Decision Support Systems' Security Model Based on Decentralized Data Platforms

Mykyta Savchenko[1], Vitaliy Tsyganok[1][0000-0002-0821-4877] and

Oleh Andriichuk[1][0000-0003-2569-2026]

[1] Institute for Information Recording of National Academy of Sciences of Ukraine,
Kiev, Ukraine
zitros.lab@gmail.com, tsyganok@ipri.kiev.ua,
andriichuk@ipri.kiev.ua

Abstract. Modern information systems, especially high-risk systems which work with important data lack proper security models. Decision support systems and recommendations they produce are extremely dependent on the data they produce recommendations on, hence they require the most secure data platform and transparent history of data input and changes. Many of such systems are centralized and are stored in a single place, like a data center or even a single machine, where important data can be lost easily. Most importantly, it can be tampered without many complications, as for the typical setup there are people who always have access to the system and its data, including people who know for sure how the system is made and who can act unnoticed, being bad actors. Through all of this can be solved by introducing proper monitoring mechanics and implementing best security practices like using secure protocols and encryption, this article describes methods which completely exclude data tampering possibility and add more important properties like data immutability, decentralization and fault tolerance on a platform level.

Keywords: security, decentralized platforms, distributed systems, decision support system, knowledge base, data immutability.

1 The Problem of Data Security of Decision Support Systems

Decision support systems (DSS) are systems that take facts and produce recommendations for decision-makers based on numerous factors. This recommendations is typically used for informational purposes only, however, the system's output might be considered as the primary conclusion for many complex operations that people usually cannot handle on their own (examples: social groups behavior research, a complex-network of facts and relationships, others) [1, 2]. The ongoing increasing of subject domains' models complexity requires an adequate and detailed representation of sets of factors and their interactions in the DSS knowledge base. Significant level of detail of knowledge base leads to redundancy, ambiguity, contradictions' presence in the base and, thus, to the deterioration of the adequacy of subject domains' models.

A structure of DSS knowledge base is shown in Fig. 1. The main elements of knowledge base are objects and connections between them. Knowledge base objects can be one of two types: target or project. Each knowledge base object has a name in short wording form. Semantic meaning of the object is specified by a keyword tuple with corresponding weights. This object can be quantitative or qualitative, threshold or quasi-linear. Projects have specific parameters: runtime and resources required. A connection between knowledge base objects can be positive or negative, it can have a time delay, compatibility groups. It is also characterized by a relative impact factor.

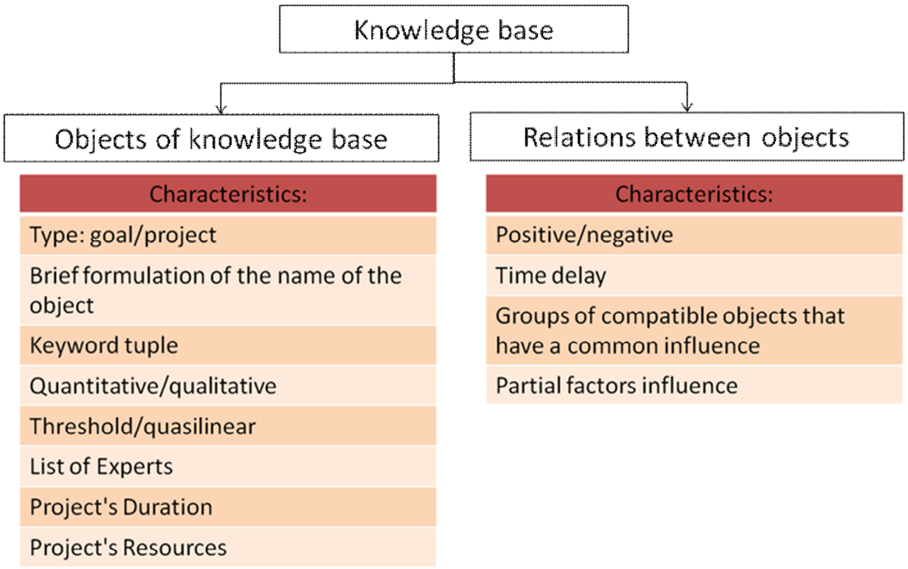

Fig. 1. Structure of DSS knowledge base

Because the recommendations of these systems depends on a data so much, a proper security model standard must be applied in order to prevent unauthorized or unexpected changes in the data, which may change the decision made by decision support system. There are several concepts related to security that should be reviewed.

