# Smart contracts

## as a specific application of blockchain technology

First reconnaissance of questions relating to legislation, regulations and future knowledge needs as a consequence of blockchain technology and more specifically, smart contracts.

Smart Contract Working Group – Dutch Blockchain Coalition

Dutch Blockchain Coalition
connect and create

www.dutchblockchaincoalition.org

Dutch
Blockchain
Coalition

connect and create

# Smart contracts as a specific application of blockchain technology

First reconnaissance of questions relating to legislation, regulations and future knowledge needs as a consequence of blockchain technology and more specifically, smart contracts.

Smart Contract Working Group – Dutch Blockchain Coalition
Olivier Rikken MSc MBA
Sandra van Heukelom - Verhage LLM
Sander Mul LLM
Jacob Boersma MSc
Inger Bijloo LLM
Pascal Van Hecke
Arne Rutjes LLM
Femke Stroucken LLM
Joost Linnemann LLM
Hidde Terpoorten MSc
Robert Reinder Nederhoed MSc

# Summary

In the course of 2016 it became clear that there was a growing need for knowledge of blockchain. The Dutch Digital Delta blockchain expert group started work in June 2016 with particular emphasis on mapping the knowledge needed. As a result of this initiative, a Working Group was assembled and given the name Smart Contract/ Legal Programming Working Group. The remit of this working group was to answer the following questions:

1. Establish which questions (and any gaps) could arise with respect to legislation and regulations that touch on the topic of smart contracts.
2. Establish which knowledge will be needed in the future and identify who or which body could meet these knowledge needs in practice and when.

The conclusion drawn from the various working group meetings and from literature studies was that, before a clear answer could be provided to the main questions, it would be necessary to define exactly what smart contracts are. And the main point here was that – as a consequence of smart contracts being in the form of computer code – smart contracts are about operational semantics and/or operational agreements, rather than denotational semantics (e.g. under which laws, subject to general terms and conditions, etc.). It was concluded that smart contracts only have a *legal* manifestation in specific cases. Which law applies (in that case), depends on the nature of that legal manifestation.

The working group ultimately reached the following conclusions:

1. A blockchain smart contract is, in the first place, a deterministic computer program that is deployed and executed on a blockchain.
2. A smart contract may have legal significance, but not necessarily so.
3. Smart contract technology can be put in place in various legal domains (private law, administrative law, criminal law) and can therefore have various manifestations.
4. Not every legal manifestation (statutory provision, obligation, etc.) lends itself to being converted into code.
5. Where conversion into code is possible, it is advisable only to do this to <u>execute</u> the recognisable (plain language) manifestation. In administrative law and criminal law – where rights and duties are established - this would seem to be the appropriate way forward on the grounds of legal certainty, but it may also be required in private law e.g., to protect consumers.
6. When the parties intend the code (itself) to create an obligation in a private law manifestation and possibly also to accept the outcome of the execution in advance, this intention should at least be laid down in writing (i.e. not in code, but in a formal language, for example). This agreement too could be registered on a blockchain.
7. When designing a solution, one must already consider the actual and legal possibilities in advance in order to (a) link the automatic execution of the contract to pre-determined terms and conditions (such as permission of the parties or a third party) and (b) 'nullify' the execution (or its consequences) in retrospect (return to the former situation, compensation, damages, etc.). Attention must also be paid to the applicable law and the competent authority (mediator, arbitrator, court, etc.) in the event of a dispute.

8. A clear distinction should always be made between *permissioned* and *permissionless* blockchains, since their governance models may be different. A permissioned blockchain can be protected by an *access control layer*. In contrast to a permissionless blockchain, not everyone can participate. Approval in advance is required. Furthermore, read and write access rights may differ for users, which also means that tasks and responsibilities can be divided up. In short, there is an organisation, frequently an alliance, behind a permissioned blockchain.
9. Personal data may be incorporated into smart contracts. Personal data are data that are directly or indirectly traceable to a natural living person. Citizens have the right to have their personal data protected (under the Dutch Personal Data Protection Act and the General Data Protection Regulation). In the case of a permissioned blockchain, it is possible to arrange who is responsible for complying with the requirements of the Dutch Personal Data Protection Act. The arrangements are different in a permissionless blockchain. No one and everyone is in charge of a permissionless blockchain and agreements of that kind are much more difficult to make, due to the lack of restrictions on access and lack of control over governance. The possibility of protecting privacy in such situations will have to be investigated further.

Evaluating the manifestations against the law results in the preliminary conclusion that major changes in laws and regulations would not appear necessary in order to deploy smart contracts in the legal order.

Three key pillars, each with two sub-pillars, have been identified with regard to future knowledge requirements. The three key pillars in which knowledge building will have to take place are:

1. Blockchain knowledge, with sub-groups on:
    a. general (technical) knowledge of blockchain
    b. knowledge of new business models and industry models as a consequence of blockchain and smart contract implementations
    c. how to deal with governance
2. Software and IT knowledge, with sub-groups on:
    a. programming languages, both those that already exist and new ones such as “Solidity”
    b. front end to back end interaction and integration, with reference to the various translations that will be made during compiling, implementations and the integration into current models and systems.
3. Legal & Risk, with subgroups on:
    a. legal, both general blockchain-related legal issues such as jurisdiction, privacy, et cetera and specialisations in various areas of the law, and risk & governance
    b. how to build a good governance and risk management structure into smart contracts and blockchain environments.

Knowledge needs to be build up in two ways in the three pillars and sub-pillars. On the one hand, in specialisations in the various sub-areas identified, and on the other, a fast-growing need for cross-expertise in these knowledge areas will result in people with multidisciplinary expertise: they will still specialise in a key pillar or sub-pillars, but they will also have thorough knowledge of one or more other pillars.

The following recommendations for subsequent steps were defined on the basis of this initial exploration:

1. A more precise exploration into legal issues that arise as a consequence of the use of smart contracts
2. With respect to the need for expertise in the pillars and sub-pillars described above, an investigation should be conducted in conjunction with the identified bodies to identify which needs can already be met and which pillars require new modules.
3. In addition, besides expanding in-depth knowledge of each pillar, work should be done on increasing cross-pillar expertise to develop multidisciplinary expertise.
4. The designation of a clear central point of contact, not only for further development of legislation and regulations but also for monitoring and developing how knowledge requirements can be met.
5. Research into the possibilities for standardising smart contracts with respect to three matters: pattern design, ontology and the standardisation of individual data elements.

# Structure of this report

This report has been prepared for readers with a general interest, for people with a technical background and for people with a legal background. Information about the structure of the report has been added due to the fact that blockchain and, more specifically, smart contacts represent the overlapping of two worlds that were previously largely separate and because the working group sessions clearly concluded that an uniform vocabulary would be highly desirable. For clarification, when we refer to smart contracts in this report, unless otherwise specified, we mean smart contracts on a blockchain.

We recommend that everyone reads the chapter on understanding and interpretation, given that, in practice, this was where the most confusion arose. Reading the piece about *smart contracts is very highly recommended,* even if you already know a lot about blockchain.

Readers with a legal background are referred to the piece that includes an in-depth discussion of legal matters: legal questions and gaps with regard to smart contracts. Here, smart contracts are explored and examined on the basis of various legal manifestations to see whether these could be encapsulated in a smart contract. For people with a more technical background and focus then, in particular, the signs that indicate that smart contracts may be more than code are essential reading in this chapter.

The chapter on knowledge requirements is relevant for readers with both a legal and a technical background, mainly because it enables them to understand which additional knowledge should be acquired to deal with smart contracts responsibly.

# Table of contents

# Introduction - background - aims

A blockchain expert group led by Ad Kroft, programme manager at Dutch Digital Delta, gathered at the Dutch Association of Insurers on 22 June 2016 at the initiative of the Dutch Digital Delta. This initiative later gave rise to the Dutch Blockchain Coalition[1].

The aim of this group was multi-faceted. Besides sharing knowledge and experience and connecting various experts in the specialist area of blockchain and related topics, various key areas were identified in which questions need to be answered before blockchain can be (more) widely deployed by government and industry. Via the Dutch Blockchain Coalition, the Netherlands wants to assume a leading position in the area of blockchain.

During one of the sessions of the Dutch Digital Delta blockchain expert group, a working group (the "Working Group") was launched, prompted by a presentation by Olivier Rikken on 28 September 2016. Its task was to conduct a mapping exercise to identify the need for knowledge development and to identify questions with regard to legislation and regulations specifically on the matter of smart contracts as one of the most important products/services [applications] building upon blockchain technology. The broadly supported international expectation is that contact with and the use of blockchain technology and, in particular, smart contracts will increase in practice.

At the first meeting of the Working group, the reason for the launch of that working group was documented as follows:

> *"The rise of smart contracts by means of various blockchains and the characteristics often associated with this ('irreversibility' in combination with actual direct payment/transfer of value caused by a valid trigger of a smart contract without still requiring the intervention of people) as well as possible new business models means that the correct (legal and risk-specific) programming of these contracts will become essential.*
>
> *That is why this exploration of current and future knowledge requirements as a consequence of smart contracts is being made. It will not only cover possible knowledge requirements in the future but also possible legal issues surrounding this topic."*

Blockchain is what is known as a 'fundamental' technology and is therefore not linked to any single or specific applications. Many applications in many industries are possible on the basis of blockchain technology. Blockchain can therefore act as the *enabler* of even more products and services. In this initial exploration and in this report, however, the Working Group has focused specifically on smart contracts.

This report is the first result to emerge from the Working Group meetings and it contains an initial survey of future knowledge requirements and questions regarding legislation and regulations. As it is an initial survey in a very volatile area, the report only provides limited answers.

[1] https://www.dutchdigitaldelta.nl/blockchain

# Aims and composition of the Working Group

When it started, the Working Group defined two different aims, which it pursued. These aims were prompted by practical problems encountered by various companies, proofs of concept and business start-ups. All were related to smart contracts.

## Aims

The Working Group's first aim was related to (the lack of) technical as well as legal knowledge of smart contracts. In practice, it was found that only a few people were able to draw up smart contracts. Moreover, the general public is not yet sufficiently familiar with the concept of *smart contracts*; the concept does not yet have a clear definition other than the technical and conceptual descriptions of what smart contracts are. That makes it more difficult for people who have read less or not yet at all about blockchain to understand what a smart contract actually is, how it is realised and how it works.

The first aim focuses on questions that could arise as a result of legislation and regulations that touch on the topic of smart contracts. After all, no specific regulations have been developed yet with regard to blockchain technology, let alone with regard to smart contracts. The first aim of the smart contract working group was therefore:

*To establish which questions (and any gaps) could arise with respect to legislation and regulations that touch on the topic of smart contracts.*

The second aim is in line with the first aim. However, its focus is primarily on a knowledge area that needs to be developed. The currently well-defined knowledge areas such as Information Technology (IT) on the one hand, and legal knowledge on the other, are probably not sufficient to clarify this definition. This points at a need for cooperation between these two knowledge areas. The second aim of the smart contract working group was therefore:

*To establish which knowledge will be needed in the future and identify who or which body could meet these knowledge needs in practice and when.*

## Composition and members of the working group

The intention right from the first meeting of the Working Group was to represent the following sectors as broadly as possible:

1. Government
2. Regulators
3. Educational institutions
4. Industry

The aim was always to have each of these sectors represented as broadly as possible too. The Working Group therefore comprises of representatives from four universities and universities of applied sciences, two regulators, four government authorities and various technical and legal businesses, ranging from business start-ups to large, well-established corporates. Ultimately, the following parties attended, contributed to and submitted input to the Working Group:

1. Government
    a. Dutch Ministry of Economic Affairs and Climate Policy
    b. Dutch Ministry of Justice and Security
    c. Dutch Ministry of Finance
    d. Dutch Academy for Legislation
2. Regulators
    a. De Nederlandsche Bank
    b. Netherlands Authority for the Financial Markets

3. Educational institutions
    a. Tilburg University
    b. Leiden University
    c. Nyenrode Business Universiteit
    d. Jheronimus Academy of Data Sciences
4. Industry
    a. Deloitte
    b. APG
    c. AXVECO
    d. BLandLord
    e. IBM
    f. Unchain
    g. Kennedy van der Laan Advocaten
    h. Pels Rijcken Advocaten
    i. Van Doorne Advocaten
    j. CMS
    k. Token Engineers

The Working Group has always been open to new members. Members of the Working Group and external people also always had the freedom to invite new parties to the table.

# Methodology

From December 2016, the Working Group met on average once a month.

During the first meeting, the Working Group agreed on the aims, expanded the list of members and "other" agreements were made for the Working Group. This included the agreement that all debates, conclusions, et cetera would be made public and therefore accessible to everyone. The technical background to smart contracts was examined in more detail in the following meeting.

During this session of the Working Group, we realised that if we continued to reason purely on the basis of the theory, the discussion would never advance beyond theory and hypotheses. The working group took the application of technology as its starting point. This application was analysed. The working group did not conduct a scientific analysis of the technology or principles of smart contracts.

For that reason, from the second session of the Working Group onwards, the discussion always started with practical examples, such as live use cases and proofs of concept from various parties in order to identify practical problems and place them within the theoretical frameworks established in various other reports. In addition, "lessons learned" from abroad were used. The following case studies from practice were dealt with at these meetings:

1. Real estate contract applications
    a. BLandLord – crowd ownership smart contracts and blockchain application – contracts of sale [2]
    b. Deloitte – Handelsgebouw Rotterdam smart contract and blockchain-based leases [3]
2. Financial services smart contract applications (non-banking)
    a. APG – various smart contract applications for pensions [4]
    b. OurSurance – peer2peer insurance and unbundling the current insurers' business model by means of smart contracts [5]
3. Government applications
    a. A summary of the various public authority blockchain and smart contract cases [6]
    b. IBM - registering, insuring and tracking electric bikes[7]

[2] https://www.blandlord.com/

[3] https://www2.deloitte.com/nl/nl/pages/over-deloitte/articles/huurcontracten-voor-het-eerst-vastgelegd-in-blockchain.html

[4] https://www.apg.nl/nl/artikel/innovatiefonds-apg/1947

[5] www.oursurance.nl

[6] https://www.blockchainpilots.nl/

[7] https://www.linkedin.com/pulse/how-blockchain-helps-reduce-bike-theft-bram-havers

A number of authors have contributed to the creation of this report. The content of this report does not necessarily reflect the opinions of the organisation where these authors work.

# Concept development and interpretation

The term "smart contract" is doubly confusing. In the first place, the application of this technology does not necessarily constitute a contract in a legal sense. In the second place, a smart contract cannot be called intelligent of itself. In essence, a smart contract does what it is told; no thought or pro-activity is involved, and all of the rules are pre-programmed. From a technical standpoint, a smart contract is perhaps best viewed as a deterministic computer program that is deployed and executed on a blockchain and it is (therefore), by definition, not under the control of a single party. The legal interpretation of a smart contract depends entirely on the specific application.

For example, a smart contract makes it possible to perform a (economic or otherwise) transaction automatically online, once the program establishes that certain conditions – established beforehand in the code – have been satisfied. In this case, one could indeed consider it an agreement, but the question is then once again whether the code constitutes the agreement or whether this is simply intended for its execution (in which case there is a written agreement, for example, or another document showing the intentions of the parties involved). But other applications are also quite plausible. Consider donations, for example, or earmarking grant funds, granting a permit, supervising, et cetera. Not only do other private law activities then come into the picture (donations), but also administrative legal activities (granting permits) and forms of administrative activities in which decision-making, supervision and enforcement merge as it were (earmarking grant funds).

A legal interpretation of the use of smart contracts requires an understanding of how blockchain technology, smart contracts and oracles work. These concepts are therefore further explained below.

## Blockchain

Blockchain is a general term for technologies intended to synchronise data stored in a distributed manner (on various computers and/or servers) via a network, so these data remain the same. To this end, a consensus protocol is put in place to guarantee the integrity of the content of the database. Cryptography plays a significant role in this. The crux of blockchain is that synchronisation takes place on a peer-to-peer basis, which means that none of the computers have control in the network. Eventually, this technology could be used to redesign or even substitute certain tasks of so-called "trusted third parties" such as a cadastre, notaries or a central bank, or even make them redundant. These tasks would primarily involve the irrefutable registration of certain information and performing standard audits.

Every participating computer will accept a proposed change in the set of data only after it has determined for itself that the change is taking place in accordance with pre-determined rules (in the most prevalent case, such a rule is: "the change must be made by the party for which it was established that they are entitled to perform that change"). As the system is peer-to-peer, i.e. without an "authoritative" party in the network, changes can be performed and forwarded from various places in the network that in and of themselves comply with the rules, but which contravene each other (i.e. they result in data sets that differ from each other).

The mechanism recorded in the blockchain software ensures that the network of computers ultimately reaches agreement about the "true" data set is called the consensus protocol. The precise content of this protocol can vary for each blockchain.

