eGov-DAO: a Better Government using Blockchain based Decentralized Autonomous Organization

Nour Diallo, Weidong Shi, Lei Xu, Zhimin Gao, Lin Chen, Yang Lu,
Nolan Shah, Larry Carranco, Ton-Chanh Le, Abraham Bez Surez, Glenn Turner

ndiallo@cs.uh.edu, wshi3@central.uh.edu, lxu13@central.uh.edu, gao@kell.vin, lchen49@central.uh.edu,
ylu17@central.uh.edu, nshah10@uh.edu, lccarran@central.uh.edu, letonchanh@gmail.com, abaezsua@central.uh.edu, gaturne@gmail.com

Computer Science Department, University of Houston, Houston, Texas, 77004

***Abstract*—E-government system has greatly improved the efficiency and transparency of daily operations of a government. However, most of existing e-government services are provided in a centralized manner and heavily rely on human individuals to control. The highly centralized IT infrastructure is more vulnerable to outside attacks. Also, it is relatively easy to compromise the data integrity by inside rogue users. Furthermore, relying on individuals to monitor and control some of the working flows makes the system error-prone and leaves room for corruption. To address these challenges, we propose to use the blockchain technology and decentralized autonomous organization (DAO) to improve the e-government system. The blockchain-based DAO system works in a fully decentralized way and is immune to both outside and inside attacks. At the same time, operations of such system is only controlled by pre-defined rules; thus, the uncertainty and errors caused by human processes are greatly reduced. We provide a concrete use case to demonstrate the usage of DAO e-government and evaluate its effectiveness.**

***Index Terms*—blockchain, DAO, e-government, transparency, auditability**

I. Introduction

Nowadays, most countries in the world have been providing *e-services*, in which the governmental public services are implemented by information and communication technologies, to serve its citizens better. Initially, developing an e-service may be challenging, due to the complexity of government policies or the disinterest of its citizens in new technologies. However, there are still opportunities to attract more interest and collaboration for the e-governance by providing a secure, rigorous, autonomous, and transparent digital system for its services. Such system e.g voting registration, driver licence processing, usually improves the government's productivity and efficiency on collecting, securing, and sharing information. Therefore, it will be more likely to be received by the citizens because of its speed and transparency. For instance, the digitalization of government's services helps to replace costly in-person or postal communication by the use of mobile phones and emails.

However, despite numerous efforts in improving the government system e.g voting system, it is still not sufficiently secure and transparent. The system is usually built on a highly centralized IT infrastructure, which is more vulnerable to outside attacks. In addition, the system is heavily controlled by human individuals, which makes it error-prone and leaves room for corruption. For example, inside rogue users can easily compromise the data integrity of the system. In this paper, we address these challenges of the e-government system by leveraging the emerging *decentralized blockchain technology* [1] to bring security, immutability, reliability, and transparency to the system. While the blockchain technology can be applied to a wide range of government services, this paper focuses on utilizing a blockchain feature called *Distributed Autonomous Organization* (DAO) for a concrete use case of the *government contracting service* [2].

Government contracting is a service that allocates public contracts to given vendors. The allocation process is inefficient since it requires multiple inter-agency interaction and involves many human labors [3]. In order to provide a simple and convenient interface, the government allocates both human and financial resources, but it results in a minimal transparency in governance. In this scenario, blockchain technology would increase transparency and trust, reduce costs, and simplify the process.

The blockchain framework introduced in this paper is generic and can be apply to any policy for government contracting. In this paper, we consider the policy of the U.S. Small Business Administration as a case study [4]. First, we describe the general application of blockchain for the U.S contracting and introduce several regulations of allocating contracts required by the U.S. Small Business Administration policies. Next, we implement a DAO framework which summarizes these requirements to regulate the allocation process. Moreover, the framework also defines the parties of the system and their activities, and models the main process of contract selection following the policies. Finally, we provide a set of enforcement rules to validate the process.

