International Telecommunication Union

ITU-T Technical Paper

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

(01/2022)

HSTP.DLT-Risk
DLT-based application development risks and their mitigations

Summary

Distributed ledger technology (DLT) implements a system of agreement on a global scale, with a huge potential for the creation of new business models in several areas. However, the development of this kind of system presents social, technological and legal challenges, which are different from the challenges related to traditional systems. Thus, it is important to provide support to developers faced with common problems affecting this platform and avoid situations that put the system objectives at risk. This report describes a common architecture used in DLT-enabled systems, identifies 30 relevant risks in this platform, and discusses ways to mitigate these risks based mainly in lessons learned. Risk analysis can assist in software project decisions, contribute to the choice of platforms for the project, and support the selection of the data to be stored in a decentralized manner. Risks can impact the development, deployment and operation of the final product. Awareness of the risks and ways to mitigate them may improve the prediction and avoidance of many problems, thus increasing projects' success rates.

Note

This is an informative ITU-T publication. Mandatory provisions, such as those found in ITU-T Recommendations, are outside the scope of this publication. This publication should only be referenced bibliographically in ITU-T Recommendations.

Keywords

Blockchain, distributed ledger technologies, DLT-based application development, lessons learned, project, risks.

Change log

This document contains Version 1 of Technical Paper ITU-T FSTP-VS-ECSR "Requirements for event centre server in video surveillance system" approved at the ITU-T Study Group 16 meeting held online, 17-28 January 2022.

Editor:	Suzana Mesquita de Borba Maranhão Moreno BNDES/Geneva University Brazil	E-mail: Suzana.Mesquita@etu.unige.ch
Contributors:	Felipe Curty BNDES Brazil	E-mail: felipecrp@bndes.gov.br
	Vanessa Almeida BNDES Brazil	E-mail: vanessa@bndes.gov.br
	Gladstone Arantes BNDES Brazil	E-mail: glads@bndes.gov.br
	Marcio Onodera BNDES Brazil	E-mail: marcio.onodera@bndes.gov.br

João Perufo
BNDES
Brazil
E-mail: jvperufo@bndes.gov.br

José Almeida
BNDES
Brazil
E-mail: josej@bndes.gov.br

Maurício Filho
BNDES
Brazil
E-mail: mvgfilho@bndes.gov.br

Paulo Henrique Alves
PUC-RIO
Brazil
E-mail: ph.alves@les.inf.puc-rio.br

Gustavo Robichez
PUC-Rio
Brazil
E-mail: guga@les.inf.puc-rio.br

Ronnie Paskin
PUC-Rio
Brazil
E-mail: ronnie.paskin@les.inf.puc-rio.br

Rafael Nasser
PUC-Rio
Brazil
E-mail: rafael.nasser@les.inf.puc-rio.br

Fabíola Greve
UFBA
Brazil
E-mail: fabiola@ufba.br

Javier Ibáñez
Comillas University
Spain
E-mail: jibanez@icade.comillas.edu

Table of Contents

Technical Paper HSTP.DLT-Risk

DLT-based application development risks and their mitigations

1 Scope

This Technical Paper proposes a typical architecture to support the development of systems that employ DLT, identifying the risks associated with these projects linked to this architecture and possible mitigation actions.

2 References

[ITU-T HSTP.DLT-RF] Technical Report HSTP.DLT-RF (2019), *Distributed ledger technologies: Regulatory framework*.

[ITU-T HSTP.DLT-UC] Technical Report HSTP.DLT-UC (2019), *Distributed ledger technologies: Use cases*.

3 Definitions

3.1 Terms defined elsewhere

This Technical Paper uses the following terms defined elsewhere:

3.1.1 DLT oracle [b-ITU-T X.1400]: A service that supplies information to a distributed ledger using data from outside of the distributed ledger system.

3.1.2 fork [b-ITU-T X.1400]: Creation of two or more different versions of a distributed ledger.

3.1.3 smart contract [b-ITU-T X.1400]: A program written on a distributed ledger system which encodes the rules for specific types of distributed ledger system transactions in a way that can be validated and triggered by specific conditions.

3.1.4 wallet [b-ITU-T X.1400]: Software and/or hardware used to generate, manage and store both private and public keys and addresses, which enable distributed ledger technology (DLT) users to transact. Some wallets may interact with smart contracts and allow single and/or multi-signature.

3.2 Terms defined in this Technical Paper

This Technical Paper does not define any particular terms.

4 Abbreviations and acronyms

This Technical Paper uses the following abbreviations and acronyms:

AML Anti-Money Laundering

DAO Decentralized Autonomous Organization

dApp Decentralized Application

DLT Distributed Ledger Technologies

P2P Peer-to-Peer

PoW Proof of Work

VPN Virtual Private Network

5 Introduction

DLT provides a digital web of trust for conducting transactions between unknown peers on the Internet; and implements a system of agreement on a global scale through a state machine replicated in a peer-to-peer network [b-Gre]. The development of DLT systems enables the creation of new business models in various sectors of the economy, such as finance, healthcare, arts, supply chain, digital identity and government [b-Mer] [b-Cas]. DLT also introduces a new development model with the potential to address large-scale challenges in our society, such as trust between anonymous peers, sharing, privacy, public transparency, immutability and auditability [b-Zha].