1.1 The Need to Trust Entire System and Rely on People

In order to trust the recommendations of decision support system, one should trust the developers of this system, algorithms behind the system, people that control and maintain the system and those who interact with it and input information [2]. Furthermore, produced recommendations can be incomprehensible or undesirable, which raise more questions rather than answers regarding system operation [3]. As the result, there are a plenty of cases in which each party have to trust the system in order to accept produced decision.

Having the decision support system that is fully under the control of one party, or even multiple parties enable them to make decisions regarding the data they own:

1. Which data to input and how to organize an input process.
2. Which data to filter from the input and which to keep.
3. How to organize a decision support process.
4. How to tune the system for a specific case.
5. How to properly present and explain the recommendations.

All of these processes rely on people who work with the system, not to mention those who input data and the system itself. In such a scheme there are too many points of failure, the main of which are people themselves. In case of undesirable results, a group of people can decide to slightly change the input data achieving the result they desire on their own. Having a system in which much of the trust relies on people is not desirable, as people never can be a source of genuine and logical decisions by their nature [1].

Instead, it is much desirable to have a system which does not require the trust to its operators and is trusted on a global level. This eliminates any human errors, as well as intentional attempts to change something in the system, leaving this change unnoticed.

1.2 Data Tampering Risks

Decision support systems are extremely sensitive to any data they store and, most importantly, take as an input, as even a slight modification in this data could drastically change the result. People who input data to the system can do errors, both accidentally or intentionally, and there is always a chance for them to play a bad role and intentionally harm the output of the system by giving an invalid or ambiguous input. Different methods can be used to reduce the input error, however, they cannot eliminate it [3].

Moreover, the input data can always be reviewed and validated by other people. Hence the only thing bad actors can perform in order to change resulting decisions is to tamper with data and hide any evidence of it.

Traditional system security models feature many ways to protect data authenticity, including encryption, data mirroring, backups, monitoring, and logging, but even if the system is set up properly and is well-maintained, there is always a risk that something can go wrong or is missing, and more importantly there is no way to ensure that the data wasn’t changed from the moment of its entry [4].

Decentralized data platforms solve this problem by introducing immutable public or permissioned ledger maintained by multiple parties that participate in a process of validating this ledger. Hence, a single entry must be trusted by all or the majority of network participants in order to be recorded on a ledger. Data is also recorded in a cryptographically secure way, excluding the possibility of previous records modification [5].

1.3 Reliability of Data Storage and Knowledge

In traditional storage systems, there is a central database which stores all the records. To prevent possible data losses in case of emergency, these systems can be set to do regular backups. However, a special care needs to be taken in order to ensure that no data was lost, damaged or tampered. Usually, the systems which guarantee all these properties are very expensive to afford and maintain.

In comparison, decentralized data platforms offer storage redundancy, but eliminate all security and maintenance risks. Additionally, the decentralized system can be configured in order to consume less storage if required [5].

1.4 Ownership of Information

While in traditional data platforms data is typically owned by one party, decentralized platforms are highly focused on the problem of information ownership. Unlike traditional data platforms, they are designed to not to concentrate all stored information, as well as computing power on a single party or even in hands of some group.

But this doesn't mean that the information is not accessible nor is accessible to all parties. If we take blockchain as an example, every network participant in a decentralized system can obtain a copy of all information available, but the information itself can be both open or encrypted. However, in a properly set up decentralized system, one can be sure that the information recorded to the decentralized storage is permanent, and all data that belong to one party won't change its hands to another, which is guaranteed by data immutability property of such systems [6].

2 Decentralized Security Approaches

In order to protect a typical centralized computer system from all possible intrusions and attacks, which also applies to decision support systems, a lot of work must be performed. This, in turn, does not exclude the possibility of a hack and cannot guarantee that the system is fully protected. In other words, no matter how the system is protected, there is always a *practical* way to attack it.