The remainder of the paper is organized as follows: In Section II we briefly introduce the background technology including blockchain, smart contract, and DAO. Section III describes the detailed design of the government DAO architecture. In Section IV we analyze and evaluate the proposed solution. We conclude the paper in Section V.

II. Background

In this section, we briefly review the background of blockchain technologies.

A. Blockchain and Smart Contract

A blockchain, or a distributed ledger, is a system involving multiple participants who achieve consensus over a data set and maintain the data locally. Blockchain systems are developed under different trust models with different consensus protocols, e.g., proof-of-work [5] and proof-of-stake [6]. A *permissioned block chain* usually has an identity management mechanism to control who can participate the block construction. Authorized users can run different Byzantine fault tolerant protocol to determine whether a block should be accepted and added to the block chain. In this paper, we only consider *public block chain* constructed using proof-of-work as it has received intensive studies.

The idea of a smart contract was first introduced by Szabo [7], which is described in Definition 1.

Definition 1: A smart contract is a set of promises, specified in a digital form, including protocols within which the parties perform on these promises [8].

Block chain provides an ideal platform for smart contracts to be executed in a decentralized way. Roughly speaking, a smart contract is a piece of program that consists of a set of rules and corresponding operations of related accounts. On a high level, the life cycle of a smart contract in a public block chain system can be summarized as follows:

1) Creation. Users involved in the contract work together to build a smart contract and use digital signatures to guarantee its authenticity. The smart contract is then submitted to the system.
2) Acceptance. Users who have received the smart contract first check its validity and then do mining to include it in a new block. The new block is then broadcast to the block chain.
3) Execution. Users in the system who have accepted a new block containing the smart contract will execute it locally according to its instructions, and obtain the result. Users then do mining to build a new block to hold the result and broadcast it to the system.
4) Result confirmation. Users who have received a block containing the result of the smart contract will verify its correctness to determine whether to accept it. In most cases, the verification is done by re-computing the smart contract and comparing the result with the one that is received.

B. Decentralized Autonomous Organization

The concept of blockchain based smart contract can be further extended to *Distributed Autonomous Organization* (DAO). A DAO functions with a set computer programs, in this case smart contracts, which define in advance the rules governing an organization [9]. Like an organization, a DAO behaves with objectives and expectations to achieve a set of goals. In theory, a DAO can be set for any reasons or goals [10].

Usually, a traditional organization is owned by an individual or stakeholders and registered in a centralized system, e.g. government. This type of organization is managed in a hierarchical structures. Directors in the organization decide the future actions and the remaining members just follow the decision. For example, in a company, the board has the power to decide how to deal with the funds and to set a goal; other employees will be assigned with tasks to complete the goal. In contrast, all participants in a DAO have the same rights to make decisions. It means that no one has special privileges in operating the organization.

Autonomy is the major feature of DAO. A DAO requires an automated program to ensure that the decisions can be executed without any manual intervention [11]. For a smart contract, if a set of events in the contract is triggered, it will be executed automatically by the decentralized system. For instance, if a contractor receives enough votes from the members of DAO, funds will be released for him or her. Typically, DAO also involves the following features:

- DAO comprises all data/requirements (resources) needed to complete a task/process.
- DAO can enforce partnerships between companies/organizations without any physical interaction because a smart contract runs automatically on nodes in the network and does not require any human interaction.
- DAO can be set up as easy as creating a company. It is a global organization open to anyone, regulated by a smart contract and operated using computer codes/coding.
- DAO will transform organizational processes and create a new platform (distributed system) that impacts our societies.

III. Blockchain based Government DAO

In this section, we adopt smart contracts and DAO to build an automated blockchain based e-government system, or government-DAO (eGov-DAO) for short. We first describe the high level architecture of the government-DAO and then provide its detailed design.

A. High Level Architecture

Participants of the eGov-DAO is divided into two groups, the DAO maintainer and the users, as depicted in Fig. 1.

