Distributed Ledger Technology & Secured Transactions: Legal, Regulatory and Technological Perspectives – Guidance Notes Series

Note 3: Distributed Ledger Technology and Secured Transactions Frameworks: A Primer

May, 2020

WORLD BANK GROUP

Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

Swiss Confederation

Federal Department of Economic Affairs,
Education and Research EAER
State Secretariat for Economic Affairs SECO

The publication of this Guidance Note was made possible due to the generous support of the Swiss State Secretariat for Economic Affairs.

Preparation of this Guidance Note was led by the Secured Transactions and Collateral Registries Team (Pratibha Chhabra, Murat Sultanov, John M. Wilson, Elaine MacEachern, and Luz Maria Salamina) within the World Bank, under the guidance of Mahesh Uttamchandani. The primary technical content was developed by Dr. Andrea Tosato (University of Nottingham; University of Pennsylvania) under the aegis of Kozolchyk National Law Center.

Special thanks to Dr. Marek Dubovec and Dr. Giuliano G. Castellano for their contributions to this report. The authors are indebted to the following peer reviewers for their excellent suggestions and thoughtful contributions to improve the content of this Guidance Note: Harish Natarajan, World Bank Group, PremaShrikrishna, World Bank Group, and Dr. Teresa Rodríguez de las Heras Ballell, Universidad Carlos III de Madrid.

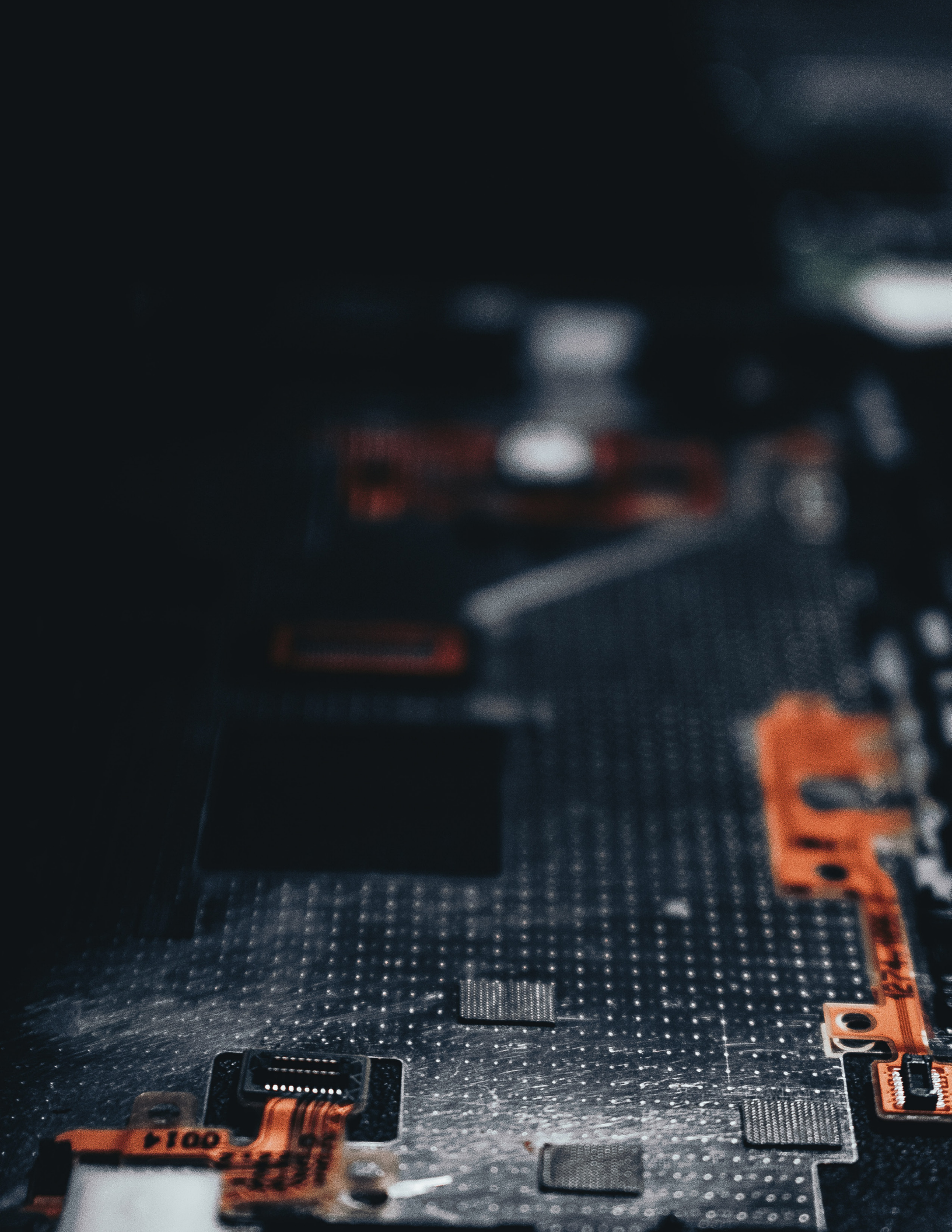
TABLE OF CONTENTS

TERMINOLOGY

For the purposes of this Guidance Note, the following terminology is used:

Distributed ledger technology (DLT) system	A distributed computerized system that enables participants (nodes) to submit, validate, and store information into a database (distributed ledger) that is disseminated, synchronized, and maintained fully or partially across nodes, without the need for intermediaries
Distributed ledger	A database that is disseminated, synchronized, and maintained fully or partially across nodes
Node	A network participant in a DLT system
Distributed script	A set of commands expressed in computer code that is stored, processed and performed by a DLT system
Smart contract	A distributed script that is deployed and processed in a DLT system containing instructions pursuant to which a set of predetermined data entries are submitted to the distributed ledger if preestablished conditions are satisfied
Digital asset	A self-contained and identifiable data entry that is created and recorded in a DLT system for the purpose of serving as an intangible unit of account
Consensus algorithm	An algorithm pursuant to which DLT system participants maintain synchronization and reach agreement regarding whether newly submitted data entries should be added into the distributed ledger
DLT protocol	The software that governs and shapes the operation of a DLT system
Decentralized application	A cluster of distributed scripts that operate in a coordinated manner to perform a predetermined range of functions
Decentralized autonomous organization	A cluster of distributed scripts that operate in a DLT system, as a decentralized organization, independently of any human intervention, pursuant to the software logic encoded in its constituent computer code
Permissioned DLT system	A DLT system in which rules and processes exist to determine which nodes can read the distributed ledger, submit new data entries, and validate data entries submitted by other nodes
Permissionless DLT system	A DLT system in which each node can read the entirety of the ledger, submit new data entries, and validate data entries submitted by other nodes
Private DLT system	A DLT system controlled either by a single person (fully private DLT) or by a group of persons (consortium/federated DLT) who regulate access
Public DLT system	A DLT system that is not controlled by any one person or group of persons and thus neither restricts who may become a node nor imposes identification requirements on participants

Native token	A digital asset that is generated and governed by the protocol of a DLT system
Non-native token	A digital asset that is generated and governed by a decentralized application that exists in a DLT system
Tokenized DLT system	A DLT system that implements digital assets
Tokenless DLT system	A DLT system that does not implement digital assets



EXECUTIVE SUMMARY

This Guidance Note provides a primer on distributed ledger technology (DLT) and highlights the junctures at which this new technology meaningfully impacts secured transactions frameworks. The aim is to identify legal and regulatory hotspots, laying the groundwork for their detailed and exhaustive analysis, which is carried out in the two companion papers (Collateral Registry, Secured Transactions Law and Practice in the Age of Distributed Ledger Technology and Regulatory Implications of Integrating Distributed Ledger Technology in Secured Transactions Frameworks). DLT is an umbrella term that describes distributed computerized systems that enable participants (nodes) to submit, validate, and store information into a database that is disseminated, synchronized, and maintained fully or partially across nodes (distributed ledger). Three facets of this nascent technology have the potential to affect materially modern credit ecosystems.

First, DLT offers the previously unavailable option to operate collateral registries through a distributed system. In the short term, this raises the crucial questions of whether this novel technology possesses the necessary features for such a task and whether it can yield tangible ameliorations compared to technological solutions presently in operation. In the medium term, it elicits the enticing possibility of novel DLT-based collateral registries that have entirely new functions and capabilities.

Second, DLT systems have the capability to create and record special data entries—commonly referred to as "tokens"—that function as units of account (digital assets). The emergence of digital assets prompts an enquiry directed at identifying which of their technical and functional features may bear significant legal and regulatory implications for their use as collateral.

Third, DLT systems can process and execute instructions expressed in computer code (distributed scripts). In secured transactions frameworks, distributed scripts could enable unprecedented automation (for example, automated registrations and self-executing security agreements) the legal and regulatory consequences of which require thoughtful consideration.

Different DLT archetypes exist. The first and most developed archetype is blockchain. In these systems, at regular time intervals, a node collects data entries into a container (a block) and then links it cryptographically to the most recent preexisting block, which in turn is linked to the one before it. Thereafter the new block is propagated to all nodes in the system, which accept it upon independently validating the cryptographic link that connects it to the rest of the chain of blocks (the blockchain). As new blocks are added, the blockchain grows in a linear manner.

Recently, software engineers have devised novel DLT archetypes that differ from blockchain in that they process data entries immediately, rather than aggregating them in blocks at regular intervals. Specifically, each newly submitted data entry is validated and then cryptographically linked to preexisting data entries; this process generates a data structure that comprises multiple, parallel chains of valid data entries, rather than a single, linear chain of blocks.

The hypothetical implementation of a DLT-based collateral registry would require choosing one of these archetypes. Blockchain would offer reliability and predictability but likely suffer from delays due to the time required to aggregate each new block. By contrast, newer DLT archetypes might offer real-time data processing yet present reliability risk due to their recent development.

