Chapter 27: Blockchain and other distributed ledgers

prof.dr. Eddy H.J. Vaassen

Tilburg University, Erasmus University Rotterdam, BDO

August 2018

1. Introduction

A blockchain is a *distributed database* that contains sequentially interlinked ('chained') clusters of transactions ('blocks') with *tokens* that follow the rules of a specific *trust protocol*. In this description three concepts stand out. Firstly, a distributed database, as opposed to a centralized database, is an organized collection of data that is integrally held at physically or logically separated storage devices. This implies that each of these storage devices contains a full copy – not a subset – of the database in question. Secondly, in essence, a token may either be a hardware device (for example, a chip card) or a piece of data (for example, a string of characters), although there are more sophisticated taxonomies that apply multiple criteria to classify tokens in blockchains (Euler 2018). A token as a piece of data is the type of token that lives on a blockchain. Specifically, in the realm of blockchain a token is a chain of digital signatures that represents (tangible or intangible) assets that are exchanged in transactions. The cryptocurrency Bitcoin can be considered a token. Thirdly, the software that manages the exchange of assets via tokens on a blockchain is dubbed the trust protocol to indicate that in a blockchain world trusted third parties are replaced by this software.

Blockchain as the technology underlying the Bitcoin was first described by the yet unknown Satoshi Nakamoto (Nakamoto 2008) but has gained traction ever since resulting in many more tokens and variants thereof, including Ether, Bitcoin Cash, Litecoin, and other (over 3000) cryptocurrencies.[1] There is not just one blockchain, each cryptocurrency has its own blockchain or has its own part of a certain blockchain. Blockchain can also be used for other tokens than cryptocurrencies. As a matter of fact, this chapter argues that applications in other tokens than cryptocurrencies will gain momentum in the forthcoming years because they enable more efficient and effective ways of transacting (by, for example, improved business processes and business process interoperability across supply chains), as well as more efficient and effective accounting, controlling, auditing and oversight practices (by, for example, linking a continuous monitoring system in a blockchain, or making additional journal entries in a blockchain to enhance the verifiability of local databases).

Although blockchain often is called a new technology, strictly speaking this is not entirely correct. Because blockchain is a combination of various existing technologies, theories and algorithms such as public and private key cryptography (Diffie & Hellman 1976), hashing and digital signatures (Haber & Stornetta 1991), smart contracts (Szabo 1994, 1997), game theory (Lamport et al. 1982), and triple-entry accounting (Grigg 2005) the innovation of blockchain lies in the smart combination of these.

Currently blockchain is at the top of the hype (Panetta 2017) and there is a reason to that. It has some inherent features that are said to disrupt the way transactions are managed and accounted for, including:

[1] See Armasu (2018) for an overview of the top 25 cryptocurrencies by market cap as per June 1, 2018.

1. It eliminates the need for trusted third parties and replaces these by trust protocols. By means of disintermediation processes become more efficient and substantially less prone to human manipulation.
2. It is built upon a distributed consensus model. A transaction can only be recorded in the blockchain if it has been validated by a majority of the nodes that participate in the network of that blockchain.
3. Once a transaction is recorded in the blockchain it cannot be removed or altered. Because of its design blockchain is the shared single-source-of-truth.

Along with the said blockchain hype goes the vast number of publications on blockchain's use cases and its underlying technology. Many of these publications are just adaptations of other publications. This may not be a problem that should interest us as such. However, due to the innovative nature of blockchain a *common body of knowledge* with agreed upon syntactical and semantical tenets is still lacking. Hence, any text on blockchain currently still makes many more arbitrary choices with respect to the syntax and semantics to be used than more mature fields. While trying to explicate the blockchain language as clearly as possible and making use of the scarce literature on blockchain ontological issues (for example, De Kruijff & Weigand 2017) this chapter is no exception since the very same arbitrary choices are made.

The remainder of this chapter first discusses Bitcoin 'under the hood'. The reason for giving an in-depth discussion of Bitcoin is that its underlying blockchain technology is extremely well thought out and has served as an example or at least as the main source of inspiration for most of the blockchain applications as we currently know them. It then continues with a section on the importance of public and private keys for digital signatures, the use of digital signatures in tokens, and data sources outside blockchains for blockchain applications. Extending our discussion from blockchain to distributed ledger technology the subsequent section then discusses some more versatile applications that go beyond the exchange of cryptocurrencies as in the Bitcoin blockchain. By exploring new types of application inevitably blockchain technology may prove itself to be not the right term. Therefore, while maintaining many of the basic principles of blockchain, this section broadens the notion of blockchain to the more generic term *distributed ledger technologies* (DLT). Along with the discussion of a classification of distributed ledgers, also smart contracts and alternative consensus mechanisms are discussed since the type of distributed ledger, the type of smart contract (if any) and the consensus mechanism are not independent. Having acquired a good sense of the language of distributed ledger technologies and to illustrate the richness of potential applications a wide variety of examples of distributed ledger technology use cases are then discussed. The chapter concludes with some guidance for deciding whether or not to engage in a distributed ledger project or remain confined to legacy centralized databases.

2. Bitcoin

Fiat money such as the Euro and US Dollar, although intrinsically just worth the price of the paper or metal used for printing or minting it, has value because it is scarce and people can use it for purchasing goods and services. Money is made scarce by a central authority (the central bank) that in return guarantees to transactors that it is always backed. Because of this backing people trust it. If there were no central authority trust would be difficult to build and maintain because one cheating transactor could undermine the entire system. The Bitcoin blockchain provides a mechanism for creating electronic money that can be trusted and for that reason used without having a central authority in place. The obvious question then arises who creates new Bitcoins if there is no central authority. The Bitcoin trust protocol, dubbed the *Bitcoin core* to distinguish it from the cryptocurrency Bitcoin, uses a mechanism called *proof-of-work* to create new Bitcoins out of thin air.

Proof-of-work

Imagine a gold mine where miners spend hours of hard work to once in a while find a nugget of gold. Nakamoto (2008) applied this notion of having to put a lot of hard work into something with a low rate of success to the creation of new Bitcoins. Whereas fiat money is created by banks giving out loans to their clients and central banks in the realm of monetary policy by buying government bonds and occasionally printing new banknotes or minting new coins, Bitcoins (and currently most other cryptocurrencies) are created by computers working hard to mine these cryptocurrencies. In the blockchain, the concept of computers working hard boils down to expending CPU power to solve a cryptographic puzzle by trial and error. So, the fact that Bitcoins have been created means that the CPU power indeed has been expended, hence the term *proof-of-work*. Mining is done by so-called mining nodes, or briefly *miners*. Although this conception of mining being quite intuitive it is not completely accurate. In the Bitcoin blockchain mining refers to the creation of new blocks of transactions, so mining new blocks, which goes in parallel with creating new Bitcoins. So, creating new Bitcoins, thus providing an incentive to miners to expend CPU power, is not the only goal of mining. The proof-of-work serves the following two overarching goals:

1. It accomplishes that a block, after the cryptographic puzzle has been solved, gets written to the blockchain.
2. It accomplishes that it becomes extremely difficult to make other changes to the blockchain than those that are based on consensus among the nodes in the blockchain.

