Human-based Consensus for Trust Installation in Ontologies

Christoph Summerer [b], Emanuel Regnath [a,*], Hans Ehm [b], and Sebastian Steinhorst [a]
[a] *Embedded Systems and Internet of Things, Technical University of Munich, Germany*
E-mails: emanuel.regnath@tum.de, sebastian.steinhorst@tum.de
[b] *Corporate Supply Chain Engineering Innovation, Infineon Technologies AG, Germany*
E-mails: christoph.summerer@gmx.de, hans.ehm@infineon.com

Abstract. Blockchain technologies enable a decentralized peer-to-peer network to reach distributed consensus on transaction data that is written into a blockchain. This data is then considered to be a single source of truth, trusted by the entire network. Many approaches focus on writing financial transaction data into the blockchain, which can be easily verified and validated by machines to reach distributed consensus. However, there exist also other types of data which requires human thinking and collaboration for validating and finding consensus. This is the case for ontologies, which are important building blocks for Semantic Web content but are currently difficult to validate and maintain and would therefore benefit from the guarantees provided by blockchain.

In this paper, we propose a novel protocol to represent the human factor on a blockchain environment. Our approach allows single or groups of humans to propose data in blocks which are verified and validated by other humans. Only if human-based consensus on the correctness and trustworthiness of the data is reached, the new block is appended to the blockchain.

Our experimental results show that this human approach is an alternative to conventional approaches but significantly extends the possibilities of blockchain applications on data that cannot be verified and validated automatically but requires human knowledge and collaboration.

Keywords: Blockchain, Consensus, Semantic Web, Ontologies, Trust

1. Introduction

The blockchain technology allows a decentralized network to agree on one global state and accept it as a trusted single source of truth. For this, (financial) transaction data is written into blocks, which are connected to each other and, by this, build a chain. Each new block secures the order and integrity of the previous blocks. The use of cryptography and hashing ensures immutability of data and, in addition, offers a high degree of transparency and traceability. Those characteristics make the blockchain technology a big "trust machine" [1]. While crypto-currencies such as Bitcoin [2] or Ethereum [3] focus on storing transactions of financial assets in the blockchain, in general this is also possible for other data types that should be immutably and transparently stored in a distributed ledger in order to exploit the blockchain characteristics, i.e. to provide a single source of truth, agreed on and trusted by an entire decentralized peer-to-peer network. However, depending on the type of content, it can often not be verified and validated automatically but requires human thinking and collaboration. This human factor not only influences the way of applying blockchain technologies for such content that cannot easily be classified as wrong or right, but also the way of reaching distributed consensus on it. For that reason, it is necessary to extend conventional blockchain and consensus processes to that human factor.

Blockchain and Semantic Web One example for this is the use of blockchain technologies to exploit their properties for the trust installation in Semantic Web content, so-called ontologies. These represent linked

*Corresponding author. E-mail: emanuel.regnath@tum.de.

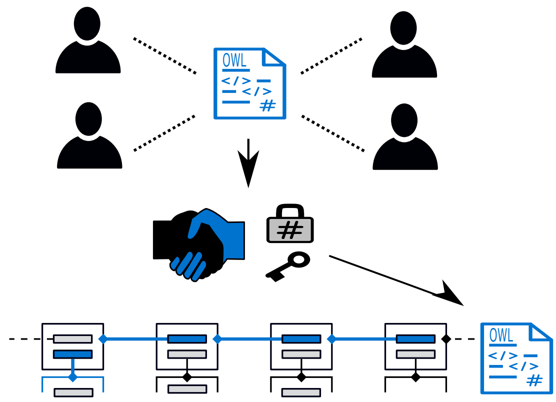

Figure 1. Concept of our human-based consensus approach: one or more human experts collaborate on an ontology file (OWL-file) and send it for verification and validation to a private blockchain network consisting of other human experts. If those agree with the proposed version of the ontology, it is appended to the blockchain where it is considered as a trusted single source of truth.

data that can be read and understood by both, humans and machines. Since there are currently only a few standards available, these ontologies are under constant development and there is no standardized way to install trust into them yet. In order to install trust into an ontology, it is desirable to track all changes applied to such an ontology on a blockchain and make use of the immutability and traceability properties of this technology. To ensure that those changes are really correct, which is a prerequisite for trust installation, the content has to be verified and validated before it can be written into the blockchain. Then, the last block in the chain represents the latest accepted and trusted version of the ontology.