DLT systems share the following five core characteristics. First, a complete or partial copy of the ledger is distributed to each node; there is neither a central data repository nor a master copy (distributed). Second, all nodes can communicate with each other directly, without the need for a third-party intermediary to enable and coordinate their data exchanges (peer to peer). Third, data entries can be added into the ledger only incrementally; once recorded, a data entry cannot be deleted or modified (append only). This quasi-absolute inalterability renders DLT systems intrinsically tamper resistant (tamper resistant). Fourth, new data entries are recorded into the ledger of a DLT system only if agreement has been reached among nodes regarding its validity, pursuant to a mechanism based on predetermined criteria (consensus driven). Fifth, cryptography ensures authentication, traceability, data integrity, and non-repudiation (cryptographically secure).

These characteristics appear theoretically well suited for collateral registries. Nevertheless, whether, in fact, DLT should be

adopted for collateral registries is a decision that should be made following a comparative assessment between this novel technology and currently implemented technical solutions. The aim should be to determine whether distributed platforms offer tangible improvements over the existing centralized systems.

DLT systems can be markedly diverse depending on their design and architecture. They can provide for different levels of user access and participation privileges based on whether they are private or public, permissioned or permissionless. The processes pursuant to which participants reach agreement regarding whether new data entries should be recorded into the distributed ledger (consensus algorithm) can be radically diverse, with notable repercussions on processing speeds, scalability, and even the carbon footprint of the platform in question. DLT systems can be either tokenized or tokenless depending on whether they implement digital assets. Moreover, the programming language adopted in a DLT system can radically alter the form, substance, and sophistication of the interactions that participants have with each other and the distributed ledger. The profound heterogeneity and variable geometry of DLT systems deserves careful consideration both when deliberating whether to implement this technology for collateral registries and in articulating the regime for the taking of security in digital assets.

DLT systems have the capability to create and record digital assets. These novel intangible assets can be implemented either at protocol level (native tokens) or application level (non-native tokens). Native tokens are implemented by the protocol that governs a DLT system and are integral to its operation (for example, Bitcoin, Ether). By contrast, non-native tokens are implemented and controlled by applications that exist within a DLT system (for example, Augur, OmiseGo). The creation, supply, and destruction of digital assets is governed by computer code and can be either fixed or modifiable. In recent times, digital assets standards have emerged to promote uniformity, transparency, and interoperability; moreover, this process has contributed to the emergence of non-fungible digital assets.

Digital assets have a broad range of possible socioeconomic uses. They can serve as means of payment or as means to obtain access to goods or services, and they may also be used to represent tangible or intangible assets that exist outside of a DLT system. Moreover, digital assets can often have multiple concurrent uses.

Typically, commercial law and financial regulation establish distinct rules for the taking of security in different asset classes (for example, goods, receivables, general intangibles, financial instruments, negotiable documents). The technological features and socioeconomic uses of digital assets might bring uncertainty to their legal and regulatory classification, demanding a pondered assessment of existing categories and possibly the creation of entirely new ones.

DLT systems can record and process distributed scripts. These are computer programs that contain instructions pursuant to which data entries are submitted and processed by a DLT system when preestablished conditions are satisfied. The Ethereum white paper theorized a DLT system that would support a form of advanced distributed scripts that were named “smart contracts.” They were described as systems that automatically move digital assets according to arbitrary prespecified rules. Smart contracts are entirely distinct from legal contracts. Smart contracts are computer programs that enable participants to submit, store, and process code in a DLT system. Legal contracts are agreements between two or more persons that are legally binding if the requirements established by the applicable law are satisfied. However, intersections between smart contracts and legal contracts occur in transactions involving digital assets; for example, a legal contract involving the transfer of digital assets may be performed and enforced through a smart contract.

A decentralized application is a cluster of distributed scripts that operate in concert to perform a predetermined range of operations. The functions that can be performed by decentralized applications are limited only by the computational capabilities of the DLT system upon which they are built. A decentralized autonomous organization is a self-governing decentralized organization that operates in a DLT system through multiple coordinated clusters of distributed scripts. In principle, a decentralized autonomous organization functions independently of any human intervention, pursuant to the software logic encoded in its constituent distributed scripts. This code is designed to replace the rules and structure of a traditional organization, eliminating the need for centralized control.

Although commercial applications are still in their infancy, the combination of digital assets and distributed scripts may result in novel and highly automated transactions between lenders and borrowers that may profoundly affect credit ecosystems, generating a range of legal and regulatory challenges.

INTRODUCTION

Ledgers have played a cardinal role in the evolution of civilization.[1] These information-recording instruments have been integral to the flourishing of international commerce, the development of banking, and the advent of capitalism.[2] In the 21st century, the emergence of distributed ledger technology (DLT) has sparked enormous interest in the new functions that ledgers might assume in the future. It has been suggested that DLT has the potential to reshape almost all facets of society and the economy, including currencies, payment systems, financial markets, property rights, access to credit, supply-chain management, trade finance, and personal identification systems.[3]

Regarding secured transactions frameworks, DLT has brought a mixture of infectious enthusiasm and cautious skepticism. On the one hand, there is hope that this new technology will empower both lenders and borrowers through disintermediation and automation, leading to greater financial inclusion and access to credit for previously underserved or unbanked constituencies. On the other, there is fear that a hasty and cavalier adoption of DLT will obstruct the flow of secured transactions with legal and regulatory uncertainties while simultaneously imposing hefty capacity-building costs and creating socially disruptive implementation challenges.

This Guidance Note provides a primer on DLT, expounding the key elements of distributed ledgers, digital assets, and distributed scripts and concurrently highlighting the junctures at which this new technology comes meaningfully into contact with secured transactions frameworks. The aim is to identify legal and regulatory hotspots, laying the groundwork for their detailed and exhaustive analysis, which is carried out in the two companion papers (that is, *Regulatory Implications of Integrating Distributed Ledger Technology in Secured Transactions Frameworks and Collateral Registry, Secured Transactions Law and Practice in the Age of Distributed Ledger Technology*).[4]

1. DISTRIBUTED LEDGER TECHNOLOGY: A NEW PARADIGM

The locution "distributed ledger technology" is used loosely across the software, financial, and legal services industries to label multiple heterogenous concepts. In this Guidance Note, it will be employed broadly as an umbrella term to encompass distributed computerized systems that enable participants (nodes) to submit, validate, and store information into a database that is disseminated, synchronized, and maintained fully or partially across nodes (distributed ledger).[5]

Every DLT system is governed by a software protocol[6] that shapes its operation. These protocols are typically built upon three mutually supportive pillars. The first pillar establishes the form and substance of the data that can be entered into the distributed ledger. The second pillar sets the parameters pursuant to which nodes communicate with each other. The third pillar determines the process according to which nodes reach agreement and maintain synchronization when new data entries are submitted for inclusion into the distributed ledger.

Three facets of this nascent technology have the potential to affect materially secured transactions frameworks.

1.1 A NOVEL DATABASE TECHNOLOGY

DLT supplies the tools to establish and maintain distributed databases that do not necessitate a central administrator, are cryptographically secure, and support the participation of a potentially unlimited number of non-trusting parties. This technology represents a marked departure from current centralized databases that store information in a single repository—typically supported by back-up facilities—and rely on an administrative entity for their operation, management, and interactions with parties who wish to access information or submit data entries.

Modern secured transactions frameworks implement centralized databases that document the existence of security rights and provide public notice of these proprietary interests to third parties (collateral registries).[7] DLT offers the previously unavailable option to operate collateral registries through a distributed system. In the short term, this raises the crucial questions of whether this novel technology possesses the necessary features for such a task and whether it can yield tangible ameliorations compared to technological solutions presently in operation. In the medium term, it elicits the enticing possibility of novel DLT-based collateral registries that have entirely new functions and capabilities.

1.2 A NOVEL FORM OF PURE INTANGIBLES

DLT systems have the capability to create and record self-contained and identifiable data entries that serve as intangible units of account (digital assets). These special data entries are commonly referred to as "coins," "tokens," or "crypto tokens". Digital assets are "simply code" and exist solely within the confines of the DLT system to which they pertain.[8]

Regarding secured transactions frameworks, the prospective use of digital assets as collateral engenders legal and regulatory challenges. From a private law perspective, it is necessary to determine precisely the body of rules and principles that govern the creation, perfection, priorities, and enforcement of security rights in these assets.[9] From a regulatory perspective, coordination with a variety of existing and novel regimes is required to enable lending against digital assets and their disposal in secondary markets.[10] An organic and cogent approach to these issues is an indispensable precondition if digital assets are to become a valuable source of collateral that supports financial innovation and fosters inclusive access to credit.

1.3 TRANSACTIONAL AUTOMATION

DLT systems have the ability to record and process instructions expressed in computer code (distributed scripts). Participants utilize distributed scripts to add new data entries into the distributed ledger, process available information, and perform transactions, such as transferring digital assets. Moreover, in some DLT systems, multiple distributed scripts can function in concert, creating complex applications and even operating without any human intervention. Overall, DLT systems intrinsically allow for unprecedented levels of transactional automation.[11]

From the perspective of secured transactions frameworks, the automation offered by DLT systems has potentially significant implications. If collateral registries were to implement this new technology, it might be possible for grantors and secured creditors to interact both with each other and the system as a whole in ways that are presently not viable. Similarly, regarding the use of digital assets as collateral, distributed scripts might allow for the automated formation, performance, and enforcement of security agreements. These scenarios raise a multiplicity of legal and regulatory challenges that deserve close consideration.