Since there must be consensus among the nodes in the blockchain network about which node will be allowed to write a block the proof-of-work requirement is called a consensus mechanism. Of all the features of a blockchain as mentioned at the beginning of this chapter the immutability of the blockchain database stands out, making it the shared single-source-of-truth. Figure 1 depicts the chain of blocks that is being held together by a series of hashes in the Bitcoin database. Note that these blocks are sequentially recorded, hence the importance of a timestamp in each block.

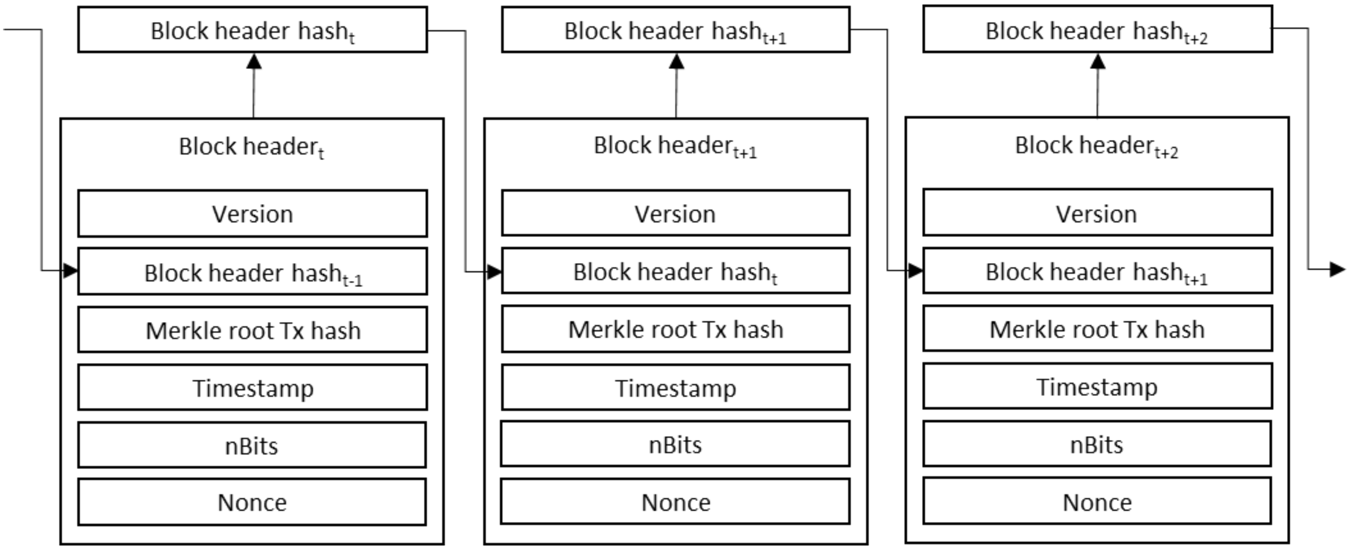

Figure 1: Hashing in the Bitcoin blockchain

A hash (also referred to as a digest) is the outcome of a hash function. Typically input data that is fed into a hash function leads to a fixed length hash. The type of hash that is used in a blockchain is a so-called *cryptographic* hash. Such an algorithm allows easy verification that the input data after hashing equals a known hash, but if the input data is unknown, it is deliberately difficult to reconstruct the input data by knowing the stored hash value. That is why such a hash is also called a mathematical trapdoor. In the Bitcoin blockchain the Secure Hashing Algorithm 256 (SHA-256) is used. Given some specified input data and consistent use of the SHA-256 this input will always lead to the same hash. Only if the input changes, the hash will change. The SHA-256 produces a 256-bit hash (32-bytes) in hexadecimal notation, so 64 digits in the range 0-9 and A-F.[2]

The Merkle root Tx hash is the digest of all the transactions in the block. By calculating the hashes of all pairs of transactions in the block (in case of an odd number the last transaction is hashed with a copy of itself) and doing the same with the resulting pairs of hashes ultimately the Merkle root hash that fully describes all the transactions in a block is calculated as one 32 bytes hexadecimal number. If one transaction is modified, the Merkle root hash changes too.

Now consider the following. In the Bitcoin blockchain a fraudster wants to change a transaction in a block that has already been recorded in that blockchain. This means that the Merkle root hash needs to be changed, and as a result also the Block header hash of the block the transaction is part of. But if a new block has already been recorded after the block with the fraudulent alteration the Block header hash of that new block also has to be changed. And if that block is followed by another newly recorded block then the Block header hash of that block has to be changed too. Given that calculating a hash from input data is easy it would not be difficult for the fraudster to change all the subsequent Block header hashes until the Block header hash of the most recent block is changed and hence the blockchain from the block with the fraudulent transaction onwards. To avoid tampering with the database and hence make changing all the subsequent Block header hashes difficult the Bitcoin blockchain uses proof-of-work.

As indicated before proof-of-work is expending CPU power to solve a cryptographic puzzle by trial and error. Nakamoto (2008) wanted a system in which on average every 10 minutes a block of transactions could be recorded in the blockchain. To accomplish this the Bitcoin core software sets a target hash with a challenge. The challenge for a miner is to find such a number (a Nonce: number used once) that after hashing with the other data from the block header produces a hash that is smaller than or equal to the target hash. If a miner finds such a number then the miner broadcasts this number to the blockchain network so that all nodes can verify that the miner has found a solution. Different from finding the Nonce, verification is easy since this merely encompasses filling in the Nonce and other data from the block header into the SHA-256 hashing algorithm and check if the outcome indeed is smaller than the target hash. If the other nodes agree that the Nonce indeed produces a hash that is smaller than the target hash, then the miner records the block that he has been working in the blockchain. At this point all the transactions in that block are recorded in the blockchain database and the miner receives a reward of (currently) 12,5 Bitcoins plus the sum of the transaction fees that the initiators of the transactions in that block have paid.[3] The transaction that

[2] To distinguish a decimal number from a hexadecimal number, the code '0x' is prepended to the hexadecimal number.

[3] The block reward for the miner who solves the cryptographic puzzle is halved every 210.000th block, which is roughly every four years. This means that the creation of Bitcoins will stop at a point far in the future: in the year 2140 the maximum total of 21 mln Bitcoins will be created, leaving only the transaction fees as an incentive for the miners to provide proof-of-work.

pays out the block reward is called the coinbase transaction, and the reward the coinbase. This is a transaction without an input and this is why this transaction creates new Bitcoins ‘out of thin air’.

After every 2016th block the target hash is calculated from the nBits field in the block header. This field sets the difficulty for solving the cryptographic puzzle. The number 2016 is the outcome of the required 10 minutes to record a block, and a period of two weeks to evaluate the difficulty of solving the cryptographic puzzle (=6 blocks per hour * 24 hours * 14 days). The Bitcoin core automatically increases (decreases) the difficulty if the time that was needed to find a nonce that led to a hash smaller than the target hash is shorter (longer) than two weeks. The value of the nBits field is calculated as follows:

1. Only re-calculate the nBits field if the current block is a multiple of 2016.
2. Take the timestamp of the current block and that of the block that is 2015 blocks before the current block.[4]
3. Calculate the difference between these two timestamps.
4. Multiply the difference by the current nBits converted from compact form to the target it represents and divide the result by 2 weeks.[5]
5. Convert the result to compact form[6] and use that as the new nBits value.