However, these ontologies can only be verified and validated by human experts and there is currently no mechanism to integrate human verification and validation into a blockchain architecture. As a result, it is not possible to use human verification and validation of data and blockchain security together, which could lead to a chaotic and inconsistent Semantic Web development where companies or individuals only trust their own ontologies and there exist many quasi-standards at the same time. To prevent this, we developed a first approach that aligns the processes on a blockchain to this human factor in order to harmonize ontology creation across several domains and stakeholders.

1.1. Contributions

We propose the combination of blockchain technologies with human verification, validation, and confirmation of data. We investigate this topic as changes applied to an ontology, to reach distributed consensus. This way, only data that has been considered to be correct and trustworthy by a majority of human experts is written into the blockchain. In particular, we enable

- the single and joint proposal-making of changes applied to an ontology,
- the stake adjustment towards the size and impact of proposed changes in an ontology,
- and the human verification, validation, and consensus-finding on changes in an ontology that eventually results in a final block representing the latest trusted single source of truth.

Our approach, illustrated in Figure 1, maps the human factor to a blockchain environment, enabling human collaboration and consensus-finding on proposals regarding changes applied to an ontology. This human verification, validation, and confirmation of data enables the use of blockchain technologies in areas apart from financial transaction data.

1.2. Blockchain, Consensus, and Semantic Web

Blockchain Formally, the blockchain $\mathbf{C}$ is a distributed database that stores data in blocks $\mathcal{B}_i$ ordered over time. The length of $\mathbf{C}$ is n and i represents the index of block $\mathcal{B}_i$ [4].

$$\mathbf{C} = \{\mathcal{B}_i | i \in 1, \ldots, n\} \tag{1}$$

Before a new block can be added to the blockchain, we consider it as a block proposal $\mathcal{P}_i$. We define a set of validators V that verify the block proposals for correctness and reach consensus on this.

After verification, each new block that is appended to the blockchain reconfirms the data of the preceding block(s) by including the cryptographic hash of the previous block in its own block data. Therefore, a hash function $hash(m)$ is applied that maps arbitrary input data m to a bit-string of fixed size h. This process is one-directional and considered as unique as every change in the input results in a different output hash. The only way to recreate the input data is by trial and error. This secures the integrity and ordering of the blocks in the chain as any change to the data of an existing block would result in different hashes of the consecutive blocks. In addition to hashing, asymmetric encryption in form of public-private-keys and digital signatures ensure a high level of security.

Consensus Instead of relying on a central authority to coordinate processes, distributed consensus methods are applied to ensure agreement within a decentralized peer-to-peer network. This network can be either accessible by everyone (public) or restricted to a certain amount of participants (private). Distributed consensus protocols then ensure that everyone in the network agrees that the information in the blockchain is true. For this, it is necessary that the actors reach consensus on what is to be written in which order into the blockchain. This makes sure that all nodes hold the same global state of the blockchain that is considered as a trusted single source of truth. Depending on the field of application and the composition of the network, there are different ways to achieve this distributed consensus. In general, a protocol is said to solve the consensus problem if three properties hold [5], [6]:

- Agreement: All correct nodes decide the same value.
- Integrity: All correct nodes decide only once.
- Termination: All correct nodes decide before timeout.

Agreement and integrity are safety properties, whereas termination is a liveness property, as defined by [7]. In addition to safety and liveness, byzantine fault-tolerance as introduced in [8] also plays an important role in many protocols, i.e. the possibility of achieving distributed consensus despite faulty components in the network. However, a study, known as the FLP Impossibility Result, states that no deterministic consensus protocol can guarantee all three properties in a fully asynchronous system [9].

Semantic Web The consensus problem, however, brings some challenges if it is to be applied to Semantic Web content, which is defined by ontologies in the Web Ontology Language (OWL). Those ontologies represent linked subject-predicate-object relationships, so-called Resource Description Framework (RDF) triples, which are not only readable and understandable for humans but also for machines [10]. There are very little standardized ontologies at the moment, which results in a dynamic further development of existing ones. To determine whether ontologies and changes applied to them are correct or not, it requires human knowledge to verify, validate, and confirm that. Conventional consensus protocols do not consider this human impact as they are designed to automatically verify and validate rather simple (financial) transaction data, for example by solving a cryptographic puzzle as in Proof of Work (PoW), the consensus protocol used by Bitcoin [2]. In order to achieve distributed consensus on the correctness and trustworthiness of ontologies, to write them into a blockchain and by this exploit the confidence-building characteristics of such, an approach must be developed that maps this human factor to a blockchain environment.

