From old to new: from internet to smart contracts and from people to smart contracts

T.J. de Graaf[1]
Leiden University, Faculty of Law, Institute of Private Law

Published in Computer Law & Security Review 35 (2019) 105322,
https://doi.org/10.1016/j.clsr.2019.04.005

Abstract

Discussing legal issues related to smart contracts on the blockchain is very topical. This article will discuss primarily smart contracts on the blockchain the conclusion and execution of which does not interact with the physical world, as well as briefly touch upon smart contracts on the blockchain which do interact with the physical world. For these smart contracts, it will be determined to what extent existing EU internet laws can help support their development and if not, what is needed to support this. In order to answer this question, the following will be discussed: the rise of e-commerce and in particular the EU internet laws supporting and regulating e-commerce, how smart contracts work and how smart contracts compare with existing technological developments and comparable legal constructs (internet, bank accounts and bank guarantees). Subsequently, it will be explained how the use of smart contracts leads to a shift of confidence, from trust in people to trust in code. On the basis of The DAO hack and the problems that arose, it will be illustrated that this shift to trust in code is not as absolute as is often thought. The article concludes that applying specific EU laws on supporting and regulating e-commerce to smart contracts is difficult for two reasons. First of all, the starting points differ: trust in people versus trust in code. Secondly, technical and practical obstacles often inhibit applying internet laws in a meaningful manner. When using smart contracts, it makes more sense to prevent problems from arising than to correct them afterwards. For this reason, it is advocated that programmers work together with lawyers to create better smart contracts and that the legislator focuses on laws dealing with auditing smart contracts code by trusted third parties and automatically equating smart contracts with written contracts with wet ink signatures. This will hopefully facilitate the rise of smart contracts on the blockchain.

1. Introduction

Smart contracts are, in short, "little programs that execute 'if this happens then do that'."[2] Smart contracts that run on the blockchain are more difficult to define in a concise manner. On the basis of a number of characteristics, they can be described as software programs:

1. that are stored and executed without an intermediary in a decentralised manner on various computers (nodes) which are connected on a peer-to-peer basis to each other in a network and owned by different people;
2. that execute 'if this then that' commands autonomously so that contractual promises are automatically executed;
3. in respect of which, as a condition precedent, a transfer of value (e.g. payment by the customer) can only take place if ultimately at least 51% of the nodes have reached consensus that the execution of the smart contract (e.g. provision of the service by the supplier) has occurred in accordance with the requirements stipulated in such coded contract; and
4. the storing of which takes place in a public ledger which cannot be changed and which is often referred to as a secure public ledger with a single source of truth.

[1] mr. dr. Tycho J. de Graaf is an assistant professor of private law at Leiden University. This article is a remastered version of my Dutch-language article 'Van oud naar nieuw: van internet naar smart contracts en van mensen naar code (I) en (II, slot)', [2018] WPNR 494 and 525, of which a number of Dutch law elements were removed and to which a number of comparative law elements as well as a discussion of smart contracts interacting with the physical world were added.

[2] https://bitsonblocks.net/2016/02/01/a-gentle-introduction-to-smart-contracts/. See for a concise definition of smart contracts also https://www.uitlegblockchain.nl/smart-contracts/.

As with any new technological development, a number of legal questions arise of which I will try to answer one: to what extent can existing EU internet laws help support the development of smart contracts and if not, what is needed to support this? In order answer this question, I will proceed as follows. I will briefly describe the rise of e-commerce and in particular the specific EU laws supporting and regulating e-commerce (hereafter: internet laws). Then, I will discuss how smart contracts work and how they compare with existing technological developments and comparable legal constructs: internet, bank accounts and bank guarantees. Subsequently, I will explain how the use of smart contracts leads to a shift of confidence, from trust in people to trust in code. On the basis of a discussion of The DAO hack and the problems that arose, I will consider whether this shift to trust in code is as absolute as is often thought. I conclude with an answer to the aforementioned question on the usefulness of EU internet laws with respect to blockchain and the extent to which they should be amended in order to support smart contract development. I will only discuss the usefulness of EU internet laws firstly because those are laws which seek, insofar possible in technologically neutral manner, to facilitate contracting by electronic means and therefore could potentially facilitate the use of smart contracts in the blockchain, and secondly because those laws are harmonised across the EU and are therefore relevant to a broad audience. For the sake of brevity and clarity, I will do all of this only on the basis of Ethereum, a leading smart contracts platform that everyone can take part in and which is therefore permissionless.[3] For the purposes of most of this article, I will discuss only smart contracts stored and executed on the blockchain without any connection to the physical, off-chain world and sometimes briefly touch upon the extra complexities introduced by smart contracts which connect to the physical word, e.g. in the shipping and airline industry.

2. The rise of the internet

The rise of the internet started at around 1990. In contrast to closed EDI systems,[4] the internet allows suppliers to sell products and services automatically to customers with whom they have no previous legal relationship. In the beginning, some questioned whether governments should be allowed to interfere with the internet. For instance, John Perry Barlow, one of the founders of the Electronic Frontier Foundation, wrote in his 1996 Declaration of the Independence of Cyberspace: "*Governments of the Industrial World, you weary giants of flesh and steel, I come from Cyberspace, the new home of Mind. On behalf of the future, I ask you of the past to leave us alone. … Where there are real conflicts, where there are wrongs, we will identify them and address them by our means*."[5]

The wishes of Barlow did not come true. On the contrary. In addition to privacy and sector specific regulation, the European Union adopted detailed directives by means of which it, through its Member States, sought to facilitate internet sales, yet at the same time protect weaker parties, in particular consumers. Examples are the (minimum harmonisation) 1997 B2C Directive on Distance Contracts[6] (now replaced by the (maximum harmonisation) 2011 B2C Consumer Rights Directive[7])

[3] See for a concise explanation of Ethereum https://bitsonblocks.net/2016/10/02/a-gentle-introduction-to-ethereum/ and for a more extensive explanation http://www.ethdocs.org/. In this article, permissioned blockchain platforms such as Hyperledger https://www.hyperledger.org or Corda https://www.corda.net are not discussed. These are blockchain platforms the access to which can be protected by means of a so-called access control layer and in which different powers can be assigned per user (or user type). See for a comparison between Ethereum, Hyperledger and Corda https://medium.com/@philippsandner/comparison-of-ethereum-hyperledger-fabric-and-corda-21c1bb9442f6.

[4] EDI is the electronic exchange of structured and standardised messages between information systems, e.g. supermarket cashiers scan products in their PoS (Points of Sale) systems, which PoS send that data to the supermarket's warehouse management system (WMS), which WMS in turn automatically places an order with the supplier if the stock of the product drops below a pre-set level.

[5] https://www.eff.org/cyberspace-independence.

[6] Directive 97/7/EC of the European Parliament and of the Council of 20 May 1997 on the protection of consumers in respect of distance contracts, OJ L 144, 4.6.1997, p. 19–27.

