Risk Type	Risk Category	Risk	Risk Description
Basic Blockchain Risks (BAS)	Key Management	BAS-1 Wallet credential theft	**Key Management: They deal with stealing of wallet or private key, or to the ability of an attacker to forge a private key or even a transaction, resulting respectively in an impersonation of a legitimate user or in the addition of a new transaction into the ledger;** normally these situations are connected to weaknesses/flaws in the cryptographic protocols or to the usage of weak keys
		BAS-2 Private key theft	
		BAS-3 Private key forging	
		BAS-4 Signature of rogue transaction	
	Cryptography	BAS-5 Weak key generation software	Key Management: They deal with stealing of wallet or private key, or to the ability of an attacker to forge a private key or even a transaction, resulting respectively in an impersonation of a legitimate user or in the addition of a new transaction into the ledger; **normally these situations are connected to weaknesses/flaws in the cryptographic protocols or to the usage of weak keys**
		BAS-6 Resilience of asymmetric keys to 0-days/quantum computing	
	Data Protection and Privacy	BAS-7 Data protection & privacy violation (header data)	BAS-7 deals with the need to avoid having personal data stored as public elements (even if in the headers) of the blockchain; under certain regulations, for example the General Data Protection Regulation (GDPR), special processing must be ensured for such pieces of information, thus the immutability and publicity of blockchain may stand in the way of compliance.
	Exploitable vulnerabilities in blockchain code	BAS-8 Lack of forward secrecy	BAS-8 addresses the possibility that a compromised private key may be used to compromise future transactions associated to that key; forward secrecy is a feature that explicitly prevents such situation.
		BAS-9 Security vulnerability in the blockchain code	BAS-9 deals with the risk of having exploitable vulnerabilities in the blockchain code.
	Consensus Management	BAS-10 Re-usage of a 'spent' asset	BAS-10 foresees the possibility that an asset (like a bitcoin) could be consumed more than once before a transaction is confirmed (i.e., verified) and thus immutable.
		BAS-11 Lack of new blockchain adoption after a hard fork	BAS-11 is relevant when after an attack, the only way to recover operations is through two measures, an "hard fork" in the chain of transaction to the last not-compromised transaction, and a software update to fix the exploited vulnerability. In this situation, recovery can only be achieved if all nodes accept and update their software timely to reduce service disruptions.
		BAS-12 Lack of means to slow down/stop consensus hijack attempts	BAS-12 refers to the possibility that consensus attacks may come to alter the standard blockchain behavior and no effective means to counter them are available.
	Wallet Management	BAS-13 Exploitation of wallet to access stored keys	BAS-13 and BAS-14 deals with the possibility to obtain blockchain keys exploiting wallet's vulnerabilities.
		BAS-14 Exploitation of wallet to access keys in transit	
		BAS-15 Loss of wallet	BAS-15 considers the possibility that an end-user loses her/his wallet, for example because it was hosted on a local device that got stolen or corrupted.
		BAS-16 Lack of detection of wallet duplication	BAS-16 on the other hand anticipates the possibility that a wallet could be stolen and no mechanisms for the detection of such situation are available.
	Scalability	BAS-17 Ledger size too big	Normally the blockchain ledger is replicated integrally in all nodes, and its size grows indefinitely. BAS-17 considers that storage capacity may not be adequately planned in advance to cope with the growing ledger size, thus causing disruptions like a denial of service.
		BAS-18 Transaction speed too slow	Transaction speed is the object of BAS-18 and it must be aligned with the performance requirements of the served use-cases.
		BAS-19 Reverting transaction block	**Sharding (BAS-19) occurs when a transaction block needs to be reverted due to an error during block validation**, while BAS-20 considers the case when validation nodes are not able to catch up with the validation requests.
		BAS-20 Validation bottleneck	Sharding (BAS-19) occurs when a transaction block needs to be reverted due to an error during block validation, **while BAS-20 considers the case when validation nodes are not able to catch up with the validation requests.**
	Regulatory, antifraud and anti-money laundering techniques and mechanisms	BAS-21 Unclear enforcement of legal constraints	**First and foremost, BAS-21 considers the compliance against the applicable legal framework; to counter critical attacks, protocol adaptations or even emergency responses can be needed,** so BAS-22 considers whether such adaptations may be implemented, and in a short time frame.
