# A Multilayer Approach to the Security of Blockchain Networks of the Future

Alexander Bogdanov[1,2], Alexander Degtyarev[1], Nadezhda Shchegoleva[1,2], Vladimir Korkhov[1], Valery Khvatov[3], Nodir Zaynalov[4], Jasur Kiyamov[1](✉), and Aleksandr Dik[1]

[1] Saint Petersburg State University, St. Petersburg, Russia
{a.v.bogdanov,a.degtyarev,n.shchegoleva,v.korkhov}@spbu.ru, {st080634,st087383}@student.spbu.ru
[2] St. Petersburg State Marine Technical University, Saint Petersburg, Russia
[3] DGT Technologies AG, Toronto, Canada
[4] Samarkand Branch Tashkent University of Information Technologies named after Muhammad al-Khwarizmi, Samarkand, Uzbekistan

**Abstract.** Decentralized computing and blockchain technology play a significant role in the implementation of modern digital economy business models. The most noticeable trends in this economy are the diversification and convergence of platforms and services, which is often achieved through undesirable fragmentation of the overall IT landscape. Business solutions represented by different blockchain networks turn out to be poorly connected, data exchange between them is difficult. The search for ways to overcome barriers between different decentralized networks leads to an increase in the importance of cross-platform integration solutions, providing the necessary level of interoperability. Such solutions must be secure both in terms of confidentiality and fault tolerance. Below is a vision of the architecture of integration gateways using the ODAP-2PC protocol, which provides crash fault-tolerance for the integration of various networks. This architecture provides transparency of interaction, reliability and continuity of audit in digital asset exchange systems or payment systems with increased requirements for interoperability.

**Keywords:** Blockchain · Distributed recovery · ODAP

## 1 Introduction

There is a growing interest in digital currencies and virtual assets as the foundation of the next generation digital economy. Blockchain technology has established itself as a reliable tool due to its properties such as immutability, transparency and controllability [1–3]. Private organizations, governments are actively researching and investing in blockchain-based digital assets, for example, by promoting new platforms for digital transactions [4]. The key task on the way to creating a digital economy is the secure connection of various networks, providing network effects between them [5–7]. Thus, the interaction of blockchains is a

O. Gervasi et al. (Eds.): ICCSA 2022 Workshops, LNCS 13377, pp. 205–216, 2022.
https://doi.org/10.1007/978-3-031-10536-4_14

key moment in this area [2,8–10]. Although significant progress has been made in the interoperability of public and private blockchains, legacy systems cannot yet seamlessly interoperate with each other [11]. Moreover, current solutions are not standardized and do not offer the possibility of seamless blockchain interaction across the enterprise, and the need for adaptability is a motivating factor for combining different blockchains to a heterogeneous ecosystem. The choice of new blockchains allows you to explore new scenarios and keep up with the times. However, each blockchain comes with its own security risks as the technology is still evolving. Therefore, developers have to choose between novelty and stability, which leads to a huge variety of options. This diversity leads to fragmentation: there are many immature solutions for blockchains (for example, without extensive testing). Until recently, blockchains did not take into account the need for interoperability, since each of them was focused on solving specific problems, which led to disparate stores of data and values.

## 2 Future Developments of Blockchain in 6G Network

Blockchain is one of the most famous technologies unlocking the potential of 6G systems. This section explores the possibilities and strengths of blockchain technology to address potential issues. As 5G connects users, services and applications, security and privacy are paramount. However, data management in 5G is more difficult than in earlier wireless networks, because the connected devices can be more diverse in type, they are expected to be very large in number, and the data they generate will be larger and more distributed (Table 1).

