TRANSFORMACJE PRAWA PRYWATNEGO
3/2020 ISSN 1641–1609

MARLENA PECYNA*, ADAM BEHAN**

SMART CONTRACTS — NOWA TECHNOLOGIA PRAWA UMÓW?

1. WSTĘP

Zagadnienie będące przedmiotem niniejszego opracowania wyrażone w jego tytule brzmi jak oksymoron prawny i prawniczy… Jednakże odmieniany przez ostatnie lata przez wszystkie przypadki blockchain i nadbudowane na nim smart contracts w dużej mierze zrewolucjonizowały lub zrewolucjonizują wiele dziedzin życia społecznego przez nowy rodzaj transakcji[1], model świadczenia usług oraz decentralizację z wykorzystaniem sieci *peer-to-peer* (p2p). Literatura (światowa, europejska, krajowa) przedmiotu jest tak bogata, że nie sposób uważać za możliwe ustosunkowanie się w pełni do wszystkich poglądów w niej zaprezentowanych, zwłaszcza że obejmuje ona różne tradycje prawne, opiera się na różnych systemach prawnych, odmiennych uregulowaniach, w szczególności z zakresu prawa umów[2]. Publikowa-

* *Autorka jest doktorem habilitowanym, profesorem nadzwyczajnym w Katedrze Prawa Cywilnego Uniwersytetu Jagiellońskiego.*

** *Autor jest pracownikiem Katedry Prawa Karnego Uniwersytetu Jagiellońskiego.*

[1] Celowo w tym miejscu używane jest określenie neutralne prawniczo.

[2] Zob. np. N. Szabo: *Smart Contract: Formalizing and Securing Relationships on Public Networks* (https://firstmonday.org/ojs/index.php/fm/article/view/548/469-publisher=First); F. Idelberger: *Connected contracts reloaded — smart contracts as contractual networks* (w:) *European Contract Law in the Digital Age*, ed. S. Grundmann, Cambridge–Antwerp–Portland 2018, s. 205–236; E. Tjong Tjin Tai: *Formalizing contract law for smart contracts*, Tilburg Private Law Working Paper 2017, nr 6 (https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3038800); R.D. Leonhard: *Forget Paris: Building a Carbon Market in the U.S. Using Blockchain — Based Smart Contracts* (https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3082450); P. Catchlove: *Smart Contracts: A New Era of Contract Use* (https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3090226); J.M. Sklaroff: *Smart Contracts and the Cost of Inflexibility*, University of Pennsylvania Law Review 2017, nr 166, s. 263–303; S. McJohn: *The Commercial Law of Bitcoin and Blockchain Transactions* (https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2874463); F. Panisi: *Blockchain and „smart contracts": FinTech innovations to reduce the cost of trust* (https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3066543); G.O.B. Jaccard: *Smart Contracts and the Role of Law* (https://papers.ssrn.com/sol3/papers.cfm?abstract_

ne są liczne opracowania o charakterze praktycznym, popularnonaukowym[3], czy też analizy prawnoporównawcze, a także ujęte z perspektywy europejskiej[4]. Z drugiej strony omawiane zjawisko współcześnie jest jedynie jednym z przejawów nieuchronnie rozprzestrzeniającej się technologizacji życia społecznego, nie mogąc pozostawać niewidocznym dla prawa, mimo że prawo nie ma szans nadążenia za nim.

Według oryginalnej koncepcji smart contracts sformułowanej przez N. Szabo[5] stanowią one algorytmiczne ujęcie transakcji (umowy) w postaci kodu źródłowego programu komputerowego, zabezpieczonego kryptograficznie lub w inny sposób, której jedną z głównych cech jest samowykonalność. Jednocześnie smart contracts różnią się od podawanych przez Szabo jako ich egzemplifikacja maszyn wendingowych (które również cechuje automatyzm wykonania świadczenia wynikający z zależności: *if* — uiszczona opłata, *then* — wydaj towar) między innymi tym, że są zapisane na blockchainie, samo nawiązanie stosunku może być dokonane automatycznie, na przykład z wykorzystaniem tzw. elektronicznych agentów (przykład DAO[6]). Można w uproszczeniu stwierdzić, że smart contracts to skompilowany kod źródłowy programu zapisany na blockchainie, który zapewnia samowykonalność oraz autonomiczną naturę postanowień przewidzianych i zdefiniowanych w kodzie,

id=3099885); R. Holden, A. Malani: *Can Blockchain Solve the Holdup Problems in Contracts?* (https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3093879); T.F. Alabi: *Taking Contracting Digital: Examination of the Smart Contracts Experiment* (https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3015843); E. Mik: *Smart Contracts: Terminology, Technical Limitations and Real World Complexity*, Law, Innovation & Technology 2017, nr 9.2, s. 1–32 (https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3038406); H. Kartik: *Analysis of Contracts in Various Formats of Blockchain* (https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2890577); Ch. Van der Elst: *Bringing the AGM to the 21st Century: Blockchain and Smart Contracting Tech for Shareholder Involvement* (https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2992804); H. Kim, M. Laskowski: *A Perspective on Blockchain Smart Contracts: Reducing Uncertainty and Complexity in Value Exchange* (https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2975770); M.L. Perugini, P.D. Checco: *Smart Contracts: a preliminary evaluation* (https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2729548); M. Raskin: *The Law and Legality of Smart Contracts*, Georgetown Law Technology Review 2017, nr 1:2, s. 305–340; L.W. Cong, Z. He: *Blockchain Disruption and Smart Contracts* (https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2985764); H.M. Kim, M. Laskowski: *Towards an Ontology — Driven Blockchain Design for Supply Chain Provenance* (https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2828369); K. Werbach, N. Cornell: *Contracts Ex Machina* (https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2936294); J. Hazard, H. Happio: *Wise Contracts: Smart Contracts That Work for People and Machines* (https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2925871); A. Saveleyev: *Contract Law 2.0: „Smart" Contracts as the Beggining of the End of Classic Contract Law* (https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2885241); P. De Filippi, A. Wright: *Blockchain and the Law*, Harvard University Press 2018; *idem*: *Lex Cryptographia. Znaczenie prawne umów i jednostek rozliczeniowych opartych na technologii Blockchain*, Warszawa 2018; J. Szczerbowski: *Place of Smart Contract in Civil Law. A Few Comments on Form and Interpretation* (w:) *Proceedings on 12th Annual International Scientific Conference NEW TRENDS 2017*, Znojomo 2017; D. Szostek: *Blockchain a prawo*, Warszawa 2018.

[3] Zob. np. A. Kraińska, R. Kuchta, J. Prokurat, P. Rutkowski: *Blockchain, inteligentne kontrakty i DAO* (https://www.wardynski.com.pl/w_publication/blockchain-inteligentne-kontrakty-i-dao/); R.H. Weber: *Smart Contracts: Do we need New Legal Rules?* (w:) *Digital Revolution — New Challenges for Law*, eds. A. De Franceschi, R. Schulze, C.H. Beck–Hart–Nomos 2019, s. 299–312; F. Möslein: *Legal Boundaries of Blockchain Technologies: Smart Contacts as Self-Help?* (w:) *Digital…*, *op. cit.*, eds. A. De Franceschi, R. Schulze, 2019, s. 313–326.

[4] Zob. *Legal and Regulatory Framework of Blockchains and Smart Contracts. A thematic report prepared by the European Union Blockchain Observatory and Forum*, 27.09.2019 (https://www.eublockchainforum.eu/sites/default/files/reports/report_legal_v1.0.pdf).

[5] Zob. powołaną wyżej klasyczną już pozycję N. Szabo: *Formalizing…*, *op. cit.*

[6] Decentralized Autonomus Organization — zob. niżej.

na warunkach w nich zawartych oraz zastosowanych do wartości majątkowych, które mogą być powiązane z blockchainem[7]. Rzecz jasna połączenie blockchaina i smart contract jest najbardziej klasycznym modelem ich funkcjonowania, jednak nie ma technicznych przeszkód, aby model automatycznie wykonującego się programu zaimplementować za pomocą setek języków programowania i struktur danych. Sięgnięcie chociażby do jednej z pierwszych regulacji dotyczącej definicji pojęcia smart contract zawartej w maltańskiej ustawie Virtual Financial Assets Act (VFA Act)[8] jasno potwierdza, że błędem jest utożsamianie smart contract wyłącznie z DLT, blockchainem, samym bitcoinem czy Ethereum.

Rozważaniom niniejszego opracowania podlega zagadnienie, czy smart contracts wychodzą poza istniejący (prawnie) model kontraktowania i opierają się na nowym paradygmacie transakcji w ramach wirtualnej rzeczywistości. W zasadzie już na wstępie można zauważyć, że sposób funkcjonowania smart contracts świadczyłby o tym, że mogą istnieć bez regulującego je systemu prawnego, reprezentując technologiczną alternatywę, wyrażoną w uniwersalnym języku matematyki, dla tradycyjnej umowy, zgodnie z określeniem „kod stanowi prawo”[9]. Jest to jednak stanowisko trudne do przyjęcia, które miałoby prowadzić do wyłączenia pewnej sfery funkcjonowania obejmującej rzeczywistość wirtualną, ogólnie mówiąc, spod systemu prawnego i jego regulacji. Dlatego liczne analizy prawnicze dążą do objaśnienia funkcjonowania smart contracts w sferze prawnej, tj. do przełożenia niejako języka smart contracts na język prawniczy i prawny, czyli wykonania operacji odwrotnej niż w zamyśle pomysłodawcy smart contracts polegającym na swoistej algorytmizacji podstawowej instytucji systemu prawnego i rynkowego, tj. umowy (jako instytucji prawnej). Nie ma pewności, czy wspomniana operacja odwrotna (próba wdrożenia smart contracts w system regulacji prawa umów i teorię umowy) doprowadzi do uzyskania pierwotnego rezultatu w postaci umowy w tradycyjnie rozumianym pojęciu. Temu zadaniu badawczemu oraz jego wynikom zostaje poświęcony niniejszy artykuł.

2. TECHNOLOGICZNIE O BLOCKCHAINIE I SMART CONTRACTS

Aby przedstawić sposób działania blockchainu oraz smart contracts, wady i zalety, a przede wszystkim zagrożenia, jakie mogą się wiązać z tą technologią ze względu na jej cechy, konieczne jest omówienie pokrótce technicznych aspektów,

[7] Zob. J. Hazard, H. Haapio: *Wise Contracts: Smart Contracts That Work for People and Machines* (https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2925871); A. Saveleyev: *Contract…*, *op. cit.*, s. 15; R.H. Weber: *Smart Contracts…*, *op. cit.*, s. 301.

[8] Zob. https://legislation.mt/eli/cap/590/eng/pdf.

[9] Zob. J. Hazard, H. Haapio: *Wise Contracts: Smart Contracts That Work for People and Machines* (https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2925871); A. Saveleyev: *Contract…*, *op. cit.*, s. 15; R.H. Weber: *Smart Contracts…*, *op. cit.*, s. 301.

które się za nimi kryją. Poniższa analiza, ze względu na ograniczony charakter niniejszego opracowania, nie może objąć wszystkich aspektów działania tej technologii, która, dodać należy, nie jest jednolita, ale posiada niezliczoną ilość odmian. Skupiać się będzie na przybliżeniu czytelnikowi istotnych z punktu widzenia późniejszych rozważań cywilistycznych aspektów jej funkcjonowania.

2.1. BLOCKCHAIN

Jakkolwiek coś, co nazywamy dziś blockchainem, było pomysłem S. Habera i W. Scotta Stornetta z początku lat dziewięćdziesiątych ubiegłego wieku[10], to dopiero praca S. Nakamoto[11] z 2008 r.[12], a w ślad za nią kod źródłowy klienta bitcoin[13] zaprezentowały praktyczny sposób użycia tego konceptu. Block chain, jak nazwał to Nakamoto w kodzie źródłowym[14], był sposobem na rozwiązanie problemu tzw. *double spending*[15] zaimplementowanym w kodzie źródłowym bitcoina.

W związku z założeniem leżącym u podstaw bitcoina, polegającego na oparciu go na braku centralnej instytucji uwierzytelniającej, tzw. *trusted third party*, którą zwyczajowo w obrocie gospodarczym są banki[16], pojawiła się konieczność matematycznego i kryptograficznego zabezpieczenia transakcji w taki sposób, aby jej obecność nie była potrzebna, a wystarczającym zabezpieczaniem środków był sam sposób zapisu danych oparty na kryptografii, ale przede wszystkim, aby system był przejrzysty dla jego uczestników i nie opierał się w żadnej mierze na zaufaniu, ale jawności kodu i algorytmów go zabezpieczających. „Wymyślenie" przez Nakamoto technologii blockchain, która złożona jest z już wcześniej istniejących technologii i pomysłów, legło u podstaw protokołu bitcoin — stworzenia pierwszego systemu, w którym tak udanie rozwiązane są powyższe problemy i który stał się przełomem w implementacji zdecentralizowanego i oderwanego od jakichkolwiek instytucji systemu wymiany dóbr.