2. DLT ARCHETYPES

A common misconception is that all DLT systems function in the same manner and share a homogenous data structure. In fact, different DLT archetypes exist, and their processes differ markedly. The first and most developed is blockchain.[12] More recently, software engineers have conceived novel DLT archetypes that aim to overcome some of the structural weaknesses of blockchain and achieve greater flexibility, scalability, and transaction-processing power.

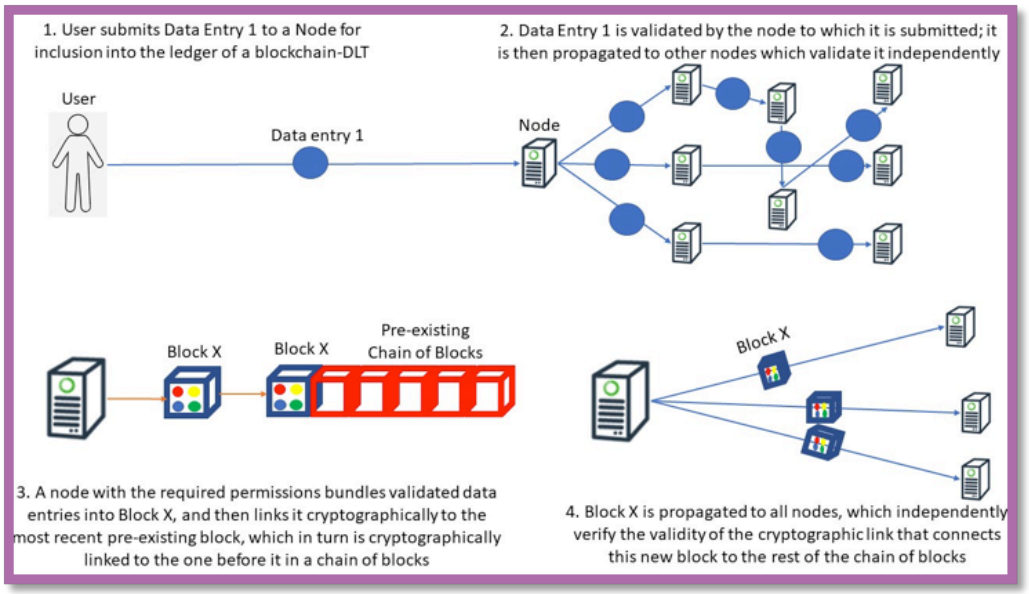
2.1 BLOCKCHAIN

The blockchain-DLT archetype was originally theorized in a white paper authored under the pseudonym Satoshi Nakamoto:[13] Bitcoin: *A Peer-to-Peer Electronic Cash System*.[14] The operation of a blockchain system proceeds in rounds that can be divided into the following four phases:

1. A user who wants to add a data entry into the ledger of a blockchain-DLT submits this information to one or more nodes.
2. The nodes that receive this data entry validate it by applying predetermined criteria and then propagate it to other nodes that validate it independently.
3. At regular time intervals, a node with the required permissions bundles validated data entries into a timestamped data container (a block) and then cryptographically links it to the most recent preexisting block, which in turn is cryptographically linked to the one before it; this process produces an interdependent chain of blocks that originate from the first block (the genesis block) of the blockchain in question.
4. The newly linked block is then propagated to all nodes, which independently verify the integrity and the validity of the cryptographic link connecting it to the rest of the chain of blocks. As new blocks are accepted, the blockchain grows incrementally in a linear manner.

The product of this process is a synchronized and distributed ledger that comprises all the data entries stored in each block of the blockchain. The most successful implementations of the blockchain-DLT archetype are Bitcoin and Ethereum.[15] It should be noted that the rules that govern the creation and propagation of new data entries, which nodes have authority to assemble new blocks and link them to the preexisting chain, acceptance of each new block by all nodes, and the relevant cryptographic primitives vary profoundly across blockchains, depending on their design and architecture.[16]

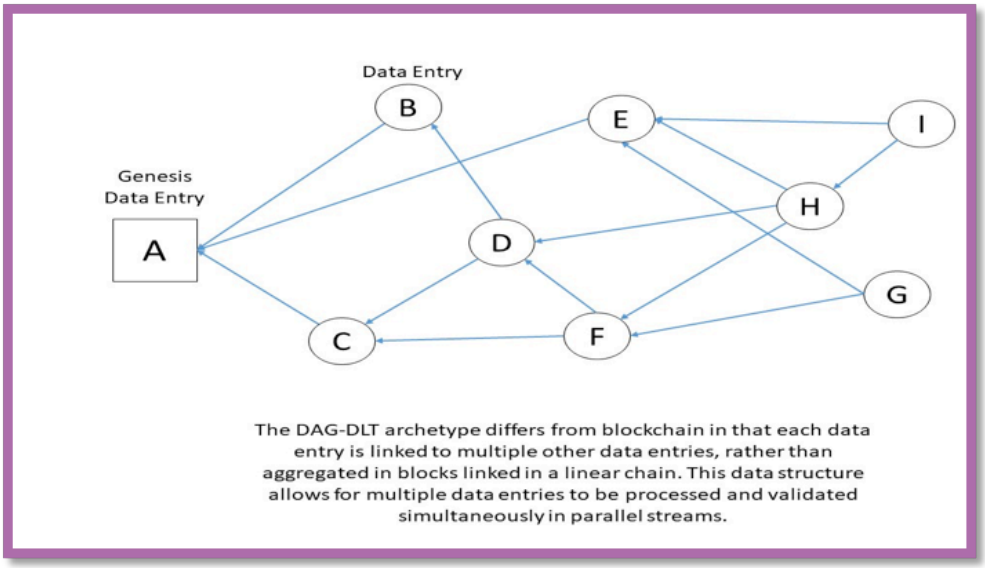
Diagram 1.1: Blockchain-DLT



2.2 NEW DLT ARCHETYPES

More recently, software engineers have devised novel DLT archetypes. They differ fundamentally from blockchain systems in that data entries are not aggregated in blocks sequentially linked in chronological order. One such DLT archetype builds and grows its ledger as a directed acyclic graph of linked data entries.[17] Each data entry—either individually or in small clusters—is cryptographically linked to multiple other data entries (two at a minimum), producing a data structure that comprises multiple, parallel chains of valid data entries, rather than a single, linear chain of blocks. This archetype allows for the submission, validation, and recording of large numbers of data entries simultaneously, far exceeding the throughput capabilities of blockchain-based DLT systems.[18]

Diagram 1.2: Directed Acyclic Graph DLT



Another novel DLT archetype, called Corda, has been developed by the R3 consortium.[19] Corda is a system that enables participants to enter into transactions and provides them with means to keep the relevant information synchronized and cryptographically secure. In Corda, nodes do not hold a complete copy of the distributed ledger that details every transaction that has occurred in the system. The ledger held by each node details only transactions in which they have been directly involved, and this information is synchronized across the ledgers of the relevant parties. Notably, Corda neither aggregates transactions in a linear chain of blocks nor constructs a single distributed ledger that documents all data entries. [20]

Box 1.1: DLT Archetypes and Collateral Registries

The creation of a DLT-based collateral registry would require choosing which archetype to implement. In principle, both blockchain and the newer DLT archetypes allow collateral registries to record security interests and provide public notice. However, these archetypes offer markedly different propositions; each has strengths and weaknesses.

The blockchain-DLT archetype has existed for over a decade. It offers a measure of reliability and predictability, as it has been implemented and operated successfully in large-scale platforms such as Bitcoin and Ethereum. However, this DLT archetype records data entries in blocks that are bundled at predetermined time intervals; this can result in slow processing times and execution delays, depending on transaction volumes and computational power. These shortcomings are a matter for concern, as timeliness and prompt searchability of filings are crucial aspects of modern secured transactions frameworks.[21]

The newer DLT archetypes record and link data entries individually, rather than aggregating them in blocks. Such a data structure bears the promise of real-time execution, prompt visibility, and the capacity to deal with high transaction volumes. However, these newer DLT archetypes are still at a nascent stage; they are under development, and their reliability remains untested.

3. DLT CHARACTERISTICS

DLT systems can be markedly heterogeneous depending on their design and architecture. Nevertheless, they share the following core characteristics.[22]

3.1 DISTRIBUTED

In a DLT system, a complete or partial copy of the ledger is distributed to each node; there is neither a central data repository nor a master copy. Notably, the criteria governing whether each node holds a complete or partial copy of the entire ledger depends on the design and architecture of the DLT system in question.

3.2 PEER TO PEER

All nodes participating in a DLT system can communicate with each other directly (peer to peer), without the need for a third-party intermediary to enable and coordinate their data exchanges. Notably, the fact that all nodes can interact directly with each other does not equate to all of them holding the necessary permissions to validate and add new data entries into the distributed ledger or even read all its content.

3.3 APPEND ONLY AND TAMPER RESISTANT

DLT systems allow for only the incremental addition of data entries into the ledger. Unless a DLT system is maliciously compromised or fundamentally restructured, once a data entry has been recorded, it cannot be deleted or modified. This quasi inalterability of the distributed ledger renders DLT systems intrinsically tamper resistant.

3.4 CONSENSUS DRIVEN

A data entry is recorded into the distributed ledger of a DLT system only if agreement has been reached among nodes regarding its validity, pursuant to a mechanism based on predetermined criteria. This process of convergence is commonly referred to as "consensus," and its dynamics vary profoundly depending on the design and architecture of the DLT system in question.

3.5 CRYPTOGRAPHICALLY SECURE

In a DLT system, cryptography ensures authentication, traceability, data integrity, and non-repudiation. Notably, the extent to which interactions among nodes are encrypted and the type of cryptography employed vary depending on the design and architecture of the DLT system in question.

