International Engineering Journal For Research & Development

Vol.4
Issue 5

E-ISSN NO:-2349-0721

Impact factor : 6.03

DECENTRALAND – A BLOCKCHAIN BASED MODEL FOR SMART PROPERTY EXPERIENCE

Prof. A. A. Chaudhari
Department of Computer Science
Prof. Ram Meghe Institute Of Tech. & Research, *Amravati, India*

Disha Laddha
Department of Computer Science
Prof. Ram Meghe Institute of Tech. & Research, *Amravati, India*

Madhulika Potdar
Department of Computer Science
Prof. Ram Meghe Institute Of Tech. & Research, *Amravati, India*

Abstract-

To allows crowd to buy or sell VR experiences in which you could be rocking out at a live concert, meeting up with a celebrity, learning about the pyramids from an ancient Egyptian or blasting through space while fighting off aliens. Interestingly, this platform is also planning a overview property experience with other features like dating, treat a get together meet, so we guess this will work like Tinder, with the added bonus of treating your date to some VR experience. Basically Block chain's ability to track products can improve crisis handling. Here the decision for the investment for maximum profit on available owned lands property. By purchasing LAND through the block chain, an immutable record of ownership is created, while smart contracts track all modifications. Once you own LAND, then it's yours to do with as you choose build houses and businesses, hang out with friends, listen to music, race cars or even go swimming with dolphins.

The best possible solution will be provided the interactive approach to deal with the owned land in context of profitable view. The aim is to used augmented reality based algorithm with block chain technology in utilizing property/land in its optimized way can be easily predicted.

Keywords— Augmented & Virtual Reality, Block chain, Property, Land

1. INTRODUCTION

Blockchain have allowed mutually mistrusting entities to perform financial payments without relying on a central trusted third party while offering a transparent and integrity protected data storage. Due to these properties, blockchain as a technology has gained much attention beyond the purpose of financial transactions – distributed cloud storage, smart property, Internet of Things, supply chain management, healthcare, ownership and royalty distribution, and decentralized autonomous organizations just to name a few.

Contrary to Bitcoin's permission less blockchain, where any writer and reader can join at any time, so-called permissioned blockchains have been proposed, where only an authorized set of entities is allowed to write and read the respective blockchain. A permissioned blockchain, however, shares similarities with a centralized database, and this naturally brings up the question whether a blockchain is better suited than a centralized database. In this work, we analyze the properties of different blockchain types (i.e. permissioned and permissionless) and contrast these properties to those of a centrally managed database. We provide a methodology to identify whether a blockchain is useful depending on the problem requirements, and if so, what type of blockchain might be appropriate. Based on our methodology, we evaluate in detail three use cases, namely (i) supply chain management, (ii) interbank and international payments and (iii) decentralized autonomous organizations and argue if and which blockchain type make sense for the specific applications. The remainder of this article is organized as follows. In Section II, we briefly describe the most important background about blockchain. In Section III we provide a structured methodology to identify if a blockchain makes sense, and if yes, which type of blockchain would be appropriate. Based on our methodology, we analyze proposed use cases in detail in Section IV. In Section V, we review related work in the area, and we conclude the article in Section VI.

II. BACKGROUND ON BLOCKCHAIN

In the following section, we detail the required blockchain background and the involved parties. The name blockchain stems from its technical structure — a chain of blocks. Each block is linked to the previous block with a cryptographic hash. A block is a data structure which allows to store a list of transactions. Transactions are created and exchanged by peers of the blockchain network and modify the state of the

blockchain. As such, transactions can exchange monetary amounts, but are not restricted to financial transactions only and for example allow to execute arbitrary code within so called smart contracts.

Before diving into the specific differences of permissionless and permissioned blockchains, we now describe the different participants of these networks. As applicable to any database system, we denote as writer any entity which writes state to the database. In a blockchain this would correspond to a participant that is involved in the consensus protocol and helps growing the blockchain. As such, a writer is able to accumulate transactions within a block and append this block to the blockchain. Related work might also denominate a writer as a validator. We denote a reader as any entity which is not extending the blockchain, but participating in either the transaction creation process, simply reading and analysing or auditing the blockchain.