Decentralized platforms, with blockchain technology as a primary example, offer a solution to above problems: they eliminate risks of tampering with the system and provide a data platform with a strictly determined way of how different parties can interact with the platform and rules of any possible data manipulations within the platform. Also, they offer a highly failure-resistant system, which means that even if half of all network nodes will crash, the system will still continue to operate normally [5].

Decentralized data platforms like blockchain guarantee by design that no single party can interrupt the normal, predictable functioning of the platform, as well as no data can be changed in an undefined way. In other words, there is no practical way to make a decentralized program produce a different result from expected, as well as tamper with any historical data [6].

2.1 Existing Decentralized Data Platforms

The first successful application of a public decentralized platform was released in 2008 by Satoshi Nakamoto, the name used by the unknown person or a group of people who originally developed Bitcoin – a peer-to-peer electronic cash system, which is still under a high demand today [9]. At that time it was barely possible to call Bitcoin a "decentralized platform", as the only functionality it served is virtual value transfers between parties. The term "decentralized platform" became well-recognized almost 7 years later, in 2015, with the release of Ethereum – world's first practically successful decentralized applications platform, which allowed people not only to transfer value over the internet but also to run any programs and computations which cannot be tampered with [10].

However, there are other emerging technologies that try to address many blockchain issues. We will briefly describe them below.

Blockchain. Bitcoin and Ethereum, as the most widely adopted decentralized platforms, are running using blockchain technology, which is one that is practically proven to be secure and stable. Blockchain's main disadvantage is its scalability issues, which other decentralized data platforms try to address. However, blockchain sharding, as the most recently introduced way to scale traditional blockchain platforms, offers a way to increase blockchain throughput without compromising security [8].

The name "Blockchain" appeared directly from an underlying algorithm meaning – the chain of blocks. Transactions in a blockchain are organized in blocks, and blocks reference each other. This structure with cryptographically secured references between blocks is called blockchain [8] (Fig. 2).

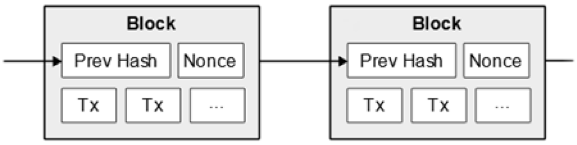

Fig. 2. Blockchain data structure

In blockchain, each next block must reference its previous block in order for the whole structure to be valid. This reference isn't just a link – it's a hash of a previous block. If a previous block's contents changes, its hash changes, and thus the reference to modified block becomes invalid. Because the whole network in blockchain validates blocks, every invalid block is just ignored, leaving no chances to someone to tamper with data and being unnoticed.

The block size in blockchain is intentionally limited, which leads to scalability issues [11]. If the block size wasn't limited, then fewer network participants could participate in forming a consensus because of hardware and network throughput limitations.

Hashgraph. Unlike blockchain, hashgraph does not pack transactions into blocks. Instead, transactions have cryptographic references to each other, forming a data structure which is also called hashgraph [12]. For example, in IOTA project, which is an early adopter of this approach, the global ledger forms a data structure called Tangle, which is a direct acyclic graph in which each next transaction reference two previous [13]. The example of the hashgraph structure is depicted in figure 3.

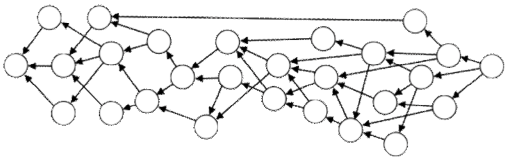

Fig. 3. Hashgraph (direct acyclic graph) structure

Hashgraph uses a probabilistic-based consensus algorithm, as an opposite to blockchain's deterministic consensus algorithms. For instance, to ensure that the particular transaction happened in hashgraph, a network participant uses a gossip protocol; it asks some of its peers about whether or not they have the same transaction recorded, and the transaction is considered valid when, for example, 2/3 of the peers respond positively.

Unlike blockchain, hashgraph doesn't have a limit of transactions. Its tangle can theoretically become very big and still be able to handle all incoming transactions. But, on the other side, hashgraph is less resistant to attacks. By having enough power, the continuous attack on a hashgraph ledger can create a "parasitic" hashgraph, which can prevent new transactions from happening as they might be considered invalid by more than 1/3 of the network, for example.