3.6 DLT CHARACTERISTICS AND COLLATERAL REGISTRIES: THE NEED FOR A COMPARATIVE ASSESSMENT

The characteristics of DLT appear theoretically well matched with the core functions of collateral registries: recording registrations pertaining to security rights and providing public notice of the existence of these proprietary interests. The distributed nature of this technology mitigates the risk of both loss of sensitive data (that is, registrations) and discontinuity of service (that is, inability to search the registry and submit new information) due to a single point of failure (for example, destruction or malfunction of a centralized database). Similarly, the peer-to-peer element of DLT systems is conducive to swift communications between grantors and secured creditors (for example, requesting and submitting a cancellation notice), as well as market participants more broadly. Moreover, the append-only structure of DLT systems, combined with their pervasive implementation of cryptography, offers a level of embedded security and tamper resistance that is desirable to ensure data integrity. Notably, the combination of all these features might be especially attractive in jurisdictions where market particiapants have been historically reluctant to place their trust in the authorities in charge of the collateral registry.

Nevertheless, whether DLT should be adopted for collateral registries should be determined on the basis of a detailed comparative assessment between this novel technology and the technical solutions currently implemented. Specifically, it needs to be objectively determined whether operating collateral registries through a distributed platform offers tangible and marked improvements over the existing centralized systems.[23]

4. DLT DESIGN AND ARCHITECTURE

DLT systems can be profoundly diverse depending on their design and architecture. The operational dynamics, user privileges, and permitted interactions can all be markedly different depending on the structure adopted by the DLT system in question. This heterogeneity is especially challenging in the context of secured transactions frameworks and calls for a functional, rather than formalistic, approach when developing legal rules and regulatory regimes for this new technology.

4.1 PUBLIC/PRIVATE, PERMISSIONLESS/PERMISSIONED DLT SYSTEMS

DLT systems can be either public or private.[24] Moreover, they can be either permissionless or permissioned.[25]

A public DLT system is not managed by any one person or group of persons. Such systems neither restrict who may become a node nor impose identification requirements on their participants.[26] A private DLT is managed either by a single person (fully private DLT) or a group of persons (consortium/federated DLT) who retain control over access to the system.[27]

In a permissionless DLT system, each node can read the entire ledger, submit new data entries, and validate data entries submitted by other nodes. By contrast, in a permissioned DLT, rules and processes determine which nodes have the power to perform each of the aforementioned operations; notably, in such systems, nodes that hold the permissions required to validate new data entries control the ledger.

A public DLT can be either permissionless or permissioned. In the former case, every participant will be able to read the ledger, submit data entries, and validate data entries submitted by others. Such a system will be completely trustless and entirely reliant on nodes reaching consensus before accepting any new data entry into the ledger.[28] A permissioned public DLT will allow all nodes to read the distributed database, yet it will have restrictions and controls to determine which nodes can propose and validate newly submitted data entries. [29]

In a similar vein, a private DLT can be either permissioned or permissionless. In a private DLT that is permissionless, any person granted access to the system by its managers will be able to read the ledger, submit new data entries, and validate new data entries.[30] By contrast, in a private, permissioned DLT, the managers of the system will establish precisely the extent to which each node can read the ledger and submit and validate data entries.[31]

Diagram 1.3: Public/Private, Permissionless/Permissioned DLT Systems

Public and permissioned	Public and permissionless
Voting platforms (Voatz) Currency exchanges (Ripple) Digital identity (Sovrin	Currencies (Bitcoin) Distributed computing (Ethereum, Waves)
Private and permissioned	**Private and permissionless**
Supply chains (Marco Polo, Voltron, Circular) Insurance compliance data (OpenIDL) Healthcare (Change Healthcare) Financial services (Interbank Information Network, IIN)	Trustless workflows (LTO Network) Mobile applications (Monet) Decentralized applications (Holochain)

4.2 CONSENSUS ALGORITHMS

DLT systems require a set of rules (consensus algorithm) pursuant to which nodes maintain synchronization and reach agreement regarding whether newly submitted data entries should be added to the distributed ledger.[32] The aim of a consensus algorithm is to ensure that all nodes arrive at the same result, even in the presence of malicious participants and technical disruptions.[33] The complexity of each consensus algorithm depends on the design and architecture of the DLT system in question. For example, in a private DLT that comprises multiple nodes of which only one—Alpha—has the permissions to validate new data entries, the consensus algorithm will be uncomplicated; all participants must agree to every new data entry propagated by Alpha. By contrast, a public DLT that comprises large numbers of anonymous nodes all holding the permissions required to validate new data entries will necessitate a complicated consensus algorithm that maintains synchronization among nodes while weeding out erratic behavior. Flaws in the consensus algorithm of a DLT system can result in a variety of critical failures, such as rendering it impossible to update the ledger, slow processing of newly submitted inputs, and even corruption of data.

Below, a short description is provided of the most widely adopted and successful consensus algorithms presently in use.

- *Proof of work (PoW)* consensus algorithms create a contest among participants—commonly described as "miners"—to earn the right to add a new validated data entry to the distributed ledger. This competition involves resolving a computational problem that is based on the current state of the ledger.[34] The first participant to find the solution attaches it to the data entry that they wish to add into the distributed ledger and then broadcasts the solution and data entry jointly to the whole network. Other participants first validate that the mathematical solution provided is correct and then accept the new data entry, adding it to their copy of the ledger. At this point, the competition starts afresh. A reward is provided to the participant who resolves the computational problem, providing an incentive to participate in the system.

  PoW consensus algorithms do not require identification of participants and can accommodate thousands of nodes. However, the time and effort required to solve the computational problem delay transaction processing. Crucially, PoW consensus algorithms impose high costs in terms of computational resources and energy consumption; this causes a range of negative externalities and can have a profoundly detrimental impact on the environment.[35] PoW consensus algorithms are used predominantly in public, permissionless DLT systems.

- *Proof of stake (PoS)* consensus algorithms award the power to add a new data entry to the ledger among competing participants based on a probabilistic selection that is proportionally determined by their "stake" in system.[36] The rationale supporting this consensus algorithms is that participants with the largest investment into the system will have the strongest interest in ensuring its undistorted and efficient operation.

  PoS consensus algorithms do not require participant identification and can scale to accommodate thousands of nodes; moreover, in stark contrast to PoW consensus algorithms, they are not especially resource intensive and thus do not pose meaningful environmental concerns. Nevertheless, PoS consensus algorithms generally present difficulties in ensuring consistency of outcome across participants and thus may struggle to ensure that data entries are rapidly recorded into the ledger. PoS consensus algorithms are used predominantly in public, permissionless DLT systems.

- *Practical Byzantine fault tolerant (pBFT)*[37] consensus algorithms provide that new data entries must be submitted to a leader node preselected either randomly or pursuant to defined criteria. The leader node sends the data entry to all participants, who in turn communicate with each other to confirm that they have received the same proposed data entry. When a defined number of confirmations (quorum) is reached, all participants update their ledger, and the system is ready to record new information.[38]

  pBFT consensus algorithms allow for rapid and definitive agreement among nodes; they are not resource intensive and thus have a very small carbon footprint. However, they require leader nodes to be validated and identified;[39] moreover, they face scalability challenges due to the exponential rise in confirmation messages as the number of system participants increases. Notably, pBFT consensus algorithms are adopted predominantly in private, permissioned DLT systems.

Diagram 1.4: Consensus Algorithms Features

	PoW	PoS	pBFT
DLT permissions	Permissionless/ permissioned	Permissionless/ permissioned	Permissioned
DLT management	Public/private	Public/private	Private
Data entry processing speed	Low	Medium/high	High
Scalability	High	High	Low
Operational costs and environmental impact	High	Intermediate	Low
Implementation	Bitcoin, Ethereum (Homestead)	Peercoin, Ethereum (Serenity)	NEO, Hyperledger Fabric, Ziliqa

Box 1.2: DLT Consensus and Collateral Registries

Although consensus algorithms vary greatly, an empirical overview of current DLT projects reveals that PoW and PoS consensus algorithms are used predominantly in public, permissionless DLT systems, whereas pBFT algorithms are implemented in private, permissioned DLT systems. This is a direct consequence of each algorithm's features, strengths, and weaknesses.

Accordingly, if a DLT-based collateral registry were designed as a private, permissioned system, pBFT consensus protocols would likely be the option of choice. By contrast, if such a system were designed as public, permissionless DLT, it is probable that either a PoW or PoS consensus algorithm would be implemented.

4.3 TOKENIZED AND TOKENLESS DLTS

DLT systems can be tokenless or tokenized. In a tokenless DLT system, the function of the distributed ledger is to record data that documents phenomena external to the system in question.[40] In tokenized DLT systems, the purpose of the distributed ledger is to record the existence and the transfers of digital assets, either exclusively or alongside other data entries unrelated to these coins or tokens. Notably, whether a DLT system is designed as tokenized or tokenless is a design choice that profoundly impacts the type of consensus algorithm that may be adopted.

Box 1.3: Tokenized/Tokenless DLT Systems and Collateral Registries

The core functions of collateral registries require the capability of recording and amending information regarding security rights and making it available to searchers.[41] Units of account are neither necessary nor instrumental to the operation of this type of public records. Accordingly, a DLT-based collateral registry could be designed as either tokenized or tokenless. A tokenless DLT-based collateral registry would exactly replicate the level of functionality of a current centralized collateral registry. By contrast, a tokenized DLT-based collateral registry could potentially contemplate the tokenization of registrations.[42] Ultimately, whether a DLT-based collateral registry were implemented as tokenized or tokenless would be a choice that would depend on the policy objectives pursued by the relevant decision makers and the socioeconomic conditions of the jurisdiction in question.

