

«CODE IS LAW» ABER «IS LAW CODE»?

Felix Gantner

DI Dr., infolex Rechtsinformatik
Bei der Kapelle 7, 3592 Röhrenbach, AT
gantner@infolex.at; <http://www.infolex.at>

Schlagnote: *Smart Contracts, Rechtssetzung, Blockchain, regulation by technology, Rechtssprache, formale Sprachen, Ethereum*

Abstract: *Im Bereich des Internets, aber auch der Softwareentwicklung beschreibt die Aussage «Code is Law» die Tatsache, dass über Programmcode das Verhalten und die Handlungsmöglichkeiten von Benutzern reguliert wird. Smart Contracts gehen den umgekehrten Weg: Rechtsnormen werden als Programmcode dargestellt, der für die Vertragserfüllung auch tatsächlich ausgeführt wird. Werden Rechtsnormen als tatsächlich ausführbar Code beschrieben, so hat dies auch Rückwirkungen auf die Rechtssprache und Gesetzgebungstechnik.*

1. Code is Law

1.1. Ursprung

Die von LAWRENCE LESSIG geprägte Wendung *Code is Law*¹ war ursprünglich eine prägnante Zusammenfassung der Tatsache, dass im Cyberspace – zum Zeitpunkt des Erscheinens seines Buchs ein neues Phänomen – zu den bis dahin bekannten Regulierungskräften²

- Markt
- Architektur
- soziale Normen
- Recht

eine weitere hinzugekommen ist: Code, also Software, die in der digitalen Welt faktisch Verhalten steuert.

Er stellt aber ausdrücklich fest, dass *Code is Law* nicht bedeutet, dass Software und Rechtsvorschriften bzw. -normen dasselbe oder gleichzusetzen sind.³ Vielmehr soll diese prägnante Formulierung darauf hinweisen, dass Code ähnlich wie die anderen Kräfte Verhalten reguliert⁴: «Das Gesetz reguliert, indem es Möglichkeiten, die der Einzelne nutzen kann, zulässt oder einschränkt, um ihn zu einem bestimmten Verhalten zu bewegen. Der Code tut dasselbe.»⁵

In der Folge zeigte sich, dass tatsächlich immer häufiger das Verhalten von Benutzern durch Code bzw. technische Architekturen gesteuert wird und *regulation by technology* entweder als Ergänzung oder vielfach auch in Konkurrenz zur Rechtsordnung eingesetzt wird.⁶ Zahlreiche Beispiele sind bekannt, bei denen Unternehmen

¹ LESSIG 2001; vgl. auch LESSIG 2006 und LESSIG 2000.

² LESSIG 2001, S. 161.

³ LESSIG 2001, S. 383: «Der Code ist nicht Gesetz, so wenig wie die Konstruktion eines Flugzeugs Gesetz ist.» Unverändert LESSIG 2006, S. 324.

⁴ WU 2003, S. 687: «The idea is, that programmers make choices that constrain online capability, and that such choices are regulatory in their effects.»

⁵ LESSIG 2001, S. 383.

⁶ Vgl. LENK 2016, S. 231 f.

ihre – vermeintlichen – Rechte lieber über Code und Produktgestaltung, als über den Rechtsweg durchzusetzen versuchten.⁷

1.2. ... und Neuinterpretation

Zusätzlich verselbständigte sich die prägnante Formulierung und wurde zu einem Schlagwort, das der Rechtfertigung für ungezügelter *regulation by technology* diene. Der kritische Ansatz von LAWRENCE LESSIG⁸ ging verloren und aus *Code is Law* wurde die Formel *Code = Law*.