- Blockchain maintainer. As we consider public blockchain constructed using proof-of-work, any one with computation/storage resources can connect to the system and contribute to block construction [12]. Some of the blockchain maintainers may not be honest, e.g., they could try to alter or remove transaction data stored in the blockchain system or provide wrong results of a smart contract. However, we assume that the majority of them are honest maintainers so that those malicious ones cannot compromise the blockchain infrastructure;

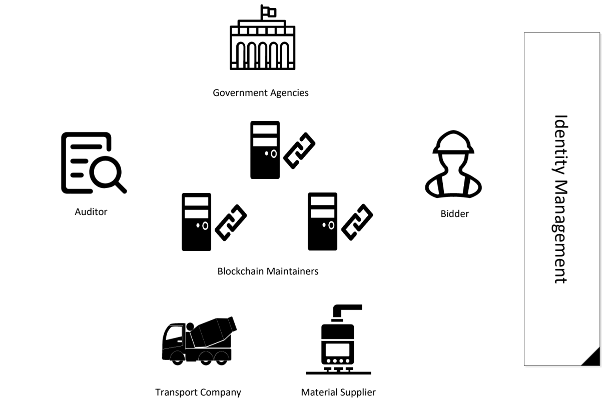

Fig. 1. Two types of participants in the eGov-DAO system: the blockchain maintainers and users including government agencies, auditors, and other vendors that work for the government. The identity management component supports identity control and does not participate in daily operations of the eGov-DAO.

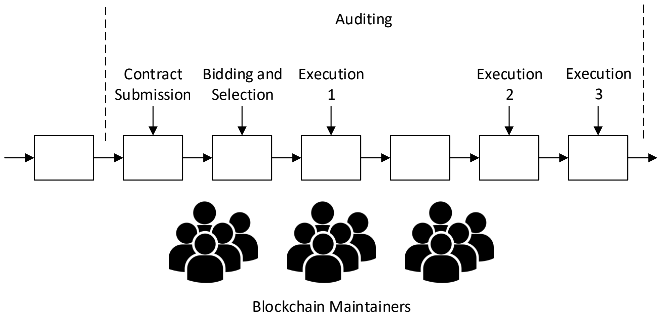

Fig. 2. An overview of the work flow of the government-DAO.

- User: The eGov-DAO has a variety of users, from government agencies who manage government projects, project auditors, to all types of vendors. They can submit different types of transaction records to the eGov-DAO, which will be verified by blockchain maintainers before accepted to be included in the blockchain. According to the rules defined in the eGov-DAO, an authority party may need to approve one transaction before it can be accepted.

Fig. 2 gives an overview on the work flow of the proposed government-DAO. For a government contract, the eGov-DAO tracks each step of it during its whole life cycle.

B. Detailed Design

There are various forms of government contracts. For example, *fixed price* where vendor can compete and go through a bidding system, and *cost reimbursement* in which the price is negotiable. All of them can be implemented using the DAO concept. Initially, the general requirements of the contract are implemented in smart contract, which will be used as the basic rules to control execution of the contract.

All users, e.g., vendor, company, and regulators, are governed under these initial requirements. According to the US Department of Defense, there are at least eleven steps to acquire a government contract. The steps can be grouped into four categories: contract preparation and submission, bidding and selection, contract execution monitoring, and auditing. At each stages of the execution, the system checks and validates that all the parties meet the requirements before moving to the next stage of the execution.

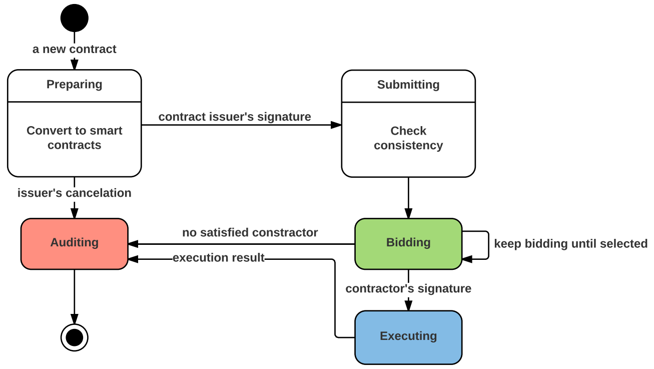

Fig. 3. The life cycle of a government-DAO. Usually, an eGov-DAO requires contract issuers, regulators, contractors, executors, and auditors to participate in.

