Survey on Performance evaluation of different Distributed Ledger Technologies (DLTs)

Adam Jakobsson
Department of Computer and Information Science
Linköping University
Linköping, Sweden
adaja901@student.liu.se

Martin Steen-Holmberg
Department of Computer and Information Science
Linköping University
Linköping, Sweden
marho300@student.liu.se

***Abstract*—Distributed ledger technology (DLT) is a technology using peer-to-peer networks to make decentralized applications. The popularity of different Distributed ledger technologies such as Bitcoin and Ethereum are rising. This paper wants to contribute with a performance comparison between the difference of private and public DLTs, the usage of smart contracts in DLTs and which scope of usage they are best fitted for. By doing a literature study, several characteristics, which affect the DLTs, are presented to help the comparison regarding the performance. We found that there are several different sectors DLTs are usable for. The differences between private and public DLTs are many and there are different aspects considered to decide which one to use. The same is concluded for smart contracts.**

***Index Terms*—Distributed ledger technology, Smart contract, Blockchain**

I. Introduction

Distributed Ledger Technology (DLT) is the term for the usage of peer-to-peer networks to decentralize services. DLTs can offer the characteristics to ensure data integrity, authenticity and provenance (the origin). An example of a DLT is blockchain. The blockchain inherits the characteristics from the DLT and extends its own characteristics to register sales, contracts and agreements to a frequently updated list. Those characteristics were intended to support crypto-currency like Bitcoin [1]. Bitcoin has over 70 million users worldwide at the end of Mars 2021 [2] and it seems it will not stop growing in popularity. Bitcoin's popularity has made more and more different DLTs arise. To mention a few there is the Hyperledger family, Ethereum and EOS. The new DLTs have differences in terms of usage, smart contracts and if it is a public or private DLT. The combination of those differences will generate different tradeoffs, which this paper aims to investigate. The main questions for this paper to investigate are:

- RQ1: How do different DLTs tradeoffs differ when using smart contracts?
- RQ2: How do different DLTs tradeoffs differ between using private vs public DLTs?
- RQ3: Where are the DLTs, compared in the questions above, best suited to use?

The research aims to investigate the recent development of DLTs and will not be going into their history. Therefore, most of the sources used will be from within the last 8 years. Also, the picked DLTs are all of the type blockchain, are required to use smart contracts and are equally divided between public and private DLTs.

II. Background

A. Distributed Ledger Technology

A ledger is a register of transactions and the transaction is divided into blocks [3]. This ledger is shared between different nodes in a network to become a distributed ledger. Every node has a copy of the ledger to maintain by connecting it to other nodes and establishing peer-to-peer connections between the nodes in the network.

1) The researched Distributed ledgers:

Ethereum: Often called the second generation of DLTs, it introduced smart contracts and can be both private and public. It can use both Proof of Work (PoW) or Proof of Stake (PoS) while public and Proof of Work or Proof of Authority (PoA) while private. To enable smart contracts Ethereum has its own language called Solidity [3].

EOS: The first smart contract platform with the use of Delegated Proof of Stake (DPoS) [4] and counted as one of the first third generation blockchain. The intent of EOS was to make a competitor to Ethereum [3] by using DPoS, increasing scalability and higher throughput [1].

Hyperledger Fabric: The first Hyperledger to come out from the Hyperledger ecosystem, having its strong points in its modular design, pluggable features and strong privacy. Because of the pluggable features you can with Hyperledger Fabric e.g. switch consensus algorithms depending on your needs [1].

This DLT uses smart contracts, two layered architecture and can have multiple ledgers within its ecosystem. The first layer of the architecture keeps track of the identity of all members of the ledger. This helps with security, making it possible creating policies dictating what the different users are allowed to do. The second layer is for creating channels and uniquely to Fabric is the attachment of a ledger to a channel. Channels are then used to segregate the interactions between certain users from other users outside of the channel, creating the possibility for users to independently separate their transactions.

Hyperledger Sawtooth: Initially developed by Intel and bound to their Intel SGX (Software Guard Execution) because

of its use of the Proof-of-Elapsed-Time (PoET) consensus algorithm. Sawtooth security is based on the execution within secure enclaves inside of the processor and then verified by a remote attestation process. In Sawtooth you can also use transaction families, grouping allowed operations and transaction types, to encapsulate business logic (contract) [1].

Sawtooth also allows for configured thresholds, putting limitations on how much you want the DLT to process at the same time before rejecting the rest [3].

B. Consensus Algorithms

Consensus algorithms are used to reach a consensus among nodes in a network. This consensus is needed to create uniformity within a network in need of conclusions.

There are two different kinds of consensus algorithms, proof-based and voting-based. For proof-based consensus algorithms the network needs to solve cryptographic problems in order to get the right to append the block. Voting-based consensus algorithms vote for it and are preferably used in private blockchains where nodes are known [5].

1) Proof of Work (PoW): Proof of work is a proof-based algorithm and relies on nodes solving a mathematical puzzle. It is for example used in Bitcoin to protect against unwanted changes to the ledger blockchain. After hashing a value with SHA 256, the system will determine if the hash value satisfies the difficulty level of the system [5]. If it is not, the hash value is too weak and needs to be recalculated. The recalculations work by changing the nonce value. A nonce value is a value that doesn't have any meaning, but is added to the block, which should be hashed, in order to create a hash value that satisfies the difficult level set by the system. The hash value is then recalculated with a new added nonce value to try and prove its satisfiability by the system difficulty level. If it is not satisfied again, the nonce value will repeatedly be changed until it is met by the system. After being accepted by the system, the nodes verify their blocks hash value and nonce value by broadcasting it to the other nodes on the network, which append it to their own ledger. This makes the systems proof of work highly secure but in need of a lot of computational power [6].

2) Proof of Stake (PoS): Replacing the miners of mathematical puzzles above by choosing authorized validators based on the amount of currency they have in the system. It keeps this system secure by penalizing the users with malicious behavior by removing their currency while rewarding the users who act correctly, hoping potential losses over failed attacks will deter attackers. This system leads to less latency, better throughput and increased power efficiency [7].