Ideą działania blockchain jest tworzenie łańcucha bloków, z których każdy kolejny będzie zależał od poprzedniego. Przez blok rozumieć można ustrukturyzowany zestaw danych, który dla łatwiejszego wyobrażenia możemy nazwać tabelą.

[10] Zob. S. Haber, W.S. Stornetta: *How to time-stamp a digital document*, Journal of Cryptology 1991, Vol. 3(2), s. 99–111. W pracy posługiwali się pojęciem *secured chain of blocks*.

[11] Do dzisiaj nie jest znana osoba (osoby) kryjąca (kryjące) się pod tym pseudonimem, zwłaszcza że w opracowaniu pojawia się wielokrotnie sformułowanie w liczbie mnogiej „we propose", „we define" (ang. my proponujemy, definiujemy). W artykule używamy liczby pojedynczej.

[12] S. Nakamoto: *Bitcoin: A Peer-to-Peer Electronic Cash System* (satoshin@gmx.com, www.bitcoin.org).

[13] Zob. https://github.com/trottier/original-bitcoin/blob/master/src/main.h#L795-L803 (dostęp: 16 czerwca 2019 r.).

[14] *Ibidem.*

[15] W największym skrócie polegający na tym, w jaki sposób, bez „trusted third party" (którą są zwyczajowo banki) zabezpieczyć system, aby niemożliwe stało się kilkukrotne wydanie tych samych środków.

[16] Akceptujemy płatności dokonane na nasze konta bankowe, wierząc, że zgromadzone tam środki są bezpieczne.

W bloku (tabeli) zawrzemy informację dotyczącą niego samego, a także zestaw transakcji/danych czy dowolnych informacji, które w bloku chcemy umieścić. Dla zobrazowania możemy przyjąć, że w bloku mamy informację, iż użytkownik X wysłał użytkownikowi Y kwotę Z. Następnie za pomocą matematycznej funkcji jednostronnie haszującej wygenerujemy hash takiego bloku.

2.1.1. HASH

Funkcje jednostronnie haszujące jak SHA–2[17] czy SHA–3 to funkcje, których idea opiera się na tym samym — z dowolnie długiego ciągu znaków (bitów) — wygenerowany zostanie inny, ale **za każdym razem ten sam** ciąg znaków. Dla przykładu weźmy popularną funkcję SHA–256[18], czyli jeden z wariantów funkcji SHA–2 używany w bitcoinie czy protokołach takich jak TLS/SSL, PGP, SSH, S/MIME, IPsec.

Za każdym razem, jeśli poddamy działaniu funkcji SHA–256 określenie „smart contract", otrzymamy wynik 256b, tj. d0665bba31dcc96a4c03ee110937b38bcad666 7ef6fc026087b02f1c084f9bac. Czytelnik w tym miejscu zastanawiać się może, dlaczego wartość 2^{256} zapisana została za pomocą 64 znaków, a nie 256 zer i jedynek. Do zapisu bowiem wykorzystano system hexadecymalny (szesnastkowy) — sposób zapisu składający się z cyfr od 0 do 9 i liter od a do f[19], zatem z 16 możliwych wartości dla każdego z 4 bitów. Każde 8 bitów[20] można więc zapisać hexadecymalnie za pomocą dwóch znaków[21]. W związku z tym, że $2^{256} = 16^{64}$, można dokonać zapisu 256. bitowej formy w formie, która jest zdecydowanie bardziej czytelna dla człowieka[22].

Co jednak istotniejsze, poddanie działaniu tej samej funkcji ciągu znaków „Smart contract"[23] wygeneruje już zupełnie inny wynik funkcji skrótu. Między innymi tę właściwość, tj. fakt, że zmiana jednego znaku w ciągu poddanym tej funk-

[17] Wykorzystywane w bitcoinie.

[18] 256 oznacza, że wynik jest wynikiem 256. bitowym; są też wersje np. 224, 256 czy 384. bitowe. Zasadniczo większa liczba bitów = większe bezpieczeństwo. Oznacza to, że liczba kombinacji bitu (0 lub 1) wynosi 2^{256}, co równe jest około $1{,}15*10^{77}$. Liczbę atomów w całym wszechświecie szacuje się na około 10^{80}. Liczba ta, jakkolwiek trudna do wyobrażenia, reprezentuje ilość możliwych kombinacji, jaką może mieć wynik tej funkcji, i jest nierozerwalnie związana z prawdopodobieństwem złamania tego zabezpieczania. Funkcja ta uznawana jest za bezpieczną (nie do złamania z wykorzystaniem nawet wszystkich komputerów świata) i opiera się na niej przynajmniej we fragmencie znaczna część używanych protokołów bezpieczeństwa.

[19] A posiada wartość 10, b=11, c=12, d=13, e=14, f=15.

[20] Czyli 256 kombinacji, gdyż 2^8=256. Jako że w informatyce liczenie zaczynamy od 0, dopuszczalne wartości w 8b = 0–255.

[21] I tak wartość np. liczba 79 będzie miała wartość heksadecymalną 4F, a binarną 1100 1111, liczbę 204 zapiszemy hexadecymalnie jako CC, a binarnie jako 1100 1100.

[22] W układzie binarnym zapis ten wyglądałby: 11010000011001100101101110111010001100011101110011001 00101101010010011000000001111101110000100010000100100110111101100111000101111001010110101100110011001111110111011011111100000000100110000010000111101100000010111100011100000010000100111110 01101110101100. Zapis hexadecymalny jest wygodnym dla człowieka sposobem zapisywania wartości binarnych.

[23] Czyli zmiana s na S.

cji zupełnie zmieni jej wynik. Jeśli poddamy działaniu funkcji skrótu ciąg miliarda znaków, a następnie zmienimy w miliardzie znaków jedno zero na jedynkę, to otrzymamy zupełnie inny wynik, co wykorzystał Nakamoto, aby zabezpieczyć bloki przed wsteczną zmianą. Posłużmy się przykładem. Jeśli bowiem pierwszy blok[24] ze wszystkimi danymi będzie posiadał hash ABCD, to drugi blok do wyliczenia swojego hashu będzie musiał sięgnąć do swojego poprzednika i do obliczenia własnej sumy wykorzystać jako jedną ze składowych sumę poprzednika ABCD, co wygeneruje na przykład podpis DEFG. Trzeci blok, aby zostać dołączony do bloku, będzie musiał obliczyć własną sumę kontrolną z wykorzystaniem sumy bloku drugiego (DEFG). W wyniku tego obliczenia powstanie suma kontrolna trzeciego bloku GHIJ. I każdy kolejny blok będzie wykorzystywał sumę swojego poprzednika, która to suma powstała z wykorzystaniem sumy jego poprzednika, a ta z wyniku jego poprzednika, i tak do pierwszego bloku.

Gdyby ktoś zatem dokonał jakiejkolwiek zmiany (nawet 1 bitu ze 100 miliardów bitów) w danych zapisanych w którymkolwiek z wcześniejszych bloków, hash miałby po przeliczeniu (który jest przy każdej transakcji przeliczany ponownie dla wszystkich bloków, począwszy od drugiego) zupełnie inną sumę kontrolną, a przez to każdy kolejny blok byłby błędnie podpisany, przez co wszystkie te bloki zostałyby odrzucone przez program, bo system wykryłby, że ktoś dokonał jakiejś zmiany.

2.1.2. DODAWANIE BLOKÓW DO ŁAŃCUCHA

Wyjaśnienia wymaga również, w jaki sposób powstają rzeczone bloki i jak do łańcucha są one dopisywane. Znów rozważania należy rozpocząć od przedstawienia pierwowzoru zaimplementowanego w bitcoinie, a następnie omówienia pokrótce alternatywnych rozwiązań.

2.1.2.1. Bitcoin

W bitcoinie dodanie bloków jest swoistym wyścigiem opartym na PoW (*Proof of Work*).

Pojęcie *Proof of Work*[25] po raz pierwszy pojawiło się w 1999 r. w pracy M. Jakobssona i A. Juelsa[26], jakkolwiek sam pomysł wykorzystania mocy obliczeniowej warunkujący dostęp do usług był dziełem C. Dwork i M. Naor zaprezento-

[24] Pierwszy blok — nazwany w Bitcoinie Genesis blok nie ma oczywiście hashu poprzedniego bloku.

[25] Zob. też *Proof of Stake (DPoS)and Delegated Proof of Stake (DPoS)*, jako alternatywne dla PoW algorytmy konsensu.

[26] M. Jakobsson, A. Juels: *Proofs of Work and Bread Pudding Protocols (Extended Abstract)* (w:) B. Preneel: *Secure Information Networks: Communications and Multimedia Security IFIP — The International Federation for Information Processing*, 1999, t. 23, s. 258–272.

wanym w ich artykule z 1993 r.[27] Autorki te przedstawiły pomysł konieczności wykonania przez procesor jakiegoś algorytmu warunkującego dostęp do usługi. Miało to być remedium na wciąż narastający problem ze spamem i atakami DDoS[28] i opierało się na banalnym pomyśle. Warunkiem skorzystania z usługi wysłania maila/wejścia na stronę miało się stać wykonanie przez komputer jakichś funkcji matematycznych. Na tyle prostych, aby nie wpłynęło to na komfort korzystania z usługi, a równocześnie na tyle skomplikowanych, aby w przypadku chęci masowej wysyłki maili (spamowanie) przekroczyło to możliwości atakującego[29]. Nakamoto wykorzystał ten pomysł w twórczy sposób, wprowadzając go do bitcoina. Szczegółom technicznym działania tego mechanizmu poświęcić można kilka opracowań, ale w poniższym fragmencie postaramy się przedstawić choć zarys działania tego mechanizmu.

Wynik funkcji SHA–256 wykorzystywany do podpisywania bloków jest liczbą naturalną, której maksymalna wartość składa się z 78 cyfr (minimalną wartością jest zero)[30]. Wartość ta oznacza zatem równocześnie liczbę możliwych jej wyników. Aby jednak nie operować na tak abstrakcyjnych wartościach, ale przedstawić sposób działania PoW, wyobraźmy sobie, że maksymalnym wynikiem, jaki można osiągnąć, jest 1 milion. Każde dane (w naszym przypadku nowy blok), jakie poddamy działaniu tej funkcji, dadzą wynik w zakresie 0 –1 000 000. Nakamoto ustalił zasadę — blok zawierający nowe transakcje doda ten, kto pierwszy znajdzie hash mniejszy niż wartość, którą nazwał target. Target jest wartością zmienną i dopasowywaną automatycznie przez algorytm bitcoina co około dwa tygodnie[31] w taki sposób, aby średni czas znalezienia nowego bloku wynosił 10 minut. Przyjmijmy na potrzeby naszego wyjaśnienia, że wartość target to 100.

Blok ma specjalne pola[32], które mogą być edytowane przez górników (ang. *miners*), czyli osoby, które przeznaczają moc obliczeniową swoich komputerów dla obliczania hashy. I tak mając blok z zestawem transakcji, dodawana jest w polu nonce wartość — na przykład 0, i obliczany dla całego bloku (w skład którego wchodzi pole nonce) hash, który przykładowo da wynik 344 333. Następnie obliczany jest ten sam blok z wartością 1, co da na przykład wynik 134 333. I kolejno inkrementując wartość nonce, obliczane są kolejne hash. Za X razem wyliczony hash wynosił będzie na przykład 456.

[27] C. Dwork, M. Naor: *Pricing via Processing or Combatting Junk Mail* (w:) *Advances in Cryptology — CRYPTO'92*, Lecture Notes in Computer Science 1993, nr 740, s. 139–147.

[28] DDOS.

[29] Jeśli przykładowo warunkiem wysłania maila będzie rozwiązanie algorytmu, który zajmie procesorowi 0,25 sekundy, to próba wysyłki miliona maili będzie wiązać się z obciążeniem procesora przez 70 godzin, skutecznie uniemożliwiając przeprowadzenie takiego ataku bez olbrzymiej mocy obliczeniowej, która z kolei będzie zbyt droga, aby taki atak był opłacalny.

[30] Zob. przyp. 21 — najwyższy wynik tej funkcji można zapisać za pomocą 64 znaków systemu hexadecymalnego, 256. binarnego albo 78. dziesiętnego.

[31] Dokładnie co 2016 nowych bloków.

[32] Nonce, jakkolwiek wykorzystywanych jest bardzo wiele sposobów, aby wygenerować nowe możliwości tworzenia hashu.