## Bitcoin - the first blockchain implementation

Blockchain's origin is Bitcoin, a so-called crypto-currency. The designer of Bitcoin wanted to create a system in which parties can pay each other online without the intervention of banks or other financial institutions (!). The question then was: in the absence of a so-called trusted third-party (see the following paragraph), who will check whether the paying party has sufficient funds to make sure that this party doesn't secretly try to transfer the same value twice (the double-spending problem). The solution was to have the network itself do this: every participating computer checks whether a transaction can be performed, and is also a witness to ensure that the balance is not issued twice. And so the blockchain was born, in fact a registry (it's often compared with a ledger) in which the history of all the trusted Bitcoin transactions performed on the network is recorded. If a computer drops out of the network, this is not a problem. After all, there are many other computers with a copy of the ledger and each computer can audit the proposed transactions independently. One remarkable property of blockchain is that all data is saved and cannot be changed unilaterally after the fact, so, in principle, data is only added.

## Blockchain as a replacement for Trusted Third Parties

We noted above that, thanks to blockchain technology, payment can be made safely with Bitcoin without the intervention of a bank. Others quickly realised that blockchain is, in essence, a generic technology that can be used in any situation requiring a data set that is maintained jointly and that cannot be manipulated by any of the parties unilaterally. In other words, blockchain allows the establishment of a shared single source of truth between two parties without these parties having to resort to a neutral, trusted, third party.

Essentially, anything can be recorded in that data: in addition to the ownership of Bitcoins, also the ownership of a good, a (legal) power, a diploma, a permit, medical information, et cetera. Blockchain technology can also be used (symbolically or otherwise) to *transfer value*. If a particular asset (a house, for example) can be identified on a blockchain, then it is also conceivable that this house can change owners via the blockchain (however, please see the example below).

An important precondition must now be made: simply the intention of one or more of the parties to create legal obligations via a blockchain application or possibly also to execute an obligation does not mean that all of the legal requirements have been met for this to succeed. For example, in technical and economic terms, the sale of a house via a smart contract can be done relatively easily, but the question of whether this so called smart contract constitutes a legally valid agreement remains. Furthermore, under current (Dutch) law, notarial intermediation is required for the transfer of a house.

The possibility of maintaining data and possibly transferring value without the intervention of a trusted third party means that the speed of business can increase, while costs can be reduced. Not only transaction costs, but also the costs of security, supervision and enforcement, for example. It can enable a self-organising group of persons/agencies to draw up its own set of rules for performing transactions and to execute these without deploying a third-party. This explains the disruptive potential of blockchain technology applications, certainly in combination with the use of smart contracts.

We indicated previously that certain Trusted Third Parties' tasks in the areas of administration audits may possibly disappear, i.e., be redesigned. At the same time, it should not be forgotten that Trusted Third Parties are often more than just glorified administrators. They can also play a role in the protection of the parties involved or of third-party rights. This can also prevent conflicts and that is also to the benefit of government.

# Permissioned vs. permissionless blockchains

One extremely important aspect of blockchain is the phenomenon of permissioned versus permissionless blockchain. Both are basically the same in that data storage takes place in a comparable manner by means of building up cryptographically linked blocks. The difference is in participation and rights. This in itself leads to entirely new discussions and facts concerning such issues as privacy and governance.

- A permissionless blockchain is a blockchain in which everyone is completely free to participate (anonymously). This means that anyone who wants to do so can participate immediately in this blockchain as a normal user or as a so-called "full node". No identification or authentication takes place on permissionless blockchains. Participants are therefore virtually completely anonymous in that sense, although pseudonymous would be more correct. So-called cryptographic key pairs are used in order to perform transactions: these are a (hash of a) public key and a secret private key. All transactions and all information in the particular blockchain are public. Everyone can also propose software updates – but an upgrade of the network takes place only if (a majority of) the participants voluntarily update the software on their own machines. So, in a permissionless blockchain, no single party is "the boss" and the chain also has no super users or equivalent structures. Whenever software updates are not accepted by a portion of the network, a network split (also called a fork) can occur, with two differing blockchains that have a shared previous history up to the point where the split in the chain of blocks occurred. A permissionless blockchain is also called a public blockchain. Bitcoin and Ethereum are the most well-known permissionless blockchains.
  Certain issues of sustainability (energy consumption) are associated with permissionless blockchains, along with costs (energy, hardware, computing power), processing speed and scalability (one block every 10 minutes for Bitcoin) and governance (distributed).
- A permissioned blockchain is protected by a so-called access control layer. Not just anyone can participate in a permissioned blockchain. These are blockchains in which an access request/approval is required and in which reading and writing privileges may vary for each user, for example. In theory, the data may even be stored on just a single computer ("node") and so a sort of superuser can be created.

> Permissioned blockchains are also called hybrid, consortium or private blockchains, depending on the number of different nodes and the types of users. Various software projects for building permissioned blockchains work under the Linux Foundation's Hyperledger project (e.g. Hyperledger Fabric, originally contributed by IBM, or Hyperledger Burrow, that is building further on Ethereum).

There is a huge difference between permissioned and permissionless blockchains with respect to governance and compliance. With respect to governance in a permissioned blockchain, a (group of) responsible party/parties is indeed designated, while in a permissionless blockchain, everyone - and, therefore, no one - appears responsible [8]. In a permissioned blockchain, privacy can be safeguarded more easily, for example, using reading and writing permissions; with the transparency of a permissionless blockchain, this is more difficult.

## Consensus mechanisms and immutability

Just as in every other computer network, blockchain applications must also take network attacks into account. In this manner, it is conceivable that a number of computers in the network collaborate in order to present a false image of the truth to other computers. This would appear to be a particular risk in so-called permissionless blockchains, where everyone is free to enter, since everyone in the world has a computer connected to the Internet, can participate anonymously in both the use of the application (e.g. paying someone) and in maintaining and safeguarding the blockchain in the context of that use. This means that the design must take into account anonymous evildoers who will try to compromise the system.

In order to counter attacks, Bitcoin and many other permissionless blockchains currently choose to work with a so-called proof-of-work system. Every 10 minutes [9] on average, this system designates a computer "at random" [10] that may propose a block of transactions to the other computers in the network [11]. Using mathematical verification (cryptography), it is easy to ascertain for the other computers that:

1. This computer has indeed earned the right to make a proposal.
   Proof: proof-of-work.
2. The proposed transactions do indeed exist, come from a party permitted to perform the transactions, and that the content has not been tampered with.
   Proof: digital signature.
3. The proposed transactions may indeed be performed in accordance with the applicable rules (e.g., the balance is sufficient). This means that it must also be demonstrable that the transaction history has not been tampered with.
   Proof: blockchain in the form of indissoluble Merkle trees linked to each other.

A consensus based on proof-of-work costs (a considerable amount of) money - namely in the form of energy, hardware depreciation and

[8] Many permissionless blockchains do have “core development teams” that provide the greatest contribution to further development of open source software and that play a de facto governance role within the community. Examples include the Bitcoin Core team and the Ethereum Foundation. However, anyone is free to add to the software and to participate in the community.

[9] The average time durations can vary for other blockchains, such as Ethereum ~15 seconds.

[10] Random in the sense that each computer has an equal chance, depending on computing power; the more computing power you have (and therefore the more you have invested), the greater your chance, of course.

[11] For how blocks with transactions are "published", see, for example: https://blockchain.info/nl

computing power that could also have been utilised elsewhere (for a higher return). The fact that the process costs money is, of course, no accident: the cost scares away "spammers". On the other hand, nothing is for free: in order to make it attractive to those of goodwill to spend money on the process of managing and securing the blockchain, the computer that is able to approve a block of transactions gets rewarded: in the case of Bitcoin, with new bitcoins (which is how bitcoins come about) and with the so-called transaction fees. In proof-of-work systems, transactions are presumed to be all the more secure (and therefore no longer reversible) the longer they have been registered. In practice, a Bitcoin transaction is considered unchangeable after an hour.

The proof-of-work mechanism is already much older than blockchain technology. This makes it the most proven mechanism at the moment to achieve consensus in a decentralised environment. This is also the reason that most permissionless and also a few permissioned blockchains use this mechanism. However, there are also multiple disadvantages associated with this mechanism, and therefore there are also many other consensus mechanisms used. Among the most prevalent alternatives [12] are:

1. Proof-of-Stake
2. Proof-of-Capacity
3. (P)BFT
4. PAXOS
5. RAFT

With permissionless blockchains, one sees primarily proof-of-work and proof-of-stake mechanisms. With permissioned blockchains, one sees greater diversity.

One of the most important elements of a blockchain is its so-called immutability. When something is placed in a blockchain, it should no longer be able to be reversed. However, some nuance should be noted here: "it cannot be reversed unilaterally". 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 [13]. But the challenge here is that if someone wants to do this in a permissionless blockchain, then all nodes in the network will have to cooperate in order to prevent a split in the network. The problem with this is that not all nodes in the network are known to everyone, so convincing everyone in the network is a difficult process. In a permissioned blockchain, all full nodes, the network's bookkeepers, are indeed known. This makes reversing transactions in a permissioned blockchain in which everyone must grant approval simpler than in a permissionless environment.

Finally: immutability simply means that it is certain that a given piece of information was once registered in a blockchain. It does not necessarily mean that that piece of information is also correct. For example, if the incorrect owner is registered in a registry of bicycle owners due to a human error, then that can no longer be expunged. But this still does not make the person in question the legal owner. In applications in which blockchain is to be used to represent real-world assets (see also the following paragraph), then it will also be necessary to build the application in such a way that the representation on the blockchain can be brought "into sync" with legal actuality (by building in exception procedures, for example).

## Native currencies versus issued assets

The most familiar application of (permissionless) blockchains among a wider audience is in so-called cryptocurrencies such as bitcoin, ether, dash et cetera. There are currently more than 750 blockchains with their own currency that can be traded publicly [14].

[12] http://www.coindesk.com/short-guide-blockchain-consensus-protocols/

[13] https://www.coindesk.com/ethereum-executes-blockchain-hard-fork-return-dao-investor-funds/

[14] https://coinmarketcap.com/currencies/views/all/

Each of those currencies is "inherent" or "native" to the blockchain on which they function: the currency is one way to link a cost price to transactions. After all, if transactions were free, then the network would be spammed to death in a permissionless blockchain - as has already happened on an Ethereum test network [15]. It is also the way in which parties that secure the network (using proof-of-work or proof-of-stake) can be compensated. The currency can exist only together with the associated blockchain, so the value of these currencies is inextricably linked with the utility the blockchain provides to the user. Applications were quickly found for blockchain that extend further than trading native currencies. With Bitcoin, using a few tricks, (such techniques as “coloured coins” or Omni and Counterparty), you can create your own "coins" and trade these via the blockchain. Other blockchains, such as Nxt (“assets”) and Ethereum (“tokens”) make this even easier - there are now hundreds of such popular crypto-assets [16].

Assets can represent monetary value, e.g. a claim on goods (assets that represent gold in a safe are popular!), a share of stock, or another type of security. In contrast to the native currency whose value is intrinsically linked to the functioning of the underlying blockchain, the value of an asset depends entirely on the issuer. As the holder of the asset, you conclude an agreement, either implicitly or explicitly, with the issuer allowing you to claim the underlying value.

However, assets can also represent such abstract things as membership, or the right to use certain software. The issuer is then often not even a legal entity, but a pseudonymous group of developers. In particular, in Ethereum's world of tokens and smart contracts, experiments are being done with all types of new business models and organisational forms, particularly ones in which token holders form a type of virtual company (a “Distributed Autonomous Organisation”), since each holder is motivated to increase the value of the token.

# Smart contracts and oracles

Smart contracts are applications that can be placed in a blockchain. At heart, a smart contract is a deterministic computer program that is deployed and executed on a blockchain. A computer program is deterministic if, given specific input and specific start values, it always generates the same output. In other words, its operation is completely predictable. In contrast to what the name might suggest, a smart contract does not necessarily mean the creation or performing of a contract or other legal act. For example, with a collection of interacting smart contracts and oracles, a business process can also be managed in a chain.

In order to determine whether the conditions for the performance of a smart contract have been met, data (input) from outside the blockchain will often be required, such as confirmation that the package has been delivered. A blockchain is "deaf and blind": the blockchain software cannot retrieve information from outside (other than that dictated by the protocol) [17]. This is where so-called oracles come into the picture. Oracles can provide input to a smart contract.

An oracle is a party (or a technical source such as a database, or a person who has been assigned that role) who plays the role of "source of the truth" for a smart contract. The other parties that use the smart contract trust that the oracle will provide the right information for the execution (of a function) in the smart contract, but cannot verify "on chain" that this was actually the right information. If parties do

[15] https://www.coindesk.com/ethereum-spam-attacks-back-time-test-network/

[16] https://coinmarketcap.com/tokens/views/all/

[17] For public blockchains, security is an important reason: the software is "sandboxed" and smart contracts have no access to the network or hard drive, for example. Since each node is located in a different environment, and therefore "sees" differing things, consensus would be impossible.

not wish to rely on a single source, they could even have various sources "vote".

The role of an oracle once again brings to mind a “Trusted Third Party”. An oracle can be solely a source of information and not involved in the execution of the contract. Furthermore, an oracle need not even know about the further use of the information provided. An oracle need not be a technical source such as a database, but a notary could also be behind an oracle, or a mediator whose signature is required for the execution of a (given function in the) contract.

Generally trusted institutes such as Royal Netherlands Meteorological Institute (KNMI), Rijkswaterstaat, et cetera, could provide digitally signed data feeds that are used as or by oracles in various blockchains to have insurance processed automatically, for example. However, as indicated previously, a designated person with the proper authorisation can also fill this role (e.g. in the form of a binding recommendation).

As stated, smart contracts can also be used (also symbolically) to transfer value. Wherever payment is made with crypto-currencies or assets/tokens, these can be "locked" in a smart contract until the moment it is established that the conditions of payment have been satisfied, or until a particular deadline has passed, allowing the committed amount to become liquid again. In certain cases, tokens can even be conditional on the actual exercise of a right. Consider a rental car, for example, that does not start unless a person is in possession of a particular virtual key.

Smart contracts are being increasingly deployed precisely for the transfer of value, for example in the so-called Ethereum blockchain. When you look at the composition of a smart contract on the Ethereum blockchain, it has these three most important main elements:

1. A balance (in which a varying amount of the crypto-currency "ether" can be saved).
2. A capability for (possibly rewritable) data storage. Here, statuses can be saved - for example, a package is on its way or has been delivered. But virtual tokens and their quantities can also be maintained here, whereby a token can represent a share of stock, for example.
3. The contract code. Based on the message that a smart contract receives, possibly in combination with values in the data storage, this code determines whether the data storage must be changed or whether crypto-currencies must be transferred.

These smart contracts have an address (more or less equivalent to an "account number") and messages or funds in crypto-currencies can be sent there. Smart contracts are reactive. This means that they do nothing until they receive a message or transaction. After receipt of the transaction, the code is activated and it determines whether something must be done with the message.

Once a contract is deployed on the blockchain, the code of the contract can no longer be changed. The balance or the storage can also no longer be manipulated other than by means of a specific message that can be sent to the contract via a transaction triggering the code. This, too, can only take place if the code contains functions permitting a change.

# The legal questions surrounding blockchain and smart contracts

The conclusion of the previous chapter is that, in the first place, a smart contract is a deterministic computer program that is replicated and executed on a blockchain. This chapter concerns the legal questions surrounding smart contracts.

Some people consider smart contracts to be the revolution that will make lawyers superfluous. After all, smart contracts make it possible for parties to cast their agreements into unchangeable program code; code which, furthermore – once the conditions of the agreement have been met – is executed automatically. In certain applications, this undoubtedly offers great advantages. However, the DAO affair demonstrated that things can also go wrong: in this case, someone absconded with millions of dollars' worth of ether (the cryptocurrency of Ethereum), because she/he recognised the error in the program code and exploited this for his/her own profit. Naturally, this raised the question of what is actually determinant: the intention with which a smart contract is drawn up, or the way in which that intention is cast into code. It also raised the question of liability of those involved. Consider, for example, the programmer, the party for whom the programmer worked, or the platform that provided the smart contract functionality. We should also note here that the jurisdiction nationality under which these questions must be answered and the court before which this must be heard are not automatically clear. Finally, the affair raised the question of whether a repair back to the previous condition was still possible and, if so, how. Ultimately, the consequence of the programming error was resolved by means of a change to the blockchain software itself, within which smart contracts are executed.

The term "smart contract" is an unfortunate term not only because it has no legal meaning, but also because it suggests that a contract is formed. As we shall explain below, smart contracts can play a role in various legal domains and we should also pay close attention to the intent behind the deployment: as a source of rights and obligations or simply for their execution. A relevant question in this context is also the extent to which (the exercise of) rights and obligations can actually be contained in program code. This question is addressed first. Various private law and public law "manifestations" are subsequently discussed. The chapter concludes with a number of general legal issues and a preliminary conclusion.

## To what extent can law be subsumed in program code?

Before addressing the legal manifestations of a smart contract, it is worth asking the question about the extent to which law and program code are actually comparable. After all, on the surface, law appears to be an algorithm: if this happens, then that is the consequence... or "if (X), then (Y)". In the 17th century, the famous mathematician Gottfried Leibniz predicted a calculus for calculating legal rights and obligations. His was a good prediction: these days, large portions of the implementation of laws and regulations are supported by information systems. But does this mean that all laws and regulations and all contractual obligations can be translated into code? Although quite a number of things might lend themselves well to automation, the idea that "code is law" also suggests problems. The legal philosopher Hart expressed this succinctly:

because of our lack of understanding of the facts, we also lack insight into the norms. Here, Hart says that reality can still surprise us. It's not just about the *unexpected* (something no one could know), but also about the *unconsidered* (something that already existed, but that no one had thought about).