Durch die Gleichsetzung von *Code* und *Law* kam es u.a. zu folgenden zwei Entwicklungen:

Zum einen wurde aus dem Software-Entwickler, also dem Erzeuger des *Codes*, ein Gesetzgeber, ein Erzeugen von *Law*. Da aber nach Ansicht von manchen Verfechtern der Idee, dass der Cyberspace ein rechtsfreier Raum ist, *Code* das einzige zulässige bzw. mögliche Regulationsmittel ist, erhält der Programmierer bzw. in der Praxis dessen Auftraggeber oder Arbeitgeber dadurch die Stellung eines absoluten, unbeschränkten Gesetzgebers. Es ist daher nicht überraschend, dass inzwischen in Zusammenfassungen⁹ von Vorträgen eigentlich selbstverständliche Aussagen wie

- You cannot escape from law.
- Daher muss Code im Einklang mit dem geltenden Recht formuliert werden.

als bemerkenswerte Thesen oder als besonders hervorzuhebende Aussagen zitiert werden.

Zum anderen wurden durch die Gleichsetzung von *Code* und *Law* auch Bewertungs- und Qualitätskriterien von Programmcode als technische Textgattung auf Gesetzes- und sonstige Rechtstexte, vor allem auch Verträge, übertragen. Die angenommene Identität der beiden Bereiche lässt das ja auch zulässig erscheinen.

Die Tätigkeit von Software-Entwicklern und Juristen kann dann gleichgesetzt¹⁰ werden. Rechtstexte werden nur mehr als eine besondere, vor allem schlecht bzw. mehrdeutig und unklar¹¹ formulierte Form des Programmcodes wahrgenommen.

Es geht dann aber nicht mehr um Fragen der effizienten Gestaltung von Rechtstexten oder die Erhöhung ihrer Verständlichkeit¹². Vielmehr wird unter Annahme eines aggressiven Vorrangs¹³ der Informatik vorgegeben, was Recht in Zukunft sein soll und wie es gestaltet wird. Unter dem Slogan *software is eating law*¹⁴ wird die Zukunft im *computational law*¹⁵ gesehen. Ziel ist es, ein besseres – oder aus Sicht der Vertreter dieser Bewegung – endlich funktionierendes Rechtssystem aufzubauen, denn «when the law becomes publicly verifiable and openly computational, it will be possible to make more definite legal statements and ensure more predictable legal outcomes»¹⁶.

Diese Aussagen und Zukunftsvisionen wären nicht besonders beachtenswert, würden sie nicht eine Grundhaltung im Bereich der Blockchain-Entwickler-Community wiedergeben. Da aber Blockchain und damit verbun-

⁷ Vgl. FILIPPI/SAMER 2016.

⁸ LESSIG 2001, S. 414: «Wenn wir die verschiedenen regulierenden Instanzen relativieren ... werden wir erkennen dass Freiheit nicht einfach dadurch geschaffen wird, dass wir dem Recht Grenzen setzen. Freiheit schaffen wir vielmehr durch Strukturen, die dem Einzelnen einen Spielraum für seine Entscheidungen belassen, so eng dieser Spielraum auch sein mag.»

⁹ REUTER 2017.

¹⁰ SIEGEL 2017: «Thinking like a lawyer and thinking like a programmer aren't that different.»

¹¹ SZABO 2002.

¹² Vgl. KAHLIG 2015.

¹³ Vgl. <http://legalese.com> (alle Websites zuletzt abgerufen am 6. Januar 2018): «They believe in world-domination of the engineering class; everything can be reduced to an algorithm and legal documents are not going to be spared.»

¹⁴ <http://legalese.com> in Anlehnung an ANDREESSEN 2011.

¹⁵ <http://legalese.com>.

¹⁶ GREENWOOD 2016.

den Smart Contracts vielfach als der nächste revolutionäre Entwicklungsschritt im Internet angesehen werden, werden im Folgenden die mit *Code is Law* und Blockchain verbundenen Begriffe und Modelle untersucht.

2. Begriffe

Gerade in der Diskussion von Smart Contracts und Blockchains werden durch Juristen und Informatiker die Begriffe *Law*, *Code*, *smart* und *Smart Contract* mit völlig unterschiedlicher Bedeutung verwendet.