[7] Directive 2011/83/EU of the European Parliament and of the Council of 25 October 2011 on consumer rights, amending Council Directive 93/13/EEC and Directive 1999/44/EC of the European Parliament and of the Council and repealing Council Directive 85/577/EEC and Directive 97/7/EC of the European Parliament and of the Council, OJ L 304, 22.11.2011, p. 64–88.

and the 2000 B2B & B2C E-commerce Directive.[8] In future, we will probably be faced with a Digital Content Directive[9] and a Sales Directive,[10] the latter of which initially only pertained to online sales, but the scope of which has now been extended to include other forms of sales as well.

2.1 Information obligations and obligations with regard to the manner of contracting

As a result of the implementation of the E-commerce Directive and the (predecessor of the) Consumer Rights Directive in the European Union's Member States, a legal framework was established as a result of which a number of legal uncertainties for suppliers to sell products and services via the internet were removed. A country-of-origin principle was introduced as a consequence of which suppliers, in principle, only have to comply with the regulations of the country from which they act (article 3 and 4 E-commerce Directive). Also, Member States were required to allow contracts to be concluded by electronic means (article 9 paragraph 1 E-commerce Directive).

At the same time, a large number of obligations were imposed on suppliers. A supplier must, for example, provide a lot of information about himself, the products and services offered, the way in which the contracting process is set up and executed as well as the contract terms and general conditions used (see articles 5 and 10 E-Commerce Directive). The supplier must also arrange his contracting process in such a way that input errors can be identified and corrected and that the receipt of orders is acknowledged (article 11 E-Commerce Directive). Consumers, on the other hand, are afforded a lot of mandatory protection, including the right to withdraw from distance contracts without reason during a period of 14 days after the day the contract is concluded (in the case of service contracts) or the day on which the consumer acquires physical possession of the product concerned (in the case of sales contracts), see article 9 Consumer Rights Directive. In most cases, the directives seek to overcome the weaker position a customer has when purchasing products or services via the internet when compared to purchasing at a brick-and-mortar store.

2.2 Equating electronic contracts with written contracts

As contracts were increasingly being concluded by electronic means, concerns arose about their legal validity. This was especially true in those cases where laws in Member States required contracts to be concluded in writing, failing which the contract would be null and void or subject to nullification. At EU level, attempts were made to remove barriers to as well as facilitate electronic contracting. The E-Commerce Directive, for instance, obliges the Member States to "ensure that their legal system allows contracts to be concluded by electronic means" and that "the legal requirements applicable to the contractual process neither create obstacles for the use of electronic contracts nor result in such contracts being deprived of legal effectiveness and validity on account of their having been made by electronic means." (article 9 paragraph 1 E-Commerce Directive).

2.3 Equating electronic signatures with wet ink signatures

At EU level, laws also came into being to equate electronic signatures with written signatures, in particular by means of the 1999 Electronic Signatures Directive,[11] which was replaced in 2016 by the

[8] Directive 2000/31/EC of the European Parliament and of the Council of 8 June 2000 on certain legal aspects of information society services, in particular electronic commerce, in the Internal Market, OJ L 178, 17.7.2000, p. 1–16.

[9] Proposal for a Directive of the European Parliament and of the Council on certain aspects concerning contracts for the supply of digital content, COM/2015/0634 final - 2015/0287 (COD).

[10] Amended proposal for a Directive of the European Parliament and of the Council on certain aspects concerning contracts for the online and other distance sales of goods, amending Regulation (EC) No 2006/2004 of the European Parliament and of the Council and Directive 2009/22/EC of the European Parliament and of the Council and repealing Directive 1999/44/EC of the European Parliament and of the Council, COM/2017/0637 final - 2015/0288 (COD).

[11] Directive 1999/93/EC of the European Parliament and of the Council of 13 December 1999 on a Community framework for electronic signatures, OJ L 13, 19.1.2000, p. 12–20.

eIDAS Regulation.[12] An interesting aspect of this regulation is that only one form of electronic signature, the qualified electronic signature, automatically has the same legal effect as a wet ink signature (article 25 paragraph 2 eIDAS Regulation).[13] In order for the addressee's electronic signature to be considered a qualified electronic signature, it must be based on a qualified certificate for electronic signatures issued by a qualified trust service provider, i.e. a provider which has received the qualified status by a supervisory body (article 3 subsections 12, 11, 23, 15, 14 and 20 eIDAS Regulation) and which I shall refer to as a certification service provider (CSP).[14] A CSP is an example of a so-called trusted third party (TTP). An addressee wishing to acquire a qualified certificate can request a CSP to provide such a certificate. A CSP may only provide such a certificate to a person if he is able to authenticate that person, i.e. successfully establish that the person is the person he claims to be.[15] Once the addressee has acquired such a certificate, the sender can use such a certificate to verify with the CSP that the public key (which he uses to encrypt his message before sending it to the addressee) belongs to the addressee. This system is also known as public key infrastructure (PKI). Obtaining and using a qualified electronic signature is quite cumbersome, yet it is the only form of electronic signature which the European legislator automatically equates with a wet ink signature.

2.4 Intermediate conclusion

Combining the above directives and regulation yields the following picture. Firstly, such laws place great emphasis on information that the supplier must provide electronically and on the way in which he must organise his contracting process. By placing so much emphasis on the supplier, the EU legislator makes it clear that he wants to create trust in the supplier and his actions. Secondly, for contracts which statute provides are required to be in writing and signed with a wet ink signature, article 9 paragraph 1 E-Commerce Directive provides that, in most cases, electronic equivalents may be used, but article 25 paragraph 2 eIDAS Regulation provides that an electronic file signed electronically is only *automatically* equated with a written document signed with a wet ink signature if, amongst other things, a TTP is involved, in this case a CSP. And for those contracts which statute does not require to be in writing or signed with a wet ink signature, statute may provide that greater evidentiary value is attached to those contracts which are nonetheless in writing and signed with a wet ink signature.[16] For that reason, parties concluding contracts electronically have an interest in using the only form of electronic signature which is automatically equated with a wet ink one: the qualified electronic signature. Yet, by requiring the use of a TTP to make use of a qualified electronic signature, the EU legislator places great emphasis on a TTP and thereby places great confidence in the centralisation of trust, namely at such a TTP. In doing so, the EU legislator introduces a single point of failure susceptible to hacking, as the Diginotar case shows.[17]

[12] Regulation (EU) No 910/2014 of the European Parliament and of the Council of 23 July 2014 on electronic identification and trust services for electronic transactions in the internal market and repealing Directive 1999/93/EC, OJ L 257, 28.8.2014, p. 73–114.

[13] In addition to the qualified electronic signature, the advanced and the 'ordinary' electronic signature exist. The use of these types of signatures does not require a trusted third party. However, the eIDAS Regulation does not determine that these types of signatures automatically have the same legal effect as wet ink signatures. See for an analysis under English law S. Mason, ‘Documents signed or executed with electronic signatures in English law’, [2018] Computer Law & Security Review 933 and the sources mentioned therein.

[14] The term certification service provider (CSP) comes from the old Electronic Signatures Directive and was used to define a service provider issuing certificates related to electronic signatures (article 2 under 11 Electronic Signatures Directive). This term will be used because it is a more specific description than the more generic term 'qualified trust service provider' used in the eIDAS Regulation (article 3 under 19 eIDAS Regulation).

