Smart contracts in the Netherlands

A legal research regarding the use of smart contracts within Dutch contract law and legal framework

Master Thesis International Business Law

Writer:	Ruben (R.W.H.G.) Schulpen
SNR:	U1253885
Master:	International Business Law (LLM)
Year:	2018
Supervisor:	L. Tavecchio

Acknowledgements

Every expectation that I had before starting my second Master of International Business Law (LLM) at Tilburg University has been surpassed this year.

With this acknowledgement, I would like to address and thank all those who helped with the creation of this thesis. Since it is impossible to mention all those who have contributed to this work, I will limit myself to those who were most important.

First of all, I would like to express my gratitude towards my supervisor Luca Tavecchio for all the useful comments, remarks and engagement through the learning process of this Master thesis. Beside my supervisor, I also want to use this opportunity to express my thankfulness towards all of the staff members of the Department of International Business Law of Tilburg University, for their devotion and passion by composing an incredible, innovative and outstanding Master's Program.

Last but definitely not least, I would also like to thank my loved ones, who have supported me throughout entire process, both by keeping me harmonious and helping me putting pieces together.

Enjoy reading my Master thesis!

Ruben Schulpen
Tilburg, August 2018

Table of contents

Introduction

Blockchain technology and smart contracts: while some consider these technologies as the biggest and most disruptive innovations since the introduction of the Internet,[1] others argue that it can be considered as a "technological bubble" which will burst within a short period of time.[2] Despite the different views regarding these innovations, it cannot be denied that a considerable amount of attention has been paid to these innovations within academic literature. Many governmental institutions, commercial entities and financial institutions are already experimenting with applications of these technologies and implementing them in existing work processes. Not surprisingly, the year 2017 has been the year in which the discussion about blockchain technology and its applications became a trending topic in many sectors.[3] As a result of the considerable amount of attention that has been paid to blockchain technology within the legal sector, 2017 can be seen as the year of blockchain technology.

While 2017 can be seen as the year of blockchain technology, the year 2018 will be the year of the "smart contracts" according to the literature and articles within the legal sector.[4] Essentially, a smart contract is a user-defined program that specifies rules governing transactions and that is enforced by a network of peers. Smart contracts carry the promise to eliminate the need of intermediaries and their associated transaction costs.[5] Although the idea of smart contracts is not relatively new (the term smart contracts was coined in the early 1990's by Nick Szabo[6]), the concept of smart contracts is starting to emerge in the legal world nowadays. The emergence of smart contracts is closely intertwined with the rise of blockchain technology since it allows its participants to have a platform on which they can create, verify and perform smart contracts. As a result of the growing interest in smart contracts, the upcoming years will be very interesting for smart contracts in the Netherlands according to the Dutch Blockchain Coalition (hereafter: "DBC") in their recent *"First reconnaissance of questions relating to legislation, regulations and future knowledge needs as a consequence of blockchain technology and more specifically, smart contracts".*[7]

---

[1] See for example D. Tapscott, "*How blockchain could change the world*", May 2016 (https://www.mckinsey.com/industries/high-tech/our-insights/how-blockchains-could-change-the-world) and see also L. Mearian, "*What is blockchain? The most disruptive tech in decades*", May 2018 (https://www.computerworld.com/article/3191077/security/what-is-blockchain-the-most-disruptive-tech-in-decades.html)

[2] See report from research firm GlobalData ("*Blockchain – Thematic Research*"), where they argue that the new technology's bubble "*will burst in the next two years [and] will have lost much of its gloss by 2025*." (https://martechtoday.com/a-new-report-bursts-the-blockchain-bubble-216959)

[3] See for example M. Mamoria, "*2017- The year blockchain went mainstream",* 2018 (https://hackernoon.com/2017-the-year-blockchain-went-mainstream-119863d5d9f3*)* or S. Friend*, "2017: The Year Blockchain Got Weird",* January 2018 (https://www.coindesk.com/2017-year-blockchain-got-weird/).

[4] See for example C. Sproule, "*As smart contracts get smarter, the rules of development will change*", 2018 (https://venturebeat.com/2018/02/18/as-smart-contracts-get-smarter-the-rules-of-development-will-change/)

[5] K. Delmolino, et al, "*Step by Step Towards Creating a Safe Smart Contract: Lessons and Insights from a Cryptocurrency Lab*", Financial Cryptography and Data Security - FC 2016 International Work- shops, BITCOIN, VOTING, and WAHC, Christ Church, Barbados, February 26, 2016, Revised Selected Papers. pp. 79–94 (2016)

[6] N. Szabo, "*Smart Contracts: Building Blocks for Digital Markets*", Extropy nr. 16 (1996)

[7] Dutch Blockchain Coalition, "*Smart contracts as a specific application of blockchain technology*", 2017 https://www.dutchdigitaldelta.nl/uploads/pdf/Smart-Contracts-ENG-report.pdf

Consequently, it can be said that smart contracts will face an interesting period in the upcoming years[8]. The attention that has been paid to smart contracts on the Internet and academic writing is worth mentioning. Despite of the attention that has been paid to smart contracts within academic writing, many legal questions arise in the Netherlands with regard to this new way of contracting. Most of the global organizations, law firms and governments cannot get around the developments with regard to smart contracts. Furthermore, some scientific studies have also been published about the legal impact and consequences of smart contracts. Unfortunately, the academic studies that have been published about smart contracts in the Netherlands are still seeking for answers on some (legal) questions, mostly due to the fact that smart contracts are still in its infancy and therefore no uniform answers can be found to these questions.

While the DBC made a first step in mapping out several follow-up steps and recommendations regarding smart contracts in the Netherlands, many questions remain unanswered. The Dutch Parliament also determined that further investigation to new topics as smart contracts and blockchain technology are necessary for the future with respect to their impact on society, regulation and legislation.[9]

Research question and sub-questions

Taken into consideration the abovementioned facts, this thesis will focus on answering the legal questions that arise with regard to smart contracts in Dutch (contract) law and examine what should be done regarding the legal framework to successfully implement smart contracts in the Dutch legal system. Consequently, the central research question will be: ***"Will smart contracts replace traditional ways of contracting and will the Netherlands (and its legal framework) be ready for smart contracts in the future?"***

Although the central research question will be the common theme throughout the whole thesis, several sub-sections dealing with legal and technical issues regarding smart contracts have to be discussed in order to come to a conclusion. As a result, several sub-questions will be answered and further analyzed in this thesis.

Therefore, the first chapter of this thesis will analyze and answer the sub-question *"What are smart contracts and how do they differ from traditional contracts?"*. In order to explain smart contracts and their underlying technology, it is necessary to make a short side-step to the explanation of blockchain technology. Although several technical aspects will be discussed and analyzed in the first chapter, the technological analysis (i.e. non-legal analysis) aims to remain broad and generic. Beside the explanation of the concept of smart contracts and their relationship with blockchain technology, several other "features" of smart contracts (like *automatability* and *immutability*) will be discussed in the first chapter. The first chapter

[8] See for example https://www.forbes.com/sites/rogeraitken/2017/11/21/smart-contracts-on-the-blockchain-can-businesses-reap-the-benefits/#61a9857c1074. According to research and advisory company Gartner, more than 25% of the global organizations will use smart contracts in 2022.

[9] Kamerstukken I, 2016/2017, 33 009

will be concluded with a final paragraph about the advantages of smart contracts. In addition, several reconsiderations with respect to the use of smart contracts will be discussed.

After defining the concept of smart contracts in the first chapter, the second chapter will discuss the second sub-question: *"What are the legal implications of smart contracts within Dutch contract law?"*. The first subject that will be discussed is whether (and to what extent) law can be subsumed in code under Dutch Law. Moreover, one of the paragraphs will be dedicated to the discussion of "code *is* law" or "code *as* law". Secondly, the question will be answered how Dutch contract law looks upon smart contracts: in other words, can smart contracts be seen as legally binding contracts under Dutch contract law? Furthermore, the legal system of the Netherlands will be analyzed with respect to the interpretation of smart contracts under Dutch Law and how potential disputes have to be dealt with in a way that also serves the purpose and nature of smart contracts.

The third chapter will analyze and answer the last sub-question: *"What are the regulatory challenges regarding the use of smart contracts in the future within the Dutch legal system?"*. The analysis will focus on how the Netherlands can solve the problems that can be expected with the use of smart contracts. The discussion will commence with the explanation of the legal and technical obstacles regarding smart contracts. The following paragraphs examine several subjects that are useful for the future application of smart contracts, like best practices and ethical guidelines. The subsequent paragraph will examine the difficulties of translating existing Dutch law into smart contract applications and discuss several options that might be used to overcome the translating difficulties. The third chapter will conclude with the analysis of the Dutch judicial system and long-term effects on the so-called "trust market".

To conclude this thesis, several recommendations and concluding observations will be made in order to answer the central research question.

Research methods

In order to answer the central research question, the focus will be mainly on the research of international (legal) academic literature about smart contracts and blockchain technology. For explaining smart contracts, blockchain technology and their underlying technology, several *technical* academic studies will be used. Compared with most of the legal academic literature, these technical studies give a more detailed description of the way in which smart contracts are used and designed. Consequently, the use of these technical studies will contribute to a better and broader understanding of smart contracts. However, as mentioned before, this technical analysis aims to remain broad and generic in order to be comprehendible. For this reason, the choice has also been made to add an Annex in which the programming of a smart contract will be clarified and explained by using an example smart contract written in Solidity.

For the analysis of smart contracts within Dutch contract law the focus will be on Dutch academic literature, case law and Parliamentary History. This will contribute to an understanding of Dutch contract law and underlying motives that contributed to the introduction of specific laws and regulations.

Chapter 1: Smart contracts

1.1: Definition of smart contract: the humble vending machine

According to a study of the US National Association of Purchasing Managers in 2003, the average Fortune 1000 company had anywhere between 20,000 and 40,000 active contracts.[10] Nowadays, the number of active contracts within a company continue to increase each year. All these contracts need to be managed since they are the lifeblood of any enterprise. Apart from the financial constraints for efficient contract lifecycle management are companies also exposed to significant legal and compliance risks in the execution and administration of contracts. As a result, many of these contracts are better served by a technical approach. Automation of contracts could be a solid solution to the problem of the becoming increasingly difficult contract management. Imagine that a contract is self-executing: the contract will, for example, trigger the payment automatically as soon as the delivery has taken place or the contract itself will automatically index its own rates when exchange rates fluctuate heavily. These simple examples of self-executing contracts, also known as "smart contracts", are in theory far more functional than their inanimate paper-based ancestors, if only for the fact that it will decrease the load on a company's contract management.[11]

The concept of smart contracts is not relatively new, although the growing attention that has been paid to smart contracts in the legal business last couple of years, suggests otherwise. The idea of smart contracts was coined in the early 1990's by computer scientist, lawyer and cryptographer Nick Szabo. Over the years, Szabo published several articles and papers with regard to the concept of smart contracts. According to one of his early papers, a smart contract can be described as *"a set of promises, specified in digital form, including protocols within which the parties perform on these promises."*[12] Another definition of a smart contract was described in another publication of Szabo where he described smart contracts as *"digital, computable contracts where the performance and enforcement of contractual conditions occur automatically, without the need for human intervention".*[13]

Szabo compared the concept of smart contracts with a vending machine of soft drinks: *"When the money is paid, an irrevocable set of actions is put in motion. The money is retained and a drink is supplied. The transaction cannot be stopped in mid flow. The money cannot be returned when the drink is supplied. The transaction's terms are in a sense embedded in the*

[10] PricewaterhouseCoopers LLP, "*Contract management: contract value and minimize risks: A necessary response to the overwhelming quantity and diversified complexity of contracts*": p.2

[11] N. Szabo, "*Smart Contracts: Building Blocks for Digital Markets*", Extropy nr. 16 (1996)

[12] N. Szabo, "*Smart Contracts: Building Blocks for Digital Markets*", Extropy nr. 16 (1996) (http://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/smart_contracts_2.html)

[13] N. Szabo, "*Formalizing and Securing Relationships on Public Networks*", First Monday 1997, vol. 2 nr.9

hardware and in the software that runs the machine."[14] In its core, the humble vending machine is the original form of a smart contract according to Szabo.[15][16]

Except for the publications of Szabo[17] and few others, it can be concluded that the concept of smart contracts stayed out of the spotlights for several decades, partly due to the lack of a supporting technology. Nevertheless, with the emergence of blockchain technology, the attention for smart contracts also changed. It is beyond this thesis to fully explain the whole technology of blockchain, however it is important to sketch the important main lines of this technology.

1.2: The change of smart contract: code, blockchain and Ethereum

Blockchain, a distributed database and computing technology managed in a decentralized manner (often autonomously), firstly became well-known as the technology behind the cryptocurrency Bitcoin in the beginning of 2008.[18] Like smart contracts, the idea of a cryptographically secured chain of blocks dates back to 1991 but it was only until 2008 that the first blockchain was conceptualized by the "mysterious" Satoshi Nakamoto[19], and was implemented and popularized through the cryptocurrency Bitcoin.[20]

The simplest form of blockchain technology entails a distributed database that autonomously maintains a continuously growing list of public records in unit of "blocks", secured from tampering and revision. Each block contains a timestamp and a link to a previous block.[21] The central aim of blockchain is creating a database system that parties can jointly maintain and edit in a decentralized manner, with no individual exercising central control.[22] One defining feature of blockchain is the ability to maintain, in a relatively cheap way, a uniform view on the state of things and the order of events – a (decentralized) consensus, a type of information that leads individuals or organizations with divergent perspectives and incentives to accept and act upon as if it is truth.[23] This *decentralized* consensus is perhaps the most important aim of blockchain which eventually makes it possible that states or sets of

---

[14] M. van Eersel & T. van den Bergh, *"Blockchain en smart contracts: toegang tot een reeks van slimme dingen"*, FRP 2017/457 (afl. 4)

[15] N. Szabo, "*Smart Contracts: Building Blocks for Digital Markets*", Extropy nr. 16 (1996)

[16] Szabo used this reference also in his foreword in *"Smart Contracts: 12 Use Cases for Business & Beyond A Technology, Legal & Regulatory Introduction"*, Chamber of Digital Commerce, December 2016

[17] Szabo also published a draft guide for setting up smart contracts: (N. Szabo, "*A Formal Language for Analyzing Contracts"*, 2002

[18] L.W. Cong & Z. He, "*Blockchain Disruption and Smart Contracts*", University of Chicago Booth School of Business (2017), p. 2

[19] It is still uncertain who exactly the creator of Bitcoin is. See an example of the "witch-hunt": https://medium.com/cryptomuse/how-the-nsa-caught-satoshi-nakamoto-868affcef595

[20] L.W. Cong & Z. He, "*Blockchain Disruption and Smart Contracts*", University of Chicago Booth School of Business (2017), p. 7

[21] L.W. Cong & Z. He, "*Blockchain Disruption and Smart Contracts*", University of Chicago Booth School of Business (2017), p. 7 but also J. Linneman, "*Juridische aspecten van (toepassingen van) blockchain*", Computerrecht 2016/218, afl. 6, p. 319

[22] L.W. Cong & Z. He, "*Blockchain Disruption and Smart Contracts*", University of Chicago Booth School of Business (2017), p. 7

[23] L.W. Cong & Z. He, "*Blockchain Disruption and Smart Contracts*", University of Chicago Booth School of Business (2017), p. 7

information have to be agreed upon by all agents via rules and protocols, without the need to trust or rely upon a centralized authority.[24]

The technology behind blockchain has various possible applications. One of the possible applications which often has been a topic of discussion is the concept of decentralized digital currencies like Bitcoin or Ether. Beside these decentralized digital currencies, it is also possible to implement digital codes in a computerized transaction protocol that automatically executes the terms of a contract in case the obligations have been fulfilled. These automated codes that have been implemented in this context are called "smart contracts". Worth mentioning is the fact that Bitcoin was the first to support basic smart contracts: in short, the network can transfer value from one person to another in case certain conditions are met but only if the network of "nodes"[25] can validate the transaction.[26] For example: if someone wants to send two Bitcoins to another person, the nodes review the entire Bitcoin network to validate that the Sender really has two Bitcoins in his wallet, and has not already sent them to someone else. Once that information is confirmed ("*if…then*"), the new transaction is going to be included in a "block". At this point, the new block is attached to the previous block. These blocks are related to each other and form a "chain", hence the name "block*chain*". In the aforementioned example, the previous block contains the information regarding the Sender and his Bitcoins, while the new block contains the related information regarding the transaction and the Receiver of the Bitcoins. Every block within a blockchain contains a "hash function" and a so-called "hash pointer" that refers to the previous block within the chain and can be seen as a reference to the location of data.

The explicit goal of Bitcoin's blockchain was creating and recording transfers of a blockchain network's native token (Bitcoin). However, programming applications using Bitcoin's scripting language was too difficult and too limited because of its resemblance to machine code. Machine code is a strictly numerical language that is designed to be executed directly or as fast as possible by a computer but incredibly difficult to read or program directly by humans. See for example this Bitcoin's scripting language[27]:

OP_DUPOP_HASH16062e907b15cbf27d5425399ebf6f0fb50ebb88f18OP_EQUALVERIFYOP_CHECKSIG

Bitcoin's scripting language was therefore discouraging for developers, user-unfriendly and also not "*Turing-complete"*.[28] Every computation which can be carried out, can be expressed and implemented in a Turing-complete programming language. Because of this, Turing

[24] L.W. Cong & Z. He, "*Blockchain Disruption and Smart Contracts*", University of Chicago Booth School of Business (2017), p. 7

[25] Large network of computers.

[26] https://www.coindesk.com/information/ethereum-smart-contracts-work/

[27] P. Kasireddy, "*Bitcoin, Ethereum, Blockchain, Tokens, ICOs: Why should anyone care?",* July 2017 https://hackernoon.com/bitcoin-ethereum-blockchain-tokens-icos-why-should-anyone-care-890b868cec06

[28] A "Turing Complete" programming language allow to compute anything computable given enough resources. See for more extensive explanation what Turing-completeness is: https://hackernoon.com/ethereum-turing-completeness-and-rich-statefulness-explained-e650db7fc1fb

completeness enables a computer to 'loop' and process its own output in repeating complex terms.[29] Bitcoin's programming language was lacking this ability and could only effectuate "simple" payments.

This changed with the introduction of the first Turing-complete blockchain, *Ethereum*. Ethereum is a cryptocurrency invented by Vitalik Buterin and was launched in 2015. It was built from the ground up using its own blockchain network and was designed to be a more generalized protocol than Bitcoin's blockchain.[30] As written in Ethereum's white paper, the intent of Ethereum was *"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(…) 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."*[31]

The Ethereum blockchain contains just like Bitcoin's blockchain a log of transaction-like events in which users send the network's native token (Ether) using the log. Unlike the Bitcoin programming code, Ethereum's programming language does not look like machine code but is similar to other common programming languages.[32] A simple example of a Solidity contract (a contract-oriented programming language for writing smart contracts) of creating and sending cryptocurrency would look like the example below, based on the programming language used in Ethereum (for a brief explanation of the Solidity example contract below, see Annex 1 of this thesis):

```
pragma solidity ^0.4.21;

contract Coin {
    // The keyword "public" makes those variables
    // readable from outside.
    address public minter;
    mapping (address => uint) public balances;

    // Events allow light clients to react on
    // changes efficiently.
    event Sent(address from, address to, uint amount);

    // This is the constructor whose code is
    // run only when the contract is created.
    function Coin() public {
        minter = msg.sender;
    }

    function mint(address receiver, uint amount) public {
        if (msg.sender != minter) return;
        balances[receiver] += amount;
    }

    function send(address receiver, uint amount) public {
        if (balances[msg.sender] < amount) return;
        balances[msg.sender] -= amount;
        balances[receiver] += amount;
        emit Sent(msg.sender, receiver, amount);
    }
}
```

[29] See for historical explanation https://hackernoon.com/smart-contracts-turing-completeness-reality-3eb897996621
[30] P. Kasireddy, "*Bitcoin, Ethereum, Blockchain, Tokens, ICOs: Why should anyone care?",* July 2017
[31] See whitepaper of Ethereum online at https://github.com/ethereum/wiki/wiki/White-Paper/f18902f4e7fb21dc92b37e8a0963eec4b3f4793a
[32] P. Kasireddy, "*Bitcoin, Ethereum, Blockchain, Tokens, ICOs: Why should anyone care?",* July 2017

With the introduction of Ethereum, a generalized framework was created for running any type of code on the blockchain in a more "easy" way compared to Bitcoin. Smart contracts are an example of applications which can now be built in a much easier and user-friendly way than in Bitcoin. Since a smart contract is a distributed contract that is represented in code and needs to clarify in its code "*if this happens then do that*"[33], it has become easier for programmers to code smart contracts with Ethereum's programming language than Bitcoin's "machine code" programming language. Unlike Ethereum's programming language, it can be assumed that it is almost impossible to program a smart contract directly with Bitcoin's machine code and at the same time clarify what automatically needs to happen if several obligations have been fulfilled. Smart contracts can accept and store Ether and data and can send that Ether to other accounts or other smart contracts. Essentially, the emerge of Ethereum made it in theory possible that smart contracts could be created much easier by programmers.[34]

It is noteworthy that smart contracts do not per se need a decentralized blockchain network to work, but they do need an underlying trusted network or mechanism, which blockchain provides conveniently and efficiently.[35] In contrast to the decentralized blockchain network, smart contracts can also work on a *centralized* trusted network where a trusted central party or authority is in control over several actions, like making changes or record the transactions (i.e. not immutable).[36] However, in the following of this thesis it will be assumed that smart contracts are stored on the blockchain network because of the fact that under such network, upon a triggering action, the network is designed to automatically produce outputs without third-party intervention, which makes the entire network system reliable and tamper-proof (i.e. unchangeable) and not dependent on the parties' trust in each other.[37] For this reason, it can be argued that this assumption is better for the common understandability of this thesis. In case the underlying trusted network or mechanism will differ (i.e. not a blockchain network), there is going to be a clear notice.

As a result of the current use of smart contracts in relation to blockchain technology and the emergence of Ethereum as platform to design smart contracts, it can be argued that the definition that Szabo used in his early publications not seem to perfectly fit any longer and need a minor addition, especially regarding the decentralized trusted network that blockchain provides. At the same time, there is still no unambiguous and widely shared definition of smart contracts in academic literature in relation to blockchain technology. In the context of blockchain technology, smart contracts can be described as follows: "*Pre-written computer*

[33] P. Kasireddy, "*Bitcoin, Ethereum, Blockchain, Tokens, ICOs: Why should anyone care?",* July 2017

[34] Until today, the "breakthrough" of the use of smart contracts stayed out.

[35] Allen & Overy LLP, "*Smart Contracts for Finance Parties*", http://www.allenovery.com/publications/en-gb/lrrfs/cross-border/Pages/Smart-contracts-for-finance-parties.aspx

[36] Such network has more in common with the traditional centralized database.

[37] Clifford Chance, "*Smart Contracts (Code vs. Contract)",* January 2018 (https://talkingtech.cliffordchance.com/en/tech/smart-contracts--code-vs-contract-.html)

logic/code, stored and replicated on a distributed storage platform (like Blockchain), executed by a network of computers and can result in changes of the ledger."[38]

It is important to note that despite of the use of blockchain technology, not all the smart contracts are created equal. While there might certainly be smart contracts which completely consist out of computer code, other smart contracts which have "real effects" make full automation not achievable at least for now. In particular, should be noted, that sometimes the term "smart contract" is used to identify a specific technology - code that is stored, verified and executed on a blockchain -, while on the other hand, the term is used to refer to a specific *application* of the technology: as a complement, or substitute, for legal contracts.[39] Some authors[40], therefore, have proposed to make a distinction between "smart contract code" (which identify specific technology) and "smart legal contracts" (which specify the application of the technology).[41] However, this thesis will focus on smart contracts as "contracts" under the Dutch contract law .

This distinction is required since the term "smart *contract*" can be confusing. After all, there are certainly different kinds of use of the abovementioned technology that does not constitute a *contract* in legal sense.[42] This issue will be further analyzed later in this thesis together with the state of the art of Dutch contract law.

1.2.1: Permissionless and permissioned blockchain

Previous sub-section made clear that blockchain technology provides a convenient and efficient underlying network for smart contract to work on. However, a noteworthy aspect about blockchain technology with regard to smart contracts that has to be distinguished, is the distinction between so-called "*permissionless*" and "*permissioned*" blockchain, also known as "public" and "private" blockchain[43]. Although both are basically the same in the way they store the data, by means of building up cryptographically linked blocks, many important differences can be distinguished which might have an impact on the participation of parties and legal identification[44].

A public blockchain can be best described as a blockchain where everyone is free to participate anonymously. This means that everyone who is willing to participate, can do so immediately as a normal user or as a so-called "full node".[45] Since public blockchains are open for everybody without exception, every participant can download it to their computers and

[38] J.B. Schmaal & E.M. van Genuchten, "*Smart contracts en de Haviltex-norm*", Tijdschrift voor Internetrecht, nr.1 maart 2017: p. 12. They based their definition on the description of BitsonBlocks-blogger, Anthony Lewis.
[39] J. Stark, "*Making Sense of Blockchain Smart Contracts*", June 2016: https://www.coindesk.com/making-sense-smart-contracts/
[40] For example: C.D. Clack, V.A. Bakshi & L. Braine, "*Smart Contract Templates: foundations, design landscape and research directions*", August 4 2016: p. 2
[41] A. de Backer & V. de Boe, "*Smart contracts in de financiële sector: grote verwachtingen en regulatoire uitdagingen*", Computerrecht 2017/252
[42] Dutch Blockchain Coalition, "*Smart contracts as a specific application of blockchain technology*", 2017: p.12
[43] Hereinafter, I will use the terminology of "public" and "private" blockchain.
[44] See paragraph 2.3 for further legal identification of smart contracts under private or public blockchain.
[45] Dutch Blockchain Coalition, "*Smart contracts as a specific application of blockchain technology*", 2017: p. 14

view the entire history of the blockchain, send and receive digital currencies, store information, and even create smart contracts within the blockchain. The most well-known public blockchains are Ethereum and Bitcoin. Furthermore, public blockchains are characterized by the fact that no identification or authentication concerning the parties (participants) is required, who are virtually anonymous on the public blockchain.[46] To perform transactions, two different sets of so-called cryptographic key pairs are used: a public key and a private key, the latter is personal for each participant and should remain secret while the other, as the name suggest, is public. All information within the blockchain is public and all of the participants can make proposals: although every participant can make a proposal within a public blockchain, an update or proposal will only take place if a majority of the participants agree with it.[47] This leads to the situation that no single participant can be seen as the central authority within the blockchain (i.e. decentralized).