Although extremely promising, DLT is recent [b-Nak] and is not yet mature and robust. Several challenges related to its infrastructure, network layers and distributed systems, fault tolerance, security, data management, platforms and technology, as well as organizational and social governance, need to be addressed to truly transform society through DLT [b-Gre] [b-Ian].

In the development of DLT-oriented software, aspects of information security, application distribution, architecture, business process modelling and specific metrics must be considered [b-Por]. Because DLT applications are hard to modify after implementation, design errors are in general much more challenging to address than in traditional systems. Additionally, DLT-oriented business modelling often introduces challenges in legal issues, governance and economic incentives that need to be planned and addressed.

The risks and mitigations presented here may be an important input to help decision-making in DLT-based projects, such as platform analysis and choice, implementation of smart contract codes, and selection of the kind of data that can be stored on a decentralized basis.

Existing works [b-Atz] [b-Cas] [b-Che] [b-LiX] [b-Zha] describe risks and are certainly useful, but they do not focus on the project, and are often limited to a specific platform. In this report, an architecture for DLT systems is presented that provides a comprehensive view of the entire development process of a DLT-based application. The report presents 30 relevant risks and their consequences. These risks were identified based on real experience of the authors and on the description of 49 use cases of ITU report [ITU-T HSTP.DLT-UC] and supported by the literature and by actual attacks and vulnerabilities. Additionally, the risks were organized according to the layers of the architecture presented. Finally, the report discusses some actions to mitigate the identified risks.

The remainder of the report consists of three clauses. Clause 6 presents an architecture for DLT systems that is useful for risk analysis. Clause 7 presents the risks that must be considered in project development. The risks are discussed considering the architecture of clause 6. Finally, clause 8 discusses some actions to mitigate these risks.

6 Background

Aiming at a more structured analysis of the risks inherent in the development of a project involving DLT technology, this report presents a very common base architecture of systems that use the technology.

As Figure 1 illustrates, the architecture is divided into layers related to the key intrinsic components of DLT: peer-to-peer (P2P) network, protocol (which includes consensus algorithm), smart contracts, and extrinsic components representing network connection, front-end application, ecosystem in which it is inserted and the governance to which it is subordinated [b-Ant] [b-Gre] [b-Gro].

The P2P network layer is formed by the nodes that will store, propose and validate new DLT states, thus establishing the distributed network, the foundation of DLT. In this network, each node can transmit messages asynchronously to its peers, discover its peers, execute message dissemination mechanisms, synchronize states, among other natural aspects in a point-to-point network. The point-to-point nodes of the previous layer need to establish a common protocol so that network states can

evolve by consensus between nodes and, securely, through cryptographic mechanisms. The risks at these two layers are discussed in clause 7.1.

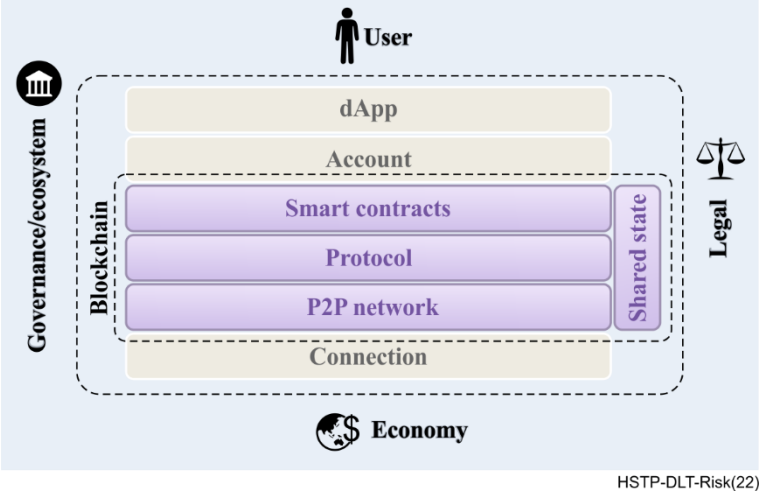

Figure 1 – Example of common architecture of DLT systems

Business rules, coded in a computer program that is executed for proposing and validating the state change discussed in the protocol layer, inherit the power decentralization feature provided by DLT and are executed in smart contracts. The risks associated with this layer are analysed in clause 7.2.

Connection allows peer-to-peer network nodes to communicate, either via the Internet or virtual private networks (VPNs). The communication layer also abstracts the infrastructure risks of isolated nodes, such as intrusion risks and node access control. The risks of this layer are widely debated in the field of computer science and telecommunications, demanding no further comment in this report.

