

Citizens Vote to Act: smart contracts for the management of water resources in smart cities.

Andrea Bracciali
University of Stirling
abb@cs.stir.ac.uk

Ioannis Chatzigiannakis
University of Rome “a Sapienza”
ichatz@diag.uniroma1.it

Andrea Vitaletti
University of Rome “a Sapienza”
andrea.vitaletti@diag.uniroma1.it

Abstract—Smart cities leverage Information and Communication Technologies (ICTs) to enhance the quality of urban services. However, it is nowadays clear that the success of a smart city largely depends on the level of engagement of its citizens. In this paper we explore to what extent disruptive blockchain technologies can be used to incentivise the democratic participation of citizen. The investigated approach extends the standard IoT cycle 1) sense data, 2) cloudify and elaborate them, and 3) push information to the users. Here, the user takes an active role by means of data-informed votes on policies, therefore influencing behaviours. We illustrate such an approach by means of a proof-of-concept decentralised application (dApp) supporting the negotiation of policies for the management of urban water resources. The dApp consists of a smart contract that manages the execution of other smart contracts (the policies) according to the data-driven choices of the community. This use case demonstrates how suitably blockchain technologies can support fair and safe access to data and user engagement in smart cities.

Index Terms—Smart cities, Blockchain technologies, IoT, dApps, Smart Contracts

I. INTRODUCTION

Smart cities integrates a variety of Information and Communication Technologies (ICTs) to enhance the quality and improve the performance of urban services with the main goal of reducing resource consumption, wastage and overall costs. Examples of smart cities services include parking and street maintenance, lighting, traffic control, energy and water management.

The growing availability of Internet of Thing (IoT) devices in the cities [5], i.e. embedded devices that can communicate and interact over the internet and can be remotely monitored and controlled, allows us to measure and possibly control a number of heterogeneous urban services and resources. Thanks to capillary measurements of a variety of phenomena is nowadays possible to decide actions that can significantly improve the quality of the life in the city. When executed collectively by the community of citizens, such actions can enable a new generation of eco-sustainable smart cities.

While IoT devices are in principle capable of both sensing and acting on the environment, in many real cases the engagement and collective participation of citizens is still needed to produce tangible benefits [22], [24].

A. Decentralized management of water resources

Most of the topics discussed in this paper may be applied to a variety of urban services and related problems. In this paper

we will focus on the management of urban water resources.

Several sources make water management data available. The interactive statistics portal by the International Water Association (IWA) [6] allows users to compare cities' differences regarding water consumption, tariff structure and regulation of water services. The portal includes data from all 5 continents from 39 countries and 198 cities. In the following we report some of the main conclusions driven in 2017 by the expert Jan Hammenecker, leader of the IWA Working Group Statistics:

- Household consumption of potable water varies from 28 to 631 litres per day per capita, a factor of 20.
- Water taxes that form part of the water bill vary between 0 and 28 percent of the total bill.
- No single tariff structure is trending worldwide. Hammenecker said that fixed charges versus variable charges and environmental charges or not, all have different advantages and disadvantages.
- Water consumption remains subsidised in many countries and cities.
- Water pricing is a useful tool for reducing water consumption in times of water scarcity.

The IWA report stresses the importance of adopting modern emerging technologies of smart metering based on IoT to accelerate the way to a smart water network. This is further confirmed by a report by the ITU-T Focus Group on Smart Sustainable Cities [16] that clarifies the importance of smart water management in the cities and identifies in the smart water meters the tools that can give to the citizens the detailed information they need to conserve water and reduce their bills. The employment of smart meters as part of IoT deployments to reduce the water consumption has been investigated in a number of papers, such as [7], [25].

Figure I-A shows a simplified reference architecture emphasizing the main building blocks of a typical IoT system for water management.

Data on water consumption is collected by smart meters and delivered to a cloud service allowing live monitoring of the consumption and providing evidences on consumption patterns to the users.

One of the goals of this framework is to provide evidence that may help citizens to better understand their consumption patterns and consequently act to reduce water consumption.