Private blockchains, on the other hand, are not accessible to everyone. Private blockchains are protected by a so-called "access control layer".[48] In case someone wants to participate in a private blockchain, that participant will need to request for approval. Where a public blockchain does not have a central authority, a private blockchain has something similar to a central trusted authority (which might be labelled as a "Trusted Third Party") to whom the participants have to ask allowance for access: writing-privileges and reading-privileges may also differ between the participants.[49] In contrast to public blockchains without any central authority, private blockchain carry out an identification and authentication of each participant. In short, the Trusted Third Party can be described as an independent entity or person that offers services that increase the reliability of the smart contracts by applying security measures: for example, identification or authentication of parties.

1.3: Differences and similarities regarding traditional agreements

Smart contracts are, as with many technology-derived solutions, the product of computer programming. These computer programs create and enforce an agreed upon performance between parties, much like the traditional paper-based contracts. The most obvious difference is that smart contracts are computer-generated and thus it is the code itself that explains the obligations of the parties. While most of the traditional contracts are written in the native language of the country in which the contract has been drafted, it can be said that smart contracts consist out of programming language of which most contracting parties and even programmers themselves, may not have a complete understanding of. As discussed in the previous paragraph, the programming language of a smart contract is very difficult and hard to understand without a certain basic knowledge about coding.[50]

[46] Dutch Blockchain Coalition, "*Smart contracts as a specific application of blockchain technology*", 2017: p. 14. Although anonymous as in "pseudonyms" would be better.
[47] Dutch Blockchain Coalition, "*Smart contracts as a specific application of blockchain technology*", 2017: p. 14
[48] Dutch Blockchain Coalition, "*Smart contracts as a specific application of blockchain technology*", 2017: p. 14
[49] In contrast to public blockchains, it might in theory even be possible that all of the data will be stored on a single node.
[50] See Annex 1 of this thesis for a brief explanation of the programming language Solidity.

While in most countries the contractual written language of contracts has been developed through decades of legislative developments and case law, the language of smart contracts (or rather: code) is still in its infancy. As a consequence of this infancy, it can be said that the legislation is based on traditional ways of contracting and not smart contracting. The legislation in the Netherlands requires in some cases that a third party, like a notary, is involved in order to validate the transaction. For example, whenever someone wants to transfer the shares of a Dutch private company to another party, the law requires that the contract is accompanied by a notarial deed.[51] The same applies to buying a house within the Netherlands, where the contract has to be in writing, registered in a public register and drawn up and signed by a civil-law notary established in the Netherlands.[52] The whole essence of a smart contract, where third parties are not involved with the self-executing contract, will be subverted with aforementioned legislation. Consequently, smart contracts are not applicable to all cases in which written contracts are required under Dutch law. This is because the law is simply not applicable for non-paper-based smart contracts.

Despite the obvious differences in design and language, it can be argued that smart contracts are not significantly different from traditional/written contracts with regard to their *purpose*. Similar as in traditional contracts, parties determine the agreements between both parties in advance which on their turn are then translated into mutual obligations that can be made dependent on the occurrence of certain triggers. Although the drafting of written contracts is significantly different from the programming of scripts, it can be said that a good contract lawyer wants to cut the agreements as much as possible in binary parts that can only be explained in one way, which make it almost similar to programming.[53] To put it in a simple way, the underlying idea is still the same: making agreements in advance and excluding several unwanted scenarios.

Nevertheless, smart contracts differ very much in execution and design in case these contracts are compared with the execution and design of traditional agreements. Traditional contracts are at their most basic level agreements coupled with the intention of the parties to be legally bound to that specific agreement. Smart contracts do have some key additional features that distinguishes them from traditional contracts: *automatability* and *immutability*. The meaning and effect of both terms will be explained separately in the following sub-sections.

[51] Article 2:196 Dutch Civil Code
[52] Article 7:2 and 7:3 Dutch Civil Code
[53] J.B. Schmaal & E.M. van Genuchten, "*Smart contracts en de Haviltex-norm*", Tijdschrift voor Internetrecht, nr.1 maart 2017: p. 14

1.3.1: Automatability of smart contracts

Traditional paper-based contracts are usually written[54] agreements between parties and contain the specific agreements where both parties need to comply with. While the majority of the contracts are *bilateral,* meaning that each of the parties have made a promise to the other, some other contracts can be seen as *unilateral*, where only one party makes a promise in exchange for an act by the other party.[55] People record their agreements since the beginning of writing in paper-based contracts to keep the other party to their promises. Therefore, contracts have an obligatory character from origin and reflect the promises that have been made by the concerning parties.

The execution of these contracts does, however, rely on the independent action of the parties who are part of the contract. The fact that the parties need to execute and comply to the contracts can be traced down to the deeply rooted moral duty *"when you make a promise, you have to keep it".*[56] However, the fact that parties made some agreements and promises to each other does not completely ensure that parties will keep their promises and comply with the agreements that have been made. Therefore, it is not surprising that most countries have a well-developed contract law in which disputes about execution of contracts is dealt with. In case a contracted party does not want to meet one of his obligations and breaches the contract, the other party can make an appeal on the legal obligation to fulfill the contract and eventually ask a court to give a verdict to enforce fulfillment of the agreed obligations.

Unlike traditional contracts where the execution relies on the independent action of the parties, it can be said that smart contracts are capable of performing (or at least partially performing) the agreements of the contracts by computers, without the parties' direct intervention.[57] As discussed in previous paragraphs, smart contracts consist out of programmed code in which the agreements are recorded. Consequently, the rights and obligations regarding the execution of the contracts are already implemented in the smart contract even before the smart contract is effectuated. These coded agreements can be seen as conditions and associated consequences: in case conditions A and B are met, consequence C will be carried out automatically by the smart contract. Smart contracts act, unlike traditional contracts, as autonomous agents that run entirely on the blockchain and "remove the human out of the loop"[58]. Smart contracts are therefore self-executing, purely on the basis of the instructions that are implemented in the code of smart contracts. In this context, the smart contract itself will ensure about the execution in case the preformatted criteria are met.

Since smart contracts are self-executing, it can be said that the contracting parties of a smart contract do not have to actually perform actions themselves to fulfill their contractual obligations and at the same time do not have to invoke their contractual obligations actively. Where traditional agreements sometimes end up in protracted conflicts because one of the

[54] To be clear, I will not take oral agreements into account in this thesis.
[55] Insurance policies are unilateral for example.
[56] E. Tjong Tjin Tai, "*Smart contracts en het recht*", NJB 2017/146: p. 178
[57] Clifford Chance Report, "*Are Smart Contracts Contracts?*", December 2017: p.3
[58] P. Kasireddy, "*Bitcoin, Ethereum, Blockchain, Tokens, ICOs: Why should anyone care?",* July 2017

parties does not want to fulfill his duties, with regard to smart contracts is this scenario in theory impossible. After all, a smart contract will automatically execute its performance (or not) solely on the basis of its pre-programmed instructions.

Furthermore, it is important to understand that a smart contract is different from a traditional contract performed via computers, when looking at an automated bank transfer for example. Most people nowadays use online banking and set up several automated bank transfers to other parties, so that they do not have to approach the bank every time they want to make a payment. A simple example is the payment of the monthly rent: in case a certain date has been reached at the end of the month, the bank will automatically transfer a predetermined amount to the landlord on behalf of the renter. However, the choice to eventually transfer the predetermined amount will still remain within the "sphere of influence" of the bank or renter, whereas either the renter or the bank[59] may intervene, terminate or change the payment at any time. Conversely, a smart contract is executed and performed autonomously, without the possibility for the parties to interfere or run the risk of being tampered with.[60] In essence, the code of the smart contract will simply carry out the transfer in case the automated transfer would have been implemented in a smart contract and makes an intervention of a third party unnecessary and impossible.[61] Accordingly, the autonomous nature of self-execution and performance rights and obligations are the fundamentals that make smart contracts "smart".[62]

The argument that is used many times, is that avoiding this intervention of third parties can be seen as a big advantage of using smart contracts. Not only does this save costs with respect to third parties but it also makes the contract less dependent on the interference of these third parties. Consequently, smart contracts will be less sensitive for whims of the other party. With regard to litigation, the reversal of the burden of litigation is a key distinction with a smart contract due to the automatability – a "wronged" party under a smart contract will have to sue to reverse performance and claim damages, instead of suing for failure of performance and a claim for damages.[63]

1.3.2: Immutability of smart contracts

Beside their self-enforcing nature, smart contracts also differ from traditional contracts in their immutability, or maybe better: *enforceability*. Due to the fact that agreements in smart contracts are embodied in programming language on a blockchain, it can be said that the

[59] The bank can choose to reject the payment because of several reasons, like for example a conflict with the duty of care of the bank.
[60] Clifford Chance, "*Smart Contracts (Code vs. Contract)",* January 2018 (https://talkingtech.cliffordchance.com/en/tech/smart-contracts--code-vs-contract-.html)
[61] J.B. Schmaal & E.M. van Genuchten, "*Smart contracts en de Haviltex-norm*", Tijdschrift voor Internetrecht, nr.1 maart 2017: p. 15
[62] Clifford Chance, "*Smart Contracts (Code vs. Contract)",* January 2018
[63] Allen & Overy LLP, "*Smart Contracts for Finance Parties*", http://www.allenovery.com/publications/en-gb/lrrfs/cross-border/Pages/Smart-contracts-for-finance-parties.aspx

concerning agreements are hardly or not changeable at all (unilaterally)[64] afterwards by the parties themselves.[65] The code of smart contracts will decide the operation and outcome of the contract during the whole duration of the smart contract, which means that the existing agreements within the smart contracts cannot be interpreted or carried out in a different way by external factors or third parties.

The binary design of the code of the smart contract prevents interpretations on the base of context or background. Consequently, contracting parties need to record the context or background of the agreements at the moment of drafting the smart contract. Although it can be argued that most traditional contracts also need to include these aspects at the moment of drafting, the context or background can also be established afterwards, by a court. In addition, the context and background of a contract are most of the time decisive for a judge in the Netherlands in their judgment, due to the well-known *Haviltex-formula.*[66] In a subsequently paragraph of this thesis will the Haviltex-formula and implications for smart contracts be further analyzed and explained.

The aforementioned immutability can be best described starting with the humble vending machine of Szabo. In case someone wants to buy a soda from the vending machine and puts a coin into the soda vending machine slot and enters the code of the desired soda, there should be no requirement for the vendor to step in to ensure that the transaction is performed.[67] The transaction should be *autonomous* and the vendor should have no further ability to *interfere* with the transaction ("tamper-proof"[68]).

The same principle can also be projected on a Bitcoin transaction: if a person submits a valid Bitcoin transfer instruction, the transaction is recorded autonomously on the Bitcoin blockchain and the blockchain will not be further modified until a subsequent valid Bitcoin transaction is entered into.[69]

Consequently, a smart contract needs to be capable of ensuring its performance in such way that it depends on the completion of an autonomous technological process ("if…then"), in order to have the property of tamper-proof execution like the vending machine and Bitcoin transaction.[70] What the definition of tamper-proof with respect to smart contracts entails, is that the enforceability of a smart contract may not necessarily lie in the fact that its performance can be enforced by a court, but, as an alternative sense of "enforceable", that it

[64] If general consensus exists among all nodes that something must be reversed, then that can indeed happen: see Ethereum's so-called hard fork of 2016 (DAO-hack)
[65] J.B. Schmaal & E.M. van Genuchten, "*Smart contracts en de Haviltex-norm*", Tijdschrift voor Internetrecht, nr.1 maart 2017: p. 14
[66] Supreme Court Netherlands (Hoge Raad), 13-03-1981, ECLI:NL:HR:1981:AG4158
[67] Clifford Chance Report, "*Are Smart Contracts Contracts?*", December 2017: p.3
[68] Description is used in Clifford Chance Report, "*Are Smart Contracts Contracts?*", December 2017: p.3
[69] Clifford Chance Report, "*Are Smart Contracts Contracts?*", December 2017: p.3-4
[70] Clifford Chance Report, "*Are Smart Contracts Contracts?*", December 2017: p.4

may be effected by an autonomous technological process, which cannot be interfered *unilaterally*[71] with by the parties, as soon as it is initiated.[72]

The fact that the programming language is leading and cannot be changed, results in an ideal where computer code governs the legal system: "*code is law*". This can result in two situations. Firstly, the situation concerning "code *is* law", where no human intervention will be possible and where in fact no more legal protection can be obtained than the code itself. In other words, only the code will be leading in that case. In the other situation, parties of smart contracts chose to regulate their legal relationship by code and not legal/written rules but still have the fundamental right to go to a court, since it is their fundamental right according to article 6 ECHR: "code *as* law".[73] The differences between both situations will be cited several times in the following of this thesis and analyzed extensively in paragraph 2.2.

1.4: Advantages of smart contracts

The features of automatability and enforceability distinguish smart contracts from the traditional paper-based contracts. Although that these features contain several advantages, it can be argued that smart contracts also have other advantages which are not related to these features. In the following sub-sections, other advantages of smart contracts will be analyzed at the current state of affairs but at the same time will also several reconsiderations be discussed[74].

1.4.1: Main advantages of smart contracts

In theory, smart contracts contain several advantages compared with traditional contracts. Firstly, the transaction costs of smart contracts are lower than the transaction costs which are involved with traditional contracts due to the fact that smart contracts are self-executing and no third parties are involved with the execution of the contracts[75]. Unlike traditional contracts, smart contracts do not need any involvement of banks, notaries or accountants due to the fact that the blockchain itself will guarantee that the party, which transfers a specific good for example, is the actual owner of the good.[76]

A consequence of the fact that smart contracts are self-executing, and no third parties are involved with the execution, is that smart contracts can be time saving with regard to the signing/execution of a deal (clearing and settlement). Since everything is done by a prescribed programming language, the execution and payment will be settled with as soon as the parties have fulfilled their duties of the smart contract. Where traditional contracts (and involved

[71] Dutch Blockchain Coalition, "*Smart contracts as a specific application of blockchain technology*", 2017: p. 16: "*If general consensus exists among all nodes that something must be reversed, then that can indeed happen, see Ethereum's so-called hard fork of 2016*" (DAO-hack)
[72] Clifford Chance Report, "*Are Smart Contracts Contracts?*", December 2017: p 3
[73] E. Tjong Tjin Tai, "*Smart contracts en het recht*", NJB 2017/146: p. 178-179
[74] I prefer this term instead of using the term "disadvantages".
[75] See also paragraph 1.3.1
[76] M. van Eersel & T. van den Bergh, *"Blockchain en smart contracts: toegang tot een reeks van slimme dingen"*, FRP 2017/457 (afl. 4): p. 44

parties) were limited by the human participation and associated time that was necessary to comply and execute a contract, it can be said that smart contracts need less time to execute since the execution of the smart contract is automated. The smart contract will be executed automatically as soon as the conditions, which are recorded in the code, are met.

Furthermore, smart contracts also entail more certainty with regard to several subjects beside only the time and cost saving aspect. As the previous paragraphs about the immutability and automatability already showed, it can be concluded that the contracting or third parties have no unilaterally influence on the execution of the smart contracts itself. After the smart contract has been activated on the blockchain, it will be almost impossible that it can be influenced by different interpretations or executions by an involved party, of which the outcome is not certain in advance. Smart contracts entail agent neutrality, in that sense that parties themselves cannot interpret the contract in their favor.[77] This results in compliance in *advance*: all contract parties know how the conditions of the smart contract will be met.[78] In traditional paper-based contracts, parties tend to explain the contractual obligations in their own favor afterwards and eventually execute the contract as soon as they believe that the conditions are fulfilled *according to their own belief*. With the use of smart contracts, this situation will almost be impossible due to the fact that the code itself will execute regardless of subjective criteria. Because of the fact that the code itself is deterministic, it will always generate the same output in case the obligations have been fulfilled as a result of the fact that the code is insensible for parties' interpretations or opinions. Parties know that with smart contracts, the contract will be executed as soon as the conditions are met. Parties cannot change anything about it whereby they have to take the eventual and predictable outcome of the contract into consideration.

Taken the aforementioned advantages into consideration, some proponents of smart contracts argue that smart contracts will take over traditional contracts since they increase efficiency, increase reliability and also reduce costs. However, despite some ambitious projects run by start-ups or financial institutions, no smart contract solution has yet significantly displaced some traditional way of contracting.[79] In the next sub-section, several reasons will be illustrated that explain why smart contracts did not replace traditional contracting so far.

1.5: Points of reconsideration regarding smart contracts

Smart contract technology is, despite its increasing interest, still in its infancy. Consequently, contracting parties still remain relatively unknowing about smart contracts, their underlying technology and their potential. A simple but logical cause for this can be related to the fact that contracting parties still want to see and understand what they have agreed in a (smart)

[77] S. Grybniak, "*Advantages and Disadvantages of Smart Contracts in Financial Blockchain Systems*", December 2017 (https://hackernoon.com/advantages-and-disadvantages-of-smart-contracts-in-financial-blockchain-systems-3a443145ae1c)

[78] M. van Eersel & T. van den Bergh, *"Blockchain en smart contracts: toegang tot een reeks van slimme dingen"*, FRP 2017/457 (afl. 4): p. 44

[79] Clifford Chance Report, "*Are Smart Contracts Contracts?*", December 2017: p.2

contract. Nowadays, most parties will not have enough reassurance about their agreements in case they have a programmed code in front of them and simply want to see a written (and paper-based) contract. The absence of basic understanding and legislative acceptance regarding programmed codes and smart contracts, play an important role in the fact that no smart contracts solution has yet significantly displaced some traditional way of contracting.

Another reason why smart contracts are still in their infancy, can be found in the fact that most of the advantages that have been discussed in the previous sub-sections, still need some more clearness and explanation. Since further consideration is therefore desirable, several points of reconsideration will be explained in the upcoming sub-sections in order to come to a more realistic image about smart contracts.

1.5.1: Involvement of third parties

According to most of the proponents of smart contracts, it is often assumed that one of the most desirable advantages of smart contracts is the fact that third-party agents will disappear. Consequently, the omitting of third parties will eventually lead to less transaction costs and less time for clearing and settlement of a contract. However, the actual situation is that third party agents do not completely disappear but will start playing a different role in the process. Where the involvement of some third-party agents like notaries, perhaps will diminish with regard to their contribution to some contracts, some other parties' involvement will on the other hand increase. Where traditional written contracts most of the time need contract lawyers in the drafting process, it can be assumed that programmers of smart contracts will be needed considerably more to draft smart contracts and need to cooperate with "traditional" lawyers. Also, the need for lawyers experienced in IT will increase in the future because programmers of smart contracts on their turn need consultation for making new kinds of contracts.[80]

Another factor worth mentioning, is the fact that it is still uncertain how some requirements, which decide whether the conditions are met and through this decide whether the transaction is completed, can be implemented within the system. Third parties will still have an active role in case these third parties need to implement information within a smart contract. These third parties or technical databases which provide information to the smart contract and serve as a "source of truth" (called "oracles"), can also make (human) mistakes[81] that eventually can lead to undesirable consequences: one example will be discussed in next sub-section ("The DAO-hack").

Although it seems that no third party will be involved with the drafting and execution of a smart contract, the reality however shows that third parties are still needed but in a different kind of way than before.

[80] S. Grybniak, "*Advantages and Disadvantages of Smart Contracts in Financial Blockchain Systems*", December 2017
[81] Unintentionally but also intentionally.

1.5.2: Immutability of smart contracts: the DAO-hack

Because of the fact that agreements in smart contracts are embodied in a programming language on a blockchain, it can be concluded that the concerning agreements are hardly or not at all changeable by the parties themselves. This will contribute to more predictability since the binary design of the code of the smart contract prevents that agreements of the smart contracts can be interpreted by their context or background.

Although third parties cannot perform a smart contract according to their own interest or context due to this immutability, mistakes made in a smart contract[82] can eventually lead to undesired outcomes. The question can arise whether an intervention possibility within the immutable character of a smart contract should be possible or desirable. This question was also raised when a hack was uncovered within an application of Ethereum, called "The DAO".

The DAO (Decentralized Autonomous Organization) was one ambitious smart contract-enabled project that made the headlines in 2016. The DAO was a fully-fledged crowd funding platform implemented on the Ethereum Blockchain, where investment proposals could be made and be presented in the form of code. Potential investors could subscribe on the proposals by paying a small amount of Ether[83] to the proposal, and eventually vote.[84]

In May 2016, a proposal was published on The DAO which (miss)used a weakness within the programming language. The proposal seemed to ask for only one payment, but eventually emptied out the complete wallets of all the investors that made a payment, due to a recursion which carried out itself over and over again.[85] Due to this weakness and hack by only one participant, many investors lost their complete investment wallets to the value of several millions. The outcome and consequence of the mistake within the code was definitely unwanted and not desirable. After the hack was discovered, the question rose whether it should be possible to change the code and create an exception regarding to the proposal that eventually created an unwanted situation and damaged many participants. This could be achieved by making a so-called "*fork*" within the code but would, however, breach the immutability of the smart contracts and allow an intervention that would be enforced by a third party.

Eventually, Ethereum announced that a fork would be applied[86] and implemented within the code of The DAO.[87] As a reaction to this decision, a part of the participants did not agree with the implemented fork and argued that it was a fundamental attack on the ideals of smart contracts and their immutability. This group eventually split up under the principle of "code *is* law" and continued under the name of "Ethereum Classic".

Apart from the group that split up, the hack of The DAO did uncover the fundamental weakness of the "myth" of immutability regarding smart contracts: almost every system or

[82] For example: mistakes made in the coding process of a smart contract.
[83] The native cryptocurrency of Ethereum.
[84] E. Tjong Tjin Tai, "*Smart contracts en het recht*", NJB 2017/146: p. 179-180
[85] E. Tjong Tjin Tai, "*Smart contracts en het recht*", NJB 2017/146: p. 179-180
[86] https://blog.ethereum.org/2016/07/15/to-fork-or-not-to-fork/
[87] https://blog.ethereum.org/2016/07/20/hard-fork-completed/

contract needs a human correction mechanism in the event that something goes wrong. Even self-executing smart contracts need a correction mechanism and cannot be completely immutable.[88] In case a mistake has been made in the code by a third party (a programmer for example), unwanted situations with far-reaching consequences can arise. This mistake may in theory not be changed or rectified in case of immutability. However, smart contracts still remain a human product and might contain mistakes, which can be seen as unavoidable when smart contracts are programmed. A recent study estimated that 12,000 Ether (currently valued at over 8 million Euro) have been irrevocably lost due to transactions where the party made a simple mistake by sending the Ether to a wrong destination address.[89]

Another recently published study[90] shows that nowadays, almost 35.000 smart contracts on Ethereum are still very vulnerable for attackers. One of the main reasons why smart contracts are still vulnerable according to the study, can be found in the fact that, unlike traditional consumer device software, smart contracts cannot be upgraded or patched once deployed due to their immutability. Because of the new ecosystem in which smart contracts are drafted, smart contracts are also very difficult to test, especially since their runtimes allow them to interact with other smart contracts and external off-chain services; they can be invoked repeatedly by transactions from a large number of users.[91] As soon as a vulnerability is uncovered, a large number of smart contracts will face difficulties. Therefore, it can be assumed that the immutable character of smart contracts might entail some difficult scenarios, especially when a mistake has been made within the programming stage of the smart contract. Immutability can be seen as a solid solution or attribute in case the computer-based smart contract will be infallible and trustworthy. Unfortunately, a smart contract still remains to be a human product, with its mistakes, flaws and unwanted outcomes. The ability to change it, might then be desirable.

1.5.3: Open standards in smart contracts

Despite the promises that smart contracts may increase efficiency, increase reliability and also reduce costs, it can be argued that smart contracts will face difficulties with respect to the anticipation on all possible scenarios that might happen in case parties agree to secure their agreements in a smart contract. Some scenarios or unwanted situations might be caused by unforeseen circumstances. Even the simplest problem, like no sufficient money to succeed with the transaction, can be caused by a situation in which both parties could not have anticipated in advance. As mentioned earlier in this thesis[92], these cases are most of the time dealt within the case law of a specific country in which specific open standards and norms

[88] E. Tjong Tjin Tai, "*Smart contracts en het recht*", NJB 2017/146: p. 179-180
[89] J. Pfeffer, "*Over 12,000 Ether Are Lost Forever Due to Typos*", Consensys.net (Mar. 9, 2018): https://media.consensys.net/over-12-000-ether-are-lost-forever-due-to-typos-f6ccc35432f8
[90] I. Nikolic et al, "*Finding the Greedy, Prodigal, and Suicidal Contracts at Scale*", School of Computing, NUS Singapore, March 14, 2018
[91] I. Nikolic et al, "*Finding the Greedy, Prodigal, and Suicidal Contracts at Scale*", School of Computing, NUS Singapore, March 14, 2018 (see chapter 1)
[92] See paragraph 1.3.1

have been developed throughout decades, to eventually deal with these unforeseeable situations that cannot be secured in regulations in advance.

Take for example the "reasonableness and fairness" (Dutch: redelijkheid en billijkheid) in Dutch contract law. This relatively open norm can tackle some unforeseeable problems with regard to contracts and explain the meaning of it by its context ("what did the parties have in mind when they drafted their contract?"). However, these open norms are incredibly difficult to secure in programming language and do not relate perfectly to the immutability of a smart contract ("code is/as law") in case a judge needs to give a verdict afterwards.[93] While every agreement within a smart contract has to be programmed *before* the contract will be effectuated since it cannot be changed later on, it can be said that the reasonableness and fairness will only come into play *afterwards* (i.e. as soon as the dispute already occurred). Consequently, programming an open norm that is used afterwards can be seen as problematic and practically impossible.

Due to the fact that the binary design of the programming language of smart contracts requires unambiguous language, it can be assumed that open and ambiguous terms (or principles) face difficulties in the programming stage of smart contracts. The more complex a contract gets, the more difficult it will be to set up a contract without any open or unambiguous language as a result of the fact that open and ambiguous terms are frequently used in comprehensive contracts. This can be related to the fact that complex contracts most of the time need some ambiguity in their writing. Ambiguity creates workability as it allows the parties to retain a measure of flexibility in performing their side of the bargain and in evaluating each other's performances. It also enables both parties to adapt to changing circumstances without having to redraft the agreement.[94] This might eventually create a deadlock situation: due to the immutable and binary design, it can be assumed that smart contracts need to be programmed with enormous precision in advance to prevent unwanted situations. However, because of this binary design some important norms or principles might not be programmed as in traditional written agreements. Consequently, the risk will arise that some norms need to be left out or programmed in a different, unambiguous way. Unfortunately, this way of programming might lead to unwanted situations which on their turn will be difficult to solve as a result of the immutable nature.

1.5.4: Technical limits of smart contracts

Another point of consideration are the technical limits of smart contracts which are stored on the blockchain. To illustrate the limitations, the most common and generalized framework to run smart contracts on will be taken as example: Ethereum. Although Ethereum was created for running any type of code on the blockchain more "easily" than its predecessors, it still has its technical limitations.

