DGS

Decentralized Government System

Alexandru Panait
alexandru.panait@aipss.ro

Abstract. Given that Bitcoin is the first applicability of Blockchain technology, we can see how blockchain technology in the financial system is redundant because it can never reach the processing speed required by the financial sector to expand sufficiently to cover the entire market. Considering the main advantages of blockchain technology, such as **cyber security** and the **property of immutability**, we can easily see how they are extremely necessary to the government system.

The current complexity of the legislation goes far beyond what an ordinary citizen can assimilate in order to effectively fulfill legal obligations. Looking at this case at macro level from the point of view of the evolution of the society, considering the increase of the population and the standards of living, we can also observe an increase of the goods owned by the citizens, and implicitly a significant increase of the procedures and documents necessary for their management. The current bureaucratic system cannot scale efficiently by hiring more officials, as this only helps increase the chaos of entropy.

The error rate of the current system of government is directly proportional to the level of understanding and applicability of the laws. Due to the level of complexity of the legal acts, checking the blackholes in the system is extremely expensive, so covering the errors of the system by legal means of correction is unsustainable due to the high costs. Thus, by moving paperwork into digital structures through smart contracts[2], all data about citizens will be managed efficiently, in a private way inside a dedicated blockchain.

I propose in this white paper a decentralized model of government, through which the concept of **trust of the citizen in civil servants** disappears, and the concept of **trust of the citizen in the system** is introduced.

1. History & Concept

Legal history and bureaucracy is closely connected to the development of civilisations and operates in the wider context of social history. Legal institutions have complex systems of rules, players and symbols and these elements interact with society to change, adapt, or promote certain aspects of civil society. Current legislation is based on old laws that have been periodically readjusted. Implicitly, the current evolution of the legislative and bureaucratic system is by creating new legal frameworks over the old ones. The traceability of adjustments is extremely inefficient, and the current complexity of the legislation is far beyond what an ordinary citizen can assimilate, in order to efficiently fulfill his legislative obligations.

The lack of education regarding the legislation, or its interpretation in favor of a party is the main cause for non-compliance with legal provisions. Thus **democracy** is suppressed in the central systems by the abuse of power expressed by the people holding public leadership positions. The error rate of the legislative system will be massively reduced, the only problem remaining is tax evasion (not applicable for cashless society). To combat tax evasion into a fully digital public administration is easier because more resources can be allocated after the implementation of the system due to the redistribution of the labor force. Even corruption can be significantly reduced through the proposed technology.

Mankind faces many social problems, pandemics, etc. Humanity must have effective tools to meet all these challenges. To promote health, citizens must interact in person with someone only if they want to, not out of obligations to the government or other constraints. The citizen must be able to execute his obligations to the state strictly in the online environment, and in the most efficient way in terms of time. The lack of efficiency of the administrative apparatus is often the reason why citizens want to interact as little as possible. This lack of efficiency can often cause the individual's lack of motivation, inspiration and action from a psychological point of view, so the potential for progress in society is massively reduced from the root. The opening of a company must be done with a single click, the management of properties and the payment of taxes must be in a single easy-to-use interface, where the citizen can also send requests to the local administration.

Thus, in order to avoid subjecting the population to inefficient stress in terms of fulfilling legal obligations, the life events of the citizen should be able to be executed through a single central portal, without requiring the assistance of a civil servant to help the citizen to perform his obligations to the state. The transition to a simplified digital legislative framework must also involve streamlining current legislation. Each new digital

framework must be developed starting from the life event of the citizen, until the satisfaction of the state's need regarding that life event, all this considering the most efficient approach in terms of data storage and transfer.

All this must be found in a central interface, in order to normalize the interaction with the government, so that the know-how can be transferred much more easily among the citizens in order to accelerate the process of streamlining the governmental act through rapid adoption.

Traditional legislation no longer has traction in regulating technologies. All attempts to create traditional complex rules for new technological realities will result in the end with failures. A clear example is the regulation of decentralized technologies.

