computer, 50(9), S. 50–57.

Marino, Bill (2016), Unpacking the Term «Smart Contract». The Word «Contract» and Ethereum, ConsenSys Blog, 10. Februar 2016, https://medium.com/@ConsenSys/unpacking-the-term-smart-contract-e63238f7db65.

Meitinger, Thomas Heinz (2017), Smart Contracts, Informatik-Spektrum, 40(4), S. 371–375.

Meyer, Bertrand (1992), Applying «Design by Contract». IEEE Computer 25(10), S. 40–51.

Mielke, Bettina/Wolff, Christian (2017), E-Justice, Justiz 3.0 und Legal Tech – eine Analyse, in: Jusletter IT 18. Mai 2017.

Missier, Paolo/Bajoudah, Shaimaa/Capossele, Angelo/Gaglione, Andrea/Nati, Michele (2017), Mind My Value: a Decentralized Infrastructure for Fair and Trusted IoT Data Trading. In: Proceedings of the Seventh International Conference on the Internet of Things, ACM, New York, S. 1–8.

Morabito, Vincenzo (2017), Business Innovation through Blockchain. Springer, Cham/CH.

Nissen, Bettina/Symons, Kate/Tallyn, Ella/Speed, Chris/Maxwell, Deborah/Vines, John (2017), New Value Transactions: Understanding and Designing for Distributed Autonomous Organisations. In: Proceedings of the 2017 ACM Conference Companion Publication on Designing Interactive Systems, ACM, New York, S. 352–355.

O'Connor, Russell (2017), Simplicity: A New Language for Blockchains. In: Proceedings of the 2017 Workshop on Programming Languages and Analysis for Security (PLAS '17), ACM, New York, S. 107–120.

Paech, Pillipp (2017), The Governance of Blockchain Financial Networks, Modern Law Review, 80, S. 1073–1110.

Peck, Morgan E. (2017), The Bitcoin Mines of China, IEEE Spectrum 54(10), S. 46–53.

Schrey, Joachim/Thalhofer, Thomas (2017), Rechtliche Aspekte der Blockchain, NJW 2017, S. 1431–1436.

Seijas, Pablo Lamela/Thompson, Simon J./McAdams, Darryl (2016), Scripting Smart Contracts for Distributed Ledger Technology, IACR Cryptology ePrint Archive, 2016, S. 1156.

Szabo, Nick (1996), Smart contracts: building blocks for digital markets. Report, http://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/smart_contracts_2.html.

Szabo, Nick (1997), Formalizing and Securing Relationships on Public Networks, First Monday, 2(9).

Vo, Hoang Tam/Mehedy, Lenin/Mohania, Mukesh/Abebe, Ermyas (2017), Blockchain-based Data Management and Analytics for Micro-insurance Applications. In: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, ACM, New York, S. 2539–2542.

Voigt, Kai-Ingo (2001), Desintermediation im B2B-Bereich — Perspektiven aus Sicht der Produzenten. In: Albach, Horst/Wildemann, Horst (Hrsg.), E-Business Management mit E-Technologien, Gabler Verlag, Wiesbaden, S. 53–72.

Wagner, Jens (2018), Legal Tech und Legal Robots. Der Wandel im Rechtsmarkt durch neue Technologien und künstliche Intelligenz, Springer Fachmedien, Wiesbaden.

Werbach, Kevin/Cornell, Nicolas (2017), CONTRACTS EX MACHINA, Duke Law Journal, 67, S. 313–382.

Weber, Ingo/Gramoli, Vincent/Ponomarev, Alex/Staples, Mark/Holz, Ralph/Tran, An Binh/Rimba, Paul (2017), On Availability for Blockchain-based Systems. In: Reliable Distributed Systems (SRDS), 2017 IEEE 36th Symposium on, IEEE, Piscataway, NJ, S. 64–73.

Wieczner, Jen (2017), The 21st Century Bank Robbery, Fortune 176(3), S. 34–41.