[93] J.B. Schmaal & E.M. van Genuchten, "*Smart contracts en de Haviltex-norm*", Tijdschrift voor Internetrecht, nr.1 maart 2017: p. 16

[94] E. Mik, "*Smart Contracts: Terminology, Technical Limitations and Real World Complexity*", 2017: p. 19

When executing a smart contract on Ethereum, each operation comes at a certain cost, whose amount is measured with a specific unit called "gas".[95] This amount of gas has to be paid in advance by the party who is initiating the transaction. The more complexity a smart contract has to handle, the more gas has to be spent by the party. However, due to security reasons there is an upper limit in the amount of gas which can be executed in the set of transactions that parties want to include in a block of a blockchain: the so-called "block gas limit".[96] The more complex a smart contract gets, the more difficult it will get to put the agreements within a smart contract.

Beside the upper limit in the amount of gas, the absence of direct access to advanced libraries can be seen as another problem for more complex smart contracts in certain fields of work. Many complex applications need to use advanced libraries or other external information sources. This is particularly the case in the financial industry, where advanced libraries are needed due to the fact that most of the financial contracts have a strong need to reuse mature financial libraries, algorithms and market data, which the blockchain has no direct access to. Even with access, there would be no space to execute those operations due to the above-mentioned "gas limit".[97] Other advanced libraries, like the data regarding the register of cadastral information of real estate, also seem to be too complex to execute automatically and without external input within smart contracts.

Although innovation of smart contracts never stands still, there might be serious questions whether or not smart contracts can technically execute very complex and extensive transactions due to their limited capacity.

1.6: Solutions based on smart contracts

Previous paragraph explained that smart contracts and their execution still face some "teething problems". Nevertheless, several experiments with blockchain and smart contracts have been set up in previous years, especially in the financial and insurance sector.

As we have seen before[98], The DAO was one good example of an experiment where investors could invest in freely tradeable "tokens" without an intervention of a third party, using blockchain and smart contracts on Ethereum. The fact that in June 2016 The DAO collected for more than $150 million (!) from private investors and became the largest crowdfunding project until then, shows that the interest in these kinds of experiments is very considerable. Although The DAO eventually turned into a catastrophe, many other experiments have successfully been made by different kind of institutions.

In the first place, banks and stock exchanges are experimenting for example with blockchain and smart contract solutions with regard to trade clearing and settlement of stock and funds. The process of trade clearing and settlement of stock includes most of the time labor-intense and complex processes with different approvals from different parties. Although

[95] See for more a technical explanation: https://blog.oraclize.it/overcoming-blockchain-limitations-bd50a4cfb233
[96] https://blog.oraclize.it/overcoming-blockchain-limitations-bd50a4cfb233
[97] https://blog.oraclize.it/overcoming-blockchain-limitations-bd50a4cfb233
[98] See paragraph 1.5.2

investment banks and other third parties have an extensive IT-infrastructure, it can be said that the process still takes a lot of time and holds the risks for mistakes.[99] Blockchain, however, holds the promise to have a jointly owned database with a guarantee of correctness, while smart contracts can offer an automated process of approvals, computing settlement amounts and eventually transfer the specified amount of money and funds.[100] As a result of the promises that blockchain holds, a consortium between forty of the world's largest investment banks and financial institutions, led by R3[101], experimented with the implementation of blockchain technology and smart contract solutions with regard to the settlement and trade clearing of stock.[102]

The same happened with so called "over-the-counter derivatives contracts". These contracts are also time-consuming and need settlement of all underlying assets. It would be efficient in case these contracts can be standardized and automated. Because of this reason, Barclays experimented with blockchain platform "Corda"[103] regarding so-called "smart derivative contracts".[104]

Furthermore, with respect to the insurance sector, Toyota Research Institute experiments with blockchain technology and smart contracts to share information with insurance companies and eventually offer personalized insurances to car drivers. Smart contracts can also help in this perspective to automate payments in case of damages to a car, without the intervention of third parties like insurance companies or experts.[105]

Abovementioned experiments are only a small selection of all the experiments that have been initiated over the past years. As the experience and knowledge about smart contracts and their underlying technology will grow in the upcoming years, it can be expected that many other experiments will be set up by governmental institutions, commercial entities and financial institutions. Therefore, it can be seen as no surprise that the Dutch Parliament already initiated to set up an investigation about the impact of smart contracts and blockchain on society, regulation and legislation.[106]

---

[99] A. de Backer & V. de Boe, "*Smart contracts in de financiële sector: grote verwachtingen en regulatoire uitdagingen*", Computerrecht 2017/252

[100] A. de Backer & V. de Boe, "*Smart contracts in de financiële sector: grote verwachtingen en regulatoire uitdagingen*", Computerrecht 2017/252

[101] A distributed database technology company, headquartered in New York, which leads a consortium of more than 70 of the world's biggest financial institutions in research and development of distributed ledger usage in the financial system.

[102] See for example the announcement of ING Bank: https://www.ing.com/Newsroom/All-news/-ING-completes-trial-of-five-emerging-block-chain-technologies-within-R3-Global-Bank-Consortium.htm

[103] Developed by R3.

[104] See https://www.coindesk.com/barclays-smart-contracts-templates-demo-r3-corda/

[105] A. de Backer & V. de Boe, "*Smart contracts in de financiële sector: grote verwachtingen en regulatoire uitdagingen*", Computerrecht 2017/252 (paragraph 2.4)

[106] Kamerstukken I, 2016/2017, 33 009

Chapter 2: Legal issues of smart contracts under Dutch law

2.1: Introduction

As previous chapter explained, it can be concluded that smart contracts show a great potential. The cumulation of the growing accessibility of smart contracts, the experiments which have been set up and growing attention which has been paid to smart contracts in legal academic writing, is supporting the evidence of the growing potential applications of smart contracts. However, at the moment of writing this thesis, no smart contract solution has yet significantly displaced some traditional way of contracting, partly due to the "failures" of the previous years and many legal questions that arise with the use of smart contracts.

Members of the Dutch House of Representatives ("de Tweede Kamer") already addressed the fact that further investigation to new topics as smart contracts and blockchain technology are necessary for the future, with respect to their impact on society, regulation and legislation.[107]

In this chapter, several legal issues and questions of smart contracts under Dutch (contract) law will be discussed and analyzed. Firstly, the question will be answered whether (and to what extent) law can be subsumed in program code under Dutch Law. One of the sub-sections of this chapter will therefore be dedicated to the discussion of "code *is* law" or "code *as* law". Secondly, the question of how Dutch contract law looks upon smart contracts will be analyzed: can smart contracts be seen as legally binding contracts/agreements under Dutch contract law and what legal manifestations are there? To conclude this chapter, the legal system of the Netherlands will be analyzed with respect to the interpretation of smart contracts under Dutch Law and how potential disputes can be dealt with in a way that also serves the purpose and nature of smart contracts.

2.2: Law in programming code

As discussed in the first chapter, smart contracts are, as with many technology-derived solutions, the product of computer programming. These computer programs create and enforce an agreed upon performance between parties, much like the traditional paper-based contracts. The most obvious difference, of course, is that smart contracts are computer-generated and thus it is the code itself that explains the obligations of the parties.

The question that can arise, is whether law and program code (maybe better: algorithms) can be seen as comparable, or even replaceable. After all, written law on a paper-based format appears to be an algorithm as well, just like a program code: "if (X) happens, then (Y)".[108] Surprisingly enough, this question is not relatively new: already in the 17th century famous mathematician Gottfried Leibniz predicted a calculus for calculating legal rights and

[107] Kamerstuk 33 009, nr. F.
[108] Dutch Blockchain Coalition, "*Smart contracts as a specific application of blockchain technology*", 2017: p. 19

obligations. Nowadays, most of the implementation of laws and regulations is supported by algorithms and information systems.[109]

However, this does not answer the question whether program code can be compared with law or even replace it. Although many aspects of a paper-based contract can be expressed within code in the form of "if…then", it can be assumed that some aspects are not suitable to be implemented within program code or algorithms. It is possible that some aspects of a contract can differ from the reality. Moreover, an outcome of a contract can surprise the parties who are involved due to the fact that the outcome may be unexpected (something that no one could know) or even unconsidered (something that exits, but no one could have thought about).[110] The norm that has been implemented in the contract can turn out to be problematic and lead to unwanted situations for one or all of the parties. For these situations, most of the time the case law within a country developed several solutions: in the Netherlands several options are created to solve these problems such as open standards (like reasonableness and fairness), principles of good governance and hardship clauses.

Unfortunately, these standards are difficult to fully implement in binary programming language (i.e. code) because these standards depend on an interpretation that comes from underlying social practices. Law itself remains to be a social practice where the meaning of rules depends on the context of it.[111] However, there is still a relationship between facts and norms: the facts determine the norms in a certain case, and the norms give the facts content. For program code itself, it will be difficult to present an outcome which also takes the social practices into account. Program code itself is deterministic and only follows the orders that are implemented in the code itself. In case an outcome might be unexpected or unconsidered, the code will not take the effect of the outcome into account.

The fact that program code cannot take some events into account, does not automatically mean that code cannot be law. Some pioneers in law and informatics even posit the proposition that algorithms *are* the law.[112] However, that proposition would mean that the nature of smart contracts and the way they include the applicable law/rules, differ from the legal system with social practices. Instead of utilizing the legal rules which have been made by the law-making processes of the government and case law, it can be argued according to some scholars that codes create their own legal rules. Therefore, there are several different ways of thinking about the nature of smart contracts.[113]

[109] Dutch Blockchain Coalition, "*Smart contracts as a specific application of blockchain technology*", 2017: p. 19
[110] Dutch Blockchain Coalition, "*Smart contracts as a specific application of blockchain technology*", 2017: p. 20. Here do the writers use the philosophy of Hart: *"because of the lack of understanding of the facts, we also lack insight into the norms".*
[111] Dutch Blockchain Coalition, "*Smart contracts as a specific application of blockchain technology*", 2017: p. 20
[112] L. Lessig, "*Code: version 2.0*", New York, Basic Book: 2006: p. 5-6
[113] M. van Eersel & T. van den Bergh, *"Blockchain en smart contracts: toegang tot een reeks van slimme dingen"*, FRP 2017/457 (afl. 4): p. 45

2.2.1: The nature of smart contracts: "code *is* law" or "code *as* law"?

As discussed before, the conditions which have been recorded in the smart contract are barely changeable by the parties themselves after the smart contract has been effectuated due to their immutable character. The code itself determines the operation and outcome of the smart contract, while the conditions cannot be interpreted or carried out in a different way by external factors or third parties unless these factors are put down explicitly in the code itself. However, it is almost impossible with the current state of the technology that smart contracts adapt to changing circumstances and the parties' revised preferences. In all probability, it will be possible in the not too distant future that artificial intelligence will be able to embrace principles where parties' intents can be enforced in smart contracts and update the code as soon as the circumstances or parties' preferences have changed, based on the principles of fairness or economic efficiency.[114] For now, it can be concluded that smart contracts still lack the ability to adapt and remain open-ended, while the more complex smart contracts also rely on external input. However, coders seek to convert certain performable obligations in the smart contract into something which has a certain degree of conceptual formal clarity that is almost similar to existing forms of computer code and would not allow the level of ambiguity that characterizes traditional human language.[115]

Because of the fact that a deterministic smart contract is immutable, there is a discussion among scholars whether or not smart contracts place themselves outside of the applicable legal system. This is expressed in an ideal where law does not rule anymore, but where code rules: where "code *is* law"[116]. Lawrence Lessig explains this ideal as follows: *"In real space, we recognize how laws regulate—through constitutions, statutes, and other legal codes. In cyberspace we must understand how a different "code" regulates— how the software and hardware (i.e., the "code" of cyberspace) that make cyberspace what it is also regulate cyberspace as it is. As William Mitchell puts it, this code is cyberspace's "law." "Lex Informatica," as Joel Reidenberg first put it, or better, "code is law."*[117]

The point that Lessig makes is that coders and software architects, by making a choice about the working and structure of IT networks and the applications that run on them, make important and often critical decisions about the rules under which the systems would be governed.[118] With these codes, programmers regulate the behavior of users of a (smart) contract, just like laws, social norms and market mechanisms do as well.

Proponents of the thought that "code is law/contract", argue that a smart contract somehow displaces the law or that the automatic performance aspect of a smart contract means that law cannot intervene in the case of a dispute over a smart contract, simply because automatic performance makes the completion of the contract a done deal.[119] The argument

[114] M. von Haller Gronbaek, "*Blockchain 2.0, smart contracts and challenges*", 2016, https://www.twobirds.com/en/news/articles/2016/uk/blockchain-2-0--smart-contracts-and-challenges#8
[115] Clifford Chance Report, "*Are Smart Contracts Contracts?*", December 2017: p. 5
[116] In some cases, also referred as "code is contract".
[117] L. Lessig, "*Code: version 2.0*", New York, Basic Book: 2006: p. 5
[118] M. von Haller Gronbaek, "*Blockchain 2.0, smart contracts and challenges*", 2016
[119] Norton Rose Fulbright LLP, "*Can smart contracts be legally binding contracts?*", November 2016: p. 16

that these proponents most of the time use, is that it can be seen as almost impossible to change a smart contract once it is effective due to the fact that it is self-executing and self-regulating ("honey-badger effect"[120]). According to them, a judge can never rule that somebody has to take a smart contract from the blockchain, since it is practically impossible to even remove it, and most of the time impossible to trace down who the parties are. The smart contract performs under the rules of the decentralized blockchain network, which eventually leads to the situation that no exact location or jurisdiction can be assigned as a result of this decentralized system. Therefore, it is hardly susceptible for human, legal intervention: the code itself is simply the law.[121]

On the other side, there are the scholars who argue that the parties who agreed to put their agreements in a smart contract, chose to regulate their legal relationship by using code but that their choice is not standing in the way to go to a court and that the legal system with its law is still applicable. The way to go to court is simply too fundamental to get rid of (article 6 ECHR) according to these scholars.[122] Moreover, they argue that smart contracts are a way of giving effect to their agreements in a digital way: these scholars tend to see "code *as* law", which is still subject to the system of law, case law and underlying social norms. According to their view, smart contracts are a digital way in which execution is given to a digital program where operation, definitions and business logic is included ("smart contract execution").[123] While the "code is law/contract" scholars see smart contracts as full-fledged contracts, it can be concluded that the "code *as* law" scholars see the smart contracts more as the automated execution of agreements that are suitable for automation.

The most obvious example of differences in way of thinking between the two groups can be found during the hack of The DAO. As discussed before[124], the two groups protested during the discussion whether or not to follow the code, even though the code contained an obvious mistake and made it possible that several millions of dollars were diverted by a hacker. The code *is* law-group did not want to change the code, while the majority of the participants proposed an opposite position: they simply did *not* want to follow the code *is* law-principle and proposed to change the code because of the fact that the situation was not expected and above all, not wanted. In its core, the discussion did not much differ from the old contradiction in contract law between text versus context. While one group sees the code (text) as something unchangeable and as a perfect display of the agreements, the other group wants to change the code in case something works out differently than agreed upon by the contracting parties (context).

Between the opposite positions of "code is/as law", is a middle ground where the "dual integration" of both positions is leading. This "hybrid"-position argues that the non-human

[120] Reference to the animal "honey badger". Honey badgers can be seen as one of the most fearless animals. Smart contracts also do not "fear" anything. Smart contracts are self-executing and self-regulating and will only do what is instructed in the code of the smart contract.
[121] E. Tjong Tjin Tai, "*Smart contracts en het recht*", NJB 2017/146: p. 179
[122] Article 6 ECHR is the Right to a Fair Trial.
[123] M. van Eersel & T. van den Bergh, *"Blockchain en smart contracts: toegang tot een reeks van slimme dingen"*, FRP 2017/457 (afl. 4): p. 45
[124] See paragraph 1.5.2.

execution is contained in computer code (like automation of payment), while the provisions and obligations of more human nature (like social practices, enforcement of fulfillment and settlement of disputes) are described in a separate part that is set up in natural language. According to this point of view, the smart contract itself should contain the more 'active' actions with the events that will be automatically executed, while the more 'passive' actions, which are needed for the settlement of potential disputes, are recorded in a "normal" contract and will be appended to the smart contract.[125]

2.2.2: Applicability of law to smart contracts

The discussion between "code is law" and "code as law" has some similarities with the first emergence of the Internet couple of decades ago. Some scholars also believed that the Internet would somehow evolve into a legal "Wild West", free from legal intervention.[126] Legislators and the courts across the globe took by contrast a different view and acted together to protect people and transactions over the Internet in a variety of ways.[127]

Therefore, some scholars do believe that probably the same will happen to smart contracts. According to Eliza Mik, there will be no Wild West scenes with regard to the changes that can be expected with smart contracting: "*Contract law has always displayed an inherent ability to adapt to new situations, without the need for major revisions of its underlying principles. Technology – while not changing contract law – adds complexity to the traditional analysis. The question is not 'do traditional principles apply?' but 'how do they apply?"*[128]

Despite the similarities of the Internet with regard to the discussion of the applicability of law, it can be argued according to my view that the discussion regarding the smart contracts differs in difficulty compared with the Internet. The self-executing nature of smart contracts and the decentralized blockchain network, which serves as an underlying trusted network or mechanism where smart contracts can be stored on, make it more complicated (not impossible) to use traditional contract law. Moreover, in a simple traditional contract law case where a judge rules that a (smart) contract needs to be removed because of fraud, it can be argued that it will be technically almost impossible to do with a smart contract that is stored on a blockchain network. Because of the fact that the smart contract will be stored on the blockchain network "eternally", it will be impossible for a single person or entity to remove it from the blockchain network in case a judge rules to do so. Due to the decentralized mechanism of the blockchain network, it can be seen as nearly impossible in a technical, but also practical way. Consequently, a single person or entity cannot remove a smart contract from a blockchain network on its own since the majority of the nodes have to agree upon it, while it also might be possible that other smart contracts on the blockchain network have

[125] M. van Eersel & T. van den Bergh, *"Blockchain en smart contracts: toegang tot een reeks van slimme dingen"*, FRP 2017/457 (afl. 4): p. 45. See also H. Schuringa, "*Enkele civielrechtelijke aspecten van blockchain*", Computerrecht 2017/254: § 3.1

[126] Norton Rose Fulbright LLP, "*Can smart contracts be legally binding contracts?*", November 2016: p. 15

[127] Norton Rose Fulbright LLP, "*Can smart contracts be legally binding contracts?*", November 2016: p. 15

[128] Eliza Mik, "Formation Online":, 159, *Contract formation: law and practice*, OUP, 2010: p.159-161

used the smart contract (which has to be removed) for their own purposes. Another question which might arise is whether these smart contracts also have to be removed and whether or not this will be possible in a technical and computable way.

In spite of the previous differences, it can be said that the underlying struggle of contracting and law, still remains the same. With smart contracting do not lawyers but coders formulate the content and scope of the smart contract that will be converted into computer executable code: in other words, the coders have the right in deciding the framework and the limits of the smart contract. The reality will be that this will become a customer-driven market[129] since parties of smart contracts will pay coders to tailor smart contracts to suit their specific needs. Coders will become akin to lawyers drafting "traditional" contracts, and coders will be assisted by lawyers specialized in the language and mechanics of smart contracts.[130]

Consequently, the connection between cyberspace and smart contracts can be made easily according to Von Haller Gronbaek: "*Just as it was quickly realised that cyberspace was not free from government interference, it must be understood that smart contracts are not only subjected to "code is law" but are governed by the law of the land. Even smart contracts with autonomous software agents as parties can trace their beginnings to human actions and will also impact human beings or other actors in the "real" world at some time. Just as there are many limits to freedom of contract in general, there will be many limits in contract law and regulation on the autonomy and self-enforcement of smart contracts. Smart contracts do not exist in a legal vacuum just as cyberspace is not cut off from the real world.*"[131]

As Von Haller Gronbaek argues, smart contracts cannot exist in a legal vacuum. As the The DAO hack already showed, there will be a Wild West scenario in case no law is applicable to smart contracts. Since the code itself is agnostic and does not know what is "good or bad", it can be assumed that a simple mistake within the code itself (deliberately or not) will have an enormous impact on the process of smart contracts. Coders with bad intentions could possibly have free play in case their malicious code will be the law, also because most people do not know how to check whether the code is good or not. The "code is law"-school, which implies that smart contracts obviate the need of any legislation or judicial protection, simply overlook the fact that the lack of recourse to legal institutions or legal principles would not only incentivize fraudsters or hackers, but also discourage the use of smart contracts with regard to several (financial) transactions.[132] After all, the "trustless" and "incorruptible" character of the blockchain is of limited significance in case the code of the smart contract is executed outside the blockchain and if the immutable and automated smart contract will be incapable in protecting the parties from several risks which are not unusual, like changed circumstances or (deliberately or not) errors in the code.[133] For these reasons alone, it becomes apparent that the contracting parties of a smart contract must retain the ability to rely on traditional legal protections.

[129] M. von Haller Gronbaek, "*Blockchain 2.0, smart contracts and challenges*", 2016
[130] M. von Haller Gronbaek, "*Blockchain 2.0, smart contracts and challenges*", 2016
[131] M. von Haller Gronbaek, "*Blockchain 2.0, smart contracts and challenges*", 2016
[132] E. Mik, "*Smart Contracts: Terminology, Technical Limitations and Real World Complexity*", 2017: p. 13
[133] E. Mik, "*Smart Contracts: Terminology, Technical Limitations and Real World Complexity*", 2017: p. 13

Legal rules provide remedies to the aggrieved parties of a smart contract that has been executed in case the contract would be deemed invalid by the court due to legal concepts such as fraud, forgery or lack of legal capacity. These legal principles are so fundamental to regulation of exchanges that it would be counterproductive if these could be circumvented alone by subjecting to the fact that the contract is executed due to its fully automated and self-enforcing capabilities.[134] Therefore, it can be concluded that law needs to be applicable to smart contracts to prevent a legal vacuum and incentivize the use of smart contracts.

2.3: Smart contracts as valid legal contracts under Dutch law

As previous sub-section explained, it can be concluded that law still needs to be applicable to smart contracts and that smart contracts cannot exist in a legal vacuum. Because of the fact that law is still applicable to these contracts, the question might arise how Dutch (contract) law looks upon smart contracts. After all, can smart contracts be considered as enforceable contracts under Dutch contract law? Also, what kind of legal manifestations are there with regard to these *automatable and enforceable agreements*? Beside the provisions with regard to agreements of the Dutch contract law, attention will also be paid to so-called *electronic contracting*, which is already provided in Dutch contract law.

However, it might be needful to take a look at the interpretation of the term "smart contract" before even getting to the core of Dutch contract law. The term itself can be confusing and might eventually lead to an undesired legal interpretation.

2.3.1: Smart contracts: not necessarily a "contract" or "smart"

The term smart contract indicates that the legal application concerns a "contract" and that it also can be seen as a "smart" solution. However, the term can be seen as double confusing.

In the first place, the legal application of the technology behind a smart contract is not necessarily a "contract". As discussed before, a smart contract can make it possible to automatically perform a transaction between two parties as soon as several obligations have been fulfilled. In this case, it will be understandable that the transaction and its execution can be seen as a contract (agreement), as it does not differ that much from a traditional paper-based contract, except for the automatability and immutability. There are, however, still many other legal applications possible with smart contracts than solely transactions between two parties. These other legal applications do not necessarily have legal implications or have to be contracts.

The Dutch legal system, for instance, has many legal acts like *multilateral private legal acts* (like contracts between two or more parties), *unilateral private legal acts* (like donations), *public legal acts* (which are governed by administrative law) and *automation of legal processes*.[135] Most of these legal acts consist several acts which can be converted into smart

[134] M. von Haller Gronbaek, "*Blockchain 2.0, smart contracts and challenges*", 2016
[135] Dutch Blockchain Coalition, "*Smart contracts as a specific application of blockchain technology*", 2017: p 22

contract applications. While the multilateral private legal acts and their execution can be seen as "contracts", it can be assumed that it will be very difficult to consider some of the public legal acts or automation processes, as a legal contract. Therefore, the term smart *contract* can be seen as infelicitous due to the fact that it will not always necessarily concern a contract in the broadest sense.

Secondly, smart contracts are not necessarily "smart". Essentially, a smart contract performs what it has been instructed to do. Because of the fact that all of the actions and obligations are pre-programmed within the code of the smart contract, it can be argued that a smart contract has a lack of any self-thinking or pro-activity.[136] Despite the prediction that in the near future artificial intelligence will provide smart contracts the ability to interpret the language of a contract and to perform the contracts themselves without any human input, as "robotic agents", smart contracts are still dependent of human input. Consequently, most of the time data from outside the smart contract and blockchain network[137] will need to be required in order to determine whether the conditions of execution have been fulfilled. Smart contracts and the blockchain network are still "deaf" and "blind": it is still not possible for a smart contract to get information from the outside by itself.[138] Therefore, smart contracts still need input from outside parties, so-called "oracles". Smart contracts are in essence a deterministic computer program that sometimes still rely on input of outside parties. Because of this, some scholars do not consider smart contracts as "smart" but rather like "robotic" or "automated".[139] Thus, smart contracts can be said to be "smart"[140] to the extent they offer the efficiency of automated contractual performance and reduce the risk of human errors with respect to the execution. But barring quantum improvements in artificial intelligence, the utility of smart contracts is limited to situations where the satisfaction or non-satisfaction of a particular factual condition is objectively ascertainable through programmatic reference to an extrinsic data source.[141]

Despite the confusing term of "smart contract" and the fact that smart contracts can be used for more legal acts than just "legal contracts", the subsequent parts of this thesis will continue with smart contracts as multilateral private legal acts (contracts/agreements) and focus more on Dutch contract law. It is beyond this thesis to fully discuss all of the legal manifestations where smart contracts can be used.[142] The central question of this section is whether Dutch (contract) law is suitable for the application of smart contracts and whether

---

[136] It is still programmed by a human and not by the smart contract itself. See also Dutch Blockchain Coalition, "*Smart contracts as a specific application of blockchain technology*", 2017: p 12

[137] I submit that a smart contract can exist outside of the context of a blockchain network. But for the comprehensibility of this thesis, I assume that smart contracts exist on the blockchain.

[138] Dutch Blockchain Coalition, "*Smart contracts as a specific application of blockchain technology*", 2017: p 18

[139] M. van Eersel & T. van den Bergh, *"Blockchain en smart contracts: toegang tot een reeks van slimme dingen"*, FRP 2017/457 (afl. 4): p. 45

[140] Report of Clifford Chance describes it as follows on page 5: *"…smart contracts are smart as in "smart watch", not as in intelligent*."