There are many examples where Ethereum and other similar technologies were adopted by businesses, government and healthcare, as well as many other fields of study. However, these platforms didn't get massively adopted yet, as they all suffer from scalability or security issues. There are many different higher-level solutions were introduced like sharding, lightning network, plasma, and others, but still, they are all vulnerable in case of weak algorithms used.

2.2 Existing Consensus Algorithms

A consensus algorithm in a decentralized network is a process used to achieve agreement on a single data value or a network state among distributed network participants. In Blockchain, a consensus algorithm is used to agree on the latest state of a ledger, which is supported by all network participants [14].

There are many consensus algorithms in existence, but there are just a few which are practically proven to work and are widely used as of 2018: Proof of Work, Proof of Stake and Delegated Proof of Stake. We will briefly introduce the main idea behind these algorithms and compare them to each other.

Proof of Work. In a Proof of Work consensus algorithm, the blockchain network agrees on a rule to accept the valid chain with the highest total complexity [15]. Everyone in a network can try to increase the total chain complexity, by adding new blocks with transactions. Network participants are economically incentivized to add transactions of other network participants to blocks they produce, as after forging a valid block they collect all transaction fees plus a block finding reward.

Adding each new block to a chain requires finding such a hash of a block that maps to a number which is below the given threshold called complexity. Finding this hash is not a trivial task; this task doesn't have a known solution due to the pre-image resistance property of a hash function used. Thus, the only way to find a valid hash of the block is to try all possible values of a random number nonce which is used in hashing.

The first network participant called "miner" who generates a valid hash from a block broadcasts it to a network, while all other network participants accept this block because its hash is below the given complexity for a block. The next block's complexity is deterministically computed from the complexity of previous blocks and time required for its mining so that no matter how powerful computers in the network are, there is always a stable approximate time between new blocks added to the network (15 seconds for Ethereum).

Because finding a right hash for a block is time and computational resources consuming task, network attracts more and more incentivized parties that try to find a right hash, making network complexity enormously big, which, in turn, makes any possible attacks to this network impractical.

Proof of Stake. Proof of Work algorithm requires a lot of energy for mining [14, 16], and this creates two problems:

- A lot of energy is consumed for a very little amount of useful work (performing transaction in a blockchain).
- A risk that parties which have supercomputers and computational data centers can break the network by addressing all their computational capabilities to it.

Proof of stake algorithm solves this problem by introducing virtual mining – a pseudo-random deterministic way of identifying which party can produce the next block, based on numerous factors. The primary factor which Proof of Stake takes into account is the party's stake – the number of virtual currency the party owns. In Ethereum, for example, this currency is Ether. This means the more currency the party has, the more blocks it can produce [17].

Proof of Stake doesn't solve the problem of one party owning almost all currency, as well as Proof of Work doesn't solve the problem of one party owning all computational powers. However, it is more efficient as it requires a very few computing resources when compared to Proof of Work.

Delegated Proof of Stake is one of the most complex consensus algorithms widely used today. It works similar to Proof of Stake with the difference in the next block producer selection algorithm. Delegated Proof of Stake block producer mechanism works more like an election, where all network participants can vote for block producers using their in-network currency, thus, increasing their chances to become actual block producers [18].

Unlike Proof of Stake, there isa limited number of block producer involved, which allows higher scalability comparing to Proof of Work and Proof of Stake, as only a limited number of network participants do actual computations, while others just consume them as is.

2.3 Typical Access Control Models

Because there are tasks that one particular decentralized data platform cannot handle, such as heavy computations or high throughput and reliability at the same time, there are several different approaches of how the decentralized network can be set up in order to fit the needs [5].

Public Decentralized Data Platforms. Public decentralized data platform is a decentralized data platform intended for a public use, meaning that any party can transact in its network. Just like the internet, but immutable, predictable and censorship-resistant. Because the network of a decentralized data platform is public, any network participant can transparently see and verify each transaction in it, ensuring that the data was not corrupted or changed [19, 20].