Permissionless Blockchains Bitcoin and Ethereum are instances of permissionless blockchains, which are open and decentralized. Any peer can join and leave the network as reader and writer at any time. Interestingly, there is no central entity which manages the membership, or which could ban illegitimate readers or writers. This openness implies that the written content is readable byany peer. With the use of cryptographic primitives however, it is technically feasible to design a permissionless blockchain which hides privacy relevant information (e.g. Zerocash). Permissioned Blockchains To only authorize a limited set of readers and writers, so called-permissioned blockchains have been recently proposed. Here, a central entity decides and attributes the right to individual peers to participate in the write or read operations of the blockchain. To provide encapsulation and privacy, reader and writer could also run in separated parallel blockchains that are interconnected. The most widely known instance of permissioned blockchains are Hyperledger Fabric and R3 Corda.

A. Properties

In the following, we describe and compare the most relevant properties that distributed ledgers and centralized systems provide.

Public Verifiability allows anyone to verify the correctness of the state of the system. In a distributed ledger, each state transition is confirmed by verifiers (e.g. miners in Bitcoin), which can be a restricted set of participants. Any observer, however, can verify that the state of the ledger was changed according to the protocol and all observers will eventually have the same view of the ledger, at least up to a certain length. In a centralized system, different observers may have entirely different views of the state. As such, they might not be able to verify that all state transitions were executed correctly. Instead, observers need to trust the central entity to provide them with the correct state.

Transparency of the data and the process of updating the state is a requirement for public verifiability. The amount of information that is transparent to an observer, however, can differ, and not every participant needs to have access to every piece of information. Privacy is an important property of any system. There exists an inherent tension between privacy and transparency. Privacy is certainly easier to achieve in a centralized system because transparency and public verifiability are not required for the functioning of the system.

Integrity of information ensures that information is protected from unauthorized modifications, i.e. that retrieved data is correct. The integrity of information is closely linked to public verifiability. If a system provides public verifiability, anyone can verify the integrity of the data; integrity can otherwise only be ensured if the centralized system is not compromised.

Redundancy of data is important for many use cases. In blockchain systems, redundancy is inherently provided through replication across the writers. In centralized systems, redundancy is generally achieved through replication on different physical servers and through backups. Trust Anchor defines who represents the highest authority of a given system that has the authority to grant and revoke read and write access to a system.

B. Tensions between Transparency and Privacy

There exist an inherent tradeoff between transparency and privacy. A fully transparent system allows anyone to see any piece of information, i.e. no privacy is provided. Likewise, a fully private system provides no transparency. However, a system can still provide significant privacy-guarantees while making the process of state transitions transparent, e.g. a distributed ledger can provide public verifiability of its overall state without leaking information about the state of each individual participant. Privacy in a public system can be achieved using cryptographic techniques but typically comes at the cost of lower efficiency. The cryptocurrency Zerocash for example makes use of computationally expensive cryptography to provide full anonymity while still providing sufficient transparency to publicly verify the ledger state.

For any kind of a high value property (real estate, cars, art) it is important to have accurate records which identify the current owner and provide a proof that he is indeed the owner. These records can be used to protect

owners' rights (e.g. in case of theft) resolve disputes make sure ownership is correctly transferred to a new owner after sale prevent sale fraud Thus it is crucial to maintain correctness and completeness of this information, and prevent unauthorized, fraudulent changes.

From the point of view of a computer security expert, currently people have to rely on a trusted third party. E.g. a government agency might be responsible for keeping track of ownership information. Sometimes, these records are not preserved in a systematic way. Is it possible to keep track of property ownership through some kind of a distributed system which won't rely on trust? What would it require? At minimum, we need a consensus about the current owner and ability for that owner to identify himself.

The same problem was solved by Satoshi Nakamoto when he created: consensus is established using the blockchain (which keeps records of previous transactions) and proof of work which makes changing historic records prohibitively costly correctness is guaranteed by protocol rules owner can be identified using public key cryptography.

3. PROPERTY REGISTRY AND CATALOG

When property transfers are secured by the blockchain, we no longer need to rely on a trusted party to verify them. However, an associated between a particular property and a genesis transaction output becomes the weakest link. E.g. suppose somebody claims that a certain coin represents ownership of the house. He can demonstrate that he is the owner of an unspent coin by signing a message using his private key, and he can demonstrate the transaction history involving that coin. But how can we check that a particular coin represents a particular house? How do we check that there are no other coins which represent it?

In the example with the car, the factory which manufactured the car was responsible for associating a colored coin with a car. A tag or a chip attached to a physical object might be used to refer to a genesis output, and thus establish an association. But this is reliable only as long as information contained in that tag or chip cannot be altered, and the cannot be detaching. (Or, rather, detaching them is impractical or prohibitively expensive.)