W naszym przypadku X uzyskany hash jest wartością mniejszą bądź równą targetowi (1000) i jako taki pozwoli dopisać nowy blok do łańcucha. Wyniku funkcji nie da się jednocześnie przewidzieć; jedyny sposób na znalezienie rozwiązania to sprawdzanie kolejnych możliwości. Jej bezpieczeństwo polega też na tym, że nie da się odwrócić jej działania. Nie istnieje algorytm, który można by zobrazować zapytaniem — co należy wstawić w pole nonce, aby otrzymać 145. Oczywiście górnicy nie poświęcają swojego czasu, sprzętu i pieniędzy bezinteresownie. Za znalezienie rozwiązania otrzymują nagrodę — obecnie[33] w wysokości 6,25 bitcoina, co przy aktualnych cenach oznacza możliwość „wygrania" w swoistym wyścigu co 10 minut równowartości około 65 tys. dolarów[34]. W momencie znalezienia właściwego hashu dla określonej zawartości bloku blok ten dopisywany jest do sieci jako kolejny, a na adres wydobywającego, który znajduje się w bloku, transferowane są nowe bitcoiny. W ten i tylko ten sposób bitcoiny są też tworzone — nie ma innego sposobu ich powstania — dodawane są do sieci tylko jako nagroda za podpisanie bloku. Jednocześnie algorytm dostosowuje trudność poprzez modyfikację targetu — jeśli po 2016 blokach na podstawie znaczników czasowych okaże się, że średni czas „wykopania" bloku wynosił mniej niż 10 minut, algorytm automatycznie dostosuje target wedle wzoru:

oczekiwany czas 2016 bloków w minutach / rzeczywisty czas w minutach

Przyjmijmy, że ze względu na wzrost mocy obliczeniowej sieci spowodowany dołączeniem wielu nowych górników średni czas wydobycia ostatnich 2016 bloków wynosił 9 minut. Do wzoru podstawimy zatem (2016*10) / (2016*9), co da nam 20160 / 18144 = 1.11[35]. Oznacza to, że algorytm zwiększy o 11% trudność, odpowiednio dostosowując target. I tak sterując wartością (w naszym przykładzie 1000), możemy ją zmniejszać do na przykład 10 i niezwykle utrudnić znalezienie prawidłowego rozwiązania, albo zwiększając do 500 000, sprawić, że średnio co drugi hash będzie poprawny. Mając na uwadze cenę bitcoina, nie może być zaskoczeniem, że o podpisanie bloku rywalizują użytkownicy na całym świecie, szukając właściwego hashu dla swojego bloku. Należy mieć jednocześnie na uwadze obecną moc obliczeniową całej sieci bitcoin, która jest kilkadziesiąt razy większa niż moc obliczeniowa 500 najmocniejszych superkomputerów świata[36]. Przeliczenie 10 minut (600 sek.) przez moc obliczeniową sieci pozwala stwierdzić, że do znalezienia rozwiązania (≤ targetowi) testowanych jest przez wszystkich górników łącznie ponad 40 tryliardów hashy[37]. Nie bez znaczenia jest też ilość prądu, jaką pochłania tak duża

[33] Pierwotnie „nagroda" wynosiła 50 BTC, ale co 210 000 bloków nagroda obniżana jest o połowę. W 2024 r. nagroda zostanie ponownie podzielona i wynosić będzie 3,125 BTC.

[34] Według ceny na 4 września 2020 r. W okresie najwyższych cen bitcoina w 2017 r. nagroda za blok była równowartością około 240 tys. dolarów.

[35] Oczywiście ruch możliwy jest w obu kierunkach, algorytm może zmniejszyć, jak i zwiększyć target.

[36] https://www.top500.org/lists/2019/06/.

[37] Network Hash Rate = 9985348008059.555*232/600 / 1018 = 71,47 TH/s * 600 sek.

sieć komputerów. Szacuje się, że całoroczne działanie sieci bitcoin pochłania ponad 62 TWh[38], co jest średnią wartością, jaką rocznie zużywa Austria, i stanowi jednocześnie około 0,25% całego światowego zużycia energii tylko dla obliczania hashy, które służą do podpisania relatywnie niewielkiej ilości transakcji[39], ale to nagroda za podpisanie bloku jest główną determinantą tak olbrzymiej mocy obliczeniowej i nakładów na tzw. farmy składające się z tysięcy układów, których jedynym zadaniem jest szukanie hashy.

W taki skrótowo opisany — ze względu na ograniczenia objętości niniejszego opracowania, sposób zaprojektowany został system, który dał impuls do rozwoju innego narzędzia, z jakim kojarzone są smart contracts.

2.1.2.2. Ethereum

Bitcoin jako protokół ma bardzo ograniczone możliwości programistyczne, został zaprojektowany do obsługi kryptowaluty o tożsamej nazwie, chociaż jest możliwość stworzenia w tym środowisku smart contracts. Dzięki takim mechanizmom jak timelocks[40], służącym warunkowaniu płatności przed zadaną datą lub numerem bloku, sidechains — wykorzystywanego przez RSK Smart Contract Network — sieć opartą na tzw. forkowaniu głównego bloku, czy mimblewimble[41], jest możliwe tworzenie aplikacji opartych na blockchainie bitcoina. Rozszerzenia te, których pierwotnie w bitcoinie nie było, były odpowiedzią na możliwości projektu, z którym naturalnie kojarzone są smart contracts, zarazem największego konkurenta bitcoina na rynku kryptowalut — Ethereum.

Pomysł Ethereum — platformy, która umożliwiłaby budowę kompletnego środowiska dla zdecentralizowanych aplikacji[42], pojawił się w 2013 r., sama platforma

[38] Zob. *Cambridge Bitcoin Electricity Consumption Index* uruchomiony przez Uniwersytet w Cambridge (https://www.cbeci.org/), jakkolwiek inne szacunki mówią o przybliżonej wartości ponad 73 TWh (https://digiconomist.net/bitcoin-energy-consumption).

[39] Około 300–350 tys. transakcji dziennie przy około 150 mln dziennie dokonywanych przez jedną tylko firmę VISA.

[40] Zob. więcej informacji dotyczących wartości nLockTime, CheckLockTimeVerify, Relative locktime czy CheckSequenceVerify, które wprowadzone są albo modyfikacjami samej aplikacji, albo BIP (*Bitcoin Improvement Proposals*; https://github.com/bitcoin/bips/blob/master/README.mediawiki).

[41] W największym skrócie umożliwiający zwiększenie stopnia anonimowości i skalowalności protokołu bitcoin poprzez umożliwienie prywatnych transakcji. Nazwa pochodzi od zaklęcia z serii książek o Harrym Potterze, które pozwalało związać język przeciwnika podczas pojedynku.

[42] „The intent of Ethereum is to create an alternative protocol for building decentralized applications, providing a different set of tradeoffs that we believe will be very useful for a large class of decentralized applications, with particular emphasis on situations where rapid development time, security for small and rarely used applications, and the ability of different applications to very efficiently interact, are important. Ethereum does this by building what is essentially the ultimate abstract foundational layer: a blockchain with a built-in Turing-complete programming language, allowing anyone to write smart contracts and decentralized applications where they can create their own arbitrary rules for ownership, transaction formats and state transition functions". Zob. V. Buterin: *Ethereum White Paper: A Next Generation Smart Contract and Decentralized Application Platform*, 2014.

natomiast wystartowała w czerwcu 2015 r. Stworzona przez V. Buterina[43], dała programistom zestaw narzędzi, za pomocą których mogli tworzyć własne aplikacje[44]. Opiera się ona na kilku składnikach czyniących ją zdolnymi do tworzenia smart contracts. Należą do nich:

A. Ethereum Virtual Machine (EVM), która opisana została w Yellow Paper[45]. Stanowi podstawę działania systemu Ethereum, udostępniając środowisko wykonywania kodu, stanowiąc fundament konsensu, jaki niezbędny jest do działania sieci, bowiem każdy aktywny węzeł sieci musi ją mieć uruchomioną, jeśli chce być jej częścią. Niezależna od sprzętu, na którym działa, ma za zadanie wykonywać kod z kolejnych bloków podpisanych przez górników. Jako że wykonywany kod na wszystkich maszynach da taki sam wynik, pozwala to na utrzymanie tożsamości stanu we wszystkich węzłach. W sieci Ethereum znalezienie bloku (czyli znalezienie noncji, która da hash bloku ≤ target) i jego propagacja przez górnika do wszystkich węzłów sieci możemy wyobrazić sobie jako polecenie wykonania na każdym z klientów (liczonych w setkach tysięcy): wykonaj operację a, b, c, które w tym bloku się znajdują. Biorąc pod uwagę założenie leżące u podstaw systemu, że na każdej z maszyn dane wejściowe są identyczne, to i wynik będzie wszędzie ten sam, przez to stan na wszystkich węzłach sieci będzie identyczny, tworząc tzw. *global truth*.
B. Solidity — obiektowy język programowania stworzony specjalnie do pisania smart contracts w sieci Ethereum, wymyślony przez G. Wooda. W jego składni widać silny wpływ C[46], Pythona czy JavaScriptu. Jego kompilacja do bytecodu pozwala na jego wykonanie na EVM. Jakkolwiek jest to najpopularniejszy język umożliwiający pisanie smart contracts, to powstały również inne języki programowania, takie jak Serpent, LLL, Mutan czy Vyper, za pomocą których można stworzyć smart contracts. Ograniczeniem bitcoina w kontekście wykorzystania go do smart contracts był język programowania, a w zasadzie jego namiastka. Bitcoin działa bowiem za pomocą bardzo prostego języka Script, który jest maszyną stosową (ang. *stack-based*), wykonywanego od lewej do prawej, a co najważniejsze — nie zaimplementowano w nim wykonywania pętli, co czyni z niego język tzw. *non Turing complete*[47]. Sprawia to, że jest zbyt

[43] Programistę ukraińsko-kanadyjskiego pochodzenia zaangażowanego wcześniej w rozwój bitcoina.

[44] Wyróżnia się tzw. White Paper — o której można mówić jako o pewnej propozycji mającej na celu zachęcenie użytkowników do skorzystania z pewnego rozwiązania technicznego. Opisuje problem i proponuje jego rozwiązanie; Yellow Paper — będący technicznym wyjaśnieniem szczegółów zaprezentowanych w White Paper, oraz Beige Paper będący uproszczoną wersją Yellow Paper (zob. White Paper, *Bitcoin: A Peer-to-Peer Electronic Cash System* czy *A Next Generation Smart Contract and Decentralized Application Platform*); zob. np. https://www.allcryptowhitepapers.com/whitepaper-overview/, oraz https://github.com/chronaeon/beigepaper/blob/master/beigepaper.pdf.

[45] Zob. G. Wood: *Ethereum: A secure decentralised generalised transaction ledger*, 2014, s. 10.

[46] Domyślny kompilator Solidity — solc jest napisany w C++.

[47] Jeśli język programowania jest w stanie wykonać każdy algorytm (bez względu, czy zrobi to wydajnie, prosto czy efektywnie), mówimy, że jest Turing complete. Nazwa wzięła się od nazwiska matematyka Alana Tu-

prosty do tego, aby utworzyć za jego pomocą bardziej skomplikowane programy, takie jak smart contracts. W przeciwieństwie do Script, Solidity jest językiem „Turing complete", będąc zaprojektowanym do tworzenia i wykonywania smart contracts.

C. „Waluta" ether (ETH), która dzieli się na walutę zdawkową wei.
1 ether = 10^{12} szabo = 10^{15} finney = 10^{18} wei.
1 ether to zatem trylion wei lub na przykład tysiąc finney.

D. Koncepcja Paliwa (Gas)
W związku z faktem, że Solidity jest językiem charakteryzującym się kompletnością Turinga, możliwe jest na nim uruchamianie pętli. Aby zabezpieczyć system przed błędami w kodzie (lub celowym działaniem), które zawierałyby nieskończone pętle, których z racji idei działania EVM nie dałoby się przerwać, a które to pętle zablokowałyby całkowicie system, oraz z powodu obaw o ataki DDoS, twórcy wprowadzili koncept paliwa (Gas). Wedle założeń każda transakcja wykonywana w ramach systemu Ethereum wymaga do jej wykonania Gas. Ilość Gas niezbędnego do wykonania poszczególnych transakcji określona została w załączniku G do Ethereum Yellow Paper[48]. I tak na przykład każda podstawowa integracja z systemem, tj. transakcja (*transaction*), zużywa 21 tys. Gas, każda operacja SHA3 kosztuje 30 Gas[49], a na przykład wykonanie funkcji „call" — 700 Gas. Nie można jednak posiadać Gas, nie jest to w żadnej mierze jednostka wymiany, ale tylko określony przez system koncept ustandaryzowanego sposobu wyceny czasu pracy systemu. Sam Gas nie ma żadnej predefiniowanej wartości, dopóki nie zostanie ona określona przez użytkownika. Cena Gas, a przez to koszt całej operacji, ustalany jest za pomocą definiowanej przez użytkownika „Gas price". Określa on w niej, ile warty jest dla niego 1 Gas[50]. Stanowi to zachętę dla górnika (który oblicza i waliduje dane, które będą zapisane w bloku) do tego, aby uwzględnił w bloku kod danego użytkownika. Określany jest on w jednostce gwei. 1 gwei to 1000000000 wei = 0,000000001 ether. W czasie wykonywania transakcji ustalany jest jednocześnie przez wykonującego tzw. *gas limit*. Wynik działania *gas price* * *gas limit* oznacza maksymalną kwotę, którą wysyłający jest skłonny wydać na transakcję, która to kwota trafi do górnika, który jako pierwszy znajdzie właściwy hash. Przykładowo zatem określając Gas Limit: 100000, a Gas Price: 21 gwei, otrzymamy:

ringa, który w 1936 r. zaproponował model uniwersalnej maszyny liczącej. Dodać należy, że języki niebędące zupełnymi są niezwykle rzadkie, a zdecydowana większość języków to języki Turing complete.