2. Our Human-based Consensus Approach

For that reason, we propose the combination of blockchain technologies with human verification, validation, and confirmation of ontology data to reach distributed human-based consensus on such and thereby install trust into the content.

For this, we build a private blockchain network consisting of human experts that are familiar with the domain of the considered ontology and build the validator set V. Each participant or even a group of participants in the network can propose changes or improvements in the ontology by including them into a block proposal $\mathcal{P}_i$, which are then submitted to the other participants (V) for verification and validation. For this purpose, a distributed consensus method is extended by the human factor.

Consensus Algorithm We propose to base this human-based consensus on Practical Byzantine Fault Tolerance (PBFT) [11], which ensures that proposals arrive and are processed in the correct order by all human peers in the network, as shown in Figure 2. This PBFT consen-

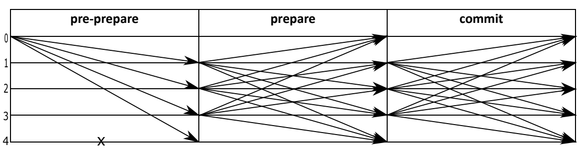

Figure 2. PBFT Consensus Algorithm. Figure inspired by [11].

sus is then extended by a voting mechanism that allows the human validators V in the network to give feedback on the proposals. After controlling the proposal $\mathcal{P}_i$, they

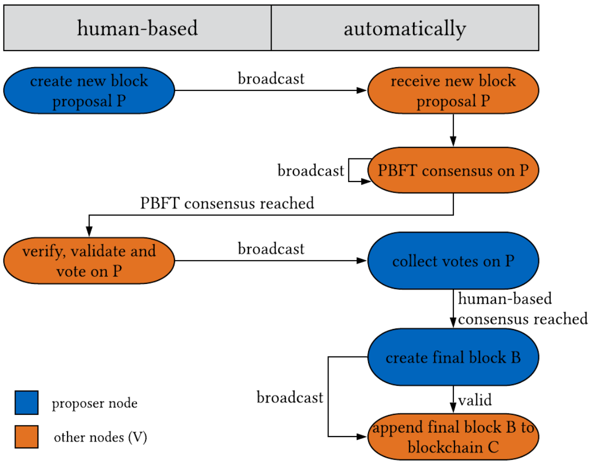

Figure 3. Human-based consensus approach.

can either agree or disagree with it by casting a positive or negative vote. Should a two-thirds majority of human experts in the network consider the proposed changes in the ontology to be correct within a given time, distributed human-based consensus is reached and the proposed version of the ontology is integrated into a new final block $\mathcal{B}_i$, which is then appended to the blockchain **C**. The last block $\mathcal{P}_n$ in **C** then represents and contains the latest accepted and trusted version of this ontology. This procedure is shown in Figure 3. Thereby, the properties of the blockchain technology are profitably used for the creation of trust into an ontology. Any change in an ontology that was validated and accepted by a majority of human experts (V) is transparently, immutably, and verifiable stored in a block $\mathcal{B}_i$ in the blockchain **C**. A user can trace the entire change history of the ontology transparently. The human-based consensus ensures that only content that has been confirmed by at least a two-thirds majority of human experts is written into the blockchain. Thereby, the content is not limited to Semantic Web content, i.e. ontologies. Our human-based consensus could be applied for any kind of data that needs to be verified, validated and confirmed by humans before it can be written into a blockchain.