But we can't rely on tags in the case with a real estate, for example, thus we need some kind of a registry which will be responsible for association between objects and corresponding colored coins. Let's assume that for a kind of objects we are interested in, we can generate property identifiers which unambiguously point to an object (e.g. coordinates, street address, device identifier etc.). Then a registry will map genesis transaction outputs to property identifiers, and property identifiers to genesis transaction outputs. Is it possible to make this registry distributed and trustless? It might work for some problem domains, e.g. in Namecoin, the first person who tries to register a name gets it. But this doesn't work in a general case.

Thus a registry needs to be a trusted third party. We can't completely escape from that model, however, we can try to minimize reliance on trust and impose rules which would make cheating hard, evident and provable.

Particularly, trust is much less of a concern when the registry is forced to operate in a transparent way and cryptographic protocols are used to authenticate information supplied by the registry. This can be accomplished by making registry's complete catalog openly accessible to everyone. I.e. anyone can request a complete catalog from registry, which will reply with a list of (property identifier, genesis transaction output) pairs, with whole message being signed with registry's public key. This alone is enough to detect basic problems (e.g. duplicate or ambiguous identifiers) and attacks (if you have two messages with different association, you can detect that this registry is faulty and prove this to others).

4. RESEARCH AND OPEN CHALLENGES

Although smart contracts have tremendous potential in solving real-life problems, most existing platforms and applications are still in their preliminary stage. Common problems smart contracts face range from semantic dependencies to the pseudonymous operation of criminal activities. In this section, we analyze limitations of existing smart contracts and solutions proposed in recent research studies, identify remaining challenges and provide insights on future directions. We categorize these challenges into three main classes, namely technology, legalization and usability and acceptance.

4.1 Technology

We discuss below the weak links and challenges in the composition and execution of smart contracts from a technical perspective. Note that we here only provide a limited number of examples, a more detailed mapping study on various issues of smart contracts can be found in.

4.1.1 Security

Security is one of the major concerns of any blockchain system and related procedure. In 2016, a re-entrancy attack in Solidity caused a loss over 40M USD and has led to a heated discussion over security issues of Etheruem smart contracts. In fact, many vulnerabilities are caused by the misunderstanding of the scripting languages.

Following the study conducted by Juels et al. in which several forms of criminal Ethereum smart contracts were explored, Luu et al. further studied security aws of existing Ethereum smart contracts including how contract execution and code behaviour are affected by the order of mined transactions, correctness of time-stamps and handling of exceptions. Delmolino et al. summarized common mistakes students made while programming smart contracts in the Serpent language. Apart from not realizing the limitation of the blockchain implementation, Delmolino et al. found that students often fail to encode state machines logically and ensure the incentive compatibility of a contract. Wang et al. categorized semantic vulnerabilities of smart contracts into transaction-ordering dependence, time-stamp dependence, mishandled exceptions, re-entry attacks and call-stack depth.

To enhance security of smart contracts, Luu et al. developed OYENTE for to analyzing and detecting security-related document bugs of smart contracts and proposed a set of improvements to the Ethereum protocol. Similarly, Securify and Mythril are also intended to ensure security of smart contracts. Some other groups are also developing alternatives. For instance, the Obsidian coin, developed by Coblenz et al., comes with a new programming language to enhance the security and usability of smart contracts. The improvement of existing smart contract languages and development of new ones should be carefully examined. Also, since the types of attacks vary from platform to platform, there is a need to understand the mechanism and vulnerabilities of particular blockchain platforms before using them.

4.1.2 Privacy

The pseudonymity of public smart contract do not necessarily guarantee their privacy. In particular, they do not guarantee unlinkability, which is crucial not only for privacy but also for fungibility. One way to protect privacy is to integrate an extra component for data protection, e.g., the Zero-Knowledge Proofs (ZKP) scheme as in ZeroCoin. Similar ideas and techniques have also been applied to smart contracts. In Hawk, a privacy-preserving compiler was built on top of the ZeroCoin protocol to enable the compilation of smart contracts with a cryptographic protocol while maintaining users' on-chain privacy and contractual security. With a minimally-trusted manager who executes the code, two users can perform actions on smart contracts without revealing the actual information.

Another branch of research is around coin mixing. For example, Coin Shuffle hides the origin of transactions among a group of users by allowing them to shuffle freshly generated output addresses in an oblivious manner. Similar proposals include Value Shuffle and CoinJoin. However, the adoption of encryption algorithms often brings extra computational overhead for the system, hence future development of privacy preserving techniques shall target light-weight solutions 4.1.3 Integrity.