[48] https://ethereum.github.io/yellowpaper/paper.pdf.

[49] Plus 6 za każdą operację hashowania 256b danych.

[50] Nie da się w tym miejscu uniknąć skojarzenia z paliwem w samochodzie, co pomoże zobrazować ten model. Każda transakcja to określona podróż, która spala zdefiniowaną ilość paliwa. Jedyne, co różnicuje koszty transakcji, to cena paliwa. Może kosztować 1, a może 1000, i to użytkownik określa, ile jest w stanie za nie zapłacić, a górnicy decydują, czy za tę cenę paliwa warto im jest „przejazd zapewnić".

*Max transaction fee: 100,000 Gas * 21 gwei = 0,0021 ether*,
co przy cenie około 395 dol.[51] za 1 ETH wyniesie około 0,357dol. przy wykorzystaniu całości 100 tys. Gas. Zwrócić należy uwagę, że parametr wskazuje na *Gas limit*, czyli górną granicę wykorzystanego Gas. Zadeklarowanie 100 tys. oznacza, że więcej nie zostanie pobrane z konta wysyłającego. Jeśli natomiast wykorzystane zostanie na przykład 30 tys. Gas, to tylko za tę wartość (30 000 * 21 gwei) użytkownik zapłaci. Jeśli jednak zadeklarowana zostanie w *Gas limit* wartość mniejsza niż wymagana (np. określimy 20 tys. Gas, podczas gdy transakcja wymaga do jej obliczenia minimum 21 tys.), to bez względu na *Gas price*, jeśli transakcja zostanie sprawdzona przez górników, to jako błędna zostanie odrzucona. Co jednak istotne, w związku z tym, że praca górnika została wykonana (praca jego komputera), kwota opłaty nie zostanie zwrócona wysyłającemu. Warto w tym miejscu zwrócić uwagę na niezwykle istotną kwestię i najpewniej zastanawiający spójnik „jeśli". Zadeklarowanie odpowiedniej ilości Gas i określenie jego ceny nie gwarantują bowiem włączenia danej transakcji w blok. To górnicy z tzw. *pool*[52] wybierają transakcje, które chcą włączyć do bloku, i zazwyczaj wybierają te najlepiej dla nich płatne. W związku z ograniczeniami przepustowości samej sieci, transakcja, która będzie miała za niski *Gas price*, oczekiwać może na potwierdzenie godzinę albo i dłużej, nie ma bowiem żadnego przymusu, który skłoniłby górnika do uwzględnienia tej transakcji w kopanym bloku, a w konsekwencji na dołączenie go w łańcuchu[53]. Innymi słowy, mogę chcieć przystąpić do jakiegoś smart contract, mieć środki i wydać dyspozycję do systemu, ale przez zbyt małą wartość *gas* górnicy nie będą zainteresowani jej walidacją i dołączeniem do bloku, przez co z przyczyn od nas niezależnych nie dojdzie do jej wykonania.

E. Jawność
Podobnie jak w bitcoinie łańcuch (blockchain) bloków w Ethereum jest jawny, a każdą transakcję można zobaczyć.

Ethereum również działa w oparciu o system PoW, jednak z powodu obaw twórców o przejęcie całego wydobywania Ethereum przez ASIC[54], jak ma to miejsce w przypadku bitcoina, twórcy zdecydowali się nieco zmodyfikować PoW, implementując właściwą funkcję nazwaną Ethash. Funkcja ta, wykorzystując funkcję skrótu Keccak[55] do obliczenia hashy, wymaga intensywnego wykorzystania z pa-

[51] Według ceny na wrzesień 2020 r.

[52] Czyli z zestawu niepodpisanych i nieuwzględnionych jeszcze w łańcuchu transakcji.

[53] Zob. https://ethgasstation.info/, gdzie zamieszczane są aktualne dane dotyczące średniego kosztu transakcji i zależności *Gas price* do czasu włączenia transakcji do bloku.

[54] *Application-specific integrated circuit* — układ scalony zaprojektowany do realizacji z góry określonego zadania. W przypadku sieci Bitcoin ASIC są projektowane wyłącznie do obliczania sum kontrolnych, posiadając olbrzymi hash rate.

[55] Która po nieznacznych modyfikacjach stała się funkcją SHA–3.

mięci (tzw. *Memory hard*)[56], ponieważ było to niezwykle trudne do implementacji na ASIC. W 2018 r. pojawiły się jednak pierwsze ASIC dla Ethash, co przyśpieszyło prace nad postulowaną wielokrotnie wcześniej propozycją zastąpienia PoW algorytmem konsensu *Proof of Stake* (zwaną również Ethereum 2.0)[57]. Obecnie, na razie w wersjach testowych, mamy dwie implementacje Casper: Casper CBC i Casper FFG. Implementacja Casper FFG, rozwijana przez twórcę Ethereum, V. Buterina, zakłada połączenie systemów PoW i PoS, tworząc hybrydowy sposób dowiedzenia wykonanej pracy. Sam protokół jest jednak obecnie w fazie testów i najpewniej jego finalna wersja przewidywać będzie czysty model PoS.

Model tworzenia Ethereum jest w dużej mierze tożsamy z modelem BTC — „wydobywany" on jest w wyniku walidacji bloku przez kopiącego, za co obecnie kopiący otrzymuje 2 ETH[58]. W przeciwieństwie jednak do bitcoina, w którym blok generowany jest średnio co 10 minut, w Ethereum nowy blok dopisywany jest średnio co 10–12 sekund[59]. Tak krótki czas generowania bloku był jednak założeniem Buterina, które miało umożliwić efektywne wykonywanie smart contracts, gdzie czas był niezwykle ważną determinantą. Należy mieć na uwadze, że ze względu na możliwość forkowania[60] bloków, blok uznaje się za bezpiecznie i trwale podpisany po kilku kolejnych blokach, co przy bitcoinie oznacza czas około godziny, a w przypadku Ethereum — około minuty.

Bitcoin i Ethereum nieznacznie różnicuje również sposób ustalania stanu adresu portfela (konta). W obydwu walutach czynność tworzenia nowego „konta" polega na wygenerowaniu klucza prywatnego, za pomocą którego generowany jest zawsze odpowiadający klucz publiczny, a następnie adres (można go porównać z quasi-rachunkiem bankowym), z którego wykonywane są transakcje. Oba systemy opierają się na krzywej eliptycznej ECDSA[61], choć różnicują je funkcje wykorzystywane do generowania adresu z kluczy prywatnych[62].

[56] „The main reason why memory hardness is important is to make the proof of work function resistant to specialized hardware. With Bitcoin, whose mining algorithm requires only a simple SHA256 computation, companies have already existed for over a year that create specialized »application-specific integrated circuits« (ASICs) designed and configured in silicon for the sole purpose of computing billions of SHA256 hashes in an attempt to »mine« a valid Bitcoin block. These chips have no legitimate applications outside of Bitcoin mining and password cracking, and the presence of these chips, which are thousands of times more efficient per dollar and kilowatt hour at computing hashes than generic CPUs, makes it impossible for ordinary users with generic CPU and GPU hardware to compete". Zob. B. Vitalik: *Dagger: a memory-hard to compute, memory-easy to verify scrypt alternative*, Tech Report 2013, hashcash.org website (http://www.hashcash.org/papers/dagger.html) (dostęp: 10 września 2019 r.).

[57] Aktualizacja Ethereum z wersji 1.0 do 2.0 określana jest kryptonimem Serenity.

[58] Ethereum ma bardzo skomplikowany system „bonusowych" płatności uwzględniający innych kopiących, którym nie udało się jako pierwszym znaleźć rozwiązania — tzw. *Omners/Uncles*.

[59] Przez znacznie mniejszą „trudność" odnalezienia bloku.

[60] Na temat forkowania sieci zob. A.M. Antonopoulos: *Mastering Bitcoin. Programming the Open Blockchain*, 2. ed., O' Reilly media 2017.

[61] Elliptic Curve Digital Signature Algorithm, a dokładnie funkcji $y^2 = x^3 + ax + b$, scharakteryzowanej w secp256k1. Zob. http://www.secg.org/sec2-v2.pdf.

[62] Ethereum wykorzystuje prekursora SHA-3 (Keccak), podczas gdy bitcoin korzysta z SHA-256 i RIPEMD-160.

Znaczącą różnicą w sposobie działania obu systemów jest system transakcji i stanu „konta". Bitcoin opiera się na modelu UTXO[63], w którym aby dokonać transakcji, system najpierw weryfikuje, czy suma niewykorzystanych UTXO dla danego adresu jest co najmniej równa wartości transakcji. Oznacza to, że nie ma w systemie informacji o wartości zgromadzonej na danym adresie (który możemy rozumieć jako swoiste konto), ale jest ona każdorazowo wyliczana jako różnica wpływów i wydatków w całym blockchainie. W Ethereum mamy klasyczny system z kontami, na których przechowywany jest ich stan. Jest to zasadnicza różnica — UTXO nie jest bowiem transakcją, ale tylko mechanizmem utrzymywania stanu transakcji, podczas gdy w Ethereum to właśnie transakcje są używane do zmiany stanu kont. W tym ujęciu Ethereum to w zasadzie automat skończony, który po otrzymaniu nowych danych wejściowych (blok), przejdzie w nowy stan. Stan — dodać trzeba — który zostanie tożsamo zmieniony na wszystkich klientach sieci[64].

Ethereum ma dwa rodzaje kont — Externally owned i Contract accounts.

A. Externally owned account to zwykłe konto użytkownika, ma adres i jest kontrolowane za pomocą klucza prywatnego. Może otrzymywać i wysyłać ETH.
B. Contract account to *de facto* program znajdujący się pod adresem konta. Tym w istocie są smart contracts. Są to programy napisane w Solidity lub innym języku znajdujące się pod zadanym adresem sieci. I tak jak każdy inny program, smart contract może w zależności od jego twórcy przyjmować dane wejściowe bądź nie, czy też wykonywać dowolne funkcje, które zostały przewidziane przez jego twórcę. Są wywoływane (uruchamiane poprzez transakcje) z Externally owned account. Aby zatem „wykonać smart contract", trzeba wykonać transakcję na jego adres. To, co stanie się potem, zależy już wyłącznie od kodu samego programu.

2.2. PROBLEMY ZWIĄZANE Z ZASTOSOWANIEM SMART CONTRACT

Interesująco przedstawia się zagadnienie zadeklarowania w czasie wykonania transakcji niewystarczającej ilości Gas. Wyróżnić tu możemy dwie sytuacje: rzadszą, tj. błąd wysyłającego, który zadeklarował niewystarczającą liczbę Gas pomimo informacji, ile Gas kosztuje wykonanie kontraktu. W takim przypadku program nie zostanie wykonany, ale cały Gas zostanie przekazany na rzecz górnika, który podpisał blok, a tym samym dokonał walidacji niepoprawnego zlecenia transakcji.

[63] *Unspent transaction output.*
[64] Co stanie się dzięki EVM.

Porównać można to z wrzuceniem do skrzynki na listy koperty, na którą naklei się niewystarczającą liczbę znaczków. Koperta zostanie ostemplowana i znaczki zużyte, ale list nie zostanie doręczony. Tak stanie się też w przypadku niewystarczającej ilości Gas. Druga i niestety powszechniejsza jest sytuacja, w której z powodu błędu lub niedopatrzenia w kodzie dochodzi do wyczerpania zadeklarowanej w Gas wartości. Jeśli bowiem program (smart contract) będzie posiadał błąd na przykład w pętli, która będzie wykonywać się w nieskończoność, to program ten będzie działać do czasu wyczerpania Gas (każda operacja w kodzie wymaga do wykonania Gas). Rezultatem takiego błędu będzie brak możliwości wykonania programu (niezawarcie smart contractu) i zużycie — dodać należy — nierefundowanego Gas. Jakkolwiek bowiem zawsze należy podawać wyższy *Gas limit* niż szacowane koszty transakcji na wypadek dodatkowych operacji, to zakreślenie bardzo wysokiego limitu powodować może wydanie całego Gas bez wykonania programu, a przez to zawarcia umowy. Jeśli bowiem smart contract jest reprezentacją umowy, której zawarcie warunkuje jego wykonanie (czyli wykonanie programu zazwyczaj połączone z transferem środków), to przez wyczerpanie limitu Gas do transferu tychże, a przez to do zawarcia umowy nie dojdzie. Zużyty jednakowoż zostanie Gas, zostawiając użytkownika z uszczuplonym portfelem i iluzoryczną możliwością dochodzenia roszczeń. Nie sposób bowiem zidentyfikować w większości przypadków twórców takiego programu.