**Table 1.** Comparison of the new generation network.

| | 5G | 6G |
|---|---|---|
| Transmission speed | 0.1 Gb/s–20Gb/s | 1 Gb/s–1 Tb/s |
| Reliability (error rate) | ($\leq 10^{-5}$) | ($\leq 10^{-5}$) |
| Solidity | ($10^6$/km$^2$) | ($10^7$/km$^2$) |
| Localization accuracy | 10 sm in 2D | 10 sm in 3D |
| Mobility | 500 km/h | 1000 km/h |
| Throughput | (10 mb/s/m$^2$) | (10 gb/s/m$^3$) |
| Delay | 1–5 ms | 10–100 ns |

Transaction privacy leak: The blockchain relies in part on transparent transactions. Consequently, in blockchain-based systems, user privacy is at risk. Finally, blockchain-based smart contracts have significantly reduced the risk of de-anonymization, thanks to a closed transaction between contract participants [13].

However, as flexible as smart contracts are, they introduce a number of new attack surfaces into the system. The three main attack vectors for blockchain-based smart contracts are vulnerabilities in the blockchain itself, in the smart contract, and in the code-executing virtual machine [14].

The network layer transmits messages related to transactions and system management. Scaling requires each node to select only raw transactions for newly mined blocks; this will effectively halve the number of transactions required. In addition, the network plane topology can be modified to improve broadcast protocols [15].

The storage layer is a ledger - a global memory that stores member state changes resulting from write and read operations mutually agreed upon by all members at the consensus level, as well as smart contracts or other state-related entities. Storage tier performance improvement methods are divided into:

- Storage sharding;
- Distributed hash tables.

6G is expected to dramatically increase the performance and number of wireless network services, resulting in a significant increase in ubiquitous computing resources.

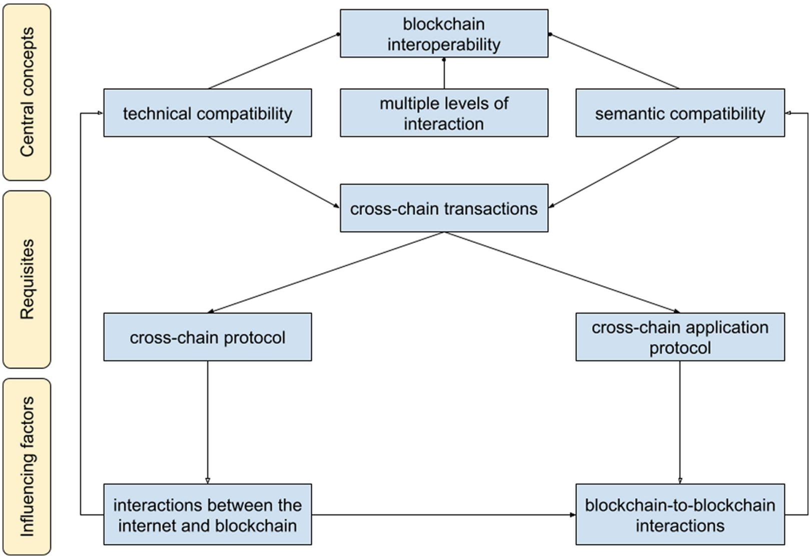
Central concepts
Requisites
Influencing factors
blockchain interoperability
technical compatibility
multiple levels of interaction
semantic compatibility
cross-chain transactions
cross-chain protocol
cross-chain application protocol
interactions between the internet and blockchain
blockchain-to-blockchain interactions

**Fig. 1.** Illustration of the relationship between different blockchain concepts.

The following types of blockchain network interaction can be distinguished [17]:

- Full state replication between blockchains;

- Blockchain cross-application support;
- Blockchain compatibility with other technologies.

Figure 1 shows the relationship between different blockchain interoperability concepts. This approach can provide interoperability at the semantic level (that is, related to the transfer of a data value that corresponds to the interaction) mapped to the application level. Based on the foregoing, let's take a closer look at multi-level protection, using the DGT platform as an example.

## 3 Layered Data Protection Approach

Most optimization algorithms require synchronization of local peer-to-peer access to WAN information. This is a separate problem, known as the aggregation problem [7], and refers to a set of functions that provide access to such components of a distributed system as network size, load average and uptime. Consider the solutions of the DGT platform, which considers the problem of fault tolerance with a multi-level data processing approach. This approach is based on the deployment of a virtual network for solving problems or when developing applications. Servers or nodes may also be located on different physical networks. Node clusters are part of a larger network division - segments, which can be of two types: public and private. In a separate network, only one public segment is possible, joining nodes can freely interact with other segment nodes. The network can have several private segments, the main difference of which from the public segment is the controlled topology (Fig. 2).