But by lowering the cost of mining and effort you are more vulnerable to malicious attacks. This limitation of this algorithm is called the "Nothing-at-stake" problem. Another drawback of the algorithm is the possibility of centralization as its wealthiest node can increase its dominance based on this system.

3) Delegated Proof of Stake (DPoS): The Delegated Proof of Stake (DPoS) consensus algorithm applies 21 validators, also called Block Producers (BPs). The 21 various validators are selected from a set of Block Producers [1]. The validators take place in various nodes of the network. The validators act as a delegate to validate the transaction to another node by using a voting process. The delegates take turns to vote on newly created blocks and validate the blocks authenticity. It is also possible to vote out a dishonourable delegate if it shows malicious behavior, but this can also lead to corruption in the voting process if used incorrectly [7].

The difference between PoS and DPoS is that PoS pursue a direct democratic policy and DPoS pursue a representative democratic policy. Another difference is that in DPoS there are fewer participants involved in the process of block validation than there is in PoS. This alleviates for faster block generation and for faster transactions provable. For DPoS it is also possible to assure efficiency by adjusting the block size and block intervals.

The limitation of DPoS as a consensus algorithm is it can have centralization tendency. This means that a validator can by themselves vote for others to become validators by manipulating the votes. This can lead to potential malicious validators to join the voting process, but as described before, dishonest validators can be voted out by the other validators.

4) Proof of Authority (PoA): This algorithm relies on having chosen authorities within the network. While an authority, a user can sign off and validate a block. By then switching which authority is allowed to validate which block, it prevents malicious users who have gained authority from validating many blocks in a row [3].

C. What is a smart contract?

Smart contracts are the idea of embedding contractual clauses into software. Introduced in blockchain by Ethereum, it works by having immutable pieces of code reside within the block's data which then is triggered by transactions. Just like any other transaction in blockchain the output becomes validated by the nodes and results are saved into the blockchain. Since they are censorship-resistant like any blockchain data it is hard to stop it without altering most of the nodes in the network [3].

D. Private vs Public Distributed Ledgers

A public distributed ledger, also called non-permissioned distributed ledger, allows anybody to create and validate blocks but also modify the ledger state by storing and updating data through transactions with other entities of the network.

A private distributed ledger, also called permissioned distributed ledger, only allows authorized people to create or validate blocks or to modify or in any other way participate within the ledger. This means a private distributed ledger is restricting the accessibility of a ledger and therefore can ensure the privacy of the data in a ledger [1].

They are not always interchangeable though as there are concepts of public-permissioned, like EOS, and private-permissionless DLTs. Public DLTs meaning everyone being able to join and information being available for anyone and permissioned meaning having authorities within the network. Private DLTs meaning not sharing any data and permissionless meaning no authorities on the network.

III. Methodology

In this section the searching of relevant literature for the researched questions is presented.

A. Searching literature

To find relevant literature for this survey, we used the three databases Google Scholar, IEEE and LiU Library. In those databases different keywords were used to find the relevant papers. The keywords which were used can be seen in Table I. The keywords were also used in different combinations and used with the Boolean operator AND.

B. Literature screening

By prioritizing papers with a higher number of citations and quickly reading abstracts to check their relevance we thinned out the first round of keywords searched. This first round is called filter 1. In the next round of filtering, called filter 2, the papers from filter 1 was read at a deeper level. This included the reading of the introductions, results and conclusions to get a better view of the papers. If the paper was classified as usable for this survey, it bypassed filter 2.

C. Categorizing

After selecting and filtering which scientific papers were relevant for our process, they were categorized into each section, to see to which of our 3 questions they were most relevant to and to see for example which DLTs, characteristics or usage area they brought up. If it was relevant enough for multiple questions or DLTs it showed up multiple times within this categorization.

IV. Result

For this section, the findings from the literature are presented. First for smart contracts, followed by private vs public ledger and finally where the DLTs are best suited.

A. Smart contracts

1) DLT performance using smart contracts: In an experimental study made by Benahmed et al. [3] they measured four characteristics for three different DLTs. The measured parameters were CPU consumption, memory consumption, load scalability and network scalability. The three DLTs were EOS, Ethereum and Hyperledger Sawtooth.

For the CPU consumption, the experiment used five different nodes to send various amounts of transaction between. The result of the measurement showed the percentage usage of a core in a multi-core CPU (e.g. 120% is the full usage of one core and 20% of a second core). The average CPU consumption for sending 10, 100 and 1000 are shown in Table II.

The Memory consumption was also using five different nodes to send various amounts of transactions between. To measure the memory consumption the experiment saved and read data through the DLT. The node, which received and sent the data, was using its RAM to store the blocks and gives an idea of the memory consumption.

To measure the load scalability, the transactions were set to 1000 and a node range of 2-9 nodes. Then the experiment measured the speed for a transaction to be sent between two nodes. The scale is the average of the transactions executed per second (tx/s).

The network scalability measurement works almost as the load scalability. The differences are that the number of nodes is set to five and it is using a range of 50 to 1000 transactions.

2) Smart contracts impact on characteristics: How does smart contracts affect the characteristics of the ledgers? Does it slow it down, help with immutability, help scale it up, etc? Here we present the results about those findings.

1) **Privacy -** In general, Nikos Fotiou and George C. Polyzos describe that the information in a smart contract is available for everyone. They describe three implications of what this property leads to:
   a) A smart contract cannot store private data (e.g. private keys).
   b) A smart contract cannot execute operations that requires secret information (e.g. creating a digital signature).
   c) A smart contract cannot create or generate secret information (e.g. create a private key).