2.3. PROBLEMY Z WALIDACJĄ UPRAWNIEŃ DO KONTA

Kolejnym problemem, na który należy zwrócić uwagę, jest kwestia walidacji uprawnień do samego konta. Jeśli bowiem nierozważny użytkownik zgubi klucz prywatny (który możemy traktować dla uproszczenia jako hasło) do swojego konta, to nie ma możliwości jego odzyskania. Nie ma instytucji, do której może pójść i legitymując się dowodem, odzyskać hasło, a tym samym dostęp do zgromadzonych na nim środków czy na przykład tokenów przypisanych do konta, a uzyskanych w wyniku wykonania smart contracts. Walidacja odbywa się bowiem przy wykorzystaniu klucza prywatnego. Głośno było swego czasu o śmierci zarządzającego największą kanadyjską giełdą kryptowalut Geralda Cottena, który jako jedyny znał hasło do zdeponowanych tam wartych ponad 190 milionów dolarów kryptowalut. Ta sama matematyka, która gwarantuje bezpieczeństwo kryptowalut, sprawia, że odzyskanie tych środków bez znajomości „hasła" (klucza prywatnego) jest niemożliwe, a ich odzyskanie za pomocą prób łamania haseł wszystkim komputerom na świecie zajęłoby tysiące lat. Co więcej, biorąc pod uwagę jawność blockchainu, każdy może te zdeponowane środki na koncie zobaczyć — nie można się do nich tylko dostać. Iluzoryczne wydają się w takiej sytuacji klasyczne środki ochrony prawnej oferowane przez tradycyjny system prawny.

3. RYZYKA TECHNOLOGICZNE ZWIĄZANE Z WYKORZYSTANIEM SMART CONTRACTS

Należy mieć na uwadze swoiste ryzyka związane z wykorzystaniem samych smart contracts w kontekście naruszenia umów zawartych z ich wykorzystaniem. Chodzi o przypadki luk w programach lub językach programowania. Powszechnie omawiany jest przypadek ataku na DAO, czyli Decentralized Autonomus Oganization będącej formą *venture capital*, na który wpływ mieli inwestorzy poprzez tokeny otrzymywane za wpłacany ether. Uruchomiona w kwietniu 2016 r., w maju dysponowała aktywami przekraczającymi 150 mln dolarów, będącymi wpłatami użytkowników systemu chcących inwestować za jej pomocą w projekty biznesowe. DAO przedstawiał się jako „hub that disperses funds (currently in Ether, the Ethereum value token) to projects". W połowie maja 2016 r. opublikowana została praca, która wskazywała na liczne błędy w oprogramowaniu DAO[65], zwracając uwagę na błędy w funkcji „recursive calls". 14 czerwca gotowe były poprawki w kodzie, które oczekiwały na akceptację społeczności. 17 czerwca 2016 r. doszło do ataku na DAO, w którym wykorzystano kombinację błędów w funkcjach split-DAO withdrawRewardFor, co umożliwiło atakującym wielokrotne wypłacanie 30 razy większej ilości tych samych tokenów, niż byli uprawnieni[66]. W ataku wytransferowano poza DAO około 3,6 mln ether o ówczesnej wartości rynkowej około 50 mln dolarów[67]. W związku z postanowieniami smart contract środki przez 28 dni znajdowały się na subkoncie, z którego przed upływem tego czasu nie można było ich wypłacić[68]. Przez te 28 dni społeczność Ethereum prowadziła ożywioną debatę, co zrobić z tą sytuacją, biorąc pod uwagę, że w DAO znajdowało się około 14% całej waluty ether. Wśród społeczności zarysowały się trzy stanowiska — odwrócić działanie hakera i przywrócić stan środków na dzień sprzed ataku (w uproszczeniu cofając się do bloku sprzed ataku) w postaci hard forka, dokonać z wykorzystaniem minerów blokady predefiniowanych operacji na blockchain, czyniąc wszystkie transakcje do wskazanego subkonta DAO nieważnymi[69], albo kontynuować działanie sieci w niezmienianym stanie[70]. W wyniku braku porozumienia spo-

[65] Kod był open-sourcowy i każdy mógł zapoznać się z nim na platformie GitHub.

[66] W największym skrócie atakujący był w stanie zażądać od smart contractu ulokowanego przez niego etheru, ale w kwocie 30 razy większej, niż był do tego uprawniony i wielokrotnie w stosunku do tych samych środków, zanim smart contract dokonał aktualizacji stanu konta.

[67] Dziś, przy cenie około 180 dolarów (wrzesień 2019) to ponad 630 000 mln dolarów.

[68] Zob. dokumentacja DAO — https://github.com/slockit/DAO/wiki/How-to-split-the-DAO.

[69] Uczynić to planowano za pomocą tzw. ataku 51%, czyli hipotetycznej sytuacji, w której ktoś, zdobywając większość mocy obliczeniowej całego systemu, miałby obliczeniową możliwość stworzyć najdłuższy łańcuch, nadpisując stare transakcje. W tym przypadku było to jednakowoż legitymizowane i ustalone w drodze konsensusu społeczności działanie polegające na obliczeniu przez górników na nowo bloków od momentu sprzed ataku.

[70] Zwracano uwagę w tym kontekście, że ceną za przywrócenie stanu blockchaina sprzed ataku będzie upadek konceptu nieodwracalności działań w blockchainie, a zatem zanegowanie jednego z paradygmatów stojących za jego popularnością i zaufaniem do niego, podnosząc, że w blockchainie kod = prawo. Warto nadmienić, że dzień

łeczności, 20 lipca 2016 r. doszło do rozłamu sieci — tzw. forkowania, z którego powstały dwa niezależne byty: Ethereum (gdzie odwrócono działanie hakera, przywracając stan sprzed ataku i zwracając środki poszkodowanym) oraz Ethereum Classic, w którym nie zmieniono efektów przestępczej działalności[71]. Konsekwencją całego zdarzenia było ukazanie możliwego niebezpieczeństwa, jakie łączy się z korzystaniem z blockchainu, a także ożywiło debatę nad koniecznością regulacji samego Ethereum. Warto w tym kontekście zwrócić uwagę na Report of Investigation Pursuant to Section 21(a) of the Securities Exchange Act of 1934: The DAO[72] przygotowany przez amerykańską Komisję Papierów Wartościowych i Giełd[73], która zwracała uwagę na to, że tokeny DAO były papierami wartościowymi[74].

Oczywiście DAO było tylko najgłośniejszym z przypadków błędów w kodzie, które doprowadziły do olbrzymich reperkusji w kwestii postrzegania bezpieczeństwa kryptowalut i milionowych strat użytkowników. Smart contracts podatne są na rozliczne ataki[75] wykorzystujące rozmaite sposoby, luki w kodzie czy samym języku programowania[76].

4. SMART CONTRACTS W PRAWIE UMÓW

4.1. PRÓBY REGULACJI PRAWNEJ

Dotychczasowe próby uregulowania technologii blockchain czy smart contracts *de facto* dowodzą zauważenia istnienia takich technologii przez poszczególnych

po ataku opublikowany został list domniemanego hakera, który stwierdza w nim, że działał wyłącznie w zakresie tego, na co pozwalał mu kod smart contractu, i jego działanie jest w pełni legalne, pisząc w nim: „A soft or hard fork would amount to seizure of my legitimate and rightful ether, claimed legally through the terms of a smart contract. Such fork would permanently and irrevocably ruin all confidence in not only Ethereum but also the in the field of smart contracts and blockchain technology. Many large Ethereum holders will dump their ether, and developers, researchers, and companies will leave Ethereum. Make no mistake: any fork, soft or hard, will further damage Ethereum and destroy its reputation and appeal". Zob. https://pastebin.com/CcGUBgDG.

[71] Sam proces poprzedził nieudany *soft fork*, który miał się dokonać 30 czerwca 2016 r. i został odwołany z uwagi na luki bezpieczeństwa. Zob. http://hackingdistributed.com/2016/06/28/ethereum-soft-fork-dos-vector/.

[72] Zob. https://www.sec.gov/litigation/investreport/34-81207.pdf.

[73] United States Securities and Exchange Commission — niezależna agencja, której zadaniem jest sprawowanie nadzoru nad przestrzeganiem federalnego prawa obrotu papierami wartościowymi oraz regulowanie i kontrolowanie rynku papierów wartościowych w USA.

[74] „Issuers of distributed ledger or blockchain technology-based securities must register offers and sales of such securities unless a valid exemption applies. Those participating in unregistered offerings also may be liable for violations of the securities laws. Additionally, securities exchanges providing for trading in these securities must register unless they are exempt. The purpose of the registration provisions of the federal securities laws is to ensure that investors are sold investments that include all the proper disclosures and are subject to regulatory scrutiny for investors' protection". Zob. Securities and exchange commission, securities exchange act of 1934, Release No. 81207/ July 25, 2017, Report of Investigation Pursuant to Section 21(a) of the Securities Exchange Act of 1934: The DAO.

[75] N. Atzei, M. Bartoletti, T. Cimoli: *A survey of attacks on Ethereum smart contracts*, Technical Report. Cryptology ePrint Archive: Report 2016, nr 1007, https://eprint.iacr.org/2016/1007.

[76] Wspomnieć warto np. the Parity wallet hack, and the Parity wallet suicides liczone w dziesiątkach milionów dolarów.

ustawodawców. Arizona House Bill 2417 z 29 marca 2017 r.[77] wprowadza art. 5, który dotyczy technologii blockchain, definiując smart contract w ust. E.2 jako program, który działa w ramach rozproszonego, zdecentralizowanego, podzielonego i powielanego rejestru podlegającego nadzorowi i pozwala na przenoszenie aktywów do tego rejestru. Doniosłe postanowienie zawarte jest w ust. C tego artykułu, bowiem wynika z niego, że smart contracts mogą funkcjonować w obrocie gospodarczym, a umowy nie mogą zostać pozbawione skutków prawnych, ważności lub wykonalności jedynie na tej podstawie, że zawierają postanowienia smart contract. Z powyższego wynika, że prawodawca objął zakresem uregulowania prawa umów także te, które zostały zawarte jako smart contracts, choć bez szerszego zakresu regulacyjnego. Istotne znaczenie ma regulacja dekretu nr 8 Prezydenta Białorusi z dnia 21 grudnia 2017 r. o rozwoju gospodarki cyfrowej[78], który podejmuje kompleksowe uregulowanie funkcjonowania technologii blockchain, obrót tokenami, a także zawiera kierunkowe rozwiązania dla regulacji cywilnej w celu wdrożenia do niej nowych instytucji opartych na „eksperymencie prawnym — Parku Nowych Technologii", którego dotyczy rzeczony dekret[79]. Zgodnie z definicją zawartą w rzeczonym dekrecie, smart contract oznacza kod komputerowy przeznaczony do transakcji zapisanych w bloku rejestru (blockchain), innym rozproszonym systemie informacyjnym do celów automatycznego wykonania oraz/lub realizacji transakcji lub wykonania innych prawnie doniosłych działań. Dekret zakłada, że w ramach Parku Nowych Technologii zostanie przeprowadzony prawny eksperyment polegający na wprowadzeniu nowych instytucji prawnych w celu umożliwienia ich wdrożenia do regulacji cywilnej Republiki Białorusi (ust. 5 dekretu). W tym celu rezydenci Parku Nowych Technologii mają zapewnione między innymi prawo do wykonania i/lub realizacji transakcji przy użyciu smart contract. Osobę, która wykonuje transakcję przy użyciu smart contract, domniemywa się za poinformowaną o jego postanowieniach, w tym wyrażonych w kodzie, chyba że wykaże okoliczność przeciwną (ust. 5.3 dekretu). Z powyższego wynika, że Park Nowych Technologii stanowi rzeczywiście swoistego rodzaju eksperyment gospodarczo-technologiczno-prawny, który obejmuje również pewne założenia w odniesieniu do funkcji smart contracts w sprawie umów. W pierwszym rzędzie *expressis verbis* dopuszczono wykorzystanie smart contracts w obrocie prawnym. Jednocześnie wprowadzono domniemanie sprowadzające się do przypisania osobie/osobom wykorzystującym

[77] Dostępny na stronie: https://www.azleg.gov/legtext/53leg/1r/bills/hb2417p.pdf.

[78] Zob. w wersji oryginalnej: *ДЕКРЕТ ПРЕЗИДЕНТА РЕСПУБЛИКИ БЕЛАРУСЬ*, 21 декабря 2017 г., № 8, http://pravo.by/document/?guid=12551&p0=Pd1700008&p1=1&p5=0; w tłumaczeniu na język angielski: *Decree of the President of the Republic of Belarus*, 21.12.2017, No. 8, http://law.by/document?guid=3871&p0=Pd1700008e.