This boils down to a lower target hash when the puzzle was too easy and a higher target hash when the puzzle was too difficult.

The Block header string is the concatenated form of the Version, Previous block header hash, Timestamp, nBits, and Nonce. Figure 2 gives the Block header composition for block 536201 in the Bitcoin blockchain, including the block header hash.

Block #536201		
Block header component	**Notation in blockchain**	**Notation for block header string**
Version	0x3fff0000	0000ff3f
Previous block header hash	000000000000000000289652dbe08d40d3a7ce52561981b564fb98ff062b8bdc	dc8b2b06ff98fb64b581195652cea7d3408de0db5296280000000000000000000
Merkle root hash	fbb64c2cc1d4826afbddea1230995d6325eac8ecc7722ccc1ff7647054c64a9c	9c4ac6547064f71fcc2c72c7ecc8ea25635d993012eaddfb6a82d4c12c4cb6fb
Timestamp	11 Aug 2018, 04:17:13	39556e5b
nBits	172f4f7b	7b4f2f17
Nonce	2,120,628,932	c43a667e
Block header string	0000ff3fdc8b2b06ff98fb64b581195652cea7d3408de0db5296280000000000000000000009c4ac6547064f71fcc2c72c7ecc8ea25635d993012eaddfb6a82d4c12c4cb6fb39556e5b7b4f2f17c43a667e	
Block header hash[7]	0000000000000000002b281e847addb6ee94f0019524922bff6cc960fd41d1be	

Figure 2: Block header hash composition for a block in the Bitcoin blockchain

[4] The Timestamp is the Unix epoch notation of the time a block is recorded.

[5] There are some more rules for exceptional cases to prevent the difficulty to decrease or increase by more than a factor 4 or to prevent having a target that exceeds the maximum target as set for the first ever mined 2016 blocks, including: if the difference is greater than 8 weeks then set it to 8 weeks, if the difference is less than half a week then set it to half a week, and if the result is greater than the maximum target (2^(256-32)-1) then set it to the maximum target.

[6] Compact form is a floating point notation with the first byte being the exponent e and the last three bytes the mantissa c in the formula: c*2^(8*(e-3)).

[7] The Block header hash is calculated by running the Block header through the SHA-256 algorithm twice.

Nodes

Bitcoins are stored in electronic wallets. A Bitcoin wallet can take various forms – one more secure than the other – but the principle for safeguarding one's Bitcoins is always the same. A Bitcoin wallet has an identification code, the wallet ID that acts as a username for any logical access control. For example, a Bitcoin wallet ID may look like: 8a15ne4d-3d6c-6745-d282-da885h64pqf9. To log into this wallet, the user needs to know this wallet ID, a password, and any other code(s) he has enabled for multi-factor authentication. A wallet ID is only used for the login process and cannot be used as an address to send Bitcoins from (an output transaction) or to (an input transaction). It is different from a Bitcoin address, which is a single-use token that is used as the send-from and send-to address in a Bitcoin transaction. A Bitcoin transaction may be compared to sending an email with the message that the sender wants to transfer a certain amount of Bitcoins to a recipient. However, unlike e-mail addresses, people may have many different Bitcoin addresses whereas a unique Bitcoin address should be used for each transaction. An example of a Bitcoin address is: 1B2S4Nf8jD3fshHodzuYhframoQsQaZEcZ. By logging on to a Bitcoin wallet the user has access to one or more Bitcoin addresses that he can use to transfer Bitcoins from or receive Bitcoins to.

It is important to notice that a Bitcoin is a token that can be moved in the blockchain. A wallet does not contain the amount of Bitcoins the owner of the wallet has at a certain moment in time, it rather calculates this position from all the input and output transactions that were done with the Bitcoins that are linked to the Bitcoin addresses the wallet owner uses. For that purpose the wallet only needs to store that part of the blockchain that contains the bitcoin addresses that are used by the wallet owner.

Wallets can run on any computer, from an app on a smartphone to a database with all the transactions that were ever done on the blockchain in a full node or a miner. Wallets do not verify, validate nor relay transactions, which miners do. Full nodes are in between wallets and miners: they relay valid transactions to other nodes so that miners can find the pending transactions to incorporate in a block, and they relay blocks that are created by the miners thus helping to synchronize the blockchain. In the original Bitcoin whitepaper (Nakamoto 2008) did not distinguish between full nodes and miners, considering both to be nodes that could validate and relay transactions, as well as create new blocks of transactions. Gradually, creating new blocks of transactions became a specialist operation that required heavy investments in computer hardware and involved large amounts of electricity usage.[8] As a result full nodes and miners became separate groups of nodes on the Bitcoin blockchain. So, whereas full nodes only relay transactions and blocks to other nodes, miners also validate and create new blocks of transactions to append to the blockchain.

Nakamoto (2008, p.3) lists the steps to run the Bitcoin blockchain network without making a distinction between full nodes and miners. Adjusting these steps to our contemporary world with separate full nodes and miners, and using the terminology as discussed in this section yields the following steps:

1. Wallets broadcast new transactions to all nodes.
2. The miners, after verification of each individual transaction, collect the new transactions into a block.

[8] The electricity consumption of the Bitcoin blockchain in 2017 was 61,4 billion kWh (=61,4 TWh), which is the total electricity consumption of countries like Austria and Switzerland.

3. Each miner works on finding the nonce for the block the miner in question is working on.
4. When a miner finds a proof-of-work, it broadcasts the block to all nodes (the other mining nodes and the full nodes).
5. Nodes only accept a block if all transactions in it are valid and the nonce yields a block header hash that is smaller than the target hash.
6. Full nodes express their acceptance of a block by relaying it to the other nodes; miners express their acceptance of a block by working on the next block in the chain, using the hash of the accepted block as the previous hash.

From this it should be clear that recording a block can only take place by a miner that has acquired the right to write that block through proof-of-work. This also implies that a fraudulent alteration to a previously written block requires the fraudster to redo the proof-of-work of the block he is changing but also of all the blocks after the block with the alteration. For example, suppose currently the miners are working on block #533064. A fraudster wants to change a transaction in block #533061 so that he can double-spend the bitcoins in that transaction. He then has to redo the proof-of-work for all the blocks that have been mined so far, so blocks #533061 till 533063, which will take on average 30 minutes (3 blocks*10 minutes). However, the other miners keep on trying to mine new blocks with valid transactions from block #533064 onwards. Two important rules in the Bitcoin core are that new valid blocks are always appended to the longest chain of blocks (and after this has been done the entire blockchain accepts this chain as the shared single-source-of-truth) and that a new block can only be mined if the previous block has been mined (as a result of the hash of the previous block being part of the cryptographic puzzle to solve for the current block). So, in case a fraudster – who can only be a miner – is trying to redo the proof-of-work of the block he wants to change and all the following blocks that have already been written to the blockchain, he has to do this faster than the other miners who are already ahead of him since they are working on block #533064, and continue with the subsequent blocks at an average pace of 10 minutes per block. Given that the other miners, just like the fraudster, have fast CPUs to deliver the proof-of-work it is extremely difficult, if not impossible, for the fraudster to catch up with the other miners and create his own version of the blockchain that becomes the longest chain. It can easily be seen that the more blocks have been written after a block that contains an invalid alteration the more difficult it becomes to make such an alteration. After 6 new blocks have been mined and hence written (these are called confirmations) it is impossible to make changes to an older block, but in practice this is already the case after 3 newly mined blocks (Figure 3).