Decentralized applications (dApps) make use of DLT to store data and/or execute smart contracts to ensure business logic and provide to the end user a better user experience. dApps can store information both on and off-chain, as well as executing business logic outside and within the smart contract. DLT platforms generally use accounts to identify their users. An account is represented at least by a pair of cryptographic keys, one public, allowing the identification of the account and another private, which gives its holder autonomy to perform actions on behalf of this account. As mentioned above, the user can relate to the proposed solution through a dApp for a better experience. However, to ensure power decentralization, the user can also directly relate to the smart contract from their account, which implies that it is important to analyse the risks considering these two types of access: direct or intermediated. DApps, account and user risks are covered in clause 7.3.

In addition to the user, other extrinsic aspects are of fundamental importance for the full development of the software and its implementation, such as governance and ecosystem (risks described in clause 7.4), legal (risks detailed in clause 7.5) and economic (risks discussed in clause 7.6). In summary, the proposed architecture presents a big picture of DLT systems. This architecture is essential to classify and understand the risks discussed in the following clauses.

7 Risk evaluation

7.1 P2P network layer and protocol risks

Risk 1 – Power centralization by the consensus algorithm or another network service. The higher the decentralization, the less likely participants are to collude to tamper with on-chain data. The lower the decentralization, the greater the need for adopting cryptographic resources (e.g., use of certifying authorities) to prevent attacks [b-Ban]. For example, attacks to block producers or attacks from the block producers to the network. One reason may happen because of the use of cloud services to host

the nodes[1]. Centralization may be due to the existence of special network nodes, such as master nodes[2], notaries[3] and specialized block producers[4]. These nodes are hotspots of vulnerability and attract denial-of-service attacks and exist on various DLT platforms designed, for example, for greater scalability or privacy, as well as in second layer additions to existing platforms such as the Lightning Network for Bitcoin [b-Poo]. The geographical location of block producers may also raise the risk of geopolitical centralization, especially in cases of permission-less networks.

Risk 2 – Low scalability and congestion. Limitation of the number of transactions executed in parallel, creating the risk that the platform will not support the processing requirements necessary for the business model to function properly. This is the case with Bitcoin's Proof-of-Work (PoW) consensus, which, by design, requires decision-making of transactions in the order of minutes [b-LiX]. In fact, congestion due to near-maximum network capacity has already hampered the operation of some platforms[5] (visited on 27/11/2019). In certain situations, congestion can even be created artificially to prevent new transactions from being sent and to allow the attacker to gain advantages, such as while using an application that needs quick responses. This is the case, for example, with auctions [b-Che].

Risk 3 – Probabilistic confirmation. This is the possibility of a transaction taking too long or even never being confirmed by the network. In theory, the PoW consensus has a probabilistic agreement, and, only after some rounds, offers high probability security for decisions [b-Nak].

Risk 4 – Loss of availability. Although more resilient than centralized solutions, some platforms may become unavailable or incapable of validating new transactions, such as in cases of network partitioning or unavailability of many validator nodes.

Risk 5 – Delays in retransmission of data. Messages propagate in the network layer in a non-deterministic period of time due to delays in data retransmission. This increases the likelihood of state reorganization, possibly invalidating previous transactions.

Risk 6 – Break of cryptographic primitives. Cryptographic primitives could be attacked, especially with the adoption of newer technologies such as quantum computing. Cryptography is a basic technology for DLT operation, such as generating algorithms for public and private keys and hash functions, as well as consensus algorithms for block production and validation [b-Gia].

Risk 7 – Platform failures. Failure to deploy or upgrade platforms can result in unintended effects and introduce vulnerabilities that facilitate attacks such as denial of service (DDoS) [b-Men] and forgery of valid signatures[6].

Risk 8 – Issues in an individual node. The availability and ability of nodes to communicate in peer-to-peer architecture can be threatened by load balancing issues in network deployment, network layer attacks or platform vulnerabilities. For example, an attacker captures a node's connection to other nodes, the attacked node is cut off from the real network, and thus unable to send transactions, which is called an eclipse attack and leaves nodes susceptible to other types of attacks [b-Hei].

---

1 https://thenextweb.com/hardfork/2019/09/23/ethereum-nodes-cloud-services-amazon-web-services-blockchain-hosted-decentralization/ (visited on 2020-05-05)

2 https://www.dash.org/masternodes/ (visited on 2019-11-27)

3 https://docs.corda.net/key-concepts-notaries.html (visited on 2019-11-27)

4 https://eosauthority.com/producers_rank (visited on 2019-11-27)

5 https://www.coindesk.com/loveable-digital-kittens-clogging-ethereums-blockchain (visited on 2022-02-18)

6 http://blog.lekkertech.net/blog/2018/03/07/iota-signatures/ (visited on 2019-11-27)

7.2 Smart contract layer risks

Risk 9 – Smart contract coding errors. These errors are extremely relevant, especially considering the code's immutability feature, and can cause feature freeze[7], unexpected operations, lack of possibility to accommodate future requirements or even introduce vulnerabilities[8]. Errors can occur in the smart contract being developed by the project or in the libraries used [b-Atz] [b-Sat]. Errors can be due to a lack of understanding of how code is executed on the DLT platform. For example, errors caused by transactions being executed asynchronously and by the execution queue being modified in an attack known as front-running [b-Che]. Errors can also be due to failure in software design, possibly due to the designer's poor understanding of the solution to be built or the ability of the technologies to be used.