[15] This can be done, for example, by having a person drop by, establishing that his/her face matches the photo on his/her passport and comparing the written signature he/she places on a piece of paper with the signature in his/her passport.

[16] See e.g. articles 156 paragraph 3, 157 paragraph 2 and 151 of the Dutch Code of Civil Procedure, which provides that a document signed with a wet ink signature is presumed to be true, subject to counterproof.

[17] See https://en.wikipedia.org/wiki/DigiNotar.

In contrast, as will be shown below, smart contracts do not place such emphasis on the supplier, a TTP and centralisation, but instead on code and decentralisation. This begs the question to what extent such e-commerce laws can nonetheless help support the rise of smart contracts and if not, what is needed to facilitate this. That question will be also be discussed below.

3. Smart contracts make their appearance

At a time when the blockchain did not yet exist, the creator of smart contracts, Nick Szabo, defined smart contracts in 1994 as follows: "*A smart contract is a computerized transaction protocol that executes the terms of a contract.*" and added: "*The general objectives of smart contract design are to satisfy common contractual conditions (such as payment terms, liens, confidentiality, and even enforcement), minimize exceptions both malicious and accidental, and minimize the need for trusted intermediaries*."[18] Take the example of smart property that Szabo also mentioned at the time: a car purchased on instalments that cannot be driven if the buyer has not fulfilled his payment obligations at the agreed times. Without blockchain, that wish would be implemented as follows. The seller has a car immobiliser installed in the buyer's car. Before the car starts, the immobilizer checks (using mobile data transmission) with the vendor's bank (or, if neutrality is desired, with a TTP on whose escrow account the buyer is required to pay) whether the buyer has paid everything he was required to pay at that moment. If that's the case, the car starts. If that's not the case, the car doesn't start. This type of implementation requires trust in the seller (that he does not abuse the technology by instructing the immobilizer not to start) as well as trust in his bank or TTP (that he correctly determines whether all required payments have been received).

The innovative aspect of smart contracts on the blockchain in combination with cryptocurrency is that the execution of the contract (the execution of the performance as well as the corresponding transfer of value, i.e. payment) takes place automatically and decentrally. Unlike with centralised actions of one supplier, his bank or his TTP (server-based, web 2.0), execution on the blockchain takes places decentrally (peer-to-peer, web 3.0). As a result, the ideas of Szabo, who wrote his article when there was no blockchain, can be put into practice in a much better manner. A simple example may clarify.

Someone would like to play a game called 'guess the number under ten' with someone else. The game works as follows: two people pay the same amount of money to the smart contract at the start of each game, the smart contract chooses a random number, and the players guess as long as one of them guesses the number chosen by the smart contract. The person who guesses the number gets his stake back and wins the other's stake. If the game were to be played via a (centralised) internet site, the players would have to have a lot of confidence: in each other that the other person also pays his stake and in the gambling company operating the internet site that it pays both stakes to the winner and does not cheat. Also, transaction costs should be paid by both parties to the gambling company and by the gambling company to its payment service provider (PSP), which takes care of receiving and paying money. Furthermore, both players are out of luck if the gambling site is hacked or down (single point of failure). By using a smart contracts platform, e.g. Ethereum, that trust is not required, the transaction costs can be reduced and there is no single point of failure. Ethereum works by means of a combination of cryptocurrency and smart contracts in a manner which I will illustrate below.

Before doing so, I note that the example has been deliberately chosen because it is simple and the entire contract is stored and executed on the blockchain (on chain). Complexities that arises by allowing the execution of the smart contract to be dependent on interaction with the physical world (e.g. by requiring input from a sensor or a third party) can therefore be ignored for now. Of course, matters become more challenging when smart contracts do interact with the physical word. An example could be to automatically have a smart contract pay a seller the cryptocurrency previously paid by the purchaser into such contract if a GPS-chip attached to a container sends a signal to the smart contract that the container has arrived in the port of destination. Or if a smart contract pays passengers a set amount of cryptocurrency paid into such smart contract by the airline carrier upon

[18] Although it is difficult to find a reliable source for Szabo's article, many assume that it can be found here: http://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/smart.contracts.html.

receipt of a signal that a flight has been delayed for x hours. However, discussing the complexity resulting from these examples is best left for now because explaining and understanding how smart contracts on the blockchain work by means of the simple gambling example is already quite a challenge. Nonetheless, some of the complexities of the shipping and airline flight examples will be revisited at a later stage.

3.1 The cryptocurrency part

Ether is required for the use of Ethereum smart contracts. Ether is a cryptocurrency (just like bitcoin[19]), which, like other cryptocurrencies, functions by means of blockchain technology explained below.[20] Ether is held in a so-called Externally Owned Account (EOA, comparable to a bank account, hereinafter also referred to as: "account") with a unique address (comparable to a bank account number) to which and from which Ether can be transferred.[21] The access to each account is secured by means of a key pair, consisting of a public and a private key. This key pair is saved in a so-called key file, a text file that can be opened and viewed in a text editor. The private key portion of the key file is encrypted and can only be decrypted using a password that is chosen when creating the account. To access an account and subsequently transfer Ether, two things are needed: the keyfile (something you have) and the password with which the private key part of the keyfile is encrypted (something you know). The unique address to which and from which Ether can be transferred, consists of the last 20 bytes of the public key.

If someone wants to transfer Ether to an account, he transfers it to one of those unique addresses. Only the person who has both the keyfile and the password with which the private key part of the keyfile is encrypted can access the Ether transferred to that account. So far, the transfer from one Ethereum account to another is similar to the transfer of money from one bank account to another with two-factor authentication. This also requires: something you have (smartphone & app) and something you know (password/PIN code). There is, however, one important difference between the two, and that has to do with the fact that opening accounts and transferring Ether from and to accounts takes place decentrally in the blockchain, instead of centrally on the servers of one or more banks. That difference requires a little more explanation.

The Ethereum blockchain keeps track of the current status of each account, as well as all transactions between those accounts. This is done in a decentralised general ledger. That ledger runs on all computers that participate in the Ethereum network and those computers are called nodes. All these nodes in turn run so-called client software,[22] including the so-called Ethereum Virtual Machine (EVM). The EVM ensures, among other things, that the general ledger is synchronised between all nodes. Therefore, each node shares the same reality, called a shared single source or truth. In order to synchronise, a peer-to-peer network protocol is used. As soon as someone wants to execute a transaction, say transfer x Ether from account a to account b, it must be determined whether account a

[19] For a simple explanation of bitcoin, see https://medium.freecodecamp.org/explain-bitcoin-like-im-five-73b4257ac833. Bitcoins, ethers and other cryptocurrencies can be exchanged for conventional fiat money at a so-called cryptocurrency exchange, see https://en.wikipedia.org/wiki/Cryptocurrency_exchange.