Considering that Bitcoin[1] is a first applicability of blockchain technology, which cannot provide by its own technology the necessary scalability and speed that is needed for the financial sector. Blockchain technology is much more suitable for systems that do not require a high speed of transactions. Normative acts and digital procedures are the most suitable digital data to be integrated in a blockchain type system, because they do not require instant speed, but only require transparency, security on a distributed storage in order to give the immutability property for all the documents.

2. Blockchain

We all know the basic principles of a blockchain, which is why I will avoid presenting them in this document. I will start by referring to Ethereum[2], being the most complete model that fits to develop the proposed system. We will adapt to the Ethereum technology the aspects related to the core blockchain, according to the following indications.

The blockchain structure in the blockchain will be structured by epochs. To allow data to be transported between nodes participating in the network much faster and more efficiently. At a first analysis, the necessary would be as follows:

- **Epoch 1** The data structure of the block in this era is exactly the same as in Ethereum, only that the message can contain 68 bytes to include the prefix of the institution. Also, everything related to smart contracts in Ethereum will not be used in this era but in Era 2. This era will only contain transactions with different network tokens and optionally messages attached to these transactions.

- **Epoch 2** contains some specific blocks that are formed once every ~1 hour. This era will contain basic smart contracts, which will define the organizational model in the blockchain for each structure.
- **Epoch 3** Contains blocks that are formed once every ~24 hours. All documents that want to be stored in full on the blockchain, not just their hash, such as those from era 1, will be introduced in this era.

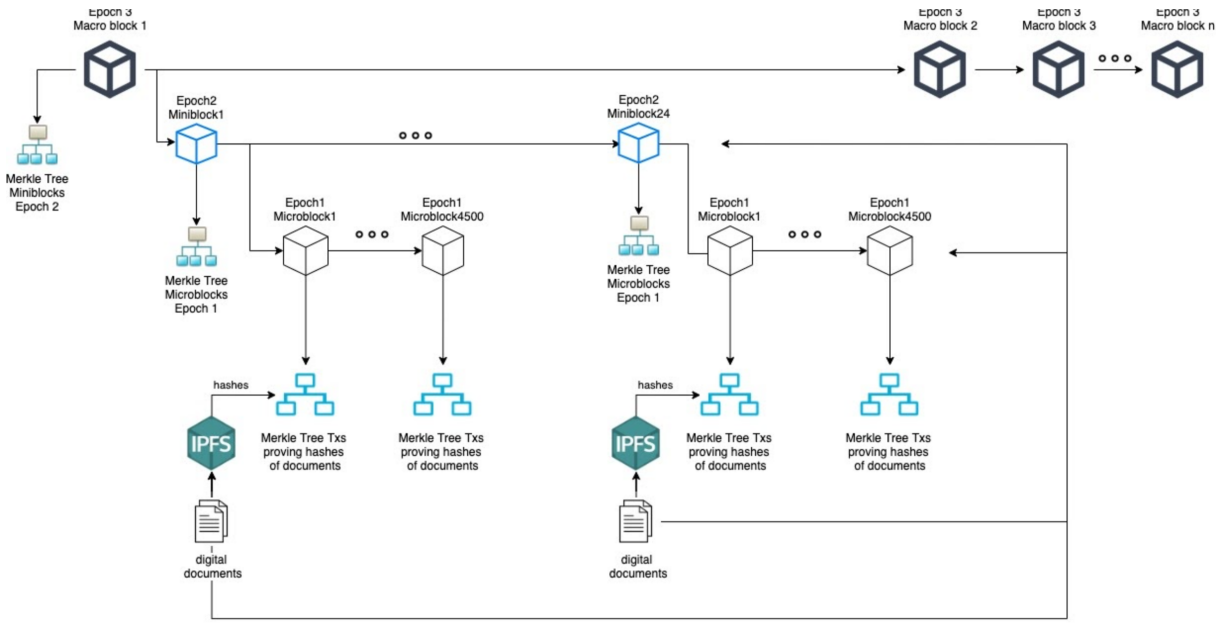


Figure 1. The architecture of proposed blockchain structure.