Ethereum is one of the most well-known public decentralized data platforms, which is able to handle a maximum of 15 transactions per second. Handling more transactions per second makes the system more centralized, as more resourced is required to support the network, while not every network participant can afford to have this resources available.

Public decentralized data platforms like Ethereum allow providing a maximal trust to a software solution based on them. However, sometimes, these platforms are not scalable enough, and for some cases, there is a point in using private or permissioned platforms.

Private Decentralized Data Platforms. Because public decentralized data platforms cannot afford big throughput due to security reasons, there is an option to launch a private network, which will be only accessible to a particular number of parties [20]. Making the decentralized network private means that it eventually loses one of its properties – decentralization, because it is being controlled by one party which keeps it private. However, several approaches could help to preserve trust in the system:

- Occasionally publishing a network state on a regular basis to another decentralized public ledger, which can be used to validate the state of a private network in the future.
- Allowing external provisioners to validate the network state on-demand, by introducing a public registry of backups made by a system regularly.

While private networks are closer to traditional centralized platforms, they borrow some useful properties from decentralized ones, like immutable network history and fault tolerance.

Permissioned Decentralized Data Platforms combine the properties of public and private decentralized data platforms, finding the golden mean between two. These platforms are often referenced as consortium platforms (for example, consortium blockchain). They function just like public platforms but have more restrictions on who can access the platform and, more importantly, who can perform which transactions [19, 20].

Speaking of the blockchain technology, in a permissioned decentralized system there may be a pre-defined set of parties who can assemble blocks. For instance, in a bank system where more than 10 banks are involved, each bank as a network participant can have an equal weight, leading to everyone have the same influence on the network. The rules are strictly defined in a system, and once one instance does not follow these rules it exists to the chain which is considered invalid by all other parties, thus leaving the network.

The biggest example of permissioned blockchain today is Hyperledger Fabric, providing a flexible blockchain-based platform that can be set up for almost any needs. While it is highly configurable, there is still no possible configuration in which the blockchain will be scalable, highly available and secure at the same time.

3 Security Model for High-Risk Systems

Decentralized data platform application can bring the following properties to the high-risk systems:

1. Data security and integrity.
2. Data immutability and transparent history of changes.
3. Reliable and fault-resistant storage.
4. Reliable and predictable data processing.

Regarding decision support systems, there are several possible applications. We describe 2 of them, based on a fully public access control model and semi-public permissioned model.

Private and consortium security models of decentralized data platforms are quite challenging to set up properly, so we do not recommend setting them up for individual organizations, rather than joining to already existing private or consortium decentralized data platforms.

3.1 System Security Model Based on Public Decentralized Data Platforms

One of the main applications is to build a system fully on top of a decentralized platform. This is always the most desirable scenario, however, it has two disadvantages:

1. It requires developing a solution which doesn’t have functional bugs or special method development which allows updating the solution without further centralization risks.

2. The reliable public decentralized platform which solves a scalability problem does not exist yet, which limits the number of operations that can be performed in a decentralized ledger.

In a case of the full decentralization, regarding decision support system, experts are using the client which is connected to a decentralized data platform directly, making the direct contributions to a global public ledger of a decentralized data platform. The entered data becomes immutable immediately after the entry, and all its possible future changes are practically excluded. The diagram demonstrating the process of information entry, processing and verifying is depicted in figure 4.

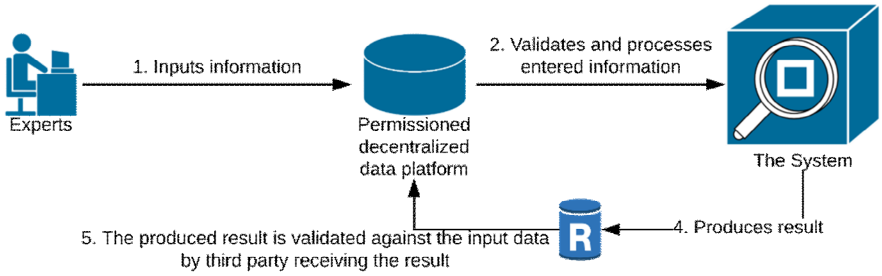
Experts
1. Inputs information
Permissioned decentralized data platform
2. Validates and processes entered information
The System
4. Produces result
R
5. The produced result is validated against the input data by third party receiving the result

Fig. 4. System security model where experts directly submit information to a permissioned decentralized system.