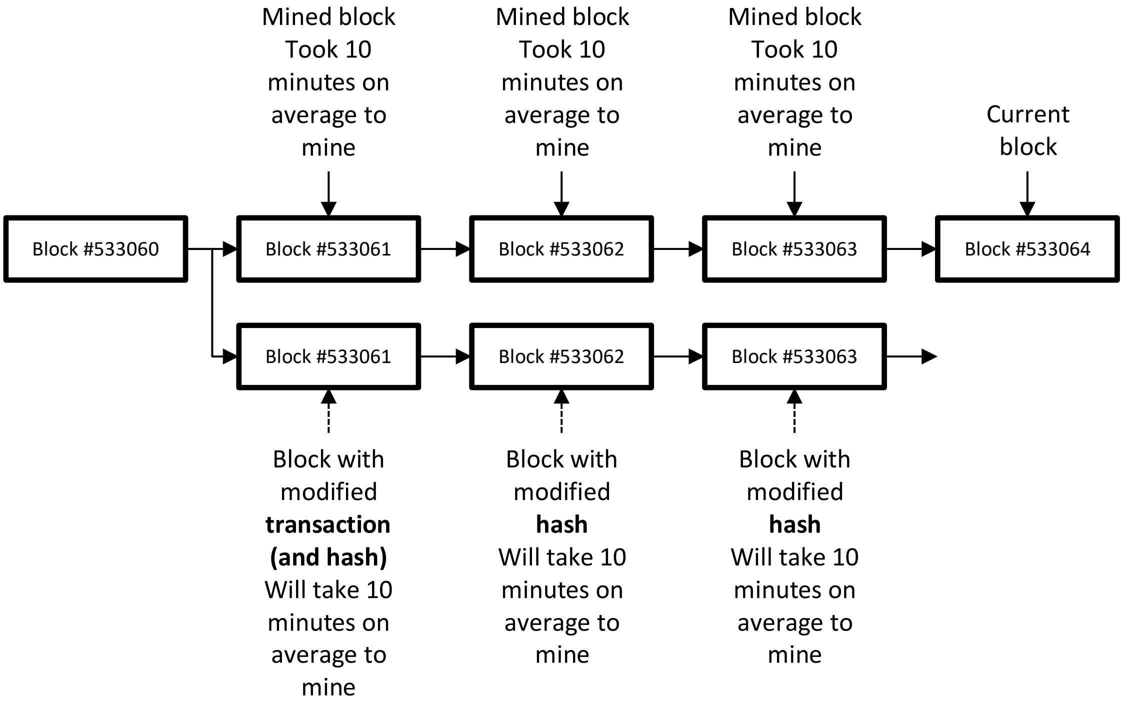
Mined block Took 10 minutes on average to mine
Mined block Took 10 minutes on average to mine
Mined block Took 10 minutes on average to mine
Current block
Block #533060
Block #533061
Block #533062
Block #533063
Block #533064
Block #533061
Block #533062
Block #533063
Block with modified **transaction (and hash)** Will take 10 minutes on average to mine
Block with modified **hash** Will take 10 minutes on average to mine
Block with modified **hash** Will take 10 minutes on average to mine

Figure 3: Proof-of-work to redo if a previously mined block is modified

In the Bitcoin blockchain proof-of-work or a majority of 51% of the hashing power is needed to validate blocks. This means that if a fraudster wants to change a block, thus creating his own version of the blockchain, he can only accomplish this with a majority of the hashing power to provide the necessary proof-of-work. If he has 51% of the hashing power or more, he can make any changes to the blockchain (in practice this turns out to be a slightly higher percentage). Hence it is crucial for the trust in the blockchain that no single miner has 51% or more of the hashing power. Currently the largest miner (BTC.com) has an estimated 29,6% of the hashing power with the largest three miners all being located in China and together having an estimated 52,4% of the hashing power (Tuwiner 2018). Therefore, the term China risk is used to indicate that there is a risk that the Chinese miners join forces and take over the Bitcoin blockchain. However, these miners most likely do not have any intention to do a 51% attack on the network since this would be extremely costly but moreover would completely undermine the trust in the network and render Bitcoins (including theirs) and the mining rigs[9] that they heavily invested in worthless. Yet, on the other hand it would be naive to think that a concentration of hashing power in just a few miners that moreover are located in one single country is not at odds with Satoshi Nakamoto's original idea of a completely democratic currency.

Open source

The Bitcoin core is open source software. By downloading and installing it on his computer a user can unilaterally decide that he wants to make his computer a node in the Bitcoin network. Such a node does not require exceptional computing power nor an enormous amount of disk space since the

[9] A mining rig is hardware that contains application specific integrated circuits (ASICs) that will only do the mining for one specific native token. So, a Bitcoin mining rig cannot do the mining for Ether, and vice versa.

software only serves to support full nodes and not mining nodes, and transactions only consist of an input address, an output address and the amount to be transferred, which consumes only limited memory space.

The Bitcoin core, as most other distributed ledger software, is not managed on-chain but on an open source software platform. This creates a conventional form of transparency that generally is considered safer than privately developed software.

3. Digital signatures, tokenization and oracles

In a distributed ledger each transaction must have a cryptographic digital signature that unlocks the funds from that transaction. Only the person who has the appropriate private key can create a valid digital signature and only a person who has his own appropriate private key can verify that the digital signature is valid (Diffie & Hellman 1976). This ensures that funds can only be spent by their owners. A system of digital signatures is based on public and private keys. Since an electronic coin is a chain of digital signatures, a brief explication of private and public and how these relate to digital signatures is needed.

Imagine that Alice and Bob use a Bitcoin wallet service (for example by downloading an app to their smartphone). As they download the wallet, they are assigned a private and a public key by the wallet, which are stored in the wallet software on their behalf. Alice and Bob can use their private and public keys to exchange information secretly, even over the public Internet provided that the public keys that they exchange and that are completely visible for the outside world are derived from the same encryption algorithm. Note that encryption with keys is different from hashing, since the encrypted data can be converted back to the original information (decrypting) through the use of the keys. As such encryption is an application of hashing. Alice and Bob choose a hashing algorithm and agree on the parameters of that algorithm, say 3^x mod 17, where x is a private key or a combination of a private key and a public key.[10] Suppose Alice has private key 7 and Bob has private key 6. Since these are private keys they don't show these to anybody. Using the chosen algorithm Alice arrives at a public key of 11 (=3^7 mod 17) that she can publicly send to Bob, and Bob of 15 (=3^6 mod 17) that he can publicly send to Alice. Filling in the values of the public (11 and 15) and private keys (7 and 6) leads to a shared secret of 8 (=11^6 mod 17, which equals 15^7 mod 17) that both Alice and Bob know because it was communicated via the Internet. Note that nobody else was able to intercept and read this shared secret because it was communicated using public keys that were hashed from private keys. So, by creating a public key out of a private key an ingenious system emerges whereby encryption is done with the private key of the sender and decryption with the private key of the recipient. This system can be tweaked for creating digital signatures to be sent along with a message. A valid digital signature gives a recipient reason to believe that the message was created by a known sender (authentication); that the sender cannot deny having sent the message (non-repudiation); and that the message was not altered in transit (integrity). This works as follows:

1. Alice sends a message in non-encrypted form to Bob.
2. Alice also calculates a hash from that document and encrypts it with her private key, resulting in a digital signature.
3. Alice sends the digital signature to Bob.
4. Bob calculates the hash from the document that he received from Alice.
5. Bob decrypts the digital signature using Alice's public key.