Hart would have us note that whenever reality surprises us, the norm can also become problematic. A standard may still be lacking, it may be unclear whether the standard applies, or appears to apply, but the question is whether this will result in undesirable outcomes. Although the law is certainly not infinitely malleable, various solutions have been found in the course of time for those instances that reality take us by surprise. A few examples of this: hardness clauses, reasonableness and fairness, the principles of good governance, the unlawful act as an act or omission in contravention of unwritten social rules, non-limitative summaries, et cetera. But even without these structures that are specifically intended to create latitude for the unexpected or unconsidered, law remains a social practice in which the meaning of rules depends on context. Consider an instance of a judge permitting a deadline to be exceeded due to force majeure without any regulation permitting this; or a standard that does not get applied because it does not appear to serve the interest of the party appealing to the standard. Or even simpler: a park ranger who would not even think of giving the kid with the remote-control car a citation simply because vehicles are forbidden in the park. In short, the interpretation of standards is underpinned by underlying social practices or, preferably: there is a circular relationship between facts and standards – the facts determine the standards that are considered, and the standards are given content by the facts.

But does the aforementioned make the idea of smart contracts hopeless? Not in the least. It would appear likelier that one school of thought (the law as an algorithm) places the emphasis on regularities in the law, while the other school of thought notes the irregularities. Viewed in this way, it is mainly about the context in which an algorithm will generally provide satisfactory results and which safety valves must be built-in to enable human intervention, since these were already provided by the law (open standard, hardness clause, etc.), or because the result is unacceptable, for example, due to unforeseen circumstances, the particulars of the case, et cetera.

Oracles are interesting in this context because they make it possible to pause the process, which also creates latitude for human intervention. The design of smart contracts that automate the execution of an agreement or regulation in a rightful way (see Chapter 4 for an additional explanation) shall then also provide for a role for a human actor who can influence or reverse the execution of a smart contract under certain conditions, in the manner of an exceptions procedure. Consider such roles as an escrow party, mediator or judge who are part of the current way of working, for example.

# Legal manifestations of smart contracts

The crux of a smart contract is that its contents (the code) cannot be manipulated (afterwards) and that its execution cannot be prevented [18]. As previously mentioned, the term "smart contract" is an unfortunate term not only because it has no legal meaning, but also because it suggests that a legally binding contract has been created.

As will be shown from the discussion of various manifestations, smart contracts can play a role in various legal domains. When discussing these forms, we presume Dutch law, although this need not always be the case. (See also further under General legal issues: Applicable law.)

[18] Ultimately, everything can be manipulated, of course, but that would require huge efforts on a blockchain. Although not permitted a more likely approach is to have a person try to manipulate the signal that goes to the contract.

The use cases studied by the working group demonstrate that smart contracts can have manifestations that represent a legal act (see manifestations 1-4 below), or that can have meaning for the law or the legal relationship in which the smart contract is deployed (see manifestations 5-7 below). If this is the case, then it must be certain that the smart contract is programmed in such a way that the legal requirements placed on the legal act for which the smart contract provides are met, or at least the requirements placed on the law or legal relationship that the parties have. In other words, the smart contract will have to represent a legal situation and the transaction generated by the smart contract must be legal. The question can then arise: "who can and may evaluate the legality of the situation?" Our opinion is that, if an IT solution is considered in which the creation, execution or compliance with legal obligations plays a role, then it is necessary to already pay attention to all relevant legal issues in the design phase. This includes setting standards in code or pure execution, methods for dealing with issues that cannot be captured in code, specific case-specific legal requirements, and more general legal questions (liability, applicable law, jurisdiction, general principles, dispute resolution, privacy and digital identity). It is therefore good to involve a lawyer.

# The most prevalent legal manifestations

Our legal system is characterised by many legal acts – from multiple-party private law legal acts, such as contracts, to unilateral private law legal acts (such as endowments) and public law legal acts. The question is whether all those various legal acts can be "captured" in a smart contract or whether the law places such "analogue" requirements on those legal acts that they cannot be replaced by the code of a smart contract. With the working group, we have identified the most prevalent legal manifestations of smart contracts.

1. Contract and/or execution of a contract
2. Suspensive condition or dissolving condition in a contract
3. Unilateral legal act
4. Decision under public law
5. Means of evidence/evidence function
6. Automatic execution of a (legal) process
7. Obligation of compliance with (fiscal) law

We expect that (many) more legal manifestations can be identified. For this reason, the list is not intended to be exhaustive, but serves solely to shine a light on the most prevalent legal acts that are executed in smart contracts. The legal requirements for each legal manifestation are summarised below and we describe the challenges that arise when those legal acts are cast in the form of a smart contract.

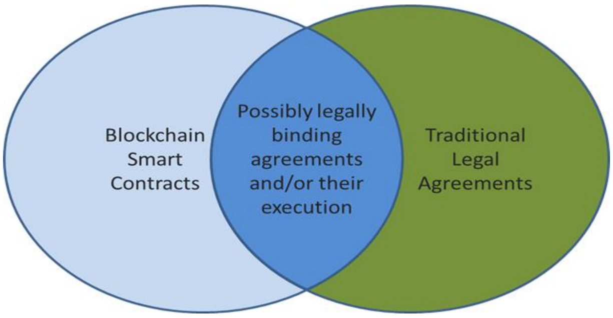
## 1. Contract and/or execution of a contract

Doing business via the Internet is not a new phenomenon. The innovative part of smart contracts is that a computer program performs certain processes automatically once the conditions for this are met. For example, the automatic payment of a web shop can be programmed into a smart contract, once the customer has digitally confirmed the delivery of the item ordered. In order to provide certainty about the "payment", the agreed amount can be "locked" into the smart contract beforehand. The address where the payment amount is saved then functions as a type of escrow account that is managed by a computer program. Naturally, this means a temporary loss of liquidity.

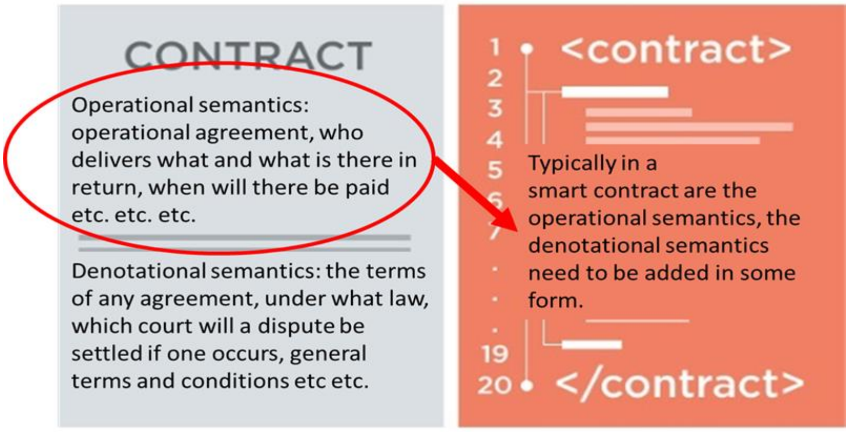
A contract is a type of agreement. According to the Dutch Civil Code, an agreement is said to exist upon offer and acceptance. In principle, the way in which an agreement comes about is not subject to a particular form. For example, an agreement can be made orally and even implicitly, namely by behaving in a particular way. One of the most important requirements for an agreement is, naturally, that it is clear to parties what has been agreed. The Dutch Civil Code therefore requires that commitments be sufficiently determinable. Smart contracts are written in a programming language such as Solidity or Go and, moreover, are often published on the blockchain in a "compiled" form, which can be read only by computers. This justifies the question of whether a smart contract can provide determinable commitments. To exclude any doubt about this, it is recommended that the commitments be described in a way that is understandable to all parties. The advantage of this is also that agreements can then be recorded that cannot be automated, or less easily so. One possible disadvantage of agreements in general language in addition to code is that these may give rise to differences in interpretation. If, and to the extent, a smart contract is indeed intended as the "pure execution" of commitments described elsewhere, then the recommendation is to record this clearly.

Should a party make a claim based on the code, then the doctrine of error could play a role, see article 6:228 of the Dutch Civil Code. Reasonableness and fairness dictates that parties must have their behaviour determined in part by the justified interests of the counterparty. Should a party make a claim based on a complicated piece of code against a non-expert party, and this former party did not reveal the precise meaning of that code, then contravention of the obligation of information can be an issue. The non-expert party could have (part of) the agreement declared null and void based on aberration or error in accordance with article 6:228 paragraph 1 under b of the Dutch Civil Code, since the expert party remained silent when it should have spoken.

In the first instance, a smart contract is just a program on a blockchain. So, there will be a collection of smart contracts whose intention is not to create an agreement. Conversely, there will also be a collection of written agreements that have nothing to do with smart contracts. In the cross-section between these two collections is a subgroup in which smart contracts are used, in any case for the automated execution of (part of) an agreement and possibly to create legal obligations as well.


Blockchain Smart Contracts
Possibly legally binding agreements and/or their execution
Traditional Legal Agreements

It is conceivable that smart contracts will be increasingly deployed in a form in which the code is inextricably linked to natural language expressions. The natural language can serve to record things that cannot be expressed in code (general conditions, applicable law, evidentiary agreement, more open standards, etc.) and possibly also to explain the intention of the code. A hybrid contract



Operational semantics: operational agreement, who delivers what and what is there in return, when will there be paid etc. etc. etc.

Denotational semantics: the terms of any agreement, under what law, which court will a dispute be settled if one occurs, general terms and conditions etc etc.

Typically in a smart contract are the operational semantics, the denotational semantics need to be added in some form.

Traditional contract

Smart contract

which combines both code (or executable data structures) and prose is sometimes called a Ricardian contract [19].

It was mentioned previously that it would be desirable to develop a standard (ontology) that expresses rights and obligations independently of any platform. One advantage of such a standard would be that everyone would understand clearly what a certain term means.

*Can a smart contract be used as a written agreement?*

The creation of agreements is not always independent of form: sometimes, the law requires that an agreement be concluded in writing. This is the case in a leasehold agreement, for example. And if the law prescribes a so-called writ, a written agreement is also required. Article 6:227a of the Dutch Civil Code describes the conditions under which an agreement that has been concluded electronically can stand in for a written agreement. This must involve an agreement for which the law does not prescribe the intervention of the court, a public body or a professional performing a public task. An agreement concluded electronically can be considered equivalent to a written agreement if:

a. it can be consulted by parties;
b. the authenticity of the agreement is safeguarded to a sufficient degree;
c. the moment of the creation of the agreement can be determined with sufficient certainty; and
d. the identity of the parties can be established with sufficient certainty.

*Re a: Ability to be consulted*

The first condition is that the electronic agreement can be consulted by the parties to the agreement. As shown by jurisprudence, the agreement must be recorded in such a way that the parties are able to access and save its contents in order to be able to *inform themselves later* about the agreement. This requirement can mean that the party that wishes to use a particular technique for this shall be obligated to make the proper technical resources available to the counterparty in order to be able to consult the contents of the agreement if that latter party does not have these resources at his disposal. As indicated previously, a contract in compiled form is legible only to computers. Can a smart contract be consulted if the source code [20] is published? Here we encounter the problem once again that not every issue is likely to lend itself to translation into code and that the need can arise for explanation of the (intended) working of the source code. We previously recommended that it should not only concern the development of code, but that the commitments should also be written down in a way that is understandable to all parties.

*Re: b Authenticity*

To a certain extent, the obligation of authenticity is easily met since a smart contract cannot be changed unilaterally. Furthermore, a change would result in a new hash value (i.e. a changed digital fingerprint). However, as mentioned, it is a question of the circumstances under which a smart contract can provide determinable commitments in and of itself. If in addition to this or as a supplement, text in regular language is used, it may be presumed that the requirement of authenticity also covers these texts.

It was not the law's intention to place more stringent requirements on this point than on paper-based agreements. For equivalence, nothing more is required than an equivalent degree of certainty about the authenticity. Correspondingly, the same applies to the moment of the creation of the agreement and the establishment of parties' identities (both topics are discussed below).

[19] http://iang.org/ricardian/

[20] "Source code" refers to the code in a programming language that is written and read by people; this, in contrast to the "compiled" form, which can only be read by computers (depending on the technology, this latter is called machine language or bytecode). Although all smart contract code is available to all participants on a blockchain, it is available in a 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 available, you can deduce the operation of the compiled form.

*Re: c Moment of creation*
On the presumption that indisputable, automatic date/time stamps are used, the moment of an agreement's creation, or at least the smart contract portion of this, can be established with sufficient certainty.
*Re: d Identity*
Although this may not always be practical, a smart contract can be made in such a way that parties remain anonymous, in the sense that parties are known only under a meaningless public key. We have previously (see “Permissioned versus permissionless blockchains”) explained why this need does not mean that the identity of a party cannot be discovered. In any case, it is clear that if a smart contract must be able to stand in for a written agreement, more information than a public key is required: after all, it must be possible to establish the identities of the parties with sufficient certainty (see also the heading "Information obligations prior to the agreement" and the paragraph on "Identity and the digital signature" below).

*Payment with Bitcoin or another crypto-currency: purchase or barter agreement?*
Smart contracts often presume payment in Bitcoin or another ‘native crypto-currency’ such as Ether. According to the district court of Overijssel, Bitcoin is not money but a means of bartering/exchange [21]. This decision could mean that, in the event of payment using Bitcoin, it does not involve a purchase but a barter agreement. Article 7:49 of the Dutch Civil Code defines barter as an agreement in which parties mutually commit to give a good to each other. Bitcoin is not a good because it is not a physical object. It would appear to be a property entitlement, because it can be valued in terms of money. The literature presumes that a property entitlement can be exchanged.

If it were indeed barter, then, according to article 7:50 of the Dutch Civil Code (a so-called linking provision), the provisions concerning purchase are applicable, on the understanding that each of the parties is considered to be a seller of the performance that it owes, and as a buyer for the part that it is due. Therefore, all the provisions concerning consumer protection in this chapter are also applicable.

The question then is whether Bitcoin concerns a property right that is *transferable* from a legal point of view. According to Dutch law, ownership, limited rights and rights of claim are transferable unless the law or the nature of the entitlement contravenes a transfer. Rights other than the ones stated are transferable only if the law expressly permits it. Given a property or a limited right is not involved, it comes down to whether Bitcoin can be considered a right of claim. If not, then the law may have to be changed in order to make Bitcoin transferable. Rank is of the opinion that this does not involve a right of claim, since a claim must be balanced with a debt. And since there is no issuer of Bitcoin, there is no debtor and therefore also no debt. He also concludes that the law should be changed in order to make Bitcoin transferable. Another approach could be that Bitcoin may well result in a right of claim, since a Bitcoin embodies a number of rights with respect to the Bitcoin community which, after all, has committed itself to a number of agreements (enforced automatically).

If Bitcoin were to be considered a right of claim, then delivery on grounds of article 3:94 of the Dutch Civil Code must take place by means of a writ. And according to article 156 of the Dutch Code of Civil Procedure, this is a *document signed* and serving as evidence. Article 156a of the Dutch Code of Civil Procedure opens up the possibility of drawing up a private writ in electronic form. The blockchain records all of the approved transactions and the associated digital signatures. It should be noted here that only the sender of the crypto-currency transaction signs, but the recipient probably signed the smart contract in which the crypto-currencies were locked up. It should also be considered that Bitcoin is a permissionless blockchain that works on the basis of pseudonyms. On close examination, this distributed ledger proves nothing more than that a certain number of Bitcoins was transferred

[21] District court of Overijssel 14 May 2014, ECLI:NL:RBOVE:2014:2667

from one public key to another. If the counterparty were to deny that the claim was transferred to him, extra evidence would therefore be needed to link him/her to the public key to which the claim was transferred. It would appear that the requirement of reproducibility of the writ is met: various programs are available that can offer a view of a blockchain. One such example is "Wallet" programs in which not only private keys are stored, but in which transactions to and from the addresses managed by the user are shown. It may be presumed that Bitcoin users are familiar with these. However, all things being considered the application of article 156a of the Dutch Code of Civil Procedure may nonetheless require a certain amount of legal flexibility.

*Obligations of information prior to the agreement*
Article 3:15d of the Dutch Civil Code places obligations on service providers in the information society. Among other things, this provision covers the purchasing of goods and services on the Internet, even if no payment is required for the services, as long as the services are offered as an economic activity[22]. Among such service providers' obligations is the requirement to provide information about his/her identity and address of establishment.