[141] D.M. Adlerstein, "*Are Smart Contracts Smart? A Critical Look at Basic Blockchain Questions*", June 2017, https://www.coindesk.com/when-is-a-smart-contract-actually-a-contract/

[142] See for extensive discussion of other legal manifestations Dutch Blockchain Coalition, "*Smart contracts as a specific application of blockchain technology*", 2017: p 22-37

smart contracts can be seen as valid legal *contracts* under Dutch law. However, it is necessary to keep in mind that the other legal manifestations have different consequences regarding their legal impact of smart contracts.

2.3.2: Smart contracts as legal contracts according to Dutch Civil Code

To answer the question how Dutch contract law looks upon smart contracts and whether smart contracts can be considered as enforceable contracts, the Dutch Civil Code ("Burgerlijk Wetboek", hereinafter "*BW*") and case law will be examined to see what the Dutch legislation and courts would recognize as a legal contract. Although the "code *is* law"-school would argue that a smart contract would put aside the law and create a "legally" binding contract, this sub-section will not go into this way of thinking. This sub-section will consider the smart contract from the *dual integration*-school, where the smart contract consists out of an automated but also (traditional) written part.

The prerequisites of a legally enforceable contract, which is a type of agreement, can be reduced to two elements: *intention* and *consideration*. The first prerequisite, the presence of intention, is always evaluated according to the parties' actions and words. According to the Dutch Civil Code, an agreement is said to exist upon offer and acceptance (article 3:32 and 3:33 BW). The specific requirement is stated in article 3:33 BW as follows: "*A juridical act requires the will (intention) of the acting person to establish a specific legal effect, which will (intention) has to be expressed through a statement of the acting person."* According to article 3:37 BW, the way in which parties would like to express their intentions is in principle free of form. This underlying principle of Dutch contract law is also known as "consensualism": agreements are made by the mere *consensus* of intention of the parties.[143] Therefore, an agreement can be formed orally, in writing, Morse code and parties may even implicitly reach expression of their intentions through the behavior of a person (by conduct).[144] With very little exceptions introduced to the principle of this freedom of form[145], it can be said that Dutch contract law abstracts from the form and focuses merely on the substance. As a result of the freedom in which parties can express their agreements, no *legal* obstacle can be found in Dutch legislation to express their agreements in code or to regulate their obligations towards each other in an automated process running on a blockchain, i.e. in a smart contract.[146]

The second prerequisite, consideration, is needed to indicate the difference between smart contracts as "legally enforceable *contracts*" and smart contracts as mere "technical tools". Consideration can be described as something that is promised or given by both parties in exchange.[147] Where one party must accrue profit or interest, the other party will need to

[143] J. Hijma, "*Rechtshandeling en Overeenkomst*", 6e druk, Deventer: Kluwer 2010, p. 150
[144] See article 3:37 BW
[145] Dutch law sometimes requires that certain agreements are in writing. See for example article 7:317 BW with regard to leasehold agreements: "*The leasehold agreement and the agreements to change or end it (to change or end the lease agreement), must be concluded in writing*."
[146] E. Mik, "*Smart Contracts: Terminology, Technical Limitations and Real World Complexity*", 2017: p. 14
[147] E. Mik, "*Smart Contracts: Terminology, Technical Limitations and Real World Complexity*", 2017: p. 14

suffer or undertake some loss or responsibility. With regard to smart contracts, parties can for example exchange tokens in return for (digital) assets. Moreover, for a legally enforceable smart contract, it is necessary that an exchange is involved in order to fulfill the prerequisite of consideration: whenever the prerequisite of consideration is not fulfilled, it must be assumed that the smart contract is used in a mere technical way.[148]

Although that each contract needs to be evaluated as a separate case each time, it can be assumed that in most of the cases there will be no legal obstacles or problems with regard to the legal enforceability of smart contracts under Dutch contract law if the abovementioned prerequisites are present. Especially in the event that smart contracts rely on already existing legally binding agreements.

2.3.3: Requirements of legal contract under Dutch contract law

As mentioned before, a contract is a type of agreement under Dutch contract law. According to the Dutch Civil Code does an agreement need to comply with three requirements in order to establish a *legally* binding contract:

1. between the contracting parties must be consensus of intention (article 6:217 BW, in conjunction with article 3:32 and 3:33 BW (= offer and acceptance))
2. the agreement needs to be sufficiently determinable (article 6:227 BW) and;
3. the agreement must not be in conflict with the law, public order and good morals (article 3:40 BW).

Regarding the first requirement about consensus of intention by means of the offer and acceptance model, it can be said that it will not give significant problems with regard to smart contracts. As discussed before, the way in which an agreement can be concluded under Dutch contract law is in principle free of a set of forms. In this context, contracting parties are free to choose to put their agreements in a code since there is no legal obstacle in case both parties agree to put their agreements in a code (of a smart contract) and eventually accept the offer that is made. However, to see whether there is consensus or not, it does not only mean that one party has to accept the offer that has been made by the other party, but it also means that the parties have to agree upon the fact that the agreements are put down in a form of a smart contract. The fact that the agreements are put down into a smart contract could eventually have (legal) implications for the parties. Moreover, in case one of the parties is not aware of the fact that a smart contract is used, it can be concluded that the consensus of intention is not fully present. If one of the parties still puts the agreements in a smart contract without the consensus of the other party, it can be said that the party can make an appeal on a legal error, according to article 6:228 BW. In sum, consensus about putting the agreements in a smart contract and not in traditional writing is essential because of this reason.

[148] For example: automation of a legal process or actuation of an application like a weather app.

With the use of smart contracts, the intention of the parties will be expressed in coding and algorithms. According to *Voulon,* this declaration of intention via codes can be labeled as a *programmed intention* of the parties which are involved with the smart contract.[149] As a result of the fact that the parties explicitly choose to implement and execute their agreements via a smart contract, this choice can be considered as additional 'proof' of the consensus between the parties.[150] The use of a smart contract for the execution of an agreement namely requires an extra step compared to the execution of a traditional agreement: beside traditional language, agreements in smart contracts also need to be implemented in programming language.

Before agreements between parties in a smart contract can be expressed in programming language, it is required that the parties agree on the content of their agreements, on what they want the smart contract to carry out and under which conditions the smart contract will execute. At this stage, the parties will make agreements, determine which agreements or which parts of agreements they want to express in a smart contract and also determine which conditions must be fulfilled before the consequence of the smart contract takes effect. These agreements, whether they are written down or oral, can already be seen as a legal agreement according to the Dutch Civil Code. After all, the expression of intention is free of form. However, as long as the contracting parties do not express their intentions in code language, there will not be a smart contract.[151] A smart contract will be created only when these agreements will be programmed in code. Consequently, this whole process of expressing intentions and implementing them in programming language before accepting the offer, makes it almost impossible that no consensus of intention is reached between contracting parties to express their agreements in a smart contract.[152]

Beside the fact that a consensus of intention between both parties is needed, it is also necessary that the smart contract needs to be *sufficient determinable* according to article 6:227 BW. Although the law itself does not mention anything about the "determinability", the Parliamentary history does mention that legal acts, like (smart) contracts, need to have "a determinable *subject*".[153] However, this does not mean that the content of a contract needs to be certain in advance: it is only necessary that the content (subject) of the contract can be determined on the basis of "predetermined criteria or normal course of business between the parties".[154] With regard to a correct execution of a smart contract, it will be necessary that all criteria consist of clear and indisputable binary codes, because a smart contract cannot think by itself. Criteria which have not been coded clear or indisputable, eventually lead to an undesired output or error of a smart contract. Because parties want to prevent these unwanted situations, most of the smart contracts will already consist out of clear and

[149] M.B. Voulon, "*Automatisch contractreren*" (diss. Leiden), Leiden, Leiden University Press: 2010: p.54
[150] M.B. Voulon, "*Automatisch contractreren*" (diss. Leiden), Leiden, Leiden University Press: 2010: p.56
[151] Although there is an agreement according to the Dutch Civil Code.
[152] Except for the exceptional cases of deception and fraud.
[153] J. Hijma, "*Rechtshandeling en Overeenkomst*", 6e druk, Deventer: Kluwer 2010, p. 160. Reference to "Parlementaire Geschiedenis van het Nieuwe Burgerlijk Wetboek", Book 6, p.895
[154] J. Hijma, "*Rechtshandeling en Overeenkomst*", 6e druk, Deventer: Kluwer 2010, p. 150

determinable language (i.e. code). Where traditional contracts most of the time consist out of open norms or ambiguous terms, it can be said that smart contracts use binary, unambiguous terms to make sure that the smart contract can be self-executing. Therefore, the requirement of sufficient determinability will not lead to serious problems with regard to smart contracting from a practical point of view: smart contracts *need* to be sufficient determinable to make execution even possible. Nevertheless, parties who agree to put their agreements in a smart contract, need to have certain knowledge about the code of the smart contract and legal impact to really understand the determinable language. However, this will not alter the fact that a smart contract on itself will be sufficient determinable according to Dutch contract law.

The third requirement, that the agreement must not be in conflict with the law, public order and good morals, speaks for itself[155]: a smart contract must not be against the Dutch law and must not contain illegal practices. This requirement alone could already be a reason not to accept the "code is law"-principle because these smart contracts cannot be "untouched" by a human correction mechanism.

It can be concluded that the requirements of a legal binding contract under the Dutch Civil Code will not cause any significant legal obstacles with regard to smart contracts. Even the fact that smart contracts automate the performance of contractual obligations is legally uncontroversial according to Dutch contract law. The Dutch Civil Code contains several articles with regard to automated and electronical contracting. The question can arise whether or not this framework for electronic contracting already consist a well-fitting framework for smart contracts.

2.3.4: Electronic contracting according to Dutch contract law

Due to the fact that Dutch contract law is most of the time free of form, it can be said that the intention of consensus regarding offer and acceptance can also be expressed via electronic form.[156] Despite the ruling principle of freedom of form, Dutch contract law requires sometimes that the agreement needs to be concluded in writing: this is for example the case with leasehold agreements[157] and buying a house.[158] In these cases where the law requires a written agreement, Dutch contract law also provides a possibility that in principle an electronic contract will be sufficient, provided that a number of conditions are met, which are described in article 6:227a BW.

Smart contracts are also created electronically (on for example Ethereum), most of the time without physical interaction between the contracting parties. In case a smart contract deals with the selling of a house and is created electronically, this smart contract needs to meet the additional conditions of electronic contracting as well. Article 6:227a BW provides

[155] Although some Dutch scholars see the article of 3:40 BW as the "ugly duckling" of Dutch contract law: see: Hijma, "*Rechtshandeling en Overeenkomst*", 6e druk, Deventer: Kluwer 2010, p. 161. It is beyond this thesis to fully explain the functioning of this article

[156] See article 3:37 BW

[157] See article 7:317 BW

[158] Article 7:2 and 7:3 BW

these conditions under which an electronic agreement can be seen as legally sufficient. Nonetheless, it is important to note that it must involve an agreement for which the Dutch Civil Code does not prescribe the intervention of a court, a public body or a professional performing a public task.[159] For these agreements, the Dutch legislator did consider it as desirable to meet the "physical form requirements" in writing, which cannot be done electronically.[160]

According to article 6:227a BW, an agreement which is concluded electronically can be considered equivalent to a legal (written) agreement if:

a. it can be *consulted* by the contracting parties;
b. the *authenticity* of the agreement is sufficiently guaranteed;
c. the *moment of conclusion* of the agreement can be established with sufficient certainty; and
d. the *identity of the parties* can be established with sufficient certainty.

To see whether smart contracts meet all the above-mentioned requirements of article 6:227a BW, all of the requirements will be discussed and analyzed separately in accordance with Dutch contract law and the Explanatory Memorandum[161] of the Adaption Act of the Directive of Electronic Commerce[162], since article 6:227a BW was introduced following the implementation of the Directive of Electronic Commerce.

a. Consultable

The first requirement is that the electronic agreement must be available for consultation by the parties of the agreement. This means that it must be recorded in such a way that the parties are able to access and preserve the content for "*subsequent cognizance"*.[163] Furthermore, this requirement may also imply that the party who wishes to make use of a specific technique, will be obliged to provide the other party with the correct technical means to consult the contents of the electronic agreement in case this is not available to the concerning party.[164]

One of the features of the blockchain network on which smart contracts are stored, is that all parties within the so-called peer-to-peer network have the same information at their disposal: in short, an identical copy of the register is stored in a database on each node. In case one party wants to make adjustments to the smart contract, it can be said that all of these adjustments need to be added to the chain of blocks through a new block. In this way, all parties thus have the same information on the blockchain. Moreover, due to the nature of the blockchain technology, the agreements which are implemented within the smart contract

[159] See section 2 of article 6:227a BW
[160] Kamerstukken II 2000-2001, 28 197, nr. 3, p.55
[161] Kamerstukken II 2000-2001, 28 197, nr. 3: p. 51 - 55
[162] Directive 2000/31/EC of the European Parliament and Council
[163] Kamerstukken II 2000-2001, 28 197, nr. 3: p. 53
[164] Kamerstukken II 2000-2001, 28 197, nr. 3: p. 53

cannot be adapted or removed without consensus from the other nodes. Therefore, it seems that smart contracts which are stored on the blockchain are recorded in such a way that the parties are able to access and preserve the content for "*subsequent cognizance",* without the risk that other parties have not the complete set of information.[165]

However, there is one problem with regard to the *consultability* of the information of a smart contract. Although all the information of the smart contract on the blockchain is accessible and also the same for all of the parties, it can be said that the available information comes in a so-called "compiled form".[166] Unfortunately, this compiled form is only legible for the computers and not the parties themselves. As a result, the compiled form alone will not lead to the desired degree of consultability that is needed according to article 6:227a BW. As stated in the report of the Dutch Blockchain Coalition (DBC), the requirement of consultability will only be achieved when the so-called "source code" is made available to the parties.[167] This so-called source code refers to the code in a programming language that is written and read by people (and not only by computers like the compiled form). Although all smart contract code is available to all parties on a blockchain, it is available in the compiled form. It can nonetheless be irrefutably demonstrated that a given source code results in a particular compiled form. In other words: if the source code is made available to all of the parties of a smart contract, it is possible to deduce the operation of the compiled form.[168]

According to my view, this will still bring a lot of uncertainty with regard to this information that can be consulted since it seems to be logical that many parties still lack the needed knowledge with regard to the "source code" or "compiled form" and translating it into information that is needed to consult and interpret a smart contract. Even programmers have problems with converting the information of source codes to useful information about the smart contract, let alone unexperienced contracting parties. Although the code might be consultable in the broadest sense of the word, it does not necessarily mean that it is useful and understandable for the average user of a smart contract. Because of this reason alone, there might be a need for a 'natural language' which describes the operation of the source code and makes the agreements clear in a way that is understandable to all parties.

b. *Authenticity*

The second requirement is that the *authenticity* of the agreement is sufficiently guaranteed. With traditional paper-based contracting, a certain degree of authenticity arises due to the fact that manipulation or changes of the agreements in writing will become apparent later on and easily notified.[169] On the contrary, the content of electronic contracts can be manipulated

[165] In case of a smart contract which is stored on the "*public* blockchain", everyone can access and consult the information that is stored. It is also possible to limit the accessibility and consultability for a predetermined group of parties by using a "*private* blockchain". See also paragraph 1.2.1.

[166] Dutch Blockchain Coalition, "*Smart contracts as a specific application of blockchain technology*", 2017: p. 25

[167] See footnote of Dutch Blockchain Coalition, "*Smart contracts as a specific application of blockchain technology*", 2017: p. 25

[168] See footnote of Dutch Blockchain Coalition, "*Smart contracts as a specific application of blockchain technology*", 2017: p. 25

[169] Kamerstukken II 2000-2001, 28 197, nr. 3: p. 53

quite easily and unnoticed if insufficient measures have been taken to protect them.[170] Therefore, the second condition requires that the authenticity of the agreement is sufficiently guaranteed. In order to fulfill this requirement, it is necessary that parties record the electronic agreement in such a way that sufficiently can be relied on the correctness and authenticity of its content.

The requirement of authenticity is to a certain extent easily met since a smart contract cannot be easily changed *unilaterally* by one of the parties. The nature of smart contracts stored on the blockchain entails the fact that agreements cannot (or hardly) be adjusted afterwards without a general consensus of all nodes. Furthermore, a change within a smart contract leads to a new hash value (i.e. "a changed digital fingerprint"[171]) and makes it possible to detect any changes easily.

According to the Dutch legislator[172], one possible way in which parties can fulfill the requirement of authenticity is by recording an agreement in an electronic file that is provided with an *electronic signature,* whereby any manipulation or change can be easily notified.[173] This electronic signature is data in electronic form that is attached to, or logically connected to, other data in electronic form and used by the signer to sign the contract.[174] This can be a (simple) scanned written signature but also an "advanced electronic signature": this advanced signature entails *encryption*. Encryption is a technology used for encrypting data of a digital contract in such a way that only authorized parties can access it and those who are not authorized can simply not. With smart contracts, asymmetric encryption is most of the time used and entails two types of so-called "key pairs": a private key that is secret and known only to the possessor, and a public key that is included in a public register. Although this technology is very complex, any subsequent modification of the data can be traced down because of the fact that the electronic signature is linked to the electronic contract.[175]

However, it was never the intention of the Dutch legislator to set out more stringent requirements on the point of authenticity of electronic contracts compared with traditional paper-based agreements.[176] Because of this, nothing more than an *equivalent degree* of certainty about the authenticity is essentially required.[177] Therefore, in case smart contracts are stored on the blockchain, the authenticity of the agreement is most of the time already sufficiently guaranteed due to the nature of the smart contracts and blockchain itself[178]. Nevertheless, by using an electronic signature, it can be argued that parties rely on an even higher degree of authenticity of the content within a smart contract. It seems practical and

[170] Kamerstukken II 2000-2001, 28 197, nr. 3: p. 53
[171] Dutch Blockchain Coalition, "*Smart contracts as a specific application of blockchain technology*", 2017: p. 23
[172] Kamerstukken II 2000-2001, 28 197, nr. 3: p. 53-54
[173] See article 3:15a-c BW
[174] Electronic signature according to the Regulation (EU) No 910/2014 of the European Parliament on Electronic Identification and Trust Services for Electronic Transactions in the Internal Market and Repealing Directive 1999/93/EC (eIDAS Directive). Implemented in art. 3:15a-c BW.
[175] Kamerstukken II 2000-2001, 28 197, nr. 3: p. 54
[176] Kamerstukken II 2000-2001, 28 197, nr. 3: p. 52
[177] Dutch Blockchain Coalition, "*Smart contracts as a specific application of blockchain technology*", 2017: p. 23
[178] It will most of the time guarantee an equivalent degree of certainty if not a higher degree of certainty, compared with written agreements.

beneficial for the functioning of a smart contract that the parties make use of an electronic signature to sufficiently guarantee the authenticity of the contract.

c. *Moment of conclusion*

Beside the consultability and authenticity of the contract does the moment of conclusion of the agreement also need to be established with *sufficient certainty*. According to the Dutch legislator, this requirement is drafted because it will be important in many cases to know at what point an agreement has been concluded between the parties. In other words, the moment of conclusion might be important to determine from which point parties have obligations towards each other or to determine whether there is a shortcoming in the fulfillment of the obligations or not.[179]

Article 6:227a BW refers to 'sufficient certainty' with regard to the moment of conclusion. Determining whether the moment of conclusion can be established with sufficient certainty depends on the "circumstances of the case, such as the state of the technology, the nature of the agreement and the capacities of the parties".[180] According to this view, it can be expected that the contracting parties take substantial measures to ensure the moment of conclusion in case the interests of the agreement are considerable.

With regard to smart contracts, this requirement can be fulfilled if the contracting parties sign the smart contract with an electronic signature which includes so-called indisputable and automatic "time stamps". As a consequence, smart contracts will ensure the moment of conclusion with an even higher degree of sufficiency than paper-based contracts.

d. *Identity of the parties*

When a contract is concluded in writing, it is usually obvious who the parties are due to the fact that the parties will be present at the moment of signing the agreement. With electronical contracting this situation will be less frequent. Therefore, the fourth requirement of article 6:227a BW requires that the identity of the parties needs to be established with sufficient certainty in case of electronic contracting. In case of a smart contract, this requirement might create several problems.

Smart contracts are (mostly) stored on the blockchain which can be public or private. As previously discussed in paragraph 1.2.1, a public blockchain is characterized by a *decentralized* identity management. Consequently, this might lead to a situation where it will be practically impossible to establish the identity of the contracting parties with sufficient certainty, due to the lack of a central authority that determines who the users of the blockchain are and the fact that almost everyone can participate in the network and can create multiple identities.[181] As a result, these public blockchain do most of the time not sufficiently identify or do a sufficient authentication of the parties, as referred to in article 6:227a BW.[182]

---

[179] Kamerstukken II 2000-2001, 28 197, nr. 3: p. 54

[180] Kamerstukken II 2000-2001, 28 197, nr. 3: p. 54

[181] By simply generating a new key pair: see J. Linneman, "*Juridische aspecten van (toepassingen van) blockchain*", Computerrecht 2016/218, afl. 6: p.320-322

[182] Dutch Blockchain Coalition, "*Smart contracts as a specific application of blockchain technology*", 2017: p. 14

Despite the fact that all transactions for all participants are visible in the blockchain and that transactions can also be linked to persons[183], it can be assumed that the public blockchain is an almost anonymous system. This anonymity might cause problems with the requirement of establishing sufficient certainty with regard to the identity of the parties. One possible solution might be an electronic signature within the public blockchain to demonstrate that the transactions, which are recorded in the blockchain, have actually been made by the authorized (and identified) user. Another possible solution to sufficient identity the parties is to create a private blockchain where the contracting parties are identified and approved in advance, for example by a Trusted Third Party.

Although smart contracts will not immediately establish the identity of the contracting parties with sufficient certainty because it can be concluded electronically, some solutions can be found in the use of electronic signatures and private blockchains where a Trusted Third Party plays an important "identifying" role.

e. *Article 6:227b BW*

Another important issue that has to be taken into account is that several additional requirements might be applicable according to article 6:227b in conjunction with article 3:15d BW, in the event a Trusted Third Party is involved with for example the identification of the parties. Article 3:15d BW creates additional obligations for *service providers of the information society*: this definition covers all services that are normally, but not necessarily, provided for remuneration, at a distance, via electronic equipment for the processing (including digital compression) and storage of data.[184] Such service provider is obliged to provide information about his identity and address of establishment.[185] Article 6:227b BW builds upon this and determines that these service providers provide information to the other party before an electronic agreement is concluded in a clear, comprehensible and unambiguous way. The information needs to contain:

(a) the way in which the contract will be concluded and in particular which actions are necessary for this;

(b) whether or not to archive the agreement after it has been established, and, if the agreement is filed, how it will be available to the other party;

(c) the way in which the other party can become aware of actions that he does not want, as well as the way in which he can rectify them before the agreement is concluded;

(d) the languages in which the contract may be concluded and:

(e) the codes of conduct and the manner in which these codes of conduct can be consulted electronically by the other party.[186]

[183] The linking to persons does not automatically lead to *identification* of the person. As stated before, people might use other names to become anonymous.

[184] See point 17 of the Consideration of Directive 2000/31 EC. This does not include agreements which are concluded through email or other individualized forms of electronic communication.

[185] Dutch Blockchain Coalition, "*Smart contracts as a specific application of blockchain technology*", 2017: p.25

[186] Dutch Blockchain Coalition, "*Smart contracts as a specific application of blockchain technology*", 2017: p.25

As soon as a Trusted Third Party is involved, these additional requirements also might to be taken into account in case it concerns a service provider according to article 6:227b BW.

f. Conclusive remarks

The abovementioned requirements of electronic contracting under Dutch contract law make clear that several important aspects have to be taken into account with respect to smart contracts in case Dutch contract law requires a written agreement, but an electronic contract also might be sufficient, provided that a number of conditions are met.

Where the use of an electronic signature within a smart contract has the ability to tackle most of the requirements, like authenticity and moment of conclusion, it can be concluded that a smart contract still faces some uncertainties relating to the identity of the parties and consultability of its content. The identity of the parties could be sufficient guaranteed by electronic signatures on a public blockchain or private blockchain, where a Trusted Third Party plays an important identifying role. However, the consultability of a smart contract remains a stumbling block according to my view. Although it is possible to deduce the operation of the compiled form in case the source code is made available to all of the parties, it still remains practically impossible to understand the smart contract for parties who lack the needed knowledge with regard to "source code" or "compiled form" and translating it to information which is needed to consult and interpret a smart contract. In the event that Dutch law requires a written agreement but also provides a possibility that an electronic contract will be sufficient, it might be problematic to meet all the additional requirements when a smart contract is used. For example, the consultability of leasehold agreements[187] is very important for the rightsholders and other involved parties. In case only a source code or compiled form is handed to these parties, most of them will have no clue what the content or meaning of the smart contract will behold. For buying or selling a house it is even legally not allowed to do it electronically on the basis of article 6:227a section 2 BW. According to the Dutch legislator, it cannot be properly maintained that in all cases it would be desirable if it could be realized electronically[188], let alone in a smart contract.

Another point which is worth mentioning is the fact that contracts which do not have to be in writing, do not per se have to comply with the requirements of article 6:227a BW due to the fact that these requirements are only legally required for contracts which have to be in *writing*, according to Dutch law. The freedom of form results in the situation that smart contracts do not have to contain the abovementioned additional requirements, even though these requirements would have a positive impact on the applicability of smart contracting on itself, according to my view. A simple transaction of cryptocurrency, for instance, does not have to comply with the requirements of article 6:227a BW since it does not have to be in writing according to the law and, therefore, can be free of form. Some smart contracts, which are stored on the blockchain, already have electronic signatures to establish sufficient

[187] See article 7:317 BW. These agreements need to be in writing.
[188] Kamerstukken II 2000-2001, 28 197, nr. 3: p. 55

certainty with regard to the identity, moment of conclusion and identity[189]. However, it would benefit the operation of smart contracts altogether if also the simplest smart contract would have to comply with these requirements (including some additions to overcome the aforementioned problems) in order to utilize all of the possibilities.

As mentioned before in subsection '*e'* of this paragraph, some contracting parties also need to provide information about their identity and address of establishment according to article 6:227b BW. In case these parties do not comply with these disclosure requirements, it can be said that it will be possible that the other party makes an appeal on a legal error and terminates the contract on the basis of article 6:228 BW. Nonetheless, this protection mechanism is only addressed to agreements which have to be in writing according to the Dutch law and can be concluded electronically in accordance with article 6:227a BW. Smart contracts in general contain complex programming language which might be difficult to understand. Deducing useful information, like the requirements of article 6:227b BW, can be seen as even more problematic. Contracting parties might than conclude smart contracts with a potential risk of legal error on the side of one of the parties. According to my view, it would be beneficial to the drafting of smart contracts if similar disclosure requirements, together with the requirements of article 6:227a BW, have to be satisfied before even concluding a smart contract under Dutch law. This will create more certainty for both contracting parties and prevent a large portion of potential disputes.[190] Unfortunately for now, no requirements for contracting by smart contracts or blockchain are included in the Dutch Civil Code. Perhaps in the near future will such a regulatory step be taken in order to run the contracting process of smart contracts in a more suitable way.