		BAS-22 Protocol evolution too slow	First and foremost, BAS-21 considers the compliance against the applicable legal framework; to counter critical attacks, protocol adaptations or even emergency responses can be needed, **so BAS-22 considers whether such adaptations may be implemented, and in a short time frame.**
		BAS-23 Untrusted end-user computer (hacked account)	BAS-23 instead considers cases where an end-user lost control of a device that is subsequently used to conduct attacks or frauds.
		BAS-24 Lack of means to verify the intent of executing a transaction	BAS-24 deals with non-repudiation, or better, to the possible absence of means to assert that a user intentionally performed a transaction, and it did not happen accidentally.
		BAS-25 Exploitation of the transaction protocol (hacked key)	On a similar line, BAS-25 looks at the possibility to exploit a session key or any other key used in the protocol, while BAS-26 considers vulnerabilities in the protocol flows.
		BAS-26 Exploitation of the transaction protocol (non-compliant transaction)	On a similar line, BAS-25 looks at the possibility to exploit a session key or any other key used in the protocol, **while BAS-26 considers vulnerabilities in the protocol flows.**
Crypto-currency risks	Crypto-Currency Risks	CC-1 Currency high volatility	Starts with CC-1, connected to the high currency volatility, which in turn may influence negatively on the consensus mechanism when getting too low or on the contrary, cause a surge of transactions if getting very high thus requiring effective scalability.
		CC-2 Usage of crypto-currency to transact malware (side channel)	CC-2 considers the possibility that the crypto-currency gets used to pay malware or any illegal goods, also completely offline.
		CC-3 Lack of compliance with anti-money laundering rules	**CC-3 deals with the possible lack of compliance against money-laundry rules coming from the financial domain,** while CC-4 considers vulnerabilities in the transaction protocol to allow double-spending of the same virtual currency.
		CC-4 Exploitation of the transaction protocol for double-spending	CC-3 deals with the possible lack of compliance against money-laundry rules coming from the financial domain, **while CC-4 considers vulnerabilities in the transaction protocol to allow double-spending of the same virtual currency.**
Hash data risks	Hash Data Risks	H-1 Hash collision	In a use-case where only document hashes are being stored as an audit trail of a document repository, H-1 considers the risk of an attacker forging a document happening to have the same hash value as a genuine one, opening-up the possibility of fraud.
Generic data risks	Capability of nodes to store arbitrary data onto the blockchain	GEN-1 Decryption of encrypted data	GEN-1 considers the risk that, as every blockchain node has access to blockchain data payloads, it has access to any encrypted data, so that one can be allowed to perform offline attacks without any kind of limitation or control.
		GEN-2 Lack of detection mechanism (repudiation)	GEN-2 considers that attacks on ledger (offline) copies will be impossible to detect, by definition.
		GEN-3 Resilience of encryption scheme (confidential data)	Another source of risk is considered in GEN-3, with respect to the resilience of the adopted encryption scheme: the advent of quantum computing, for example, may break our current cryptography in some years. Therefore, blockchain data confidentiality should not be considered as guaranteed forever.
		GEN-6 Storage of malicious data	GEN-6 states that, if arbitrary data can be stored in the blockchain, the mechanism could be abused for example to distribute viruses or illegal content, as it happened with Bitcoin.
	Personal Data Protection Regulations	GEN-4 Enforcement of the right to be forgotten	GEN-4 and GEN-5 looks at personal data protection regulations. The former considers the "right to be forgotten" as enunciated by GDPR, i.e., the need for an entity (data controller) to delete all personal data of an individual if she/he requests to do so. Naturally if personal data are stored in a blockchain (or used as identifiers), their deletion conflicts with its immutability property. The latter, the GDPR definition of personal data comprises also IP addresses and other elements that may be part of headers used in transaction or other logged messages. As their processing is strictly regulated (for example, an organization must also enforce the right to be forgotten also on those pieces of information), potential violations of data protection regulations must be assessed also for what concerns headers and any other element of the blockchain.
		GEN-5 Data protection & privacy violation (payload data)	GEN-4 and GEN-5 looks at personal data protection regulations. The latter, the GDPR definition of personal data comprises also IP addresses and other elements that may be part of headers used in transaction or other logged messages. As their processing is strictly regulated (for example, an organization must also enforce the right to be forgotten also on those pieces of information), potential violations of data protection regulations must be assessed also for what concerns headers and any other element of the blockchain.