Article 6:227b of the Dutch Civil Code builds on this further. The purpose of this provision is to increase faith in e-commerce by preventing both unwanted agreements and the creation of agreements with unintended content. Article 6:227b of the Dutch Civil Code focuses in particular on agreements that are established via a website. The provision does not apply to agreements that arise via e-mail or another individualised form of electronic communication. According to the first paragraph of this provision, a service provider in the information society must provide clear, understandable and unambiguous information to the counterparty – prior to the creation of an electronic agreement – about:

a. the way in which the agreement shall arise and, particularly, which activities are required for this;
b. whether or not the agreement shall be archived after this has been created and, if the agreement is archived, the way in which the counterparty can consult this;
c. the way in which the counterparty can become informed of activities (obligations) that he does not desire, along with the way in which he can rectify these before the agreement is established;
d. the languages in which the agreement can be concluded;
e. the codes of conduct to which he has subjected himself and the way in which the counterparty can inform himself of these codes of conduct electronically.

In consumer exchanges (parties are not acting in the exercise of a profession or company), this concerns mandatory law.

According to the second paragraph, the provider must make the conditions of the agreement available to the counterparty prior to or upon the conclusion of the agreement, i.e. not those general conditions referred to in article 6:231 of the Dutch Civil Code, in such a way that the counterparty can store these conditions in a way that these are accessible to him/her for purposes of information later.

An agreement that has arisen under the influence of a service provider's non-compliance with the obligations stated in paragraph 1 in the header and under a, c or d can be nullified. If the service provider does not comply with his obligations stated in paragraph 1 and under a or c, then it is suspected that an agreement has come about under the influence of this.

Throughout the time period that the service provider has not provided the information referred to in paragraph 1 under b and e and paragraph 2, the counterparty can terminate the agreement.

[22] Dutch Parliamentary Papers II 2001/02, 28197, 3, p. 12/3.

The first paragraph of article 6:230m of the Dutch Civil Code contains a summary of the information obligations that apply to a trader if the consumer concludes an agreement remotely or outside the sales area. In these cases, the consumer is extra vulnerable and the trader must provide at least the following information to the consumer in a clear and understandable manner:

1. the primary features of the goods or services, and the degree to which this is appropriate considering the carrier and the goods or services;
2. the trader's identity, such as his trade name;
3. the geographic address at which the trader is located and the trader's phone number, fax and e-mail address, if available, and the geographic address and identity of the trader on whose account he acts, if applicable;
4. the total price of the goods or services, including all taxes or, if the nature of the good or service cannot be reasonably calculated beforehand, the way in which the price is to be calculated and, in such a case, all additional freight, delivery or postage costs and any other costs or, should these costs not be able to be reasonably calculated beforehand, the fact that such extra costs may possibly be owed. In the event of an agreement of undefined duration or of an agreement comprising a subscription, the total price includes the total costs for each invoicing period. If a fixed rate applies to such an agreement, then the total price also includes the total monthly costs. If the total cost cannot be reasonably calculated beforehand, the way in which the price must be calculated is reported;
5. the method of payment, delivery, performance and the time period within which the trader commits to delivering the goods or services and, to the extent applicable, the trader's complaint-handling policy;
6. information concerning the (existence of) right to dissolution.
7. a reminder of the existence of the legal guarantee that the item delivered must correspond with the agreement;
8. the duration of the agreement, to the extent applicable or, if the agreement is for undefined duration or tacitly renewed, the conditions for terminating the agreement;
9. if applicable, the existence and the conditions of guarantee amounts or other financial guarantees that the consumer must pay or offer at the trader's request;
10. if applicable, the functionality of digital content, including any applicable technical security facilities;
11. the possibility of access to extrajudicial complaint and dispute resolution procedures with which the trader is associated and the way in which access is gained.

These provisions illustrate how important it is that a party, and consumers in particular, is informed of exactly what is being agreed – what is to be delivered and at what price, the duration of the agreement, the method of cancellation, who the business transaction is with, et cetera. It should also be noted that the existence and the conditions of guarantee amounts or other financial guarantees must be stated: in the event of a smart contract, a guarantee is often provided by locking Bitcoins or another crypto-currency into the contract.

*Explanation of agreements*

To the extent that an agreement has arisen, the question may rise as to the *explanation* of that agreement. The crux of a smart contract is that its contents (the code) cannot be manipulated (afterwards) and that its execution cannot be prevented. If you are not careful, you can be confronted with a fait accompli. This would strongly suggest building dispute-resolution facilities into the smart contract. In addition to the smart contract, you could also structure a smart contract such that the consequences of the first contract are annulled/rectified, for example, after the intervention of a trusted third party. The question is then, indeed: what can be repaired digitally and what requires an act in the physical world?

If a smart contract is executed and the question arises as to whether the consequences were indeed intended, then "classical law" can also provide assistance. The Dutch Civil Code contains mandatory law in the first instance, which applies in any case. In the second instance, the Civil Code contains regulatory law that applies if parties do not deviate from this. Reasonableness and fairness also play a role with the explanation of agreements: they can work both in a supplemental manner about what was agreed, and in a limiting fashion. According to the well-known Haviltex criterion, one must look not only at the literal wording for the explanation of a contract, but also at the meaning that parties might reasonably assign these provisions back and forth under the given circumstances and at what they might reasonably expect of each other in that case.

The aforementioned discussed the creation and interpretation of agreements and obligations that may exist prior to the creation. Agreements can also terminate. Initially, there may be complete compliance. The agreement is then, in principle, completed (delivery and payment have taken place). It is also conceivable that parties agree to dissolution, or that the agreement is cancelled unilaterally, or that the agreement is dissolved by means of court annulment. It is also possible that an agreement was never found to have arisen. In the latter case, it cannot be said that the agreement was terminated, but it can mean that the (consequences of) certain actions must be reversed. We will not go into further detail about these aforementioned situations at this point. But it should be clear that these must also be taken into account when the smart contract is programmed.

## 2. Suspensive condition or dissolving condition

In short, to suspend means to delay. A suspensive condition is therefore a condition that delays the occurrence of a particular legal consequence. Consider an agreement with a suspensive condition, for example. The agreement comes about (therefore gaining legal force and meaning) only after the suspensive condition is met [23]. However, prior to the effectuation of the suspensive condition, there is a conditional commitment between the parties. It is important to realize that the suspensive condition does not have an independent meaning; it becomes relevant in its manifestation only if linked to another legal manifestation such as the agreement or a unilateral legal act [24].

The suspensive condition would appear to lend itself well to the concept of smart contracts. After all, this is, in fact an "if this happens, then that happens" type of application of blockchain technology. It is important that the suspensive condition of the intended legal act is established precisely beforehand so that a commitment and its starting moment are clear.

A number of questions arise surrounding this manifestation. For example: what happens if the suspensive condition is met in a different way or on a different principle than is provided for in the code of the smart contract? The law, for instance, determines that if a party had an interest in non-compliance with the suspensive condition, or if it hindered compliance, then the condition is nonetheless considered to have been met if reasonable and fairness require it [25]. Or what happens in the event that a unilaterally focused legal act is rescinded prior to the occurrence of the suspensive condition - on the presumption that rescission would be possible in the "analogue world"?

[23] See for example, District court of Leeuwarden 28 April 2014, ECLI:NL:RBLEE:2010:BM3063, r.o. 5.2.

[24] See for example, HR 31 May 2002, ECLI:NL:HR:2002:AE0745, NJ, 2002/470.

[25] Article 6:23 of the Dutch Civil Code.

It is important that these types of situations are taken into account beforehand when shaping smart contracts.

A subsequent question is: how should we deal with statutes of (time) limitation? If no action is taken by a creditor for a long period of time to collect the asset in the smart contract, the claim will lapse after a given time period. This lapsing of a claim that is due means, in fact, that the claim can no longer be legally enforced after a given deadline has passed. Every type of claim (arising from an agreement or otherwise) has its own statute of time limitation that is regulated in various places in the law. When that deadline is taken into account, a smart contract can provide a solution. One can link the payability of a claim, i.e. make the claim dependent on, a suspensive condition. Compliance with the suspensive condition then initiates the time limitation statute [26].[4]

Finally, we mention here the dissolution of an agreement under suspensive conditions before the suspensive condition takes effect. Naturally, an agreement subject to a suspensive condition can also be dissolved. On legal grounds, the counterparty of a party evincing shortcomings is able to dissolve the agreement as long as the consequences of the shortcoming justify the dissolution. After dissolution, a commitment arises for both parties to reverse the performances they have already received. If such a rewinding of the transaction is impossible, then the value is compensated instead. In such situations, we run up against the immutability of smart contracts. This can possibly be resolved by excluding the possibility for (complete) dissolution, or by taking dissolution possibilities and their consequences into account beforehand in the code.

The dissolving condition is the opposite of the suspensive condition: this makes the (conditional) commitment lapse upon the event taking place. The dissolving condition also occurs in combination with other legal manifestations. The dissolving condition would appear to be less suited to a smart contract than the suspensive condition. After all, a smart contract records exactly what will happen if a given (previously recorded) event takes place, while a conditional commitment arises in the event of a dissolving condition, such that the operation of a given legal act is made dependent on a future uncertain event [27].[5]

The greatest difficulty for a smart contract in the form of a commitment subject to dissolving conditions would appear to be in the consequences of the occurrence of the dissolving condition. After all, the occurrence of the dissolving condition results in the dissolution of the commitment. Parties to the commitment (or those involved) must be returned to the state in which they existed at the moment of the conclusion (or occurrence) of the commitment. Any performances (or other actions) made during the commitment must be undone. For example, in the event of the dissolution of the agreement, any payments made based on the agreement will be undue payments and must be refunded. If possible, such things must be carefully provided for in the coding of a smart contract.

Just as with the suspensive condition, this commitment can also be dissolved on the grounds of reasons other than the dissolving condition. Exclusion of the possibility for dissolution would appear to be a solution. The same goes for the situation in which the dissolving condition – separate from the actual dissolution based on law as mentioned above – is substantiated in a different way from what the smart contract provides for. This challenge was also encountered with the suspensive condition.

## 3. Unilateral private law legal act

As we have noted above with the contract, for example, legal acts (acts aimed at a particular legal consequence) can involve multiple parties, but they can also be unilateral. The first distinction that can be made with unilateral legal acts is between the unilateral public law legal act and the unilateral

[26] HR 23 December 2016, ECLI:NL:HR:2016:2988

[27] Article 6:21 of the Dutch Civil Code.

private law legal act. Section 4 below deals with the administrative decision, a public law legal act.

Unilateral private law legal acts can subsequently be subdivided into the directional unilateral (private law) legal acts and the non-directional unilateral legal act. As examples of (unilateral) directional legal acts, consider cancelling a real property lease agreement, say, or firing an employee. Such legal acts as these are characterised by the fact that they come about by means of a statement directed to one or more persons designated either directly or implicitly, and by the fact that they first take effect if the particular statement has reached that person or those persons. A non-directional unilateral legal act does not require permission from another person, nor is the receipt by a particular person required. One example of this is renouncing community.

It is important that the unilateral legal act has multiple forms of manifestation that are included in multiple places in the law. The various manifestations may be part of a different (legal) regime. For purposes of this study, we will attempt to keep things general.

**Unilateral directional private law legal act**
In nearly every case, a separate legal act takes place prior to a directional legal act, such as cancellation, dismissal or annulment (of a legal act).

Considering this, it would appear obvious for directional legal action to be included in the smart contract of the particular other legal act. For example, it is conceivable to describe the situation in an agreement in which cancellation or dismissal is possible and to link the directional legal act to that situation automatically (which could possibly be established using oracles). However, it is important first to search the jurisprudence for other solutions, which may also be required sometimes in the consideration of reasonableness and fairness. This must remain possible. Furthermore, it is quite likely that certain situations will be missed in which one might wish to perform the directional legal act, but in which the smart contract has not provided for this.

A subsequent challenge – separate from the suspensive and dissolving conditions that may be placed on a unilateral legal act – is the rescission of such a legal act. After all, a unilateral directional legal act only carries legal consequences once the intentionality has reached the party for whom the offer is intended. One can still withdraw a statement that has not yet been received [28]. The coding of a unilateral legal act must take this possibility into account.

The annulment of a unilateral directional legal act is an even greater challenge, considering that the annulment, in principle, is retroactive up to the moment of the performance of that legal act. Commitments then arise to annul performances that have already been performed, but which were not due. The coding of a smart contract must take this into account.

**Unilateral non-directional private law legal act**
This category includes drawing up a will, on the one hand. It is beyond the remit of this report to discuss all of the (extremely complex) legal provisions to which this manifestation is subject. But all of these legal provisions must be taken into account if a will is recorded in the form of a smart contract. In any case, it is worth mentioning that an heir can either accept or reject a will; and these two options are unilateral non-directional legal acts of themselves. I.e. a unilateral non-directional legal act on a non-directional unilateral legal act. The coding must take this into account as well.

[28] Rescission must be distinguished from revocation. Rescission keeps an offer from coming about. There was never any commitment on the part of the offering party. (Relevant only in the situation that the offer has not yet reached the counterparty.) Revocation carries legal consequences if the statement of revocation has reached the person to whom the offer was made article 3:37, paragraph 3). Revocation is possible only if the message/offer is not accepted, but can no longer be revoked if the counterparty has already sent the acceptance but this statement has not yet reached the offering party (article 6:219, paragraph 2).

On the other hand, we know that there are manifestation forms of unilateral non-directional legal acts that must not be followed by another (legal) act for their operation. In addition to the examples stated, these include: recognition of a child, offering up possessions and issuing a "403 statement" in corporate law.

These forms also have varying legal regulations that must be taken into account. If we look at the 403 statement [29], for example, then we see that this statement is just one of the conditions under which a legal entity belonging to a group is exempted from the structural requirements of the annual report. In short, that statement alone is not sufficient.

## 4. Decision under public law

Public law decisions are issued by administrative bodies (such as municipal executives, ministers and tax inspectors) and are subject to public administrative law. Administrative law is responding slowly to the possibilities of the digital era. Electronic communications between citizen and government are increasingly customary and will be supported in the near future by the Dutch Modernisation of Electronic Administrative Communications Act [30]. At this time, a message can be sent from the board electronically to one or more addresses as long as the party addressed has indicated that he is sufficiently available in this way (article 2:14, paragraph 1 of the Dutch General Administrative Law Act; consider, for example, a message indicating that a requested permit or grant has been issued or refused). This would appear to present few problems to requesters who download, for example, an app and submit a request in this manner. But it would present problems concerning decisions not granted upon request and messages to third-party stakeholders.

In turn, a citizen can send electronic messages to an administrative body as long as the administrative body has indicated that it is "open" in this manner. By making, for example, an app available, this condition would appear to be met. In addition, a request can also be signed with an electronic signature if the method used is sufficiently reliable, considering the nature and content of the electronic message and the purpose for which it is used (article 2:16, paragraph 1 of the Dutch General Administrative Law Act). However, a second considerable challenge lies in making the distinction between electronic messaging *possible* and making it mandatory. So, in this case, between working solely with, for example, an app and the underlying smart contract and also (as an option for the citizen) making the option available of working with an app and an underlying smart contract. The board cannot *demand* that citizens forward their messages electronically or only receive them electronically [31].

In addition to the fact that administrative bodies regularly communicate with citizens, they also make public law administrative decisions. A public law decision is a written decision of an *administrative body,* containing a public law legal act (article 1:3, paragraph 1 of the Dutch General Administrative Law Act).

[29] A written statement that the consolidating legal entity files with the trade registry of the Dutch Chamber of Commerce in which it states its joint and several liability for the debts of the exempted legal entities (subsidiaries).

[30] The Dutch law concerning the modernisation of electronic administrative communications changes that portion of the Dutch General Administrative Law Act that relates to electronic administrative communications. The legislative proposal is currently submitted to the Council of State's advisory committee. If this legislative proposal is accepted, then this has two concrete consequences for administrative bodies: 1) an obligation to open digital channels for any formal electronic message sent to the administrative body; 2) an obligation to adapt digital channels in such a way that legal requirements are met (only request necessary information, sending confirmations of receipt, making electronic forms available, burden of proof for administrative body, notification of refusal of wrongly addressed message).

[31] Unless this is regulated by law, such as the obligation of companies to submit tax returns digitally. Based on the law, individuals can choose to file a paper tax return, use a tax return diskette or may file directly via Internet.

There are two main categories of decisions: orders/decrees for an individual or concrete instance (article 1.3, paragraph 2 of the Dutch General Administrative Law Act) and decisions of a more general intent (such as generally binding provisions, policy regulations and plans). This concerns the first category: orders.

An *individual* order is aimed at one or more stakeholders (generally, but not always, the requesting party). A *concrete* order is not aimed at one or more stakeholders (but concerns a concrete instance or object such as an object designated as a monument). An order focuses on establishing in a *binding* manner or *creating* or *terminating* a legal involvement (a relationship with a legal meaning). As to the latter, examples are: 1) permits, waivers, releases and such, in which something is permitted that would otherwise be forbidden; 2) the provision of status in which a particular legal regimen applies to someone or something; 3) issuing grants or other governmental performances such as benefits, loans, scholarships and guarantees; 4) recommending that something be done, omitted or clarified; 5) approval, quittance or annulment of a decision of a decentralised administrative body by another administrative body.