2.1. Smart

Während im allgemeinen Sprachgebrauch *smart* die Bedeutung von «klug», «schlau», «intelligent» oder ähnliches hat, wird im Kontext von Blockchains *smart* mit eindeutig identifizierbar gleichgesetzt. Ein Objekt wird *smart*, wenn ihm in dem System oder vielleicht sogar systemübergreifend ein eindeutiger Bezeichner zugeordnet wird.

So hält MELANIE SWAN zu einem auf einer Blockchain aufbauenden Register fest: «In this system, all property could become *smart property*; this is the notion of encoding every asset to the blockchain with a unique identifier such that the asset can be tracked, controlled, and exchanged (bought or sold) on the blockchain.»¹⁷

Da keine EDV-Anwendung, die Daten in einer Datenbank dauerhaft speichert, ohne eindeutige Bezeichner auskommt, könnten alle diese gespeicherten Daten als *smart* bezeichnet werden. Alleine durch solche eindeutigen Bezeichner wird Software jedoch nicht klug, schlau oder intelligent.¹⁸

2.2. Code

Im englischen Sprachraum verstehen Juristen unter *Code* «a systematic collection or revision of laws, rules or regulations»¹⁹, also kodifizierte Rechtsvorschriften.

Im Gegensatz dazu wird im Bereich der Informatik unter *Code* der Programmcode, also ein in einer Programmiersprache geschriebener Text verstanden.

Prinzipiell unterscheiden und trennen Juristen und Techniker problemlos die beiden Bereiche, doch Verfechter von *Code = Law* fällt folgende verkürzende Argumentation leicht:

- (1) Code: collection of rules/Sammlung von Regeln
- (2) Code: Folge von Anweisungen und Regeln, die ein Computer abarbeitet.

Da es sich nach dieser Argumentation in beiden Fällen nur um Regeln handelt, sind juristischer *Code* und Programmcode dasselbe. Es ist bald eine automatische Übersetzung möglich von juristischem *Code* in Programmcode und umgekehrt möglich.²⁰ Juristen der Zukunft werden daher wie Programmierer arbeiten.²¹

2.3. Law

Law, also Gesetz ist der Begriff, bei dem es zwischen Juristen und Technikern zu den größten Differenzen und Verwirrungen über den Inhalt kommt.

Für einen Juristen ist es selbstverständlich, dass – kurz und äußerst unscharf zusammengefasst – es sich bei einem Gesetz, also einer *Sammlung von Rechtsnormen*, um ein soziales Phänomen handelt. Ein Gesetzgeber

¹⁷ SWAN 2015, S. viii (Hervorhebung im Original).

¹⁸ Geschickt werden in Texten, die die Revolutionierung des Rechts mit Blockchain-Technologie bewerben, die beiden Bedeutungen vermischt, wenn unter der Überschrift «Smart Words for Smart Contracts» von «smart (mostly young) lawyers» gesprochen wird. (SIEGEL 2017).

¹⁹ GARNER 1999, S. 250.

²⁰ «Soon, we will be able to scan our legacy documents, translate them into simple language, then into code, and then into legal ecosystems that plug and play, reducing vast amounts of complexity and opening new possibilities we haven't even thought of yet.» (SIEGEL 2017).

²¹ «I think in five years most law schools will be teaching, probably requiring, coding skills.» (SIEGEL 2017).

gestaltet durch einen gesetzgeberischen Willensakt die Wirklichkeit und setzt das Gesetz in Kraft (oder außer Kraft), um das Verhalten der Normunterworfenen zu regulieren. Ein Gesetz hat daher eine *zeitlich begrenzte Geltung* (Beginn – Ende) und ist *veränderbar*. Der Gesetzgeber verwendet Begriffe zur Beschreibung des Norminhalts. Deren Bedeutung muss durch *Interpretation* festgestellt, um den Inhalt des Gesetzes herauszufinden. Eine Interpretation kann richtig oder falsch sein, das Gesetz selbst nicht.