Token System To handle the process flow as shown in Figure 3 and provide a compensation and incentive for the human effort and time that is necessary to create, verify, and validate data or content that should be written into the blockchain, we propose a non-monetary token system **T**, based on stake $\mathcal{S}$ and reward $\mathcal{R}$.

$$\mathbf{T} = \{\mathcal{S}, \mathcal{R}\} \tag{2}$$

We distinguish between reward tokens for the proposer of proposal $\mathcal{P}_i$, $\mathcal{R}_P$, and reward tokens for the validators in V, $\mathcal{R}_V$.

$$\begin{aligned} \mathcal{R}_V &= 1 \cdot \mathcal{S} \\ \mathcal{R}_P &= 3 \cdot \mathcal{S} \end{aligned} \tag{3}$$

For each proposal, a certain number of tokens has to be deposited as a stake $\mathcal{S}$. This number of tokens is adjusted to the size and impact of the proposal as larger changes in the data require on the one hand more effort in the creation, and on the other hand also more effort for their verification and validation. If the proposal is rejected, this stake gets lost. This prevents the network from being flooded with too many proposals. In contrast, if the proposal is accepted by a two-thirds majority, the proposer gets rewarded by a multiple of the deposited stake $\mathcal{R}_P$ and also the validators get a reward in form of tokens $\mathcal{R}_V$.
In the case of joint proposals, i.e. proposals made by a group of peers, the stake and the reward is equally distributed among the involved peers. By this, a human peer that actively participates in the network by making high-quality proposals and validating such, results to have a higher token balance than another peer that makes low-quality proposals and spends no effort in verification and validation of proposed data. Therefore, the token balance of each peer can be also considered as a kind of reputation value. This token system **T**, based on stake $\mathcal{S}$ and reward $\mathcal{R}$, allows only peers with a certain amount of tokens, gained by honest and active participation in the network, to make new proposals $\mathcal{P}_i$. Malicious participants will be denied this opportunity due to the lack of tokens to be deposited as a stake. In addition, those peers can be removed from the network by a two-thirds majority decision. New peers can also be added in the same way. Our approach creates a network that manages itself dynamically and whose expertise guarantees the correctness of the data in the blockchain by means of human-based consensus.

Metrics To make the human-based decisions more transparent and traceable, we introduce to measure some processes and include the results directly in the block data. By this, the last block in the chain does not only represent the latest accepted and trusted version of human-confirmed data, but also metrics that give information about how this was achieved. For this purpose, we propose to add information about the proposer and information about the proposal itself. Thereby, also external consumers of data can understand who made which proposal $\mathcal{P}_i$ at which time and by how many other humans in V it was validated and considered to be correct and trustworthy. We further measure how much time Δt it took to reach a majority for each proposal $\mathcal{P}_i$ to result in a final new block $\mathcal{B}_i$ and how many tokens $\mathbf{T}$ were involved in the processes as stake $\mathcal{S}$ and reward $\mathcal{R}$. This is to further increase trust in the data, since it has not only been verified and validated by humans but is also traceable as to how exactly the human-based consensus came about.

2.1. Implementation

For the implementation of our approach, conventional blockchain technologies and distributed consensus methods could not be applied as they do not offer any mechanism to integrate human verification and validation of data which is necessary to ensure the correctness and, therefore, trustworthiness of the content that should be written in the blockchain. Hence, we implemented a prototype of our human-based consensus approach by ourselves, using the Go programming language, which offers advantages in terms of speed, platform interoperability, multi-threading, safety and user-friendliness. The code of our implementation is publicly available at [12] for review and further research. We also made use of go-libp2p, a modular network stack that allows different transport protocols, multiplexing and sockets, encrypted connections and communications, publish-subscribe, runtime freedom, and peer discovery and routing. Packages for cryptography and other blockchain-related features complete the tool list. Furthermore, we based our implementation on the Inter Planetary File System (IPFS), a distributed file system that allows to store and share data within a decentralized peer-to-peer network. This allows us to make use of the unique peerID provided by the IPFS to unambiguously identify the peers in the network. In addition, we can store the ontology data off-chain by putting only its unique hash $hash(m)$ in the block data of $\mathcal{P}_i$ and $\mathcal{B}_i$ to keep the message exchange data low and save storage space for the blockchain. By this, we implemented a private blockchain network where peer connections are based on Transmission Control Protocol (TCP) streams. Peers are uniquely identified by their IPFS peerID. The network can be dynamically reduced or extended by peers if a two-thirds majority of existing peers in V agrees. Each peer or even a group of them can propose new blocks $\mathcal{P}_i$ as long as the token balance allows to deposit the necessary stake $\mathcal{S}$. The message-exchange for the human-based consensus is realized by go-libp2p's decentralized publish-subscribe solution, called gossipsub. Messages can also be directly addressed to single peers via their unique IPFS peerIDs. The message flow as shown in Figure 3 is realized by two concurrent functions, so-called goroutines. One function handles the human input while the other function handles the underlying processes. Messages are always encrypted and provided with information about sender and subject to identify and process them correctly. If human-based consensus is reached, the blockchain $\mathbf{C}$ contains blocks $\mathcal{B}_i$ where the last block $\mathcal{B}_n$ represents the latest trusted and agreed on version of the ontology. Furthermore, we provide metrics about the proposer and the consensus-finding process itself to increase the transparency. Figure 4 shows the pseudo-code of our block structure.