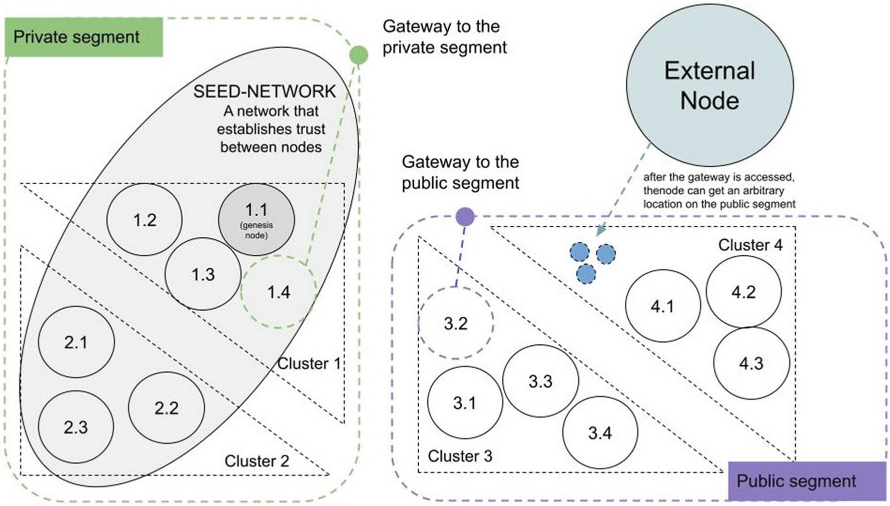
Private segment
Gateway to the private segment
SEED-NETWORK
A network that establishes trust between nodes
1.1
(genesis node)
1.2
1.3
1.4
Cluster 1
2.1
2.2
2.3
Cluster 2
External Node
Gateway to the public segment
after the gateway is accessed, thenode can get an arbitrary location on the public segment
Cluster 4
4.1
4.2
4.3
3.2
3.3
3.1
3.4
Cluster 3
Public segment

**Fig. 2.** DGT network topology and node attaching

The initial implementation of the network, also called the "static core", which is a group of nodes that form special trust relationships (the public keys of such nodes are known in advance and are registered at the time of the core deployment). Joining other nodes requires the processing of node certificates for private shards, or a dynamic connection for public shards.

DGT is positioned as a platform for distributed and decentralized computing, where the system processes data regardless of the specific application task. To solve a specific task, it is required to set up a family of transactions, as well as an add-on of the applied client part [19,20]. In fact, the DGT software is the set of typical nodes, which provide interaction with other nodes, data validation and insertion of new data into the storage (registry), also called DAG or State. It is aimed at supporting consortium-based networks. This means that a node can join the network if certain conditions are met. In the simplest terms, this could be checking the host for a certificate. Depending on the implementation of the anchor mechanism, the degree of openness of the network varies - from completely open (public) to completely closed (private). Nodes are combined into groups called clusters. The initial interaction is carried out through connections between nodes with one dedicated node in the cluster - Leader. The leader collects data from transaction checks at each node. Such checks are called "votes". If the number of votes exceeds a certain threshold, then the transaction is considered approved in the cluster and awaits arbitration - performed outside the cluster (additional verification). Within a cluster, nodes interact with each other via dedicated channels, also called permalinks.

Following Sawtooth, DGT is a multi-transactional system in which several families of transactions can be addressed. Each family is processed by a separate transaction processor. Transaction families complement the technology of smart contracts, and also allow you to set the boundaries of the availability of different types of transactions for different network segments. But this approach cannot provide complete system protection, as server components or the server itself may fail.

## 4 Failure Recovery

To ensure a fair exchange of assets, blockchain gateways must work reliably and be able to withstand various attacks. Thus, a disaster recovery strategy should be a major factor in the design of blockchain gateways, where specific recovery protocols can be developed as part of the digital asset transaction protocol between gateways. The recovery protocol associated with the failover strategy ensures that the source and target DLTs are changed sequentially, i.e. that assets taken from the source DLT are preserved in the destination DLT and no double spending can occur.