[79] Technologia Blockchain została również zdefiniowana w ramach m.in. ustawodawstwa Republiki Gibraltar — zob. Financial Services (Distribiuted Ledger Technology Provoders) Regulations 2017/204 (https://www.gibraltarfinance.gi/uploads/publications/2018/11/16/TkAln_20171013_20Ellul_20_20Co_20Article_20-_20Gibraltar.pdf), a także Republiki Malty w ustawie z dnia 5 lipca 2018 r. Virtual Financial Assets Bill (http://www.justiceservices.gov.mt/DownloadDocument.aspx?app=lp&itemid=29079&l=1).

smart contracts wiedzy/powiadomienia o treści postanowień transakcji, w tym wyrażonych w kodzie. Jest to propozycja regulacji, która nawiązuje do koncepcji tzw. konsensu poinformowanego[80], jednakże dekret przyjmuje dopuszczalność wykazania okoliczności przeciwnej, czyli braku wiedzy o postanowieniach transakcji, w tym wyrażonych w kodzie programu. W tym zakresie prawodawca białoruski dopuszcza jednak możliwość podważenia skutków transakcji opartej na smart contracts, ograniczając ryzyko rzeczywistej nieznajomości treści ich postanowień. Powyższe stawia pod znakiem zapytania pewność transakcji opartej na smart contracts, jeśli jednak rzeczywisty konsens ma znaczenie pierwszorzędne, chociaż się go domniemywa. Jak wspomniano, z samego dekretu wynika pewien eksperymentalny charakter tej regulacji i jej propozycje można uznać za wstępne.

4.2. SMART CONTRACT A POJĘCIE UMOWY

Powyższa analiza ukazująca technologiczny aspekt smart contracts, jak również przykłady regulacji prawnych stanowią punkt wyjścia do dalszych rozważań. Wysiłki wielu badaczy zmierzają do wykazania, że smart contract jednak stanowi umowę w rozumieniu tzw. klasycznego prawa cywilnego, tj. podejmują próbę ustalenia zgodnych oświadczeń woli stron wyrażonych w celu nawiązania wiążącego stosunku umownego[81]. W szczególności wskazuje się na cel wykorzystania smart contracts, który w wielu przypadkach sprowadza się do nawiązania transakcji, na podstawie której dochodzi do przeniesienia praw majątkowych do dóbr cyfrowych opartych na blockchainie. Mimo że wykonanie smart contracts jest automatyczne, to wymaga wyrażenia woli stron w celu jego „uruchomienia", której upatruje się w chwili podjęcia decyzji przez daną osobę o wykorzystaniu smart contract jako swoistego „agenta" w celu zawarcia pewnej umowy i o związaniu jego działaniami. Dana osoba powierza zatem smart contract zawarcie umowy i jednocześnie jej wykonanie, a element zaufania zamiast między stronami występuje w relacji do komputerowego algorytmu[82]. Podnosi się, że smart contracts mogą być „zawarte" i wykonane jedynie w formie elektronicznej. Postanowienia smart contracts stanowi kod (program) komputerowy, a same smart contracts mają podwójną naturę, bowiem z jednej strony stanowią niejako swoistą elektroniczną „formę" obejmują-

[80] Zob. niżej.

[81] Zob. np. M.L. Perugini, P.D. Checco: *Smart Contracts…*, *op. cit.* (https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2729548); M. Raskin: *The Law…*, *op. cit.*, s. 305–326; L.W. Cong, Z. He: *Blockchain…*, *op. cit.* (https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2985764); H.M. Kim, M. Laskowski: *Towards…*, *op. cit.* (https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2828369); K. Werbach, N. Cornell: *Contracts…*, *op. cit.* (https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2936294); J. Hazard, H. Happio: *Wise Contracts…*, *op. cit.* (https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2925871); A. Saveleyev: *Contract…*, *op. cit.* (https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2885241).

[82] Zob. A. Saveleyev: *Contract…*, *op. cit.*, s. 11 (https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2885241).

cą treść stosunku między stronami, a z drugiej, będąc programem komputerowym (kodem), jako takie stanowią obiekt z zakresu własności intelektualnej podlegający ochronie. W związku z tym, że smart contracts są w istocie kodem komputerowym, ich postanowienia są wyrażone w języku formalnym, ściśle zdefiniowanym. Wykładnia smart contracts ma również niejako automatyczny i autonomiczny charakter i opiera się na logice 1 : 0 (prawda : fałsz)[83]. Rozbieżności co do znaczenia postanowień smart contracts nie występują w odniesieniu do stron smart contract, ale, o ile, to ewentualnie pomiędzy osobą zlecającą sporządzenie smart contract a osobą, która tworzy ten kod, jednakże nie wywołują one skutków między stronami smart contract. Sam język smart contract opiera się na schemacie „if…, than", tj. warunkowości. „Zawarcie" smart contract skutkuje jego samowykonalnością niezależnie od woli jego stron oraz niezależnie od działania w tym zakresie osoby dłużnika i okoliczności.

Czy wobec powyższego smart contract może być w jakimkolwiek zakresie utożsamione z pojęciem umowy w klasycznym prawnym jej rozumieniu jako zgodnym oświadczeniu woli co najmniej dwóch stron mających na celu osiągnięcie konsensu i wywołanie skutków prawnych? Rzecz jasna chodzi w tym przypadku o smart contract, który nie stanowi jedynie kodu/programu komputerowego służącego do wykonania wcześniej zawartej umowy, jak również nie o smart contract, którego celem jest wywołanie określonych skutków niebędących odzwierciedleniem zawieranej transakcji. Pytanie dotyczy w tym miejscu możliwości przypisania smart contract stanowiącemu emanację umowy takich jej cech, które doprowadzą do stwierdzenia istnienia umowy między stronami. Jak wyżej wskazano, podnosi się, że złożenia oświadczenia woli można upatrywać w chwili podjęcia decyzji i dokonaniu opłaty (Gas) przez dany podmiot za skorzystanie z mocy obliczeniowej umożliwiającej przeprowadzenie danej transakcji[84]. Przypomnieć jednak należy, że oświadczenia woli składające się na umowę powinny dotyczyć jej treści (niezależnie od tego, czy jest negocjowana, czy nienegocjowana indywidualnie), wyrażać wolę wywołania skutków prawnych, a sama umowa powstaje przez złożenie tych oświadczeń woli. Jeśli chodzi o smart contract, to brak jest składania oświadczeń woli w tym znaczeniu, strony najczęściej są wobec siebie anonimowe, a co najważniejsze, dojście do skutku — „zawarcie" smart contract — zależy od działania osoby trzeciej, tj. „górnika" (*miner*), który doprowadzi do potwierdzenia transakcji (zapisania w blockchainie). Rzeczony górnik nie jest pośrednikiem (pełnomocnikiem, posłańcem) żadnej ze stron, nie działa w jej imieniu, ale wykonuje niezależną czynność techniczną, bez której smart contract nie zostanie „uruchomiony". Przypomnieć również należy, że opłata mniejszej niż wymagana wartości Gas nie doprowadzi do uruchomienia smart contract, mimo że osoba działała w celu doprowadzenia do skutku, zaś górnik tę wartość otrzyma za dokonanie walidacji niewłaściwego zle-

[83] Zgodnie z tzw. *Boolean logic*.

[84] Zob. A. Saveleyev: *Contract…*, *op. cit.*, s. 11 (https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2885241); K. Kowacz: *Smart contract w prawie umów* (praca magisterska), Kraków 2019, s. 26.

cenia transakcji. W tym przypadku można jednak twierdzić, że warunek uruchomienia smart contract nie został spełniony przez zlecającego. Z kolei do braku uruchomienia smart contract może dojść również w przypadku błędu w samym kodzie, który spowoduje wyczerpanie limitu Gas bez efektywnego wykonania programu, czyli do sytuacji, w której osoba ma wolę uruchomienia kodu/smart contract, ale ze względu na błędy w kodzie nie jest w stanie tego osiągnąć. Wracając do przykładu wyżej z listem i błędnie naklejonymi znaczkami, można zwrócić uwagę, że nie dojdzie w tym przypadku do złożenia oświadczenia woli, bo nie dojdzie ono do adresata. Zbyt mała ilość Gas lub jego nieefektywne wykorzystanie ze względu na błąd w kodzie i jego wadliwe działanie nie doprowadzi do wykorzystania smart contract zgodnie z założeniem osoby, która podejmuje czynności mające na celu jego uruchomienie.

4.2.1. SMART CONTRACT JAKO SPOSÓB ZAWARCIA UMOWY I JEJ (SAMO)WYKONANIA

Powyższe ukazuje, że smart contract jako taki nie jest umową i z umową w sensie prawnym nie powinien być utożsamiany. Natomiast smart contract może być uznany za sposób zawarcia i jednocześnie (samo)wykonania umowy, o ile taki jest jego cel i treść kodu (programu). Może być również sposobem wykonania umowy zawartej w inny sposób. Smart contracts mogą występować również jako niezależni (autonomiczni) „agenci", którzy negocjują, dokonują wyboru smart contract, którego celem będzie zawarcie i wykonanie umowy lub innej czynności wywołującej skutki prawne[85]. Jeśli celem wykorzystania smart contract ma być doprowadzenie do przypisania danemu zdarzeniu polegającemu na uruchomieniu kodu oraz jego zapisaniu na blockchainie skutków umowy lub jej wykonania, to w chwili dokonania wszystkich czynności technicznych przez osobę dążącą do powstania tych skutków następuje początek składania oświadczenia, które ma pełnić funkcję oświadczenia woli, a samo jego złożenie następuje z chwilą automatycznego wykonania odpowiedniego algorytmu przez kod, a następnie włączenia go do blockchainu przez górnika. Z tą chwilą następuje również etap samowykonania umowy przez obie (wszystkie) strony. Wykorzystanie smart contract w stosunkach umownych wymusza zatem dopuszczenie skuteczności automatyzmu zawarcia umowy oraz jej jednoczesnego samowykonania. W istocie zatem można zauważyć, że smart contract otwiera drogę do przyjęcia modelu samowykonalnej umowy, na którym się *de facto* zasadza. Jest to umowa zawierana przy wykorzystaniu automatyzmu algorytmu zapisanego w kodzie i jednocześnie samowykonują-

[85] Zob. też: *Legal and Regulatory Framework of Blockchains and Smart Contracts. A thematic report prepared by the European Union Blockchain Observatory and Forum*, 27.09.2019, s. 31–32.

ca się również przy wykorzystaniu kodu. Jest to równocześnie model, który pozbawia oświadczenie woli znaczenia woli rzeczywistej w tradycyjnym rozumieniu[86]. Nie jest to jednak zjawisko nowe, ale raczej postępujące lub przybierające kolejną postać. W tym modelu, którego istotą jest samowykonalność umowy, oświadczenie woli jednej strony nie kształtuje w rzeczywistości zaufania drugiej strony do jego treści, a owo zaufanie koncentruje się raczej na kodzie, co ma następnie znaczenie dla ustalenia treści umowy i jej wykonania[87], bo wykonanie następuje automatycznie, zgodnie z treścią kodu. Kod stanowi umowę, tzn. kształtuje jej treść, która jest jednocześnie samowykonalna. Powyższe nie oznacza jednak, że kod to prawo (parafrazując „Code is law"[88]) w tym znaczeniu, że umowa zawarta przy wykorzystaniu smart contract nie powinna podlegać regulacji prawnej, ale ta regulacja powinna być jednakowoż dostosowana do modelu umowy, której dotyczy i jest oparty na technologii smart contract i blockchain lub im podobnej. W szczególności należy wziąć pod uwagę rozkład lub ograniczenie bądź zmianę punktu ciężkości ryzyk związanych z samowykonalnością umowy. Przy tej okazji można zauważyć, że umowa zawierana za pomocą smart contract opiera się raczej na modelu umowy z systemu amerykańskiego *common law* opartego na Restatement of Contracts (Second)[89], w ramach którego umowa koncentruje się na *promise*, która sama w sobie jest źródłem zobowiązania, jeśli wywoła u drugiej strony zaufanie, które podlega ochronie[90], aniżeli nawet na angielskim *common law*, w którym *promise* wsparte jest przez *consideration*[91], czy kontynentalnym[92]. Nie należy jednakże pominąć zmiany rozumienia i znaczenia *promise*, także zbliżenia jej funkcji do oświadczenia woli, co pozwala jednocześnie na powiązanie modelu kontynentalnego z modelem *common law*. Powyższe zastrzeżenie nie wyklucza jednak dalszego spostrzeżenia, że ze względu na „clickwrap" charakter umowy zawieranej z wykorzystaniem smart contract, w którym strony (lub co najmniej jedna z nich) *de facto* nie znają treści zawieranej umowy, odwołanie do koncepcji klasycznych staje się nieadekwatne. Jedynie takie ujęcie *promise*, któremu przypisuje się skutki zaufania drugiej strony — w tym przypadku nie tyle do oświadczenia strony, ile do kodu, stanowi podstawę rozważań o pojęciu umowy zawartej z wykorzystaniem smart contract[93].

[86] Zob. M. Pecyna: *Merger clause jako zastrzeżenie wyłączności dokumentu, klauzula integralności umowy, reguła wykładni umowy*, Warszawa 2013, s. 249–261 i przywołana tam literatura.

[87] *Ibidem*, s. 250–255.

[88] Zob. L. Lessing: *Code and Other Laws of Cyberspace*, New York 1993, s. 3, oraz R.H. Weber: *Rose is a rose is a rose is a rose — what about code and law?*, Computer Law & Security Review 2018, nr 34, s. 701–706.