Risk 10 – Contracts with waste of resources. Codes that are not well optimized are a risk as they can waste a considerable amount of computational and financial resources during their deployment and/or execution. This is especially relevant to permission-less DLTs, where operations are often monetized via crypto-assets [b-Auer].

Risk 11 – Non-execution due to parameter variability. Platform limitations regarding storage, runtime, or any other parameters may prevent a transaction from being executed. When such parameters are variable, there is a risk that they will not meet the requirements at the time of the transaction. In the case of Turing-complete platforms, the problem is even greater because the execution itself causes unavoidable unpredictability. For example, Ethereum has a limit to the maximum amount of computational effort allowed per block, which may even vary over time [b-Zha].

Risk 12 – Untrusted data or rules external to the smart program. Developing a solution often requires off-chain searches for data or rules by a mechanism known as an oracle[9], e.g., digital representations of physical objects or weather services. This use introduces the risk that the data is erroneous, tampered with or with versioning incompatible with expectations.

Risk 13 – Exposing information. An old discussion on free software repeats itself in the universe of DLT. Making public the source code of software increases your exposure to risks, such as coding errors and logical impossibilities that can be exploited. In the case of DLT, to increase product confidence, many solutions choose to make the source code public. Advertising certainly brings greater knowledge to the attacker, but it also brings greater transparency and community knowledge to the improvement of code. Exposing transactions data is also a concern. For example, a company may not want to reveal who are their suppliers and the value of their transactions. In addition to being a legal issue, it may cause competitive and business issues.

7.3 Risks of dApp layer, account and user

Risk 14 – Key management issues. Users can store their keys carelessly, share them with others, use an infected hardware[10], generate them in a fragile way[11][12], get hacked, be persuaded to provide them, or simply lose them. Key loss can also occur directly in databases of custody companies such as exchanges[13]. Exchanges may also be victims of attacks on their IT infrastructure[14].

---

7 https://www.parity.io/a-postmortem-on-the-parity -multi-sig-library-self-destruct/ (visited on 2019-11-27)

8 https://www.coindesk.com/understanding-dao-hack-journalists (visited on 2019-09-27)

9 https://provable.xyz/ (visited on 2019-11-27)

10 https://decrypt.co/26025/rubygems-bitcoin-stealing-software-reversinglabs (visited on 2020-05-05)

11 https://cointelegraph.com/news/blockchain-bandit-has-stolen-45-000-eth-by-guessing-weak-private-keys-report-claims

12 https://www.wired.com/story/blockchain-bandit-ethereum-weak-private-keys/ (visited on 2021-09-07)

13 https://www.bbc.com/news/world-us-canada-47203706 (visited on 2019-11-27)

14 https://www.wired.com/story/hack-binance-cryptocurrency-exchange/ (visited on 2019-11-27)

Risk 15 – User errors. While users should be aware of the risks inherent in the technology used, either directly or through intermediaries (e.g., exchanges), it is very difficult to eliminate usage errors. In this ecosystem, it is common for assets to be controlled solely by their owner, so that there is no intermediary or authority to turn to if an error is made. Given the complexity of working with public and private keys, it is not difficult for a layman to send funds, tokens or other assets to the wrong address. Errors may also occur in the payment of transaction fees. In Ethereum, for example, such fees are considered a structural incentive for miners. The values are set by the user and have no limits, and very low fees can be generated (which results in non-confirmation of the transaction) or absurdly high ones[15]. There is a possibility that the user risk arises from the contract owner himself, who may, for example, wrongly apply a method that destroys the smart contract[16].

Risk 16 – dApp issues. The project's dApp layer solution can be compromised and result in security issues such as misinformation and data or resource theft. An example is an attack on a WEB server that stores the software front end and includes a malicious script or display of false information [b-Che].

Risk 17 – Phishing to a similar page or use of compromised interface software. In this case, users do not lose their private keys, but they may be misled to send resources[17] or disclose information to malicious websites or programs. An example is the case of MyEtherWallet [b-Che].

Risk 18 – Inability to send a transaction to DLT. If the user does not control a DLT node, he must trust that a node will send it on his behalf. Once the user sends the signed transaction, there is no risk of changing the submitted content. However, the input node may choose to censor user transactions; or, the node that the user has purposely chosen (or the node that was configured by default in the software used) may be down and the user may not know how to configure the use of an alternate node.

Risk 19 – Incompatibility of versions. Off-chain software integrations into the dApp layer present versioning incompatibility risks. As a practical example, the introduction of privacy mode in Metamask[18] account management software caused errors in dApps not fitted to the new version unless the user knows how to configure the privacy mode.