   In the example of creating or storing a private key, every person can see in the smart contract what the stored private key value is and how a private key is created by studying the algorithm [8].
   While for example Ethereum only storage of smart contracts is in bytecode, new programs that analyze the bytecode now have put the privacy of these contracts under risk [9].
2) **Immutability -** Nikos Fotiou and George C. Polyzos describe in general that smart contracts are immutable. When a smart contract is deployed, nobody can modify the smart contract. If there exist a flaw in the smart contract after deployment, the flaw stays in the smart contract forever due to there being no way to grant an update [8].
   Because of immutability, when you send cryptocurrency to a smart contract address there might be an error causing the receiver's private key not to be visible to the sender. This leads to your transaction being statistically impossible to recover because of the low probability of guessing the address. Trying to use computational power to do so would be impractical both in resources and in time. This is called *Cryptocurrency transfer loss* [9].

TABLE I
KEYWORDS USED FOR SEARCHING LITERATURE

Private DLT	Public DLT	Private vs public DLT
DLTs private public	DLT	Distributed ledger
Distributed ledger performance	Distributed ledger performance comparative	DLT tradeoffs
Smart contracts DLT	DLT smart contracts	Distributed ledger technology AND private AND public
DLT AND smart contracts AND tradeoffs	DLT AND smart contract AND comparison	Smart contracts AND distributed ledger technology

TABLE II
SMART CONTRACT EXPERIMENTAL RESULTS

DLT	CPU consumption (%)			Memory consumption (MB)			Load scalability (tx/s)	Network scalability (tx/s)
	For 10 transactions	For 100 transactions	For 1000 transactions	For 10 transactions	For 100 transactions	For 1000 transactions		
Ethereum [3]	110	120	125	230	460	460	10	4-11
EOS [3]	5	10	10	10	10	10	21	21
Sawtooth [3]	65	65	18	50	40	190	3-7	3-10

3) **Security -** If a participant in a smart contract breaks their conditions of the contract, they would be punished by losing their paid deposit and establishing timed commitments according to protocol, which is a way to ensure values are returned back to its previous owner. But this can be exploited by others looking to hurt you or simply by bad luck, as a timed DoS attack can make you miss fulfilling your conditions on time, making you suffer losses despite wanting to cooperate. This sort of problem is called *Unilateral abortion* [9].
4) **Upgradability -** Smart contracts upgradability differs from DLT to DLT. On Ethereum upgrading smart contracts is not possible while on EOS they are, even without forking [1].

3) Problems with smart contracts: In a systematic study from D. Macrinici et al. [9], they identified 16 problems for smart contracts within blockchain. The 16 problems are presented below.

1) **Consensus mechanism** - This problem relates to the different consensus algorithms creation of smart contracts. Regarding the PoW consensus algorithm, where a large amount of energy is consumed while mining, there exists risks for centralization and in the network, where the mining takes place, it tends to fall for attacks. For the PoS consensus algorithm, there is a decrease in security and a loss of transactions completeness. In PoS, some risk for centralization is reduced but there are still some centralization risks left (e.g. stakeholders control in stake pools).
2) **Sacrificed performance for scalability** - There are tradeoffs in performance to accomplish high scalability. There is for example heavy work to handle a large number of transactions in DLTs such as Bitcoin and Ethereum. Other heavy work in need of a decrease is the hardware and resource requirements which operate within a decentralized network.
3) **Unpredictable state** - In general, when sending a transaction which require a smart contract there is a doubt about the state of the contract while the transaction is running. D. Macrinici et al. bring up two different situations for this. The first is when a contract state is changed by other transactions, which could be executed first. The second situation is when a fork is needed to be executed. As a consequence of those two situations, an exploit was raised so that an attacker could trick honourable users to use their malicious library. This library is then updated into the smart contracts and makes an entry point for the attacker.
4) **Generating randomness** - This problem refers to an issue where the source of randomness should meet three criteria to generate a randomness. The three criteria are, available globally, verified independently and unpredictable. Two suggestions to solve the problem was to use the blocks hashes or timestamps but those two suggestions were vulnerable to other attacks.
5) **Timestamp dependencies** - When using the timestamp from a block as a condition to let central operations be executed, security problems arise within smart contracts.
6) **Lack of reimbursement** - This problem is the unfinished handling of preconditions. This problem encloses the situation when a party is going to lose in a elected abortion, the payment to the other party is also aborted. An example could be a poker tournament where a player needs to pay a fee (which is included in the prize for winning the tournament) to take part in the tournament and when the player loses, the player takes back the fee and leaves.
7) **Unilateral abortion** - This problem is already described in the section IV-A2 regarding security.
8) **Lack of privacy/preserving privacy** - The problem regards privacy concerns are already presented in the section IV-A2.
9) **Call to the unknown** - This problem is in a general term a flaw, making it possible to execute a DAO-attack, where an attacker can make participants donate ether (Ethereum cryptocurrency) to contracts of the attacker's preference. The attacker can then withdraw the funds with the help of a malicious drawback function. For fur-

ther explanation of the DAO-attack and the vulnerability, it is presented in section IV-B7.

10) **Exception disorder** - There exist two ways exceptions were treated. The first way was that the execution stopped and the side issues were degenerated combined with the gas consumption. The second way was that the exception was reproduced in the chain and the side issues reappeared in the contract again.
11) **Gasless send, out of gas exception** - Gas in a DLT context refers to the amount of ether when a transaction is made, which will pay the execution cost for said transaction. It is possible during the transaction to run out of gas and the sender needs to pay back to the miner to cross the limit. The problem is that there is a max limit of the amount of gas a user needs to perform a transaction, but the limit is not trustworthy as it doesn't always work. If the limit is exceeded, an external function is supposed to be called. This function could fail, making the limit not trustworthy as the consequences could be that the user possibly pays a higher price than expected for exceeding the gas limit.
12) **Type cast mismatch** - Type cast mismatch can emerge when a message is sent to external contracts which can include a typecast address in the smart contract interface and continue to call a function in the smart contract. The typecast which was included made the compiler unable to decide if it was a valid or invalid typecast.
13) **Re-entrancy** - The re-entrancy was the most severe issue in the DAO-attack, which is described in section IV-B7. The re-entrancy gives the control to the smart contract caller from the founder. With the control of the caller, the caller could withdraw money from another account.
14) **Immutable** - This is already described in the section IV-A2 regarding immutability.
15) **Stack overflow** - A stack overflow exception was possible to get when a smart contract was called by someone other than the creator or if the smart contract was self-created.
16) **Cryptocurrency transfer loss** - This is already described in the section IV-A2 regarding immutability.