4.4 CRYPTOGRAPHY

Cryptography is an essential component of DLT systems.[43] It ensures that only authorized parties can access (confidentiality) and modify (integrity) information. Moreover, it provides the tools for all authentication processes, including identification of participants (entity authentication), tracking of communications (data origin authentication), non-repudiation of data entries, and accountability. Core elements of distributed systems, such as peer-to-peer communications, consensus algorithms, and the linking of blocks are all structurally reliant on cryptography.

At present, the vast majority of DLT systems implement some form of asymmetric cryptography (also known as "public key cryptography").[44] Persons who wish to submit data entries are required to use their public/private keys to digitally sign their instructions.[45] Moreover, the power to transfer digital assets—such as coins or tokens—is exercised through public/private key systems.[46]

Regarding secured transactions frameworks, the integral role played by cryptography in DLT systems prompts two considerations. First, the creation and operation of a DLT-based collateral registry would require extensive capacity-building efforts to train both operators and users to utilize the cryptographic technologies in question comfortably. Moreover, such a system would require a robust contingency plan to ensure data integrity and continuity of service in the event of a failure of the relevant cryptographic technology. Second, digital assets are intrinsically linked and reliant on the cryptography of the DLT system in which they exist. If this cryptography is compromised, the value of the associated digital assets is entirely lost.

4.5 PROGRAMMING LANGUAGE

All DLT systems have the ability to process and execute computer programs submitted by participants (distributed scripts).[47] The extent to which these distributed scripts can perform complex operations and actions (for example, data entries involving multiple participants or data entries conditional on a hypothetical event or time lapse) varies depending on the programming language adopted by the DLT protocol in question. For example, the Bitcoin protocol implements a programming language with limited capabilities that does not support complex distributed scripts.[48] By contrast, the Ethereum protocol features a sophisticated programming language that allows for the development and operation of "arbitrary" distributed scripts.[49]

5. DLT ASSETS

DLT systems have the capability to create digital assets. This subsection provides an overview of the technical and functional features of this novel type of pure intangibles, highlighting elements of legal and regulatory significance for their use as collateral.[50] Two caveats must be borne in mind. First, although the categories and terminology considered below are widely adopted, they are neither universally accepted nor homogenously understood by software developers, economists, and lawyers. Second, as the development of digital assets is still in its infancy, it is inevitable that this field will evolve at a fast pace for the foreseeable future.

5.1 NATIVE TOKENS AND NON-NATIVE TOKENS

Digital assets can be implemented at either protocol level (native tokens) or application level (non-native tokens).[51]

Native tokens are implemented by the protocol that governs a DLT system, and they are integral to its operation. For example, the Bitcoin protocol automatically issues native tokens (eponymously named Bitcoins) that are conferred as rewards to nodes that actively contribute to the achievement of consensus within the system.[52] In a similar vein, the Ethereum protocol awards native tokens (named Ethers) to nodes that partake in its consensus process; moreover, Ethers also serve as the currency to pay the fees charged by this DLT system to process new data entries and execute scripts. [53]

Non-native tokens are implemented and governed by applications[54] that exist within a DLT system (for example, Augur, Chainlink). For example, the Ethereum protocol allows users to create applications that issue non-native tokens. These digital assets are connected to the application from which they originate, and they can have a role in its operation, typically serving as participation rights or access to services (for example, OmiseGo, and the Basic Attention Token).

In principle, both native and non-native tokens are suitable for use as collateral.[55] It should be noted that the valuation, monitoring, disposal, and regulatory profiles of native and non-native tokens differ markedly.[56] Native tokens require consideration only of the DLT protocol to which they are associated. By contrast, non-native tokens necessitate a two-tier assessment that analyzes both the underlying DLT protocol and the application from which they originate. The latter is especially problematic, as the originating application might hold the necessary permissions to modify the features (for example, the number of tokens in circulation) and functions (for example, transferability) of the non-native tokens in question, dramatically altering their viability as collateral both from a legal and regulatory perspective.[57]

5.2 CREATION, SUPPLY, AND DESTRUCTION OF DIGITAL ASSETS

The creation, supply, and destruction of digital assets are governed by either the relevant DLT protocol or application, depending on whether they are native or non-native tokens. The determinative parameters that control these events are established in computer code; they can be either fixed or modifiable.

Typically, creation of digital assets occurs in two ways. First, the DLT protocol or the application in question provides that a set amount of digital assets is generated either when the system commences operation or at another moment in time.[58] Second, the DLT protocol or the application in question establishes that a predetermined quantity of digital assets is produced if certain conditions are met, typically in response to actions or events that are deemed relevant to the distributed system in question.[59] It is not uncommon for these two methods of creation to coexist. For example, in Ethereum, several million native tokens were minted de novo when this DLT system was initially founded; concurrently, the Ethereum protocol is designed to award native tokens to participants who contribute to the consensus process by validating and propagating new data entries.

The supply of digital assets can be either capped (deflationary model) or uncapped (inflationary model). From a technical perspective, this is a trivial design choice that is entirely in the hands of the persons in charge of the relevant DLT protocol or application. For example, the Bitcoin protocol establishes that its native tokens are capped at 21 million units. By

contrast, from a functional perspective, whether supply is capped or uncapped is a design choice that profoundly affects the socioeconomic dynamics of the digital asset in question.

The destruction of digital assets (commonly referred to as "burning") can take one of three primary forms. First, the person in control of a digital asset transfers its control to an entity that is irreversibly inactive, effectively removing them from circulation.[60] Second, the person in control of a digital asset deletes their ownership record from the ledger, making it impossible for anyone else to claim it in the future. Third, the application to which certain non-native tokens are attached provides for a kill switch that unilaterally results in their destruction.

The ease of creation, multiplication, and destruction that characterizes digital assets is not unique, as many other forms of tangible and intangible property are subject to similar supply dynamics. However, digital assets are singular in that their existence, scarcity, and abundance can be altered almost instantaneously, without incurring costs. This feature complicates significantly valuation, pricing, and risk assessments for market participants. Moreover, it creates a range of non-trivial challenges for policy and law makers intent on developing a coherent legal and regulatory regime for the use of digital assets as collateral.[61]

5.3 DIGITAL ASSETS STANDARDS

The first DLT protocols implemented digital assets—mostly native tokens—that were profoundly diverse. Their technical features and functional profiles were specific to their appertaining system. Such heterogeneity reduced transparency and completely prevented interoperability. This is readily apparent by comparing the native tokens implemented by some of the oldest DLT systems, such as Bitcoin, Litecoin, Namecoin, Peercoin, NEO, and Monero.

In recent times, DLT systems have started to develop and implement standards for digital assets, with special attention to non-native tokens.[62] The aim is to promote uniformity of processes for creating non-native tokens, as well as their technical features and functions (for example, transferability, traceability, and fungibility). For example, in Ethereum, the ERC-20 standard sets out the parameters for the creation and distribution of a non-native token that, among other things, is transferable, fungible, and easily trackable.[63]

Regarding secured transactions frameworks, standardization of digital assets has material implications for their use as collateral, as it increases the information available to potential secured creditors. Notably, if a grantor offers standardized tokens as collateral, a lender can rely on publicly available information associated with the standard in question to carry out due diligence regarding their features and functions.[64]

5.4 FUNGIBLE AND NON-FUNGIBLE DIGITAL ASSETS

DLT assets were originally designed as fungible. Like gold bars or ancient coins, they were interchangeable and indistinguishable from one another. For example, Bitcoins and Ethers are fungible digital assets. Although this design choice was partly due to technical limitations, it also reflected the original use case of native tokens as virtual currencies.

Recently, DLT software innovation has enabled the creation of non-native tokens that are non-fungible in nature. Analogously to their fungible kin, these digital assets can be transferred, stored, and tracked easily; however, they are unique, indivisible, and can store large amounts of metadata.[65]

5.5 POSSIBLE APPLICATIONS OF DIGITAL ASSETS

Digital assets can have a range of possible socioeconomic applications. Although no universal classification perfectly captures this fast-evolving landscape, the following tripartite categorization has acquired increasing recognition internationally:[66]

1. Digital assets intended to be used as a means of payment for goods or services or as money/value transfer. These are often referred to as "payment tokens" or "cryptocurrencies."[67]
2. Digital assets intended to provide access to an application or service by means of a blockchain-based infrastructure. These are often referred to as "utility tokens."[68]
3. Digital assets intended to represent tangible or intangible assets that exist outside of a DLT system, such as a debt

obligation of, or an equity interest in, the issuer or any other type of movable or immovable property. These are often referred to as "asset tokens" and "security tokens."[69]

Notably, digital assets can often be considered "hybrids" either because they possess the characteristics of more than one of these categories or because they are capable of changing functions when predetermined conditions are satisfied. A notable example of hybrid tokens are stablecoins.[70]

Regarding secured transactions frameworks, the broad and variable nature of the possible applications of digital assets is highly problematic. Typically, commercial law and financial regulation establish distinct rules for the taking of security in different asset classes (for example, goods, receivables, general intangibles, financial instruments, negotiable documents). The chameleonic nature of digital assets might bring uncertainty to their legal and regulatory classification, demanding a pondered assessment of existing categories and possibly the creation of entirely new ones.[71]

6. DLT ACTIONS

DLT systems have the capability to store, process and perform distributed scripts. This subsection describes the actions that are possible by means of these computer programs and highlights their potential impact on secured transactions frameworks.