7.4 Governance and ecosystem risks

Risk 20 – Lack of trained professionals or knowledge to develop using the platform. Skilled people are needed to enable the development of projects using the platform. In addition, the more knowledge available, whether in tutorials, forums, previous projects, etc., the easier it is to train new professionals.

Risk 21 – Platform does not evolve over time. The evolution of the platform allows for the introduction of technical improvements that may be relevant for use in the project. Some platforms maintain a continuous flow of resources to support maintenance projects and establish a governance structure to facilitate evolutionary maintenance decisions[19] while others do not have this.

Risk 22 – Lack of control over the governance of the project's network/platform and the risk of decentralization of the governance of the application. The evolution of network or platform governance may not meet the demands of a project. For example, a decision may cause a network or platform to stop accepting a particular type of transaction or significantly change how transactions

15 https://medium.com/moatcoin/eth-gas-26d221c5c4c2 (visited on 2019-11-27)

16 https://www.parity.io/a-postmortem-on-the-parity-multi-sig-library-self-destruct/ (visited on 2019-11-27)

17 https://cointelegraph.com/news/network-of-fake-bitcoin-qr-code-generators-stole-45-000-in-march (visited on 2020-05-05)

18 https://metamask.io/ (visited on 2019-11-27)

19 https://eos.io/ (visited on 2019-11-27)

are charged by the network[20]. If the decisions are based in tokens (proof-of-stake consensus algorithm), there is also the risk that token custodians vote following their own interests[21]. Decisions in the network/platform can have a huge impact on the project. Decentralizing application governance implies greater participation in decisions on how to evolve the project at the same time that it introduces the risk of hindering the evolution of the application itself since it makes the decision more difficult.

Risk 23 – DLT forks. Forks can arise due to the difficulty of the governance structure making decisions, either because governance is not adhering to platform objectives or because it is not clear in advance which decisions can be made and how they can be implemented. For example, the fork stemming from the attack on the DAO[22] was caused by community divergence in identifying an error in the contract that allowed the diversion of millions of dollars. Forks may also arise for technical reasons, such as a bug in a new software release or the introduction of a technical upgrade in probabilistic consensus algorithms. Because nodes are managed in a decentralized manner, the community needs to be aware and make the necessary updates[23]. Forks can lead to network partitioning, generate inconsistency, lower network security by possibly decreasing the number of nodes in the resulting network, among others[24].

Risk 24 - Societal risks. A DLT project may create a relevant impact on players of the status quo or introduce systemic risks, and face resistance for different parts of society. This was the case of the project Libra[25], which was initially proposed by Facebook with the goal to create a global stable coin.

7.5 Legal risks

Risk 25 – Lack of proper legislation and regulations where the project will produce effects and repercussions. Inadequacy of existing laws can lead to litigation – administrative, civil or criminal. Examples are compliance with tax and accounting, financial markets, competition, intellectual property, civil liability and contracting legal frameworks, and with public-law mandatory regulations, namely AML and data protection laws[26]. In addition, legal vagueness or vacuum also undermine the potential of using a software project. For example, several projects may not make sense if the legislation does not recognize that DLT records are legally valid or establishes that crypto-assets cannot be freely used. Under the scope of this risk, it is also included the risk of non-compliance with existing mandatory laws because of legal uncertainty or lack of uniform interpretation by public institutions, agencies or judges [b-Ibaa] [b-Ibab].

Risk 26 – Difficulty in determining competent jurisdiction or court. In some decentralized projects, it may be difficult to determine the jurisdiction that will govern the consultations, controversies and conflicts arising out of the use of the DLT product or service. It is important to note whether the states covered by the project address the issue in a narrow-restrictive manner, only enforcing rules regarding facts occurring within the territory of the state, or attractive-expansively, enforcing legislation regarding facts occurring within the territory of the state, but also in other situations where there are connections or links that authorize enforcement, such as participation or relationship with citizens of the state.

---

20 https://blog.aragon.org/istanbul-hard-fork-impact/ (visited on 2020-05-05)

21 https://cointelegraph.com/news/the-steem-takeover-and-the-coming-proof-of-stake-crisis (visited on 2020-05-05)

22 https://coincentral.com/ethereum-classic-vs-ethereum/ (visited on 2019-11-27)

23 https://www.coindesk.com/markets/2019/05/17/unpatched-ethereum-clients-pose-51-attack-risk-says-report/ (visited on 2022-02-18)

24 https://www.academia.edu/39881756/ (visited on 2019-11-27)

25 https://techcrunch.com/2019/06/18/facebook-libra/ (visited on 2021-12-12)

26 http://www.planalto.gov.br/ccivil_03/_Ato2015-2018/2018/Lei/L13709.htm (visited on 2019-11-27)

Risk 27 – Misuse or abuse of application. Such conduct encompasses the use. Users can make uncompliant and eventually criminal use of an application, even if such use was not initially intended by the designers. For example, users can save inappropriate images[27] or text by making use of an application feature.