- Consensus & Mining

The whole system will be Proof-of-Stake (PoS), so the consensus which allows the block creation will be done through the governance token 0x00. This token can be held in the form of a single quantity, i.e. one unit, by each SCTI. Thus the consensus will be given by all citizens, through the proof of existence in the system. By doing this, the system operates transparently on all the data of the public administration, and the consent is given permanently by all citizens. Each has a unique chance, in direct proportion to undermine the next block. The initiator of this blockchain must ensure that the blockchain is private in the first instance, until there are enough SCTIs to meet a safe consensus, without the risk of a 50%+1 attack-. This token can also be revoked by SCTI MASTER, in case of death.

The mandatory existence of a monetary asset in any given blockchain is to avoid network spam from bad actors.

Thus, anyone who wants to publish any message on the blockchain will have to pay an insignificant fee. In our system, this fee is not necessary, because only SCWs and SCS will be allowed to publish on the Blockchain. The requests from each citizen signed as SCTI will be able to be published only by SCS through the central platform. Each SCTI is limited to a maximum number of daily interactions to avoid network congestion.

- Special Tokens for Voting Systems

The system also allows presidential / local elections based on each SCTI. On the election day, a special election token is being issued and airdropped to all citizens. This election token can be used only on that day by the voters. Each eligible candidate will have a unique voting wallet-. The system will know how to move from each SCTI that unique token in the candidate's voting wallet. All votes will be completely anonymized through Zero Knowledge like confidential transactions and ring signatures[3].

This can take the administration to the next level, where local decisions are made by the community, not by the authority that was once selected by the community. This eliminates the possibility that the elected administration will not perform in accordance with the electoral promises and the real needs of the citizens.

- Scalability & Sharding

Optimization: For a given key to find in which shard is stored, a prefix is being attached to the key in order to deterministically predict the shard in which the data is being stored.. Each hash will have 64-bytes with a 4-byte prefix that will be used by each local shard. Maximum number of shards is 65,536, which should be enough.

EXAMPLE: **0001**aa58b21b01d6b8a99c1a5856962dbac36c758a79dc0a77c2e013ce2c39ecdc8a
Aaa10a7cb27e52927eacabbb7ecc738b0fea50b3967945257c43a67eb753cb465bd0

Because each local government and structure has its own set of rules that derive from the legislative frameworks, to make the system more efficient, each local public administration will have its own shard. This shard will communicate only with the shard of the upper structure. This will avoid graph communication between shards. When searching for a document on a different shard, the system will know that the document is in that shard from the attached prefix, which is unique to each institution. Thus, for the system to work, the shards do not have to communicate important data with each other extremely frequently.

Most documents will only have the fingerprint (hash) stored in the blockchain, in Epoch 1, not the entire document. The strictly necessary documents will be stored in Epoch 3. It is recommended to store only SCL, and for the rest of the documents to avoid occupying the space unnecessarily.

3. Smart Contracts

I've noticed that institutional organizations work based on 4 essential elements: **Law, Hierarchy, Identities and Documents**. All these main elements have secondary elements of maximum importance for the system to properly function. To achieve this, for each of the elements mentioned before, we will create a smart contract. In order to be able to do an action in the system you need a **digital identity**. Once you have a digital identity, you can issue **documents** in the system. The documents circulate according to a **procedure**. This procedure works on the basis of an **organizational chart** with **public servants**, which is related to the hierarchical organization of the **structure**. All structures are also subordinated to each other in the form of a hierarchical tree. All this structure is defined according to the **legislative text**. It is recommended to avoid hard coding these main elements, in order to allow later an easier modeling and improvement of the structures in the system.

Identities, institutions, and elements in organization charts will essentially be just identification hashes. In the case of digital identity and the elements in the diagram, they will be associated with a public key, through the prism of which private key you can prove that you publish something in the blockchain on behalf of that digital entity. In the system anyone can generate a set of keys at any time, but it will have no value if the public key is not associated with any element. Thus, in case of losing a key pair, the system will be able to reorganize in order to provide a new key to the real owner.