[20] The legal qualification of cryptocurrencies from a contract and property law perspective is a complicated issue. Some argue that a bitcoin is an (absolute) property right, but subsequently run into trouble with the numerus clausus rule (i.e. the closed system of property rights) because a bitcoin (right) is not specifically mentioned in statute and is therefore not transferable, see with respect to Dutch law: F.H.J. Mijnssen, *Verbintenissen tot betaling van een geldsom (Mon. BW nr. B39)*, Kluwer 2017, paragraph 1.6 whilst referring to art. 3:83, paragraph 3 Dutch Civil Code. As a consequence, some argue, the transfer of bitcoins takes place exclusively on the basis of contract law, is to be considered a ‘Realakt’ and does not result in a change to the property law positions, see with respect to German law C. Engelhardt & S. Klein, ‘Bitcoins – Geschäfte mit Geld, das keines ist. Technische Grundlagen und zivilrechtliche Betrachtung’, [2014] MMR 355. I believe that the physical form of the wallet holding the private key by means of which bitcoins can be transferred is a documentary intangible, see T.J. de Graaf, ‘The qualification of bitcoins as documentary intangibles’, to be published this year in the European Review of Private Law (ERPL).

[21] http://www.ethdocs.org/en/latest/account-management.html#accounts.

[22] There are about eight different implementations of clients, see http://ethdocs.org/en/latest/ethereum-clients/choosing-a-client.html.

contains at least x Ether and if so, if x Ether can be validly transferred from account a to account b. This is done by using miners. Miners are nodes which fully automatically package together several transactions in a block and automatically solve a complicated cryptographic puzzle and in doing so strive to be the first whose block is added to the block chain. The miner which solves the puzzle first[23] (winner of the speed competition) wins the right to propose adding the mined block to the blockchain. He is therefore called a proposer.

Subsequently, the other nodes get a chance to prove that the proposer is not doing the right thing (in a quality competition).[24] If one of the nodes proves that the proposer is wrong (and thus wins the quality competition), the process starts all over and all nodes have the chance to become a proposer. If at least 51% of the nodes agree that the proposer is right, there is consensus. At that moment, that transaction (together with the other verified transactions in that block) is added to the blockchain as a block. That block is immutable in the sense that each block refers back to the previous block by means of a so-called hash pointer. As a result, every change in a block is immediately visible and such a change will not be accepted. The proposer whose block is added to the blockchain is rewarded for this by receiving (i) new Ether from the network (by means of a so-called block reward) and (ii) gas,[25] say transaction costs, from those whose transactions are in the block added to the blockchain.

This decentralised form of achieving consensus ensures that Ethereum transactions are almost flawless, the data stored in the blockchain cannot be changed and Ethereum is almost never down. After all, if one or many nodes is hacked or down, the remaining nodes continue to operate and go on with their work as if nothing is wrong. This is similar to what happens to the internet when an internet node is down, but this is fundamentally different to what happens when a bank is hacked or down. In the latter case, there is a risk that the bank's account holders (temporarily or even permanently) lose access to their money. Up to now, this description applies to virtually every cryptocurrency, including bitcoin. What makes Ethereum special and distinguishes it from a cryptocurrency sec, is the smart contracts part.

3.2 The smart contracts part

The Ethereum platform offers the possibility of having a smart contract executed fully automatically and decentrally on a peer-to-peer basis. Applied to the 'guess the number under ten' game, this works as follows. A programmer programs the game as a smart contract in, for example, Solidity (a programming language specifically designed for developing smart contracts) or another user-friendly programming language modelled on an existing programming language such as JavaScript or Python.[26] The code in which the programmer has programmed the game is called the source code or, in Ethereum-speak, the contract source. Once the contract source of the smart contract is finished, the programmer uses software (a compiler) to convert (i.e. compile) the contract source to a code (called EVM bytecode) that can be executed by the Ethereum Virtual Machine (EVM) on all nodes.[27] That EVM byte code is then, usually with the aid of a browser (the Mist browser), deployed in the blockchain, as a result of which it will be decentrally stored on all the nodes. The EVM byte code and,

[23] In practice, miners work together in so-called mining pools and share the profit they receive if their pool wins.

[24] I have not been able to determine whether Ethereum, like bitcoin, works with a quality competition, but I assume that that is the case.

[25] Gas is a cryptofuel whose price is determined on the basis of the available computing power of the nodes. Gas can be purchased with Ether. Gas is owed in order to reward nodes for verifying transactions, but also in order to ensure that the nodes are not overloaded by DDoS attacks (the simultaneous bombing of a node with nonsense traffic by several computers at the same time) or by having to calculate infinite loops (infinitely performing the same calculation).

[26] For an experiment to translate the buyer's suspension right into code, see T.F.E. Tjong Tjin Tai, 'Formalizing contract law for smart contracts' (September 18, 2017)', Tilburg Private Law Working Paper Series [2017-6], https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3038800.

[27] http://ethdocs.org/en/latest/contracts-and-transactions/contracts.html#what-is-a-contract.

almost always also the contract source, are public.[28] Anyone can review that code and judge whether the code is correct and will be executed in a correct manner.

It is important to realise that a so-called contract account is programmed into every smart contract.[29] On a contract account, just like on an EOA, Ether can be received and transferred. Transfer of Ether from an EOA to a contract account is the first step needed to initiate the execution of a smart contract on the Ethereum blockchain. Nothing happens without such first transfer. In our example, the smart contract only 'starts' when one of the participants has transferred his wager (in Ether) from his EOA to the smart contract and this wager has been received on the smart contract's contract account. When that wager is received, the smart contract awaits the receipt of the wager (in Ether) from the other participant, after which the game begins. Once the wager is deposited in the smart contract, only the execution of the code can result in payment. In our example, the code transfers the wagers of both participants to the EOA of the person who has won the game. And because the code is self-executing and runs decentrally on the blockchain, it cannot be stopped, neither by the programmer of the smart contracts code nor by the participants in the game.[30] The smart contract transfers the Ether which was won from its contract account to the EOA of speed and quality competition described above have been executed correctly and consensus has been reached. As a result, this form of gambling on the blockchain is also called provably fair gambling.

That the code executes itself and cannot be stopped is described by Lessig as 'code is law'[31] and by Wright and De Filippi as 'lex cryptographia'.[32] This is important to emphasise: once Ether is deposited in a smart contract, then only the execution of the self-executing code determines whether and if so, when and to which EOA Ether is transferred. So, with an EOA, a person can transfer the Ether in that account, whilst with a smart contract, that 'right' cannot be exercised by a person, but only by code. As a result of such self-executing mechanism, parties using smart contracts have no need for penalty clauses incentivising proper performance.[33]

4. Trust in who and what?

The fact that smart contracts run decentrally on the blockchain and are self-executing, has many consequences for the trust required when concluding a contract. Any contract is concluded between one person or legal entity and another person or legal entity. That is true now and does not change when concluding a smart contract on the blockchain. What does change, however, is in whom or what the contracting parties need to place their trust. Traditionally, the contracting parties had to trust each other. In a simple sales agreement, the purchaser would have to trust the seller that he would deliver the product purchased on the date, in the quantity and with the quality agreed upon. And the seller would need to trust the purchaser would take receipt of the product and pay the price agreed upon. By contrast, if parties conclude a smart contract on the blockchain and the fulfilment of the each parties obligations takes places entirely on the blockchain, each contracting party need only trust that the smart contracts code performs in accordance with the respective party's requirements. A party need not trust that the other party fulfils its obligations because the consequences of fulfilment and non-fulfilment are pre-programmed into the smart contract and once the smart contract executes, there is

[28] The code can be checked at https://etherscan.io. Although it seems possible to deploy the EVM byte code without the contract source on the blockchain, that is not in accordance with the intent behind the Ethereum blockchain because it is only possible to verify whether the code is correct and will be executed correctly with the help of the contract source.