7.6 Economic risks

Risk 28 – Unexpected variations in costs associated with network use. Costs may be charged, for example, by transaction or by deploying a new application on the network. Unexpected variations in costs can become a risk to application success, even making some business models unfeasible. For example, if an application supports low-value transactions, high usage costs could make it unfeasible. These variations may also occur due to increased network usage. For Bitcoin and Ethereum, the fee to be paid is set by the user who submits the transaction. However, block producers tend to choose high-fee transactions to be processed first. Thus, as the volume of transactions increases, fees tend to rise due to the dispute between users to prioritize their transactions.

Risk 29 – Variations in the remuneration of block producers. Consideration should be given not only to the variation in amounts received, but also to the costs involved, such as storage requirements and computational capacity. Ultimately, such variations can have a fatal impact on the consensus algorithm itself. Lower remuneration causes greater network fragility in the face of attacks like the 51% attack [b-Gre]. In the case of Bitcoin, the tendency for remuneration to be dominated by fees rather than the creation of new bitcoins may be a long-term problem, which makes halving periods particularly relevant [b-Aur]. Other PoW-based algorithms present similar problems.

Risk 30 – Variation in the value of crypto-assets. Not only because variations in crypto-assets may cause exposure to the two previous risks (28 and 29), but also because this is a risk that has its own dynamics[28]. For example, a crypto-asset such as bitcoin may have its value highly affected by market speculation only, with no increase in network usage and no halving occurring.

8 Risk mitigation

Mitigation is a way of reducing the likelihood or the impact of the materialization of a risk. This clause discusses actions to mitigate the risks presented in clause 7.

8.1 Mitigation of network layer and protocol risks

These layers' risks are mostly inherent to the platform being used, whether due to its software's technical architecture, the number of applications currently using the network, the number of attacks that can occur on its infrastructure or the remuneration that block producers receive in comparison to their cost (as well as the remuneration of alternative investments to which block producers may have access). Mitigating the risks of these layers is more effective if project leaders can influence the choice of platform, network, or network parameters to be used; for example by increasing the number of nodes participating in the consensus or improving connection speed between nodes.

The ITU assessment criteria of distributed ledger technologies [b-Bai] can be a good source of parameters to consider when selecting an appropriate platform or a consensus algorithm in a platform which supports pluggable consensus. Each consensus algorithm implements a proposal for decentralization, with consequences on scalability and on the possibility of attacks [b-Bal] [b-Ban].

The consensus algorithm also determines the risk level of probabilistic confirmation and loss of availability (risks 3 and 4). The famous CAP theorem states that partitions cannot be tolerated without losing availability or consistency in the face of network uncertainties. In this case, probabilistic

27 https://www.bbc.com/news/technology-47130268 (visited on 2019-11-27)

28 https://www.coindesk.com/these-3-factors-were-likely-behind-bitcoins-price-surge-to-5k (visited on 2019-11-27)

consensuses, more common in permission-less platforms, ensure availability but generate forks that compromise consistency. In turn, Byzantine fault tolerance deterministic consensuses, generally used in permissioned networks, in general, compromise availability while ensuring consistency [b-Gre].

Hybrid solutions, such as networks permissioned to participate in consensus but not permissioned to receive and verify data, may be a good option for increasing the decentralization of a permissioned network without sacrificing the time required to reach consensus. One example of a hybrid network is Rinkeby[29].

It is worth further comment on risks 2 and 8. Regarding risk 2, permission-less platforms usually offer economic mechanisms to mitigate the risk of users manipulating network usage by sending too many transactions or a transaction that does not terminate. Permissioned network projects that do not charge for network use should have a mechanism to prevent such indiscriminate use, especially if the network is available to any Internet user. To mitigate risk 8, several actions are possible. First, it is necessary to monitor the load balancing of network nodes and to engage the network governance. One action more focused on the project might be to alter the dApp layer to balance the use of non-congested nodes. In addition, one way to mitigate the risk of misreading a node, without compromising decentralization, is to use dApps layers that obtain data from more than one node, possibly simultaneously, to check for possible inconsistencies.

8.2 Mitigation of smart contract risks

There are several alternatives for mitigating smart contract layer risks. Risks 9 and 10 are closely related to the software engineering process. Porru et al. [b-Por] propose the introduction of new symbols into UML to allow the particularities of DLT systems to be properly represented and the definition of architectural design patterns, such as hybrid on-chain and off-chain systems. Design patterns, good development practices and automatic code generation standards based on tested models are also discussed in the literature [b-Ban] [b-Sat]. Some design patterns address the need to evolve smart contracts by, for example, defining a proxy that points to an actual implementation of the smart contract. Application governance may be used to decide when to migrate from an outdated smart contract to an updated one. In addition, the use of code analysis tools [b-Dik] and formal methods [b-Sat] also help to improve the quality of the software produced.

Security infrastructure of traditional systems makes use of external devices such as firewalls, proxies, etc., but these devices have limited application in DLT projects, especially regarding permission-less platforms. This characteristic makes the software even more vulnerable to external attacks.