Smart Contract risks	Smart Contract Risks	SC-1 Privacy breach through vulnerability in smart contract	SC-1 is about a possible re-identification of a person, or information disclosure on internal data processing, because of some data leaking from a Smart Contract implementation either because of a vulnerability or because of a design flaw.
		SC-2 Security vulnerability in the smart contract	**On a similar line, SC-2 consider the possibility that a Smart Contract implementation contains a vulnerability. Since deployed Smart Contracts cannot be revoked, it is a matter of reducing the attack surface to its minimum.** And, since the execution of the Smart Contract is mandatory, in the hypothetical case where the blockchain node can call external URLs, it could possibly define a specially crafted contract that connects to an external URL, letting it replicate through nodes and then repeatedly execute it to obtain a distributed denial of service attack (SC-3).
		SC-3 Smart contract-powered denial of service	On a similar line, SC-2 consider the possibility that a Smart Contract implementation contains a vulnerability. Since deployed Smart Contracts cannot be revoked, it is a matter of reducing the attack surface to its minimum. **And, since the execution of the Smart Contract is mandatory, in the hypothetical case where the blockchain node can call external URLs, it could possibly define a specially crafted contract that connects to an external URL, letting it replicate through nodes and then repeatedly execute it to obtain a distributed denial of service attack (SC-3).**
		SC-4 Lack of means to stop a smart contract from running	A problem is also represented by a Smart Contract property, that normally prevents them to be stopped after being deployed (SC-4).
		SC-5 Programming code allows side effects	SC-5 moreover considers situations where Smart Contracts can be expressed using imperative languages such as C: their execution may have side effects, e.g., a function can modify a variable which is not its return value, thus potentially altering some of the blockchain functionalities.
		SC-6 Design flaw in smart contract	Even worse consequences may happen in case of **design flaws (SC-6)** or zero-day vulnerabilities (SC-7). Therefore, considering the importance and the sensitivity of Smart Contracts, technologies like Hyperledger Fabric *) foresee Smart Contract deployments via an endorsement process, to reduce the risk of error. The Smart Contract developer defines two documents along with the Smart Contract: a policy, defining rules by which the Smart Contract abides, and an endorsement policy, containing what specific nodes called endorsers shall abide to. Then, a defined number of endorsers need to authorize the Smart Contract deployment. This shall happen after the endorsers tested the Smart Contract according to the defined policies. However, in this setting, the endorsement rules may be too weak to allow the detection of security flaws (intentional or accidental) (SC-8) (see Table 7).
		SC-7 Zero-day in smart contract code	Even worse consequences may happen in case of design flaws (SC-6) or **zero-day vulnerabilities (SC-7)**. Therefore, considering the importance and the sensitivity of Smart Contracts, technologies like Hyperledger Fabric *) foresee Smart Contract deployments via an endorsement process, to reduce the risk of error. The Smart Contract developer defines two documents along with the Smart Contract: a policy, defining rules by which the Smart Contract abides, and an endorsement policy, containing what specific nodes called endorsers shall abide to. Then, a defined number of endorsers need to authorize the Smart Contract deployment. This shall happen after the endorsers tested the Smart Contract according to the defined policies. However, in this setting, the endorsement rules may be too weak to allow the detection of security flaws (intentional or accidental) (SC-8) (see Table 7).
		SC-8 Weak endorsement rules for deployment	Even worse consequences may happen in case of design flaws (SC-6) or zero-day vulnerabilities (SC-7). Therefore, considering the importance and the sensitivity of Smart Contracts, technologies like Hyperledger Fabric *) foresee Smart Contract deployments via an endorsement process, to reduce the risk of error. The Smart Contract developer defines two documents along with the Smart Contract: a policy, defining rules by which the Smart Contract abides, and an endorsement policy, containing what specific nodes called endorsers shall abide to. Then, a defined number of endorsers need to authorize the Smart Contract deployment. This shall happen after the endorsers tested the Smart Contract according to the defined policies. **However, in this setting, the endorsement rules may be too weak to allow the detection of security flaws (intentional or accidental) (SC-8).**
Permissioned ledgers risks	Permissioned Ledgers Consensus Risks	PERM-1 Refusal to process a transaction	In such context, PERM-1 deals with the risk that transaction validations is not enough attractive for verifiers, if for example the resources to be committed in Proof-of-Work consensus mechanism are too expensive in comparison with the validation benefits.