Gateways allow the seamless transfer of digital currencies and virtual assets between these systems. The Internet Engineering Task Force is currently working on an asset transfer protocol that works between two gateway devices. In a layered approach, when increasing throughput, communication failure may occur, to solve this problem, you can use the ODAP protocol.

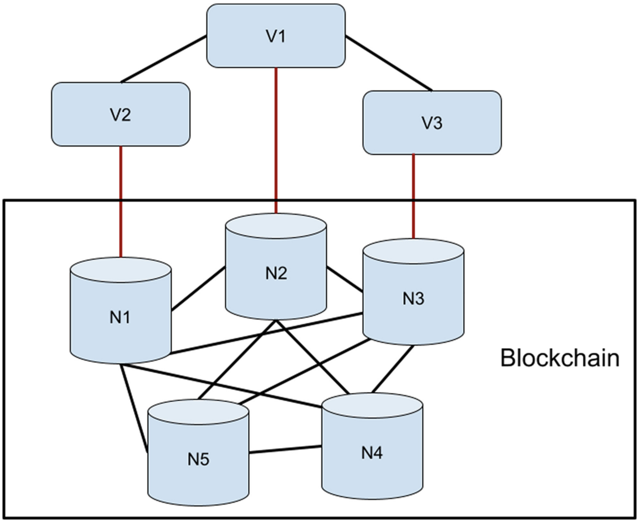

**Fig. 3.** The structure of a multi-level network. V1, V2 and V3 are the validator nodes, N1, N2, N3, N4 and N5 are the nodes of the blockchain network

## 5 Open Digital Asset Protocol (ODAP)

ODAP is an internetworking protocol that handles multiple cross-border digital asset transactions using asset profiles (asset schema) and the concept of gateways. The most common layered network architecture is the client-server architecture. It includes two ranks of communication participants, the rank of the client and the rank of the server, while the rank of the server is dominant in the network. This model is the basis for the centralized exchange and storage of information, and the most common network architecture in the modern Internet. The structure of this network is shown in Fig. 3. It can be seen from the figure that "Node 1" is a server and all network clients access it with requests. It can also be seen that the other nodes do not communicate directly with each other, they do not imply such a possibility. All inter-client interactions occur either through the mediation of the server, or do not occur at all. In this case, when the server fails, in fact, the entire network fails and the client nodes absolutely lose the opportunity to receive the service provided by the server node.

Crash fault-tolerant (CFT) systems can fail on $n/2$ nodes, where $n$ is the number of nodes. As long as there are most nodes with the latest state, failures are tolerable [18]. The primary backup model defines a set of $n$ hosts (or nodes) that, as a group, provide fault tolerance for the service, thereby increasing availability. In this model, the application client sends messages to the primary node $P$. The primary nodes forward message updates to the backup set $B = B_1, ..., B_n$ when it receives a message. Backup server $k$ propagates a new incoming message to backup server $k + 1, k \leq n, k \in R$. Node $P$ is notified of the update when $n$-node failover is reached. The message has been replicated to at least $n$ nodes.

If such an acknowledgment cannot be received $P$, the message update request is resubmitted. If $P$ fails, then a new leader $P_{new} \in B$ is chosen. If the standby node receives a request from an application client, it forwards it to $P$, accepting it only when the latter sends an update request. When an update is received, $P$ sends a message update to its right neighbor, sending back an acknowledgment.

Another recovery mechanism is self-healing [16]. It is assumed that during self-healing, when nodes fail, they sooner or later recover. Although this mode is cheaper than primary backup because it uses fewer nodes, less messages exchanged, and less storage requirements.

## 6 Distributed Recovery Procedure

One of the key requirements for deploying asset transfer gateways is the high availability of the gateways. The distributed recovery procedure then improves the fault tolerance of the layered nodes through fault tolerance. Next, we present an overview of the ODAP-2PC. ODAP-2PC supports two alternative fault tolerance strategies:

– Self-healing mode: after a failure, the gateway eventually informs other parties about its recovery and continues the protocol execution;
– Primary backup mode: if a node goes down indefinitely, a backup is created using the log storage API to retrieve the most recent version of the log.