Thus I define the following structure for each smart contract:

(SCW) - The smart contract for workers/public servants

This smart contract will define the hierarchical structure in the blockchain. In the first instance, it will be an independent object in the blockchain, without having any reference to other elements in ecosystems. Only by publishing inside an older block the first SCO that will refer to the SCS hash will the connection with the other elements in the ecosystem be produced. This smart contract will often occupy a terminal node position in the entire governing ecosystem.

Data Structure:

- *Random ID Hash (Unique hash in the network used for identification of the SCW)
- *Position Name (Ex: Department Chief, Accountant, Advisor, etc.)
- Previously ID Hash (Required only if this smart contract is an update of an old one)
- *Associated Public key (The public servant key)
- *Associated SCTI (Citizen Identification for the public servant)

Special Functions:

1. Update Smart Contract (Using this function an updated copy of the initial smart contract can be published. This one will have a new random hash, and will refer to the old hash. This can be made only with root signature from the related SCO).

(SCS) - The smart contract for structures

This smart contract will define the structure inside the blockchain. In the first instance, it will be an independent object in the blockchain, without having any reference to other elements in ecosystems. Only by publishing inside an older block the first SCO that will refer to the SCS hash will the connection with the other elements in the ecosystem be produced.

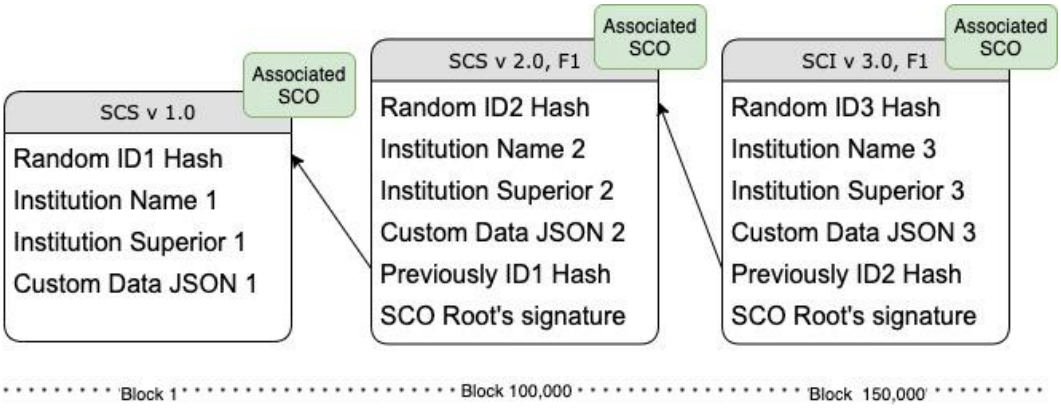
Examples of structures: Planet, Continent, Union, Country/Government, Ministry, Regional Institutions, Local Institutions, Hospitals, Schools, Departaments, Councils, etc.

Data Structure:

- *Random ID Hash (Unique hash in the network used for identification of the SCS)
- *Structure Name
- Structure Superior Hash (To which structure is this subordinate, in case of empty, is a root structure)
- Previously ID Hash (Required only if this smart contract is an update of an old one)
- SCO root signature
- Custom Data JSON (Any other data required to be associated with the structure for further usage)

Special Functions:

2. Update Smart Contract (Using this function an updated copy of the initial smart contract can be published. This one will have a new hash, and will refer to the old hash. This can be made only with SCO's schnorr, after an SCO will be associated with the structure). The first SCS will always be the origin and it's hash should always be used for identification of the SCS.



(SCO) - The smart contract for organization chart

This smart contract at the first iteration aims to associate with an SCS and define its operating structure. SCOs as data contains a tree. Each node from the tree being represented either only existing initial SCS hashes or only new hashes, in this case meaning that the organization chart refers to one where the nodes are officials, not entities such as structures. Trees for public servants where each node has a new hash representing the position identifier and an associated hash from an (SCTI).

SCS and SCO should be published in the same block in order to avoid blockchain configuration by an attacker which will link to all empty SCS and uninitiated SCO.