B. Public vs Private Distributed ledger

Here we will use different characteristics to compare where different private and public distributed ledgers have their strengths and weaknesses and how it depends on their private or public design.

The outcome of an analysis of comparing different DLTs by Chowdhury et al. [1] can be seen in Table III. They are comparing different DLTs against different parameters. The comparison between different DLTs and the characteristics are described in the following sections.

1) Scalability: Scalability is the measurement for how much data a system can manage during a given period of time. Because of the lack of need for hard computationally intensive cryptographic puzzles, using Proof-of-Elapsed-Time, often used in private DLTs, increases scalability [1]. Another consensus algorithm, Proof of Work, often used in public DLTs, also ensures scalability, even though other characteristics such as throughput and latency suffer while doing so [7].

As then seen in for instance Table III, the quantitative evaluation from Chowdhury et al. shows that scalability on block size and block time is more configurable on private DLTs than on public ones [1].

According to the research of S. Benahmed et al., out of the compared DLTs, scalability for loading seems to be strongest with EOS at 21 transactions per second. Ethereum reaches a peak when approaching around 100 transactions and then stagnates at around 10 transactions per second. Sawtooth at the same peak as Ethereum starts rejecting transactions with a steady decline after 100 transactions. When looking at network scalability we can see a similar picture with EOS being stable at 21 transactions per second while Ethereum and Sawtooth starts lower and declines, with Sawtooth declining much faster [3].

2) Cost: Cost, also known as transaction fee, is the cost, if there is any, for each transaction which stores or manages data in the ledger [1].

Cost is something mostly used on public DLTs to execute transactions and smart contracts. The cost exists to prevent the infinite loop from happening in the code [1] [10] [11].

3) Privacy: Privacy is the ability to cover information of individual data. Because of everyone's ability to create and validate blocks all information stored in public DLTs are accessible for everyone. This makes public DLTs a bad choice when wanting privacy or handling sensitive information if it isn't handled well enough with for example encryption. [1].

4) Identity and Auditability: Identity is the qualities that makes a person/thing unique from others [1]. Auditability is the ability of an auditor to accomplish proper results in a review of a record [12]. In the Table III the identity is how an entity is identified with the respect of auditability.

5) Robustness and Resilience: Robustness is the quality or how unlikely a system fails due to errors. Resilience is the ability to manage an adversity and return successfully back to the condition before the adversity [1]. In Table III robustness and resilience is referred to as the ability to manage against different types of attacks and uncommon errors.

6) Flexibility: Ethereum proves more flexible in storing data than bitcoin as it has more data types that can be stored in any smart contract [1].

Using Proof-of-Elapsed-Time increases flexibility as it allows Sawtooth to be used in both private and public ledgers.

TABLE III
CHARACTERISTICS RESULTS

DLT	Scalability		Cost	Power consumption	Privacy	Identity and Auditability	Robustness and Resilience
	Block size (MB)	Block time (s)					
Ethereum [1]	Implicit restriction	15	Yes	High	Data as smart contract and transactions are visible for everyone	Has strong support for auditability and accountability when an entity can be verified properly	Strong resiliency against data and code immutability because of its consensus algorithm
EOS [1]	1, dynamically configurable	0.5	Yes	N/A	Data as smart contract and transactions are visible for everyone	Has strong support for auditability and accountability when an entity can be verified properly	Has a robust P2P structure while its resiliency is less than that of Ethereum. This because EOS is only using 21 validators, making it easier for corruptness and collusion among the validators to occur. Can provide resiliency against data and code immutability if the validators perform as intended.
Sawtooth [1]	Configurable	N/A	No	Very low	Support privacy by using a private ledger where individuals who are known can engage	Has strong support for auditability and accountability because of the usage of public key cryptography to verify an identity	Have a robust P2P network and the resiliency is depending on the number of validators.
Fabric [1]	Configurable	0.5-2	No	Very low	Support privacy by using a private ledger there only authorized persons can engage. To maintain privacy within a network of peers, Fabric applies a concept called channel which allows this.	Has strong support for auditability and accountability because of the usage of PKI-based identification which requires an identity to be registered and issued with required keys	Have a robust P2P network and the resiliency is depending on the number of endorsers and the number of orders.

7) Security: The public DLTs such as Ethereum and EOS have their data and transactions open in the ledger for everyone and are therefore not suitable for managing sensitive data. To identify entities, public DLTs uses cryptographic pseudonyms, which harden the possibility to audit and is open for Sybil attacks [1].

Sybil attack is when an entity, which is corrupted, can introduce several nodes by itself to take control over the systems and sabotage redundancy. Redundancy was first introduced to large-scale peer-to-peer networks to resist security threats which malicious entities can contribute to. Redundancy is then bypassed by Sybil attacks [13].

Back to Ethereum, its lack of upgradability for smart contracts, presented in section IV-A2, made it vulnerable to attacks such as DAO. Ethereum is vulnerable to this attack because of its use of the cryptocurrency named Ether. Ether uses smart contracts but can't update them if new features are added or if a bug is found in the current version of the smart contract. For this attack, a bug in the currently used version of the smart contract makes it possible for attackers to retrieve a large amount of money [1].

DAO-attack was an attack on a system named Decentralized Autonomous Organization (DAO) which was created on Ethereum. DAO uses Ethereum's cryptocurrency Ether to send transactions with the help of smart contracts. The DAO system works by investors exchanging Ether for tokens to vote on an approval or rejection of a project. The vulnerability in the DAO system was a flaw in the smart contract called *Call to the unknown*. The flaw was used by an attacker to withdraw their balance, which was stored in DAO, repeatedly before the balance was adjusted [14].