The analytical and experimental results of our implementation are evaluated and discussed in Section 4.

```
// Definition of type "Block"
type Block struct {    // block B
Index           int    // index i
Timestamp       string // time of final
    ↪ block B creation
File            string // IPFS hash
Proposer        string // IPFS peerID
AuthorMetrics   string // token balance,
    ↪ number of P, etc.
ProposalMetrics string // delta t, stake,
    ↪ number of V
NetworkMetrics  string // number of peers in
    ↪ the network
PrevHash        string // hash(B), i-1
}

// Blockchain: a slice of type "Block"
var Blockchain []Block // blockchain C
```

Figure 4. Pseudo-code of the block structure used in our implementation.

3. Related Work

Consensus-based Ontologies The need for consensus as a necessary prerequisite for trust installation into ontologies was already recognized by Nagy and Vargas-Vera in [13] and [14] before the blockchain technology became popular. The authors come to the conclusion that contradictory interpretations of Semantic Web content (ontologies) are counterproductive for the installation of trust and propose a fuzzy voting model in order to achieve a coherent state that represents the democratic majority opinion about the Semantic Web content in question. They rely on the assumption that a majority of a group of voters is more likely to make the right decision than a random single voter is. In [15], Duong et al. take up this approach but focus not only on finding consensus on ontologies, but also on considering the consensus quality. From an original version of an ontology, branches are created and edited by individual experts. These are then to be merged into a new, improved version of the ontology. Reaching consensus within the group of editors is decisive for this. Therefore, distance values between the different branches are measured and used as an indicator for consensus quality before a new version of the ontology is merged. This approach focuses on the collaborative processing of ontologies but does not consider any incentives for the experts being involved in the process. Furthermore, there is no mechanism that prevents the network from being flooded with too many branches. In addition, neither the approach of Nagy and Vargas-Vera nor the one of Duong et al. is intended to be applied on a blockchain environment.

Blockchain-Secured Ontologies Iancu and Sandu propose to apply blockchain technologies to implement the trust layer of the Semantic Web [16]. Following their idea, the blockchain's immutability and transparency properties enable to certify and track every change in an ontology or single ontology statements, i.e. RDF triples. For domain-specific ontologies, they suggest permissioned blockchains, otherwise unpermissioned ones were the better choice. However, there is no consensus part in their approach. Everybody with access to the respective blockchain network can make changes in an ontology and write them into the blockchain. Whoever wants to use the ontology then, be it a human or a computer, must decide for oneself whether the author of the changes is trustworthy and whether the changes he or she made are really correct. There is no decision-making aid for contradictory information. In that approach, the blockchain is more used as a distributed logbook rather than acting as a trust machine providing a single source of truth.

To reduce this weakness, Fill and Haerer propose the concept of Knowledge Blockchains in [17] to track who added what change at what time. In order to guarantee the correctness of the changes, access rights are assigned to people who can only modify certain parts in an ontology for which they have a permission. In addition, automatic checks should be carried out to identify inconsistencies in applied modifications in order to ensure a high quality of the blockchain entries. To show the practical application of their concept, they created a prototypical implementation based on the ADOxx library that, however, does not cover all proposed functionalities. An extension and full evaluation of their concept and the corresponding implementation was envisaged for future work. Nevertheless, also in this work no distributed consensus is used to create a fair ordering and agreement on the applied modifications.