Data Structure:

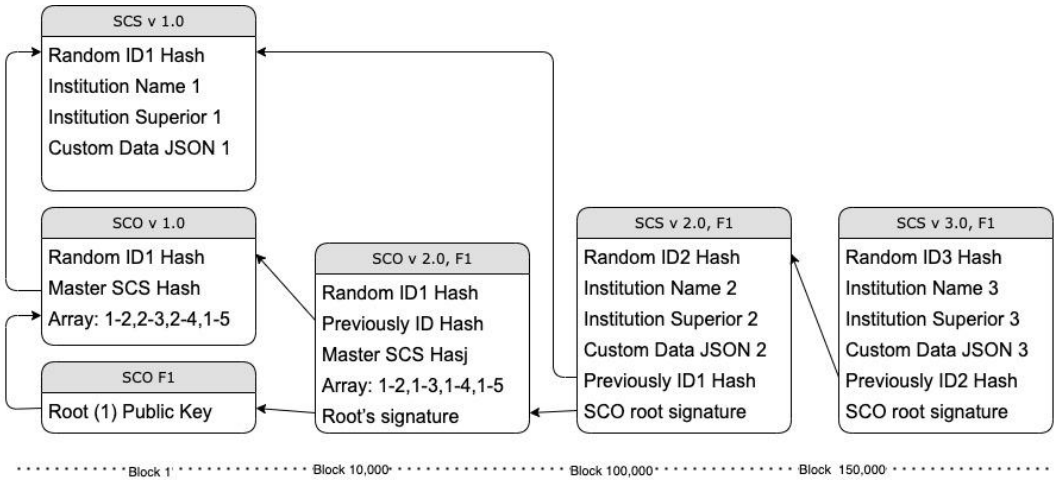
- *Random ID Hash - Unique hash in the network used for identification of the SCO
- Master SCS Hash
- Previously ID Hash - Required only if this smart contract is an update of an old one
- *Hashes Array - The hashes should be related to an SCW(for executant) or SCS(Recommended only for departments, councils or other

substructures from the same structure. Avoid referring to a structure.). The hashes should be in the array two by two. The parent to be even and the child to be odd. Ex: 1,2,2,3,2,4,1,5; Where the association is 1-2,2-3,2-4,1-5;

- Root's Pub Key (Required only after Initiate)

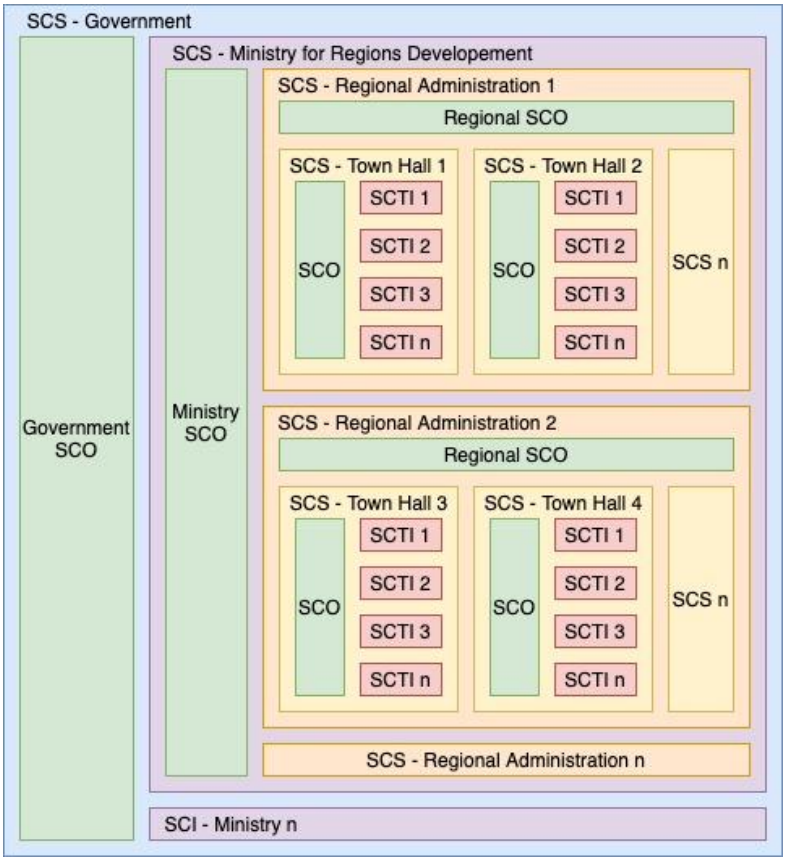
Special Functions:

1. Initiate Smart Contract (Assign a public key to the root node)
2. Update/Forking Smart Contract (Using this function an updated copy of the initial smart contract can be published. This one will have a new hash, and will refer to the old hash.)
3. Update Node Pub Key (Assign a public key to one the direct child)
4. Recover root key (Only by signing the root in the SCO of the Master SCS, the root key can be changed.)



This will let the system know retroactively the status of the entire system at each moment of time by block number, in order to perform checks. The vector in the SCO will be used to define different SCP for fome SCD.

The ecosystem of each government will be formed on the basis of a hierarchical combination of SCO and SCS as in the diagram below. Thus a hierarchy of SCS and SCO can be defined for any type of organization. Thus, the SCS ultimately defines a group. Each SCS has an associated SCO, which is used for all procedures (SCP) for the efficient circulation of documents (SCD), which targets the activity and assets of citizens in the system (SCTI). Government SCS being a root type structure.



(SCTI) - The smart contract for trusted identity

In the case of certified identities in the system, an SCTI MASTER associated with an SCS is required. Once an SCS is assigned an SCTI MASTER, then no lower SCS can be assigned another SCTI MASTER. The root of the SCO associated with an SCS can assign other lower SCSs as being SCTI Issuer, that can use the SCTI functions in the system, using their SCS's identification hash.

Data Structure SCTI MASTER:

- *Random ID Hash (Unique hash in the network used for identification of the SCTI MASTER)
- *Master SCS Hash
- Previously ID Hash (Required only if this smart contract is an update of an old one)

- *Hashes Array (SCS that became SCTI ISSUERS and are allowed to publish new SCTI in the system).

Special Functions SCTI MASTER:

1. Initiate Smart Contract (Assign a SCTI Master to an SCS)
2. Update/Forking Smart Contract (Using this function an updated copy of the initial smart contract can be published. This one will have a new hash, and will refer to the old hash. It will be mainly used to add new SCS's hashes in the Array)
3. Create SCTI (This can be used by any hash from the array, that can prove the ownership of the current private key of the SCO's root associated).

Data Structure SCTI:

- *Random ID Hash (Unique hash in the network used for identification of the citizen)
- Associated Pub Key (Referring to the public-private key pair that can sign for the given citizen)
- SCTI ISSUER hash

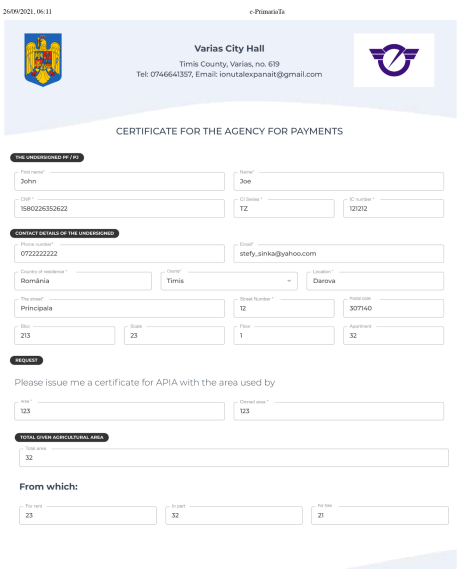
Special Functions SCTI (Only for SCS Issuer only):

- Change the Associated Pub Key (In case the citizen is losing his key pair)
- Change SCTI ISSUER hash (In case the citizen is moving in a different Town)

To avoid attacks on the network consensus, each SCTI Master will be able to publish a limited number of identities in the system for each block. This, at launch, will hinder the process of enrolling all citizens in the system. It will be possible to add 1/10 more people than in any of the previous blocks that has the maximum number of new identities added. But never more than the security figure, which is 500,000.