		PERM-2 Lack of incentive to secure a sidechain	Again, on consensus mechanisms, in multichain ledgers, i.e., where it is allowed to have one or more "branches" of the main blockchain (often referred as sidechains), PERM-2 states the risk of not having enough verifiers for guaranteeing the immutability of the side-chains on top of that of the main blockchain.
		PERM-3 Consensus hijack by hack of regulators	Still on consensus, PERM-3 and PERM-4 concentrates on the validation side. As recalled, in general permissioned blockchains are used when the number of trusted verifiers is assumed to be low, which simplifies block validation and consensus mechanism; however, it can be possible that an attacker takes over one or more verifiers and at the same time, launches a denial-of-service attacks on all others in order to rewrite a part of the blockchain. PERM-3 captures this situation.
		PERM-4 Abuse of policy rules for chain takeover	For the same objectives, attackers may review blockchain policies to find vulnerabilities allowing them to take over on the blockchain (PERM-4).
	Other Permissioned Ledgers Risks	PERM-5 Targeted denial of service for guiding block validation	Denial of service for verifiers is the object of PERM-5, as also seen in PERM-3; in this case, it is contemplated the risk that block validation is delayed, to allow the creation of a fork with such a significant history to become the main chain once the attack terminates (exploiting the blockchain reconciliation procedures).
		PERM-6 Disclosure of internal processes	PERM-6 is about the disclosure of internal processes due to the blockchain transparency properties. Blockchain requests and transactions are signed by the initiating peers. In institutions, where several individual persons all run their own peer, the details of who is involved with each transaction will become visible on the ledger. This might be a confidentiality issue for the institution which may want to act as a unique body.
Permissionless ledgers risks	Other Permissionless Ledgers Risks	PLESS-1 User re-identification via transaction analysis	The permissionless ledgers specific risk list (in Table 3) starts with PLESS-1, that looks at the possibility to re-identify users' identities by analyzing their transactions; as blockchain does not provide complete anonymity with this respect, this risk should always be taken into account.
		PLESS-2 Block mining too simple	Naturally consensus management is critical in permissionless blockchain and on top of the risks in the basis section, PLESS-2 considers the case when block mining is too simple and easy to perform, for example due to the advent of dedicated hardware; in such situation, certain notes may gain unfair advantage and be able to take over the blockchain control and to rewrite a part of the history (the ledger records)
	Denial -of-Service Attacks	PLESS-3 Transaction spamming by rogue nodes	Risks associated to denial-of-service attacks are considered in PLESS-3 and PLESS-4. **The former looks at attacks performed by rogue nodes triggering transactions (or transaction attempts), if not prevented by protocol or platform measures,** while the latter considers the risk that verifiers do not get compensation for their mining activity thus, they do not have incentives to perform block validation. This situation may be by default or caused by circumstances, in any case it may be a serious risk for the sustainability of the solution under analysis.
		PLESS-4 No currency for mining	Risks associated to denial-of-service attacks are considered in PLESS-3 and PLESS-4. The former looks at attacks performed by rogue nodes triggering transactions (or transaction attempts), if not prevented by protocol or platform measures, **while the latter considers the risk that verifiers do not get compensation for their mining activity thus, they do not have incentives to perform block validation.** This situation may be by default or caused by circumstances, in any case it may be a serious risk for the sustainability of the solution under analysis.
	Smart Contracts in permissionless ledgers	PLESS-5 Front-running attack	PLESS-5 considers the usage of Smart Contract in permissionless ledgers: normally, transactions are broadcasted to all nodes, giving the basis for front-running attacks where, just before an important buy or a sell takes place, a rogue node submits respectively a buy or sell transaction, in order to exploit the fluctuations caused by the important operation. Such situation may happen for example if transaction processing order is regulated by a combination of rules (like 'biggest amount first', 'biggest transaction fee first') that results vulnerable to this attack.
	Financial Regulations Noncompliance	PLESS-6 Lack of means to identify an address owner	Financial regulations noncompliance is considered in PLESS-6 and PLESS-7, respectively for the lack of means to identify a user (in direct contrast with PLESS-1) or to block a transaction for illegal purposes.
		PLESS-7 Lack of means to block ongoing illegal transactions	Financial regulations noncompliance is considered in PLESS-6 and PLESS-7, respectively for the lack of means to identify a user (in direct contrast with PLESS-1) or to block a transaction for illegal purposes.