[89] Zob.*:* § 75 Restatement of Contracts (Second), Official Text 1981, s. 242–243, G. Gilmore: *The Death of Contract*, Ohio State University Press 1995, s. 76–84.

[90] Zob. np. Ch. Fried: *Contract as Promise. A Theory of Contractual Obligation*, Harvard University Press 1981, s. 11; P.S. Atiyah: *Essays on Contract*, Oxford 1986, s. 121.

[91] Zob. np. G.H. Treitel: *The Law of Contract*, London 2003, s. 67–160.

[92] Nie sposób jednak nie zauważyć, że z kolei regulacja Uniform Commercial Code zakłada, że istotą umowy jest porozumienie stron (*agreement*) (zob. § 1–201 (3) i (11) UCC).

[93] Zob. szerzej współcześnie o *promise*: M. Hogg: *Promises and Contract Law. Comparative Perspectives*, Cambridge University Press 2011, s. 3–24.

Natomiast, jak wyżej zostało wspomniane, sam smart contract stanowi kod/program komputerowy podlegający ochronie na zasadach własności intelektualnej, a jego wady/usterki mogą skutkować odpowiedzialnością[94]. Poniżej zostaną przedstawione podstawowe z punktu widzenia teorii umowy aspekty, na które należy zwrócić szczególną uwagę[95].

4.3. USTALENIE TREŚCI UMOWY ZAWARTEJ Z WYKORZYSTANIEM SMART CONTRACT

Kwestia wykładni umowy zawartej z wykorzystaniem smart contract jest złożona. Z jednej bowiem strony cechą większości systemów prawnych jest przyjęcie zasady wykładni umowy, zgodnie ze wspólnym zamiarem stron niezależnie od dosłownego brzmienia[96], a z drugiej, istotą smart contract jest wyrażenie treści transakcji[97], która przy wykorzystaniu smart contract jest zawierana za pomocą właściwego języka programowania w kodzie programu[98], interpretowanego za pomocą logiki formalnej (prawda/fałsz; 1/0). Według tak ustalonej treści transakcja podlega samowykonaniu. Ze względu na działanie smart contract trudno doszukiwać się wspólnego zamiaru stron, jak również wątpliwości interpretacyjnych, które miałyby zostać rozstrzygnięte na korzyść jednej ze stron (zgodnie z regułą *contra proferentem*), czy też stosować model rozsądnej osoby w celu ustalenia znaczenia niejasnych sformułowań umowy[99]. W zasadzie zastosowanie smart contract do zawarcia umowy skutkuje koniecznością przyjęcia zastosowania reguły analogicznej do *plain meaning rule* czy *parol evidence rule*[100] i to wolnej od kontrowersji, które z powyższymi są związane. Smart contract jako sposób zawarcia i wyrażenia umowy pełniłby funkcję dokumentu, który stanowi centrum rozważań związanych z *plain meaing rule* i *parol evidence rule*. Z wykorzystaniem smart contract można powiązać domniemanie jasności treści umowy skutkujące wyłączeniem dopuszczalności poszukiwania znaczenia odmiennego niż wynikające z kodu oraz prowadzenia dowodów na tę okoliczność. Kod wyznacza zatem w tym przypadku treść umowy. Obalenie tego domniemania nie jest wykluczone, o ile strony są zidentyfikowane oraz *de facto* zawarły umowę w tradycyjnym rozumieniu, a smart contract wykorzystały jako sposób zawarcia umowy i jej formę, natomiast doszło do błędu przy

[94] Zob. też A. Saveleyev: *Contract…*, *op. cit.*, s. 13 (https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2885241).

[95] Ze względu na ramy opracowania zostanie pominięta kwestia funkcjonowania smart contracts w obrocie konsumenckim, bowiem temu zagadnieniu zostanie poświęcona odrębna publikacja.

[96] Zob. szerzej np. M. Pecyna: *Merger clause…*, *op. cit.*, s. 187–211.

[97] Celowo w tym miejscu ominięte zostało pojęcie umowy.

[98] Przykłady języków były podanej wyżej.

[99] Zob. szerzej prawnoporównawczo M. Pecyna: *Merger clause…*, *op. cit.*, s. 211–220.

[100] *Ibidem*, s. 188–211.

sporządzaniu programu mającego treść tej umowy odzwierciedlać. Niemniej ze względu na sposób funkcjonowania smart contracts i ich wykorzystania w praktyce obrotu należy zauważyć, że taki przypadek może się zdarzyć z iluzorycznym prawdopodobieństwem. Nie może mieć w takim przypadku również znaczenia wola osoby, która zleca przygotowanie/napisanie kodu, mającego wyrażać treść określonej transakcji, w razie, gdyby wykładnia smart contract dokonana zgodnie z zasadami logiki formalnej doprowadziła do odmiennych ustaleń niż pierwotne założenie. Jest to kwestia ewentualnej odpowiedzialności programisty za napisanie kodu, którego treść nie jest zgodna z wytycznymi[101]. Kod programu zasadza się na warunku „jeśli x, to y", co determinuje konieczność takiego formułowania treści transakcji, aby dochować wspomnianej relacji przyczynowo-skutkowej. W tym zakresie istotna jest współpraca pomiędzy specjalistami (prawnikami, programistami, ekonomistami itp.). Można powiedzieć, że pełna skuteczność zastosowania smart contracts w obrocie zasadza się na ich swoistej interdyscyplinarności.

4.4. WAŻNOŚĆ I WIĄŻĄCY SKUTEK UMOWY ZAWARTEJ Z WYKORZYSTANIEM SMART CONTRACT

Przyjęta w niniejszym opracowaniu perspektywa oceny prawnej smart contract jako sposobu zawarcia umowy (o ile do umowy smart contract ma służyć) pozwala na stwierdzenie, że nieporozumieniem jest rozważanie, czy smart contract może być nieważny na przykład jako bezprawny (w sensie niezgodności z ustawą lub wykorzystywany do bezprawnych celów). Kod komputerowy nie podlega bowiem ocenie z punktu widzenia ważności/nieważności, chociaż można sobie wyobrazić, że warunek zapisany w kodzie będzie sprzeczny z prawem (ustawą). Ocenie pod względem ważności podlegać może natomiast umowa zawarta z wykorzystaniem smart contract i w razie spełnienia przesłanek nieważności taki skutek może zostać jej przypisany, mimo że zostanie umieszczona w blockchainie i wykonana. Podobnie przedstawia się sprawa z kwestią wad oświadczeń woli, czy to skutkujących nieważnością, czy wzruszalnością, w szczególności jak błąd czy podstęp. Każda z wad oświadczeń woli jako instytucja prawa cywilnego wymaga jednakże szczegółowych rozważań w kontekście jej przesłanek, których wykładnia i zastosowanie winny być dostosowane do smart contracts. W szczególności jeśli wziąć pod uwagę przesłanki błędu prawnie doniosłego, które odnoszą się do treści czynności prawnej oraz istotności w tym znaczeniu, że czynność prawna nie byłaby dokonana, jeśli składający oświadczenie woli wiedziałby, że jest w błędzie[102], to należy zwrócić uwagę na to,

[101] Podobnie A. Saveleyev: *Contract…*, *op. cit.*, s. 14 (https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2885241).

[102] W tym miejscu przesłanki powołane zostały na podstawie regulacji polskiego kodeksu cywilnego (art. 84 k.c.), ale chodzi nie o analizę konkretnego systemu, ale przedstawienie pewnego zamysłu teoretycznego w ocenie sposobu funkcjonowania w ramach porządku prawnego smart contracts.

co następuje. Ustalenie treści czynności prawnej/umowy zawieranej z wykorzystaniem smart contracts powinno nastąpić zgodnie z zasadami wykładni adekwatnymi dla smart contracts, tj. przy założeniu, że treść czynności zapisana kodem jest jasna i zrozumiała. Jednak przy takim założeniu również może wystąpić błąd co do przedmiotu czynności/umowy, który skutkować będzie możliwością uchylenia się od skutków prawnych oświadczenia woli złożonego pod wpływem błędu. Błąd musi istnieć w chwili zawarcia umowy. W przypadku smart contracts oznaczać to będzie moment wykonania kodu i jednocześnie wykonania umowy. Ze względu jednak na specyfikę obrotu internetowego, zwłaszcza z wykorzystaniem nowych technologii, należy uznać, że przydatność klasycznych wad oświadczenia woli jest bardzo ograniczona, jeśli nie nikła. Przesłanki klasycznych rodzajów wad oświadczeń woli są bowiem nieprzystające do umowy zawartej z wykorzystaniem technologii, w tym smart contracts, w szczególności z powodu pozbawienia doniosłości oświadczenia woli w klasycznym rozumieniu tej koncepcji prawa prywatnego. Dlatego też w odniesieniu do obrotu *online* wprowadza się rozwiązania prawne, które mają spełnić wielorakie funkcje (jak np. tzw. konsumenckie prawo do odwołania umowy). Zważywszy na sposób działania smart contracts, należy rozważyć ewentualnie wprowadzenie ogólnego, tzn. dla całości obrotu, niezależnie od charakteru stron, prawa do odwołania umowy lub innego, które pozwoli na doprowadzenie do wyłączenia skutków prawnych umowy opartej na ryzyku wady oświadczenia woli.

Umowy zawarte z wykorzystaniem smart contracts podlegają także regulacji wzorców umownych, czy szerzej — klauzul nienegocjowanych indywidualnie, a prawidłowością obrotu przy takich transakcjach jest sytuacja, w której umowa będzie miała narzucony charakter w całości. Ze względu jednak na sposób zawierania umowy z wykorzystaniem smart contracts ważne jest zastrzeżenie o nieadekwatności regulacji dotyczącej inkorporacji wzorców umownych do stosunku umownego, która co do zasady zakłada doręczenie wzorca lub co najmniej umożliwienie łatwego zapoznania się z jego treścią przed zawarciem umowy[103]. Kontrola wzorców umownych w ramach umowy zawieranej w oparciu o smart contract koncentruje się na kontroli treści, na zasadach uzależnionych od charakteru umowy (konsumencka/między przedsiębiorcami/powszechna).

W związku z możliwością doprowadzenia do upadku umowy zawartej z wykorzystaniem smart contracts, co podważa pewność niepodważalności umieszczonej transakcji w blockchainie, należałoby rozważyć wprowadzenie domniemania ważności transakcji zapisanych w bloku, którego obalenie mogłoby nastąpić przez odpowiedni wpis w innym bloku, który odzwierciedlałby rzeczywisty stan rzeczy i stan prawny danej transakcji. Powstaje pytanie, kto do takiego wpisu na podstawie oświadczenia złożonego przez uprawnionego byłby uprawniony, w szczególności w razie sporu pomiędzy stronami w tym przedmiocie. Jest to jednocześnie pytanie

[103] Zob. np. art. 384 k.c.

o kwestię jurysdykcji oraz możliwości ingerencji organów rozwiązujących spory w dokonywanie wpisów w blockchainie. Nie jest wykluczone, aby kwestia rozwiązywania pewnych sporów była przedmiotem samoregulacji smart contracts[104], ale jednocześnie jest to ingerencja kodu komputerowego w materię regulacji procesowej i związanych z nią gwarancji. Dodatkowo należy zwrócić uwagę, że przesłanki wad oświadczeń woli czy też innych instytucji prowadzących ostatecznie do nieważności umowy lub innej sankcji mają charakter często ocenny i sprowadzają się do przypisania dyskrecjonalnej władzy sędziego, który mocą urzędu posiada kompetencję do wiążącego orzeczenia w danym sporze. Algorytm takich mocy nie ma, jeśli nie mówimy o hipersztucznej inteligencji, która ma imitować działanie mózgu człowieka.

4.5. WYKONANIE I NARUSZENIE ZOBOWIĄZANIA

Założeniem i swoistego rodzaju dobrodziejstwem smart contract jest zapewnienie umowie zawartej z jego wykorzystaniem samowykonalności, tzn. jej wykonania bez udziału celowego działania lub woli jej stron. Niemniej nie jest zasadne stwierdzenie, że wspomniana samowykonalność wyłącza możliwość przyjęcia, że umowy zawarte przy wykorzystaniu smart contracts nie stanowią źródła zobowiązań stron, bowiem założeniem leżącym u podstaw zobowiązania, w tym jego wykonania, jest wola dłużnika odnosząca się również do jego wykonania, oraz wola wierzyciela odnosząca się także do przyjęcia świadczenia[105]. W tym zakresie należy zauważyć, że wykonanie zobowiązania nie musi się opierać na tzw. umownej teorii wykonania, ale na teorii realnego wykonania lub celowego świadczenia[106]. Wykonanie umowy nie musi zatem być oparte na tzw. porozumieniu wykonawczym, ale mogą nastąpić czynności faktyczne w celu wykonania zobowiązania. Rzeczona celowość działania w tym przypadku nie opiera się na czynnościach dłużnika oddzielonych od aktu zawarcia umowy, ale jest od niego zależna. Jednocześnie systemowo trudno uznać, że wykonanie umowy zawartej z wykorzystaniem smart contract nie opiera się na istnieniu zobowiązania, bowiem powstaje wtedy pytanie o tytuł/przyczynę przejścia prawa do danego dobra (szeroko rozumianego) z jednej osoby na drugą. Brak przypisania stronom zobowiązań wyłączałby możliwość ustalenia naruszenia umowy i zastosowanie skutków prawnych tego naruszenia. Samowykonalność umowy nie zabezpiecza przed istnieniem wad jej przedmiotu. Wobec tego zasady dotyczące naruszenia zobowiązania w zakresie, w którym automatyzm wykonania umowy nie

[104] Por. A. Saveleyev: *Contract…*, *op. cit.*, s. 22 (https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2885241).