We assume that the gateway does not lose its long-term keys (public/private key pair) and can re-establish all TLS connections. In the main-backup mode, we assume that after a period $\delta_t$ of failure of the main gateway, the backup gateway unambiguously detects this failure and assumes the role of the main one. Failure is detected using a conservative value of $\delta_t$. To do this, a backup gateway essentially does the same thing as a gateway in self-healing mode:

– reads the log and continues the process. In this mode, the log must be shared between the primary and backup gateways. If there is more than one backup, a leader election protocol must be run to decide which backup will take the lead role.
– In both modes, logs are written before operations to ensure the atomicity and consistency of the protocol used to exchange assets. The log data is considered as a resource that may be internal to the DLT system.

There are several situations where a failure can occur [18]. On Fig. 4 shows the GS (source gatway) failing before it performs the verification operation for the GR (recipient gateway) steps 1 and 2. Both gateways retain their storage APIs. In self-healing mode, the gateway eventually recovers (step 3) by creating a recovered message in the form (step 4). The unbroken gateway requests the log entries that the failover gateway needs (steps 5, 6). In particular, the GS obtains the required log entries in step 7 and compares them with its current log. The GS then attempts to reconcile the changes with its current state (step 8). After processing, if both versions of the log match, the log is updated and

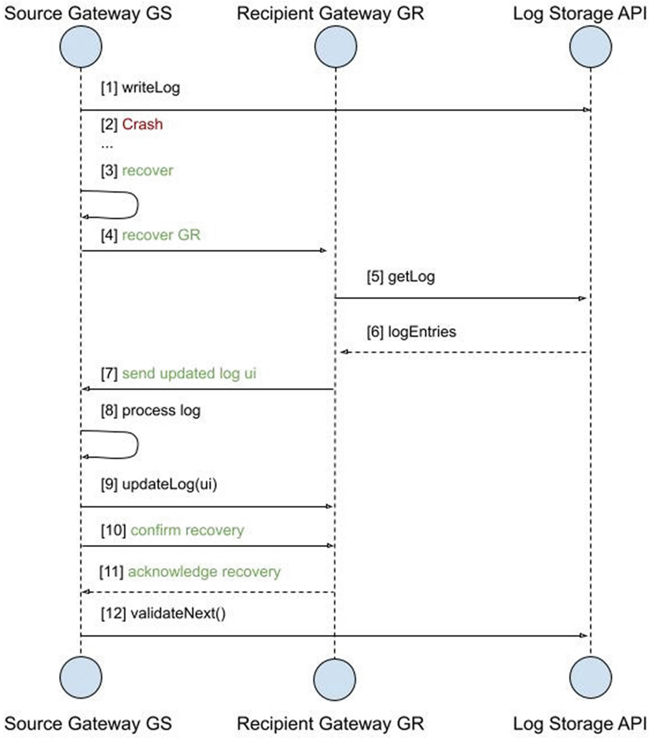

**Fig. 4.** GS failure before issuing initialization-validation GR

the process can continue. If the logs differ, then the GS calls the updateLog primitive, updating its log (step 9) and thereby allowing the failed gateway to recover the current moment.

In this particular example, step 9 will not occur because the exec-validate, done-validate, and ack-validate operations were not performed by GR. If the Log Storage API is in Shared mode, no additional synchronization steps are required. After that confirms successful recovery (steps 10, 11).

A set of experiments was carried out with a two-rank processing system as shown with fault tolerance on the DGT platform. Initially tested on a stable with 24 presti nodes over 1000 transactions and it resulted in an average throughput of (Fig. 5) 0.009 s per transaction.

In Fig. 6 simulated forced failure of 6 out of 24 nodes while processing 1000 transactions, in this scenario, the node leader started to process voting rounds among 18 nodes and continued to process data. This processing position took an average of 0.0077 s (Fig. 6) of time.