The risk of non-execution due to parameter variability can be mitigated by the project developer, who needs to consider limiting protocol variables during software building. The risk of using external data can be minimized by using IoT devices for real-world data collection, by using decentralized oracles and by receiving messages signed by trusted sources.

The risk of exposing source code bears analogy to choosing a permissioned or permissionless DLT network. Very simply, in a permissionless network anyone can become a network node and thus access on-chain records and the smart contract code (even if reverse-engineered from the compiled version), while in a permissioned network, only previously authorized nodes can do it by default. Mitigation, in this case, depends on whether project leaders prefer to make the code public or not, but possibly involves substituting the DLT network. The use of privacy techniques, like zk-snarks [b-Gro], or even the minimization of information that should go on-chain, may mitigate the risk of exposing information.

29 https://www.rinkeby.io/ (visited on 2019-11-27)

8.3 Mitigation of dApp, account and user risks

Mitigation of key management risks and user errors involves education and adoption of procedures. The likelihood of these errors tends to decrease as users utilize multiple DLT solutions, and as the knowledge required for management and use becomes more disseminated, as with other technologies.

One way for the project to minimize exposure to key management risks is not to offer custody services, indicating that users should be responsible for their keys, possibly contracting one or more third parties capable of providing such a service. Project leaders should consider whether their keys can be stored offline or in dedicated hardware to minimize network attacks. Another option for mitigating this risk is to ask users to associate a digital certificate with their DLT accounts, so that the digital certificate can be used in case of DLT account loss or theft to generate a new account and transfer linked resources, as proposed by [b-Ara]. To minimize the risk of misuse, some technical solutions can be QR code generators and contact lists.

Risks 16 and 17 are common to other web applications that have mitigation techniques already widely discussed in the literature.

The risk of the user being unable to submit a transaction to the DLT network can be mitigated if the user could be a network node or know how to submit transactions using different node options. Once again, education is an important factor for risk mitigation.

The risk of version incompatibility can be minimized if the project software is distributed with everything necessary for its use. However, sometimes the user may prefer not to use some of the distributed software. For example, the user may have more confidence in using an independent account management software and not the one provided by the project. Therefore, project leaders must keep up with market changes to avoid errors in such cases.

8.4 Mitigation of other risks

Project leaders have low autonomy to minimize risks 20-23, from governance and ecosystem. However, knowing these risks helps to predict system limitations and avoid surprises. Possible mitigation measures are training professionals, documenting knowledge about a platform, hiring specialized consultancies and adopting platforms with less risk to not evolve, e.g., with continuous flow of resources or that are maintained by solid companies.

Risks 22 and 23 are not only related to the chosen platform since they are influenced by the network and the governance of the application. Project leaders can try to influence the governance structure or contribute to the platform's or network's own sustainability, but such actions will generally have a small effect, especially if the community is large. Using a permissioned network with few nodes and known institutions could bring greater governance control and minimize forks, since it will be easier to perform off-chain agreements. The degree of the decentralization of the application governance must be chosen taking into consideration the importance of its decentralization to the business model and its necessary pace of decisions. To avoid forks, project leaders should also be concerned with keeping nodes up to date with network changes (such as patches and upgrades). To minimize risk 24, project leaders may need to change the project itself, or to break its deliverable in small parts with the goal to cope with society reaction.

The risk of lacking legal adequacy and the difficulty of determining jurisdiction can and should be mitigated by taking measures discussed with DLT experts serving in concerned public agencies, court members and representatives, or a legal professional. Legal uncertainty should be discussed in depth from a jurisprudential and doctrine standpoint. Judicial and jurisprudential certainty can be assessed within a probabilistic view, expressed quantitatively as a percentage of the possibility of the risk materializing, after relevant law and economic analysis. Risks relating to laws can be assumed and proven viable after such cost-benefit analysis. To mitigate the risk that a developed solution will be used for unplanned purposes, it is important to compose a terms of use document and request its acceptance by users of the software.

Some possibilities for mitigating economic risks are: (a) adopting a platform where these risks are more controllable, such as the use of permissioned networks, especially when block producers are not motivated by economic incentives associated directly with network use; (b) not using free market-priced crypto-assets; and (c) taking actions to expose project leaders to opposite economic risks, such as the purchase of certain financial instruments (e.g., derivatives).

Bibliography

[b-ITU-T X.1400] Recommendation ITU-T X.1400 (2020), *Terms and definitions for distributed ledger technology*.

[b-Ant] Antonopoulos, A.M., Wood, G. (2018), *Mastering Ethereum: building smart contracts and dApps. O'Reilly Media.*

[b-Ara] Arantes, G.M., D'Almeida, J.N., Onodera, M.T., Moreno, S.M.D.B.M., Almeida, V.D.R.S. (2018), *Improving the process of lending, monitoring and evaluating through blockchain technologies: An application of blockchain in the brazilian development bank (bndes).* 2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData). pp. 1181–1188. IEEE.