In comparison to the public DLTs we have the private DLTs Hyperledger Fabric and Sawtooth, which were created to solve the different problems with public DLTs. In a private DLT the identity of an entity must be verified in the network which leads to accountability and auditability increasing. The verification of entities will mitigate the Sybil attack. However, there is another aspect to be considered about private DLTs. Private

DLTs cannot grant the same amount of security regarding immutability of data and code compared to a public DLT. This depends upon if the integrity should be compromised in any DLT, it requires to corrupt or collude the majority of the validating nodes, also called the 51% attack. It is a difficult task for an attacker of a public DLT to control all those validating nodes, which contains thousands of miners or stakeholders. Compared to private DLTs, where there is only a few validating nodes, countable on one hand, making these attacks potentially easier [1].

In general, while a private DLT higher the confidentiality it lowers the integrity of the ledger. This because having a small number of known nodes exposes the network topology, increasing the risk of a successful partition-based attack on the network and therefore increases the risk of losing immutability [15].

8) Immutability: Public DLTs ensure a much stronger immutability of data than private DLTs. This because of the difference in the number of validators/miners in the chain which often are higher in public than private DLTs [1]. They help reduce the concentration of power and therefore the power of data manipulation by those with authority that is otherwise used in private DLTs [7].

9) Maintainability: To update DLT software and protocols there needs to be a majority of nodes for the proposal. Using a public-permissionless DLT makes that difficult based on the potential lack of verification by nodes and the normally large number of nodes within the ledger. That fact makes it hard to maintain and update a large public-permissionless DLT. Public-permissioned DLTs on the other hand has more control over which nodes to let in making it easier to maintain, while private-permissioned DLTs are the best at maintainability since their nodes are also needed to be verified. This meaning you can contact them directly when trying to push for an update [15].

10) Centralization: Decentralization helps reduce the need for trust of a central agency while reducing server, operation and development costs. But as decentralization requires more trust in general, consensus algorithms are what is needed to reduce this need. Bottlenecks are also reduced because of this lack of central agency [7].

Proof of Stakes system of selecting and rewarding the wealthiest nodes risks getting a system centralized or getting unfair distribution. When talking about blockchain, public DLTs are more decentralized than private DLTs, who still need some sort of centralized authority to work.

11) CPU usage: In this area, out of the DLTs compared, EOS seems to be the winner regarding low consumption. On the other hand, it has shown to have slower processing during CPU-intensive loads. This may prove EOS to be useful in projects using public DLTs and needing consistent speed. Sawtooth instead reduces pressure when reaching its configured bottleneck and starting to reject transactions. This means it's CPU usage decreases when load increase, showing Sawtooth is good when needing limits for hardware such as when using DLTs together with IoT systems. Ethereum showed big struggles compared to this as it consumed 110 % of the CPU on average [3]. Specific numbers are available at Table III.

12) Memory usage: Here again EOS shows a great advantage having memory consumption being less than a quarter of Ethereum's. Sawtooth starts not quite as low as EOS but still much lower than Ethereum. When it reaches its threshold, it again starts rejecting transactions which leads to less memory consumption [3]. Specific numbers are available at Table III.

13) Energy consumption: A. A. Monrat, O. Schélen and K. Andersson present the energy consumption regarding Bitcoin which uses the proof-of-work (PoW) consensus algorithm. While the transactions are being sent between each peer in the network, the miners are consuming a large amount of electricity. The rising popularity of Bitcoin has made the Bitcoin network consume more electricity than some countries according to a report by the International Energy Agency. From their report back from 2017, Bitcoin had an energy consumption of 47 TWh that year, placing them at position 53 in the world. Countries like Peru, Hong Kong and Iraq had lower energy consumption that year. They was placed in the range of 40-43 TWh per year [7]. During the year of 2021, Bitcoin is estimated to consume 109 TWh, placing them at position 34 in the world. Now they are ahead of countries like Kazakhstan, Philippines and Belgium which have an energy consumption less than 100 TWh per year [16]. As Bitcoin causes this big amount of energy consumption, it contributes to making a bigger impact on the carbon footprint. As an example, in China the coal-fired power plants provide the extra electricity used by Bitcoin.

A. A. Monrat, O. Schélen and K. Andersson also suggest two possible solutions for the high energy consumption that Bitcoin constitutes. The first suggestion is to change the design of the infrastructure of blockchain. The second suggestion is to change the consensus algorithm to proof-of-stake (PoS), which consumes less energy because of that the selected miners don't have competition when it comes to verifying blocks [7].

C. Best suited usage for different distributed ledgers

Here we examine how different strengths and weaknesses of DLTs can be used in certain areas.

1) Internet of Things: The number of Internet of Things (IoT) ecosystems has been quickly growing and was according to Business insider expected to have over 24 billion devices installed by 2020. This means many machines needing to connect and communicate. Blockchain helps reduce time and money needed in this process by allowing for decentralization. It's reducing the need for authority and together with smart

contracts it helps enable automated transactions between machines. Obstacles for blockchain working smoothly with IoT include energy consumption, hardware cost, transaction fees and scalability. Scalability issues emerge as IoT ecosystems require a large amount of transactions between many nodes which leads to delays if the system uses blockchain technology. This leads to blockchain and therefore, all of the DLTs examined in this paper having some potential issues being deployed in IoT ecosystems [17].

2) Logistics: Expensive bureaucratic delays, tampering of data and faulty tracking and tracing are common issues in the logistic market. They show up when trying to satisfy requirements from government and citizens and when trying to keep track of documents, contracts and other information for every object whenever it switches carriers. Decentralizing and creating immutable ledgers, as for example blockchain does, could help to solve these issues and would improve security, speed, trust and transparency in supply chains [18].