Gänzlich unterschiedlich ist der Gesetzesbegriff der Technik. Hier geht es – ebenfalls kurz und äußerst unscharf zusammengefasst – um Naturgesetze, die *unveränderbar* und *zeitlich unbegrenzt* gelten. Geltung bedeutet dabei, dass durch das Gesetz eine Beziehung in der Natur/Wirklichkeit korrekt abgebildet wird. Wenn ein Naturgesetz nicht gilt, dann ist es falsch. Beschrieben wird ein Naturgesetz mit der Sprache der *Mathematik*, wobei bei der Anwendung des Gesetzes *Variablen* konkrete *Werte zugewiesen* werden. Die Bedeutung einer Variablen, ihre Semantik, ist im Allgemeinen bekannt und genau festgelegt. Sie muss nicht durch Interpretation festgestellt werden.

Die beiden Gesetzesbegriffe sind so unterschiedlich, dass sie nicht zur Deckung gebracht werden können. Wenn aus der Analogiebeziehung *Code is Law* die Formel *Code = Law* werden soll, dann muss der technisch naturwissenschaftliche Gesetzesbegriff in den juristischen Bereich übertragen und dort implementiert werden. Für den eigentlichen juristischen Begriff bleibt im Bereich des *Code = Law* kein Platz, wie der Begriff der *Smart Contracts* zeigt.

2.4. Smart Contract

Smart Contracts sind keine Verträge im juristischen Sinn. Vielmehr sind sie Programme, die in einem Blockchain-System ausgeführt werden.

Die Verwendung des Begriffs *Contract* kann nur durch eine Uminterpretation des juristischen Ausgangsbegriffs im Sinne der *Code = Law*-Formel gestützt werden.

Dies kann in einem im Blog-Beitrag «Code is Law is Code»²² eingebetteten Foliensatz²³ nachvollzogen werden. Dort wird in einer Beschreibung mathematischer Eigenschaften einer formalen Sprache für die Formulierung von Verträgen von der juristischen Definition eines Vertrags ausgegangen: «A contract is a binding agreement between two or more persons that is enforceable by law.» Da eine verbindliche Übereinkunft in dieser Allgemeinheit mathematisch nicht fassbar ist, wird diese juristische Definition als *conventional contract* bezeichnet und abweichend etwas definiert, was mit einem Vertrag nur wenig zu tun hat: der *deontic e-contract*. «An e-contract is a machine-readable contract» bzw. ausführlicher «A contract is a document which engages several parties in a transaction and stipulates their (conditional) obligations, rights, and prohibitions, as well as penalties in case of contract violations.» Aus seiner Willenserklärung wird ein Dokument und dafür – für den in dem Dokument den zulässigen Text – wird dann eine formale Grammatik definiert. Diese Contract Specification Language könnte dann die Grundlage für eine Programmiersprache sein.

Ein *deontic e-contract* ist daher ein Dokument, das einen Text enthält, der der Grammatik für *deontic e-contracts* genügt. Natürlich ist es u.U. möglich, den Inhalt eines juristischen Vertrags, den Inhalt der Übereinkunft in einem Text, der dieser Grammatik genügt, festzuhalten. Aus juristischer Sicht ist es jedoch nicht notwendig. Wenn es trotzdem gemacht wird, dann wird noch immer ein Vertrag in e-contract-Sprache abgefasst und nicht ein *e-contract* geschlossen. Anhänger der *Code = Law*-Formel würden das anders sehen und davon ausgehen, dass sie – zumindest für den Blockchain-Bereich, für den sie Code-Gesetzgebungskompetenz haben – ein neues Rechtsinstitut definiert haben.²⁴

²² WENG WONG 2017.

²³ PACE/PRISACARIU/SCHNEIDER 2007.

²⁴ Vgl. <http://legalese.com>: «Academic contract formalisations are just now beginning to satisfy the necessary properties for reduction to software practice.»