The eGov-DAO automatically executes all the transactions and the results are available to the public including all users. The system first verifies that contract preparation is submitted, validates and adds it to the blockchain, then moves to the bidding step. Fig. 3 demonstrates the life cycle of a government contract in the eGov-DAO system.

User registration. Unlike the blockchain maintainer, to become a user of the eGov-DAO, an entity has to go through a registration procedure before he/she can participate in. For example, a D-U-N-S Number, like the social security number for a company, provides a unique form of identification and records all the information about the company [13]. This number can help others like lender and business partner to have an idea about the company. When the entity registers, it has to provide such information to an authority and the authority will issue a certificate to it as its identity in the eGov-DAO. In most cases, the certificate is a public/private key pair where the public key is embedded in a digital certificate [14], [15]. The rules to use and manage the certificate are also embedded in a smart contract, and no human needs to be involved.

Contract preparation and submission. A traditional government contract needs to be converted to the form of a smart contract, i.e., all clauses in the contract need to be written in the language supported by the eGov-DAO [16]. Algorithm 1 gives an example of an eGov-DAO contract. After generation of the contract, the contract issuer digitally signs the contract and shares it with other regulators. Each regulator or actor checks whether the contract is consistent with related rules and generates a digital signature using his/her private key. After all these procedures are finished, the contract and related signatures are sent to the eGov-DAO and available to the public.

Note that contract contents include both detailed requirements/description of the project and contractor selection criteria. The contract also includes a set of milestones where the contractor should submit information to the eGov-DAO. All the information is encoded into corresponding smart contracts.

Bidding and selection. After the contract is confirmed on the blockchain of the eGov-DAO, the bidding procedure starts. Fig. 4 shows a sequential diagram of this procedure. More specifically, all contractors who are interested can prepare a proposal, generate a signature of the proposal, and submit to the eGov-DAO. Each submitted bidding will be checked by participants of the eGov-DAO, and only those that satisfy the pre-defined criteria will be accepted and recorded in the blockchain. Then participants run the procedure embedded in the smart contract to select the winning contractor [17]. Information of the selected contractor is also embedded in a block and stored on the blockchain. When this block is confirmed, the selected contractor should follow the requirements/description of the project.

Note that the bidding information usually needs to be kept in secret before the selection procedure. As the blockchain is publicly available, a protection mechanism is required. A cryptographic commitment scheme can be used to address this problem. The scheme allows one to commit a chosen statement while keeping it hidden from others, with the ability to reveal the committed statement later [18]. For the eGov-DAO, each contractor puts his/her bidding information into a commitment and submits to the blockchain, which prevents others from learning the bidding. At the selection step, each contractor can open his/her commitment and the binding property prevents one from modifying the committed statement.

Algorithm 1 provides an example pseudo code for the smart contract to manage bidding procedure.

Monitoring contract execution. When the selected contractor meets one milestone defined in the contractor, he/she submits required information of the milestone to the eGov-DAO. A third party such as the project supervisor can also provide information for the milestone. Participants of the eGov-DAO checks whether submitted information meets the requirements defined in the contract to determine if the contractor can proceed to the next step. The decision is also recorded in a block and appended to the blockchain [19].

From Fig. 5, we learn that smart contracts are executed by the available nodes in the network, and the execution result will be submitted to the blockchain for others to verify. Once the system reaches consensus on the result, it will be added to the blockchain of which existing participants preserve a complete copy.

Auditing. Because of the immutability property of blockchain, the eGov-DAO provides a good support for auditing. All contract related information, e.g., project requirements, received bidding information, and contract execution/inspection records, is stored on the blockchain and difficult to modify. An auditor can easily trace back each step to see whether there is a violation.