2.4: Interpretation of smart contracts under Dutch law

The previously discussed "The DAO-hack" made a fundamental contrast clear whether or not "code is law" or whether there has to be a human correction option. The majority of the DAO participants (around 80%) did not want to follow the code and formulated another view on smart contracts and its code: a view which is more in accordance with the law of contracts. *Scott* argued as follows with regard to contracts and automation of it: *"Contracts are representations of frequently ambiguous, unpredictable and messy relationships between imperfect humans with imperfect knowledge. Such relationships cannot easily be pre-programmed, and much of the work of lawyers involves resolving and interpreting contracts in light of changing realities."*[191] The "The DAO-hack" was unfortunately no exception to Scott's view.

It sounds perhaps utopian to move away from a system where no law is applicable and where code will be law, especially within a short period of time. With regard to the current

[189] For example: smart contracts which are stored on a private blockchain of a Trusted Third Party.

[190] Disputes about legal error about with whom they conclude a smart contract or what they agreed to, will occur less frequent. In that case do parties need to comply with requirements of consultability, identity etc. (see also article 6:227a and 6:227b BW)

[191] B. Scott, "*How Cryptocurrency and Blockchain Technology Play a Role in Building Social and Solidarity Finance?*", UNRISD Working Paper 2016-1: p.13

Dutch legal system and jurisprudence, it will also be incredibly difficult to have a "code *is* law"-based system in the near future, or even at all.

In the event that contracting parties decide to put their agreements in a smart contract, many things can go wrong. One of the most obvious problems that may arise, is the situation that the smart contract executes which is eventually not in accordance with the intention of one (or more) of the contracting parties. Several reasons might cause this problem with smart contracting.

Firstly, since most of the contracting parties nowadays do not know how to program a smart contract, it can be said that most of the time a programmer will be hired to set up the smart contract. The programmer knows how to program a smart contract and will program it according to the instructions that he received from the contracting parties. However, programmers are not experienced contract lawyers and might program the smart contract not in accordance with the initial intentions of the contracting parties. In the worst-case scenario, this might lead to an unwanted outcome and eventually lead to a dispute between the contracting parties.[192] In case the contracting parties would know how to program a smart contract, it can be assumed that less problems would arise with regard to the aforementioned problem. Unfortunately, a very small percentage of people know how to program a smart contract at the moment of writing this thesis. Therefore, it can be seen as plausible that in the near future problems will arise with regard to this situation.

Secondly, another cause can be found in the fact that the agreements have to be put down in programming language. Presumably, some agreements will be incredibly difficult to fully implement within programming language and eventually lead to a situation where the smart contract does not perfectly match with the intentions of the parties. Moreover, those who program the smart contract[193] may use certain functions of the programming language in a wrong way or misinterpret the operation of the programming language: a *bug* or *error* within the smart contract arises in these situations.

As a result, disputes might arise between contracting parties regarding their agreements that have been codified in a smart contract. In this case, parties can go to the court for a ruling about the explanation and interpretation of the smart contract. Nonetheless, smart contracts are a new way of contracting compared with the traditional paper-based contracts. A traditional contract is very different compared with a smart contract on the blockchain: the way of setting up a smart contract, executing a smart contract and maintaining a smart contract is completely different. The question might arise whether or not the current interpretation doctrine of the Dutch courts is still applicable when it comes to the interpretation of smart contracts.

One problem that also has to be tackled before even discussing the interpretation doctrine, is how the judge(s) can deduce the intentions of the contracting parties from a smart contract that has been concluded in programming language. It can be assumed that the vast

[192] It is beyond this thesis to discuss the possible liability of the programmer regarding the contracting parties.
[193] It can be programmers or contracting parties themselves.

majority of the judges does not know how to read the programming language of a smart contract, let alone to deduce the intentions of contracting parties out of it[194].

The first possible solution of explaining the content of the smart contract and possibly come to the intentions, is to make use of an expert on the field of smart contracting. According to article 152 (and paragraph 6 of Ninth Section) of the Dutch Code of Civil Procedure, evidence can be delivered by all means. Consequently, this expert might give the judge some extra information and "translate" the programming language into language that is understandable for a judge and point out several features, bugs or errors and make it possible for a judge to give a ruling regarding a dispute about a smart contract.

While the solution of an expert is used afterwards, other solutions in advance that will help judges with "reading" smart contracts might be used even before a problem arises. As already discussed in paragraph 2.2.1, one solution can be found in adding a written part to the smart contract with the parts that cannot be easily translated into programming code and point out the intentions of parties. Stated differently, this means that the non-human execution is contained in computer code (like checking whether the conditions have been fulfilled), while the provisions and obligations of more human nature (like intentions, enforcement of fulfillment and settlement of disputes) are described is a separate part that is set up in natural written language. The smart contract can then be executed by an automated contract (code), where some specific agreements that cannot be coded that easily, are attached as a traditional written agreement. However, the written agreement and automated part do have to use the same structure in order to be useful for a thorough explanation of the smart contract. Another solution which can be used in advance might be developing a standard where certain 'fixed' codes in a smart contract have a 'fixed' meaning and that for these codes the judge will explain them in only one way. By using this code, programmers and contracting lawyers need to take these standards into account when they set up a smart contract. Programmers may also be obliged to write up their actions when they are programming the smart contract in order to track down the intentions.

Abovementioned options might eventually help a judge to deduce the intentions of the contracting parties from a smart contract. Translating and understanding a smart contract can be seen as the base for a judge to give a verdict about a dispute. After all, judges will bring no justice to smart contract disputes without a decent level of knowledge about the content and intentions, that can be seen as the base of the smart contract.

2.4.1: Haviltex-norm and textual interpretation norm

Contracting parties under Dutch law have the possibility to go to the court in case they cannot solve the problems between themselves.[195] Under Dutch law, contracts need to be assessed and interpreted on the basis of the rules of law that apply under international private law.

[194] See paragraph 1.2 for the example smart contract which has been written in Solidity. It can be assumed that most judges will have no clue about the meaning of the content or intentions of this contract.

[195] To bring an action to the court and getting a fair trial are even a fundamental right of people according to article 6 ECHR and article 17 of the Dutch Constitution ("Grondwet").

Because of the fact that (smart) contracts contain many agreements between parties, it can be said that these agreements under Dutch law always need to be assessed and interpreted by the so-called *objectified subjective interpretation* according to the Dutch Supreme Court, in case of disputes or conflicts between contracting parties: this standard is also known as the "Haviltex-norm".[196] Instead of pure textual interpretation and assessment of a contract, all circumstances (context) of the contract play a decisive role with regard to the assessment under Dutch law and the context of that specific contract will always be valued according to the reasonableness and fairness of article 6:248 BW ("*redelijkheid en billijkheid*").

In some exceptional cases the context will not play a decisive role in the assessment of the contracts and will the *textual* terms of the contract itself be leading, according to the ruling of the Dutch Supreme Court in the case of *Meyer Europe/PontMeyer*.[197] These cases do normally relate to commercial contracts between professionally operating parties. Factors like the nature of the transaction, the scope and detail of the contract, the manner of creation, the presence of an entire agreement clause and extensive negotiations between the parties play a decisive role whether or not the textual interpretation can be followed by the court.[198] Consequently, the question may arise whether the textual interpretation might be leading in case smart contracts are concluded as a commercial contract between professionally operating parties after extensive negotiations. Professionally operating parties can, for example, decide to sell their shares or company (asset transaction)[199] by using smart contracts.

However, the Dutch Supreme Court did rule in the so-called *Lundiform/Mexx*-case[200] that the Haviltex-norm is still leading for the interpretation and explanation of contracts. Even if the contracts are commercial contracts between professionally operating parties, other circumstances of the contract play a decisive role in the (non-textual) interpretation of the contract: even entire agreement clauses, which can be seen as clauses to accept the outcome of a contract, are subject to the objectified subjective assessment (= Haviltex) and need to be valued according to all circumstances and reasonableness and fairness.[201] With regard to smart contracts, this starting point would mean that the code which contains all of the agreements and obligations needs to be assessed according to its context and will be valued according to the open norm of reasonableness and fairness.

As a result, the abovementioned assessment of smart contracts does not relate to the principle of code *is* law and smart contracts in general.[202] The deterministic (binary) design of the code within the smart contract prevents that agreements of the smart contracts can be

[196] Called after the court ruling: HR 13 maart 1981, ECLI:NL:HR:1981:AG4158, NJ 1981, 635 (Haviltex)
[197] According to HR 19 januari 2007, LJN AZ3178, NJ 2007/575 (Meyer Europe/PontMeyer) en HR 29 juni 2007, LJN BA4909, NJ 2007/576 (Derksen/Homburg).
[198] HR 29 juni 2007, LJN BA4909, NJ 2007/576 (Derksen/Homburg)
[199] In a way of speaking: this is an example and might hypotactically be possible in this case. However, it might in reality be technical impossible due to the technical limitations of smart contracts (see paragraph 1.5.4).
[200] HR 5 april 2013, ECLI:NL:HR:2013:BY8101 (Lundiform/Mexx)
[201] Also according to HR 5 april 2013, ECLI:NL:HR:2013:BY8101 (Lundiform/Mexx)
[202] J.B. Schmaal & E.M. van Genuchten, "*Smart contracts en de Haviltex-norm*", Tijdschrift voor Internetrecht, nr.1 maart 2017: p. 16

interpreted by their context or background: it simply executes what has been put into the code. When the deterministic code of the smart contract *is* law, no context will be taken into account with the assessment of the content and consequences of the smart contract.[203] Additionally, the smart contract will perform as soon as the conditions within the smart contracts are met, even if the resulting situation is unwanted or unconsidered by contracting parties.[204] Where parties could prevent or stop execution and start exercising their rights under a "traditional" contract in case of changed circumstances or in case the resulting situation turns out to be different than expected, it can be assumed that this will be practically impossible with agnostic and self-executing smart contracts. The circumstances and context, that are leading for the Haviltex-norm are simply not compatible with the principle of "code *is* law" and smart contracts in general. In all probability, artificial intelligence will in the near future help putting the context within the code and, for instance, anticipate on changed circumstances. Unfortunately, for now, it can be concluded that the state of the technology is not so advanced that the context and circumstances can be completely implemented within the programming language of a smart contract.

Another point to mention with regard to the Haviltex-norm can be found in the fact that the ruling in the Haviltex-case relates to *contractual provisions* in an agreement. As a result, the question can arise whether or not the code of a smart contract fits within the meaning of "contractual provision". Codes can namely be seen as something completely different than provisions due to the fact that codes do not immediately reflect the intentions of parties. Codes are put into a smart contract to make it possible to execute the contract. Because of this reason, it can be said that codes reflect an *executional* element: codes make it possible that a smart contract will work, in other words *execute*. Written provisions in a traditional contract will also make the contract "work" but the written agreement's execution is not dependent on these written provisions since the execution is done by the parties themselves. Hence, it can reasonably be argued that codes have a different effect compared to written provisions. Where the Haviltex-norm reflects provisions in a traditional written contract, smart contracts entail codes which are stored on the blockchain and used in a way that make a smart contract executable. This way of contracting is completely new and not even regulated until now.

2.4.2: Revised interpretation norm and factors for smart contracts

Due to the fact that smart contracts (and especially the codes) have a different effect and execution compared to traditional contracts, it can be assumed that the current interpretation norms (i.e. Haviltex and textual norm) might cause some problems and that contracting by means of smart contracts on the blockchain require a different approach and weighing of

[203] J.B. Schmaal & E.M. van Genuchten, "*Smart contracts en de Haviltex-norm*", Tijdschrift voor Internetrecht, nr.1 maart 2017: p. 16

[204] This might be caused by an error (bug) within the code itself.

factors when it comes to the interpretation. After all, interpretation according to a strict Haviltex-norm or textual interpretation might cause friction with respect to smart contracts.

Smart contracts are set up to express the agreements and intentions of the contracting parties in programming language. Therefore, it sounds logical to look at the meaning and intentions that the parties could reasonably give in the context of the smart contract. The leading Haviltex-norm would be suitable according to this point of view. However, as already discussed in the previous paragraph, some elements differ very much from the traditional way of contracting. On the contrary, a pure "textual"[205] interpretation would also not be practical since it will be incredibly difficult to interpret the code in a way that can be translated into language that is understandable for judges, or even the contracting parties themselves. It can be concluded that a pure textual interpretation of a smart contract also might lead to an unwanted outcome in case some intentions are translated poorly into programming language.

Although smart contracts differ very much from traditional contracts, some possible aspects can be taken into account to come to a more suitable interpretation doctrine. The choice between a more "Haviltex-based interpretation" or to give a bigger consideration to a more textual interpretation, can be made by taking the following factors into consideration.

The first important factor that has to be taken into account is which kind of blockchain network is used. As discussed in paragraph 1.2.1, a distinction can be made between public and private blockchains. One important difference between these two blockchain networks can be found in the identification and authentication: while public blockchains do not have any central authority that takes care of the identification and authentication of the participants and almost everyone can participate on the public blockchain, it can be said that a private blockchain, on the other hand, does an identification and authentication of each participant and only allows a confined number of participants. In contrast to participants of a private blockchain, most of participants on a public blockchain do not know with whom they conclude a smart contract. Consequently, the intentions of the participants are most of the time not known by each other. Furthermore, transactions which are put on a public blockchain are most of the time different compared with transactions on a private blockchain. Bitcoin-transactions, for example, are placed on a public blockchain and do most the time not require the knowledge of the other party's intentions due to the nature of the transaction. On the contrary, selling shares or casting a vote on private blockchains require most of the time knowledge about a person's intention. Therefore, using public blockchains might be a reason to give a bigger consideration to a more "textual" interpretation to the code of the smart contract, while using a private blockchain requires looking to all circumstances and intentions (context) of the contract.

The second factor is the nature of the smart contract and the knowledge of the contracting parties about smart contracts. Although the Dutch Supreme Court ruled that the

[205] Although programming language within a smart contract does not consists real "text", in the following of this thesis will the term "textual" be used to indicate the different use of interpretation methods of smart contracts. While the Haviltex-norm uses the context as leading principle, it can be said that the textual interpretation is the direct opposite of this norm. The use of the term "textual" can therefore be seen as pure clarifying term to point out different views on the interpretation of a smart contract.

Haviltex-norm is still leading in the case of commercial contracts[206], it can be argued that a textual interpretation would be more suitable in case the smart contract is set up by professional acting parties who have knowledge about programming and smart contracts. It can be expected from these professional acting parties that they record their mutual rights and obligations accurately in a smart contract. Therefore, a more textual interpretation of a smart contract might be more applicable in these situations, also due to the fact that in these cases professional acting parties consciously and with knowledge of smart contracts chose to put their agreements in a smart contract.

Beside the abovementioned factors, it can also be argued that the use of a programmer might lead to another interpretation norm. In case contracting parties have enough knowledge about smart contracts and know how to program a smart contract, it can be reasonably assumed that a judge may lay bigger emphasis on a textual interpretation of the code within the smart contract. On the contrary, when a programmer would program the smart contract on behalf of the parties, the intention of parties might be more leading due to the fact that a programmer might make a mistake or set up a code which is not in accordance with the intention of the parties. After all, it is still practically impossible to translate natural language into code without making a significant compromise in the quality of the output of such conversion.[207] In case a programmer would draft the smart contract, several measures could be taken to make the intentions of the parties clear like writing down each step that the programmer took to express the intentions into codes.

In conclusion, the abovementioned factors could be taken into account by a judge to interpret a smart contract in a better way compared to a pure "Haviltex-based" or textual norm and do justice to the nature of smart contracts in general. Programming language is completely different than traditional written language and requires a different approach when it comes to interpretation and assessment of a smart contract. Although all circumstances will need to be taken into account, different factors will play a more decisive in the interpretation of smart contracts compared with traditional contract.

2.5: Termination of smart contracts under Dutch law: repudiation[208] and rescission[209]

At the moment of writing this thesis, no Dutch court ruling has been given regarding a dispute about of smart contracts.[210] It can reasonably be expected that in the near future a court ruling has to be given regarding a dispute about smart contracts. In all probability, a Haviltex-norm based interpretation will be leading for the assessment of the smart contracts since all contracts (including electronic contracts) under Dutch law have to be assessed according to

[206] HR 5 april 2013, ECLI:NL:HR:2013:BY8101 (Lundiform/Mexx)
[207] H. Surden, '*Machine Learning and Law*' (2014) 89 *Washington Law Review* 87, 91-95
[208] Translated in Dutch: "ontbinding". Translations taken from article of R.P.J.L. Tjittes, "*Veelvoorkomende misverstanden bij gebruik van Anglo-Amerikaanse termen in het internationale contracteren*", Contracteren 2008/2: p.41. This article discusses the best translation of Dutch law into English.
[209] Translated in Dutch: "vernietiging".
[210] Research has been done in the digital database of court rulings in the Netherlands: see website of https://uitspraken.rechtspraak.nl.

that norm. Consequently, all circumstances and context will then need to be taken into account and valued according to the reasonableness and fairness to eventually judge whether or not the outcome of the smart contract will be affected.

With traditional written agreements, several situations could occur that could turn in to a dispute between the contracting parties. For instance: one of the contracting parties fails to comply with the agreements that have been made. In case this dispute would go to the court, the result could be *repudiation* of the contract ("ontbinding") on the basis of article 6:265 BW.[211] Repudiation of a contract does not have a retroactive effect according to Dutch law, which means that the agreement will terminate from the moment of repudiation. Furthermore, repudiation of a contract can also be done by the parties themselves by writing, even without the intervention of a judge.[212] Beside repudiation of a contract, it will also be possible that *rescission* of a contract ("vernietiging") will be invoked. Rescission of a contract can be invoked in case there is a lack of will ("wilsgebrek") on the side of one of the contracting parties and might be caused by several reasons, like fraud or error. Unlike repudiation does rescission have a retroactive effect[213] according to article 3:53 BW and results in the fact that the rescission works back till the date on which the agreement has been concluded: in other words, the contract will be considered as if it never existed before. The consequences and events that already occurred, have to be turned back according to Dutch law, except in case the consequences cannot be undone easily or without far-reaching effects.[214]

However, the aforementioned termination possibilities of contracts under Dutch law will raise some interesting questions with regard to self-executing and "immutable"[215] smart contracts. The first question can be raised about repudiation of a smart contract according to article 6:265 BW. For instance: if one party does not comply with the agreements that have been made, the other party could invoke his right of repudiation on the basis of article 6:265 BW. Due to the fact that smart contracts will automatically execute as soon as the conditions have been met, most of these situations will occur less frequent since they remove the human intervention out of the loop. Smart contracts can be more efficient due to the fact that they will execute automatically and will not be dependent of the whims of the contracting parties. With respect to failure of complying with agreements like payment on a certain date, it can be expected that it will occur less frequent compared with traditional agreements, due to the fact that the contract itself will execute. On the other side, in case the conditions are not met, the smart contract will simply not execute. Unfortunately, still some situations might arise where one of the parties can invoke their right of repudiation with smart contracts. Take for example the situation in which parties agreed to pay in installments depending on several conditions that have been put down in a smart contract. In the event that some of the

[211] Unless the shortcoming, in view of its special nature or minor significance, does not justify this repudiation with its consequences (see Section 1 of Article 6:265 BW).
[212] A written notice to the other party is already sufficient in many cases.
[213] Ex tunc situation.
[214] See article 3:53 section 2 BW. The judge may than rule differently in these cases.
[215] Paragraph 1.5.2 discussed the immutability of smart contracts, which can be seen as a "myth" according to my view.

conditions have been fulfilled and automatically paid by the smart contract, it could occur that one condition and payment might not be fulfilled for a long time. Consequently, one of the parties may than invoke his right of repudiation on the basis of article 6:265 BW and reimbursement of the payments that have already been made, according to article 6:271 BW. Nevertheless, repudiation and reimbursement of the smart contracts will be incredibly difficult as soon as they are stored on the blockchain. Unlike traditional agreements, it can be assumed that smart contracts face more difficulties when it comes to termination since they are irrevocable and immutable as soon as they are effectuated and stored on the blockchain.[216]

Moreover, a similar problem will arise with regard to the rescission of a smart contract. As discussed before, rescission does have a retroactive effect and means that the rescission works back till the date on which the agreement has been concluded: the consequences or events that already occurred have to be turned back according to the Dutch law.[217] Turning back the consequences due to the retroactive effect and turning back to the previous situation will be incredibly difficult with irrevocable and immutable smart contracts, especially when these smart contracts are stored on the blockchain. Imagine the situation in which a contracting party has to turn back all of the consequences of a smart contract on a public blockchain. First of all, the majority of the nodes need to agree with the change and, secondly, all of the consequences need to be changed as well. It can be assumed from a practical and technical point of view that this situation is definitely not desirable for the efficiency and process of smart contracting on a blockchain (i.e. computationally impossible).

2.5.1 Possible solutions for terminating smart contracts

As a result of the irrevocable and immutable nature of smart contract and blockchain, it will hardly be possible (if not impossible) to change or to terminate a smart contract. Especially the retroactive effect and associated consequences will cause many difficulties. Another difficult aspect is the change itself: even if all parties within a blockchain agree to change it, the contract *itself* cannot be changed as in "adding or changing" some components of the code.[218] Due to these difficulties do some alternative solutions need to be included to solve the abovementioned challenges.

A first possible option which could be done by the parties themselves is building a so-called "exit option" within a smart contract, which has to be implemented in the smart contract from the beginning (i.e. the programming of the smart contract). This exit-option could terminate a smart contract in case the majority of the nodes agrees to terminate it (i.e. a consensus mechanism) and transfer the amount of cryptocurrency that is still on the smart contract to a new smart contract.[219] This exit option is suitable for some adaptations within a smart contract which will not require drastic changes or invoke a retroactive effect. For

[216] H. Schuringa, "*Enkele civielrechtelijke aspecten van blockchain*", Computerrecht 2017/254: § 3.2
[217] See article 3:53 BW. Dutch contract law considers these contracts as if they never existed.
[218] E. Tjong Tjin Tai, "*Smart contracts en het recht*", NJB 2017/146: §8
[219] This new smart contract can be held by an independent third party and act like an escrow account.

example: in case new regulations apply to the smart contract or if one of the parties changes their bank account number, an exit option will make these changes possible. Although the contract itself will not be changed, the newly activated contract will contain the needed changes after the exit option has been invoked.

In short, an exit option might be useful in cases where the parties want to change a minor part and both parties also agree to change it. However, there might also be situations where parties do not agree about the possible solution. As previously discussed in this paragraph, some situations can arise where one party might invoke his right of repudiation or rescission and a court might need to intervene and solve the situation. As a consequence, these cases ask for other solutions which possibly can be used without both parties' approval.

In case a smart contract contains an unwanted bug or error which might have a negative impact on one of the contracting parties, a human intervention possibility can be seen as useful mechanism. *Greg Medcreaft*, chairman of the Australian Securities and Investments Commission (ASIC), argued that institutions (like financial institutions and courts) should also be able to use so-called "kill switches" to stop computers automatically executing smart contracts "in times of stress".[220] These kill switches can also be used to prevent that illegal activities could be put into action. For instance: a smart contract could be set up to kill a person and send an amount of cryptocurrency to a specific person if he or she killed a specified person. The required information of killing a person could be implemented into the smart contract by an oracle by confirming the person's dead.[221] Another possible illegal smart contract would be selling drugs through a smart contract.[222] Also less harmful situations could be prevented, like an obvious mistake in the amount of cryptocurrency that has to be paid. In these cases of an obvious mistake, it can be said that under Dutch law always need to be checked whether the other party could have *legitimate expectations* about the offer. According to article 3:35 BW[223], an agreement will be concluded even in case one of the contracting party's will is lacking but that the other party could *reasonably rely* on the offer that has been made to him, under the given circumstances.[224] However, in case the accepting party *should have known* that it was an obvious mistake and the other party did not want to conclude an agreement under the given terms (lacking consensus of intentions), a court can rule that no agreement was concluded due to a lack of will.[225] Under these circumstances an exit option would not be desirable due to the fact that one of the contracting parties may not

---

[220] J.B. Schmaal & E.M. van Genuchten, "*Smart contracts en de Haviltex-norm*", Tijdschrift voor Internetrecht, nr.1 maart 2017: p. 14

[221] This could be checked in a death register from a certain municipality or country.

[222] Similar to the online Internet drug marketplace Silk Road.

[223] English translation of article 3:35 BW: *"Towards him who has interpreted another person's statement or behavior, in accordance with the meaning that he reasonably could give to it under the given circumstances, as a statement with a certain content of this other person addressed to him, cannot be appealed to the absence of a with that statement corresponding will (intention)."*

[224] Asser/Hartkamp & Sieburgh 6-III, 2014: nr. 139. Most of the time can contracting parties only rely on the offer of the other party. In case the circumstances under article 3:35 BW are fulfilled, an agreement (contract) can be concluded without the offering party's consensus of intention about the offer and acceptance.

[225] These "too good to be true"-cases are not considered as legal binding agreements in case the accepting parties should have known that a mistake has been made. Only when the expectations are legitimate and the accepting party could not have known that a mistake had been made, will a legally binding agreements be established.

want to change or terminate the smart contract.[226] However, the use of a kill switch in a smart contract by a court would make it possible to prevent these unwanted situations, even without the parties' consent. Perhaps will artificial intelligence in the near future recognize these unwanted or illegal situations and prevent activating these smart contracts in advance. Unfortunately for now, it can be concluded that the state of artificial intelligence is not at that point yet.

Beside an exit option or kill switch, it might also be possible to assign an independent oracle within the smart contract which can provide the smart contract with certain information and decide what will happen in case of a dispute: a court or arbitration court could, for example, settle a dispute between the contracting parties. Just as with traditional contracts a court could decide whether or not a contracting party can invoke their rights of repudiation or rescission. This input has to be delivered in a way that can be processed by a smart contract.[227] In case of repudiation, a court could use a kill switch method to terminate the smart contract. Since repudiation does not have a retroactive effect, it can be said that the smart contract will terminate from the moment of repudiation. The situation will become more difficult when one of the parties would like to invoke their right of rescission, as a result of the retroactive effect (i.e. the contract is considered to be never existed). One possible solution to solve this problem would be the possibility of the court to set up a new smart contract that will have an exact opposite outcome compared to the previous smart contract and 'overrule' the smart contract that caused the dispute. For example: payments that already have been made could be paid back in case the new smart contract contains that specific obligation. This could also be used in situations where an obvious mistake has been made by one of the contracting parties and the other party should have known that a mistake had been made but still accepted the offer (with all of the negative consequences as a result). In these situations, it can be seen as lawful for a court to overrule the previous smart contract and turn back the consequences.