6.1 DISTRIBUTED SCRIPTS: "SMART CONTRACTS" [72]

Distributed scripts are computer programs that contain commands pursuant to which determinate data entries are submitted to the distributed ledger when preestablished conditions are satisfied. Distributed scripts are characterized by a high degree of automation, as they are self-operating and resistive to outside influences; their computer logic can be described as "if this, then that." Once a distributed script is recorded into a distributed ledger, it remains active until either it is fully performed or it hypotheticals become unrealizable.[73]

Famously, in the white paper introducing Ethereum, Vitalik Buterin theorized a DLT system that would support a novel type of advanced distributed scripts that he named "smart contracts."[74] He described them as "systems which automatically move digital assets, according to arbitrary pre-specified rules" and used the metaphor of "cryptographic 'boxes' that contain value and only unlock if certain conditions are met."[75] Following the success of Ethereum, the implementation of distributed scripts of this nature (smart contracts) has become a main staple of DLT systems.

The conjugation of digital assets and smart contracts has spawned enormous interest. The prospect of controlling and disposing of digital assets—potentially representing currency, tangibles, intangibles, and any other form of property—through distributed scripts that are peer to peer, decentralized, persistent, cryptographically secure, and fully automated has drawn the attention of businesses, consumers, legislators, and politicians. On one hand, there is optimism that this new technological paradigm might yield efficiency gains, reduce transaction costs, usher in new business models, and increase financial inclusion. On the other hand, there is fear that this new technology might promote unfair commercial practices, social inequality, and the wholesale circumvention of existing legal and regulatory safeguards.[76]

Smart contracts and legal contracts are distinct concepts. Smart contracts are computer programs that are stored and processed by a DLT system and submit data entries to the relevant distributed ledger if certain conditions are met. Legal contracts are agreements between two or more persons that are binding and judicially enforceable if the requirements established by the applicable law are satisfied.[77] Accordingly, it should be recognized that smart contracts do not necessarily involve legal contracts and vice versa. Coextensively, it should also be acknowledged that smart contracts and legal contracts will intertwine in transactions involving digital assets.[78]

The interplay between smart contracts and legal contracts can potentially assume a broad variety of configurations. At present, empirical studies[79] show that smart contracts are predominantly used as tools to perform legal contracts.[80] For example, two persons may enter into a legal contract—possibly recorded in a conventional paper document—under which one party undertakes to transfer 2 Bitcoins to the other at a determinate moment in time. In their agreement, the contracting parties may stipulate that the transferor must submit a smart contract to the Bitcoin DLT system that contains the necessary computer commands to transfer 2 Bitcoins at the agreed time. In addition to their use as tools to perform legal contracts, both computer scientists and legal scholars etc suggest that, in the near feature, persons might be able to use smart contracts as all-encompassing instruments that record the terms of their contractual agreements and embody the code through which they are performed and enforced.[81] Indeed, there are multiple DLT platforms that are currently developing such smart contracts for real estate, commodities and other commercial transactions.[82]

Regarding secured transactions frameworks, parties that intend to use digital assets as collateral may choose to rely on smart contracts as a mechanism to automate performance of their security agreement. For example, Alice and Bob could enter into a legal contract under which Bob lends 20 non-native tokens to Alice in return for Alice promising both to repay this debt by a certain date and to transfer control of 5 native tokens to Bob as collateral. Alice and Bob could agree to perform part of their obligations using an escrow smart contract pursuant to which Bob's 20 non-native tokens are released to Alice only

when she has given control of the agreed collateral to Bob.

In similar vein, secured transactions involving digital assets may feature smart contracts not only for the performance of the relevant contractual obligations, but also for enforcement purposes. For example, a lender and borrower may enter into a security agreement pursuant to which native tokens used as collateral must be placed under the control of a smart contract that is set to liquidate these digital assets automatically, if the borrower defaults on their obligations. Notably, this technology would enable parties to predetermine and automate many relevant aspects of this liquidation process, including time, price-point and market.

Persons may even consider using smart contracts to record and communicate the terms of their security agreements, as well as automate their performance and enforcement. For example, Alice may offer loans of non-native tokens to the general public at a specified interest rate, subject to the transfer of a determinate number of native tokens as collateral on the part of the borrowers. For this purpose, Alice could deploy a smart contract that automatically extends such loans upon receipt of the borrower's cryptographically signed instructions and collateral. Such a smart contract would embody the terms of Alice's security agreement, provide prospective borrowers a mechanism for their acceptance, and largely automate performance of the agreed contractual obligations.

In principle, the use of smart contracts in secured transactions frameworks is appealing. This technology has the potential to automate the formation, performance and enforcement of security agreements, simplifying these transactions, reducing transaction costs—such as negotiation, contract-drafting and enforcement expenses—and ultimately promoting inclusive access to credit. However, it should be borne in mind that smart contracts also present challenges. In the first place, the legal regime governing smart contracts is still under development and rife with uncertainty. Across jurisdictions, commentators, courts and lawmakers are currently exploring the extent to which general contract law rules and principles adequately suit this novel technology.[83] In the second place, the automated nature of smart contracts is generally obtained at the expense of reduced flexibility. Notably, the level of precisions imposed by computer code does not suit open-texture contractual terms, such as "good faith" and "reasonableness".[84] Moreover, once they have been deployed, smart contracts cannot be easily altered or adjusted, and reversing their operations is both costly and complex.[85]

6.2 DECENTRALIZED APPLICATIONS

A decentralized application is a cluster of distributed scripts that operate in concert to perform a predetermined range of operations.[86] Decentralized applications typically are open source, implement non-native tokens, have a built-in consensus mechanism, and offer a graphical interface to facilitate interactions with the general public. The functions that can be performed by decentralized applications are limited only by the computational capabilities of the DLT system upon which they are built. Recently, decentralized applications have been developed to provide decentralized data storage, escrow services, and digital assets exchanges.

Decentralized applications have the potential to affect secured transactions frameworks on multiple fronts. First, in line with the broader trend toward automated lending solutions, decentralized applications might be used to automate and decentralize the entire life cycle of a secured transaction, from the issuing of loans to the taking of collateral, its custody, and eventual disposal on default. In a similar vein, they can create decentralized exchanges for native and non-native tokens that, among other things, allow persons to enter into transactions in which digital assets are used as collateral. At present, secured lending decentralized applications remain largely untested in terms of reliability and effectiveness, yet several projects are already in operation and have attracted notable interest.[87]

6.3 DECENTRALIZED AUTONOMOUS ORGANIZATIONS

A decentralized autonomous organization (DAO) is a self-governing decentralized organization that operates in a DLT system via multiple coordinated clusters of distributed scripts. In principle, a DAO functions independently of any human intervention pursuant to the software logic encoded in its constituent distributed scripts. This code is designed to replace the rules and structure of a traditional organization, eliminating the need for centralized control. The ownership structure and participation rules of a DAO are defined by its original creators and can vary substantially. Famously, the first attempts at creating DAOs have not been successful, yet a considerable amount of resources continues to be invested toward their development.[88]

Although DAOs are still at a very early stage of development, it is possible to envision a future in which they might significantly affect secured transactions frameworks. In principle, DAOs could be designed to operate autonomously as secured lenders and borrowers. This would raise a galaxy of legal and regulatory conundrums, including issues of legal personality, contractual liability, tortious liability, and public policy.

6.4 DLT, ARTIFICIAL INTELLIGENCE, MACHINE LEARNING, AND DEEP LEARNING

Artificial intelligence is the science and engineering of making computers behave in ways that replicate human intelligence.[89] Machine learning is a branch of artificial intelligence. Its aim is to develop algorithms that enable computer programs to improve automatically in the performance of their tasks by parsing datasets to find both common patterns and singularities.[90] In turn, deep learning is a subbranch of machine learning. Its aim is to develop algorithms that enable computers to develop novel solutions to overcome past failures that occurred in the performance of their tasks.[91]

The increasingly sophisticated analytics tools provided by artificial intelligence, machine learning, and deep learning and the large datasets stored in DLT systems have the potential to affect the secured transactions ecosystem in the future. The confluence of these two technology streams is likely to aid decision making, reduce transaction costs, and increase the level of automation throughout the life cycle of secured transactions. Nevertheless, concrete applications are still in development and will require extensive testing.

FINAL REMARKS

This Guidance Note provided a primer on DLT, identifying the junctures at which this new technology comes into contact with secured transactions frameworks. Moreover, it highlighted crucial areas where digital assets and smart contracts are likely to cause overlaps between secured transactions law and other branches of the law—such as financial regulation, corporate law and intellectual property law—giving rise to "commercial law intersections".[92] Three issues emerged as deserving special consideration.

First, DLT introduces a novel distributed model for the operation of databases, raising the question of whether its adoption for collateral registries might be the next step in their technological evolution. Sections 2–4 emphasized that DLT systems can be structured in a variety of different ways (for example, public/private, permissionless/permissioned, tokenless/tokenized), profoundly affecting their operation and user experience. Accordingly, it was suggested that any initiative to implement a DLT-based collateral registry should be conditional on a two-step analysis that first determines the design and architecture that would best suit the task at hand, and then appraises whether such a distributed system would provide any tangible benefits over the centralized systems currently in operation.

Second, DLT systems have the capability to create digital assets that may be deemed a valuable source of collateral to secure performance of voluntary obligations. Section 5 described the technical and functional traits of these novel intangible assets, expounding their heterogenous nature, the processes associated with their creation and destruction, the ongoing trend toward standardization, and the recent emergence of non-fungible tokens. It was suggested that these features render the taking of security in digital assets a source of multifarious legal and regulatory challenges that prompt a thorough reassessment of the existing body of rules and principles.

Third, DLT systems can record and process distributed scripts pursuant to which data entries are submitted and processed by a DLT system when preestablished conditions are satisfied. These computer programs can perform a broad range of increasingly sophisticated operations. Section 6 described the technical nature and functionalities of smart contracts, distributed applications, and distributed autonomous organizations, revealing the high level of transactional automation that they offer to market participants. Although commercial applications are still in their infancy, it was observed that the combination of digital assets and distributed scripts may result in novel and highly automated interactions between lender and borrowers that may profoundly affect credit ecosystems, generating a range of legal and regulatory challenges.