[29] http://ethdocs.org/en/latest/account-management.html#accounts.

[30] Stopping is only possible if a possibility to do so has been programmed in the smart contract code and that is something that, as we will see below, is not compatible with the philosophy of a smart contract.

[31] L. Lessig, *Code and Other Laws of Cyberspace*, Basic Books 1999 and L. Lessig, *Code version 2.0*, Basic Books 2006, http://codev2.cc. See also R.H. Weber, 'Rose is a rose is a rose is a rose – what about code and law?', Computer Law & Security Review [2018] 701.

[32] A. Wright & P. De Filippi, 'Decentralized blockchain technology and the rise of lex cryptographia', [2015] 1 https://ssrn.com/abstract=2580664 and P. Filippi & A. Wright, *Blockchain and the Law. The Rule of Code*, Harvard University Press 2018.

[33] P. Cuccuru, 'Beyond bitcoin: an early overview on smart contracts', International Journal of Law and Information Technology [2017] 179.

no way of stopping it. In the simple gambling example, if the code has been programmed correctly, the smart contract only starts to self-execute if both parties have paid an equal amount of wager into the contract account. Upon the winning party having guessed the correct number, the smart contract fulfils the losing party's obligation to pay the winning party by paying the wagers into the winning party's EOA. Neither party is able to prevent the obligations from being fulfilled in the aforementioned manner between the moment the wager paid in and out of the contract account. As a consequence, traditional contracts require the contracting parties to trust each other and smart contracts require the contracting parties to trust the code. Trust in people therefore shifts to trust in code, as will be explained in more detail below.[34]

4.1 Trustless of people and context

Seeing that for the correct execution of a smart contract trust in people is no longer needed (neither in the counterparty nor in an intermediary such as a bank, guarantor or (other) TTP), smart contracts are also referred to as trustless. That is true if it refers to trustless with respect to the contracting parties that they do what they agreed to. After all, if the code is correct, it does not matter whether the other party is reliable or not. The difference between trust in people (called trust by communication) and trust in code (called trust by computation) is also reflected in the various ways in which public and private key pairs are used on the internet and in the blockchain. In public key infrastructure (PKI), the public and private key pair that the eIDAS regulation is based on, the emphasis is on a centralised TTP (i.e. the certification service provider) to authenticate individuals (verify that they are who they say they are). In the blockchain, however, no third party (in the form of a person) needs to authenticate a person. From Ethereum's perspective, this is not necessary, because for the safe receipt and transfer of Ether it is only relevant that there is an EOA to which and an EOA from which Ether can be transferred and that those EOA's may be accessed through the use of the private key associated with each respective EOA, not whether the person presenting the private key is who he says he is.

From the point of view of legal certainty, this confidence in code is a good thing; how smart contracts code executes itself is, after all, more predictable than how people act. By no longer making human intervention possible whilst the smart contract executes, parties are afforded more legal certainty than, for example, by using an abstract first demand bank guarantee, the traditional instrument of choice used as payment security when minimal human discretion is desired. At first sight, such a bank guarantee offers a lot of security. After all, the bank can and may only check whether the documents submitted by the beneficiary of the bank guarantee (e.g. a statement of default) correspond formally (at face value) with the requirements set out in the guarantee. The bank can may only pay out when all requirements included in the guarantee are satisfied.[35] However, the payment of a bank guarantee remains dependent on the bank and its conduct, something hard-core smart contract proponents will despise because no payment is made if the bank goes bankrupt or if it decides, correctly or incorrectly, not to pay. In other words, because a smart contract pays independently of available assets of a third party or its (potential) randomness, it offers more (payment) security than an abstract bank guarantee on first demand.

The reasons why a bank does not pay out can, however, be quite valid. A bank does not have to, and in fact may not, pay out an abstract bank guarantee on first demand in case of fraud, deceit or arbitrariness.[36] This brings us to another reason why smart contracts provide more (payment) security than traditional instruments such as a bank guarantees: decontextualisation. When assessing whether fraud is involved, a bank will look at the context which is (often) fed by its customer: the

[34] For an analysis of trust in the networked era, see E.L.O. Keymolen, *Trust on the line. A philosophical exploration of trust in the networked era*, doctoral thesis Erasmus University Rotterdam, Wolf Productions 2016, https://repub.eur.nl/pub/93210 and for trust in code Stephen Mason and Timothy S. Reiniger, '"Trust" Between Machines? Establishing Identity Between Humans and Software Code, or whether You Know it is a Dog, and if so, which Dog?', Computer and Telecommunications Law Review [2015] 135.

[35] R.F. Bertrams, *Bank Guarantees in International Trade. The Law and Practice of Independent (First Demand) Guarantees and Standby Letters of Credit in Civil Law and Common Law* Jurisdictions, Kluwer Law International 2013, chapter 13.

[36] R.F. Bertrams, op. cit., chapters 14-16.

circumstances beyond the literal text of the bank guarantee. Or a judge does so in summary proceedings and requires or prohibits the bank to pay. This dependency is also non-existent in a smart contract. The code follows its 'if this, then that' structure: if the 'this' is met, then the ‘that’ automatically follows or, in the example of the 'guess the number under ten' game: as soon as a player has guessed the number, the wagers of both players are paid to him. The circumstances of the case are irrelevant, even if there is fraud.

This begs the question why a legitimate company, other than a criminal, would use such a mechanism. The answer to this question depends in whom or what the contracting parties place more trust. In case of an abstract bank first demand guarantee the party putting up the bank guarantee places trust in people: (i) the other party that he does not misuse the guarantee by calling it when the actual circumstances do not so justify, (ii) the bank that it does not collude with the calling party and (iii) the judiciary that litigation, if required, will result in a fair judgement (whatever that may be). In case of a smart contract on the blockchain, parties place trust in the code that it is programmed in such a way that a fraudulent party cannot misuse it. The ultimate difference between the two is that in case of a smart contract on the blockchain, whether it is a good idea to conclude a contract and letting it self-execute is something which can be objectively determined by verifying the code, whereas in case of traditional contracts whether it is a good idea to conclude a contract and having parties (and trusted third parties) perform their obligations is something which can only be subjectively determined by trusting people. Which of the two is preferable is very much in the eye of the beholder and will be influenced, amongst other things, by the reputation, culture and rule of law of the people and countries concerned (in case of traditional contracts) and tech-savviness of the contracting parties (in case of smart contracts on the blockchain).