Although the execution of smart contracts is regulated by hard-coded software programs and performed by all network participants, the data fed to smart contracts is still controlled by outside parties and cannot be fully trusted. Town Crier by Zhang et al. serves as a bridge between smart contracts and popular websites to secure the data-delivery. Deployed on the Intel Software Guard Extensions (SGX) hardware that provides a secure enclave for software processing, Town Crier can reliably fetch data from trusted websites to blockchain smart contracts, however, it does not ensure the integrity of data fed towards users. In most cases, users cannot directly access data on a blockchain or smart contract. Instead, they do so via wallet apps developed by other parties, which makes data integrity out of users' control.

4.2 Legalization

Before permissioned smart contracts become ready for a wider adoption in business procedures, many fundamental issues are yet to be solved. Notably, there is still lack of formalized ways of composing smart contracts to suit various design purposes, especially when legal contents are involved. From a legal perspective, there is lack of regulation and policies over smart contracts. It is sometimes hard for blockchains and smart contracts to obtain government approval. By now there is still the issue of enforceability and jurisdiction with this technology. When evaluating opportunities, organizations should carefully evaluate the effect of such lack of government acceptance. Scripting languages need to be regulated in a way to be more comprehensive and easy-to-use for both technical and non-technical people. In the case of Solidity, Frantz et al. have proposed a reasonable way of mapping contractual semantics to software declarations that covers the 5 essential components, i.e. "Attributes", "Deontic", "Aim", "Conditions" and "Or else" (or "ADICO"). According to the

authors, to successfully convert between institutional constructs and smart contracts, both directions need to be taken into consideration.

4.3 Usability and Acceptance

4.3.1 Usability

Smart contracts as logic-based computer programs have a limited level of interactivity and do not allow people to negotiate and make changes based on the later agreed modifcations like in traditional contracts, and they are not exible with exceptions such as glitches. Also, due to the P2P nature of blockchains, letting ordinary users control their data directly is risky, and the exchange rate can be unpredictable when crypto-currencies are involved.

4.3.2 Acceptance

Despite the hype of blockchains and smart contracts in both public and consortium domains, there are still a number of misconceptions about the technology. Firstly, there have been an inated expectation and many unrealistic use cases. Secondly, even with proper use cases, it can be hard to persuade stakeholders and users to accept the new technology. This could result in extra development costs and a low return on the investment. Some of the proposed use cases are in fact more efficient to implement via traditional databases. Hence, those who are interested in developing smart contract applications should keep in mind what can be achieved and what can not with it, as well as the development cost.Further, a summary of applications and challenges associated with them.

5. CONCLUSION

Smart contracts are gaining an increasing popularity in both public and private domains as they enable peer-to-peer operation on public blockchains and have the potential to improve efficiency and transparency in business collaborations. However, the current form of smart contracts are still limited in their ability to fulfill all expectations. We believe the future development should mainly focus on improving semantics of smart contracts, their integration with existing procedures, as well as their usability, acceptance and legality. If smart contracts can be made to work with enhanced security, legality and exibility, we can foresee a wider adoption of smart contracts.

REFERENCES

[1] Anonymous credit. http://diyhpl.us/~bryan/irc/bitcoin- satoshi/weidai/msg00398.html. Accessed: 2019-05-21.

[2] Blockchain for health data and its potential use in health it and health care related research. https://www.healthit.gov/sites/default/ files/11-74-ablockchainforhealthcare.pdf. Accessed: 2019-05-21.

[3] Chain of things. https://www.chainofthings.com. Accessed: 2019-05-22.

[4] Danish political party may be _rst to use block chain for internal voting. http://www.newsbtc.com/2014/04/22/danish-political-party-may-first-use-block-chain-internal-voting. Accessed: 2019-05-21.

[5] Eosio. https://eos.io. Accessed: 2019-05-21.

[6] Etherscan. https://etherscan.io/accounts/c. Accessed: 2019-05-21.

[7] Github. https://github.com. Accessed: 2019-05-21.

[8] Hyperledger. https://www.hyperledger.org. Accessed: 2019-05-21.

[9] Hyperledger: blockchain collaboration changing the business world.

https://www.ibm.com/blockchain/hyperledger.html. Accessed:2019-05-21.

[10] Initial coin offering (ico). https://www.investopedia.com/terms/i/initial-coin-offering-ico.asp. Accessed: 2019-05-26.

[11] Introducing project "bletchley". https://github.com/Azure/azure-blockchain-projects/blob/master/bletchley/bletchley-whitepaper.md. Accessed: 2019-05-21.

[12] Mythril. https://github.com/ConsenSys/mythril. Accessed: 2019-05-28.

[13] Neo smart economy. https://neo.org. Accessed: 2019-05-21.15