The complexity of logistics and its diversity in industry standards would make it hard to implement blockchain on its own. But thanks to new inventions, including smart contracts, it has granted freedom and versatility to the creation of applications. Therefore, using a DLT supporting smart contracts is important. In *An Application of Ethereum smart contracts and IoT to logistics* they propose using a blockchain based Ethereum smart contract application.

3) Governing: Reducing corruption and making it easier for accountability are a few of the benefits that can be achieved based of the transparency, auditability and easier access you could get from using DLTs in government. Using smart contracts can reduce costs, administration and human errors while the security of a specific DLT could potentially higher a governments IT-security, preventing spam and DDOS attacks [19].

While increasing security in some sense it can also lower it in others. DLTs using Proof of work (PoW) are for example susceptible to 51 % attacks which, if successful, could take full control over what blocks get added and maybe even rewrite some old blocks and their history.

4) Smart cities, sharing economy and social compliance: Using DLTs as digital tokens you can create a digital deposit system to incentives keeping social contracts, allowing for systems more reliant on other people in society, such as the sharing economy, to be viable. For this to work you need a DLT without transaction fees, which would scare away people fond of their digital tokens, and with high throughput, since slow processing of transactions would then slow down society and scare away people from using the systems at all. Another important factor is privacy, as allowing for this information about breaking social contracts to get out to the public would have consequences. For privacy reasons it's also important that there will be no record of these things for central entities either. Instead, this system will notify authorities when the persons digital tokens have run out, meaning they will no longer be allowed to use the benefits of these social contracts. For example, if you run past 20 red traffic lights you might no longer be allowed to drive [20].

5) Voting system: Even though electronic voting (e-voting) has been used before it is not widespread and still has issues with security as data can be tampered with and traces can be eliminated by authorities in control of the database. DLTs, and blockchain specifically, can reduce the risk of this problem by making the database public and distributed. Blockchains, together with the consensus algorithm Proof of Work, have good availability, verifiability and integrity. This leads to the tampering of data being extremely hard. To then follow the rules of elections it is needed to have a permissioned blockchain, to confirm identity and the amount of votes cast at a certain node [21].

These characteristics together with the rest of the design choices of the report *Blockchain based e-voting recording system design* seemed to prove quite scalable. Implementing the system with Python and doing a non-functional test with the same amount of nodes as election places in Indonesia, the report found that it worked and that each node took an average of 0,24 seconds to create a block and each block had an average of 216,04 bytes required space.

6) Copyright: DLTs can be used to transparently display and track ownership, use and compensation for different kinds of work. Smart contracts can be used for licensing and using permissionless DLTs can help decentralize the process, making creators more autonomous and the licensing process more efficient [22].

7) Peer-to-Peer transactive energy exchanges in local energy markets: With rising levels of electricity coming from not only renewable resources, such as wind and solar, but also distributed energy resources of renewable energy, the energy market has started to fluctuate more than before. To handle peaks of energy demand there's a need to balance this. As electric loads now can auto-regulate their power absorption based on demand responses from the grid while the number of Internet of Things devices and prosumers, users who can both consume and provide the electricity grid, have started to grow this has opened up for users to regulate the supply and demand more. This can be exemplified in users being able to charge their electric vehicles and heat their homes when there's an electrical surplus of energy in the grid. This can be based on value, defined by prices guaranteeing the benefits of all users [23].

If this system were to be centralized it would not be economically feasible as the high volumes of transactions needed to pass through the centralized servers would make the systems hard to scale up. Centralization also means servers creating single points of failure exposing the service to attacks against the system. Therefore, decentralized DLTs are of huge help. Other aspects important for the infrastructure and therefore the

DLT used are high throughput, failure resistance, data integrity, low energy consumption and security and privacy to protect user information.

For picking which DLT to use for this, those who use Proof of work, Bitcoin and Ethereum for example, seem to have an upper hand on data security as its hard cryptographical puzzle makes the system sealed against malicious trading. But as its energy consumption is too high the better option suggested is Proof of Stake. While it also has shortcomings that other cryptocurrency solves, PoS is a good candidate and can avoid its shortcomings by using permissioned architecture with hard-to-forge stake values. In *A Survey and Evaluation of the Potentials of Distributed Ledger Technology for Peer-to-Peer Transactive Energy Exchanges in Local Energy Markets* they for example find an architecture based on PoS that uses 0,001 % of the energy that PoW does.

V. Discussion and conclusion

In this section, RQ1, RQ2 and RQ3 are discussed based on the result. The research question is discussed in the same order as they are numbered.

A. Is it worth it to have smart contracts?

Using smart contracts in a DLT enables several possibilities. By using a smart contract, it opens up to create decentralized applications. Those applications run autonomously and don't have to rely on a system entity. Smart contracts inclusion in decentralized applications, makes smart contracts a part of the ledger. This makes the execution of the smart contract and the smart contract immutable.

Regarding the privacy concerns, smart contracts are open for everyone. This implies that secret information cannot be generated from smart contracts, execution of operations requiring secret information cannot be done in smart contract and storing secret information cannot be done in smart contract. Developers developing smart contracts should be careful when deciding what data should be stored in the smart contract.

However, a drawback with smart contracts is the upgradability characteristics. This drawback exists due to smart contracts being immutable. The public DLTs Bitcoin and Ethereum both lack the feature of upgradability, while the private DLTs, Hyperledger Fabric and Sawtooth, both support it. There is a public DLT supporting this upgradability, and that is EOS. Despite that there are DLTs not being able to update the smart contract their funds can still be stolen. This due to new features being added or bugs existing in the smart contract, making the smart contract vulnerable for attacks. An example is the DAO-attack against Ethereum. A way to handle the lack of upgradability for public DLTs is to implement a kill switch to the smart contract. If there is a bug or a new feature in the smart contract, making the smart contract vulnerable for attacks, the kill switch is a piece of code that transfers back the funds to the contracts owner and prevents other users from interacting with that contract [8].