We call an order "*bound*" to the extent that the results and content are determined by the underlying legal regulation; it is *free* to the extent that the administrative body assigns evaluation or policy latitude to the legal regulation. The bound order lends itself well to the concept of smart contracts since there is little latitude between the regulation and the application in a concrete instance. Unfortunately, orders with a completely bound nature are rare; orders are often mixed, i.e. partially bound and partially free. This presents the greatest challenge: the greater the board's freedom of evaluation – particularly – in individual cases, the less predictable the results of the administrative body's evaluation. This makes the use of the smart contract more difficult.

The question then arises: how much freedom of evaluation may the administrative body adopt via a smart contract – beforehand, separate from the individual case – and how much room is there then to do justice to individual interests and circumstances? In cases in which the results of the administrative consideration or evaluation cannot be predicted beforehand, oracles can be used to get the results of administrative evaluation formation into the smart contract. Oracles can also be used to include third-party judgements and recommendations (either mandatory or those considered necessary). Possibly – if mandatory or desirable – after the administrative body has carefully inspected the creation or has involved the recommendation in its own further assessment. Beyond this, oracles can be used to include information from their own and external data sets, sources and registers directly. As long as the administrative body is required to base itself on this, there would appear to be no problem. If a legal requirement is missing and if the data set, source or register involved is not managed by the board itself, then we may encounter complications. In addition to working with oracles, possible complications can be countered by chopping up the decision-making process into two parts (a pre-study, for example, request process, decision-making), or by using a smart contract only to streamline and automate the internal process (in which input is entered manually, and/or processed using own verified data sets, for example).

All of the activities of administrative bodies – including those activities in the context of decision-making – must be in accordance with the law, of course. This also includes the general principles of proper governance. In that context as well, working with smart contracts can present both opportunities and challenges. This is discussed below with a more extensive treatment of the general principles of proper governance.

## 5. Means of evidence/function

On the basis of article 152 of the Dutch Code of Civil Procedure, proof can be provided by any means not forbidden by law. On the basis of article 156 of the Dutch Code of Civil Procedure, writs are signed copies intended to serve as proof. Further, the law makes a distinction between authentic and private writs [32].

With the introduction in 2010 of the Dutch Documentation and Electronic Legal Transactions Act [33], the electronic private writ was made equivalent to the written private writ. Article 156a, paragraph 1 of the Dutch Code of Civil Procedure states namely that it is possible to draw up a private writ in a manner other than in writing. An electronic document with a digital signature (requirements placed on a writ) therefore qualifies as a private electronic writ. However, if the legal requirement exists to provide a private writ, then that can only be done in another manner with the express permission of the party to whom the writ must be provided (article 156a, paragraph 2 Dutch Code of Civil Procedure). In addition, on the basis of the provisions in article 6:227a of the Dutch Civil Code, the requirements included in that article must be met:

1. it must be possible for all parties to consult the writ;
2. the authenticity must be sufficiently safeguarded; this means that the content of the writ has not been and cannot be manipulated;
3. it must be possible to establish with certainty when exactly the writ was created;
4. it must be possible to establish with certainty the identity of the parties.

If the aforementioned requirements are met, then it can be said that the requirement of writing is met in any case and the private electronic writ provides the required material burden of proof between parties, just as the written private writ [34].

Furthermore, parties can conclude a mutual evidentiary agreement. In this, parties can agree to deviate from legal evidentiary rights, who (in that case) must prove what and what evidential strength is assigned to electronic particulars. Deviation from legal evidentiary rights may take place only under certain circumstances. For example, an evidentiary agreement will not be accepted if the Dutch Civil Code states this, along with when it relates to the evidence for facts to which the law attaches consequences, which the parties are not free to determine themselves (article 153 of the Dutch Code of Civil Procedure).

The question that now rises is whether that which is included in a blockchain or arises from a smart contract complies with the aforementioned principles and (therefore) can be considered compelling evidence, and/or that parties can agree to accept whatever is included in a blockchain and/or whatever arises from a smart contract as compelling evidence. In general, it can be noted that hindrances arise from the use of blockchain technology as such, or from smart contracts, that prevent compliance with the legal requirements. So, it is indeed a requirement that the identity of the parties can be established with sufficient certainty, along with the fact that an evidentiary agreement expresses the will of the parties. Here, too, the greatest challenge lies in the way in which the agreement of wills between parties can be demonstrated. A determination must be made as to the way in which it is clear to the other party that he/she should reasonably understand the statement sent to him/her as an associated intention of will.

[32] One example of an authentic writ is a notarial, for example, or a writ from a civil servant official. Private writs are writs that are not authentic.

[33] Dutch Parliamentary Papers II 2007-2008, 31 358, no. 3.

[34] This applies only with respect to private writs. For the service of a notarial writ by electronic means, the law must be changed with respect to various points.

## 6. Automatic execution of a (legal) process

With this manifestation, it concerns complying with or performing processes on which the law places requirements, with the assistance of a smart contract. On the governmental side, supervisory or investigation processes can be considered, or the procedure for administering an administrative penalty, for example (Section 5.4.2 of the Dutch General Administrative Law Act). For this, the process must comply with certain legal requirements and certain (sequential) mandatory steps must be taken (before proceeding to enforcement, the publication of the report or the issuance of an administrative penalty, respectively).

As noted previously, all of an administrative body's activities must be in accordance with law, including the general principles of proper governance. So this applies to both factual acts and to acting in the context of decision-making. Supervision and investigations are domains with considerable actual activity (data is demanded, collected and analysed, locations are visited, etc.). Section 5.2 of the Dutch General Administrative Law Act adds two specific standards for supervisors' activities to the general standards for administrative activity: the obligation of legitimation (article 5:12 of the Dutch General Administrative Law Act) and the standard of material carefulness in article 5:13 of the Dutch General Administrative Law Act (supervision may not extend further than is necessary).

The greatest challenge faced here is in the way in which physical/analogue reality and the smart contract are conjoined. Going through legal process steps will often remain a physical procedure. It appears that physical processes cannot be coded. The way in which proof is provided in a smart contract, and with which resources, has yet to be determined. Working with oracles could offer possibilities here. Involving such factual activities in the execution of smart contracts places two challenges before us:

1) How can a smart contract safeguard the accuracy of substantive input based on factual activities? The smart contract is based on the input, not on the actual situation (if this turns out to be otherwise).
2) How can we safeguard that the actual activity is performed in the proper manner? Wherever this is not the case, this could result in collected data either partially or entirely not being permitted to be involved in the rest of the process.

From the aspect of the citizen, consider notification systems that have increasingly replaced permit systems in recent years. Certain activities are then not forbidden except with a permit; they are, in principle, permitted if the generally provided rules are taken into account or if the requirements for permissibility are met. One of those rules or requirements is then the obligation to report the intention beforehand. In some notification systems, the notification of the intention is insufficient: data or documents must also be submitted or an indication must be given as to how certain focal points will be taken into account.

Sometimes the receipt of the notification is sufficient, sometimes a response with confirmation of receipt or acceptance must be made. And sometimes additional requirements or bespoke provisions can be set or the reported activity can be forbidden for a limited number of reasons, possibly within a given deadline. If an activity is still forbidden, or if provisions or limitations are linked to the report, then this involves a public law decision. If the deadline for doing this lapses without response from the administrative body, then legal permission is said to exist. Objection and appeal are possible in both situations and the comments above about public law decisions apply.

Regardless of whether it is the turn of the government or administrative body, the actual, rightful compliance with the legal requirements can be "recorded" in a blockchain (thereby substantiating the principle of transparency). Authorisations can be used to demonstrate compliance with "Chinese walls" (in view of the ban on prejudicial bias) and professionalism (regarding the application of the formal principle of carefulness) [35]. The sequence of the legal steps can be programmed into a smart contract – by indicating that a phase of hearing and rebuttal follows an investigative phase, for example. The result or the consequence of the various phases can be brought in via oracles (that link to data sets, for example, or that process "manual" input) and can be registered in the blockchain on which the smart contract runs. If the decision must be made at the end of such a process, then the previous remarks about public law decisions apply.

## 7. Obligation of compliance with (fiscal) law

It is expected that the payment of taxes can largely be automated through the use of smart contracts.

For example, it should be possible to link a consequence automatically to the tax consequences that parties (and their advisers) link to a transaction – i.e. the payment of (e.g.) the VAT (and/or transfer tax) due and that parties describe as such. It could even be established, for international transactions (if sufficient parameters are entered properly) in which country taxes are due, along with the guiding principle and the tax rate.

One large challenge is the situation in which the nature of the transaction determines the amount of the tax due and in which that nature cannot be derived from an external database. In such situations, tax authorities will not want to rely on the estimates of those taxes due and will want to retain the authority to charge additional taxes. In such a situation, the smart contract and the blockchain cannot be used as a basis to irrevocably establish the tax that is due.

Let's consider Dutch real estate as an example. When buying or selling real estate in the Netherlands, VAT and/or transfer tax can be due depending on the nature of the property. At this moment, there is no objective source showing whether such an object can be seen as a building site, for example, a "renovated construction" building or as a residence. As for the latter, the municipal permission (that can be verified via the website ruimtelijkeplannen.nl) expressly provides only an indication of the possible qualification as a residence. These types of qualification discussions occur with all types of transactions and with the combined delivery of various goods and/or services for which deviating rates apply, for example. This can (also) possibly be resolved with the use of oracles, with the authorised body (outside the smart contract) providing an evaluation in the event of qualification issues.
At the same time, it is quite conceivable that goods and services that can indeed be qualified can be registered in a blockchain and that the payment of what is due is regulated by a smart contract. The possibility for additional charges at a later date must also be built in then too.

It is important to notice that many different taxes exist, each with separate legal provisions. Considering the huge diversity in terms of requirements, we will not discuss those requirements exhaustively. It should be emphasised that the specific requirements for the relevant taxes must be taken into account when structuring the smart contract. Furthermore, taxes are public law decisions for which the principles sketched above apply. Appeal is open in such cases, and this must be regulated.

# General legal issues

In addition to the manifestations, general legal issues also exist concerning smart contracts. Consider, for example, liability, applicable law, jurisdiction, general principles, dispute resolution, privacy and identity.

[35] The principles stated here and other general principles of proper governance are discussed in more detail below.

## Liability

The question of the exact meaning of liability is a question that is answered primarily in the domain of private law. One speaks of (legal) liability in the event of an illegal act or breach that is attributable to a person (or company). In that case, that person is liable and he/she must compensate for the damage arising from this.

In the event of default, liability concerns the damage arising as a consequence of non-compliance (or improper) compliance with a contract. In principle, in the event of an illegal act, there is not a commitment, but damage is "simply" caused by a person's act or omission. Someone can be liable for an illegal act that he committed himself (culpable liability). One speaks then of being "at fault". One can also be liable for the deeds of another person: one then speaks of "qualitative" liability or risk liability. For example, a parent is liable for damages that his child (younger than 14) has caused. Another (private law) form of liability concerns product liability or professional liability, for example. Although we can certainly think up more legal challenges that result in (private law) liability, we will keep it here to a couple of prominent ones.

The first challenge is the fact that blockchain technology makes it possible to act under a pseudonym. The chapters on Contracts has already explained that – in cases in which the law requires a written agreement – an electronic agreement can qualify as being a written agreement only if the identity of the parties can be established with sufficient certainty. However, the question remains as to whether the cryptographic signature of a party who interacts with a smart contract does indeed safeguard sufficient certainty in this regard. After all, that cryptographic signature safeguards the fact that this party can and may appeal to the smart contract, but this does not mean that the identity of that party can also be established with certainty (see the subject Digital Identity for more on this).

Assigning liability in the event of an unlawful act is at least as complicated. For example, if the code of the smart contract is hacked by an unknown party. The same applies here, too: how do you hold someone liable if you don't know who it is? Although this situation may be easier to resolve than the "anonymity problem", the second challenge concerning smart contract liability lies in the fact that many relationships surround the smart contract and that there are still very few best practices. In the chapter on Applicable Law, the following relationships were mentioned: (i) the person who organises coding of the smart contract; (ii) the programmer; (iii) in some cases, the party providing input for the smart contract; and (iv) in some cases, the "beneficiary" of the output of the smart contract.

We can also imagine that an external party is involved in a smart contract that functions as a "guard" for a smart contract's output, for example. However, because this appears to clash so much with the basic philosophy of blockchain technology (namely, the removal of the necessity of so-called trusted third parties), we have purposely not addressed this here.

When is someone liable with respect to a smart contract? The parties who have a smart contract coded could conceivably hold each other liable in the event of breach. But when does this occur? After all, what happens on the basis of a given input is agreed to exactly beforehand – it is recorded in a code. Certainly when the input (activation) is dependent on an oracle (and not on a person), establishing when a breach can be said to exist is complicated. If the input depends on a natural (or legal) entity, one could imagine that the agreed input is not delivered or is improperly delivered and that the elaboration of the smart contract is different from what was agreed to (or not!). This brings us to the following consideration: breach on the part of a party to a contract entitles the other party to (i) compliance with compensation of damages; (ii) replacement compensation; (iii) suspension of the obligations; and (iv) dissolution of the agreement. The solutions related to compensation of damages can be regulated outside the smart contract. As we stated earlier, suspension must be possible on the basis of the code of the smart contract. If this has not been taken into account during the coding of a smart contract, then this is not possible. Also in the event of dissolution we see a challenge that we encountered earlier: parties (or those involved) to the commitment must be brought back to the state in which they existed at the moment of the conclusion (or occurrence) of the commitment. However, the execution of a smart contract cannot be prevented.

A "programmer's" liability would appear easier to establish: this party acts on the basis of an assignment. If he does not perform the assignment, or not in the manner that parties have agreed, then he is liable for the damages arising as a consequence. This liability also lies outside the smart contract.

Finally, this brings us to the insurability of liability in connection with smart contracts. Generally, you can insure yourself against many types of liability claims. But since no best practices exist yet for smart contracts, it remains questionable as to whether it applies here. Do insurance companies see a future in insuring liability risks in connection with smart contracts? And, if so, under which conditions?

## Applicable law

One important question frequently asked concerns the law applicable to smart contracts. And here, applicable law means: "which country's law applies?". On the presumption that this concerns Dutch international private law, this question arises only if the jurisdictional choice has not been made beforehand.

To achieve clarity about applicable law, a number of steps must always be taken:

1. Which legal manifestation is involved?
2. What are the nationalities of the parties involved?
3. To which (international) regulations (treaty, acts, etc.) is the commitment between the parties subject?
4. Which national law is designated as the applicable law by international regulations in the specific case?

We saw previously that a smart contract is initially difficult to accommodate in a (legal) definition and, further, that the concept – if there were indeed a legal manifestation (or an overlap with one) – cannot simply be made equivalent to an agreement (contract!). This makes the question of which legal system applies to the relationship between parties more difficult, since each legal manifestation has its own regimen [36] for handling that question. Subsequently, with respect to smart contracts, multiple relationships exist: (i) the person who organising coding of the smart contract; (ii) the programmer; (iii) in some cases, the party providing input for the smart contract; and (iv) in some cases, the "beneficiary" of the output of the smart contract. A different legal system may be applicable in all of those relationships. From a technical perspective, the various persons mentioned may also be embodied in a single person. Taking a look at the legal manifestations that we have identified in the study for this report, we can state that, for each manifestation, at least one person is involved with an expression of will aimed at a legal act. Finally, as far as we can tell, there is the complicating factor that all smart contract activities are in fact performed by or using nodes. The location of the node(s) involved and the domicile of the person involved with an expression of will and that legal act need not always be one and the same.

[36] Laws and regulations.

## Jurisdiction – international[37]

Jurisdiction is another word for legal power and relates to the area over which a governmental body has authority. The legislative, executive and judicial powers all have their own specific jurisdiction. In this report, we discuss only the judicial powers: courts.

Should it come to a dispute concerning a smart contract, then the question that follows the question of applicable law is: which court is competent? Here it is also true[38] that this question will only arise if no choice of jurisdiction has been made, i.e., a choice is made beforehand for the competent court (or other judicial body such as an arbitration institute).
In principle, to get an answer to the question of which court is competent, the same steps must be taken as with the question about applicable law:

1. Which legal manifestation is involved?
2. What are the nationalities of the parties involved?
3. To which (international) regulations (treaty, acts, etc.) is the commitment between the parties subject?
4. Which court has been designated as the competent court by international regulations in the specific case?

Just like the question about applicable law, the first problem, or challenge, with the question of the competent court is, that various legal manifestations can have a different regimen concerning this. If we look at a relationship between parties, each of whom has his own domicile[39] in a differing member state within the EU, for example, then the EEX regulation[40] generally applies. However, this regulation applies only to civil and trade cases. Furthermore, article 1 of the regulation states explicitly that this does not apply to issues involving taxes, customs or administrative law cases. So the legal manifestation in which the smart contract is cast must be clear (and whether this has been done). This still does not resolve the challenge: in such a case, the result can lead to the competence of multiple courts. In that case, one can select the court before which the dispute will be heard.