[10] Hashing algorithms are mostly based on modulus (MOD) functions.

6. Bob can now verify that the received hash from the document equals the decrypted digital signature.

In a distributed ledger verification is programmed in the protocol that manages the ledger.

As indicated a coin, as an instantiation of a token, is a chain of digital signatures. In many distributed ledgers tokens are needed to make the system tick, whether it be as an incentive for nodes to maintain the ledger, as a digital representation of virtual or physical assets, or both. When a token serves as a digital representation of assets, tokenization becomes an important part of the functioning of the distributed ledger, which creates its own dynamics regarding the interaction between the real world and the ledger.

Tokenization is a method that converts rights to an asset into a digital token. With digital assets tokenization is fairly easy since the only thing that needs to be done is linking the digital assets to the also digital token. This can be as simple as creating a digital signature from the digital asset in combination with some private key. When the digital assets changes ownership, the token also moves from the old to the new owner with only a limited risk that the assets and the token become decoupled. With physical assets this is different since the link between the physical asset and the token must be continuously safeguarded and in many instances it is extremely difficult to uniquely identify a physical asset with a tag that is inseparably connected to the asset. For example Everledger provides a platform for recording expensive diamonds in a blockchain (Kemp 2017). To tokenize each diamond the unique characteristics of the diamond must be determined and attached to the diamond. A code that is attached to the diamond in the form of a laser inscription on the girdle will generally not be sufficient to guarantee that it is the right diamond and not a falsification or an otherwise cheaper piece. In addition to a laser inscription on the girdle also physical characteristics of the diamond must be measured to arrive at a digital thumbprint for each diamond. Such a thumbprint consists of 40 metadata points, including the laser inscription and the stone's color, clarity, cut, and carat weight. These digital thumbprints are then written to the blockchain.

There are various classifications of cryptographic tokens varying from a simple dichotomy (cryptocurrencies and other tokens) to a whole multidimensional typology as presented by Euler (2018). At this point this chapter will not discuss the entire richness of cryptographic tokens in theory but will in the following sections on use cases refer to certain types of token that may be suitable in the designated distributed ledgers.

Oracles are the measurement systems that collect data from real life to feed these into a distributed ledger and that receive data from distributed ledgers to initiate pre-defined actions in real life. As such oracles are closely related to the Internet of Things. Oracles can be classified into inbound and outbound, and into software and hardware devices. Examples of inbound software oracles are: temperature, inventory level, cash receipt, price change, and a train delay. Examples of inbound hardware oracles are: RFID/NFC chips, GPS, WiFi, drones, and movement sensors. Examples of outbound software oracles are: sending an order confirmation, placing an order, making a journal entry, transferring money, and populating picking list. Examples of outbound hardware oracles are: switching on a heating, opening a lock, moving a robot, launching a drone, and picking goods from the warehouse. Since oracles live in a world outside of the distributed ledger they introduce failure points. Currently it is being investigated if and how trustless computation oracles can mitigate this issue (Teutsch 2017).

4. Extending blockchain to distributed ledger technology

Although blockchain is the most widely known term it is more accurate to speak of the more generic term distributed ledger technology (DLT). DLT includes blockchain but there are many blockchain variations that make use of DLT without having all the characteristics a typical blockchain has. This section discusses the potential of smart contracts for DLT applications, alternative consensus mechanisms in other distributed ledgers than the blockchain, and ultimately arrives at a classification of distributed ledgers.

Smart contracts

Unlike the name might imply a smart contract is not a legal contract but rather a piece of program code that initiates at least one action if the conditions in that contract are met. An example of a smart contract is the execution of a payment after ordered goods have been received and a certain period of time has elapsed. A smart contract is not necessarily built upon a blockchain. As a matter of fact the concept of smart contracts was proposed by Szabo (1994, 1997), long before the idea of decentralized cryptocurrencies on a blockchain was launched by Nakamoto (2008). However, by building a smart contract on a blockchain its execution becomes guaranteed without having the need for calling upon a trusted third party or a judiciary system to enforce compliance with the contract. Hence, a smart contract running on a blockchain can execute business logic using tamper-proof technology that is upfront compliant and unstoppable.

The Bitcoin blockchain can be used for smart contracts, but since Bitcoin was not developed for other applications than secure transactions between parties without having the need for a trusted third party Bitcoin-based smart contracts are quite conservative and don’t allow for smart contracts that account for the richness of most real-life business transactions. Ethereum is a blockchain that was specifically designed for running smart contracts that go beyond the applications the Bitcoin blockchain originally was designed for. In the Ethereum blockchain any smart contract is possible, including an investment fund, a land registry, an accounting system with cross-organization transaction validation, a voting system, an exchange for energy, a tracking and tracing system for products within their supply chains, and a token that can only be spent to acquire designated assets. Given the versatility of the Ethereum blockchain the question arises why Ethereum has not become the main blockchain platform at the expense of Bitcoin. The simple answer is that versatility comes with a price: by being conservative in its applications Bitcoin is extremely secure whereas Ethereum has sacrificed some security to become more versatile and flexible. Smart contracts may have bugs in them that make them vulnerable to all kinds of attacks and fraud schemes. For that reason smart contracts, unlike the Ethereum blockchain itself, need to be subject to regular audits.

Although, as indicated, smart contracts are not legal contracts they are based on legal code. Because smart contracts are program code, the more standardized and rule-based the legal code, the greater the potential for high quality program code in smart contracts (McKinney et al. 2018).

Alternative consensus mechanisms

A consensus mechanism is needed to secure that only nodes that have received the right to write (blocks of) transactions to a distributed ledger can do so and to keep the ledgers synchronized. Distributed ledger technology either makes use of a proof model such as proof-of-work or some variant of Byzantine fault tolerance (BFT) as its consensus mechanism. BFT is the ability of a distributed ledger to function as desired and correctly reach sufficient consensus despite some malicious nodes defaulting or relaying incorrect information to other nodes. Its objective is to minimize the influence these malicious nodes have on the correct functioning of the distributed ledger and on the desired consensus that should be reached by the honest nodes. As explicated, in the Bitcoin blockchain the consensus mechanism is proof-of-work. Since this mechanism requires an enormous amount of electricity that is used to solve, utterly pointless, cryptographic puzzles it is not considered sustainable by many. For that reason alternative consensus mechanisms that still produce sufficient proof or fault tolerance to grant a node the right to write to the distributed ledger are being developed, including proof-of-stake (POS), practical Byzantine fault tolerance (PBFT), and federated Byzantine agreement (FBA).