Algorithm 1 A smart contract manages bidding in the government-DAO.

```
Input:
chairperson:address ← address of chairperson
voters:array ← list of voters
proposals:array ← list of proposals

function VOTE($proposal\_i$, $voter\_a$)
    if voter_a.voted then return false
    if proposal_i ≥ proposals.length then return false
    $voter\_a.voted \leftarrow true$
    $voter\_a.vote \leftarrow proposal\_i$
    $proposals[proposal\_i].voteCount \leftarrow$
$voter\_a.weight$ return true

function WINNINGPROPOSAL
    $win\_c \leftarrow 0$
    for $prop \leftarrow 0$,$prop < proposals.length$,$prop$++ do
        if $proposals[prop].voteCount > win\_c$ then
            $win\_c \leftarrow proposals[prop].voteCount$
            $\_winningProposal \leftarrow prop$
    return $\_winningProposal$

Comments:
The goal of function “winningProposal” is to find out a
proposal which receives maximum votes.
```

IV. Analysis and Evaluation

In this section, we analyze and evaluate the proposed government-DAO from security and performance perspectives.

A. Security of the Government-DAO

There two major types of threats to the government-DAO system (eGov-DAO):

- Data integrity. Data is the foundation for all functions provided by the eGov-DAO. If an attacker can alter/delete existing data or insert new data to the historical data, it may cause serious consequences. The blockchain structure can avoid all these risks to protect the integrity of the data. In order to compromise data stored in the blockchain, an attacker has to compete with all the honest users who maintain the blockchain system on producing new blocks. It is a big challenge for both proof-of-work and proof-of-stake, and the probability of success is very low [6], [20].
- Rule integrity. In the eGov-DAO system, rules are logics embedded in smart contracts. Rule integrity means that an attacker cannot influence the execution of a smart contract to get preferred results. Rules (smart contracts) are embedded in blocks before they are executed so the attacker cannot modify them directly. According to the smart contract execution model, everyone will run a contract to see the results rather than simply accepting one. Therefore, as long as the majority of users of the

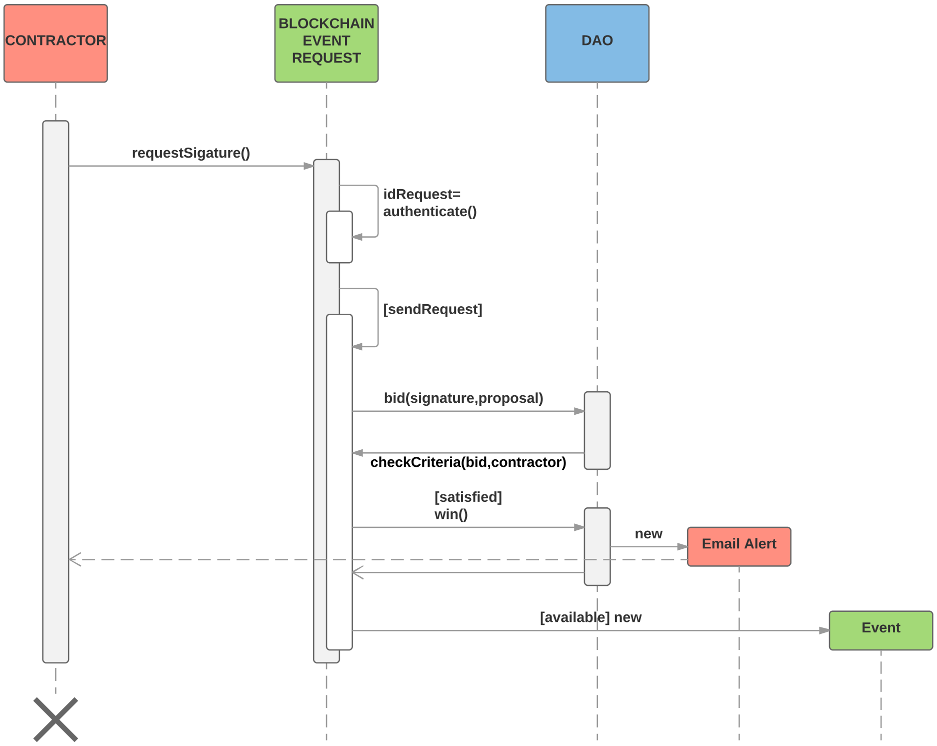

Fig. 4. Multiple contractors start to bid for the contract and the selected one will be notified by DAO and blockchain.