The second challenge was also mentioned previously: the location of the node(s) involved and the location (or domicile) of the parties involved with the smart contract, which need not be the same. However, the question is whether this will lead to a problem. After all, one arrives at the question of which court is competent to hear a dispute only once the (legal) entity is known against whom one initiates proceedings. Furthermore, in that case, one will want to initiate proceedings and take measures (and, to date, be able to do so only) against a (legal) entity and not against a node or – more broadly – a system. For this reason, the question sometimes arises as to whether a system as such should not be able to participate in legal relationships and/or to have an independent position. Currently, this is not the case and not legally possible.

## General principles of proper governance

As noted previously, all of an administrative body's activities must be in accordance with the law and "the law" must also include the general principles of proper governance. If we focus more specifically on decision-making processes, then we can distinguish between formal and material principles.

[37] This chapter discusses the question of the court that is appointed from an international perspective. The question of which court is competent (absolute competence) and the method of proceedings are not discussed here. This question can also prompt challenges.

[38] Presuming Dutch international private law.

[39] With regard to a natural person, domicile means "the town or city he lives in" and, in the absence of that city, the place of his actual accommodation. With respect to a legal entity, it means there where he is established in accordance with legal provisions or according to his articles of incorporation or regulations.

[40] The Council Regulation (EC) no. 44/2001 of 22 December 2000 concerning judicial competence, the recognition and enforcement of decisions in civil and trade cases.

The formal principles relate to (a) the preparation and (b) the decision-making and structuring of decisions. Material principles concern the content of decisions, which is determinant for the decision's legal consequence. The general principles of proper governance are partially codified in the Dutch General Administrative Law Act and partially grounded in unwritten law. In the context of the applicability of the deployment of smart contracts, we consider the following points deserving of special attention:

**Principle of transparency (article 3:2 of the Dutch General Administrative Law Act (partially)):** Transparency means that stakeholders can avail themselves of the information they require in order to assess whether they are being treated fairly and in order to make the right choices. No favouritism or unfair competition may occur. The decision-making must also be transparent with respect to both the procedure and the justification. The smart contract, the code, will therefore have to be "explained" to the parties involved in any accessible manner. In our view, this applies to all smart contracts representing a legal act. See the heading below under "Other focal points".

**(Formal) principle of carefulness (article 3:2, Sections 3.3-3.5, 4.1.1 and 4.1.2 of the Dutch General Administrative Law Act):** When preparing a decision, an administrative body must gather the necessary knowledge concerning the relevant facts and the interests to be balanced (article 3:2 of the Dutch General Administrative Law Act). How can a smart contract determine this at the individual level? Do we have to rely on the use of oracles here as well? The administrative body is subject to an obligation of investigation. A citizen stakeholder can be expected to provide certain information only if there is a special reason for this. If it concerns a decision made upon request, then the stakeholder may be expected to provide the board with the relevant information at his disposal. If the administrative body is lacking the data for processing or doubts its accuracy, then it must contact the requester. The administrative body must collect any information that it can collect more easily than the requester can.

The nature of the request or the decision to be made can mean the administrative body must gather recommendations or organise participatory proceedings, even if no specific legal obligation for this exists. If the decision is based on investigation of facts and behaviours performed by an adviser, then the administrative body must itself ensure that this investigation has taken place in a careful manner (article 3:9 of the Dutch General Administrative Law Act). Similar to this, a comparable obligation may also apply to all data from oracles (i.e. wherever this involves data sets and registers used, as well as third-party opinions and findings that are "manually" entered).

**Defence principle (articles 4:7 and 4:8 of the Dutch General Administrative Law Act (partially)):** The hearing obligations in articles 4:7 and 4:8 of the Dutch General Administrative Law Act give the stakeholder the option of expressing their opinion either in writing or orally. So an oracle may have to be created in order to process the opinion (heard) in the smart contract.

**Obligation to consider interests; principle of speciality (article 3:4, paragraph 1 of the Dutch General Administrative Law Act):** The administrative body must involve *all relevant interests directly* in the consideration (to the extent a limitation arises from a legal provision or from the nature of the authority to be exercised). Can a smart contract make such a distinction? Is "manual" monitoring otherwise always necessary because of the board? Must one fall back on the use of oracles here too?

Concerning the admissibility of a regulated act, only the interests of the requester and the interests that the legal regulation will serve may be considered. Concerning the provisions to be linked to the decision, the interests of other stakeholders must also be considered. The question arises: can this be captured by code in advance?

Obligation of effective and published justification (articles 3:46-3:48 and 3:50 of the Dutch General Administrative Law Act): Every decision must be based on proper justification (article 3:46 of the Dutch General Administrative Law Act). As a principal rule, these are stated with the publication of the decision, along with a statement of the legal basis – if possible (article 3:47, paragraphs 1 and 2 of the Dutch General Administrative Law Act). The statement can – unless this is still requested – remain absent if it *can be reasonably assumed that there is no need for this*[41]. In our opinion, a justification that (for example) the smart contract presumes that a permit is granted or indeed refused is not sufficient.

Principle of faith: The faith engendered by the administrative body and that has resulted in justified expectations may not be betrayed. What does this mean if there is an error in the code? Or a difference between what follows from the code and what was otherwise presented to the parties?

Principle of equality: Equivalent cases must be treated in the same way; non-equivalent cases to be treated unequally in accordance with the degree to which they differ. Suitable and necessary distinctions may be applied. Can a smart contract make this distinction or is reliance on oracles once again the only way out?

Material carefulness and the principle of proportionality (article 3:4, paragraph 2 of the Dutch General Administrative Law Act): The burdens resulting for a person arising from a decision may not be heavier than is strictly required. They may not be disproportionately heavy in comparison with the objectives that the decision wishes to serve. A decision made in the general interests may not present heavy burdens to just one or some of the stakeholders. Can this all be contained in code?

### Evaluation framework of the administrative court

Separately or in mutual correlation, the aforementioned principles are used by (administrative) courts as principles for evaluating automated decision-making processes that are contained in smart contracts. In its decision of 17 May 2017, the highest administrative court – the Administrative Jurisdiction Division of the Council of State[42] – found that in the situation of an automated decision-making process, a lack of insight into the choices made and the data and presumptions used can result in unequal positions for the parties in the process. This will come about if citizens cannot check the basis on which a decision is made. From their perspective, automated decision-making based in a program can then be considered a black box. To prevent this, in those cases the government is obliged at its own initiative to fully publish in good time the choices made and the data and presumptions used so that these are accessible to the citizens. This must also be done appropriately, according to the Administrative Jurisdiction Division of the Council of State; simply publishing the code of the smart contract is therefore not sufficient. It may be presumed that the government must translate the code for the citizen into an understandable language. If the government cannot comply with this, then all of the parties, including the citizens may evaluate the choice made and the data and presumptions used (or have this be evaluated) and,

[41] In such cases (in which something "is permitted, if" is linked to the criterion dependent on a (subjective) evaluation of an individual case) it would not appear possible to program this beforehand into a smart contract. One could then rely on the use of oracles. One can circumvent this in certain cases (in which "is permitted, if", but is always permitted in any case) simply by always doing so. Particularly if this can be automated into the smart contract via programming and it brings no extra associated executive burdens (it may even possibly reduce them then). Furthermore, in certain cases, capabilities can also be built into the app or the website with which requesters and such come into contact with the smart contract (indirectly, e.g., by building in an indication via a checkbox as to whether additional explanation is required).

[42] ABRvS 17 May 2017, ECLI:NL:RVS:2017:1259, r.o.14.2 ev. See previous ABRvS 16 September 2015, ECLI:NL:RVS:2015:2938 and ABRvS 7 September 2016, ECLI:NL:RVS:2016:2415

if necessary, dispute this with reasons. In the view of the administrative court, this is the only way that true legal protection is possible.

## Dispute Resolution

In the event of a dispute about the correct execution of a contract or other legal agreement, there are multiple forms of dispute resolution available, such as the decision of a court or mediator.

The same holds true for a dispute between parties that have a legal agreement in the form of a smart contract, with the difference that smart contracts can offer extra functionality that substantially simplifies the signalling of a dispute. In addition, if value is transferred via a smart contract, it can be regulated that there is always a guarantee of value transfer, or that a refund of the value takes place, since the party cannot destroy the value in the meantime. In those cases, the smart contract can be compared with an escrow account or third-party bank account, in which value can be released if both parties indicate via a message (i.e. a voting mechanism) that the agreements for the final value transfer have been met. If one does not wish to provide such a voting mechanism – in which the parties' opinions exclusively lead the transaction – then an alternative is that the parties involved can designate an oracle that determines whether the transaction's requirements have been met. One could agree, for example, that if a database for a weather service indicates a storm at a given location and moment, then automatic payment of an insurance amount proceeds, instead of meeting to determine whether there was actually a storm. When drawing up the smart contract, one agrees beforehand to the status of the agreed oracle as either refutable presumption or, if so agreed in an evidentiary agreement, as binding proof.

In the absence of consensus, resolution can be provided, for example by building in a signalling function with which parties can present their dispute immediately to a third party. That third party can offer mediation or a binding decision, for example. In the period of time during which the conflict is unresolved, the value can remain in the smart contract. It is conceivable that the resolving body, such as a mediator or court, would receive the authority to determine the party in the smart contract to whom the included value is (re-) granted. In the event of a protracted conflict, this does have the consequence that parties have no access to the value in the smart contract.

It is therefore important that the resolution of conflicts is agreed beforehand: who will take the role of mediator or dispute-resolver and with what authority? Clear agreements when drawing up a smart contract are therefore extremely desirable.

## Privacy

Privacy concerns the protection of personal information. Personal information is data that is either directly or indirectly traceable to a living natural person. On the basis of the Dutch Personal Data Protection Act (now) and the General Data Protection Regulation (as of May 2018), citizens have various rights with respect to their personal information, including the right to correction of that personal data, its removal and the right to be forgotten (GDPR).

Personal information may be processed in smart contracts. In that case, as a result of applicable laws, qualification issues will initially arise. For example, the law makes a distinction between a “data controller”(the party ultimately responsible) and a “data processor” (who works on behalf of the data controller)[43]. Differing legal requirements apply to the data controller and to the processor. We can presume, for example, that all participants in/users of blockchains and smart contracts in which personal data is exchanged are *data controllers* and must comply (independently) with (all) legal requirements. It is less clear whether the other parties participating in the blockchain (i.e. all parties who run nodes) also fulfil a particular status based on the Dutch Personal Data Protection Act.

[43] https://en.wikipedia.org/wiki/General_Data_Protection_Regulation#Scope

We can imagine that those nodes must be considered to be processors of personal data. If this is the case, then they too must satisfy the principles of the Dutch Personal Data Protection Act and must conclude processing agreements with the data controllers, for example.

Just as we have seen a number of times before, a distinction needs to be made between permissionless and permissioned blockchains for compliance with the requirements in the area of privacy law. In that latter variant, it is quite possible to exercise influence on the governance of the blockchain and (among other things) to regulate who is responsible for compliance with the Dutch Personal Data Protection Act requirements. Accordingly in these cases it can be determined by who and in which manner citizens' rights are safeguarded, such as the right to correction. When starting up the blockchain, agreements about this can be made among the participants. This is different for permissionless blockchains. There, both no one and everyone is in control at the same time; and these types of agreements are much more difficult to make due to the free-access possibility and lack of control of governance. The possibility for protecting privacy in such situations will have to be investigated further.

## Digital Identity

In order to give smart contracts meaning in the legal world, there must be a reliable system of digital identification (of both natural and legal entities) and of authorisation. At the same time, blockchain can itself be a platform for recording and anchoring the identity and authorisation of persons.

To guarantee reliability, it is both desirable and necessary to link (the physical manifestation of) a person (inseparably) to a digital identity, and to record that link reliably and to (be able to) audit the requirements for each transaction. This requires a constant match between (the physical manifestation of) a person and that person's digital identity. Among other methods, this could be achieved by enriching a person's digital identity with his biometric data, or to use his biometric data to obtain access to digital systems.

Currently, citizens in the Netherlands do not yet have such a digital identity. The current means of identification and authorisation in digital systems is limited to entering and checking a digital access proof *without* a constant state-of-the-art link taking place with (the visible manifestation of) the person who holds the digital proof of access. Because of this, it cannot be determined that the holder of the digital proof of access is the person to whom the digital proof of access is issued, or that the holder of the digital proof of access is actually the person who is authorised to view or affect data in a smart contract.

The Dutch Digital Delta is currently working on standardised, interoperable and broadly applicable blockchain solutions (in line of action 1) for the identification of persons, legal entities and objects. The progress of this line of action should be continued to be monitored.

## Preliminary conclusions

Based on the above, the working group has arrived at the following preliminary conclusions:

1. A blockchain smart contract is, in the first place, a deterministic computer program that is replicated and executed on a blockchain.
2. A smart contract may have legal significance, but this is not necessarily so.
3. A smart contract can be put in place in various legal domains (private law, administrative law, criminal law) and can therefore have various manifestations.
4. Not every legal manifestation (statutory provision, obligation, etc.) lends itself to being converted into code.
5. Where conversion into code is possible, it is advisable only to do this to <u>execute</u> the recognisable manifestation. In administrative law and criminal law –

where rights and duties are established - this would seem to be the appropriate way forward on the grounds of legal certainty, but it may also be required in private law, to protect consumers.

6. When the parties intend the code to create an obligation in a private law manifestation and possibly also to accept the outcome in advance, this intention should at least be laid down in writing (i.e. not in code, but in an agreement, for example). This too could be done through the blockchain.
7. The actual and legal possibilities must always be considered in advance in order to (a) link the automatic execution of the contract to pre-determined terms and conditions (such as permission of the parties or a third party) and (b) 'nullify' the execution (or its consequences) in retrospect (return to the former situation, compensation, damages, etc.). Attention must also be paid to the applicable law and the competent authority (mediator, arbitrator, court, etc.) in the event of a dispute.
8. A clear distinction should always be made between *permissioned* and *permissionless* blockchain, since their governance models may be different. A permissioned blockchain can be protected by an *access control layer*. In contrast to a permissionless blockchain, not everyone can participate. Approval in advance is required. Furthermore, read and write access rights may differ for users, which also means that tasks and responsibilities can be divided up. In short, there is an organisation, frequently an alliance, behind a permissioned blockchain.
9. Personal data may be incorporated into smart contracts. Personal data are data that are directly or indirectly traceable to a natural living person. Citizens have the right to have their personal data protected (under the Dutch Personal Data Protection Act and the General Data Protection Regulation). In the case of a permissioned blockchain, it is possible to arrange who is responsible for complying with the requirements of the Dutch Personal Data Protection Act. The arrangements are different in a permissionless blockchain. No one and everyone is in charge of a permissionless blockchain and agreements of that kind are much more difficult to make, due to the lack of restrictions on access and lack of control over governance. The possibility of protecting privacy in such situations will have to be investigated.

Evaluating the manifestations against the law results in the preliminary conclusion that major changes in laws and regulations would not appear necessary in order to deploy smart contracts in the legal order. Nonetheless, several questions have been identified. Close attention to the particular regulations applicable to the smart contract in a specific case is also required, and to general legal issues.

# Blockchain: legal aspects in the long term

Blockchain technology and smart contracts are relatively new phenomena. Therefore, it is inevitable and correct that this study of the legal aspects of the technology emphasises the possible legal hindrances to its development. But it is also a good idea to consider – even briefly – a greater question: imagine that blockchain and smart contracts live up to their presumed ideals and that the world changes in the way the believers say it will. What would the consequences then be for laws and regulations and for legal practice?

Lawyers who had to advise about the legal impact of the World Wide Web, shortly after its birth in the early 1990s, discussed domain name disputes, for example, or copyright infringements or the liability of intermediaries. These were all important topics. But now, more than 20 years later, we can see that the (indirect) influence on laws and regulations has been much greater. Under the influence of the web, regulations about protection of personal information, financial services (particularly payments) and competition have changed significantly. Without the web, other regulations would not even exist, or certainly not in their current form. Consider regulations concerning electronic trade, for example, or electronic identities and confidential services.

When considering the long-term impact of blockchain, two important features play a role: blockchain involves recording information and creating faith in that recording. Our current laws and regulations have a wide variety of obligations concerning the recording of data. Many other legal provisions have the objective of increasing faith in that recording. Many provisions and agreements also ensure this.

One example of the obligation for oneself to record data is the bookkeeping obligation for legal entities (article 2:10 of the Dutch Civil Code). Other obligations exist to record data at third parties. This often involves third parties pursuant to or designated by law. Consider the obligation of an enterprise to register in the trade registry, for example (at the Chamber of Commerce, based on the trade registry law). The fact that the information is reported by a central authority designated by the government is itself a way of increasing faith in the recording. But there are also other ways. For example, confidence in an annual report to be drawn up by a legal entity (and, with this, in the accounts kept by the legal entity) is increased by the legally mandated accountant's audit. To increase confidence even further, such audits may only be performed by an accountant who has a permit from the Netherlands Authority for the Financial Markets. Faith in the recording is increased in this manner step-by-step. The criminality of forgery can also be considered a legal measure aimed at increasing confidence in the recording of data. Market parties aim to increase confidence by means of their services are also supervised under certain conditions (by force of the European eIDAS regulation for providers of confidential services).

Many contractual provisions also relate to the recording of data. Consider the obligation on the part of a supplier to record the information used to send an invoice or for providing so-called service level reports. An audit right is often claimed to increase confidence in that recording. Since the supplier often has insufficient faith in his customer, it is usually determined that the audit may only be conducted by an independent third party.