	Scalability Risk	PLESS-8 Power consumption too big	Proof-of-Work for validating new blocks costs an increasing amount of energy as there are more nodes validating blocks and as the reward diminishes as it is the case with Bitcoin. There is a risk that the power consumption becomes too big or too expensive, naturally dragging all verifiers to geographic areas where power is cheaper. This in turn concentrates validation nodes closer to each other, opening-up an increased risk of denial of service (in case of localized power outage) or of chain hijack (if a high number of nodes is operated from a few number of network infrastructures and that they are falling under control of an attacker, which may be a cybercriminal or an organization, scaling up to the hosting government).
Other (situational) risks	Non-Categorical Situational Risks	OTH-1 Security vulnerability in the platform code (node-hosting cloud platform)	Often blockchain applications rely on cloud platform services, in particular for node hosting and management. OTH-1 contemplates the possibility that a security vulnerability in the cloud platform may severely affect the network of blockchain nodes.
		OTH-2 Fraudulent spending via locking, side-chain transacting, then unlocking	**OTH-2 focuses on a technique to exploit sidechains (when available, as described for PERM-2) to consume the same asset more than once; this technique relies on the possibility (if available) to revert a sidechain.** This allows an attacker to consume an asset in a sidechain for a transaction, obtain the good or service acquired, and then having the sidechain reverted, so that the asset can be spent again in the main blockchain. Again, on sidechains, when one needs to be reverted, complex operations are needed in order to ensure the integrity of the chain(s), that may result in extra computational load on the nodes and thus paving the way towards a denial-of-service attack (OTH-3).
		OTH-3 Denial of service upon (big) sidechain revert	**OTH-2 focuses on a technique to exploit sidechains (when available, as described for PERM-2) to consume the same asset more than once; this technique relies on the possibility (if available) to revert a sidechain.** This allows an attacker to consume an asset in a sidechain for a transaction, obtain the good or service acquired, and then having the sidechain reverted, so that the asset can be spent again in the main blockchain. Again, on sidechains, when one needs to be reverted, complex operations are needed in order to ensure the integrity of the chain(s), that may result in extra computational load on the nodes and thus paving the way towards a denial-of-service attack (OTH-3).
		OTH-4 Exploitation of pruning mechanism	Again, on potential attacks, OTH-4 looks at pruning, a technique available in certain technologies which consists in storing only parts of the ledger. However, this feature, if not carefully conceived, might open-up the possibility of making fraudulent blocks look genuine, abusing hash collision.
		OTH-7 Transaction revert very hard to achieve	OTH-7 considers situations where a transaction must be reverted, in direct contrast with the immutability property of blockchain. Specific rules might be used if this risk must be considered.
	Interoperability	OTH-5 Lack of possibility to share between different ledgers	On a completely different perspective, OTH-5 anticipates a requirement that at this stage seems likely to be developed in the future by the main blockchain technologies, the interoperability between different ledger. Ad-hoc solutions to serve specific use cases are already possible today, for example with the adoption of sidechains to synchronize transactions on both ledgers.
		OTH-6 Lack of common wallet usage for different ledgers	Still on interoperability, but from another angle, OTH-6 considers the actual fragmentation in current wallet offers, as each blockchain requires a different wallet format.
	Wallet Technologies	OTH-8 Wallet address hard to visualize	Lastly, and again on wallet technologies, OTH-8 highlights the difficulties associated with wallet addresses. Such addresses are used for transactions and look like long character strings. It will be very difficult for a user to make the distinction between two addresses, opening the possibility to tamper with addresses

Sources

- Cédric Hebert and Francesco Di Cerbo, *Secure blockchain in the enterprise: A methodology*, SAP Security Research, France 25 June 2019.
- ENISA, *Distributed Ledger Technology & Cybersecurity, Improving Information Security in the Financial Sector*, Technical Report 978-92-9204-200-4, 10.2824/80997, ENISA, 2017

Footnote

*) C. Cachin, Architecture of the hyperledger blockchain fabric, *Workshop on distributed cryptocurrencies and consensus ledgers*, 310, 2016, http://www.zurich.ibm.com/dccl/papers/cachin_dccl.pdf, geraadpleegd op 10 mei 2021.

Explanatory note

Level 1 grouping (Risk Type) according to the sources
Level 2 grouping (Risk Category) inferred from the risk definition(sources)