(SCD) - The smart-contract document

Each SCD MASTER will define an official document template (Ex: rest leave application, certificate, fiscal statement, land book, etc.). An SCD MASTER can be published by any SCS. Once published on the blockchain, it will render the official structure of all documents of the same type.



When filling an SCD MASTER, a .scd file will be generated which will be a document with a JSON that will contain all the data of the official file and will be stored off chain. The registration of this file will be done on the blockchain only by hashing the content of this .scd. This proves that at block x the file was registered in the system, and anyone who can read the data in the file can validate this. I recommend that the central government platform be able to interpret these files in a UI / UX friendly way. As in the example above.

Data Structure SCD MASTER:

- *Random ID Hash (Unique hash in the network used for identification of the SCD MASTER)
- Previously ID Hash (Required only if this SCD is an update of an old one)
- *Publisher SCI hash
- *Title
- *JSON DATA (Sections of the document, inputs, texts, etc.) Example:

{SCD_MASTER_HASH, Previously_HASH, SCI_HASH, TITLE,
{SCD_MASTER_HASH1,SCD_MASTER_HASH2,
{ELEMENT_HASH1,ELEMENT_HASH2,
{ELEMENT_METADATA,
ELEMENT_PRIVACY_TYPE (1,2,3),
ELEMENT_TYPE(In case of select, it will contain also the options)}}}}

OPTIMISATION: Where SCD_MASTER_HASH is the identification hash of another SCD_MASTER, such as "Citizen identification data", which contains the Elements: unique identification code, name, surname, date of birth, etc. Thus this template for identifying citizens is published only once and reference is made to that structure in any SCD_MASTER that contains the related structure, for example any declaration.

All these SCD templates will exist in the central app as universal document builder. Thus the reference of the same data will be the same everywhere.

ELEMENT_HASH represents each element, it can be defined anyway, and the universal platform will know how to interpret each element, so the user experience will be extremely simple. For example: map(coordinates),text, inputs, selects, checks, etc. Each element can have validators based on regex built in, so the user is limited in making the wrong completion of the template.

Like identification data, which only needs to be read once. Thus, even if the reading key is lost in the future, there will be no problems because that data will no longer be relevant.

Data Structure SCD:

- SCD MASTER HASH
- *To whom (Privacy chosen in order to avoid attacker to get more informations about the victim)
- *Issuer unique ID HASH |
- CCBH - Control Current Block Height (Will be automatically provided by the universal platform). While verifying the document, few blockheight delays are accepted.
- *JSON DATA (Data and values according to SCD)
- Document Public Hash
- SCP associated hash
- SHA256 (Document Public Hash) Using Issuer current Pvt Key
- Document Private Hash | Hash(Document Public Hash, Sha256) Will be used to generate an unique QR Code

Special Functions SCD:

- Update/Forking Smart Contract (Using this function an updated copy of the initial smart contract can be published. This one will have a new hash, and will refer to the old hash.)

(SCP) - The procedural smart-contract

SCP will combine elements such as: SCD, nodes from SCO, according to some SCLs. Their main purpose is to define the set of signatures required for a document to be validated by the institution in the system.

The SCP will define the necessary procedure that each SCD must go through. Some SCDs will be the root of the steps in the SCP, other SCDs may be elements that are attached to different steps in an SCP. Thus we will have two types of documents. We will have SCDs that behave as root, and from these start procedures, or SCDs that behave as attachments, and these are assigned during a procedure.

The steps in the procedure are executed off chain, according to the published on chain procedure.Also an SCP will contain a vector with the hashes of the nodes in the SCO that must approve this document. Only after the complete signing of the document by the officials is the publication on the blockchain.

Data Structure SCP:

- SCL HASH
- Array with steps and SCD hashes
- Array with required signatures

Special Functions SCP:

- Update/Forking Smart Contract (Using this function an updated copy of the initial smart contract can be published. This one will have a new hash, and will refer to the old hash.)