To summarize, a smart contract makes transactions between two nodes immutable. The immutability comes with some drawbacks. Some DLTs don't support upgrading the smart contract due to the immutability which can cause the consequences that funds are lost because of different attacks.

B. Should you use a private or public DLT?

1) Bigger systems: To avoid slowing down the system when scaling them up the optimal choice would be to use private DLTs. This due to their more configurable block size and block time combined with its easier maintainability and consensus algorithms needing less computational power.

2) Smaller systems: Here a public DLT is more resilient against adversity than a private one as private DLTs heavily rely on being able to identify their nodes for accountability instead of having protective measures installed such as the consensus algorithm Proof of Work. Therefore, the number of validators are very important for private DLTs in order to avoid attacks.

3) Energy consumption: Due to the big amount of energy Bitcoin consumes, it can be seen as a big problem. Bitcoin is estimated to consume 109 TWh in 2021. In comparison to Ethereum, which energy consumption taken from the Ethereum Energy Consumption Index is estimated to be 41 TWh in 2021 [24], which is far less than Bitcoin. This can also verify A. A. Monrat, O. Schélen and K. Andersson's second suggestion about changing consensus algorithm. To change the consensus algorithms, in such popular and widely used DLTs as Bitcoin, will be an enormous work and probably very hard.

For private DLTs, such as Hyperledger Fabric and Sawtooth, the energy consumption is lower and would be seen as a better option than the public DLTs Bitcoin and Ethereum.

4) Vulnerabilities - what DLT protects against what?: Private DLTs have their strong sides with it's accountability and audibility which can protect against malicious behavior such as the sybil attack because of its possible authority. It also has better privacy, protecting your private information against snooping eyes.

Public DLTs on the other hand has it's strong sides on being able to protect against corruption due to its openness with information and being able to protect against data tampering due to the computational heavy workload needed for changing the ledger.

5) Centralization: In general, public DLTs are more decentralized than private DLTs. This depends on whether a private DLT needs some centralized authority to authenticate users if it's a permissioned or permissionless DLT. A public DLT is available for everyone to use. A drawback with having the DLT available for everyone could be that the data would be exposed unless you for example have the data under strong encryption. If the data for example is not encrypted, any sensitive information cannot be created or stored in a public DLT without regarding the privacy concern. Private DLTs are in that case a better choice where only authenticated users can take part of sensitive information. The strength with having data available for everyone in public DLTs is that it makes it hard to change the data. First of all, since the data is open, anyone can see the manipulation of the data. Second, because

of the different consensus algorithms used in public DLTs, a malicious validator needs to bypass the majority of validators to be able to pass a change in the data. This strength makes the data believable and correct.

6) Centralization - Proof of Work vs Proof of Stake: While both being among the best in this area, of the ones we found, we found different sources saying different things regarding which one of these were weakest against centralization. Daniel Macrinici et al. brings up that there are risks for centralization with Proof of Work which contradicts what we've read on other sources, but they did not explain it in which way so we can't compare. But as their paper was regarding smart contracts, centralization might be affected differently by Proof of Work depending on if smart contracts are used or not. Therefore, we cannot conclude anything and recommend further research being done.

7) Specific strengths and weaknesses of the compared DLTs: Ethereum as the sole public permissionless DLT here stands on its strengths of immutability and decentralization while lacking in maintainability and performance.

EOS has strengths clearly visible in performance in its constant low usage of computational power. At the same time, its low number of validators opens up for corruption and collusion if they are not verified properly or won't performs as intended.

Hyperledger Sawtooth has the possibility to configure bottlenecks to reduce its transactions, leading it to be very adaptable and could help with systems with lower computational power such as those in IoT systems. The consequence of this bottleneck is that the needed memory consumption quickly increases when reaching the threshold. Otherwise, it also holds strong security thanks to its use of Intel SGX.

Hyperledger Fabric has the strength of being pluggable, making you able to switch consensus algorithm depending on your needs. It also has strong security and privacy thanks to its architecture, making it possible to create different policies for different users and separating each user's transactions independently.

C. Where can you use the DLTs?

1) Internet of Things: As all of our researched DLTs were blockchains none of them would be suitable for Internet of Things. Instead of blockchain, Tangle which is based on Directed Acyclic Graph (DAG) and uses the cryptocurrency IOTA, seems to be a good choice for IoT ecosystems. Having achieved infinite scaling, offline transactions and zero-cost transactions it shows promise for having aspects relating to the needs of them.

Scalability works for Tangle, thanks to DAG, in parallel compared to blockchain which uses appending of transactions. Tangle also has better security as it doesn't rely on cryptography the same way blockchain does. As Tangle uses Quantum-resistant cryptography it is also more future proof than those DLTs using blockchain [17], [23].

2) Logistics: Demanding the use of smart contracts, logistics rule out the use of Bitcoin. The usage and importance of decentralization in this business also makes it important to use blockchain to ensure a system people can trust. Just automating and ensuring integrity could solve many of their problems, which blockchain happens to have.

Since we didn't find any information about certain consensus algorithms tested for this business, we don't know how much throughput is possible or needed. Therefore, it might be better to use Proof of Work to ensure better decentralization, but only if the amount of transactions is small enough to be sustainable.

3) Governing: Government is an important part of society and therefore has security high on its priority list. Therefore, avoiding Proof of Work should be of priority as you do not want another government with more computational power to take over your systems. Using public DLTs which have public recording of information can be of importance when fighting corruption.

4) Smart cities, sharing economy and social compliance: To incentives citizens to use a system with DLTs it can't use DLTs with fees, ruling out ledgers such as Ethereum. To avoid being used for example governmental surveillance and social judgement it has a need for using a private DLT preserving privacy.

5) Voting system: Using a permissioned blockchain to help confirm identities and number of votes cast when voting, while also making the database public, allows for decentralizing, and therefore reduces the risk of database manipulation during e-voting.