The client used for the data entry by experts uses the decentralized identity – an account in terms of the decentralized data platform. These accounts are registered before the data entry

Thus, after the result is processed by the decision support system, each recommendation given by the system is compared and verified against input data, which is 100% legit. Moreover, in case of decentralized data platform can handle the load and computing resources required to process the input data and produce a decision, the whole decision support system can be built on top of the decentralized data platform.

3.2 System Security Model Based on Unscalable Public Decentralized Data Platforms

Taking into account that the public security model currently does not fit all the needs of a scalable and, as a result, reliable system, we suggest another approach to creating a reliable and secure trustless system for high-risk decision support systems based on permissioned decentralized data platforms.

This security model decreases the required space of stored data within the decentralized platform, making it cheaper to use. For example, it can reduce the cost of each complete decision support process to $0.03 in equivalent, when using Ethereum decentralized platform usage (the average cost of one transaction in the main Ethereum network, as of 11/8/2018).

In this security model, the data is entered into the system using a regular centralized database. After the data is entered by all experts, this data is cryptographically hashed with a strong hashing algorithm and is published to a public decentralized data platform as a proof of the data state. Later, the input information is disclosed and experts confirm that the resulting hash corresponds to an input data indeed. This data can be confirmed by any provisioning expert and saved for further proof that the output was produced from the legit input. This procedure is somehow similar to obtaining a digital signature for every information entered to the system, but without a practical way to change the signature itself.

Figure 5 demonstrates this approach. The only information that goes to a decentralized secure ledger is a hashed information, which stands as a proof of legit input data that is compared to the original input after the decision support process is finished.

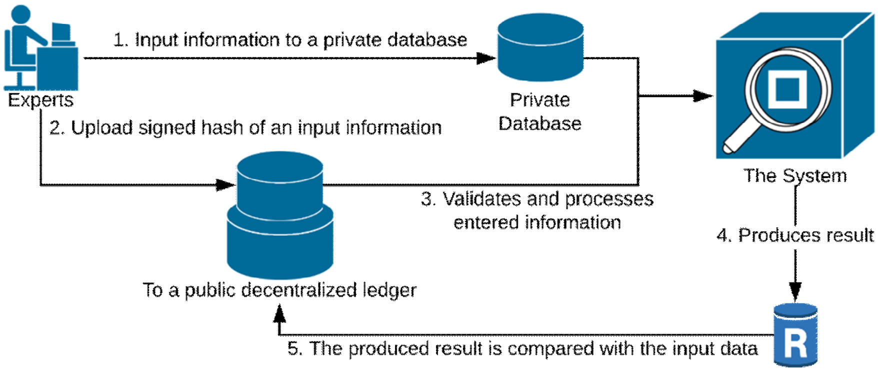
Experts
1. Input information to a private database
Private Database
2. Upload signed hash of an input information
The System
3. Validates and processes entered information
4. Produces result
To a public decentralized ledger
5. The produced result is compared with the input data
R

Fig. 5. System security model example where only hashed information is submitted to a decentralized data platform, making it cheaper to operate

In this scenario it is important to mention that the original data can be lost, resulting in having a hash that does not correspond to any data. But at least, for the consumer, this will mean that the process of decision support was not organized properly and the produced decision can be forged.

Conclusion

High-risk systems, including decision support systems, require the most secure program and infrastructure environment to function. Decentralized data platforms, which is an emerging technology nowadays, is the only way for systems and applications to *practically* exclude any tampering risks and intrusion possibility. By utilizing the decentralized security model proposed in this article high-risk systems can expect their data and processing to be safe and predictable regardless of how the internal system is developed and maintained.