In short, there are many legal and contractual regulations governing the reporting of information and increasing the reliability of that recording. A multiplicity of parties is active in the "confidence market": accountants, notaries, organisations that manage central registries such as the Kamer van Koophandel (Dutch Chamber of Commerce), Kadaster (the Landregistry), Netherlands Vehicle Authority, DUO (Dutch educational finance bodie) and also private and confidential-service providers. To a certain extent, blockchain can itself generate confidence in what is rewarded in the blockchain although not completely, because complete confidence requires more than the certainty that data is not altered after the fact. Nonetheless, if blockchain technology is truly to succeed, it will undoubtedly have far-reaching consequences for laws and regulations, contracts, and for players in the "confidence market". Fortunately, there is still plenty of time to think about the long-term effects of blockchain.

# Analysis of knowledge requirements

## Introduction

The concepts of blockchain and smart contract have already been explained and the legal questions covered. This chapter discusses the knowledge required to be able to use blockchains and smart contracts responsibly. One of the responses to the DAO affair was that there is a need for a new discipline, legal engineering, in which various disciplines are united or at least practitioners in these disciplines, lawyers, IT specialists, risk managers, et cetera collaborate productively. There is also a need for methodologies, tooling, standards, et cetera. The Netherlands would seem to be in a good position to take legal engineering forward, since a great deal of experience (for example in the area of 'rule management') has already been gained through performing public service tasks. The Netherlands also has a large academic body of knowledge. First of all, however, we will examine the experiences and practical needs of various members of the business community, before going on to forge a link to experiences in executing and academic knowledge surrounding rule management.

## Individual observations and experiences

Currently, the largest and most frequently used permissionless blockchain for developing smart contracts is Ethereum. The Ethereum Virtual Machine was specifically designed to handle smart contracts. The most widely used programming language for making smart contracts on Ethereum is Solidity. This language is not well known among most lawyers and risk managers, and only limited to IT specialists. These IT specialists however often lack the legal knowledge necessary for drawing up contracts or do not have sufficient experience in risk management to be certain that the contracts contain no loopholes. There is a risk that loopholes would enable malicious individuals to trigger illegal contract transactions (as has been extensively documented following the DAO hack) which may be in breach of current legislation and regulations or which could even enable hackers to transfer currency from the contract illegally, as in the Parity hack.

Knowledge of Solidity is currently in short supply. Only a few programmers in the Netherlands have truly mastered this programming language. Added to that, programming a contract in Solidity is not enough to enable it to actually work. For it to work, the contract would have to be compiled in hex code and published on a blockchain through a transaction. In order to then communicate with a contract, extensive knowledge of the ABI (abstract binary interface) would be needed. That is precision work. If a name is not entered correctly, the contract can no longer recognise which function is being triggered. That is usually done through a JavaScript and HTML interface. And this leads to multiple difficulties: to be able to draw up a contract as well as communicate effectively with it quite quickly requires knowledge of various types of programming languages.

Apart from knowledge of code, it is important to aim for standardisation at various levels, certainly given that every blockchain is currently still developing its own smart contracts. We have broken these levels down as follows:

1. Firstly, the use of design patterns: best practices and recurring code structures for particular legal elements. There is a requirement for the code structure in a smart contract for dispute resolution, for instance, to be designed in the same way on the Ethereum blockchain as on Hyperledger Fabric.
2. Secondly, it is important to look at standardisation of ontology. What we call an owner in a smart contract should ideally also be such in another smart contract in another blockchain. That would greatly enhance the legibility for people without IT expertise.
3. Thirdly, it is advisable to look at standardisation of data elements. The fact that we understand an owner to mean the same in different contracts is a first step; the description of these owners should also be the same (for example, an owner comprises three letters, 10 numbers and then three more letters).

It is highly likely that in the near future, when an audit is conducted, they will analyse the smart contract code on a blockchain in relation to its justification for a technological solution. To verify that it represents what was once promised on a website or in a statute book, it would have to be decompiled. And that is difficult to do at the moment. It is currently only possible to check whether a particular input of source code actually led to this contract in bytecode by publishing or sharing source code.
As indicated at the beginning, it is expected that legally qualified people might have to do this in the future. To date, it has hardly appeared in any study programme and is therefore extremely rare.

In addition, there is evidence that people who might possibly want to master the computer language in many cases lack general knowledge of blockchain. That was initially the case at OurSurance, where there was a lack of understanding at the time of how smart contracts operate. This problem manifested itself in the original smart contract and led to the situation described in the University of Maryland's documentation (among others): from the point of view of the programmer the contract was correct, but from the point of view of risk and compliance, it did not meet requirements. Experience has shown that it is even more crucial than before to have a well-crafted process design for business models built on blockchain and smart contracts to meet all the compliance requirements. Examples of this are correct financial processing of incorrect transactions in whatever form, or how to deal with data storage in permissionless blockchains.

Several projects have shown that knowledge has to go beyond specialism in a particular field. A thorough knowledge of elements that transcend a field of practice can contribute very effectively to a well-crafted design and implementation of blockchain and smart contract solutions.

Examples include the degree of confidentiality of the data. Only by understanding the desired functionality can the blockchain and smart contract specialist ask the right questions and prevent incorrect assumptions from being made. This will become more urgent if a consortium of companies is involved where collaboration takes on new forms. Besides managing the blockchain network, all the details will have to be considered at smart contract level. Which information is shared, what should be recorded in the smart contract and what should be recorded elsewhere, how will the consequences be effectuated, how will updates be implemented or the agreement terminated? What are the requirements for a third party to join, should standards be set for the security of computer programs that carry out transactions on behalf of the counterparty? Answering these and similar questions demands a great deal of flexibility from those involved in these types of negotiations and the ability to analyse a large number of scenarios.

The more a smart contract has the characteristics of an agreement, the more important it is that parties understand and knowingly accept the content. A comprehensible appendix, as is increasingly enclosed with general terms and conditions, could be the answer. In addition, programming languages will probably be developed which describe the reality at a higher level of abstraction and by so doing, make it more understandable. Obviously, every user of a smart contract will have to have a certain amount of basic knowledge and common sense. We can furthermore assume that audits will be conducted by experts who will raise the alarm if something is not quite right, as currently happens with open source software and general terms and conditions of well-known services. But then there will have to be a sufficient number of these experts around – which, given the speed of developments, is certainly not guaranteed at the moment.

Similarly for compliance, it is extremely important that we bridge the gap between the legal and technical world; there is an urgent need for lawyers who can understand technology. Past experience has shown that technicians often fail to take legal requirements seriously ('there aren't any sensitive personal details involved so it'll sort itself out') and that there are lawyers who do not understand technology well enough. These challenges cannot be solved with a simple training course.

To draw up a successful smart contract in permissionless blockchains, where in most of the cases one must pay a fee for every single transaction, knowledge is required that goes beyond programming or an entrepreneurial mentality: a certain understanding of game theory and economics is also required. The user will have to be distinctly motivated to make use of the contract and any improper use must be eliminated. We saw the adverse effects a bug can have in the DAO hack, but a 'bug' in a smart contract could also be an overlooked trigger that produces undesirable results.

# Rule management

Practical experience with smart contracts among business people illustrates that there is a need for a robust method to translate legal norms into specifications to enable automated implementation and to guarantee their operation down to the level of machine code. The Netherlands has already acquired plenty of experience in this, such as in the area of 'rule management'. There is also already a large academic body of knowledge in the country. The idea behind rule management is that people in business cannot read programming code and so they need to be offered something that enables them to check that the rules have been translated correctly, or that enables them to do it themselves. Depending on how formalised the tool used for the translation is, it will be suitable to a greater or lesser degree for automated conversion into executable programming code. The specifications that emerge from the legal norms are preferably recorded in a (vendor-neutral) layer between the norms and the IT solution to be used for executing these norms. On the assumption that every effort will be made to automate the translation from models to code, data objects, configurations, et cetera, the pros and cons of being user friendly and the degree of formalisation required for automated conversion will have to be weighed up. Any decisions that still have to be made about this translation should also be discussed with the relevant parties and then documented.

The first advantage of introducing an intermediate layer for recording the specifications is that the disciplines involved in business can tackle the specifications in what is for them a (slightly) more natural way, by presenting them in a visual setting, a decision table or in a standard that resembles natural language such as SBVR, for example. A second advantage is that the knowledge present in the organisation will not be lost in code, but will be recorded in an intermediate layer (based on standards), which will also avoid any dependence on a specific vendor. A third advantage is that the specifications – as long as they are sufficiently formal – can be examined for logical consistency: have all eventualities been considered, are there no objections et cetera? So-called ‘reasoners’ can moreover ‘discover’ knowledge through drawing logical conclusions. A fourth advantage is that the knowledge becomes easier to reuse and therefore becomes an asset in itself. Finally, subdividing specifications into logical components opens the way to service-oriented solutions (knowledge as a service) which can be shared  – whether or not for benefit – with third parties.

When legal norms are translated into more or less formalised models, it is essential that the specifications can always be associated with the norm, whether that is a law, a policy rule or a contractual obligation. This will make the translation traceable and will furthermore make it possible to perform an impact analysis on the consequences of a change to the norm (what if?). The data can also be examined during this analysis, for example, to see how many existing contracts are affected by the change in the law? Ideally, every translation (right down to the machine language) should be traceable, forwards and backwards.

It is conceivable that, in due course, the intermediate layer will become less of an environment for translating legal texts and increasingly an environment in which design work can be performed directly, with the aid of reusable objects (e.g. configurable) standard contracts and design patterns for common legal constructions.

There is already a vast body of knowledge about rule management in IT. To date, however, few studies have been done into the integration of risk management and legal issues when it comes to smart contracts and blockchains. The number of studies is currently on the rise, judging by the number of PhDs working on this topic, and a good number of master’s students have already conducted proof of concept / proof of technology research into this topic too. The SARNET project, DL4LD and other large NWO projects are addressing these matters and work is being done on finding solutions with a variety of other partners by taking an interdisciplinary approach and including lawyers, accountants and auditors and naturally AI experts and IT specialists, including cryptography experts.

In a project that formed part of an undergraduate security class at the University of Maryland [44], students were reminded of the fact that besides the traditional accurate programming of contracts like these, they also had to take account of general risk issues such as financial risk and operational risk (can someone commit fraud in a smart contract, etc.?). It turned out that in almost all cases, the students needed to repeat their programming several times in order to produce acceptable smart contracts, not simply from a programming point of view, but also on the basis of risk management. The legal framework that might be needed was not even included in this. This aspect is addressed in other study programmes such as Policy Making and Rule Management at the University of Amsterdam.

A different study at the National University of Singapore/Yale-NUS college [45] focused attention on another component: the technical translations that are made. Because contract code like this is typed and not placed on a blockchain as such, but is compiled into bytecode, there has to be an assurance that this compiled code is the correct representation of the code above it. This raised a number of areas for improvement that could be addressed. This study only focused on the translation of a high-level computer language into a low-level computer language, but in practice, an additional layer will be placed above this one in the future: the translation of natural language into computer language, since not many individuals are good at reading computer languages. Attention must therefore be given to the fact that several languages have to be translated correctly. Hyperledger Composer is an example of a higher level of abstraction, which attempts to close the gap between reality and blockchain. It can be used

[44] https://eprint.iacr.org/2015/460.pdf

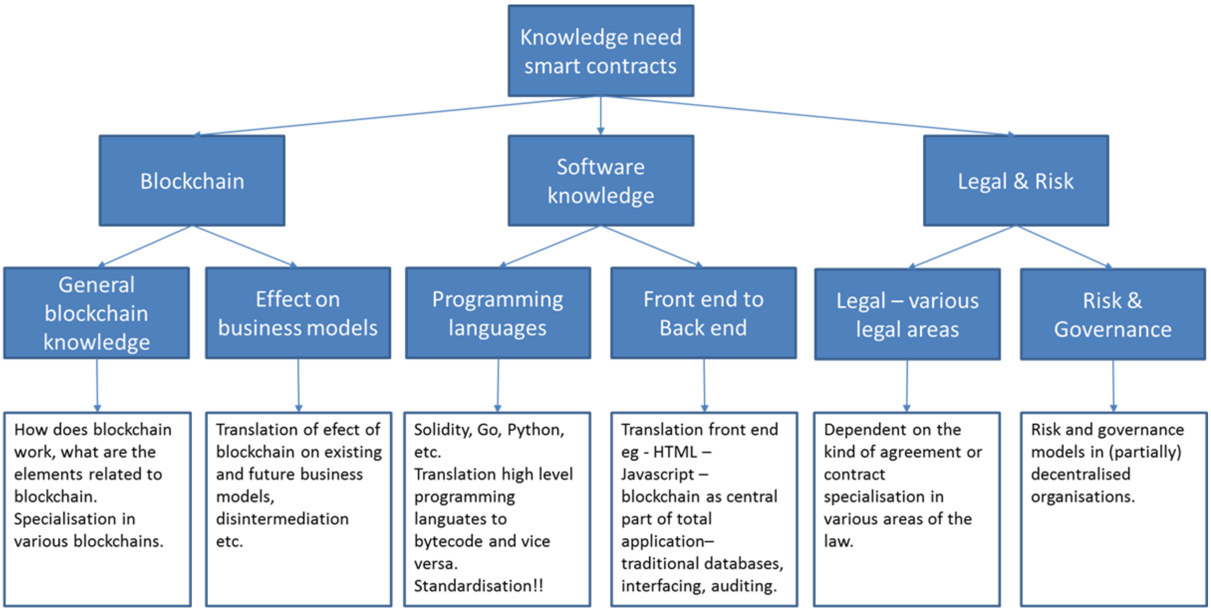
[45] https://www.comp.nus.edu.sg/~hobor/Publications/2016/Making%20Smart%20Contracts%20Smarter.pdf

to model applications that run on a Hyperledger Fabric blockchain.
One aspect that the quoted studies did not look at extensively is the translation of legal language into code. It is worth looking closely at what ‘rule management’ has to offer in this respect. This translation and the audit trail (the translation of the norms through the various layers of the system as well as demonstrated correct execution) will be exactly what lawyers, auditors and regulators will most likely focus on. For that reason, these matters should be considered carefully at the design stage, together with matters such as dispute resolution, the possibilities of hybrid contracts (language and code) et cetera.

A movement towards ‘rule management’ in the context of smart contracts is certainly welcomed. Possible standards for blockchain and distributed ledger technologies and smart contracts are being developed as part of ISO/TC 307. An international standard for expressing legal norms in smart contracts can probably be developed in this context (in due course) to ensure interoperability and also to prevent vendor lock-in, in line with the standardisation requirements for smart contracts identified earlier in this chapter. Apart from standards for contracts, the Dutch Blockchain Coalition also considers it necessary to have standards for other matters, such as the identification of people, legal entities and objects.

# Clustering knowledge requirements for identified sub-areas

As mentioned previously, knowledge requirements will go beyond technical or legal knowledge alone. Apart from the knowledge required in these two areas, a broader perspective needs to be considered in order to implement smart contracts and business models effectively on the basis of blockchain and smart contracts. An initial visualisation of knowledge requirements for the future is set out below.


Knowledge need smart contracts
Blockchain
Software knowledge
Legal & Risk
General blockchain knowledge
Effect on business models
Programming languages
Front end to Back end
Legal – various legal areas
Risk & Governance
How does blockchain work, what are the elements related to blockchain. Specialisation in various blockchains.
Translation of efect of blockchain on existing and future business models, disintermediation etc.
Solidity, Go, Python, etc. Translation high level programming languages to bytecode and vice versa. Standardisation!!
Translation front end eg - HTML – Javascript – blockchain as central part of total application– traditional databases, interfacing, auditing.
Dependent on the kind of agreement or contract specialisation in various areas of the law.
Risk and governance models in (partially) decentralised organisations.

Literature analyses of what is currently known about knowledge requirements for blockchain and smart contracts, and especially analyses of the various individual use cases, have identified three pillars in which knowledge development is needed and where knowledge is needed in order to deal effectively with smart contracts and blockchain business models in the future.

These three pillars are:
1. Blockchain knowledge
2. Software knowledge
3. Legal & Risk knowledge

All pillars in which there are indications that knowledge development is required can be broken down further into a number of sub-areas. Each of these sub-areas has a large number of specialist sub-areas. This initial exploration does not go further than a division in sub-areas. The recommendations for follow-up studies will include a recommendation to look at the specific knowledge areas and specialisations in the sub-areas in more depth.

## Blockchain knowledge

The blockchain knowledge pillar is subdivided into two sub-areas:

*1. General blockchain knowledge*

This sub-area encompasses everything about what blockchain is and how it works. This includes, but is not limited to, knowledge about how blockchain operates, knowledge of cryptography, knowledge of the various elements and characteristics a blockchain can have, knowledge of the different types of blockchains, knowledge of various types of consensus mechanisms, and knowledge of products and services on blockchain such as cryptocurrencies and smart contracts.