[b-Atz] Atzei, N., Bartoletti, M., Cimoli, T. (2017), *A survey of attacks on Ethereum smart contracts (sok).* International Conference on Principles of Security and Trust. pp. 164–186. Springer.

[b-Auer] Auer, R. (2019), *Beyond the doomsday economics of 'proof-of-work' in cryptocurrencies.*

[b-Bai] Baixue, Y., Kai, W., Hu, R. (2019), FG DLT D3.3 *Assessment criteria for DLT platforms*. <https://www.itu.int/en/ITU-T/focusgroups/dlt/>

[b-Bal] Baliga, A. (2017), *Understanding blockchain consensus models.* Persistent.

[b-Ban] Bano, S., Sonnino, A., Al-Bassam, M., Azouvi, S., McCorry, P., Meiklejohn, S., Danezis, G.: Sok (2019), *Consensus in the age of blockchains.* Proceedings of the 1st ACM Conference on Advances in Financial Technologies. pp. 183-198.

[b-Cas] Casino, F., Dasaklis, T.K., Patsakis, C. (2018), *A systematic literature review of blockchain-based applications: current status, classification and open issues*. Telematics and Informatics.

[b-Che] Chen, H., Pendleton, M., Njilla, L., Xu, S. (2019), *A survey on Ethereum systems security: Vulnerabilities, attacks and defenses.* arXiv preprint arXiv:1908.04507.

[b-Dik] Dika, A., Nowostawski (2018), *Security vulnerabilities in Ethereum smart contracts.* IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData). pp. 955–962.

[b-Gia] Giechaskiel, I., Cremers, C., Rasmussen, K.B. (2018), *When the crypto in cryptocurrencies breaks: Bitcoin security under broken primitives.* IEEE Security & Privacy 16(4), 46–56.

[b-Gre] Greve, F.G., Sampaio, L.S., Abijaude, J.A., Coutinho, A.C., Valcy, ´I.V., Queiroz, S.Q. (2018), *Blockchain e a revolu¸ca˜o do consenso sob demanda.* Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC)-Minicursos.

[b-Gro] Groth, J. (2010), *Short pairing-based non-interactive zero-knowledge arguments.* International Conference on the Theory and Application of Cryptology and Information Security. pp. 321–340. Springer.

[b-Hei] Heilman, E., Kendler, A., Zohar, A., Goldberg, S. (2015), *Eclipse attacks on bitcoin's peer-to-peer network.* 24th {USENIX} Security Symposium ({USENIX} Security 15). pp. 129–144.

[b-Ian] Iansiti, M., Lakhani, K.R. (2017), *The truth about blockchain.* Harvard Business Review 95(1), 118–127.

[b-Ibaa] Ibáñez, J. (2017), *Derecho de Blockchain.* Thomson Reuters Aranzadi. Cizur Menor.

[b-Ibab] Ibáñez, J. (2019), *Derecho de Blockchain, Aranzadi, Cizur Menor, 421-423; ÍD.* Blockchain: primeras cuestiones en el ordenamiento español, Dykinson, 19-24.; 141, 144.

[b-LiX] Li, X., Jiang, P., Chen, T., Luo, X., Wen, Q. (2017), *A survey on the security of blockchain systems.* Future Generation Computer Systems.

[b-Men] Mense, A., Flatscher, M. (2018), *Security vulnerabilities in Ethereum smart contracts.* ACM Proceedings of the 20th International Conference on Information Integration and Web-based Applications & Services. pp. 375–380.

[b-Mer] Al-Megren, S., Alsalamah, S., Altoaimy, L., Alsalamah, H., Soltanisehat, L., Almutairi, E., et al. (2018), *Blockchain use cases in digital sectors: A review of the literature.* 2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData). pp. 1417–1424. IEEE.

[b-Nak] Nakamoto, S. (2008), *Bitcoin: A peer-to-peer electronic cash system.*

[b-Poo] Poon, J., Dryja, T. (2016), *The bitcoin lightning network: Scalable off-chain instant payments.*

[b-Por] Porru, S., Pinna, A., Marchesi, M., Tonelli. (2017), *R.: Blockchain-oriented software engineering: challenges and new directions.* 2017 IEEE/ACM 39th International Conference on Software Engineering Companion (ICSE-C). pp. 169–171.

[b-Sat] Sato, N., Tateishi, T., Amano, S. (2018), *Formal requirement enforcement on smart contracts based on linear dynamic logic.* IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData). pp. 945–954.

[b-Tos] Tosh, D.K., Shetty, S., Liang, X., Kamhoua, C.A., Kwiat, K.A., Njilla, L. (2017), *Security implications of blockchain cloud with analysis of block withholding attack.* Proceedings of the 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing. pp. 458–467. IEEE Press.

[b-Zha] Zhao, J.L., Fan, S., Yan, J. (2016), *Overview of business innovations and research opportunities in blockchain and introduction to the special issue.*