It is worth noting that the overall success of such resource management strongly depends on the active and collective par-

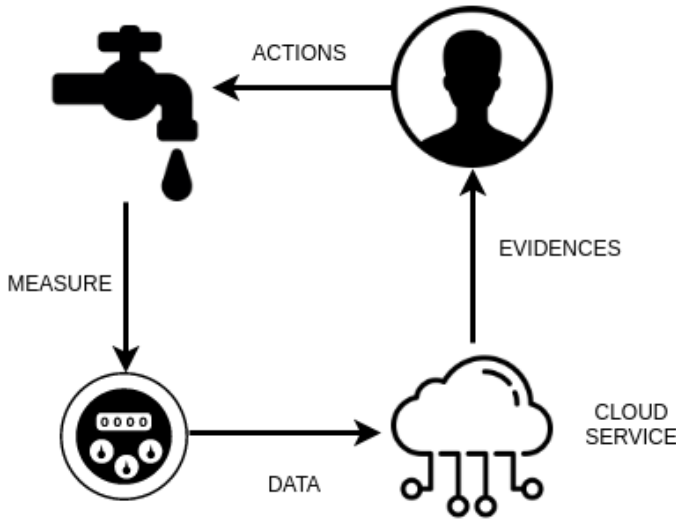


Fig. 1. A simplified reference architecture of a typical IoT system for water management. It is currently based on the centralized Client Server paradigm.

ticipation of citizens. The virtuous behaviour of an individual is commendable, but the risk is that it is literally “a drop in the ocean” if not accompanied by the joint action of the community.

Nowadays, the Client/Server paradigm dominates the cloud services market. However, blockchain technologies, an emerging Peer-to-Peer (P2P) *decentralised* framework, have the potential to revolutionise this market. After early stages of development, when blockchain technologies have been mainly employed in the financial domain, e.g. Bitcoin, nowadays there is a growing interest in exploring their applicability in other domains. Decentralized Applications (DApps)¹ are a new class of applications coded in programs running on the blockchain. DApps may provide a variety of services over the underlying P2P infrastructure that up to now have only been provided in Client/Server architectures.

The P2P nature of DApps and the lack of a central authority as in the Client/Server paradigm, is the key ingredient to implement infrastructures supporting new forms of democratic engagement of the citizens.

A recent report by Fluence Labs [13] presents the state of the DApps ecosystem surviving 160 projects. The main findings can be summarized in the following points:

- DApps is a modern trend: 72% of the projects started in 2018
- 87% of the projects run on Ethereum
- A quarter of the surveyed projects are gaming DApps.
- About half of the projects used a centralized tools to connect to the Ethereum blockchain.
- Transactional fees prevailed as the central monetization model for most projects.

¹In this paper we do not enter into the debate on the differences between DApp and DAO (Decentralized Autonomous Organizations).

- New user onboarding was mentioned by more than three quarters of the respondents as the major obstacle to adoption.

B. Contribution of the paper

In this paper we present a Proof-of-Concept (PoC) DApp supporting the active and collective participation of citizens to resource management, with the specific goal of reducing water consumption within a smart city context.

The goal of this paper is to demonstrate how the distinguishing features of DApps can actually foster that active and collective citizen engagement that is key to a real impact. We will do this by engineering the main building blocks of a PoC DApp that allows citizens to select policies and actions to be performed through a democratic voting process. Besides, engagement is also fostered by a “fair” incentive framework, naturally supported by DApps, which can suitably reward adherence to those behaviours that the community elects as most beneficial. Technically, we will discuss the design choices that lead to a decentralised management of *smart contracts* implementing water management *policies*. The decentralised nature of the architecture guarantees fairness to all citizens in the decisional process, without dominant positions or information imbalance.

To the best of our knowledge, the PoC here presented is an innovative exploitation of the benefits of blockchain technologies in the context of policy management and citizen engagement in smart cities.

II. INTRODUCING SMART CONTRACTS FOR POLICY MANAGEMENT

The reasons for the participation of citizens to a process can be complex and multi-facet, but at first glance, in a very simplistic model, there are at least three main reasons: a) *obligation*, when the participation is mandated by some rules or laws, e.g. taxes must be paid; b) *convenience*, when there is a tangible or intangible benefit in participating, e.g. lottery rewards and reputation in open source development; c) *motivational*, when people participate because embrace the ideals of an initiative, e.g. a strike.

Note that these motivations are not mutually exclusive; citizens pay taxes not only because it is mandatory (obligation), but also because they can take advantage from the services implemented thanks to their contributions (convenience).

Starting point is to assume the availability of suitable IoT devices, i.e. smart water meters, capable to measure the performance of a target process, i.e. reducing water consumption. However, we want then to have citizens engaged in pursuing behavioural changes. It is well-known that behavioural changes are the effective way to better performances in resource management, e.g. closing the water while brushing teeth can save up to 20 litres, and taking a shower can save up to four times the water necessary for a bath.