Verträge sind in die übergeordnete Rechtsordnung eingebettet und durch den Gesetzgeber kann in sie eingegriffen werden, indem z.B. durch Konsumentenschutzbestimmungen bestimmte Vertragsinhalte für ungültig erklärt werden. Gemäß der *Code = Law*-Formel sind Verträge in Blockchain-Umgebungen nur in diese eingebettet und nicht mehr veränderbar. Sie sind – zumindest vom Konzept her – von der Rechtsordnung abgekoppelt.²⁵

Die in Blockchain-Umgebungen lauffähigen *Smart Contracts* haben daher folgende Eigenschaften:

- Sie sind *smart* in dem Sinn, dass jedes einzelne dieser Programme eindeutig identifiziert ist.
- Der Code des *Smart Contracts* ist – entsprechend dem Konzept von Gesetzen im Sinn von Naturgesetzen – unveränderbar.
- Eine Auslegung der Code-Regeln ist nicht möglich. Das Programm reagiert – entsprechend dem Konzept von Gesetzen im Sinn von Naturgesetzen – nur durch Auswertung und Bearbeitung der mit Werten befüllten Variablen im Code.
- Entscheidend ist entsprechend der *Code = Law*-Formel nur der Programmcode des *Smart Contracts* und der Blockchain-Umgebung. Die zuständige Rechtsordnung wird ausgeblendet.

3. Modell der Rechtsanwendung im Code = Law-Universum

Rechtstheoretische Modelle der Rechtsanwendung sind zwischen den beiden in Abbildung 1 dargestellten Extremen angesiedelt.²⁶

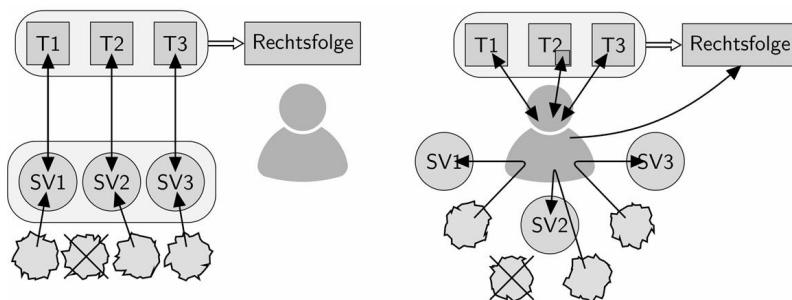


Abbildung 1: Beobachter- und Teilnehmerstandpunkt

Die eine Extremposition ist der Standpunkt des aktiven Teilnehmers am Vorgang der Rechtsanwendung. Zu Beginn des Rechtsanwendungsprozesses sind nur die Tatsachen, die Objekte der Welt vorhanden. Durch juristische Sachkenntnis beurteilt der Rechtsanwender diese Tatsachen und erzeugt die Elemente der Sachverhaltsbeschreibung SV. Gleichzeitig interpretiert er die Tatbestandsmerkmale T der in Frage kommenden Rechtsnormen in Hinblick auf SV. In einem iterativen Vorgang werden die SV und T bzw. die in den Elementen von SV und T verwendeten Begriffe auf einander hin ausgelegt. Dies geschieht so lange, bis die Rechtsfolge abgeleitet oder ausgeschlossen werden kann.

In diesem Modell ist der Rechtsanwender das Zentrum und der Ausführende des Rechtsanwendungsprozesses.

²⁵ Vgl. z.B. das Code is Law-Konzept, aber auch die Diskussionen im Anschluss an den Vorfall, bei dem die *Smart Contracts* der Ethereum-Blockchain Anwendung *The DAO* durch Hacker kompromittiert wurden und dadurch großer finanzieller Schaden entstand: «The original vision of the Ethereum creators was that computer code should, quite literally, be treated as law in their community and serve as replacement for legal agreements and regulation. The DAO creators embraced this vision and noted that participants should look exclusively to the application's code as dispositive on all matters. The code was the contract and the law for The DAO.» (MURCK 2016) Vgl. dazu auch GERARD 2017, S. 101 ff.