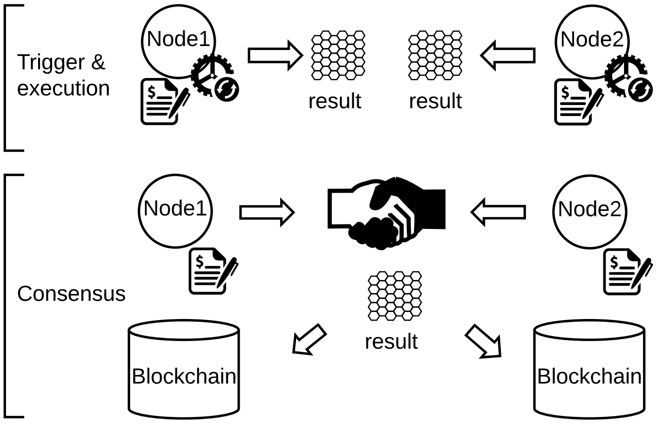

Fig. 5. Smart contract execution can be triggered by an event internally or externally; the execution of a contract would produce a result on which all participants agree.

system are honest, an attacker cannot compromise the rule integrity by providing a wrong execution result as it will be rejected by the majority.

Another potential security concern is data confidentiality stored on the blockchain that the eGov-DAO relies on. Because we are using public blockchain, everyone can access data stored there. In most cases this is not a problem because the government has to make such information public eventually. In case the data is sensitive, all parties involved can negotiate a data protection key and encrypt everything before submiting to the eGov-DAO. This mechanism can protect the data from unauthorized accesses but also reduces the transparency of the government. Therefore, data that needs to be protected should be defined in the eGov-DAO in advance to make sure that this function will not be abused.

B. *Performance of the Government-DAO*

Modern governments are becoming increasingly complex and the eGov-DAO has to handle a large amount of workload. A good thing is that most of these works do not have a high requirement on latency. For example, in the process of bidding and selection, a latency in several days is still acceptable to have the final result. The performance of the eGov-DAO is heavily affected by the underlying blockchain infrastructure, and various methods have been developed to improve its throughput, latency, and scalability [5], [21], [22]. All these techniques can be applied to improve the performance of the eGov-DAO so that it will have the capability to process an increasing amount of work.

V. Conclusion

Today, e-government services, e.g., the contracting service, do not have a fully automated and efficient system that can both integrate all the participants and provide transparency. In this paper, we propose the blockchain-based government-DAO (eGov-DAO), which is the first system allowing real-time monitoring and analysis of an e-government service. The system provides transparency, accountability, immutability, and more importantly, a better national resource management to the service. This system reserves all records for auditing, thus limiting litigation between parties involved and increasing the speed of allocation and execution of contracts. Our design makes the system user-friendly, which requires minimum training for the users. Finally, we believe that the decentralize nature of the eGov-DAO makes it attractive to both public users and business community, given their huge amount of interest in blockchain technology recently.

For the last couple of years, both governmental and business services have been hacked several times, from ransomwares to denial-of-service attacks. The blockchain-based government-DAO definitely solves these security problems while still reduces costs of building and maintaining complex IT infrastructures. This solution helps a government to save unlimited amount of resources, manage more efficiently government business, and reduce the risk of giving contracts to companies that lack the capacity to fulfill them by implementing a transparent and secure e-government system with a minimum cost.

References

[1] V. Buterin *et al.*, "A next-generation smart contract and decentralized application platform," *white paper*, 2014.

[2] M. Del Castillo, "Ethereum executes blockchain hard fork to return dao funds," 2016.