4.2 Trustful of code, but not in case of The DAO hack?

Back to the creed 'code is law'. If we find that creed to be acceptable, it presupposes that we can determine whether the code is correct. That lays bare another problem: how do we know that the code is correct? Of course, smart contracts are fully transparent: everyone can check their code, but who has the expertise to do so? In the example of the 'guess the number under ten' game, this is not that difficult for an average programmer. However, as the complexity of the smart contract increases, it will become more and more difficult to check that code, especially for the average user. So that user will most likely engage a third party to check that code. Ironically, that is precisely what blockchain adepts do not want with their disintermediation mantra: trusting a third party. And yet trust in third parties will likely also play a role in smart contracts: in advance to check code and, in exceptional cases, afterwards to reverse unwanted consequences.

Take The Decentralised Autonomous Organization (The DAO) developed by Slock.it. The DAO was an organisation that existed exclusively as an Ethereum smart contract and came to life on 30 April 2016. In the The DAO smart contract, investors were able to deposit Ether in exchange for tradable tokens with which they acquired control (Initial Coin Offering (ICO)[37]). As soon as enough Ether was collected, The DAO really started to function. A company seeking to attract funding would submit a business plan and if the majority of the votes would approve the business plan, the smart contract would automatically transfer Ether to the EOA of that company. However, a bug in the smart contracts code was used/abused by a hacker, as a result of which on 17/18 June 2016 one third of the funding (3.6 million Ether) was withdrawn from The DAO and parked in another smart contract, the so-called child DAO.[38] Pursuant to the Ethereum smart contract’s code, that Ether could, however, only be transferred from the child DAO to another EOA after a moratorium of 28 days.

A salient detail, however, is that the makers of The DAO code, Slock.it, proudly announced in their blog of 5 April 2016 that one of the world's leading security audit companies, Deja Vu Security, had conducted a security review of the The DAO smart contracts framework, whereby "no stone was left untouched during these five whole days of security analysis."[39] This does not mean,

[37] https://bitcoinmagazine.com/guides/what-ico/.

[38] https://www.coindesk.com/understanding-dao-hack-journalists/ and in a general sense https://en.wikipedia.org/wiki/The_DAO_(organization).

[39] https://blog.slock.it/deja-vu-dao-smart-contracts-audit-results-d26bc088e32e.

however, that Deja Vu Security overlooked anything. After all, it is unclear what the scope and outcome of the audit was. Also, Deja Vu Security let it be known that for confidentiality reasons, it would not communicate particulars without written consent from Slock.it and we therefore have no means of knowing their side of the story.[40] It is apparent, however, that Slock.it engaged a third party to check its code and, by blogging about the audit, tried to convince others that the code was safe.

Although the Ethereum protocol was not compromised in the The DAO hack, the Ethereum foundation (Ethereum's maker) took it upon itself to retrieve the 'stolen' Ether. After having established that a less invasive soft fork was not a good idea for safety reasons,[41] the foundation came up with a hard fork that would transfer all the Ether in The DAO and the child DAO to a new smart contract, the WithdrawDAO recovery contract.[42] The only purpose of the new smart contract would be to receive and subsequently pay back all Ether to the token holders (with an exchange rate of 100 DAO tokens = 1 Ether). That hard fork would be implemented through updates of the Ethereum clients (software that, as mentioned, runs on all nodes and with which the nodes participate in the peer-to-peer Ethereum network).[43] The Ethereum Foundation asked the community to vote on the automatic rollout of that hard fork using Carbonvote, voting software which was developed ad hoc and with which every Ether holder could vote for or against the hard fork with one vote per Ether. 87% of the votes were cast in favour of the hard fork,[44] the fork was rolled out and 85% of the miners were mining on the new fork very rapidly.[45]

The way in which the hard fork was proposed, decided upon and rolled out reveals a number of problems. Most attention was paid to the battle between, on the one hand, those who categorised the hack as abuse, wanted to oppose it and saw the hard fork as justice and, on the other hand, those who did not see the hack as abuse, but as smart use of transparent code and the hard fork as abuse of the system. Parallels were drawn with the banking crisis and the bail-out of banks that were too-big-to-fail. The prevention of another crisis and the subsequent bail-out was an important reason to develop the blockchain in the first place. The irony is that the Ethereum hard fork very much looked like a bail-out. Questions were also raised about the way in which the vote was announced (blog post, Twitter and Reddit), the duration of the vote (24 hours) and the percentage of the total number of Ether holders that cast a vote.[46]

From a legal point of view, what's intriguing is not so much the justness of the outcome, but the way the dispute between the hacker and the rest was resolved.[47] Apparently, the Ethereum foundation and the clients' software developers considered themselves entitled to change the rules of the game after having acquired the backing of a part of the community (whether representative or not) and possibly in a doubtful manner. Then it was up to the nodes of the network to decide to install the update if they agreed with that decision and wanted to continue working according to the new rules of

[40] "*Hi Everyone, Adam Cecchetti CEO of Deja vu Security here. For legal and professional reasons Deja vu Security does not discuss details of any customer interaction, engagement, or audit without written consent from said customer. Please contact representatives from Slock.it for additional details*." https://www.reddit.com/r/ethereum/comments/4ota1q/the_truth_about_the_security_audit_stephen_tual/.

[41] The soft fork was suggested in https://blog.ethereum.org/2016/06/17/critical-update-re-dao-vulnerability/ and https://blog.ethereum.org/2016/06/24/dao-wars-youre-voice-soft-fork-dilemma/, and would cause the 'stolen' Ether to be frozen as a result of which it could not be spent. The security problem is described in http://hackingdistributed.com/2016/06/28/ethereum-soft-fork-dos-vector/ and https://blog.ethereum.org/2016/06/28/security-alert- dos-vulnerability-in-the-soft-fork/.

[42] For a general explanation of forks, see https://www.coindesk.com/short-guide-bitcoin-forks-explained/ and for an explanation of the Ethereum DAO hard fork https://www.coindesk.com/understanding-dao-hack-journalists/.

[43] For the most widely used client (go-ethereum or geth) released by the Ethereum foundation see https://blog.ethereum.org/2016/07/15/to-fork-or-not-to-fork and for Slock.it's support for the hard fork see https://blog.slock.it/what-the-fork-really-means-6fe573ac31dd.

[44] http://v1.carbonvote.com.

[45] https://blog.ethereum.org/2016/07/20/hard-fork-completed/. Some of the nodes that did not agree with this fork founded Ethereum Classic and continued under the creed code is law: https://ethereumclassic.github.io.

[46] https://bitcoinmagazine.com/articles/op-ed-why-ethereums-hard-fork-will-cause-problems-coming-year/.