[226] As discussed before, an exit option will be a good solution in case both parties agree about changing it.
[227] H. Schuringa, "*Enkele civielrechtelijke aspecten van blockchain*", Computerrecht 2017/254: § 3.2

Chapter 3: Future of smart contracts and Dutch legal framework

3.1: Legal vs. technical obstacles of smart contracts

Before starting to discuss the future of smart contracts and investigate deeper into the question what should be done regarding the Dutch legal framework to successfully implement smart contracts in the Dutch legal system, it might be important to make a reflection about the obstacles that are encountered with the use of smart contracts.

Smart contracts are, as discussed in previous chapters, contracts that are represented in code and executed by computers. Originally, smart contracts were contemplated within a limited and niche range of transactions, such as in the area of financial instruments. Progressively, however, the surrounding narrative has become broader, implying that all contracts can be made smart or that many different obligations can be enforced by code.[228] Many technical writings regarding smart contracts argue that smart contracts and blockchain technology are capable of solving the majority of the (legal) problems that are encountered nowadays. However, these technical writings are often driven by ideologically charged arguments that associate the underlying technological features with solving social and economic issues.[229] It is most of the time assumed that technologies like smart contracts and blockchain technology, due to their *innovative* and *disruptive* character, are capable of solving problems that are caused by the lacking abilities of contracts and/or law, which in their turn are not capable of solving these problems and even holding back innovative ways of solving these problems (i.e. using blockchain and smart contracts).

However, as previous chapters made clear, relatively few fundamental *legal* obstacles can be discovered in Dutch (contract) law regarding the use of smart contracts. Nonetheless, as with every new technology, it can be argued that there will be certain (new) legal obstacles to overcome. After all, smart contracts on a blockchain are a completely new way of contracting: the way of setting up a smart contract, executing a smart contract and maintaining a smart contract can be seen as completely different compared to traditional ways of contracting. Therefore, it can be seen as a logical consequence that some changes have to be made within the law to overcome these obstacles. For this reason, the comparison can be easily made between the debate that was held during the early 2000's with the rise of the Internet and the current blockchain debate.[230] As discussed in paragraph 2.2, some scholars believed (and still believe) that the Internet and smart contracts would somehow not need any form of law or legal intervention at all. They tend to sketch a situation where the law is the underlying problem of holding back innovation and use of new technologies. However, these arguments simply overlook the fact that the lack of recourse to legal institutions or legal principles would not only incentivize fraudsters or hackers, but also discourage the use of smart contracts with regard to several (financial) transactions.

[228] E. Mik, "*Smart Contracts: Terminology, Technical Limitations and Real World Complexity*", 2017: p. 1
[229] E. Mik, "*Smart Contracts: Terminology, Technical Limitations and Real World Complexity*", 2017: p. 2
[230] See also paragraph 2.2.2

Ironically enough are the *technical* obstacles itself (i.e. technical limitations of smart contracts) the reason of the hold off with respect to solving the legal problems, instead of the law or legal system. A simple reason for this can be found in the fact that the created expectations of blockchain and smart contracts in technical writings as "legal problems solvers", are running ahead of what the current technology really can deliver. Most of the people nowadays are becoming accustomed to life-changing technologies like the Internet, e-mail and smartphones. As a result, the majority of these people expect more and more from new technologies.[231] Nevertheless, it can be concluded that smart contracts are still facing several technical limitations which make it practically impossible to replace all traditional ways of contracting. For example: the more complex a smart contract gets, the more difficult it will be to put all of the agreements within a smart contract due to so-called "block gas limits" or incapability of reusing mature financial libraries, algorithms and market data.[232] Due to these limitations, smart contracts cannot replace complex traditional written contracts, let alone solving their legal obstacles or uncertainties.

Beside these technical limitations, it can be said that the scalability and applicability of smart contract-based contracting on a large scale remain to be minimal. Probably the biggest current scalability issues with public blockchain networks, like Ethereum for example, are that every node in the network has to process all transactions and has to store the entire state of every account balance, contract code and storage. Despite the fact that this provides a large amount of security, it greatly limits scalability to the point that a blockchain cannot process more transactions than a single node.[233] Although one would think that a network with thousands of nodes should be able to have more throughput than a single node, it can be concluded that this is not the case with Ethereum or other public blockchain networks in general.[234] As a result, some projects have been set up to solve these scalability issues within Ethereum.[235] On the contrary, permissioned (i.e. private) blockchain networks have the ability to set up different types of nodes with different responsibilities which make it possible to allow them to configure a network of nodes to scale independent of each other.[236] Another problem that can be found is setting up a framework for smart contracting, which remains to be a big hurdle for most companies nowadays. Even if companies can overcome the technical and scalability issues, companies still face the problem that most of their clientele have no willingness to change the way of contracting. This creates, unfortunately, a prisoner's dilemma: the most significant benefits of smart contract adoption come when numerous commercial entities begin to automate their data-driven interactions using smart contracts stored on a blockchain. However, in a single commercial entity's context, in the short-term

[231] See also blog of E.P.M. Vermeulen, "*Technology Isn't Fast Enough in a Platform Economy*": find online at https://hackernoon.com/technology-isnt-fast-enough-in-a-platform-economy-d6627fd97a27
[232] See also paragraph 1.5.4 regarding these technical limitations.
[233] M. Scherer, "*Performance and Scalability of Blockchain Networks and Smart Contracts*", UU, 2017: p. 21
[234] M. Scherer, "*Performance and Scalability of Blockchain Networks and Smart Contracts*", UU, 2017: p. 21
[235] See for example *Plasma*. Plasma is a proposed framework for incentivized and enforced execution of smart contracts which is scalable to a significant amount of state updates per second (potentially billions) enabling the blockchain to be able to represent a significant amount of decentralized financial applications worldwide.
[236] M. Scherer, "*Performance and Scalability of Blockchain Networks and Smart Contracts*", UU, 2017: p. 21

the design and deployment of smart contract systems would for many applications be less efficient than carrying on with a centralized software stack.[237] As a result of the prisoners' dilemma, many commercial entities or institutions do not want to make the needed investments regarding smart contracts, despite the fact that many of these entities show a great interest for using smart contracts stored on a blockchain[238]. Unfortunately, this creates a situation where almost no entity is willing to switch to smart contract-based contracting.

As the past already proved, the technology shall improve in the upcoming years. It is important to note that the idea of smart contracts was already coined in the early 1990's but the idea did not gain a lot of interest, due to the fact that the technology at that time was not at a point where smart contracts could get implemented easily. With the introduction of blockchain technology, the idea of smart contracts also gained more interest since it established a technology that made implementation on a larger scale possible. Despite the technical limitations for now, it can be assumed that these technical limitations will be solved for a major part in the near future. However, due to the fact that smart contracts are a completely different way of contracting, it also requires a different way of legislative approach. It is tempting to weigh smart contracts against traditional ways of contracting and use current legal approaches to overcome the legislative issues of smart contracts. As previous chapters explained, smart contracts and their underlying technology differ in a variety of ways with respect to traditional contracts. Ignoring these differences in the legislative adaptations would work against smart contracts in general. Therefore, it can be concluded that the law has to adapt on some points before it really holds back the implementation of smart contracts, especially since no regulatory recognition has been made in Dutch law so far. After all, new technologies like blockchain technology and their applications are already changing the technological landscape today (see for example Bitcoin).

3.2: Future regulations, ethics and principles

Chapter 2 of this thesis made clear that the current contract law framework of the Netherlands is sufficiently prepared to encompass smart contracts. Beside several minor concerns, no fundamental *legal* obstacles can be found. At the moment of writing this thesis, no Dutch legislation has been introduced that clarifies contracting by means of smart contracts.[239] On the contrary, some federal states in the United States[240] have already introduced or considering to introduce legislation that purports to clarify that contracts cannot be "denied legal effect, validity or enforceability merely because the contract is processed, executed, or otherwise enforced via smart contract computer code."[241] However, some of these statutes contain definitions of terms such as "blockchain," "distributed ledger

[237] https://monax.io/learn/smart_contracts/

[238] See paragraph 1.6 for examples of smart contract experiments that are initiated in many industries.

[239] Only some encouragements to investigate deeper into the possibilities of smart contracts and their legal impacts have been made by the Dutch Parliament so far: see Kamerstukken I, 2016/2017, 33 009

[240] Among these states were Arizona, California, Florida, Nebraska, Nevada, New York and Tennessee. Arizona and Tennessee already passed legislation regarding smart contracts.

[241] J.D. Hansen et al, "*More Legal Aspects of Smart Contract Applications*", Perkins Coie LLP, March 2018: p. 21

technology" and "smart contracts" that have come under significant fire by academics due to the fact that these terms were not consistent enough or not in accordance with the underlying technology.[242]

The regulatory attempts of the federal states in the United States show the difficulties of regulating smart contracts and other blockchain applications. Even finding a consistent term of "smart contract" can be seen as problematic, partly due to the different gradations or categories which are described as smart contracts: there are smart contracts which completely consist out of computer code, while there are also smart contracts which exist out of a "real" contract, where the execution is fully or partially automated. According to *Mik*[243], it can be argued that some inconsistencies arise due to the poor translation of technical writings about smart contracts into a correct use of legal terms and principles. Nonetheless, it is very difficult to translate programming language of a smart contract into consistent legal terms that are used nowadays. However, laws or regulations should not attempt to define technologies that do not have a widely held definition in their relevant technical communities since it could lead to confusion and unnecessary complexity.[244]

Irrespective of the relative accuracy of the definitions contained in such laws, it is also difficult to regulate something which is constantly being disrupted as a result of technological changes. Smart contracts and blockchain technology are constantly updated and tested by programmers, mathematicians or fanatics online. The number of projects which have been set up to overcome the imperfections of smart contracts are a perfect example of how changeable and disruptive the technology can be. In case the Dutch Parliament would decide to regulate smart contracts today, that legislative process could take up to two years.[245] As a result, the risk will arise that the regulation will be inconsistent or not up-to-date with the current state of the technology at the moment of implementation. It can be assumed that this situation is not desirable for the applicability or development of smart contracts.

One possible way to overcome this problem is to adapt the process of regulating these new technologies. In the Dutch financial sector, for instance, have already some experiments been set up with so-called "*regulatory sandboxes"*, which make it possible for innovative companies to test products, services or business models in practice without having to comply with all legal obligations and limitations.[246] Not only will this include advantages for companies that want to use new technologies (like using smart contracts), but the government itself will benefit from it as well since the government will be able to see what the effect and impact of innovations are in practice, so that the risks, social value and the benefit for potential users can be assessed in a better way. Moreover, policies regarding new technologies can be easily adjusted if necessary in the light of new developments. As the regulatory attempts of the

[242] J.D. Hansen et al, "*More Legal Aspects of Smart Contract Applications*", Perkins Coie LLP, March 2018: p. 21
[243] E. Mik, "*Smart Contracts: Terminology, Technical Limitations and Real World Complexity*", 2017: p. 2
[244] According to Alexander Hinkes of Stern University, New York: https://www.technologyreview.com/s/610718/states-that-are-passing-laws-to-govern-smart-contracts-have-no-idea-what-theyre-doing/
[245] See for observation of the efficiency of Dutch legislative power: H. ten Napel, "*Snelheid, efficiëntie en transparantie: het Nederlandse wetgevingsproces in transitie*?", Beleid en Maatschappij 2013 (40) 4: p. 439
[246] See https://www.dnb.nl/en/supervision/innovationhub/maatwerk-voor-innovatie-regulatory-sandbox/index.jsp

federal states in the United States showed, it is incredibly difficult to set up a regulatory framework for smart contract at this moment due to the fact that the underlying technology simply changes to fast to regulate in a proper and all-encompassing way. Investigating implementation of smart contracts by using a regulatory sandbox could possibly give a good measurement instrument to see whether, and also how, smart contracts can be used in Dutch contract law, without automatically applying the current Dutch contract law framework, which can be seen as not well-suited for some aspects of smart contracts.

However, according to my view, the Dutch legislator should not create a complete new set of regulations because of the fact that the current Dutch contract law already supports the formation and enforceability of smart contracts for the greater part.[247] As the regulatory attempts of the federal states of the United States showed, it can be concluded that the implemented legislation did not make a significant contribution to the legislation at all since it created more confusion than clarity. Because of this, the American Chamber of Electronic Commerce (ACEC) also concluded that enactment of the state legislation regarding smart contracts is "*unnecessary and potentially undermines the growth of the industry*", since the ACEC believes that existing U.S. law, without further revision, supports the formation and enforceability of smart contracts under state and federal law.[248] There are, unfortunately, specific legal situations with respect to smart contracting that are not covered by Dutch contract law at this moment. For example, as previously discussed in paragraph 2.3.4, it can be said that the legislation with respect to electronic contracting does not completely match with smart contracting and its underlying technology. Without making a complete new set of regulations (that do not make a contribution to regulate smart contracts at all), some legal situations could be tested within the regulatory sandbox even before adding some additions to the already existing framework. After all, inconsistently drafted legislation will confuse the marketplace and potentially hinder innovation.[249]

3.2.1: Best practices for smart contracts

To overcome all possible eventualities regarding smart contracts, some (Dutch) scholars argue that "best practices" have to be developed besides a regulatory framework.[250] As *Tjong Tjin Tai* argues, these best practices can be found in the current framework of contract law. According to him, developing best practices might be interesting since it can be argued that contract law consists out of knowledge and experience about disputes regarding contracts and what can be seen as the best solution for these potential problems.[251] Compliance with best practice-solutions that a community of experts have certified as "best", seems a sensible way to mitigate risks of inefficiencies or mistakes within a smart contract. From this

[247] See paragraph 2.3.3 and 2.3.4 about the framework of contracting and electronic contracting within Dutch contract law.
[248] https://digitalchamber.org/policy-positions/smart-contracts/
[249] https://digitalchamber.org/policy-positions/smart-contracts/
[250] See E. Tjong Tjin Tai, "*Smart contracts en het recht*", NJB 2017/146: §
[251] E. Tjong Tjin Tai, "*Smart contracts en het recht*", NJB 2017/146: §8

perspective, it can be argued that setting up a framework of best practices within the Netherlands could be a good first step of securing some key aspects of smart contracting that dictate the recommended course of action in certain situations.

However, setting up a framework of best practices will still face problems on the long-term due to the fact that an almost similar point of consideration could be made as with setting up a regulatory framework. While previous paragraph made clear that it is difficult to regulate something which is constantly being disrupted as a result of technological changes, the same could be argued regarding best practices. Because of the fact that new technologies change rapidly, no guarantee can be made that the best practices regarding smart contracts will give a well-suited recommended course of action over the long-term. At the same time, most smart contracts nowadays still entail existing real-world equivalent contracts that need a form of legal recognition in order to prevent or solve problems. Therefore, setting up best practices for smart contracts can be considered as a necessary first step towards a broader acceptance, while also creating certainty in several situations that might be encountered with the use of smart contracts. Although it can be assumed that the underlying technology of smart contracts will evolve and might bring some new challenges regarding the recommended course of action, it seems more than useful for now to set up best practices that are focused on smart contracts and their underlying technology to create a first broad level of acceptance and legal certainty.

Nonetheless, basing best practices on *current* contract law[252] might not give the best solution for a complete new way of contracting. As smart contracts are still in its infancy, many technological changes are expected in the upcoming years which might influence the potential effect of a framework of best practices. While best practices of current contract law are a result of practical experiences and cases in the field of "traditional" contracting, it can be assumed that these experiences and cases are lacking with regard to smart contracts. The lack of experience might eventually lead to a framework of best practices that is not suitable for smart contracts or create an opposite effect. Since concluding smart contracts will bring other challenges, other ways of solving or preventing these challenges have to be presented. However, I agree with *Tjong Tjin Tai* that some best practice-based solutions have to be set up, especially since it can be considered as a first step towards a broader acceptance of smart contracting, while also representing the most efficient or prudent course of action. From another perspective, it appears clear that these best practices cannot be based on the actual contract law due to the fact that this new technology undermines the foundations of the contract law. For this reason, I believe that while best practices can be seen as an indispensable element of legal acceptance, these best practices also need to be established in a way that suits the underlying technology of smart contracts.

One aspect which also has to be taken into account with regard to best practices, is the fact that smart contracts (which are stored on a blockchain) most of the time face difficulties regarding the applicability of one specific framework of best practices. To illustrate the situation, consider public blockchains where it is possible that many participants have

[252] As Tjong Tjin Tai argues in his article.

different nationalities and, consequently, different applicable frameworks of law or best practices. In the event that best practices are based on a specific framework of contract law within a country, undesired outcomes could arise since every country has different remedies in contract law to solve potential disputes. As a result, creating a transnational approach by setting up an international framework of best practices with regard to smart contracts, could be considered as a possible solution for this problem.

Beside best practices in general, there is also one specific possible way of solving some challenges in advance and for the long-term: creating an ethical and style guideline for the *programming* of smart contracts that can be considered as a part of the best practices. This guideline should also contain best-practice based solutions for *coding* of the smart contract that have to be used in advance, like code review, testing, audits and correctness proofs. These guidelines have to be followed before even concluding and enforcing a smart contract.

3.2.2: Ethics regarding smart contracts

Smart contracts are agnostic, which means that a smart contract itself does not have the ability to make a decision about what can be considered as "good" or "bad": a smart contract simply executes what has been programmed into the code as soon as the conditions have been met. As a consequence of the automatic execution in case the conditions have been met, a smart contract is "smart" as in "robotic" or "automated". Because of the fact that all of the actions and obligations are pre-programmed within the code, it can be concluded that a smart contract has a lack of any self-thinking or pro-activity. After all, smart contracts are programmed by humans and not by an artificial intelligence application.

As soon as something goes wrong with the execution of a smart contract, the mistake has been made by a person in the stage of programming. In other words: the "bad" happens even before the smart contract is put into work. Nonetheless, mistakes are unavoidable but can be made accidentally or on purpose. As discussed in paragraph 2.2.2, the lack of recourse to legal institutions or legal principles does not only incentivize fraudsters or hackers, but also discourage the use of smart contracts. One important foundation of what can be seen as "bad" within a certain field, depends most of the time on the set of conceptions, decisions or actions that people use to express what they consider as "good" or "bad", i.e. norms and values. These conceptions or values are not only captured in several legislative or regulatory frameworks, but also in reflective guidelines of what can be seen as acceptable or not.[253] This reflection on norms and values is often described as "ethics" and can be seen as the cornerstone of defining acceptable conduct within, for example, the field of law. Professions like lawyers or notaries need to comply with professional ethical guidelines that are specifically drafted for their field of work and specify their rights and responsibilities but also indicate what they regard as responsible or proper conduct by explaining the standards and values that should guide in their behavior and decision making.

[253] E.H. Schotman, "*Ethiek en recht in kort bestek*", 2017

Therefore, it might be useful to set up ethics in the Dutch legal system with respect to the *coding* part of smart contracts. These ethics might on their turn prevent incentivizing fraudsters or hackers to misuse smart contracts. Ethics do most of the time already play an important role in setting standards about what can be considered as acceptable within the traditional process of concluding written contracts. However, most of the subjects which are in opposition to several contractual or legal ethical subjects within traditional written contracts are stopped or removed by contract lawyers, auditors or notaries (i.e. intermediaries), due to the fact that these subjects might be against the law or ethical guidelines. With smart contracting these intermediaries will be taken out of the process and will "ethical screening" need to be established by other ways. Another point worth mentioning is the fact that programmers will start playing an important role in the ethical screening. As discussed is paragraph 1.5.1, it can be assumed that some third parties will start playing a different role in the process of concluding a smart contract, compared to the traditional way of contracting. Where traditional contracts are drafted by contract lawyers, it can be reasonably assumed that smart contracts need to be drafted by programmers due to the fact that most lawyers will not know how to program an enforceable smart contract. As a consequence, the ethical guideline for smart contracts might shift from the traditional intermediaries (i.e. lawyers and notaries) to programmers.

Therefore, before even concluding a smart contract, it might be necessary that a programmer needs to comply with several specified ethical rules about the programming of a smart contract. One major advantage of using an enforceable guideline that has to be taken into account before concluding a smart contract is the fact that this has the potential of solving most of the disputes before implementing a smart contract. Smart contracts might have far-reaching consequences, especially when these smart contracts are stored on a blockchain and characterized by their nature that is hard to change afterwards. Regulatory frameworks that are used afterwards might give solutions to disputes but might also turn out to be problematic to carry out as the "bad" already took place. Nowadays, programmers already have to take several guidelines into account, like a style guide of how to code something in a proper style that is readable and maintainable. An ethical guideline regarding several ethical subjects would be useful for incentivizing the use of smart contracts, programming a smart contract or participating in a smart contract. According to my view, this ethical guideline should be included in the best practices of smart contracts, among other important set of actions like code reviewing, testing, auditing and correctness proofs.

Although ethics play an important role in defining responsible or proper conduct and specify what standards and values should guide in making decisions, it can be assumed that it will be difficult to rely on other persons' ethical decisions, especially when the conclusion of a smart contract will take place on an *anonymized* blockchain network. There is no guarantee that a random stranger will be ethical in making decisions or pursue their own self-interest detriment of others. Contracts depend on being able to trust that promises are kept and ethical decisions will be made during the whole life-cycle of a contract. In essence, trust and reputation among the contracting parties play an important role in securing the promises that

have been made. Trust or reputation can partly be guaranteed in case both parties need to be identified and have to ask for approval and authentication in a private blockchain. As a consequence, it can be said that trust and reputation are more present in these private blockchain networks compared to anonymized public blockchain networks since there is already a "screening" of the party with whom they want to conclude a smart contract. Unfortunately, just as with traditional contracting, this will still not give a complete guarantee that the other contracting party will be ethical or pursue their own self-interest. However, trust (and consequently reputation) within a blockchain network[254] could be guaranteed to the utmost extent by making use of several trust models for smart contracts, like making use of deposits, review agents or communicate about experiences with other participants (so-called "gossiping").[255] These trust models are created for penalizing "corrupt" participants and preserve participants who act in a responsible or proper manner and according to specified norms and values.

Although trust and reputation might be created on the basis of several trust models, another problem which might occur with ethics can be found in how to make sure that ethics are implemented within a smart contract. An *enforceable* style guide of what a participant might program within a smart contract can already be a first step in the removal of bad or unethical smart contracts. Furthermore, automated security tools might perhaps screen the "bad" smart contracts out of the blockchain within a couple of years. For now, the best solution seems the way of creating trust and reputation between participants of smart contracts by setting up an enforceable and automated guidebook within the programming language for setting up smart contracts.

Additionally, it can be argued that, unlike regulations, ethics most of the time not change drastically within a short period of time. The technology and programming languages will change over the years and will logically bring some new ethical challenges. However, the underlying ethics do most of the time remain the same.[256] Nevertheless, it can be argued that points of view about what can be seen as "ethical", differ among several cultures, countries, industries or technical applications and will become more apparent with supra-national smart contracts. Most of the countries or industries do already have different ethical guidelines with respect to contracting. Therefore, it might be useful for the Dutch legislator (as a first step) to set up an ethical guidebook with respect to the coding of smart contracts within Dutch contract law. Supra-national smart contracting platforms might, on the contrary, set up their own international ethical guidelines that need to be followed in order to effectuate a smart contract on their platform. After all, effectuating unethical smart contracts, which are hardly or practically impossible to change afterwards, can be seen as undesirable. Taken together, the abovementioned points towards the conclusion that a first delineation of ethics within in

[254] Both private as public blockchain networks.
[255] D. Harz, "*Trust and verifiable computation for smart contracts in permissionless blockchains*", 2017: p. 21-23
[256] Think about illegal things within setting up contracts in general. These things are most of the time based on socially accepted (or not accepted) values and standards.

the coding-process of smart contracts, can be seen as indispensable in the process of creating a first broad level of acceptance and legal certainty.

3.3: Coding existing Dutch law into smart contract applications

Although there is no Dutch legislative framework with respect to smart contracts at the moment of writing this thesis, it can be expected that in the near future regulatory additions will be implemented. Without any regulatory additions regarding new technologies, Wild West scenarios (i.e. legal vacuum) will be foreseeable. However, it can be concluded that it is incredibly difficult to set up a legal framework and programming existing legal rules, doctrines and legal interpretation in a smart contract, as the legislative attempts of some federal states within the United States already showed. Consequently, the question might arise how to tackle the problems with respect to the implementation of a Dutch legislative framework (also with potential new specific additions for smart contracts) in smart contract applications.

Some scholars[257] argue that a hybrid approach could be a (temporary) solution to overcome the first legislative problems. According to them, it can be argued that a practical approach of combining a "normal" contract with smart contracts might be a first step in preventing a legal vacuum. This "split-contract" approach, or "dual integration", tends to include a written contract on the blockchain network as an addition to the coded part of the smart contract.[258] This approach could eventually turn into the programming of existing legal rules, doctrines, precedent, and their existing legal interpretation into smart contract code. In essence, such existing rules would relate to comparable real-world transactions and could be added as smart contract parameters and require that compliance will take place with existing law and doctrines.[259]

The aforementioned hybrid approach has several benefits. Firstly, it provides the contracting parties with a very high degree of regulatory certainty within a short period of time.[260] Regulatory certainty will on its turn increase and incentivize the commerce in the use of smart contracts. Secondly, it can be reasonably expected that regulators will lower their cost of supervision and enforcement in case existing legal rules and doctrines are implemented in these self-enforcing smart contracts, due to the fact that smart contracts cannot execute unless all regulatory conditions and parameters are fully complied with.[261]

However, according to *Kaal*[262] this hybrid approach will only give benefits within the first period to overcome a complete legal vacuum without any degree of regulatory

[257] See for example M. van Eersel & T. van den Bergh, *"Blockchain en smart contracts: toegang tot een reeks van slimme dingen"*, FRP 2017/457 (afl. 4): p. 45 and also H. Schuringa, "*Enkele civielrechtelijke aspecten van blockchain*", Computerrecht 2017/254: § 3.1

[258] J.B. Schmaal & E.M. van Genuchten, "*Smart contracts en de Haviltex-norm*", Tijdschrift voor Internetrecht, nr.1 maart 2017: p. 14

[259] W.A. Kaal and C. Calcaterra, "*Crypto Transaction Dispute Resolution*", The Business Lawyer Spring 2018: p. 44

[260] W.A. Kaal and C. Calcaterra, "*Crypto Transaction Dispute Resolution*", The Business Lawyer Spring 2018: p. 44

[261] W.A. Kaal and C. Calcaterra, "*Crypto Transaction Dispute Resolution*", The Business Lawyer Spring 2018: p. 44

[262] W.A. Kaal and C. Calcaterra, "*Crypto Transaction Dispute Resolution*", The Business Lawyer Spring 2018: p. 44-45

certainty.[263] On the long term, it can be assumed that the hybrid approach will not be sufficient. *Kaal* assumes that in the event smart contracting evolves over time, fewer smart contracts will have a real-world equivalent. In fact, it may over time become increasingly difficult to find real world legal solutions that can become effective parameters for smart contracts.[264] Nonetheless, in contrast to Kaal's assumption, it can be argued that at the moment of writing this thesis smart contracts still entail existing contracts that have real-world equivalents and solutions that are still effective parameters for smart contracts[265].