In proof-of-stake the right to write to the blockchain is given on the basis of a random allocation process that weighs the stake a node has in the total amount of native tokens (for example, Ethers) of that blockchain. In this variant the money that normally would be expended on capital investments (mining rigs for proof-of-work) is now expended on the purchase of the tokens whereas no additional money is spent on energy as in proof-of-work.

In practical Byzantine fault tolerance the assumption is that the amount of malicious nodes in the network cannot simultaneously equal or exceed one-third of the overall nodes in the system in a given time window. The more nodes there are in the network, the more unlikely it becomes that one-third of the nodes are malicious (Castro & Liskov 1999). In essence, all of the nodes in the PBFT model are ordered in a sequence with one node being the leader and the others the backup nodes. The role of the leader switches per consensus round. Nodes need to prove that a message came from a specific peer node, but also need to verify that the message was not modified during transmission, hence the importance of digital signatures.

In federated Byzantine agreement the nodes do not have to be known and verified in advance, participation in the distributed ledger is open, and control is decentralized. Nodes can choose what other nodes to trust. System-wide quorums (the required minimum number of nodes) emerge from choices made by individual nodes.

Depending on the desired type of distributed ledger one consensus mechanism may be more suitable than the other. Consensus models used by distributed ledgers are largely driven by the type of application the ledger expects to support and the threats it envisages to the integrity of the ledger. Typically the permissionless ledgers such as Ethereum and Bitcoin achieve robust consensus among very high numbers of unknown, and hence untrusted peers using computational or memory complexity while sacrificing transaction finality and throughput (Baliga 2017). On the other hand, the permissioned, consortium ledgers such as R3 for banking are less scalable but have a much higher throughput that ensures faster transaction finality. When looking at distributed ledger technology to solve business problems, the scale of the intended network, the trust between participants, performance, confidentiality, and the required disintermediation are important criteria for determining the platform and the consensus model to use.

Classification of distributed ledgers

Due to the immaturity of the research field consensus on blockchain semantics is still lacking (de Kruijff & Weigand 2017). There is a lot of discussion, mainly in blogposts, on the dimensions along which to compare various types of distributed ledger. A much debated issue is the distinction between private and public distributed ledgers on the one hand, and permissioned and permissionless distributed ledgers on the other hand (for example, Kolisko 2018, Kravchenko 2016, BitFury Group 2015). Trying to find common semantics regarding this distinction appears to be quite an onerous task. However, since the transitions between the various types of distributed ledger are mostly not discrete but rather continuous, and moreover many applications take hybrid forms that combine distributed ledgers with traditional centralized databases, understanding the distinction between private and public on the one hand, and permissioned and permissionless on the other hand merely serves as a starting point for the design of distributed ledgers that match the specific problem that needs to be solved.

Ethereum and Bitcoin are permissionless public distributed ledgers. Here permissionless means that there is no designated party that assigns certain nodes the right to validate and write transactions or blocks of transactions to the ledger. In a permissionless distributed ledger the right to validate and write transactions or blocks of transactions is acquired through the consensus mechanism that the distributed ledger uses. In a permissionless system any node can validate and write transactions or blocks of transactions as long as the rules of the distributed ledger's consensus mechanism are followed. It should be noted that just acquiring the right to validate and write transactions is not the incentive for nodes to make substantial CPU capacity available and expending substantial amounts of energy. The real incentive is that nodes can earn native tokens[11], for example Bitcoins or Ethers, by doing so. If in a distributed ledger any node that runs the right software (for example the Bitcoin core software) can broadcast a transaction, i.e. initiate a transaction in the distributed ledger network, then that ledger is considered a public ledger. As represented in Figure 4 there are more types of digital ledger than permissionless public distributed ledgers.

	Broadcasting transactions (private/public)	
Validating and writing (permissioned/ permissionless)	Broadcasting transactions is limited to a pre-defined set of nodes (private)	Any node that runs the right software can broadcast transactions (public)
Validating and writing (blocks of) transactions is granted by a central authority to a pre-defined set of nodes (permissioned)	(2) Permissioned private distributed ledgers	(1) Permissioned public distributed ledgers
Any node can validate and write (blocks of) transactions within the used consensus mechanism (permissionless)	(3) Permissionless private distributed ledgers	(4) Permissionless public distributed ledgers

Figure 4: Classification of distributed ledgers

Some general characteristics distinguish permissionless from permissioned distributed ledgers and private from public ones. Firstly, in a permissionless distributed ledger the needed consensus mechanism is either proof-of-work, proof-of-stake or federated Byzantine agreement because a

[11] A native token is a cryptocurrency that is the sole medium of exchange on a specific blockchain, is tradeable on a market, and hence has value in the real world. Bitcoin and Ether are the most well-known examples of native tokens. Because of their value these tokens serve as the incentive for nodes to validate and write blocks of transactions to the blockchain.

central authority that determines which node is going to write (blocks of) transactions to the blockchain is not needed. Secondly, in a public distributed ledger a token is needed because such a ledger is secured through cryptography (note that a token is a chain of digital signatures that make use of public and private keys) and the token is needed as an incentive for the nodes to maintain the ledger. Thirdly, in a permissionless distributed ledger transactions are clustered into blocks to solve the double-spend problem. In that sense permissionless distributed ledgers are the only true blockchains whereas permissioned ledgers do not necessarily need to make use of blocks since the double-spend problem is solved by a central authority. For that reason the term blockchain should only be used for distributed ledgers that indeed cluster transactions in blocks to be appended to the blockchain. In all other cases the generic term distributed ledger should be used. Corda as used by R3 is an example of a permissioned private distributed ledger that is inspired on blockchain but does not record blocks of transactions but just individual transactions and moreover does not completely synchronize the entire ledger but just the subset that is relevant for the nodes that were involved in the transactions in question. As such Corda is closer to traditional centralized databases than to permissionless public distributed ledgers. Yet, through its partial decentralized set-up it belongs to the family of distributed ledgers.

5. Distributed ledger technologies for managerial purposes

Distributed ledger technology has many managerial applications that are yet unknown. Through experimentation and bold ventures new potential applications are still being discovered. However, there are some promising use cases, such as the already mentioned Everledger, that may serve as anchors for further experimentation and discovery. Moreover, there are also major fields where distributed ledger technology is still unexplored, including: accounting, control, audit, and oversight. This section discusses some current use cases thereby also giving direction to interesting strands for further exploration.

Identity

The Dutch Ministry of Internal Affairs has commissioned research into the combination of digital identity and digital ledger technology. With digital identity so-called zero-knowledge proof can be given about otherwise personal information, including: age, health history, fines, income, and other legal sentences, and payment history. For example, one of the projects was age control while purchasing age-restricted goods, such as cigarettes or alcohol. The owner of a shop only needs to know that the person trying to buy cigarettes or alcohol is older than 18 years, but currently has to ask for a passport or identity card containing much more personal data. Another example is the personal information a healthcare professional needs to know about a patient, such as allergies, prescribed medication, reanimation desired, blood type, person to contact in case of emergency, and health history in general, so that the healthcare professional can provide the right care. An obvious device to use that can replace the many proprietary ID cards (drivers license, insurance pass, bank card, library card, office pass, customer loyalty passes and the like) is the person's smartphone. In a once-only action an individual's smartphone is prepared for identity on mobile device by having her go to the city hall for linking her smartphone via an app to her citizen record at the municipality. By putting such a system in a distributed database, the ecosystem of participants can grow while the person involved can still determine who is able to see what personal data about her.