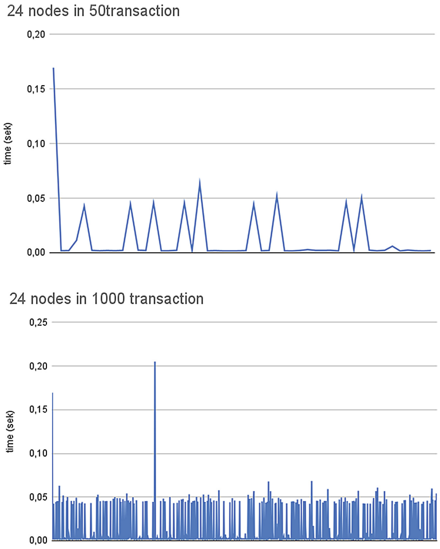

**Fig. 5.** Transaction processing graph with 24 nodes deployed.

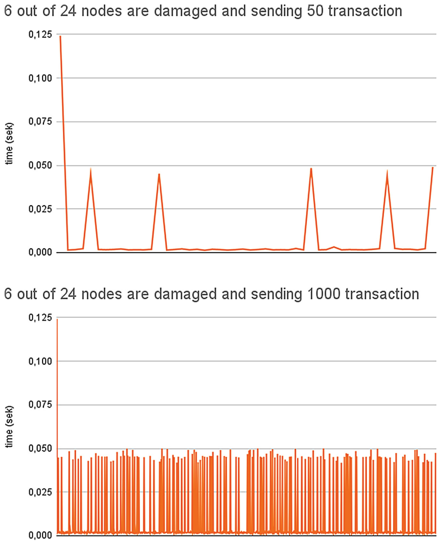

**Fig. 6.** Graph of transaction processing with 24 deployed nodes which forced a system failure on 6 nodes.

In the third scenario, it was decided to run on 12 nodes to simulate fault tolerance at 24 nodes. Despite such a failure in the number of nodes, the PBFT consensus round of voting was stable, processing one transaction in 0.0073 s on average (Fig. 7).

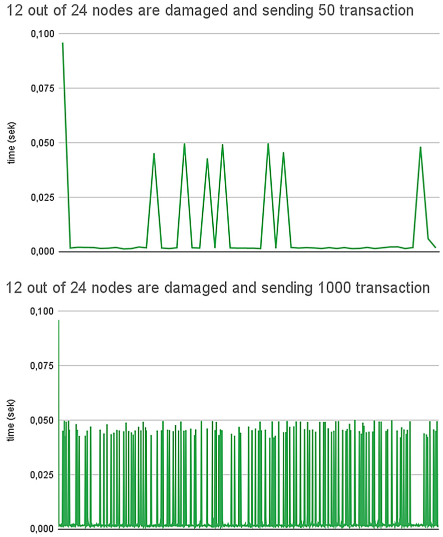

**Fig. 7.** Graph of transaction processing with 24 deployed nodes which forced a system failure on 12 nodes.

Each node is a set of services that interact with each other and are responsible for organizing the network, storing data, and processing transactions. Even a single node delivers a significant service that supports client applications via APIs. At the same time, a number of network capabilities of the platform can be used only if there are several nodes. Therefore, thanks to this technology, we eliminate the main drawback of a distributed system - data failure, which allows us to reduce the risks associated with the failure of a part of the system.

# 7 Conclusion

In this article, we have presented a possible view of how to overcome barriers between different decentralized networks leading to an increase in the importance of cross-platform integration solutions, providing the necessary level of interoperability. Such solutions must be secure both in terms of confidentiality and fault tolerance. Integration gateways using the ODAP protocol, which provides crash fault-tolerance integration of various networks. We have shown that our solution is fault tolerant by using a multi-rank distributed recovery mechanism approach.

The research is partially funded by the Ministry of Science and Higher Education of the Russian Federation as part of World-class Research Center program: Advanced Digital Technologies (contract No. 075-15-2020-903 dated 16.11.2020).