Blockchain-Consensus Combination Using blockchain and consensus in combination is roughly suggested by Hoffman et al. in [18], however not for Semantic Web content but for academic publications. The authors propose that blockchain technologies could track the interactions of scientific publishers and contributors for academic publications with the help of a smart contract that replaces a trusted third party. This would result in a platform for decentralized collaboration between humans that returns a single version of truth without relying on the power of a centralized unit like a journal or conference. That approach suggests many interesting points on how blockchain technologies can be linked to non-financial (transaction) data. This includes the off-chain storage of data as well as (financial) incentives for active and honest participation and collaboration of humans in the network. In that approach, everybody with access to the network can write data into the blockchain. In contrast to other approaches, however, the signatures of the involved actors, i.e. the authors and reviewers, are collected to express a kind of agreement, and therefore confirmation of data. By this, only content signed by a sufficient amount of actors will be considered to be correct and trustworthy. The practical applicability of this approach was demonstrated in an implementation based on Ethereum smart contracts. Nevertheless, also in that approach, no distributed consensus is found within an entire network but rather signed agreements are reached. In addition, the case of scholastic publications cannot be fully transferred to Semantic Web content (ontologies) as the engineering process of collaboratively developing an ontology is different from authors creating a scientific paper and passing the reviews before publication. The smart contract that replaces a trusted third party in [18] relies on a role model, distinguishing between authors, reviewers and annotators. Scientific papers are always created initially by the authors. They are also responsible for the further development of their work, taking into account the feedback from reviewers and annotators. This process cannot be compared with the joint development of an ontology in which the author of the ontology is not so important and accordingly an ontology can be further developed on the basis of the preliminary work of another author. The implementation of the approach by Hoffman et al. based on Ethereum smart contracts also has the disadvantage that this is associated with fluctuating costs, which are based on the dynamic gas price. This favors that new papers are proposed primarily when the costs are lowest and are processed primarily when the incentive is highest, which leads to an unbalanced environment.

4. Evaluation and Discussion

Our approach combines the benefits of human collaboration, enabled by joint proposals, and distributed human-based consensus-finding with typical blockchain advantages such as traceability and immutability of data. In contrast to many other state-of-the-art cryptocurrencies, however, we do not aim for storing transactions of financial assets in the blockchain. Instead, we focus on any type of data that needs to be verified, validated and confirmed by humans. This was tested on the example of Semantic Web content, i.e. ontologies. However, we are not restricted to that, which on the one side makes our human-based consensus approach flexible to be applied for many different fields of application. On the other side, exactly this human factor makes it difficult to evaluate our approach. The human behavior is very difficult to simulate and can vary from domain to domain. For this reason, we decided to focus on an analytical comparison of our approach with others and added a short experimental validation to prove its practical functionality.

4.1. Analytical Comparison

Table 1 compares our human-based consensus approach with other related approaches that were presented in Section 3.

Table 1
Comparison of our approach with related work.

	Consensus	**Blockchain**	**Joint proposals**
[15]	✓	✗	✓
[16]	✗	✓	✗
[17]	✗	✓	✗
[18]	(✓)	✓	✗
Our Work	✓	✓	✓

Following the work presented in section 3, especially human-based consensus on the correctness and trustworthiness of data has been identified as a prerequisite for trust installation in ontologies. Our approach is the only one that combines this human-based consensus with blockchain benefits. Furthermore, we mapped human collaboration, represented by the possibility of joint block proposals with shared risk and reward, to a blockchain environment.

Implementation The implementation of our approach uses conventional techniques and tools and is written in the Go programming language, which is also supported by Hyperledger Fabric, amongst others, to ensure interoperability with other (blockchain) technologies. Our implementation furthermore enables to dynamically adjust our network of human experts by removing or adding peers during runtime. Therefore, in comparison to other technologies, no member list has to be created in advance and maintained. Since a non-monetary token system **T** is used for the human-based consensus, the overall costs are rather low. These costs only consist of operating costs and costs for the invested working time of human experts participating in the consensus-finding process, but there are no fees or other payments involved. In addition, we do not have to struggle with exchange rate fluctuations of other crypto-currencies that are bound to real money. Also here it is hard to state concrete numbers of how much the cost is to operate the environment. In general, following [19], the operating costs are a bit higher for private blockchains than for public ones. However, since our approach considers human-based data, in comparison with (financial) transaction data it will not achieve a too high number of transactions. Furthermore, the network size will also be limited to a small number of experts, and therefore the costs will also remain manageable and will at best pay for themselves quickly when the real benefit of trust installation enabled by human-based consensus is achieved.