REFERENCES

1 M. Weber, General Economic History (1927); F. L. Nussbaum, A History of Economic Institutions of Modern Europe (1933), 160; J. Schumpeter, Capitalism, Socialism and Democracy (1943), 123; R. Mattessich, Prehistoric Accounting and the Problem of Representation: On Recent Archeological Evidence of the Middle East from 8000 B.C. to 3000 B.C., Accounting Historians Journal 1987, 71–91.

2 J. Gleeson-White, Double Entry (2011); B. S. Yamey, Scientific Bookkeeping and the Rise of Capitalism, The Economic History Review 1949, 99–113.

3 M. J. Casey & P. Vigna, The Truth Machine (2018); P. DeFilippi & A. Wright, The Blockchain and the Law (2018); K. Werbach, The Blockchain and the New Architecture of Trust (2019).

4 Legal and regulatory aspects concerning the implementation of DLT in the secured transactions frameworks are considered in Collateral Registry, Secured Transactions Law and Practice in the Age of Distributed Ledger Technology (IFC Guidance Note, 2020) [hereinafter DLT-STCR Law and Registry Note] and Regulatory Implications of Integrating Distributed Ledger Technology in Secured Transactions Frameworks (IFC Guidance Note, 2020) [hereinafter DLT-STCR Regulation Note], respectively.

5 See Nat'l. Inst. Of Standards & Tech., Blockchain Technology Overview, IR8202 (2018), https://doi.org/10.6028/NIST.IR.8202; H. Natarajan, S. K. Krause, and H. Luskin Gradstein, Distributed Ledger Technology (DLT) and Blockchain, FinTech Note 1, World Bank Group, Washington, DC, 2017; I. Bashir, Mastering Blockchain: Distributed Ledger Technology, Decentralization, and Smart Contracts Explained, (2018), 32; E. Ganne, Can Blockchain Revolutionize International Trade? World Trade Organization (2018); Legal Framework for Distributed Ledger Technology and Blockchain in Switzerland, the Federal Council Report (2018), 17.

6 Throughout this Guidance Note, the word protocol is used exclusively to refer to the software governing a DLT system to avoid confusion with other software elements often described with this same word (for example, consensus protocol).

7 A detailed analysis of the legal implication of taking digital assets as collateral and the integration of DLT in collateral registry is provided in DLT-STCR Law and Registry Note, supra note 4.

8 In the words of SEC Director William Hinman, "A digital asset itself is simply code." William Hinman, Digital Asset Transactions: When Howey Met Gary (Plastic), Remarks at the Yahoo Finance All Markets Summit: Crypto (June 14, 2018), https://www.sec.gov/news/speech/speech-hinman-061418.

9 DLT-STCR Law and Registry Note, supra note 4, at 7–17.

10 See, generally, DLT-STCR Regulation Note, supra note 4.

11 The range of actions that can be performed via scripts executed on a DLT system will be considered in Section 6 below.

12 The literature on the history of Bitcoin has grown exponentially in the past years. Ex multis, see Tapscott Blockchain Revolution: How the Technology behind Bitcoin Is Changing Money, Business, and the World (2nd ed.) (2018); M. J. Casey & P. Vigna, The Truth Machine (2018); P. DeFilippi & A. Wright, The Blockchain and the Law (2018); K. Werbach, The Blockchain and the New Architecture of Trust (2019).

13 For a journalistic enquiry on the identity of Satoshi Nakamoto, see https://www.nytimes.com/2015/05/17/business/decoding-the-enigma-of-satoshi-nakamoto-and-the-birth-of-bitcoin.html.

14 Available at https://bitcoin.org/bitcoin.pdf.

15 As of October 2019, the Bitcoin blockchain comprises over 600,000 blocks, while the Ethereum blockchain has almost 9 million blocks.

16 DLT design and architecture will be considered in Section 4 below.

17 For an overview of this DLT archetype, see S. Lee, Explaining Directed Acyclic Graph (DAG): The Real Blockchain 3.0 (2018), available at https://www.forbes.com/sites/shermanlee/2018/01/22/explaining-directed-acylic-graph-dag-the-real-blockchain-3-0/#74ac9aa0180b. For an exhaustive technical analysis, see F. Benčić & I. Žarko, Distributed Ledger Technology: Blockchain Compared to Directed Acyclic Graph, 2018 IEEE 38th International Conference on Distributed Computing Systems, 1569–1570.

18 IOTA, Hashgraph, and NANO are examples of DLT systems based on the directed acyclic graph DLT archetype.

19 See R. G. Brown, The Corda Platform: An Introduction, available at https://www.corda.net/content/corda-platform-whitepaper.pdf, and M. Hearn, Corda: A Distributed Ledger, available at https://www.corda.net/content/corda-technical-whitepaper.pdf

20 Ibid.

21 DLT-STCR Law and Registry Note, supra note 4, at 13–14.

22 For a more detailed discussion, see H. Natarajan, S. K. Krause, and H. Luskin Gradstein, Distributed Ledger Technology (DLT) and Blockchain, FinTech Note 1, World Bank Group, Washington, DC, 2017.

23 For this comparative analysis, see DLT-STCR Law and Registry Note, supra note 4, at 7–17.

24 See E. Ganne, Can Blockchain Revolutionize International Trade? World Trade Organization (2018); V. Buterin, On Public and Private Blockchains, Ethereum blog, August 7, 2015, available at
https://blog.ethereum.org/2015/08/07/on-public-and-private-blockchainblockchains

25 See T. Swanson, Consensus-as-a-Service: A Brief Report on the Emergence of Permissioned, Distributed Ledger Systems, available at https://www.ofnumbers.com/2015/04/06/consensus-as-a-service-a-brief-report-on-the-emergence-of-permissioned-distributed-ledger-systems/, and A. Berke, How Safe Are Blockchains? It Depends (2017), available at https://hbr.org/2017/03/how-safe-are-blockchains-it-depends.
26 Notable examples are Bitcoin and Ethereum.
27 Notable examples are Hyperledger Fabric and Quorum. The Libra project is a recent example of a private DLT system.
28 Bitcoin is a notable example.
29 For example, Ripple affords unrestricted read access and data entry submissions yet restricts validation through gateways.
30 FastTrackTrade is a notable example.
31 Hyperledger Fabric is a notable example.
32 Consensus is a challenge not just in DLT systems but in all distributed computing and multi-agent systems. See G. T. Nguyen & K. Kyungbaek, A Survey about Consensus Algorithms Used in Blockchain, Journal of Information Processing Systems 2018; A. Baliga, Understanding Blockchain Consensus Models, available at https://pdfs.semanticscholar.org/da8a/37b10bc1521a4d3de925d7ebc44bb606d740.pdf.
33 See M. Fischer, N. Lynch, & M. Paterson, Impossibility of Distributed Consensus with one Faulty Process, no. MIT/LCS/TR-282, Massachusetts Inst of Tech Cambridge Lab for Computer Science (1982).
34 It should be noted that, typically, the computation problem based on the current state of the ledger requires all participants who wish to compete to retain an updated version of the ledger.
35 See M. Blinder, Making Cryptocurrency More Environmentally Sustainable, Harvard Business Review, March 2018, available at https://hbr.org/2018/11/making-cryptocurrency-more-environmentally-sustainable.
36 PoS consensus algorithms have numerous variants. Heterogenous criteria are adopted to measure each participant's stake. There is also substantial diversity in the logic governing the probabilistic selection process. Differences also exist regarding whether participants must prequalify to be eligible to update the ledger.
37 Seminally, M. Castro & B. Liskov, Practical Byzantine Fault Tolerance, OSDI 1999.
38 There are numerous variants of the pBFT consensus algorithms that differ in how they stage and coordinate communications across participants; see A. Baliga, Understanding Blockchain Consensus Models, available at https://pdfs.semanticscholar.org/da8a/37b10bc1521a4d3de925d7ebc44bb606d740.pdf.
39 Recently, a variant of the pBFT consensus algorithm called Federated Byzantine Agreement (FBA) has risen to great notoriety. FBA seeks to retain the strength of pBFT while concurrently allowing for a public accessibility to the DLT system in question. See D. Maziers, The Stellar Consensus Protocol: A Federated Model for Internet-Level Consensus, available at https://www.stellar.org/papers/stellar-consensus-protocol.pdf.
40 Corda is a notable example of a tokenless DLT system.
41 DLT-STCR Law and Registry Note, supra note 4, at 7.
42 Id. at 4, 17.
43 For a complete analysis of the cryptographic primitives supporting DLT systems, including hash functions and symmetric cryptography, see H. Natarajan, S. K. Krause, and H. Luskin Gradstein, Distributed Ledger Technology (DLT) and Blockchain, FinTech Note 1, World Bank Group, Washington, DC, 2017.
44 For a primer, see D. Stinson & M. Paterson, Cryptography: Theory and Practice, chapter 4 (2018).
45 For example, if Bob wishes to send a digitally authenticated message to Alice, he can attach to the relevant text his signature encrypted using his private key. Upon receipt of this message, Alice will consider it authentic only if she is able to decrypt the cyphered signature using Bob's public key. Both Bitcoin and Ethereum implement such a system.
46 In an asymmetric cryptography system—such as Bitcoin or Ethereum—participants agree to a cypher and then retain two keys. These two keys are mathematically linked, yet it is computationally infeasible to derive one from the other. Data encrypted using one of these two keys can be decrypted only using the other. Each participant openly discloses one of their keys (the public key) while preserving confidentiality of the other (the private key). Transferring a digital asset: If Alice wants to send an encrypted message to Bob, she encrypts the text using Bob's public key and then sends the encrypted data to him. Upon receipt, Bob deciphers the encrypted data using his private key and obtains Alice's message.
47 Distributed scripts are discussed extensively in Section 6 below.
48 The Bitcoin script programming language is called Script; because of the limited range of computation operations that it provides for, it is described as non-Turing complete.
49 The Ethereum script programming language is called Solidity, and it is described as Turing complete to indicate that it emulates a Turing machine. It allows for the design and operation of arbitrary distributed scripts—that is to say that they can perform any type of computational operation. See The Annotated Turing: A Guided Tour through Alan Turing's Historic Paper on Computability and the Turing Machine; V. Buterin, A Next Generation Smart Contract and Decentralized Application Platform, GITHUB, https://github.com/ethereum/wiki/wiki/White-Paper [https://perma.cc/F8NP-8EZ6].
50 An exhaustive analysis of the legal and regulatory profiles of the use of digital assets as collateral is carried out in the DLT-STCR Law and Registry Note, supra note 4, and the DLT-STCR Regulation Note, supra note 4, respectively.
51 The transition of the Binance Decentralized Exchange from ERC-20 non-native tokens on the Ethereum DLT to their own DLT system and native tokens (BNB) offers a good case study to understand the difference between native and non-native tokens; see https://www.binance.com/en/support/articles/360027291331.
52 In fact, the Bitcoin protocol automatically awards Bitcoins to a node that resolves the mathematical problem involved in the PoW