[47] See also W. Reijers, I.S. Wuisman, M. Mannan, P. De Filippi, C. Wray, V. Rae-Looi, A.Cubillos Vélez & L. Orgad, 'Now the Code Runs Itself: On-Chain and Off-Chain Governance of Blockchain Technologies', [2018] *Topoi* https://doi.org/10.1007/s11245-018-9626-5.

the game, or decide not to install it if they did not agree with that decision and wanted to continue according to the old rules of the game. It is also remarkable that the Ethereum foundation deemed the votes of Ether holders decisive, not votes of the The DAO token holders and the companies who asked for funding (the participants The DAO, i.e. the contracting parties). Bearing in mind Barlow's heartfelt cry, quoted above, there is an argument to be made in support of the self-regulating power of this new form of technology (in this case for self-regulation by the Ethereum foundation, the community, the client software's software developers and the nodes), but from a legal point of view this form self-regulation is (almost) unjustifiable. Law exists to prevent people taking the law into their own hands and, if necessary, protect a minority against a majority as well as to ensure at the very least that decision-making procedures are executed in a fair and just manner.

Because of the justness of the outcome, it will quickly be assumed that the end justified the means, but the risks inherent to this way of self-regulating intervention should not be neglected. These risks can be made more apparent by asking a number of (partly rhetorical) questions about a more difficult scenario. What would have happened if a small investor were to lose his stake by a hack and from which loss the majority of the other investors would have benefited? What prevails in that case: a sense of justice or the mantra code is law? And would the foundation, community, developers and nodes consider intervening in this case? If so, how would they want to legitimise such intervention? Knowing this, how much trust would small investors have in such a DAO and what does that mean for the credibility and (hence) the success of that DAO?

In any case, it becomes apparent that even in the blockchain world code is not always law and, in an exceptional case, the execution of a smart contract can be halted by third parties. Trust in code was suddenly, somewhere halfway down the line, replaced by trust in the foundation, community, developers and nodes. It also become apparent, and that is more worrying, that the legitimacy of the self-regulatory intervention is questionable from a legal point. The participants in the The DAO contract saw that their dispute with the hacker was resolved not by a judge, but by the (majority of the) foundation, the community, developers and nodes.

5. Applicability of internet laws

With the foregoing in mind, are existing internet laws sufficiently tailored to apply in a meaningful way to smart contracts?[48] This question is topical because the European Commission announced that it wants to create the right enabling environment for blockchain.[49] As soon as something new comes into being, it is pertinent to first investigate whether it suffices to apply existing laws. From a legal point of view, this requires an investigation into whether existing internet laws can be applied to smart contracts. In essence, as shown above, these laws aim to eliminate the (presumed) disadvantage that a customer has when purchasing on the internet when compared to purchasing from a brick-and-mortar store as well as strengthen the buyer's confidence in the supplier and his actions. This is done, in part, by obliging the supplier to provide his (potential) customers with all sorts of information about himself and the manner of contracting and, also in part, by obliging him to structure his contracting process in a certain way as well as grant consumers the right of withdrawal (pursuant to the E-Commerce Directive and the Consumer Rights Directive). Furthermore, the (presumed) uncertainty that parties have in relation to the legal validity of electronic documents with respect to written documents is also solved with the help of laws, more specifically: the E-commerce Directive and the eIDAS regulation). This is done by requiring the use of a third party (TTP) in order to be able to

[48] See also C. Millard, 'Blockchain and the las: Incompatible codes?', Computer Law & Security Review [2018] 843, who rightly notes on p. 846 that "reconciling new technologies with established legal principles and regulatory models is often messy and protracted."

[49] Commissioner for the Digital Economy and Society Mariya Gabriel commented: "I see it as a game changer and I want Europe to be the forefront of its development. The EU Blockchain Observatory and Forum is an important step in that direction.", press release European Commission, European Commission launches the EU Blockchain Observatory and Forum, 1-2-2018 , http://europa.eu/rapid/press-release_IP-18-521_en.htm. In the UK, the government is considering "reviewing the current legal and regulatory framework for ensuring that it facilitates the use of smart contracts.", see section 2.39 https://www.lawcom.gov.uk/project/13th-programme-of-law-reform/. In the Netherlands, the government is also experimenting with blockchain, see https://www.blockchainpilots.nl.

automatically equate an electronic document with an electronic signature to a written document with a wet ink signature.

Many commercial parties that wish to sell products or services on the internet have an interest in complying with those laws. Traditionally, they sell more when buyers trust them. And one way to gain trust, is by providing information about yourself and by complying with internet laws. However, there is no (or less of a) need to do so with smart contracts. Because smart contracts execute themselves, trust in the code is important, not trust in the supplier. If the supplier feels no need to comply with these laws and (therefore) also does not provide information about himself, enforcement by courts of law becomes difficult, if not impossible. And if the supplier has no physical address and his assets are unknown, it is difficult to litigate against him and execute his assets if he is ordered by a court to pay a sum of money. In addition, smart contracts automatically execute themselves and, for technical reasons, the customer therefore has no way of exercising his rights until the moment the smart contract has been executed. This makes the exercise of suspension and similar rights illusory. As long as the implementation of the smart contract occurs entirely on chain, parties can evade laws and authorities relatively easy. In that case, the wish of Barlow will come true.

The question to be answered is not whether laws apply to smart contracts on the blockchain or not, because they obviously do, but whether the contracting parties have a practical need (or, on the contrary, are disincentivized) to provide information about their identities. In the gambling example, the conclusion and fulfilment of all obligations (gambling and payment) takes place entirely on the blockchain and there is no need for contracting parties to provide information about their identities. Parties may even be disincentivized to provide such information if gambling is illegal in their countries of residence, they do not want to give a bad loser the opportunity to come knocking at their door or they are adamant about their privacy. As more and more especially recreational activities are conducted entirely online, e.g. gaming and augmented reality, there may be less of a need to provide personal information (because this is not necessary in order to enjoy those activities) and more of a need by some to protect their privacy. Some smart contracts on the blockchain may even be considered an enabling technology to avoid laws from being enforced. In the absence of a personal information and information about assets, contracting parties will not be able to litigate offline and also regulatory authorities and the judiciary will be hard-pressed to regulate parties and enforce laws. Again, laws do apply to smart contracts, but the enforcement of laws may be extremely difficult if not impossible if the smart contract is concluded and fulfilled entirely on the blockchain. As a law-abiding citizen, I do not endorse this development, on the contrary, but merely point out its consequences.

Of course, once smart contracts interact with the physical world, matters are different.[50] The carrier delivering the container to which the GPS chip is attached may physically prevent the smart contract payment from being triggered by not entering the port of destination, and the airline’s aeroplanes may be arrested if the smart contract does not execute as the passengers had envisaged. In those cases, trust in the code is required (that it self-executes in accordance with what parties expect), yet because parties are able to physically frustrate such self-execution that trust in code is in dire need of being supplemented with trust in people. And that trust in people pre-supposes those people, i.e. the contracting parties, provide information about themselves in order to be able to litigate against each other. That information can be provided in a number of different ways. Either parties conclude a contract offline and implement a part of their contract in a smart contract, e.g. sign a written sales contract with wet ink and program into a smart contract a mechanism which, once triggered, pays cryptocurrency into a designated EOA. In that case internet laws do not apply because the contract is not concluded and/or executed entirely online, but only a part of it is executed by means of and on the blockchain.[51] Or parties conclude the entire contract by means of the blockchain, in which case all

[50] In that case, smart contracts do not exist in a legal vacuum, just as cyberspace is not cut off from the real world, as expressed by Martin von Haller Groenbaek, 'Blockchain 2.0, smart contracts and legal challenges', [2016-June/July] SCL magazine, https://www.scl.org/articles/3668-blockchain-2-0-smart-contracts-and-legal-challenges.