(SCL) - The smart contract for Legislation

Will define the legislative text, and will also contain the digital rules of applicability of the legislation in the system.

All these operations based on smart contracts will have easy-to-use interfaces to perform the usual activity for both citizens and public servants. For more complex operations, the creation of new smart contracts with integration into the interface is necessary. Subsequently, the smart contract template, which can be multiplied at the level of any structure, and even readapted.

4. Known blockchain vulnerabilities

- The presented system is **Quantum Safe** In theory, quantum computers are a threat to current encryption schemes. Thus, the system does not publicly provide sensitive data, because only the fingerprints of the documents are stored, as well as all the smart contracts that present the organizational models of the institutions. In the event of an attack by breaking cryptography, the only compromised data will be the documents

stored in Age 3. If they are encrypted, there is a chance that the encryption cannot be broken to access the data in the document.

- **Losing a key** is no longer a problem. If a citizen loses his key, based on the unique hash, it can be replaced by the authorities. The special voting token can be used locally if a master's key in administration is compromised. The system must present a special smart contract, through which a voting token is issued at the level of the targeted authority, where it can be voted by all SCTIs assigned to that structure. Thus the system is very safe in case of key loss, the community always having the power to help in case of compromising access to the system due to human inattention.

- **The 50% + 1** attack is no longer a problem, given the consensus model. Thus a consensus of 50% + 1, in this system if it is reached, also represents a democratic consensus. What is not a problem, but in fact, is the basis of democracy.

- Avoiding the storage of sensitive data is another advantage of the system. Through offchain digital documents, the system will not contain data that can be compromised in the event of an attack.

5. General Observations

- For better performance, the structures SCS, SCO, SCW, SCTI can be implemented as special transactions, and SCD, SCP and SCL remain in the form of smart contracts, because they require a subsequent greater dynamics.

- The presented concept can be developed in a special blockchain, under the complete control of the government, where all the data is public for the citizens and the whole decentralized system is coordinated by the government. Anyone with the ability to own and verify the blockchain, but only the government with the ability to publish data on the chain.

- The system can also contain smart financial contracts, through which a new token can be issued to represent a financial asset, or anything else that needs to be quantified.

- The network consensus uses PoS, so it is environmentally friendly

6. Conclusion

I presented a model of decentralized governance through a scalable blockchain technology. Where all the frameworks of government organization and management are presented transparently on the chain. All documents that refer to the components of the state have a hash based fingerprint stored on the Blockchain. Voting for the next block is decentralized to all citizens, and the digital organization of the system is permanently under the control of the government, with the possibility of agreement of the population. The system provides the necessary transparency to the citizens, and helps streamline institutional procedures to optimize the administrative act. Starting from the Ethereum Protocol, with the necessary modifications for the efficiency of the system, and with the intelligent contracts necessary for the governmental organization. Smart Contracts have been designed to allow the readjustment of the organization to more optimal versions. Any necessary blockchain protocol changes can be added to the presented document.

Special Thanks to:

- Ph.D. Dragan Boscovic, Supervisor
- Alexandru Ionut Budisteanu, Blockchain Advisor

References:

1. Satoshi Nakamoto, “Bitcoin”, https://bitcoin.org/bitcoin.pdf
2. Vitalik Buterin, “Ethereum”, https://blockchainlab.com/pdf/Ethereum_white_paper-a_next_generation_smart_contract_and_decentralized_application_platform-vitalik-buterin.pdf
3. Uriel Feige, Amos Fiat, 1 and Adi Shamir, “Zero-Knowledge Proofs of Identity”, https://fisher.wharton.upenn.edu/wp-content/uploads/2020/09/Thesis_Terrence-Jo.pdf
4. Gregory Maxwell, Andrew Poelstra1 , Yannick Seurin2 , and Pieter Wuille1, “Schnorr Signature”, https://eprint.iacr.org/2018/068.pdf
5. Gus Gutoski1 and Douglas Stebila2, “Hierarchical deterministic Bitcoin wallets that tolerate key leakage”, https://eprint.iacr.org/2014/998.pdf