Liveness Guarantees Since our human-based consensus is based on a PBFT consensus method that ensures that proposals are processed in the correct order, its properties in terms of safety, liveness, fault-tolerance and transaction finality can also be used for the extension by the human factor. By this, our approach offers a high level of safety as well as immediate transaction finality. When choosing between liveness and fault-tolerance, we decided on liveness because malicious peers can be removed from the network or get punished automatically by the token system. In case of conflicts, i.e. that no majority can be reached for a new proposal, there is a timeout included to ensure that a new round will start even though the current round has not yet been finished. In that case, the current round is stopped and deposited stake is paid back to the proposer(s). However, this means that we have to rely on synchronous clocks in our network. By this, even in case of conflicts, our human-based consensus guarantees liveness.

Communication Overhead However, limitations result in terms of scalability of our approach. The PBFT consensus requires already a high message exchange which is intensified by the human factor, as proposals, votes and decisions have to be exchanged and processed in the entire decentralized peer-to-peer network. This message exchange rises with increasing the network size. Especially joint proposals cause a communication overhead as there are even more messages needed than for a single block proposal. This message exchange limits the scalability of our approach.

4.2. Experimental Validation

We tested the implementation of our approach on the example of the Digital Reference, an ontology developed at Infineon Technologies AG representing the semiconductor supply chain and supply chains containing semiconductors. Real measurements on the TCP ports of connected peers in our simulated private blockchain network consisting of up to ten human participants showed that the number of received and sent bytes of messages containing block proposals, votes and final blocks, increases linearly with the number of connected and involved peers in the network. This is especially an issue for the peer proposing a new block since this one has to process most of the related message data in the implementation of our approach, which can be seen in Figure 5. At a certain network size, this could become a problem, limiting the scalability of our approach.

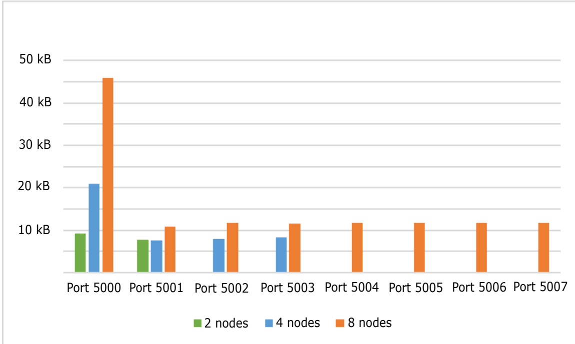

Figure 5. Measured TCP bytes for a single proposal at ports of peers in different network sizes.

The human factor in the consensus process has also an impact on the performance. The human verification, validation, and confirmation of data requires not only a high number of messages as discussed in the analytical section, but also far more time compared with the automatically checked (financial) transaction data considered in other approaches. Our experiments have shown that, depending on network size and composition as well as proposal and data type, the time between proposal-making and consensus-finding can vary between seconds and minutes or even hours and days. This can be limited by setting a timeout, assuming we are in a synchronous network. However, if the timeout is set to a shorter time frame, the experts may not have enough time to verify and validate the proposals correctly. In contrast, setting the timeout to a higher time frame means also delaying the entire consensus-finding process, especially in case of conflicts. In addition, data can only be verified and validated if there is any proposal. This means that no fixed transaction rate or latency can be calculated or specified.

To sum up, our approach strongly depends on the behavior of the human experts in the network. This behavior is hardly predictable, which is why no fixed information on performance, transaction rate etc. can be given. The human factor of this approach in combination with the used PBFT consensus method can cause larger amounts of messages to be exchanged and processed, depending on the number of peers, the structure of the network and the number of (parallel) proposals in it. Especially the proposing peer, which in our implementation processes the majority of related messages, has to deal with the increasing message exchange that results from scaling the network size. The number of messages to be exchanged may therefore tend to be slightly higher than with other blockchain technologies. However, we are currently, to the best of our knowledge, the only approach that combines the benefits of human collaboration, human-based consensus and blockchain technologies at the same time.