## References

1. Catalini, C., Gans, J.S.: Some simple economics of the blockchain. Working Paper 22952, National Bureau of Economic Research, December 2016. http://www.nber.org/papers/w22952. https://doi.org/10.3386/w22952
2. Hargreaves, M., Hardjono, T., Belchior, R.: Open Digital Asset Protocol draft 02, Internet-Draft draft-hargreaves-odap-02, Internet Engineering Task Force (2021). https://datatracker.ietf.org/doc/html/draft-hargreaves-odap-02
3. Viriyasitavat, W., Da Xu, L., Bi, Z., Pungpapong, V.: Blockchain and Internet of Things for modern business process in digital economy-the state of the art. IEEE Trans. Comput. Soc. Syst. **6**(6), 1420–1432 (2019)
4. Pentland, A., Lipton, A., Hardjono, T.: Time for a new, digital Bretton Woods, Barron's, June 2021. https://rb.gy/yj31vq
5. Pawczuk, L., Gogh, M., Hewett, N.: Inclusive deployment of blockchain for supply chains: a framework for blockchain interoperability. Technical report, World Economic Forum (2020). https://www.weforum.org
6. Hardjono, T., Lipton, A., Pentland, A.: Towards an interoperability architecture blockchain autonomous systems. IEEE Trans. Eng. Manag. **67**(4), 1298–1309 (2019). https://doi.org/10.1109/TEM.2019.2920154
7. Tam Vo, H., Wang, Z., Karunamoorthy, D., Wagner, J., Abebe, E., Mohania, M.: Internet of blockchains: techniques and challenges ahead. In: 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 (Smart-Data), pp. 1574–1581 (2018)
8. Pillai, B., Biswas, K.: Blockchain Interoperable Digital Objects Innovative Applications of Blockchain Technology View project Blockchain Interoperable Asset Classes View project (2019)
9. Borkowski, M., Sigwart, M., Frauenthaler, P., Hukkinen, T., Schulte, S.: DeXTT: deterministic cross-blockchain token transfers. IEEE Access **7**, 111030–111042 (2019). arXiv
10. Schulte, S., Sigwart, M., Frauenthaler, P., Borkowski, M.: Towards blockchain interoperability. In: Di Ciccio, C., et al. (eds.) BPM 2019. LNBIP, vol. 361, pp. 3–10. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-30429-4_1
11. Belchior, R., Vasconcelos, A., Guerreiro, S., Correia, M.: A survey on blockchain interoperability: past, present, and future trends. ACM Comput. Surv. (2021). arXiv:2005.14282. http://arxiv.org/abs/2005.14282
12. Li, X., Jiang, P., Chen, T., Luo, X., Wen, Q.: A survey on the security of blockchain systems. Future Gener. Comput. Syst. (2017)
13. Wood, G., et al.: Ethereum: a secure decentralised generalised transaction ledger. Ethereum Project Yellow Paper **151**(2014), 1–32 (2014)
14. Atzei, N., Bartoletti, M., Cimoli, T.: A survey of attacks on Ethereum smart contracts (SoK). In: Maffei, M., Ryan, M. (eds.) POST 2017. LNCS, vol. 10204, pp. 164–186. Springer, Heidelberg (2017). https://doi.org/10.1007/978-3-662-54455-6_8
15. Croman, K., et al.: On scaling decentralized blockchains. In: Clark, J., Meiklejohn, S., Ryan, P.Y.A., Wallach, D., Brenner, M., Rohloff, K. (eds.) FC 2016. LNCS, vol. 9604, pp. 106–125. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-53357-4_8
16. Bernstein, P.A., Hadzilacos, V., Goodman, N.: Concurrency Control and Recovery in Database Systems. Addison-Wesley, Boston (1987)

17. Belchior, R., et al.: A survey on blockchain interoperability: past, present, and future trends. ACM Comput. Surv. **54**(8), 168 (2021)
18. Belchior, R., Vasconcelos, A., Correia, M., Hardjono, T.: HERMES: fault-tolerant middleware for blockchain interoperability. Future Gener. Comput. Syst. (2021)
19. DGT One Pager, official and short DGT Platform Description
20. DGT. The Blockchain Handbook