Another argument made by *Kaal* is that it will become increasingly difficult to find real world legal solutions for smart contracts in case smart contracts will not be effectively encouraged to stick to traditional legal standards and translate legal rules, doctrines and intentions of contracting parties into the programming language of smart contracts.[266] Also, the intentions of the legislator, court, or administrative agency in passing an applicable rule, are almost impossible to translate into binary programming language.[267] However, the problem of not being capable of translating intentions of legislator, court or administrative agency into programming language contribute in favor to the arguments of using a hybrid approach, according to my view. As the hybrid approach makes clear, smart contracts originate as documents written in natural language that require subsequent translation into code. Natural language does sometimes require some ambiguity or open norms to allow some flexibility within written contracts. Therefore, translating natural and ambiguous language into binary programming language can be seen as problematic. The difficulty of this process is generally underestimated, especially with more complex contracts.

3.3.1: Difficulties in translating process

Despite the promising progress in the areas of machine learning and natural language processing, it is still impossible to translate natural language into code without making a significant compromise in the quality of the output of such conversion.[268] Approximations seem permissible in automated translations between natural languages, where the overall meaning of a sentence can be gleaned from the context. However, this will become difficult when it comes to legal provisions that are drafted with precision and where one single word may give rise to unintended consequences and in the worst case, a dispute.[269] The more complex a contract will get, the more difficult it will be (if not impossible) to put the agreements in code. Another problem is the fact that the translation of natural language into

[263] This is also acknowledged by *Van Eersel* and *Van der Bergh:* M. van Eersel & T. van den Bergh, *"Blockchain en smart contracts: toegang tot een reeks van slimme dingen"*, FRP 2017/457 (afl. 4): p. 45. They argue that for the first period, a hybrid approach will give the possibilities for smart contracts and blockchain technology to be used optimally.
[264] W.A. Kaal and C. Calcaterra, "*Crypto Transaction Dispute Resolution*", The Business Lawyer Spring 2018: p. 45
[265] And perhaps for a long period after writing this thesis.
[266] W.A. Kaal and C. Calcaterra, "*Crypto Transaction Dispute Resolution*", The Business Lawyer Spring 2018: p. 44-45
[267] W.A. Kaal and C. Calcaterra, "*Crypto Transaction Dispute Resolution*", The Business Lawyer Spring 2018: p. 44-45
[268] E. Mik, "*Smart Contracts: Terminology, Technical Limitations and Real World Complexity*", 2017: p. 16. Referring to H. Surden, '*Machine Learning and Law*' (2014) 89 *Washington Law Review* 87, 91-95.
[269] E. Mik, "*Smart Contracts: Terminology, Technical Limitations and Real World Complexity*", 2017: p. 17

code does not constitute a straightforward process of converting legal language into computer-readable instructions but requires the prior interpretation of the legal language before even converting it.[270] Almost every contract needs some form of interpretation, to write down the agreements of contracting parties in a correct and dispute-preventive way. Moreover, the intention and interpretation of a legal rule needs to be clear even before programming and 'translating' a smart contract. Since most of the contracting parties nowadays do not know how to code, it can reasonably be expected that this responsibility will rest on the shoulders of the programmer. Furthermore, it can be assumed that these programmers (or developers) lack the needed knowledge of translating natural written language into programming language without fully complying with the applicable legal rules and the principles governing contractual interpretation. Moreover, it must not be forgotten that the process of interpretation is not limited to situations when words are ambiguous because it may not immediately be apparent that a particular word or expression is capable of multiple interpretations. In the context of smart contracts, the problem would not lie in the parties disagreeing over the meaning of words but in the possibility of incorrect interpretations by those who decide how to convert a particular obligation into code.[271] As a consequence, most contracts contain gaps and require that terms be implied to make the agreement workable in practice. However, gaps within a *smart* contract are not practical or desirable due to the binary design.

One possible solution could be setting up a system where contract lawyers and programmers would work together and translate the contractual language into programming language. However, this still requires knowledge on both sides of contractual language and programming language. Most of the traditional written contracts are not drafted from scratch, but drafted according to several contract models, templates and previous experiences of the contracting parties with regard to the drafting of some specific contracts. Most of the time do contracting parties also need some assistance of persons or entities with specific knowledge about certain aspects. Unfortunately for now, this specific knowledge and experience about smart contracting is simply not available to the contracting parties. Therefore, it can be assumed that neither one of the parties (i.e. lawyers or programmers) possesses all the required knowledge and will not be able to ascertain whether the code of the smart contract correctly reflects the originating legal document. As a consequence, this might eventually lead to unwanted outcomes.

3.3.2: Direct coding for smart contracts

Another solution might be writing smart contracts in programming language from the outset, to prevent problems with translating natural language into programming language.[272] In this sense, smart contracts could in fact reduce ambiguity or gaps due to the fact that these

[270] E. Mik, "*Smart Contracts: Terminology, Technical Limitations and Real World Complexity*", 2017: p. 17
[271] E. Mik, "*Smart Contracts: Terminology, Technical Limitations and Real World Complexity*", 2017: p. 18
[272] E. Mik, "*Smart Contracts: Terminology, Technical Limitations and Real World Complexity*", 2017: p. 18-19

contracts need to be capable of a single interpretation. Since most lawyers are unlikely to become programmers and vice versa, it is suggested by some scholars that even if smart contracts cannot be coded directly, contracts could be drafted with encoding in mind.[273] [274] To this end, the contracting parties (and their lawyers) could reorient the manner in which they express their agreement to facilitate its subsequent translation into code: in other words, describing their obligations in a formalized, structured manner and provide objective, measurable criteria that must be met for such obligation to be considered performed.[275] By contracting in such way will a smart contract know how to perform from the outset without the danger of contractual or interpretation mistakes.

However, the drafting of a smart contract does still need to be created by lawyers and programmers together. This in turn requires that those two groups need to be able to communicate via a common language. Smart contracts should thus be written by lawyers in a legal language that is logical, clear, unambiguous, and comprehensible to programmers and vice versa.[276] For simple commercial contracts, like the selling of cryptocurrencies, it can be argued according to my view that this will cause no significant problems. However, for more complex contracts will the abovementioned direct coding through a common language be more complex. This will be caused due to the fact that complex contracts most of the time need some ambiguity in their writing. Ambiguity creates workability as it allows the parties to retain a measure of flexibility in performing their side of the bargain and in evaluating each other's performances. Moreover, it enables both parties to adapt to changing circumstances without having to redraft the agreement.[277] While ambiguity can be seen as a feature of a written agreement, it can be said that ambiguity most of the time will be seen as an error or bug in the programming language of a smart contract due to its binary design: the nature of a smart contract will not allow any ambiguity at all. Some ambiguity or uncertainty within a contract is most of the time anticipated and tolerated, since some circumstances can be seen as unforeseen or practically impossible to think about.[278] For the performance of certain obligations is some form of flexibility or ambiguity needed. According to *Mik,* coding smart contracts for complex and uncertain environments will be extremely difficult due to the number of things that can go wrong and the inability to predict how the evolving commercial context will affect the contractual relationship.[279] It must also be assumed that a completely unambiguous contract would, inevitably, be extremely long due to the fact that many of its obligations would have to be described with a large number of eventualities in mind. As paragraph 1.5.4 already described, smart contracts on a blockchain have a limited space within a block. Consequently, it can be seen as an unworkable situation in case extremely long contracts have to be implemented in a blockchain with limited space.

[273] E. Mik, "*Smart Contracts: Terminology, Technical Limitations and Real World Complexity*", 2017: p. 18-19
[274] Or not even needing a lawyer at all.
[275] H. Surden, "*Computable Contracts*" (2012) 46 *UC Davis Law Review* 635, 674-675
[276] F. Al Khalil, et al., '*A Solution for the Problems of Translation and Transparency in Smart Contracts*' (2017) 3, 8, 9
[277] E. Mik, "*Smart Contracts: Terminology, Technical Limitations and Real World Complexity*", 2017: p. 19
[278] Take for example some "force majeure" events.
[279] E. Mik, "*Smart Contracts: Terminology, Technical Limitations and Real World Complexity*", 2017: p. 20

Another problem is the fact that not every contractual obligation or legal concept can be translated into exact and unambiguous coding. Some open legal concepts in Dutch contract law, like "reasonableness and fairness" ("redelijkheid en billijkheid"), are impossible to translate into binary programming language. Many contractual obligations (like reasonableness and fairness for instance) are based on reasonable care, where the parties must undertake or refrain from certain actions without having to produce a measurable outcome. As a result, it may be difficult to reduce these provisions to sequences of steps and to provide objective benchmarks against which they can be evaluated by a self-enforcing smart contract.[280] Therefore, it might lead to the (perhaps unsatisfying) conclusion that not all obligations can be expressed in code and, as a consequence, not all contracts can be "smart" with the use of direct coding.

3.3.3: Dual integration of smart contracts

As previous sub-section explained, some elements cannot be implemented within programming language. This led to the conclusion that not all contracts can be smart contracts. However, some simple questions can arise: can normal written contracts really implement and fully tackle all of the problems with regard to open legal concepts like reasonableness and fairness? Does ambiguity really add flexibility in all cases or is it incapability of the contracting parties to write down unambiguous terms? Are the same problems with respect to coding of complex contracts not also encountered with implementing them in traditional written contracts?

Smart contracts are still facing many issues with respect to implementing complex contractual obligations and replacing highly-sophisticated traditional contracts. However, the same issues are faced nowadays by traditional paper-based contracts. Cutting of smart contracts on the basis of "not being able to express all obligations into code" might therefore be too short-sighted, according to my view. Open norms like reasonableness and fairness are most of the time used to settle a dispute *afterwards* by a court, while coding and implementing the intentions within a smart contract has to be done in advance. In short, written contracts are also unable to fully eliminate all gaps or potential disputes.

According to my view, the real underlying problem might actually be that legal systems are unlikely to resolve disputes stemming from smart contracts solely on the basis of their code. As written agreements also leave some gaps and ambiguity to settle possible disputes afterwards, the same should also be done with smart contracts. However, it can be assumed that it is much easier for courts to understand the underlying intentions and problems when they have a written agreement in front of them instead of a coding part of a smart contract. Furthermore, courts might want to apply the defaults of settling a dispute for similar prose agreements, which might lead to an outcome that may not reflect the intentions of the smart contracting parties. Because of the risks involved in enforcing smart contracts by code alone, it might nonetheless be useful to utilize "dual integration" of some kind. As discussed before,

[280] E. Mik, "*Smart Contracts: Terminology, Technical Limitations and Real World Complexity*", 2017: p. 21

dual integration means that a smart contract will be linked to a document in natural language that can be enforced by a court.

Paragraph 3.3 discussed the argument of *Kaal,* which said that the hybrid approach (i.e. dual integration) will only give benefits within the first short-term period. Although this theorem *might* be true, this theorem is based on the *assumption* that the hybrid approach will not be sufficient as smart contracting evolves over time and that fewer smart contracts will have a real-world equivalent. For now (and perhaps in a very long period), the technology and smart contracts are not that evolved that they do not have real-world equivalents. They simply represent already existing contracts. In sum, the created expectations of smart contracts are running ahead of what the current technology really can deliver.

Another reason which might be causing the disagreements about using dual integration or not, can be found in the difference of terminology and use of smart contracts. Dual integration might not be very useful in the case where smart contracts are used and seen as "smart contract *code*". However, when smart contracts are used as commercial contracts and can be seen as "legal binding contracts", it can be said that dual integration might be useful to express all of the clauses and open norms that protect parties from various edge-case liabilities, which are not always suitable for representation and execution through code. As paper-based written contracts are also not capable of tackling all of the challenges with respect to these legal norms, it can be assumed that smart contract will be neither capable in doing so (for now).

Therefore, a system of dual integration might be most suitable for legally binding contracts within the Netherlands, due to the fact that it is not possible to translate all of the natural language into programming language and setting up a system of so-called "direct coding" is not practical. As already discussed in paragraph 2.4, several clauses and information within a legal binding contract in the Netherlands play an important role in the interpretation of (smart) contracts and settling disputes. In case of a dispute, the document that is linked to the smart contract will play a decisive role in settling the dispute, while the advantages of smart contracts can also be utilized. As mentioned before, the real underlying problem might be the fact that the legal systems are unlikely to resolve disputes stemming from smart contracts solely on the basis of their code. Therefore, the solution might be found in dual integration since it allows the courts to understand smart contracts in a more comprehensible way. Furthermore, this dual integration can also be seen as a first step in solving the prisoner's dilemma, which is faced nowadays with the use of smart contract-based contracting. After all, dual integration might provide the contracting parties with a very high degree of regulatory certainty within a short period of time and also increase and incentivize the commerce in the use of smart contracts.

3.4: Judicial system in the Netherlands

It can be assumed that smart contracts could have an enormous legal impact in the Netherlands. As chapter 2 of this thesis made clear, there are no real fundamental legal obstacles with respect to smart contracts as legally binding contracts within the system of

Dutch contract law. The previous sub-sections of this chapter also made clear that several aspects have to be done regarding the legislative structure of the Netherlands to support the implementation of smart contracts on a larger base. Beside these legal aspects, it can be reasonably argued that a well-suited dispute resolution needs to be set up to create a solid base for smart contracting. Consequently, courts within the Netherlands need to be able to solve disputes regarding smart contracts, to eventually incentivizing the use of smart contracts, programming a smart contract or participating in a smart contract. After all, a lack of conflict resolution mechanisms would undermine and hinder its broadening evolution and applicability.

Before discussing possible changes within the Dutch judicial system, it is important to mention several obstacles which can be faced with blockchain-based smart contracts. As mentioned before, participants of smart contracts remain most of the time anonymous, especially in case of public blockchains. As a result of this anonymity, national courts cannot have jurisdiction over blockchain-based smart contracts since it is unlikely that a court could find out who transacted or set up the smart contract via the anonymized (public) blockchain. Settling a dispute by a court without identifying the contracting parties will almost be impossible. Another question that might arise, is which jurisdiction (i.e. which court) will be competent to rule over the smart contract. Since the blockchain network is a decentralized online network, many (anonymous) participants all around the world might be involved with the smart contract. This question will most of the time arise in the event that no choice of jurisdiction has been made in the smart contract in advance, or whether the blockchain network is an open and public (anonymized) network. These challenges require a different approach of dispute resolution, which will be discussed in paragraph 3.4.2. Before setting up a dispute resolution for these cases, some changes in the existing judicial system will be discussed in case of a smart contract where a Dutch court is competent to rule over the smart contract, and where the contracting parties are known (i.e. not anonymous)[281].

3.4.1: Changes with regard to existing judicial system

Contracting parties have a right to go to a Dutch court for starting a judicial procedure in the event that contracting parties made clear in their smart contract that a court within the Netherlands will be competent to rule about the dispute. Normally, the Dutch Court of Civil Law Cases will be competent to rule about a dispute regarding smart contracts.[282] However, the question will arise whether these civil judges have enough knowledge and competence to fully understand the content of a smart contract (i.e. the code of a smart contract) due to the fact that the programming language of a smart contract can be seen as substantially different compared to natural language. Although dual integration[283] might help judges to understand

[281] This might for example be the case with private blockchain-based smart contracts.
[282] Disputes concerning rent, hire purchase, employment contracts and monetary claims up to € 25,000 are dealt with by the subdistrict court ("Kantonrechter").
[283] See paragraph 3.3.3

smart contracts in a better way, other solutions might also come into play to overcome the lack of knowledge and competence.

As already discussed in paragraph 2.4, a first possible solution to explain the content of smart contracts can be found in the use of experts on the field of smart contracting. According to article 152 (and paragraph 6 of Ninth Section) of the Dutch Code of Civil Procedure, evidence can be delivered by all means. Consequently, this expert might give the judge some extra information and "translate" the programming language into language that is understandable for a judge and point out several features, bugs or errors and make it possible for a judge to give a ruling regarding a dispute. Another change which might be useful, is developing a standard where certain 'fixed' codes in a smart contract have a 'fixed' meaning and that for these codes the judge will explain them in only one way. This "standard" will create more certainty with regard to the settling of disputes, which eventually might incentivize the use of smart contracts.

However, the aforementioned standard and use of experts might eventually lead to the situation where very specific legal norms and doctrines will be developed that differ in many ways from traditional contracting. This is partly caused by the different nature and underlying technology (i.e. use of blockchain) of smart contracts. As a result of this difference in nature and technology, it can be argued that a specific "Chamber" within the Dutch judicial system can be developed which is specialized on settling disputes regarding smart contracts and blockchain technology. Since the way of contracting is different, other ways of settling disputes have to be established. This specialized chamber could, according to my view, be effectuated to the example of the Enterprise Division ("Ondernemingskamer") of the Amsterdam Court of Appeal.

The Enterprise Division is a specialized Dutch judicial body which has the task to assess (legal) disputes of Dutch *companies* and make binding verdicts about these disputes. The main advantages of a such specialized judicial body is the speed of which verdicts are concluded and measures are taken. In addition, the Enterprise Division has the special ability to implement so-called "immediate provisions"[284] at every stage of the dispute.[285] Moreover, another characteristic of the Enterprise Division is that beside three judges, also two councils are added who have a broad knowledge and experience in the business practice, as for instance (former) director, supervisory director or accountant. This background contributes to the depth of the treatment at the hearing and the persuasiveness of the judgment, certainly in case of procedures that are conducted in (relatively) unexplored areas.

According to my view, it can be argued that setting up a similar specialized chamber for blockchain/smart contracts within the Dutch judicial body would benefit the dispute resolution with regard to smart contracts. The two councils, who for instance are specialized on the field of blockchain and/or smart contracting, could give an enormous contribution to the depth and persuasiveness of the eventual judgment. As smart contracting is an innovative and new way of contracting, new ways of settling disputes have to be introduced. Settling a

[284] Dutch: "onmiddelijke voorzieningen".
[285] See article 2:349a section 2 BW

dispute with only the knowledge of judges will not be enough with hard to fathom technologies and programming language. Taken from this perspective, the new judicial body regarding smart contracts could also be equipped with comparable "immediate provisions" just like the Enterprise Division, like using a kill switch[286] or the ability to require the parties to create a new transaction[287] which reverses undesirable outcomes of the coded and executed transaction that was disputed. During the treatment of a dispute could the specialized body also use the previously discussed translation module and revised interpretation norm.[288]

Everything taken together, it can reasonably be argued that the abovementioned specialized body including the immediate provisions, specialized councils, revised interpretation norm and translation module, contribute to a well-suited and fast dispute resolution with regard to smart contracts. Especially in cases where a Dutch court within the Netherlands will be competent to rule about the dispute regarding a smart contract.

3.4.2: Distributed jurisdiction

As already shortly mentioned in the beginning of this paragraph, some problems might be encountered with respect to the dispute resolution of several types of smart contracts. Where disputes about smart contracts that are used as "real" contracts could be solved with the discussed solution in paragraph 3.4.1, it can be assumed that anonymized and public smart contracts will face difficulties. Since the blockchain network is a decentralized online network, many (anonymous) participants all around the world might be involved with a specific smart contract. In some of these cases will national courts have no jurisdiction over blockchain-based smart contracts due to the fact that it is unlikely that a court can find out who transacted or set up the smart contract via the anonymized (public) blockchain. Without identifying the contracting parties, it can be said that it will be practically impossible to settle a dispute by a court or decide which law (i.e. jurisdiction) is applicable.

However, a lack of governance and conflict resolution mechanisms would undermine the democratized trust created by blockchain technology and hinder the broadening evolution and applicability of smart contracts in general, especially with public smart contracts.[289] For these types of smart contracts, regulatory alternatives might be necessary due to the impossible situation of consistently identifying the parties in any dispute and the associated problems of applying the existing legal infrastructure.[290] With the anonymity of contracting parties in smart contracts, it becomes unclear which specific laws or rules could be coded into the smart contract and who the contracting parties are. These smart contracts become supra-national and create additional burdens and strengthen legal uncertainty of existing legal rules.

[286] See also paragraph 2.5.1
[287] See also paragraph 2.5.1
[288] See also paragraph 2.4.2
[289] For example: cryptocurrency transactions or a service contract involving services pertaining to cyberspace, such as programming services to create a given webpage.
[290] W.A. Kaal, C. Calcaterra, "*Smart Contract Dispute Resolution – The Need for an Open Source Blockchain Platform Ecosystem*", June 2017: https://medium.com/@wulfkaal/smart-contract-dispute-resolution-the-need-for-an-open-source-blockchain-platform-ecosystem-e6318610fdef

The problems which might be encountered in the near future, are best described by *Wulf & Calcaterra*[291]*: "Finally, as Ethereum and other blockchain-based smart contracting networks become increasingly autonomous, legislators simply will not have the ability and authority to dictate to contracting parties and Ethereum itself whether or not to code existing legal rules into a given smart contract. Anonymity means the government cannot identify citizens with access to a free internet who use the blockchain. Nor can the government prevent citizens from adding contracts anonymously to the blockchain. Autonomy means the government cannot stop the blockchain miners from automatically adding contracts because the protocols do not recognize distinctions between good and bad contracts that any outside body dictates. Also, the distributed nature of the blockchain means the government cannot control the nodes which maintain the blockchain without an effective world-wide governing body—which is obviously nowhere near feasible at this point in history—not even if the nodes were known to the government, which they are not."*

The abovementioned quote mentions several problems with respect to dispute resolution regarding smart contracts. With anonymized blockchains, it becomes practically impossible to address which jurisdiction is applicable due to the fact that many nationalities and regulations might be involved. To illustrate the impractical situation, consider the DAO-hack already discussed in paragraph 1.5.2, where thousands of participants around the world were damaged as a result of the hack. In such scenario, if we follow the letter of the international private law[292], there are several courts that have jurisdiction both nationally and internationally. When no choice of jurisdiction has been made before effectuating the smart contract, courts might claim jurisdiction on the basis of residency of the contracting parties.[293] When contracting parties have their residency in different countries, most of the time a court within the jurisdiction of the defendant is going to be able to claim jurisdiction ("forum rei").[294] However, this will raise several questions. First of all, who will the contracting parties and courts point out as defendant and hold liable for the damages? It can be seen as practically impossible for a court or contracting party to find the source of the hack and point out a specific person or entity within an anonymized blockchain that has to pay the damages. Moreover, courts also lack the ability to stop or change the smart contract within a public and anonymized blockchain due to its unchangeable character. Secondly, when no defendant can be pointed out, several courts might claim jurisdiction on the basis of the residency of the plaintiff.[295] This situation will also be undesirable and not practical at all, due to the fact that several different courts around the world will give a verdict about the damages and try to get the damages paid or turn back the unwanted situation. It can be assumed that this will lead

[291] W.A. Kaal and C. Calcaterra, "*Crypto Transaction Dispute Resolution*", The Business Lawyer Spring 2018: p. 46

[292] For European countries could the Directive 44/2001 play a decisive role is deciding which court could claim jurisdiction.

[293] L. Strikwerda, "*Inleiding tot het Nederlandse Internationaal Privaatrecht*", 2015: p. 223. See also article 4-6 of EEX-Directive II with respect to claiming jurisdiction for contracting parties within the EU. Claiming jurisdiction on the basis of forum rei, can be seen as leading principle.

[294] L. Strikwerda, "*Inleiding tot het Nederlandse Internationaal Privaatrecht*", 2015: p. 223

[295] L. Strikwerda, "*Inleiding tot het Nederlandse Internationaal Privaatrecht*", 2015: p. 224, 250-260. Especially when the contracts contain consumer related subjects.

to impractical situations when several different courts from around the world try to stop or turn back the unwanted situation. In short, no specific national court will be capable of claiming jurisdiction because of the enormous number of participants around the world, anonymous parties and absence of real power to stop or change the smart contracts within the blockchain network. These smart contracts on the blockchain have so-called "distributed jurisdiction".[296] Since in these cases no specific jurisdiction can be assigned from a practical and legislative point of view, other dispute resolutions have to be implemented.

One possible solution is an open source platform ecosystem of smart contracting dispute resolution that allows participants to opt into the conflict resolution mechanisms.[297] By doing so, the platform itself settles the possible disputes without harming the anonymity or rights of the contracting parties, since no jurisdiction needs to be chosen. It might turn out as a negative impact in case a specific jurisdiction is applicable, while one of the contracting parties did not choose or expected that the specific jurisdiction was applicable. When a blockchain network itself has a way of settling dispute through an open source platform ecosystem, it can be said that the contracting parties have to agree with this way of settling disputes without making a decision about specific national jurisdiction. As a result, the distributed jurisdiction is covered by the open source platform ecosystem itself, where the same rules apply to all of the participants around the world. From a practical point of view, this leads to more certainty and efficiency, if only for the fact that not all of the involved jurisdictions have to decide which jurisdiction is applicable.

Although the Netherlands itself cannot enforce such open source platform ecosystem on its own, some initiatives could be started by experimenting with such platform ecosystems and using regulatory sandboxes. Some initiatives have already been started by several applications to create such platform by companies themselves.[298] However, supra-national initiatives have to be examined on a large scale to overcome the lack of dispute resolution with regard to public and anonymized smart contracts. The European Securities and Markets Authorities (ESMA) held a first analysis with respect to the use of smart contracts in the security markets regulations.[299] In their first step towards an analysis, they concluded that the current regulatory framework will not be prohibitive in the short term for technologies like blockchain and smart contracts, but at the same time does ESMA acknowledge that it is not clear how several regulatory aspects can be solved.[300] One of these problems is the applicable jurisdiction. Projecting the findings of ESMA on a larger scale (i.e. not only financial instruments), it can be said that some international initiatives have to be set up with respect

[296] Term is used from the article of W.A. Kaal and C. Calcaterra, "*Crypto Transaction Dispute Resolution*", The Business Lawyer Spring 2018

[297] W.A. Kaal, C. Calcaterra, "*Smart Contract Dispute Resolution – The Need for an Open Source Blockchain Platform Ecosystem*", June 2017

[298] See for example Aragon or Semada Platform.

[299] European Securities and Markets Authority, "*The Distributed Ledger Technology Applied to Securities Markets*", February 2017

[300] European Securities and Markets Authority, "*The Distributed Ledger Technology Applied to Securities Markets*", February 2017: p. 15-18

to the question regarding distributed jurisdiction. After all, a second "The DAO-hack" is definitely not desirable.