Land registry

Honduras (using the Factom platform) has experimented with a blockchain-based cadastral register but has stalled the project in 2015. Ghana is still experimenting with an on-chain cadastral register, as are Sweden, the Republic of Georgia, and some local governments such as Cook County in Chicago (Shin 2016). These countries explore the possibilities of DLT for land title registrations for various reasons. For example, Honduras and Ghana did not have proper cadastral records for real estate. As a result there was never any assurance whether or not the transfer of ownership of land and buildings was recorded. With a cadaster, a trusted third party, this problem is solved. But with DLT emerging a more efficient solution might be to create a DLT based cadastral register that serves as the shared single source of truth regarding who owns which piece of land or building.

Tickets (concerts, museums, football games)

Guts tickets provides and app that is built upon a blockchain for selling tickets directly from the performer to the visitor. In addition it maximizes the price a reseller can charge to 120% of the official price. Guts has financed its business via an initial coin offering (ICO) with the Ethereum-based token GET. Payment of tickets does not need to be done in this token. As in most viable DLT use cases payment is done in fiat money, i.c. Euros.

Elections

A vote can be considered a token. If the double spend problem with that token is solved, then a vote can only be casted once. This may be the future of democratic voting systems that have been experimenting with various IT applications but that often failed due to a lack of security. Distributed ledger technology in combination with identity on an electronic device (most likely a smartphone) provides the technical solution.

Track & trace supply chain

Startup Seal had developed a blockchain-based system for tracing branded products back to their origin. By scanning a built-in NFC chip with a smartphone app the potential buyer can see whether or not the product is counterfeited. The same track and trace technology can be used for tracing back agricultural products that are on the shelves in our grocery stores to their origins. The potential buyer will then be able to find out whether there have been any links in the supply chain that are non-compliant with certain corporate social responsibility guidelines or regulations. For example, coffee can be traced back to the plantation where the beans were grown to find out about the labor circumstances and usage of pesticides.

Internet auctions

Internet auctions are organized by a trusted third party, but with blockchain it is possible to buy and sell directly via the internet without parties having to know and trust one another. When digital assets are traded a smart contract can manage the exchange of money and assets completely on chain. In case of physical goods a solution to the trust issue is that the seller ships the goods prior to the payment by the buyer, with a clause (i.e., a programmed business rule) in the smart contract that the payment will automatically be made after the shipment has been delivered by the carrier to the buyer. As is inherent to smart contracts the action of making the payment is unstoppable once delivery has taken place.

Registration and management of intellectual property

Renowned DJ Hardwell has put the rights to his music into a distributed database. He expects that the technology will result in more transparency and a more honest distribution of rights and funds within the foreseeable future. Services such as Spotify, YouTube and Facebook, via his application, will exactly know how much they need to pay for each track. The same approach can be followed for any form of intellectual ownership, including: books, videos, photos, articles, and art. Hardwell sure isn't the only artist that puts a firm belief in distributed ledger technology. British award winning singer Imogen Heap has been working on the Mycelia project for quite some time. Mycelia records the rights to her music in a blockchain and via a smart contract automatically settles payments she (and others in the chain) is legally entitled to.

Personal healthcare budget

When a personal healthcare budget is given to a patient that can only be spent on designated healthcare products, a whole system of checks and balances needs to be put in place for compliance purposes. Usually checking is done retrospectively. Putting the budget in a token on a distributed ledger that can only be spent on transactions the token is programmed for creates a system that leads to spending being compliant by default.

Healthcare files

The Dutch government has developed a working prototype of a distributed database that contains all the healthcare related information per individual patient. The patient himself gives permission to various parties to access his information and the data are recorded in an immutable ledger. Since there are many different parties involved in healthcare applications, a shared (yet permissioned) database greatly enhances the efficiency of healthcare processes.

Micropayments via internet

Micropayments are made possible by third party applications such as PayPal, Amazon Pay, and Stripe. The user is required to create an account in these platforms to be able to make a micropayment transaction. Such a system has several disadvantages, including: high transaction fees, payment delays, intransparency, and complexity of user interfaces in combination with these being proprietary and hence unique for each producer. A distributed ledger that uses smart contract for fair distribution is a feasible solution to these problems. Bitcoin startup Blockstream has released a micropayment processing system that it claims makes it simpler to build bitcoin apps on top of its Lightning Network.[12]

Energy

The Brooklyn Microgrid project is an initiative of a small group of New Yorkers who have solar panels on their rooftops and who want to make as efficiently as possible use of the collected energy by moving surpluses from one participant to other participants who have shortages without having time-consuming and expensive middlemen to manage this process. For that purpose they had a technology partner (energy startup LO3) design and build a private distributed ledger application to exchange energy between the participants in this little ecosystem.

[12] Lightning adds an extra layer to the Bitcoin blockchain to enable cheaper and faster payments but with the same security backing of the Bitcoin blockchain.

Accounting, audit and control

Despite meaningful applications of distributed ledger technology in accounting, audit and control being quite easy to imagine the development of such systems is still lagging. However, given the immutability and hence a near 100% reliability of a DLT based accounting system, auditors and other accountants are now becoming increasingly interested in a concept that has been labeled triple-entry accounting *avant la lettre* by Grigg (2005). It turns out that DLT is a great enabler of triple-entry accounting by creating a distributed database that is managed by each entity in the ecosystem and that serves as the shared single source of truth. By making journal entries not only locally in each entity's ERP system but also in the distributed database a normative position is continuously maintained against which the local databases can be checked. This systems aims to improve the quality of data to a near 100% reliability.

6. Discussion

Given the many (potential) use cases and the discussed DLT related concepts it is now possible to give some guidelines for evaluating if distributed ledger technology is a feasible solution for certain business problems. The overarching question should always be if DLT is taken into consideration because there is a real business problem or because there are political or publicity motives. The phrase 'we need to do something with blockchain' can be heard too often without the speaker really understanding what is meant by blockchain or related technologies. So, a thorough analysis of the problem, as always, is quintessential.

The conditions for DLT to be taken into consideration are:

1. There is a need for a shared database in which all the users are able to read or verify everything, but no single user controls who can write what.
2. There are multiple parties who write data to the shared database, verify or read these data.
3. Those parties are members of different legal or economical entities. Because organizational boundaries are crossed an in-house company solution with bilateral agreements between parties to share data is less efficient than a solution with one (albeit distributed) database.
4. There is no or limited trust between these parties because the parties don't know one another or they have reason to distrust one another.
5. It is not economically or technically feasible to put a trusted third party in place.

If all these conditions are met, a distributed ledger may be considered. The decision to actually make a DLT design also is moderated by various other factors, including: complexity of regulations (regulations that can easily be transformed into business rules are better suited for DLT applications than more complicated regulations), desired reliability (when 100% reliability is not needed, DLT may not be needed either), and the degree of standardization (the more standardization in the process to model in the DLT application, the more viable a DLT solution may be).