[51] The B2B and B2C E-Commerce Directive applies to one-to-many information society services and not to the case at hand of one-to-one electronic executions of one-to-one contracts concluded offline, see recital 18 and the definition of information society services in the E-Commerce Directive (art. 1 paragraph 2 and art. 2 under (a)

internet laws need to be abided by and the contract will be concluded in the same we as we are accustomed to when concluding a contract with a webshop. Conceptually, a lot of ‘passive’ contractual terms (such as liability, choice of law, jurisdiction clause and (other) boiler plates) could be incorporated into the contract as well as statutory information requirements be fulfilled by including them in the source code of the smart contract as comments. Yet doing so would presumably not constitute valid offer and acceptance of such contractual terms or be in line with the statutory requirements with respect to the manner such information should be provided as set out in art. 5 and 10 of the E-Commerce Directive (i.e. “easily, directly and permanently accessible” and “clearly, comprehensibly and unambiguously prior to the order being place” respectively) and art. 6 of the Consumer Rights Directive (i.e. “clear and comprehensible manner”). Alternatively, a click-wrap agreement could be programmed into the smart contract, the acceptance of which would be required before the contract may be fully concluded and executed.

Anyway, going back to an entirely on chain situation, the absence of personal information and information about assets prevents the use of the standard repertoire of remedies and enforcement. In light of this, it is (more) important to determine how confidence in code can be increased. After all, if code is law, then there is a risk that code-savvy persons abuse code-naive persons[52] or, put in a more neutral manner, that there are errors in the code that lead to undesirable consequences which are difficult if not impossible to undo. In order to diminish these risks, it is obvious that we should focus on the quality of the code. After all, if the code does what the parties expect, problems should not arise and there will be no need to correct them afterwards. I realise that code is almost never free of errors,[53] but if the reality is that code is law, the less errors there are and the less impact they have, the less susceptible contracting parties will be to abuse or unexpected consequences. To that extent, the attention of traditionally solving problems before, during or after the implementation of the contract will shift to solving them prior to their occurrence, in the pre-contractual phase. This means that programmers should develop better smart contracts together with lawyers. This also means that verifying the reliability of the code is crucial. Since most customers cannot do this themselves, they will want to call in a third party, an auditor. In that respect, a part about auditing audit of smart contracts code could be added to the eIDAS Regulation.[54] Also, the customer's lack of recourse against his supplier could be compensated by introducing a liability regime towards the auditor if the audit proves to have been incorrectly conducted. Furthermore, a new regime could be added to the eIDAS regulation that allows smart contracts on the blockchain to which cryptocurrency has been transferred using key pairs without using a TTP, are automatically equated with written contracts with written signatures insofar it concerns legal validity,[55] either with or without an unqualified audit opinion by an auditor.[56] In case an auditor is involved, the following scenario seems most likely: he first tests and approves the code, and only after such approval the code may be used for real transactions. This reminds me of acceptance testing in case of software development. Ironically, by means of these proposals, we are introducing a TTP (the auditor), whilst those introducing smart

E-Commerce Directive and art. 1 paraph 1 under (b) Directive 2015/1535 replacing Directive 98/34/EC). The B2C Consumer Rights Directive applies to a large extent to contracts concluded at a distance (which does not apply to the case at hand) and where the directive applies regardless of the manner in which it is concluded, it can be abided by in the offline contract and will not impact the execution of the smart contracts part (see e.g. the information requirements set out in art. 5 of the directive).

[52] See in a warning way Bill Marino and Ari Juels, "Setting standards for altering and undoing smart contracts", Rule technologies. 10th International Symposium, [2016] RuleML 151 https://link.springer.com/content/pdf/10.1007%2F978-3-319-42019-6.pdf and Marcella Atzori, 'Blockchain technology and decentralized governance: is the state still necessary?', [2015] SSRN https://ssrn.com/abstract=2709713.

[53] As Stephen Mason and Daniel Seng, editors, *Electronic Evidence*, University of London 2017, http://ials.sas.ac.uk/digital/humanities-digital-library/observing-law-ials-open-book-service-law/electronic-evidence, nrs. 6.57 et al aptly demonstrate and explain.

[54] See E. Valgaeren and J.J. Linnemann, 'Blockchain ontketend’, [2017] Computerrecht 346, who believe that the blockchain may make some or all of the eIDAS rules redundant, but that aspects that are regulated by that eIDAS regulation can serve as a source of inspiration for regulating certain aspects of the blockchain.

[55] Insofar as a legal form requirement applies to the contract in question.

[56] The way in which these ideas can be implemented is beyond the scope of this article.

contracts seek to get rid of TTP's. Perhaps such a TTP can itself eventually consist of a smart contract deployed on the blockchain.

The alternative would be to let internet laws apply one-on-one to smart contracts and the parties involved. If suppliers comply with them, the possibilities for customers to correct input errors and to turn to suppliers in the event of non-performance would increase. However, in many cases suppliers have, as mentioned, little or no interest in complying with these laws and customers and suppliers cannot enforce these laws or will have great difficulty enforcing them. After all, suppliers are often unknown (as a result of which it is not or very difficult to litigate against them) and their assets are often unknown (so that execution of court judgments is not possible or very difficult). In those cases, it is better to increase the trust in the code by means of the aforementioned audits and the aforementioned regime of equating, rather than to rely on enforcement which is not possible or hardly possible.

6. **Conclusion**

We saw that smart contracts differ fundamentally from e-commerce contracts concluded via the internet. From a legal perspective, it is especially relevant that in the internet era, a lot of trust is placed in people and their actions (in particular the supplier, banks and (other) trusted third parties). By means of internet laws, the customer is given means to act against a supplier breaching contract and electronic is equated with written in order to comply with formal requirements. Through smart contracts, that trust in people is replaced by trust in code. However, that code cannot be understood by the average user. The DAO hack illustrates that if things go wrong, people (will have to) intervene, but that there are all kinds of technical and legal difficulties with this solution. Although it may be attempted to apply existing internet laws on smart contracts, they are difficult to apply in a meaningful manner. This is due to the difference in starting points: trust in people versus trust in code. That is also due to technical and practical obstacles: for smart contracts stored and executed entirely on the blockchain it is technically almost impossible for parties who have not provided information about themselves or their assets to intervene *prior* to execution of the smart contract and *after* execution it almost impossible or very difficult to litigate against a stranger as well as find and execute assets previously unknown. All in all, when using smart contracts, it makes more sense to prevent problems from arising than to correct them afterwards. For this reason, I advocate that programmers work together with lawyers to create better smart contracts and that the legislator focuses on laws regarding auditing smart contracts code by trusted third parties and automatically equating smart contracts with written contracts with wet ink signatures. Hopefully, this will facilitate the rise of permissionless smart contracts on the blockchain.