²⁶ Vgl. MASTRONARDI 2009, S. 31 ff.

Ganz im Gegensatz dazu steht das Beobachtermodell. Hier hat der Rechtsanwender nur die Funktion zu überprüfen, ob die Zuordnung von Sachverhalt und Tatbestand möglich ist. Die Sachverhaltsbeschreibung und die Semantik der Tatbestandselemente sind bekannt. Die Zuordnung von T und SV ist eine Überprüfung der Übereinstimmung der einzelnen Elemente. Wenn die Zuordnung möglich ist, dann ist die Ableitung der Rechtsfolge ein Automatismus.

Keine der in der Rechtstheorie vertretenen Positionen entspricht einer der beiden Extremstandpunkte. Das Modell der *Smart Contracts* erscheint jedoch wie eine direkte Umsetzung der Kombination der Extrempositionen beider Modelle. Abbildung 2 zeigt die Struktur dieser Rechtsanwendungssituation.

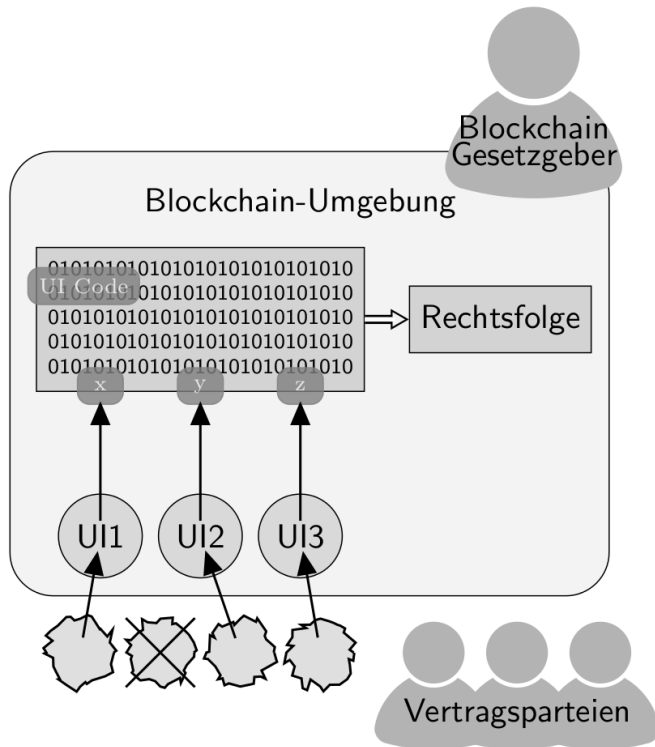


Abbildung 2: Rechtsanwendung mit Smart Contracts

Die Vertragsparteien sind durch die Übertragung der Implementierung des Vertrags auf ein Programm, das in der Blockchain-Umgebung läuft, in die Beobachter-Rolle gedrängt worden und können wegen der Unveränderbarkeit in der Blockchain nicht auf die Rechtsanwendung selbst Einfluss nehmen.

Die Rolle des Teilnehmers übernimmt das Programm *Smart Contract*, das aber nicht wie bei den vorherigen Modellen in die Rechtsordnung eingebettet ist, sondern in die Blockchain-Umgebung. Diese bestimmt technisch, wie die Befehle des Programms umgesetzt werden und ermöglicht auch den Zugriff des Programms auf die Sachverhaltsdaten (in der Abbildung UI1-3), die als Werte in die Variablen (in der Abbildung x-z) eingetragen werden. Der Tatbestand der «programmierten Rechtsnorm» bezieht sich auf die Variablen und erzeugt eine Rechtsfolge, die über die Blockchain-Umgebung umgesetzt werden muss.