5. Conclusion

We proposed the use of blockchain technologies in combination with human verification, validation, and confirmation of data to reach distributed consensus on such and thereby install trust into ontologies. Our human-based consensus method to ensure correctness and trustworthiness of data in combination with the exploitation of blockchain characteristics is a powerful combination to create confidence in non-financial transaction data. Our token system **T** and the on-chain provided metrics ensure that processes run even though humans are involved, and make them more transparent. The human-based consensus ensures that the last block $\mathcal{B}_n$ in the blockchain **C** contains the latest trusted version of an ontology, a single source of truth. Our approach is, to the best of our knowledge, the only one combining human collaboration, human-based consensus-finding on ontologies and blockchain technologies at the same time. The implementation of our approach has shown that it can be practically applied. However, the human factor restricts its performance and scalability. Nevertheless, our approach represents an innovative way to install trust in data by reaching human-based consensus on its correctness and exploiting the benefits of blockchain technologies.

References

[1] Economist, The trust machine - The promise of the blockchain, (Accessed on 07/08/2019).

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

[3] G. Wood et al., Ethereum: A secure decentralised generalised transaction ledger, *Ethereum project yellow paper* **151**(2014) (2014), 1–32.

[4] J. Garay, A. Kiayias and N. Leonardos, The bitcoin backbone protocol: Analysis and applications, in: *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, Springer, 2015, pp. 281–310.

[5] L. BAIRD, The swirlds hashgraph consensus algorithm: fair, fast, byzantine fault tolerance, Technical Report, SWIRLDS-TR-2016-01.

[6] M. Correia, G.S. Veronese, N.F. Neves and P. Verissimo, Byzantine Consensus in Asynchronous Message-Passing Systems: a Survey, *Int. J. Crit. Comput.-Based Syst.* **2**(2) (2011), 141–161. doi:10.1504/IJCCBS.2011.041257.

[7] L. Lamport, Proving the correctness of multiprocess programs, *IEEE transactions on software engineering* (1977), 125–143.

[8] L. Lamport, R. Shostak and M. Pease, The Byzantine generals problem, *ACM Transactions on Programming Languages and Systems (TOPLAS)* **4**(3) (1982), 382–401.

[9] M.J. Fischer, N.A. Lynch and M.S. Paterson, Impossibility of distributed consensus with one faulty process, Technical Report, Massachusetts Institute of Technology, Cambridge Lab for Computer Science, 1982.

[10] P. Hitzler, M. Krotzsch and S. Rudolph, Foundations of semantic web technologies (2009).

[11] M. Castro, B. Liskov et al., Practical Byzantine fault tolerance, in: *OSDI*, Vol. 99, 1999, pp. 173–186.

[12] TUM-ESI, Our Prototypical Implementation in Go, https://github.com/tum-esi/human-bc-consensus.

[13] M. Nagy and M. Vargas-Vera, Reaching consensus over contradictory interpretation of semantic web data for ontology mapping, in: *Intelligent Computer Communication and Processing, 2009. ICCP 2009. IEEE 5th International Conference on*, IEEE, 2009, pp. 63–66.

[14] M. Nagy, M. Vargas-Vera and E. Motta, Managing conflicting beliefs with fuzzy trust on the semantic web, in: *Mexican International Conference on Artificial Intelligence*, Springer, 2008, pp. 827–837.

[15] T.H. Duong, M.Q. Tran and T.P.T. Nguyen, Collaborative Vietnamese WordNet building using consensus quality, *Vietnam Journal of Computer Science* **4**(2) (2017), 85–96.

[16] B. Iancu and C. Sandu, A Cryptographic Approach for Implementing Semantic Web's Trust Layer, in: *International Conference for Information Technology and Communications*, Springer, 2016, pp. 127–136.

[17] H.-G. Fill and F. Härer, Knowledge Blockchains: Applying Blockchain Technologies to Enterprise Modeling, 2018. doi:10.24251/HICSS.2018.509.

[18] M.R. Hoffman, L.-D. Ibáñez, H. Fryer and E. Simperl, Smart Papers: Dynamic Publications on the Blockchain, in: *European Semantic Web Conference*, Springer, 2018, pp. 304–318.

[19] E..Y. LLP, Total cost of ownership for blockchain solutions, 2019, (Accessed on 03/08/2020).

[20] L. Mucciaccio, L. Piccicuto and R. Derie, Avence White Paper, 2017, (Accessed on 06/20/2019).

[21] K. Janowicz, B. Regalia, P. Hitzler, G. Mai, S. Delbecque, M. Fröhlich, P. Martinent and T. Lazarus, On the prospects of blockchain and distributed ledger technologies for open science and academic publishing, *Semantic Web* (2018), 1–11.