[105] *Ibidem*.

[106] Por. w tym zakresie np. F. Zoll (w:) *System prawa prywatnego*, t. 6, *Zobowiązania — część ogólna*, pod red. A. Olejniczaka, Warszawa 2018, s. 1048–1060.

wyklucza nienależytego wykonania zobowiązania, należy uznać za mające zastosowanie. Do wyobrażenia natomiast jest sytuacja, w której wykonywanie uprawnień z tytułu naruszenia zobowiązania będzie również podlegać mechanizmowi smart contract, to jest określone uprawnienia, w tym przesłanki skorzystania z nich, będą treścią kodu w ramach smart contract.

5. PODSUMOWANIE

Niniejsze opracowanie ukazuje niektóre problemy związane z wykorzystaniem smart contracts w obrocie prawnym. Założeniem było oparcie rozważań technologicznych na podstawach, których zrozumienie stanowi punkt wyjścia do rozważań o prawie. Jeśli bowiem uznać, co nie wydaje się kontrowersyjne, że technologie blockchain oraz smart conracts mogą stanowić sposób funkcjonowania obrotu dobrami prawnymi, to należało zidentyfikować węzłowe kwestie technologiczne i ryzyka z nimi związane, a także prawne, które determinują miejsce smart contracts w prawie umów. Powyższe ma bowiem znaczenie *de lege ferenda*. Niewątpliwie omawiana technologia (zresztą nie tylko ta) zmienia perspektywę, wprowadzając model umowy samowykonalnej, ale z drugiej strony unaocznia pewien poziom uniwersalizmu podstawowych koncepcji prawa cywilnego. Jednocześnie globalny zasięg funkcjonowania smart contracts wprowadza do systemów prawnych o różnych tradycjach pewien spójny model umowy. Takie wyobrażenie umowy miał twórca pojęcia smart contract i mechanizmu funkcjonowania tego programu. Powyższe z pewnością powinno oddziaływać na proces „modernizacji" prawa cywilnego w celu jego przystawalności i zachowania roli regulacyjnej obrotu prawnego. Jednocześnie należy zauważyć, odnosząc się do założenia badawczego tego opracowania, że przełożenie klasycznej umowy na język kodu komputerowego nie doprowadziło w konsekwencji do uzyskania tego samego modelu umowy opartego jedynie na innym sposobie jej zawarcia. Sposób zawarcia umowy, a taka funkcja została przypisana smart contracts, determinuje stanowisko o modelu umowy i jej pojęciu. Jest to punkt wyjścia do myślenia o prywatnoprawnej regulacji umowy zawartej z wykorzystaniem smart contract. Wspomniane przykłady dostrzeżenia technologii blockchain oraz smart contracts przez ustawodawców, uwzględnienia ich funkcji w ramach obrotu dają wyraz temu, że smart contracts zostały dopiero niejako zaimplementowane (mówiąc językiem technologicznym) do regulacji prawa umów. Jednak dogłębna analiza sposobu działania smart contracts i technologii blockchain nakazuje podjęcie szerszej debaty o prawnym ujęciu smart contracts w prawie umów ze względu na swoistość umowy zawieranej z ich wykorzystaniem oraz na ryzyka z tym związane. Odrębną kwestię stanowi również przystawalność norm z zakresu ochrony konsumenckiej do umów zawieranych przy wykorzystaniu smart contracts. Chodzi o takie możliwości technologii, które w jakimś zakresie wyeliminują lub

mogą wyeliminować ryzyka uzasadniające przyznanie określonym podmiotom w ramach rynku szczególnej ochrony, w szczególności dotyczące wyboru najkorzystniejszej oferty. Z drugiej strony szczegółowego zbadania wymaga sfera swoistych dla technologii smart contracts zagrożeń, które rodzą nowe ryzyka wymagające ingerencji ustawodawczej. Niewątpliwie jednak prawo cywilne teraźniejszości obejmuje również technologię prawa cywilnego, niezależnie od tego, jak „egzotycznie" by to nie brzmiało.

BIBLIOGRAFIA

Alabi T.F.: *Taking Contracting Digital: Examination of the Smart Contracts Experiment*, SSRN Electronic Journal 2017.

Antonopoulos A.M.: *Mastering Bitcoin. Programming the Open Blockchain*, 2. ed., O' Reilly media 2017.

Atiyah P.S.: *Essays on Contract*, Oxford 1986.

Atzei N., Bartoletti M., Cimoli T.: *A survey of attacks on Ethereum smart contracts*, Technical Report. Cryptology ePrint Archive: Report 2016, nr 1007.

Buterin V.: *Ethereum White Paper: A Next Generation Smart Contract and Decentralized Application Platform*, 2014.

Catchlove P.: *Smart Contracts: A New Era of Contract Use*, SSRN Electronic Journal 2017.

Cong L.W., He Z.: *Blockchain Disruption and Smart Contracts*, SSRN Electronic Journal 2017.

De Filippi P., Wright A.: *Blockchain and the Law*, Harvard University Press 2018.

Dwork C., Naor M.: *Pricing via Processing or Combatting Junk Mail* (w:) *Advances in Cryptology — CRYPTO'92*, Lecture Notes in Computer Science 1993, nr 740, s. 139–147.

Fried Ch.: *Contract as Promise. A Theory of Contractual Obligation*, Harvard University Press 1981, s. 11.

Gilmore G.: *The Death of Contract*, Ohio State University Press 1995, s. 76–84.

Haber S., Stornetta W.S.: *How to time-stamp a digital document*, Journal of Cryptology 1991, Vol. 3(2), s. 99–111.

Hazard J., Haapio H.: *Wise Contracts: Smart Contracts That Work for People and Machines*, Social Science Research Network 2017.

Hogg M.: *Promises and Contract Law. Comparative Perspectives*, Cambridge University Press 2011.

Holden R., Malani A.: *Can Blockchain Solve the Holdup Problems in Contracts?*, NBER Working Papers 25833, National Bureau of Economic Research, Inc. 2019.

Idelberger F.: *Connected contracts reloaded — smart contracts as contractual networks* (w:) *European Contract Law in the Digital Age*, ed. S. Grundmann, Cambridge–Antwerp–Portland 2018, s. 205–236.

Jaccard G.: *Smart Contracts and the Role of Law*, Private Law Theory 2018.

Jakobsson M., Juels A.: *Proofs of Work and Bread Pudding Protocols (Extended Abstract)* (w:) B. Preneel: *Communications and Multimedia Security IFIP — The International Federation for Information Processing*, Secure Information Networks 1999, t. 23, s. 258–272.

Kartik H.: *Analysis of Contracts in Various Formats of Blockchain*, Contracts & Commercial Law Journal 2017, t. 18, nr 12.

Kim H., Laskowski M.: *A Perspective on Blockchain Smart Contracts: Reducing Uncertainty and Complexity in Value Exchange*, ICCCN 2017.

Kim H.M., Laskowski M.: *Towards an Ontology — Driven Blockchain Design for Supply Chain Provenance, Conference: Submitted: Workshop on Information Technology and Systems (WITS)*, Dublin, Ireland 2016.

Kowacz K.: *Smart contract w prawie umów* (praca magisterska), Kraków 2019.

Kraińska A., Kuchta R., Prokurat J., Rutkowski P.: *Blockchain, inteligentne kontrakty i DAO*, 2016.

Legal and Regulatory Framework of Blockchains and Smart Contracts. A thematic report prepared by the European Union Blockchain Observatory and Forum, 27.09.2019.

Leonhard R.D.: *Forget Paris: Building a Carbon Market in the U.S. Using Blockchain — Based Smart Contract*, SSRN Electronic Journal 2017.

Lessing L.: *Code and Other Laws of Cyberspace*, New York 1993.

McJohn S.M., McJohn I.: *The Commercial Law of Bitcoin and Blockchain Transactions*, Uniform Commercial Code Law Journal 2016, Forthcoming, Suffolk University Law School Research Paper nr 16–13.

Mik E.: *Smart Contracts: Terminology, Technical Limitations and Real World Complexity*, Law, Innovation & Technology 2017, nr 9.2, s. 1–32.

Möslein F.: *Legal Boundaries of Blockchain Technologies: Smart Contacts as Self-Help?* (w:) *Digital Revolution — New Challenges for Law*, eds. A. De Franceschi, R. Schulze, C.H. Beck–Hart–Nomos 2019, s. 313–326.

Nakamoto S.: *Bitcoin: A Peer-to-Peer Electronic Cash System*, 2019.

Panisi R.: *Blockchain and „smart contracts": FinTech innovations to reduce the cost of trust*, SSRN Electronic Journal 2017.

Pecyna M.: *Merger clause jako zastrzeżenie wyłączności dokumentu, klauzula integralności umowy, reguła wykładni umowy*, Warszawa 2013.

Perugini M.L., Checco P.D.: *Smart Contracts: a preliminary evaluation*, SSRN 2015.

Raskin M.: *The Law and Legality of Smart Contracts*, Georgetown Law Technology Review 2017, nr 1:2, s. 305–340.

Restatement of Contracts (Second), Official Text 1981.
Saveleyev A.: *Contract Law 2.0: „Smart" Contracts as the Beggining of the End of Classic Contract Law*, Information & Communications Technology Law 2017.
Sklaroff J.M.: *Smart Contracts and the Cost of Inflexibility*, University of Pennsylvania Law Review 2017, nr 166, s. 263–303.
Szabo N.: *Smart Contract: Formalizing and Securing Relationships on Public Networks*, First Monday 1997, 2(9).
Szczerbowski J.J.: *Place of Smart Contract in Civil Law. A Few Comments on Form and Interpretation* (w:) *Proceedings on 12th Annual International Scientific Conference NEW TRENDS 2017*, Znojomo 2017.
Szczerbowski J.J.: *Lex Cryptographia. Znaczenie prawne umów i jednostek rozliczeniowych opartych na technologii Blockchain*, Warszawa 2018.
Szostek D.: *Blockchain a prawo*, Warszawa 2018.
jong Tjin Tai E.: *Formalizing contract law for smart contracts*, Tilburg Private Law Working Paper 2017, nr 6.
Treitel G.H.: *The Law of Contract*, London 2003.
Van der Elst Ch., Lafarre A.: *Bringing the AGM to the 21st Century: Blockchain and Smart Contracting Tech for Shareholder Involvement*, SSRN Electronic Journal 2017.
Weber R.H.: *Rose is a rose is a rose is a rose — what about code and law?*, Computer Law & Security Review 2018, nr 34, s. 701–706.
Weber R.H.: *Smart Contracts: Do we need New Legal Rules?* (w:) *Digital Revolution — New Challenges for Law*, eds. A. De Franceschi, R. Schulze, C.H. Beck–Hart–Nomos 2019, s. 299–312.
Werbach K., Cornell N.: *Contracts Ex Machina*, Duke Law Journal 2017, t. 67, nr 2.
Wood G.: *Ethereum: A secure decentralised generalised transaction ledger*, 2014.
Zoll F. (w:) *System prawa prywatnego*, t. 6, *Zobowiązania — część ogólna*, pod red. A. Olejniczaka, Warszawa 2018, s. 1048–1060.

Słowa kluczowe: blockchain, smart contracts, Ethereum, umowa samowykonalna.

MARLENA PECYNA, ADAM BEHAN

SMART CONTRACTS — NEW TECHNOLOGY OF CONTRACT LAW?

S u m m a r y

The article concerns selected issues regarding smart contracts from the perspective of private law, in particular the concept of a contract, determination of its content, principles of performance and breach of an obligation. The legal analysis jest supplemented with technological aspects that show the essence and mechanism of operation of Blockchain, smart contracts, Ethereum. The legal doctrine generally referred to the technological aspects of smart contracts, attempting to include them in the traditional contract law system. The article contends that this is the wrong approach. The authors argue that a smart contract as such is not a contract, but a computer program (code) that can be a manner of concluding a contract and at the same time self-executing it. The qualification of a smart contract as a method of conclusion a contract, and not the contract itself, determines the conclusions on other aspects of concluding the contract, its interpretation, etc. Considerations concerning the model of a self-executing contract are formulated around this thesis. This model determines the proposals for regulation of the principles of performance of the contract, the consequences of the breach of contract and others areas of concern.

Keywords: Blockchain, smart contracts, Ethereum, self-executing contract.