Der Betreiber der Blockchain kann im Sinne der *Code = Law*-Formel als Gesetzgeber angesehen werden. So lange die Blockchain-Umgebung nicht in eine Rechtsordnung eingebunden ist, ist er dies absolut und nur durch

die technischen Möglichkeiten beschränkt. Dass diese absoluten Möglichkeiten auch genutzt werden, hat der Fall von *The DAO* gezeigt, bei dem das zentrale Grundprinzip der Blockchain-Technologie, die Unveränderbarkeit der Blockchain, durch die Betreiber in einem Einzelfall aufgehoben wurde.²⁷

In der Blockchain-Umgebung wird auch das Schlagwort *Law is Code* umgesetzt. Dies auf zweifache Weise:

- Werden im Programm *Smart Contract*-Vereinbarungen, die durch einen Vertrag beschlossen wurden, also durch den Vertragsschluss erzeugte Rechtsnormen, umgesetzt, so kann dies nur über Programmcode geschehen. In diesem Sinn ist diese Abbildung in Software ein Beispiel für *Law is Code*.
- Die programmatische Ausgestaltung der Blockchain-Umgebung, in der die einzelnen Blockchains als Anwendungen laufen, gibt vor, wie eine konkrete Blockchain ausgestaltet werden kann und welcher Programmcode in Blockchains überhaupt ausgeführt werden kann. Die Blockchain-Umgebung entspricht daher als Metaebene der Verfassung für den *Smart Contract*-Code. In diesem Sinn gilt auch für die Umsetzung dieser Regelungskonzepte *Law is Code*.

4. Rückwirkungen auf die Rechtsordnung

Neue Technologien wie die Blockchain-Technik, die den Anspruch erhebt, die Rechtsordnung oder Teile davon effizienter, transparenter, verständlicher oder auch nur allgemein besser zu machen, wirken auch auf das Rechtssystem zurück.

Wie dargestellt kommt es im Zusammenhang mit dieser Technik zu einer Umdeutung von Rechtsbegriffen, die aber erstmals nicht nur Konzept bleibt, sondern auch gleich in Blockchain-Umgebungen implementiert werden kann.

Jedoch, nur weil manche Konzepte im Bereich der Blockchain Technik effizient sind, müssen und sollten diese nicht allgemein in der Rechtsordnung Geltung erlangen.

Es ist diesbezüglich Vorsicht angebracht, insbesondere weil gerade Betreiber von Blockchain-Umgebungen in ihrer Rolle als «absolute Gesetzgeber» kein Interesse an Beschränkungen ihres Einflusses haben. Vielmehr wird versucht, unter Umdeutung des Slogans *Code is Law* den Einfluss und die Regelungskompetenz des Rechtssystems so weit wie möglich zurückzudrängen bzw. zu untergraben.

5. Literatur

ABEGG, LUKAS, Code is law? Not Quite Yet, coindesk, <https://www.coindesk.com/code-is-law-not-quite-yet/>, 27. August 2016.

ANDREESSEN, MARC, Why Software Is Eating the World, Andreessen Horowitz, <https://a16z.com/2016/08/20/why-software-is-eating-the-world/>, erstmals veröffentlicht in The Wall Street Journal am 20. August 2011.

ANJUM, MUHAMMAD SALMAN, Blockchain Smart Contract: Code is LAW, LinkedIn, <https://www.linkedin.com/pulse/blockchain-smart-contract-code-law-muhammad-salman-anjum-feellogical-/>, 27. August 2016.

FILIPPI, PRIMAVERA DE/SAMER HASSAN, Blockchain technology as a regulatory technology: From code is law to law is code, *First Monday*, Volume 21, Number 12–5 December 2016, <http://firstmonday.org/ojs/index.php/fm/article/view/7113/5657>, DOI <http://dx.doi.org/10.5210/fm.v21i12.7113>.

GARNER, BRYAN A. (Hrsg.), *Black's Law Dictionary*, West Group, St. Paul, 1999.

GERARD, DAVID, *Attack of the 50 Foot Blockchain*, 2017.

GREENWOOD, DAZZA, Law Itself is the Killer Blockchain App, *Computational Law*, https://computationallaw.org/law-itself-is-the-killer-blockchain-app-5ccf7d86d8d8?lipi=urn%3Ali%3Apage%3Ad_flagship3_pulse_read%3Bdhk%2F4EsfRjq%2FIzq0Gk3XvQ%3D%3D, 11. Juni 2016.