References

1. Saaty, T. L. (2010). Principia Mathematica Decernendi - Mathematical principles of decision making - Generalization of the Analytic Network Process to neural firing and synthesis. Pittsburg: RWS Publications.
2. Tsyganok V., Kadenko S., Andriychuk O., Roik P. Usage of multicriteria decision-making support arsenal for strategic planning in environmental protection sphere */ Journal of Multi-Criteria Decision Analysis*. 2017;**24**:227–238.
3. Driscoll, J.W., 1978. Trust and participation in organizational decision making as predictors of satisfaction. Academy of management journal, 21(1), pp.44-56.
4. Tsyganok V.V., Kadenko S.V. & Andriichuk O.V. Using different pair-wise comparison scales for developing industrial strategies. *International Journal of Management and Decision Making*. – 2015. – vol. 14, issue 3. – P. 224-250.
5. Tsyganok V.V.,KadenkoS.V., Andriichuk O.V. Usage of Scales with Different Number of Grades for Pair Comparisons in Decision Support Systems / *International Journal of the Analytic Hierarchy Process*. – 2016. – vol.8, issue 1. – P.112-130.
6. Андрійчук О.В. Метод змістової ідентифікації об'єктів баз знань систем підтримки прийняття рішень / *Реєстрація, зберігання і обробкаданих.* – 2014, - Т.16, №1 – С.65-78.
7. Dhillon, G. and Torkzadeh, G., 2006. Value□focused assessment of information system security in organizations. Information Systems Journal, 16(3), pp.293-314.
8. Pilkington, M., 2016. 11 Blockchain technology: principles and applications. Research handbook on digital transformations, p.225.
9. Puthal, D., Malik, N., Mohanty, S.P., Kougianos, E. and Yang, C., 2018. The blockchain as a decentralized security framework. IEEE Consum. Electron. Mag., 7(2), pp.18-21.
10. Seebacher, S. and Schüritz, R., 2017, May. Blockchain technology as an enabler of service systems: A structured literature review. In International Conference on Exploring Services Science (pp. 12-23). Springer, Cham.
11. Swan, M., 2015. Blockchain: Blueprint for a new economy. "O'Reilly Media, Inc.".
12. Nakamoto, S., 2008. Bitcoin: A peer-to-peer electronic cash system.
13. Wood, G., 2014. Ethereum: A secure decentralisedgeneralised transaction ledger. Ethereum project yellow paper, 151, pp.1-32.
14. Croman, K., Decker, C., Eyal, I., Gencer, A.E., Juels, A., Kosba, A., Miller, A., Saxena, P., Shi, E., Sirer, E.G. and Song, D., 2016, February. On scaling decentralized blockchains. In International Conference on Financial Cryptography and Data Security (pp. 106-125). Springer, Berlin, Heidelberg.
15. Hoxha, L., 2018. Hashgraph the Future of Decentralized Technology and the End of Blockchain. European Journal of Formal Sciences and Engineering, 1(2), pp.29-32.
16. Popov, S., 2016. The tangle Whitepaper. cit. on, p.131.
17. Zheng, Z., Xie, S., Dai, H., Chen, X. and Wang, H., 2017, June. An overview of blockchain technology: Architecture, consensus, and future trends. In Big Data (BigData Congress), 2017 IEEE International Congress on (pp. 557-564). IEEE.
18. Vukolić, M., 2015, October. The quest for scalable blockchain fabric: Proof-of-work vs. BFT replication. In International Workshop on Open Problems in Network Security (pp. 112-125). Springer, Cham.
19. O'Dwyer, K.J. and Malone, D., 2014. Bitcoin mining and its energy footprint.
20. Bentov, I., Gabizon, A. and Mizrahi, A., 2016, February. Cryptocurrencies without proof of work. In International Conference on Financial Cryptography and Data Security (pp. 142-157). Springer, Berlin, Heidelberg.

21. Larimer, D., 2014. Delegated Proof of Stake. Bitshares. org. From Bitshares.org, last accessed 2016/11/21.
22. Cachin, C., 2016, July. Architecture of the hyperledger blockchain fabric. In Workshop on Distributed Cryptocurrencies and Consensus Ledgers (Vol. 310).
23. Zheng, Z., Xie, S., Dai, H.N. and Wang, H., 2016. Blockchain challenges and opportunities: A survey. Work Pap.–2016.
24. LNCS Homepage, http://www.springer.com/lncs, last accessed 2016/11/21.