3.5: Long term effect on Dutch trust market and corresponding legislation

Beside the abovementioned long-term effects regarding the use of smart contracts, another long-term effect that was addressed by the Dutch Blockchain Coalition in its report[301] is related to the potential effects on the recording of contract party-related information in contracts and the concerned potential disruption of the associated "trust market". In short, the current Dutch law and regulations contain many obligations regarding the recording of information of entities or contracting parties themselves.[302] Many other regulations do also have an aim of increasing trust in these recordings.[303] Some examples of such obligations are the accounting obligation for legal entities on the basis of article 2:10 BW or the obligation to register within the Chamber of Commerce on the basis of the so-called "*Handelsregisterwet*"[304]. The fact that these recordings are reported by an authority that is established by the Government does contribute to the creation of trust. Furthermore, the fact that every year an annual report has to be set up by a commercial entity that will be audited by an accountant who is licensed, creates even more trust. Some of these provisions or agreements regarding the recording of information are most of the time implemented within a contract itself. For example, some supply-agreements contain the obligation for the supplier to record the way of sending all the invoices. To create trust in the way of recording these invoices, an independent third party will serve as audit to check the way of recording.[305] Beside audits do also other parties play an important role in creating trust, like accountants, notaries and organizations that control several central registers (i.e. "trust market"). As a result, these intermediaries add trust and human judgement to a contract.

Smart contracts and blockchain technology can also record information and have, in theory, the potential to remove the intermediaries out of the process and create enough trust and security themselves. However, for creating the same level of security and trust with regard to the aforementioned recording obligations, more is needed than just the immutable and self-enforcing character of blockchain and smart contracts. Although some projects[306] already have been started up to reach a better level of security and trust regarding smart contracts, an equivalent level of trust and security will be reached as soon as smart contracts will be able to take into account compliance with existing (and ever growing) regulations[307]. A recent study of *Al Khalil*[308] showed that for this type of compliance, it will be needed to develop a "common language" between legislator, lawyers and developers of smart contracts.

[301] Dutch Blockchain Coalition, "*Smart contracts as a specific application of blockchain technology*", 2017: p. 43
[302] For clarification, privacy related data (i.e. GDPR) will not be discussed in this thesis.
[303] Dutch Blockchain Coalition, "*Smart contracts as a specific application of blockchain technology*", 2017: p. 43
[304] Translated "Commercial Registers Act".
[305] Dutch Blockchain Coalition, "*Smart contracts as a specific application of blockchain technology*", 2017: p. 43
[306] For smart contract audits see *Hacken*: https://hub.hacken.io/smart-contracts-audit
[307] After all, smart contracts cannot exist in a legal vacuum.
[308] Al Khalil et al., "*Trust is Smart Contracts is a Process, As Well*", University College Cork, April 2017: p. 8. Al Khalil did also investigate deeper into the legislation regarding financial instruments.

As *Al Khalil* argues, the legislators and lawyers author and use contracts written in that language, while the developer uses it as a specification guiding the implementation. Unfortunately, such common language is not available at the moment of writing this thesis and will even face the challenge whether it will be possible at all to translate such language into executable code with complete compliance, according to *Al Khalil*.[309] Moreover, the DBC also addressed the aforementioned problems but does (unfortunately) not go deeper into the answer and mentions only that *"it will undoubtedly have far-reaching consequences for laws and regulations, contracts and players in the "confidence market"* (i.e. trust market)[310], in case blockchain technology and smart contracts will deliver their presented promises.

Although technology led to the development of tools and abstractions to help developers in programming smart contracts, it can be said that it is still not possible to comply with existing (and ever growing) regulations that are needed for creating security and trust without the use of intermediaries. Beside the fact that smart contracts are still dependent on third parties and their input for compliance, several changes or adaptations that probably need to be done by these parties are practically impossible. Smart contracts cannot be upgraded or patched once deployed, due to their immutable nature.

According to my view, it can be concluded for now that smart contracts are not capable of recording information and creating enough trust regarding the abovementioned recordings without the use of intermediaries. The assumption that intermediaries are fallible and not trustworthy, while computers on the other hand are trustworthy and infallible, is way ahead of the reality. In more complex contracts is some form of human judgement needed, especially when smart contracts are not capable of coding complex terms, complying with existing regulations and add independent considerations. Consequently, the potential long-term effects mentioned by the DBC can be seen as unrealistic.

[309] Al Khalil et al., "*Trust is Smart Contracts is a Process, As Well*", University College Cork, April 2017: p. 8

[310] Dutch Blockchain Coalition, "*Smart contracts as a specific application of blockchain technology*", 2017: p. 43. I believe that the choice of "confidence" is not a representative translation. I prefer using "trust".

Concluding observations

This thesis analyzed the question whether smart contracts will replace traditional ways of contracting and also whether the Netherlands (and its legal framework) will be ready for the implementation of smart contracts in the future. As mentioned earlier in this thesis, some consider blockchain technology and smart contracts as the biggest and most disruptive innovations since the introduction of the Internet, while others argue that it can be considered as a "technological bubble" which will burst within a short period of time. Although it can be said that smart contracts show a potential of changing the current legal landscape, it can also be said that many legal questions arise with respect to this new way of contracting. Consequently, the central purpose of this thesis was to investigate deeper into the legal questions that arise with regard to smart contracts in Dutch contract law and what should be done regarding the legal framework to successfully implement smart contracts in the Dutch legal system.

Originally, smart contracts were contemplated within a limited and niche range of transactions, such as in the area of financial instruments. Progressively, however, the surrounding narrative within technical writings has become broader, implying that *all* contracts can be made "smart" or that many different obligations can be enforced by code. It is often assumed that because a particular technology is innovative or revolutionary (or to use the most overused term: "*disruptive*"), it is also commercially useful or capable of solving actual legal problems. With regard to smart contracts, it is often argued that the current legal problems of traditional ways of contracting will be solved due to the *automated* and *immutable* nature of smart contracts. However, this thesis showed that there are still several limitations and points of reconsideration regarding smart contracts in general and their potential of replacing traditional ways of contracting.

As it can be concluded from this thesis, several sets of problems can be uncovered regarding smart contracts. Firstly, the DAO-hack made clear that the *immutable* character of a smart contract can be seen as a "myth" and that smart contracts still entail many technical limitations which make it practically impossible to implement all of the current contractual and legal provisions within the programming language of a smart contract. Secondly, while it is often argued that the *law* itself holds back the implementation of smart contracts, it can be concluded from this thesis that the *technical* limitations of smart contracts and their underlying technology hold back implementation on a large scale. Moreover, relatively few fundamental *legal* obstacles can be detected within Dutch contract law for the use of smart contracts since Dutch contract law already supports the formation and enforceability of smart contracts for the greater part. As a consequence, the real fundamental obstacles of significantly replacing some traditional ways of contracting can be found in the *technical* limitations of smart contracts. One possible reason for this can be found in the fact that the created expectations of blockchain and smart contracts in technical writings as "legal problems solvers", are running ahead of what the current technology really can deliver.

Nonetheless, it can be reasonably expected that in the near future simple written contracts might be replaced by smart contracts, as already is happening at the moment of writing this thesis. On the other hand, more complex contracts or agreements might face fundamental challenges with respect to binary and rigid smart contracts. While simple contracts can be deduced to obligations in a formalized, structured manner and provide objective, measurable criteria that can be put into programming language, it can be argued that complex contracts entail underlying legal and social norms or practices that cannot be simply converted into binary language (i.e. code). Technical writings about smart contracts often ignore the fact that law and contracts are more than just "words on paper". Beside the challenges of translating existing (Dutch) law into smart contract applications, it can also be concluded that smart contracts remain dependent of information outside of the blockchain (most of the time provided by "oracles"), as soon as the contracts get more complex. Therefore, criticists often argue that smart contracts are not that "smart" but just robotic sets of agreements and will not be capable of solving complex problems themselves. However, smart contracts must simply be seen as programs operating in distributed computing environments and not as a *semi-mythical technology* that will solve all of the legal problems that are faced nowadays by itself. In short, some realistic way of thinking might be in place instead of proclaiming ideologically charged arguments.

Furthermore, no legal recognition has been made in Dutch law about smart contracts at the moment of writing this thesis. Since no fundamental legal obstacles can be found, simple adaptations can be made within Dutch contract law as a first step of securing some key aspects of smart contracting. Smart contracts still entail many advantages (for example security, automation and efficiency) but need to be applied into a legal system where there is room for human intervention possibilities (The DAO-hack can be taken as example for this). As a result, the Netherlands could make a first good contribution to a broader acceptance and implementation of smart contracts in case some regulatory adaptations are made. After all, *legal recognition* can be seen as an indispensable step for smart contracts to see whether they will replace some traditional ways of contracting. While Dutch contract law will not stand in the way of putting agreements in a smart contract, several additional requirements need to be set up to make it possible to conclude smart contracts in a secure and legally binding way. As discussed in this thesis, the additional requirements that are used for electronic contracting could serve as a starting point to create more legal certainty regarding the use of smart contracts. Also, revising interpretation norms in case of disputes that take the underlying technology of smart contracts into account can contribute to a broader acceptance of smart contracts, while setting up best practices, ethical guidelines and the use of "dual integration" might create even more legal certainty and acceptance of smart contracts. In sum, the Dutch legal framework will be ready for the implementation and recognition of smart contracts in case several minor adaptations will be made that respect the nature of smart contracts and their underlying technology.

To conclude this thesis, it can be said that smart contracts will not replace *all* ways of traditional contracting and also not solve *all* the legal problems due to their immutable and automated character, as is argued many times in technical writings. The fact that a new technology might be *disruptive* does not automatically mean that it will solve all of the problems. On the other hand, it can also be argued that some ways of contracting do not face significant problems and will not face improvements with the use of smart contract technology. In that case, arguing that blockchain technology and smart contracts will solve problems where no problem can be found, might create even more problems or confusion. In other words, the rush of solving legal problems with blockchain technology and smart contracts does not always go up in all of the situations. Nevertheless, the discussion regarding the use of smart contracts and whether it will replace traditional ways of contracting made clear that contracts (and humans) are not perfect and uncovered several weaknesses that are faced nowadays with the conclusion of contracts. As a consequence, solutions for all sorts of (contractual) problems are implemented and new concepts and structures are thought of. For now, it can be concluded that smart contracts are still in its infancy and need time to evolve before they can be implemented on a larger scale and used in more complex commercial contracts. Nonetheless, as the expectations of smart contracts and blockchain technology might be running ahead of what the *current* technology really can deliver for now, it cannot be underestimated for the longer term. According to my view, smart contracts will entail a more innovative character than just improving *existing* contracts since it will create a start of new paradigms and concepts about contracts that might not be envisioned right now. Consequently, a first reconnaissance and recognition of smart contracts within Dutch contract law can be seen as a necessary first step towards a broader acceptance and, additionally, help to improve contractual relationships between commercial entities, financial institutions, governments and consumers. Although this thesis made clear that Dutch contract law will not stand in the way of concluding smart contracts in general, several experiments could be set up to test what kind of *impact* smart contracts will have within the legal sector, to eventually set up regulatory additions that serve smart contracts and their underlying technology. As discussed in this thesis, regulatory sandboxes could provide a good “practice area” where the risks, social value and benefits for potential users of smart contracts can be assessed in a better way, without making “hasty” decisions about regulating the key aspects. After all, smart contracts and blockchain technology are just at the beginning of their diffusion of innovation.

Bibliography

Literature

Al Khalil
F. Al Khalil, et al., '*A Solution for the Problems of Translation and Transparency in Smart Contracts*' (2017)

Al Khalil
F. Al Khalil et al., "*Trust is Smart Contracts is a Process, As Well*", University College Cork, April 2017

Asser & Hartkamp
Asser/Hartkamp & Sieburgh 6-III, 2014

De Backer & De Boe
A. de Backer & V. de Boe, "*Smart contracts in de financiële sector: grote verwachtingen en regulatoire uitdagingen*", Computerrecht 2017/252

Clack, Bakshi & Braine
C.D. Clack, V.A. Bakshi & L. Braine, "*Smart Contract Templates: foundations, design landscape and research directions*", August 4 2016

Cong & He
L.W. Cong & Z. He, "*Blockchain Disruption and Smart Contracts*", University of Chicago Booth School of Business (2017)

Delmolino
K. Delmolino, et al, "*Step by Step Towards Creating a Safe Smart Contract: Lessons and Insights from a Cryptocurrency Lab*", Financial Cryptography and Data Security

Van Eersel & Van den Bergh
M. van Eersel & T. van den Bergh, *"Blockchain en smart contracts: toegang tot een reeks van slimme dingen"*, FRP 2017/457 (afl. 4)

Hansen
J.D. Hansen et al, "*More Legal Aspects of Smart Contract Applications*", Perkins Coie LLP, March 2018

Harz
D. Harz, "*Trust and verifiable computation for smart contracts in permissionless blockchains*", 2017

Hijma
J. Hijma, "*Rechtshandeling en Overeenkomst*", 6e druk, Deventer: Kluwer 2010

Kaal & Calcaterra
W.A. Kaal and C. Calcaterra, "*Crypto Transaction Dispute Resolution*", The Business Lawyer Spring 2018

Lessig
L. Lessig, "*Code: version 2.0*", New York, Basic Book: 2006

Linneman
J. Linneman, "*Juridische aspecten van (toepassingen van) blockchain*", Computerrecht 2016/218, afl. 6

Mik
E. Mik, "Formation Online", 159, *Contract formation: law and practice*, OUP, 2010

Mik
E. Mik, "*Smart Contracts: Terminology, Technical Limitations and Real World Complexity*", 2017

Ten Napel
H. ten Napel, "*Snelheid, efficiëntie en transparantie: het Nederlandse wetgevingsproces in transitie*?", Beleid en Maatschappij 2013 (40) 4

Nikolic
I. Nikolic et al, "*Finding the Greedy, Prodigal, and Suicidal Contracts at Scale*", School of Computing, NUS Singapore, March 14, 2018

Scherer
M. Scherer, "*Performance and Scalability of Blockchain Networks and Smart Contracts*", UU, 2017

Schmaal & Van Genuchten
J.B. Schmaal & E.M. van Genuchten, "*Smart contracts en de Haviltex-norm*", Tijdschrift voor Internetrecht, nr.1 maart 2017

Schotman
E.H. Schotman, "*Ethiek en recht in kort bestek*", 2017

Schuringa
H. Schuringa, "*Enkele civielrechtelijke aspecten van blockchain*", Computerrecht 2017/254

Scott
B. Scott, "*How Cryptocurrency and Blockchain Technology Play a Role in Building Social and Solidarity Finance?*", UNRISD Working Paper 2016-1

Strikwerda
L. Strikwerda, "*Inleiding tot het Nederlandse Internationaal Privaatrecht*", 2015

Surden
H. Surden, '*Machine Learning and Law*' (2014) 89 *Washington Law Review* 87

Surden
H. Surden, "*Computable Contracts*" (2012) 46 *UC Davis Law Review* 635

Szabo
N. Szabo, "*Smart Contracts: Building Blocks for Digital Markets*", Extropy nr. 16 (1996)

Szabo
N. Szabo, "*Formalizing and Securing Relationships on Public Networks*", First Monday 1997, vol. 2 nr.9

Szabo
N. Szabo, "*A Formal Language for Analyzing Contracts"*, 2002

Tjittes
R.P.J.L. Tjittes, "*Veelvoorkomende misverstanden bij gebruik van Anglo-Amerikaanse termen in het internationale contracteren*", Contracteren 2008/2

Tjong Tjin Tai
E. Tjong Tjin Tai, "*Smart contracts en het recht*", NJB 2017/146

Voulon
M.B. Voulon, "*Automatisch contractreren*" (diss. Leiden), Leiden, Leiden University Press: 2010

Reports

Allen & Overy LLP, "*Smart Contracts for Finance Parties*"

Chamber of Digital Commerce, *"Smart Contracts: 12 Use Cases for Business & Beyond: A Technology, Legal & Regulatory Introduction"*, December 2016

Clifford Chance, "*Smart Contracts (Code vs. Contract)",* January 2018
https://talkingtech.cliffordchance.com/en/tech/smart-contracts--code-vs-contract-.html

Clifford Chance Report, "*Are Smart Contracts Contracts?*", December 2017

Dutch Blockchain Coalition, "*Smart contracts as a specific application of blockchain technology*", 2017

European Securities and Markets Authority, "*The Distributed Ledger Technology Applied to Securities Markets*", February 2017

Norton Rose Fulbright LLP, "*Can smart contracts be legally binding contracts?*", November 2016

PricewaterhouseCoopers LLP, "*Contract management: contract value and minimize risks: A necessary response to the overwhelming quantity and diversified complexity of contracts*", 2003

Legislation

Directive 2000/31/EC of the European Parliament and Council

Dutch Civil Code, Book 2, 3, 6 and 7

Parliamentary History

Kamerstukken I, 2016/2017, 33 009

Kamerstuk 33 009, nr. F.

Kamerstukken II 2000-2001, 28 197, nr. 3

Parlementaire Geschiedenis van het Nieuwe Burgerlijk Wetboek", Book 6

Case Law

HR 13 maart 1981, ECLI:NL:HR:1981:AG4158, NJ 1981, 635 (Haviltex)

HR 19 januari 2007, LJN AZ3178, NJ 2007/575 (Meyer Europe/PontMeyer)

HR 29 juni 2007, LJN BA4909, NJ 2007/576 (Derksen/Homburg)

HR 5 april 2013, ECLI:NL:HR:2013:BY8101 (Lundiform/Mexx)

Internet articles and blogs

D.M. Adlerstein, "*Are Smart Contracts Smart? A Critical Look at Basic Blockchain Questions*", June 2017, https://www.coindesk.com/when-is-a-smart-contract-actually-a-contract/

S. Grybniak, "*Advantages and Disadvantages of Smart Contracts in Financial Blockchain Systems*", December 2017, https://hackernoon.com/advantages-and-disadvantages-of-smart-contracts-in-financial-blockchain-systems-3a443145ae1c

M. von Haller Gronbaek, "*Blockchain 2.0, smart contracts and challenges*", 2016, https://www.twobirds.com/en/news/articles/2016/uk/blockchain-2-0--smart-contracts-and-challenges

W.A. Kaal, C. Calcaterra, "*Smart Contract Dispute Resolution – The Need for an Open Source Blockchain Platform Ecosystem*", June 2017: https://medium.com/@wulfkaal/smart-contract-dispute-resolution-the-need-for-an-open-source-blockchain-platform-ecosystem-e6318610fdef

P. Kasireddy, "*Bitcoin, Ethereum, Blockchain, Tokens, ICOs: Why should anyone care?",* July 2017, https://hackernoon.com/bitcoin-ethereum-blockchain-tokens-icos-why-should-anyone-care-890b868cec06

J. Pfeffer, "*Over 12,000 Ether Are Lost Forever Due to Typos*", Consensys.net (Mar. 9, 2018): https://media.consensys.net/over-12-000-ether-are-lost-forever-due-to-typos-f6ccc35432f8

C. Sproule, "*As smart contracts get smarter, the rules of development will change*", 2018, https://venturebeat.com/2018/02/18/as-smart-contracts-get-smarter-the-rules-of-development-will-change/

J. Stark, "*Making Sense of Blockchain Smart Contracts*", June 2016: https://www.coindesk.com/making-sense-smart-contracts/

E.P.M. Vermeulen, "*Technology Isn't Fast Enough in a Platform Economy*": https://hackernoon.com/technology-isnt-fast-enough-in-a-platform-economy-d6627fd97a27

Webistes

https://www.forbes.com/sites/rogeraitken/2017/11/21/smart-contracts-on-the-blockchain-can-businesses-reap-the-benefits/#61a9857c1074

https://www.coindesk.com/information/ethereum-smart-contracts-work/

https://hackernoon.com/ethereum-turing-completeness-and-rich-statefulness-explained-e650db7fc1fb

https://hackernoon.com/smart-contracts-turing-completeness-reality-3eb897996621

https://github.com/ethereum/wiki/wiki/White-Paper/f18902f4e7fb21dc92b37e8a0963eec4b3f4793a

https://blog.ethereum.org/2016/07/15/to-fork-or-not-to-fork/

https://blog.ethereum.org/2016/07/20/hard-fork-completed/

https://blog.oraclize.it/overcoming-blockchain-limitations-bd50a4cfb233

https://www.technologyreview.com/s/610718/states-that-are-passing-laws-to-govern-smart-contracts-have-no-idea-what-theyre-doing/

http://solidity.readthedocs.io/en/v0.4.24/

https://www.mckinsey.com/industries/high-tech/our-insights/how-blockchains-could-change-the-world

https://martechtoday.com/a-new-report-bursts-the-blockchain-bubble-216959

https://hackernoon.com/2017-the-year-blockchain-went-mainstream-119863d5d9f3

Annex 1: Solidity contract example[311]

Solidity is a contract-oriented, high-level language for implementing smart contracts. It was influenced by C++, Python and JavaScript and is designed to create smart contracts on Ethereum. Instead of written provisions in natural language, Solidity consists out of programming language which describes what will be automatically executed as soon as several requirements have been fulfilled.

Before analyzing the example Solidity contract, it will be necessary to discuss first some structure elements of a Solidity contract: this will make the example Solidity contract easier to understand. Although Solidity contracts are similar to other object-oriented languages, it can be said that some aspects still differ. Solidity contracts are perfect to use and create smart contracts for voting, crowdfunding, blind auctions, multi-signature wallets. The basic structure elements of a basic Solidity contract (and the example) are the following:

```
pragma solidity ^0.4.0;

contract SimpleStorage {
    uint storedData; // State variable
    // ...
}
```

- **State variables**: state variables are values which are permanently stored in the Solidity contract storage (see blue marking in the example above). A contract in the sense of Solidity is a collection of code (its *functions*) and data (its *state*) that resides at a specific address on the Ethereum blockchain. The abovementioned line "`uint storedData`" declares a state variable called "`storedData`" of type "`uint`" (unsigned integer of 256 bits). You can think of it as a single slot in a database that can be queried and altered by calling functions of the code that manages the database. In the case of Ethereum, this is always the owning contract.

```
pragma solidity ^0.4.0;

contract SimpleAuction {
    function bid() public payable { // Function
        // ...
    }
}
```

- **Functions**: functions are the executable units of code within a contract. These are stated as "`function`" within the Solidity contract (see also the example above). Functions can be called directly or indirectly: this means that the function will be called within the contract itself (directly) or from an external message (indirectly).

[311] Example pieces of code are used from website http://solidity.readthedocs.io/en/v0.4.24/

```
pragma solidity ^0.4.21;

contract SimpleAuction {
    event HighestBidIncreased(address bidder, uint amount); // Event

    function bid() public payable {
        // ...
        emit HighestBidIncreased(msg.sender, msg.value); // Triggering event
    }
}
```

- **Events**: events within a Solidity contract, stated as "**event**", are dispatched signals that the smart contracts can fire when they are triggered (see the example and blue marking above). Events are inheritable members of smart contracts. When they are called (triggered), they cause the arguments to be stored in the transaction's log - a special data structure in the blockchain. These logs are associated with the address of the contract and will be incorporated into the blockchain and stay there as long as a block is accessible (most of the time forever).

The example Solidity contract that is used in paragraph 1.2 will now be discussed, since the basics of the structure of a Solidity contract are mentioned

Example Solidity contract

```
pragma solidity ^0.4.21;

contract Coin {
    // The keyword "public" makes those variables
    // readable from outside.
    address public minter;
    mapping (address => uint) public balances;

    // Events allow light clients to react on
    // changes efficiently.
    event Sent(address from, address to, uint amount);

    // This is the constructor whose code is
    // run only when the contract is created.
    function Coin() public {
        minter = msg.sender;
    }

    function mint(address receiver, uint amount) public {
        if (msg.sender != minter) return;
        balances[receiver] += amount;
    }

    function send(address receiver, uint amount) public {
        if (balances[msg.sender] < amount) return;
        balances[msg.sender] -= amount;
        balances[receiver] += amount;
        emit Sent(msg.sender, receiver, amount);
    }
}
```

```
pragma solidity ^0.4.21;
```

The first line of code is the annotation with a so-called “version pragma”, which rejects being compiled with future compiler versions of Solidity that might introduce incompatible changes. The first line simply tells that the source code is written for Solidity version 0.4.21 or anything newer that does not break functionality (up to, but not including, version 0.5.0). This is to ensure that the contract does not suddenly behave differently with a new compiler version (the versions update very regularly). The keyword pragma is called that way because, in general, pragmas are instructions for the compiler about how to treat the source code.

```
contract Coin {
```

A contract in the sense of Solidity is a collection of code (its **functions**) and data (its state) that resides at a specific address on the Ethereum blockchain. The abovementioned contract will implement the simplest form of a cryptocurrency, as is stated in the second line.

The (//) are the starting point of implementing some single line comments within the Solidity contract. With these comments, participants may add some extra information. The keyword “**public**” automatically generates a function that allows a participant to access the current value of the state variable from outside of the contract. Without this keyword, other contracts have no way to access the variable (private vs. public). The public accessibility is important for most of the cryptocurrency transactions: Bitcoin for example is a public blockchain network.

The line “address public minter”; declares a state variable of type address that is publicly accessible. The address type is a 160-bit value that does not allow any arithmetic operations. It is suitable for storing addresses of contracts or keypairs belonging to external persons. The **event** (= dispatched signals which the smart contracts can fire when they are triggered) describes sending coins from one address to another address about a specific amount of coins, which are mentioned in the following line:

```
event Sent(address from, address to, uint amount);
```

The following **functions** of the example contract describe the executable units of code within the contract as soon as the **event** is triggered. In the example below, the functions describe the sending of the coins from and to the specific balances of the contracting parties (see “sender” and “receiver”). The second and third function also update the coin balances of the sender and receiver. Important to note is the “if”-factor, which can trigger something or not. The fact that it is public means that all of the participants within the blockchain network can see from which account coins have been send to another account. The transaction is added to the blockchain:

```
function Coin() public {
        minter = msg.sender;
    }

    function mint(address receiver, uint amount) public {
        if (msg.sender != minter) return;
        balances[receiver] += amount;
    }

    function send(address receiver, uint amount) public {
        if (balances[msg.sender] < amount) return;
        balances[msg.sender] -= amount;
        balances[receiver] += amount;
        emit Sent(msg.sender, receiver, amount);
    }
}
```

As a result of the abovementioned structures, the example Solidity contract will automatically send an amount of cryptocurrency from one participant to another if several actions are fulfilled. Although the code itself is written in language that readable (i.e. not only numbers), it can be assumed that it will not be understandable for the majority of participants and/or judges. Understanding the underlying intentions can definitely be seen as problematic.