GREENSPAN, GIDEON, Smart contracts and the DAO implosion, *MultiChain*, <https://www.multichain.com/blog/2016/06/smart-contracts-the-dao-implosion/>, 22. Juni 2016.

²⁷ MURCK 2016; vgl. auch Fn. 25.

- KAHLIG, WOLFGANG, Ist Rechtsklarheit überhaupt erwünscht? In: Glück, Beate/Lachmayer, Friedrich/Schefbeck, Günther/Schweighofer, Erich (Hrsg.), Elektronische Schnittstellen in der Staatsorganisation. Festschrift zum 60. Geburtstag von Dr. Josef Souhrada, Österreichische Computer Gesellschaft, Wien 2015, S. 109–117.
- LENK, KLAUS, Die neuen Instrumente der weltweiten digitalen Governance, Verwaltung und Management, 2016, Heft 5, S. 227–240.
- LESSIG, LAWRENCE, Code und andere Gesetze des Cyberspace, Berlin Verlag, Berlin 2001.
- LESSIG, LAWRENCE, Code 2.0, Basic Books, New York 2006.
- LESSIG, LAWRENCE, Code Is Law, Harvard Magazine, <https://www.harvardmagazine.com/2000/01/code-is-law-html>, 1. Januar 2000.
- MASTRONARDI, PHILIPPE, Angewandte Rechtstheorie, Haupt Verlag, Bern 2009.
- MURCK, PATRICK, Who Controls the Blockchain?, Harvard Business Review, <https://hbr.org/2017/04/who-controls-the-blockchain>, 19. April 2016.
- PACE, MENG/PRISACARIU, CRISTIAN/SCHNEIDER, GERARDO, Model Checking Contracts – A Case Study, ATVA'07 Tokyo, Japan, <http://www.cse.chalmers.se/~gersch/slides-talks/slides-ATVA-07.pdf>, 22–25. Oktober 2007.
- ROARK, BRIAN/SPROAT, RICHARD, Computational Approaches to Morphology and Syntax, Oxford University Press, New York 2007.
- REUTER, MICHAEL, Is Code Law? Eine rechtliche Einordnung von Blockchain und Smart Contracts durch Dr. Markus Kaulartz, Datarella, <http://datarella.de/is-code-law-eine-rechtliche-einordnung-von-blockchain-und-smart-contracts-durch-dr-markus-kaulartz/>, 26. Januar 2017.
- SCHNEIDER, JOCHEN/SCHROTH, ULRICH, Sichtweisen juristischer Normanwendung: Determination, Argumentation und Entscheidung In: Kaufmann, Arthur/Hassemer, Winfried (Hrsg.), Einführung in Rechtsphilosophie und Rechtstheorie der Gegenwart, C. F. Müller Juristischer Verlag, Heidelberg 1989, S. 421–464.
- SIEGEL, DAVID, It's Time for Smart Law, Medium, <https://medium.com/@pullnews/its-time-for-smart-law-f218598dee92>, 2. Mai 2017.
- SORGE, CHRISTOPH, Modellierung in Recht und Informatik In: Oostrom, Samuel van/Weth, Stephan (Hrsg.), Festschrift für Maximilian Herberger, juris GmbH, Saarbrücken 2016, S. 877–884.
- SWAN, MELANIE, Blockchain, O'Reilly, Sebastopol 2015.
- SZABO, NICK, A Formal Language for Analyzing Contracts, <http://nakamotoinstitute.org/contract-language/>, 2002.
- WENG WONG, MENG, Code is Law is Code, Medium, <https://medium.com/@Legalese/code-is-law-is-code-4492c864f33f>, 20. Januar 2017.
- WU, TIM, When Code Isn't Law, Virginia Law Review, 2003, Vol 89 Nr 4, S. 679–751.