[3] M. Duggan, "Does contracting out increase the efficiency of government programs? evidence from medicaid hmos," *Journal of Public Economics*, vol. 88, no. 12, pp. 2549–2572, 2004.

[4] T. A. Denes, "Do small business set-asides increase the cost of government contracting?" *Public Administration Review*, pp. 441–444, 1997.

[5] M. Vukolić, "The quest for scalable blockchain fabric: Proof-of-work vs. bft replication," in *International Workshop on Open Problems in Network Security*. Springer, 2015, pp. 112–125.

[6] S. King and S. Nadal, "Ppcoin: Peer-to-peer crypto-currency with proof-of-stake," *self-published paper, August*, vol. 19, 2012.

[7] N. Szabo, "Formalizing and securing relationships on public networks," *First Monday*, vol. 2, no. 9, 1997.

[8] M. Giancaspro, "Is a smart contractreally a smart idea? insights from a legal perspective," *Computer Law & Security Review*, 2017.

[9] W. Dilger, "Decentralized autonomous organization of the intelligent home according to the principle of the immune system," in *Systems, Man, and Cybernetics, 1997. Computational Cybernetics and Simulation., 1997 IEEE International Conference on*, vol. 1. IEEE, 1997, pp. 351–356.

[10] M. Swan, "Blockchain thinking: The brain as a dac (decentralized autonomous organization)," in *Texas Bitcoin Conference*, 2015, pp. 27–29.

[11] C. Jentzsch, "Decentralized autonomous organization to automate governance," *Online-Publikation: https://download. slock. it/public/DAO/WhitePaper. pdf.(Stand: 23.06. 2016)*, 2016.

[12] S. Barber, X. Boyen, E. Shi, and E. Uzun, "Bitter to betterhow to make bitcoin a better currency," in *International Conference on Financial Cryptography and Data Security*. Springer, 2012, pp. 399–414.

[13] D. Neumark, J. Zhang, and B. Wall, "Where the jobs are: Business dynamics and employment growth," *The Academy of Management Perspectives*, vol. 20, no. 4, pp. 79–94, 2006.

[14] A. Slagell, R. Bonilla, and W. Yurcik, "A survey of PKI components and scalability issues," in *25th IEEE International on Performance, Computing, and Communications Conference - IPCCC 2006*. IEEE, 2006, pp. 475 – 484.

[15] C. Adams and S. Lloyd, *Understanding PKI: Concepts, Standards, and Deployment Considerations*. Addison-Wesley Professional, 2003.

[16] R. M. McNab and F. Melese, "Implementing the gpra: Examining the prospects for performance budgeting in the federal government," *Public budgeting & finance*, vol. 23, no. 2, pp. 73–95, 2003.

[17] M. Bayati, D. Shah, and M. Sharma, "A simpler max-product maximum weight matching algorithm and the auction algorithm," in *Information Theory, 2006 IEEE International Symposium on*. IEEE, 2006, pp. 557–561.

[18] M. Bellare and B. Yee, "Forward-security in private-key cryptography," in *CT-RSA*, vol. 2612. Springer, 2003, pp. 1–18.

[19] F. Idelberger, G. Governatori, R. Riveret, and G. Sartor, "Evaluation of logic-based smart contracts for blockchain systems," in *International Symposium on Rules and Rule Markup Languages for the Semantic Web*. Springer, 2016, pp. 167–183.

[20] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008.

[21] I. Eyal, A. E. Gencer, E. G. Sirer, and R. Van Renesse, "Bitcoin-ng: A scalable blockchain protocol," in *13th USENIX Symposium on Networked Systems Design and Implementation (NSDI 16)*, 2016, pp. 45–59.

[22] Z. Gao, L. Xu, L. Chen, N. Shah, Y. Lu, and W. Shi, "Scalable blockchain based smart contract execution," in *Parallel and Distributed Systems (ICPADS), 2017 IEEE 18th International Conference on*. IEEE, 2017.