*2. Effect on business and industry models*

This sub-area encompasses everything about the possible effect of blockchain on process models, business models and industry models. In this area, knowledge will have to be acquired about general modelling, redesign, design thinking in the widest sense and also some historical model development: where do current models come from? This will aid understanding of what the effect of blockchain could be on future models and will thus enable, for example, the development of an effective governance structure around decentralised smart contract-based models.

The first of these two sub-areas is rather more technical than the second, as it deals with the actual operation of blockchain and its elements. The second sub-area is less technical and will have to focus more on the business and social impact as a consequence of blockchain in combination with a historical awareness of how organisations and societies have developed.

## Software (and IT) knowledge

The second pillar is the most technical in nature and encompasses the technical software knowledge needed to deal with smart contracts. This pillar can also be subdivided into two sub-areas:

*1. Programming languages*

This sub-area is traditionally the world of programmers. In addition to traditional programming languages such as Java, JavaScript, Python, C++ and C#, and Go, the acquisition of knowledge required here mainly refers to knowledge of new programming languages such as Solidity, currently the most important programming language for smart contracts, but also forthcoming concepts, which are more visually oriented, such as Babbage. Especially in the last-mentioned language, there is currently a major shortage of programming knowledge. The development of other new languages will also have to be constantly monitored.

*2. Front-end to back-end interaction and integration*

This sub-area focuses on correctly translating what happens at the front end (such as a website built using HTML/CSS/JavaScript) into what is executed at the back end. This could be the correct execution in traditional databases, but obviously also in smart contracts on a blockchain. This translation goes further than just a digital front end such as a website, and also applies to such things as PDF summaries given to an end user on a site, which can be downloaded. The written promises and expectations communicated to end users obviously have to be correctly translated through the various layers, often requiring multiple programming languages to be linked together.

Knowledge will also have to be acquired about sideways integration. This might imply linking smart contracts to traditional databases that act as an “oracle”, for instance.

Apart from the front to back-end interaction, knowledge will also have to be acquired with regard to standardisation, as described earlier in this report with reference to design patterns, ontology and the standardisation of data elements. Such knowledge is not only needed in order to ensure that internal integration and interaction is in order, but also to enable good communication and interaction with other smart contracts, even eventually across blockchains.

As indicated above, this is the most technical sub-area, alongside the first sub-area of blockchain knowledge as described earlier.

## Legal & Risk

The third pillar of knowledge requirements is the one relating to legal & risk. This pillar can also be subdivided into two sub-areas:

*1. Different areas of the law*

This is probably one of the broadest sub-areas to be identified. In the first place, to be able to deal effectively with smart contracts, there is a need for detailed knowledge of general law and a thorough understanding of the specific key areas in law as a consequence of blockchain, as identified earlier in this report. In addition, specialisation in various areas of the law may be required where forms of contract are used that may also be drawn up in the form of a smart contract.

*2. Risk & Governance*

Given that many smart contracts can retain value and that part of their purpose in being executed may be to transport value between various parties, additional attention needs to be paid to matters such as operational and financial risks and how these can already be mitigated in the design. Where necessary, scope should also be provided in the smart contract design for action to be taken, if necessary. There is an even greater need to consider this in advance given the immutable nature of the code in a smart contract on a blockchain. Thorough knowledge of risks and the management of such risks, including clear responsibilities, is therefore necessary when drawing up a robust smart contract. This goes further therefore than only making the code watertight against hackers: it includes being able to deal with unintended actions.

As already mentioned, these sub-areas can themselves be defined to a much greater extent into various specialisations and super-specialist areas. There is an expectation, also described further on in this report, that there will be a growing need for these individual specialist areas on the one hand, but that on the other hand, a large number of the current gaps in knowledge are primarily found in the synergy between these individual knowledge areas. There will be a growing need for multidisciplinary fields of practice in this context.

## Other knowledge areas

The clustering of knowledge areas identified in this exploration focuses for the most part on the knowledge requirements for working on or with smart contracts, or processes related to smart contracts, as per the objective of this working group. Since blockchain is obviously broader than smart contracts, knowledge of other types of blockchain-related spheres of activity is needed of course in many more areas. A good example is the field of cryptocurrencies, where knowledge of economics and financial markets is more important to traders, but these areas have not been included in this exploration.

# Cross-functional knowledge

It is extremely important – perhaps even more so than when other systems are implemented – that experts are involved who have an in-depth knowledge of the domain and of the technology in question, as well as knowledge of the legal and risk aspects.

There is a wide variety of blockchain and smart contract implementations, which can easily lead to misunderstandings about the characteristics, requirements, possibilities and impossibilities of the solution options. And it is exactly at this interface of different expertise that there is a shortage of professionals. Besides expanding knowledge in the individual specialist areas, increasing numbers of multidisciplinary knowledge workers will be needed.

These people should be sufficiently specialised in one of the pillars to be able to work as professionals in that area, but should also be sufficiently well-informed about the other pillars that they can communicate with these professional groups and in this manner, could take on a coordinating, leading role.

## Subsequent steps and possible parties for developing knowledge requirements

Industry can play a role in the short term with regard to training requirements by arranging a variety of master classes and courses. No central coordination is needed for this, since initially, market forces will be sufficient for this to happen. A long-term solution will, however, also have to be discussed.

An enduring solution for the knowledge requirements that arise as a consequence of blockchain and smart contracts will have to be found in the long term. It might be found in collaborative ventures between universities and faculties, such as law faculties and technical faculties, which, up to now, have not yet formally worked together.
An encouraging note here is that these types of cross-functional study programmes have been fairly successful in the past. Examples include Life Sciences & Technology study programmes or study programmes that combine Systems Engineering, Policy Analysis and Management with Industrial Engineering and Management Science.

There is clearly a role for traditional educational and research institutions such as universities and universities of applied sciences in all this. The first universities to come to mind in the Netherlands that can go deeper into the individual elements are those that specialise in technology such as Delft, Eindhoven and Twente, and those that specialise in law such as Amsterdam, Groningen, Leiden, Nijmegen, Maastricht, Rotterdam, Utrecht and Tilburg. This is particularly the case for faculties that have been working on rule management, IT and law for a long time. In relation to multidisciplinary study programmes, the existing joint-degree partnerships such as the combinations involving Leiden, Delft and Rotterdam or between Tilburg and Eindhoven can be assessed to see if the programmes can be set up as a separate study programme. This should also be investigated at a non-university level.

Given that this study is an initial exploration, it would be advisable to examine this matter in greater depth with a variety of the parties mentioned above. The content and need for various subject clusters between the specialist areas should be discussed in more depth, as well as where current study programmes already offer solutions (or partial solutions). For example, there may be a need for a subject cluster in the proposed multidisciplinary study programme that places more emphasis on legal aspects or alternatively more emphasis on technical aspects.

Since this will require collaboration between many different parties, we propose assigning the coordination to the National Blockchain Coalition in the form of a subsequent working group for

Human Resources. This group can obviously partly comprise people who have worked on this exploratory study in order to be able to guarantee consistency and rapid progress.

With regard to timelines, it would be advisable to commence soon, given the speed in developments in blockchain and smart contracts, and the sums invested. We therefore propose that the follow-up working group meets for the first time in November 2017, with a target of 2018-2019 for further details to be worked out.

# Summary of recommendations and subsequent steps

*A more focused approach from government with regards to educational requirements and development of legislation and regulations.*
The blockchain expert group was a very good start, all the more because the various experts now know each other better and can find each other more easily. Because of this, more is known about the various initiatives being taken when it comes to blockchain in the Netherlands. Unfortunately, that has highlighted the fact that there are still too many initiatives working independently of each other (and unbeknown to each other). If in the Netherlands we want to be more efficient in this area, we will have to coordinate these initiatives more effectively to ensure the transition is faster and more successful.

*Conduct further research into knowledge requirements in relation to blockchain and smart contracts*
Due to the composition of the working group and the methodology, the major focus in this exploratory study with regards to future knowledge requirements was the immediate requirements in the field (government bodies and industry) rather than on scientific knowledge requirements. It would be advisable to concentrate more on the scientific side of knowledge requirements in a subsequent process and, in addition, to conduct deeper research into what is already available with regards to the acquisition of knowledge.

*Research into deeper, multidisciplinary study programme IT and law*
Given two areas which have traditionally enjoyed little synergy, technology (specifically computer programming) and law, can converge in smart contracts, it is advisable to explore the need for and explore possibilities of setting up a multidisciplinary study programme through collaboration between universities specialising in technical subjects and universities specialising in law. In doing so, IT and law should not be approached as a holistic truth, but students should be able to acquire a real understanding and experience of computer languages and law. Given that smart contracts could represent various types of legal contracts, students should be able to choose a specific field in law in which to specialise.

*Further research into questions and gaps in legislation*
As indicated earlier in this report, in many cases several questions on legal aspects of smart contracts still need to be answered. It is therefore advisable for the subsequent group to include several people at the start who have been involved in this working group. This is advisable from the point of view of continuity, certainly in view of the growing number of use cases.

*Documenting acquired knowledge in fixed definitions*
One of the most important findings of the working group is that there was a considerable lack of clarity as regards definitions before we started. The semantics led to almost Babylonian confusion in many cases. It is therefore advisable to document clear-cut definitions to enable future debates and exploratory studies to progress as unambiguously and smoothly as possible.

*Research into possible standardisation*
Given that standardisation of smart contracts cannot be regulated for each blockchain platform, standardisation will have to focus on the form and content of smart contracts and in particular, design patterns, ontology and standardisation of individual data elements.

# Frameworks – Use Cases discussed

## Blandlord - Crowd ownership on Bitcoin Blockchain

Blandlord introduces crowd ownership in the Netherlands and deploys blockchain technology to enable this.

Crowd ownership stands for joint possession. Ownership is distributed among a group of owners; this fits the trend of the sharing economy. Each co-owner can participate and shares in the revenues pro rata. This makes crowd ownership egalitarian and democratic: a group of equals takes joint responsibility.

Blandlord uses smart contracts to share ownership, for the financial flows and for owners' participation in decisions. This makes joint possession possible, such that thousands of owners participate completely in a decentralised manner and profit from their ownership.

## Deloitte - Handelsgebouw Rotterdam

### Blockchain in real estate

At the start of 2017, Deloitte Real Estate, in cooperation with the municipality of Rotterdam and the Cambridge Innovation Centre (CIC) worked on the first concrete blockchain application for lease agreements.
By using blockchain technology, a uniform source of real estate information arises that various stakeholders can use, and in which multiple audits of the same data are no longer required. Five important steps were taken in the project.

1. Digitisation of building information

An important first step was the creation of a blockchain ledger with real estate information for each building registered on the blockchain. By digitising the building, a digital ownership coin was "struck" for the building. This so-called "token" is a digital fingerprint of the building, including all of its associated details. The digital fingerprint refers to a database in which the building's details are stored. Existing registers such as *Kadaster* (the Landregistry) and the BAG are used at this moment as the basis of this first step. This information can be supplemented by other basic data such as the building's floor space, energy label, the structure of the building and maps. These are just a few of the data points that are now included in the building's file/passport. By deploying blockchain technology, this data is given a timestamp and an irrefutable record is made of the moment and the database from which the information is taken.

2. Digitising the ownership situation

The next step after digitising the building information is linking an owner to the digital coin that the building or portion of the building represents. In order to do this, a digital identity must be created for the owner. Existing registries, such as Kadaster (the Landregistry) and the Kamer van Koophandel (Dutch Chamber of Commerce), are also used for this step. In practice, these registries have been used in the Netherlands for years, and the legal system relies on these registries. Recording digital identity in an irrefutable manner is a big issue in the current state of technology. Central authorities (such as the Dutch Chamber of Commerce mentioned earlier) are used to perform transactions. In a future situation in which blockchain has become common practice, the role of the central parties (the trusted third parties) will change considerably.

3. Transfer of ownership

The real estate market has countless players, stakeholders and involved parties. The only constant factor in this entire process is, in fact, the real property. After a number of years, ownership transfers from party A to party B. This equally applies for the tenant, for financing and for other obligations. In the current situation, real estate is transferred using a notary. The financial, legal and tax obligations of both the current owner in the future owner are visualised and involved in the transaction.

Using the digital tokens (digital ownership point) represented by the building and a reliable digital identity, transferring ownership will become simpler. Using an online transaction, ownership of the building can be transferred more easily. The holder of the so-called ownership coin is the only party who will be entitled to "encumber” those coins with legal obligations such as a rental contract. In current practice, this will represent a significant turnabout in thinking and countless legal and tax questions will arise that require resolution.

4. Entering into lease agreement

The next step is signing lease agreement simply and online. With the increasing need for more flexible use of office space, such as in the start-up community in the Groothandelsgebouw, the process of entering into leasel obligations must be changed radically. Within the so-called customer journey of the tenant/user, countless optimisations can be achieved. In the context of the project, we chose a contract template module that enables multiple parties to work in a digital environment on the signature of a rental contract. Based on the final negotiating results, the contract is drawn up specifically for the specific tenant. Using a blockchain transaction, this lease agreement is signed digitally by one of the parties to the contract and sent to the counterparty for signature. The counterparty receives a notification and also signs the contract digitally after inspecting the contract. Once again, this involves a blockchain transaction. After the completion of these steps, a version of the lease agreement signed by both parties will be added to the blockchain. This contains an irrefutable record of the agreements that the landlord and tenant have made with each other. The most important advantage is that it saves considerable time in the process for both the landlord and the tenant.

5. Making contract information accessible to third parties

Throughout the lifecycle, the real estate owner will share information about his building with third parties. With the bank in the context of (re-) financing, with an auditor in the context of the annual audit of the books, with an assessor for purposes of evaluation, and with a (potential) buyer in the event the building is sold. During each of these occasions, the various parties involved perform audits of the actuality and completeness. For the parties involved in the network, blockchain offers the potential to use the same (decentralised) source of information.

Furthermore, blockchain ensures that the data recorded can be shared with third parties in a reliable, uniform, secure and rapid manner.

## OurSurance - Peer2Peer insurance

OurSurance is a Peer2Peer insurance project in which smart contracts on the Ethereum blockchain are used to link people needing insurance on a 1-to-1 basis to investors who wish to invest in individual policies. Here, the insured party and the investor do not know each other. Anyone who wishes to do so can become an investor on the platform – from individuals to companies. After a request has been submitted, the investor is linked to the insured party using a smart contract via an auction mechanism, including the possible amount of the claim. Because of the smart contracts, the insured party runs no risk of non-payment in the event of a valid claim. Claims processing takes place through mutual voting, such that an independent third party must cast the deciding vote in the event of a conflict. OurSurance has no part in the ultimate flow of premiums and any payout. After concluding the insurance policy, the policy exists as a smart contract on the Ethereum blockchain and the insured party gets a PDF summary. All of the policies are one-time policies. Afterwards, the insurance contracts are concluded.

## APG - Pension value transfer

In a joint experiment, APG and PGGM studied how value transfers among pension funds can be enabled using smart contracts. If a person takes a new step in his or her career then they might switch from one pension fund to another via a different (collective) employment agreement. At that moment, an administrative process is started in which the employee must show by means of various letters and forms that the accrued pension value can be transferred from one pension fund to the other. In the experiment, smart contracts were used to automatically check whether and where a person has collected the accrued pension value, in order to transfer the pension value automatically, or via a single click of a mouse, to the employee's pension fund used by his new employer. So in this experiment, the smart contract contains the functionality for checking an individual's link to another pension fund on the basis of his data and for processing the transfer of the pension value.

## IBM - Bike plan

IBM and RDW focus on the bicycle owner in this working prototype, which runs on the permissioned blockchain Hyperledger Fabric. The owners of electric bicycles can register and transfer their bikes, prove ownership and report any theft. The "smart lock" on the bike notes the location of the bike when it is locked. In the event of theft, police can see the bike's most recent position and respond immediately. Insurance companies can process the claim automatically using the smart contract. Various aspects – such as the original value, the date the insurance policy was concluded on, whether the bike was locked and whether the theft was reported – can all be included in the decision as to whether or not to pay out. This is a good example of how the development of an ecosystem can quickly result in significant and sometimes unexpected innovations and how sensors can reinforce the power of a smart contract. The project received the 2017 Computable Awards in the category "Government IT project of the Year".

Dutch
Blockchain
Coalition

connect and create

Dutch Blockchain Coalition

connect and create

Founding partners

ABN·AMRO

alliander

Brightlands

CGI

Experience the commitment®

C'M'S'

Law . Tax

CWI

Centrum Wiskunde & Informatica

ECP

Platform voor de InformatieSamenleving

ENEXIS

NETBEHEER

ING

kamer van koophandel

KNB

Koninklijke Notariële Beroepsorganisatie

Rijksdienst voor Identiteitsgegevens
*Ministerie van Binnenlandse Zaken en Koninkrijksrelaties*

Ministerie van Economische Zaken en Klimaat

Ministerie van Infrastructuur en Waterstaat

Ministerie van Justitie en Veiligheid

nationale nederlanden

NWO Nederlandse Organisatie voor Wetenschappelijk Onderzoek

Port of Rotterdam

pwc

Rabobank

Radboud Universiteit

RDW

Tilburg University

TNO innovation for life

TU Delft

de volksbank

www.dutchblockchaincoalition.org