6) Peer-to-Peer transactive energy exchanges in local energy markets: To balance a fluctuating energy market there is a need for an automated, decentralized and secure infrastructure with possibilities for high throughput. Using blockchain helps with decentralization and while Proof of Work also helps with that. Proof of Work has an upper hand on security but it consumes way too much energy and computational power to be suitable. Instead using a permissioned Proof of Stake DLT with hard-to-forge stake values can help get a good system, with less energy consumption but then also lower security.

REFERENCES

[1] M. J. M. Chowdhury, M. S. Ferdous, K. Biswas, N. Chowdhury, A. S. M. Kayes, M. Alazab, and P. Watters, "A comparative analysis of distributed ledger technology platforms," *IEEE Access*, vol. 7, pp. 167 930–167 943, 2019.

[2] Statista, "Number of blockchain wallet users worldwide from november 2011 to march 28, 2021," Available at https://www.statista.com/statistics/647374/worldwide-blockchain-wallet-users/ (2021/04/09).

[3] S. Benahmed, I. Pidikseev, R. Hussain, J. Lee, S. M. A. Kazmi, A. Oracevic, and F. Hussain, "A comparative analysis of distributed ledger technologies for smart contract development," in *2019 IEEE 30th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, 2019, pp. 1–6.

[4] eos.io, Available at https://eos.io/ (2021/04/18).

[5] S. Pahlajani, A. Kshirsagar, and V. Pachghare, "Survey on private blockchain consensus algorithms," in *2019 1st International Conference on Innovations in Information and Communication Technology (ICIICT)*, 2019, pp. 1–6.

[6] I. G. A. K. Gemeliarana and R. F. Sari, "Evaluation of proof of work (pow) blockchains security network on selfish mining," in *2018 International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*, 2018, pp. 126–130.

[7] A. A. Monrat, O. Schelén, and K. Andersson, "A survey of blockchain from the perspectives of applications, challenges, and opportunities," *IEEE Access*, vol. 7, pp. 117 134–117 151, 2019.

[8] N. Fotiou and G. C. Polyzos, "Smart contracts for the internet of things: Opportunities and challenges," in *2018 European Conference on Networks and Communications (EuCNC)*, 2018, pp. 256–260.

[9] D. Macrinici, C. Cartofeanu, and S. Gao, "Smart contract applications within blockchain technology: A systematic mapping study," *Telematics and Informatics*, vol. 35, no. 8, pp. 2337–2354, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0736585318308013

[10] R. A. Canessane, N. Srinivasan, A. Beuria, A. Singh, and B. M. Kumar, "Decentralised applications using ethereum blockchain," in *2019 Fifth International Conference on Science Technology Engineering and Mathematics (ICONSTEM)*, vol. 1, 2019, pp. 75–79.

[11] S. Bouraga, "An evaluation of gas consumption prediction on ethereum based on transaction history summarization," in *2020 2nd Conference on Blockchain Research Applications for Innovative Networks and Services (BRAINS)*, 2020, pp. 49–50.

[12] D. Liberto, "Auditability," Available at https://www.investopedia.com/terms/a/auditability.asp (2021/04/26), 2021-04-06.

[13] J. R. Douceur, "The sybil attack," in *Revised Papers from the First International Workshop on Peer-to-Peer Systems*, ser. IPTPS '01. Berlin, Heidelberg: Springer-Verlag, 2002, p. 251–260.

[14] C. Shier, I. Mehar, A. Giambattista, E. Gong, G. Fletcher, R. Sanayhie, M. Laskowski, and H. Kim, "Understanding a revolutionary and flawed grand experiment in blockchain: The dao attack," *SSRN Electronic Journal*, 01 2017.

[15] N. Kannengießer, S. Lins, T. Dehling, and A. Sunyaev, "Trade-offs between distributed ledger technology characteristics," *ACM Comput. Surv.*, vol. 53, no. 2, May 2020. [Online]. Available: https://doi.org/10.1145/3379463

[16] Digiconmist, "Bitcoin energy consumption index," Available at https://digiconomist.net/bitcoin-energy-consumption (2021/05/02), 2021.

[17] A. Ahi and A. V. Singh, "Role of distributed ledger technology (dlt) to enhance resiliency in internet of things (iot) ecosystem," in *2019 Amity International Conference on Artificial Intelligence (AICAI)*, 2019, pp. 782–786.

[18] L. Augusto, R. Costa, J. Ferreira, and R. Jardim-Gonçalves, "An application of ethereum smart contracts and iot to logistics," in *2019 International Young Engineers Forum (YEF-ECE)*, 2019, pp. 1–7.

[19] S. Ølnes, J. Ubacht, and M. Janssen, "Blockchain in government: Benefits and implications of distributed ledger technology for information sharing," *Government Information Quarterly*, vol. 34, no. 3, pp. 355–364, 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0740624X17303155

[20] P. Ferraro, C. King, and R. Shorten, "Distributed ledger technology for smart cities, the sharing economy, and social compliance," *IEEE Access*, vol. 6, pp. 62 728–62 746, 2018.

[21] R. Hanifatunnisa and B. Rahardjo, "Blockchain based e-voting recording system design," in *2017 11th International Conference on Telecommunication Systems Services and Applications (TSSA)*, 2017, pp. 1–6.

[22] B. Bodó, D. Gervais, and J. P. Quintais, "Blockchain and smart contracts: the missing link in copyright licensing?" *International Journal of Law and Information Technology*, vol. 26, no. 4, pp. 311–336, 09 2018. [Online]. Available: https://doi.org/10.1093/ijlit/eay014

[23] P. Siano, G. De Marco, A. Rolán, and V. Loia, "A survey and evaluation of the potentials of distributed ledger technology for peer-to-peer transactive energy exchanges in local energy markets," *IEEE Systems Journal*, vol. 13, no. 3, pp. 3454–3466, 2019.

[24] Digiconmist, "Ethereum energy consumption index(beta)," Available at https://digiconomist.net/ethereum-energy-consumption (2021/05/02), 2021.