Once the choice is made to enter into a DLT application the choice for the type of distributed ledger must be made. This boils down to determining if the ledger should be public or private, permissionless or permissioned, and whether a (native) token is needed to make the application work. It should thereby be kept in mind that the trade-off between confidentiality needs (private, permissioned is better at this than public, permissionless) and disintermediation needs (the more open, public and permissionless, the better disintermediation may work). It also should be noted that hybrid forms of distributed ledgers are much more likely to be successfully implemented than pure forms such as the Bitcoin blockchain. In that sense the boundaries between the various types of distributed ledger are rather fuzzy than well-demarcated.

From the analysis in this chapter some promising classes of use case emerge. In accordance with the designations Greenspan (2016) distinguishes, these classes comprise lightweight financial systems, provenance tracking, inter-organizational record keeping, and multi-party aggregation. *Lightweight financial systems* are aimed at transactions with digital assets in which the economic stakes are relatively low. Putting this in a distributed ledger requires a token to be incorporated. Examples are loyalty points, local currencies (including gift cards), and crowdfunding. *Provenance tracking* also requires a digital token. Such a token is linked to the asset whose provenance must be tracked. The token then travels on the distributed ledger along with the travel of the asset in the real world. Examples include wine, diamonds, brand articles, agricultural products but also purchase contracts and ownership certificates of real estate. *Inter-organizational record keeping* on a distributed ledger does not require a token to be administered. Instead the ledger acts as a tool for administering any type of data (financial or non-financial, quantitative or non-quantitative). Examples include maintaining an audit trail, a contract register and other control registers for checking purposes. Finally, *multi-party aggregation* is closely related to inter-organizational record keeping but whereas the latter is mainly aimed at maintaining a system of checks and balances, the former is merely aimed at making access to data more efficient. The idea is that multiple parties in a DLT ecosystem have designated rights to access the ledger and read, verify or write data. Examples include a system that grants subsidies based on a wide variety of data sources, users having to submit multiple different forms with (almost) identical data to various different parties as in applications for permits, licenses, or exam results.

In the near future, as a result of the enormous attention that is given to blockchain and other distributed ledger technologies, many new and yet unknown applications will see the light. It will be a challenge for practitioners, educators and researchers to join forces in design science, action research, grounded theory and other practical problem solving research paradigms to explore a great variety of DLT projects to move the domain forward.

References

Armasu, L. (2018). *Top 25 Cryptocurrencies By Market Cap*. Available online at: https://www.tomshardware.com/picturestory/778-biggest-cryptocurrencies.html#s2, June 1

Baliga, A. (2017). Understanding Blockchain Consensus Models. Whitepaper Persistent. Available online at: https://pdfs.semanticscholar.org/da8a/37b10bc1521a4d3de925d7ebc44bb606d740.pdf. April

BitFury Group (2015). *Public versus Private Blockchains, Part 2: Permissionless Blockchains*. Available online at: https://bitfury.com/content/downloads/public-vs-private-pt2-1.pdf, 20 October

Castro, M., and B. Liskov (1999). Practical Byzantine fault tolerance. In: *Proceedings of the Third Symposium on Operating Systems Design and Implementation*, USENIX, New Orleans

Dai, J., and M.A. Vasarhelyi (2017). Toward Blockchain-Based Accounting and Assurance. *Journal of Information Systems*. Vol. 31, No. 3, Fall, 5-21

Diffie, W., and M.E. Hellman (1976). New directions in cryptography. *IEEE Transactions on information theory*, IT-22 (6), pp.644-654

Euler, Th. (2018). *The Token Classification Framework: A multi-dimensional tool for understanding and classifying crypto tokens*. Available online at: http://www.untitled-inc.com, January 18

Greenspan, G. (2016). *Four genuine blockchain use cases*. Available online at: https://www.multichain.com/blog/2016/05/four-genuine-blockchain-use-cases, May 10

Grigg, I. (2005). *Triple Entry Accounting*. Systemics, Inc. Available online at: http://iang.org/papers/triple_entry.html

Haber, S. and W. Stornetta (1991). How to time-stamp a digital document. *Journal of Cryptology*, 3(2)

Kemp, L. (2017). *Putting Bling on the Blockchain: the Everledger Story*. Available online at: http://institute.swissre.com/research/library/Rdm_Blockchain_Leanne_Kemp.html. January 25

Kolisko, L. (2018). *In-depth on differences between public, private and permissioned blockchains*. Available online at: https://medium.com/@lkolisko/in-depth-on-differences-between-public-private-and-permissioned-blockchains-aff762f0ca24, 3 April

Kravchenko, P. (2016). *Ok, I need a blockchain, but which one?* Available online at: https://medium.com/@pavelkravchenko/ok-i-need-a-blockchain-but-which-one-ca75c1e2100, 26 September

Kruijff, J. de, and H. Weigand (2017). Understanding the Blockchain Using Enterprise Ontology. In: E. Dubois and K. Pohl (2017). *Proceedings of the 29th International Conference on Advanced Information Systems Engineering*. Essen (Germany), June 12-16, pp.29-43

Lamport, L., R. Shostak, and M. Pease (1982). The Byzantine generals problem. *ACM Transactions on Programming Languages and Systems*, July, 4 (3), 382-401

McKinney, S. A., Landy, R., & Wilka, R. (2018). Smart Contracts, Blockchain, and the Next Frontier of Transactional Law. *Washington Journal of Law, Technology & Arts*. Vol.13, Issue 3, Spring

Mougayar, W. (2016). *The business blockchain, Promise, practice, and application of the next internet technology*. New Jersey: John Wiley & Sons

Nakamoto, S. (2008). *Bitcoin: A peer-to-peer electronic cash system*. Available online at: https://bitcoin.org/bitcoin.pdf

Narayanan, A., J. Bonneau, E.W. Felten, A. Miller, and S. Goldfeder (2016). *Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction*. Princeton University Press

Panetta, K. (2017). *Gartner Top 10 Strategic Technology Trends for 2018*. Available online at: https://www.gartner.com/smarterwithgartner/gartner-top-10-strategic-technology-trends-for-2018/, October 3

Shin, L. (2016). *Republic of Georgia to Pilot Land Titling on Blockchain with Economist Hernando de Soto*. BitFury. Available online at: https://www.forbes.com/sites/laurashin/2016/04/21/republic-of-georgia-to-pilot-land-titling-on-blockchain-with-economist-hernando-de-soto-bitfury/#5df3211544da, April 21

Szabo, N. (1994). *Smart Contracts*. Available online at: http://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/smart.contracts.html

Szabo, N. (1997). Formalizing and securing relationships on public networks. *First Monday*, Volume 2, Number 9, September 1. Available online at: http://journals.uic.edu/ojs/index.php/fm/article/view/548/469

Tapscott, D., and A. Tapscott (2016). *Blockchain Revolution, How the technology behind Bitcoin is changing money, business and the world*. Portfolio Penguin

Teutsch, J. (2017). *On decentralized oracles for data availability*. TrueBit. Available online at: http://people.cs.uchicago.edu/~teutsch/papers/decentralized_oracles.pdf

Tuwiner, J. (2018). *Bitcoin Mining Pools*. Available online at: https://www.buybitcoinworldwide.com/mining/pools/, June 30

Yermack, D. (2017). Corporate Governance and Blockchains. *Review of Finance*, 7-31