consensus; this is the incentive that motivates miners to participate in this DLT system. See N. Sakamoto, Bitcoin: A Peer-to-Peer Electronic Cash System, available at https://bitcoin.org/bitcoin.pdf.
53 See V. Buterin, A Next Generation Smart Contract & Decentralized Application Platform, available at http://blockchainlab.com/pdf/Ethereum_white_paper-a_next_generation_smart_contract_and_decentralized_application_platform-vitalik-buterin.pdf.
54 See subparagraph 6.2.
55 DLT-STCR Law and Registry Note, supra note 4, at 21–23, and DLT-STCR Regulation Note, supra note 4, Section 6.
56 For a detailed analysis of the regulatory and governance profiles of secondary markets for native and non-native tokens, see, generally, DLT-STCR Regulation Note, supra note 4, Section 7.
57 For an analysis of the relevant regulatory implications, see DLT-STCR Regulation Note, supra note 4, Section 7.4. For a private law assessment, see S. Cohney, D. Hoffman, J. Sklaroff, and D. Wishnick, Coin-Operated Capitalism, Columbia Law Review 2019, 591–676; see DLT-STCR Law and Registry Note, supra note 4, at 21.
58 This process of creating digital assets is commonly described as "minting." The public offering of DLT assets created in this manner is commonly described as an "initial coin offering" (ICO); see DLT-STCR Regulation Note, supra note 4, at 12–13.
59 The typical example of such a creation process is "mining" in the Bitcoin protocol. A Bitcoin is awarded to the node that resolves the computational puzzle created by the PoW consensus.
60 For example, in Ethereum, this can be achieved by transferring native tokens to a predetermined account that has no owner such as the genesis account.
61 See DLT-STCR Regulation Note, supra note 4, at 20.
62 For example, in Ethereum, software developers have crafted and refined several open-source standards for non-native tokens, such as ERC-20, ERC-223, ERC-721, ERC-777, ERC-1337, ERC-1155. Recently, the International Token Standardization Association has started a database aimed at tracking all token standards available in different DLT systems; see https://itsa.global/what-we-do/#Tokenbase.
63 See https://eips.ethereum.org/EIPS/eip-20.
64 See S. Cohney, D. Hoffman, J. Sklaroff, and D. Wishnick, Coin-Operated Capitalism, Columbia Law Review 119 (2019), 591–676.
65 In Ethereum, a standard has been developed (ERC-721) to standardize the features and streamline the creation process of non-fungible, non-native tokens; see erc721.org. A notable example of the implementation of this standard is cryptokitties; see https://www.cryptokitties.co/about
66 For an exhaustive and detailed analysis of existing regulatory classifications, see DLT-STCR Regulation Note, supra note 4, at 27.
67 Bitcoin and Bitcash are notable examples.
68 Filecoin is a notable example.
69 Maecenas and KWHCoin are notable examples.
70 For a complete analysis of stablecoins, see DLT-STCR Regulation Note, supra note 4, Section 6.2.
71 See DLT-STCR Law and Registry Note, supra note 4, at 18–24. For an analytical framework to determine the applicable regulatory regimes when digital assets are taken as collateral, see DLT-STCR Regulation Note, supra note 4, Section 5.
72 The term "smart contract" was originally coined by Nick Szabo and defined as: "A a set of promises, specified in digital form, including protocols within which the parties perform on these promises." See N. Szabo, Smart Contracts: Building Blocks for Digital Markets 1996, available at https://perma.cc/YC35-2MXQ. Drawing inspiration from the "humble vending machine" Szabo posited that "the basic idea behind smart contracts is that many kinds of contractual clauses . . . can be embedded in the hardware and software we deal with, in such a way as to make breach of contract expensive … for the breacher" See N. Szabo, Formalizing and Securing Relationships on Public Networks, First Monday 1997, available at https://journals.uic.edu/ojs/index.php/fm/article/view/548/469.
73 In greater detail, see H. Natarajan, S. K. Krause, and H. Luskin Gradstein, Distributed Ledger Technology (DLT) and Blockchain, FinTech Note 1, World Bank Group, Washington, DC, 2017; S. Cohney & D. Hoffman, Transactional Scripts in Contract Stacks, Minnesota Law Review 2021, available at https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3523515.
74 Interestingly, Vitalik Buterin subsequently came to regret this choice of terminology, recognizing that it created great confusion, rather than bringing clarity; he stated that "at this point I quite regret adopting the term 'smart contracts'. I should have called them something more boring and technical, perhaps something like 'persistent scripts" https://perma.cc/Q4GY-AAQR
75 See V. Buterin, A Next Generation Smart Contract & Decentralized Application Platform, available at http://blockchainlab.com/pdf/Ethereum_white_paper-a_next_generation_smart_contract_and_decentralized_application_platform-vitalik-buterin.pdf.
76 This problem is often referred to as "regulatory arbitrage"; see DLT-STCR Regulation Note, supra note 4, at 11.
77 See J. M. Sklaroff, Smart Contracts and the Cost of Inflexibility, University of Pennsylvania Law Review 2017, 263.
78 See K. Werbach & N. Cornell, Contracts ex machina, Duke Law Journal 2017, 313.
79 M. Bartoletti & L. Pompianu, An Empirical Analysis of Smart Contracts: Platforms, Applications, and Design Patterns, in M. Brenner et al. (eds), Financial Cryptography and Data Security (2017) 494-509; H Surden, Computable Contracts, UC Davies Law Review 2012, 629.
80 See M. Raskin, The Law and Legality of Smart Contracts, Georgetown Law Technology Review 2017, 305; K. Werbach & N. Cornell, Contracts ex machina, Duke Law Journal 2017, 313; J. Cieplak & S. Leefatt, Smart Contracts: A Smart Way to Automate Performance, Georgetown Law and Technology Review 2017, 418; P. Cuccuru, Beyond bitcoin: an early overview on smart contracts, International Journal of Law and Information Technology 2017, 179–195.
81 See J. G. Allen, Wrapped and Stacked: 'Smart Contracts' and the Interaction of Natural and Formal Language!, European Contract Law Review 2018, 317-320.
82 The Corda Enterprise and OpenLaw platforms are notable examples. For an overview see R. Brown, The Corda Platform: An

Introduction available at https://www.corda.net/content/corda-platform-whitepaper.pdf, and OpenLaw, The Smart Contract Stack available at https://perma.cc/4MA6-Y5XS respectively.
83 For a recent example, see United Kingdom Jurisdiction Taskforce, Legal Statement on Cryptoassets and Smart Contracts (2019), available at https://35z8e83m1ih83drye280o9d1-wpengine.netdna-ssl.com/wp-content/uploads/2019/11/6.6056_JO_Cryptocurrencies_Statement_FINAL_WEB_111119-1.pdf
84 See J. M. Sklaroff, Smart Contracts and the Cost of Inflexibility, University of Pennsylvania Law Review 2017, 291–298.
85 S. Cohney & D. Hoffman, Transactional Scripts in Contract Stacks, Minnesota Law Review 2021, available at https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3523515.
86 See V. Buterin, A Next Generation Smart Contract & Decentralized Application Platform, available at http://blockchainlab.com/pdf/Ethereum_white_paper-a_next_generation_smart_contract_and_decentralized_application_platform-vitalik-buterin.pdf. By contrast, for a much broader definition of decentralized applications, see David Johnston et al., The General Theory of Decentralized Applications, Dapps, available at https://github.com/DavidJohnstonCEO/DecentralizedApplications.
87 MakerDAO, Dharma, and BlockFi are notable examples.
88 M. I. Mehar et al., Understanding a Revolutionary and Flawed Grand Experiment in Blockchain: The DAO Attack, Journal of Cases on Information Technology 2019, 19–32.
89 The literature on this topic has grown exponentially in the recent past. See N. Nilsson, Principles of Artificial Intelligence (2014); S. Russell & P. Norvig, Artificial Intelligence: A Modern Approach. (2016); M. Minsky, Steps toward Artificial Intelligence, Proceedings of the IRE 49.1 (1961) 8–30.
90 See T. Mitchell, Machine Learning (1997).
91 See T. Sejnowksi, The Deep Learning Revolution (2019).
92 See Giuliano G. Castellano & Andrea Tosato, Commercial Law Intersections, Hastings Law Journal 2021, available at https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3558378.

Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

Swiss Confederation

Federal Department of Economic Affairs,
Education and Research EAER
State Secretariat for Economic Affairs SECO