As observed in [22], smart cities often do not optimally reach their objectives if the citizens are not suitably involved in their design. The authors observe that the scientific literature

acknowledges the essential participative role of citizens in smart cities and argues that the notions of empowerment of citizens and “democratization” of innovation should be part of the definition of smart cities.

DApps running on blockchains have the potential to support new ways of democratic participation and citizen empowerment. Noticeably, blockchains introduce decentralised computation on a P2P network: the (honest) majority of the peers validates and guarantees the fairness and correctness of the results of the computation.² DApps are typically implemented in the form of *smart contracts*, i.e. computer programs that may encode agreements, policies, rules and penalties that can not be arbitrarily altered once agreed and run on the blockchain. Beyond automatically enforcing such obligations, smart contracts can also autonomously transfer digital assets between parties, e.g. rewards, as stipulated by the content of the contracts.

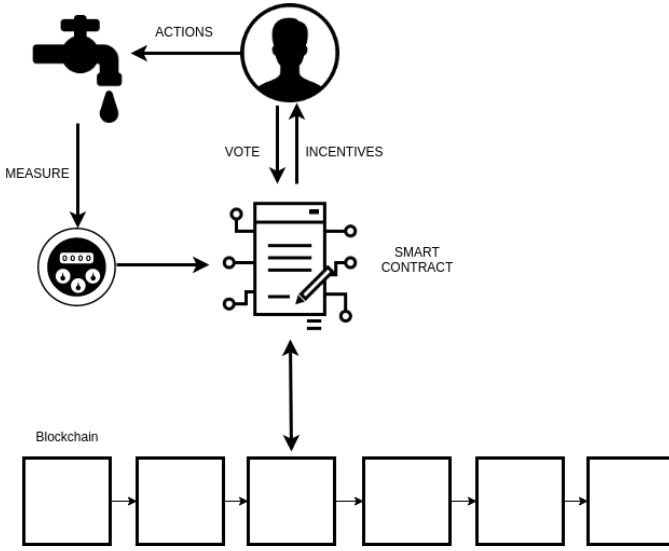


Fig. 2. A simplified picture of a DApp for water management. Contrary to the centralized architecture depicted in I-A the back-end of this architecture is implemented on smart contracts running on the P2P decentralized blockchain infrastructure.

To support the active engagement and democratic participation of users, our proposed DApp will basically implement two main principles:

- Citizens propose smart contracts that encode measures of the effectiveness of water management policies. Typically, such smart contracts relies on IoT data to monitor the application of policies. Citizens also select a smart contract by a fair vote. Such smart contracts, capable of attracting the greater consensus from citizens, becomes currently operative. Selection may occur regularly, on demand, or on specific conditions.
- To further encourage the participation of citizens, the operative smart contract will also be in charge of dis-

tributing incentives to virtuous citizens, namely citizens that most actively contribute to fulfil policies, i.e. successfully reduce the water consumption. Incentives come from treasury mechanisms, which are out of scope in this paper (one can imagine a suitably minimal fee). It is worth noting that when voting for the smart contracts, citizens actually also select an incentive framework.

We consider two types of smart contracts: the *proposal* and the *manager*.

A proposal smart contract must essentially define a) how to measure the contribution by each citizen to the reduction of water consumption as measured by IoT sensors, and b) how to distribute incentives according to that contribution. Citizens are free to present proposals, namely alternative solutions that are democratically voted by the citizens themselves.

The manager smart contract is in charge to a) manage the interface with IoT sensors and safety of data, b) manage the voting process at each occurrence of it, and c) make operational the most voted proposal, which will monitor the application of policies and devolve incentives accordingly. Several variations to this general scheme are possible, of course, but for the sake of this paper such a general formulation will be adequate. Several technical details about implementation are discussed in Section III

This form of democratic co-design of policies and rewards are meant to support the participative process initially discussed within the context of smart cities: policies, their success criteria and associated incentives are proposed and selected bottom-up and are not imposed top-down, thanks to the innovative features of decentralised computation, as supported by blockchain technologies. In the following we sketch two simple examples of smart contracts designed to encourage and support a sustainable use of the water. Note that both behaviours relies upon a measurable metric.

- **Smart Contract 1:** incentives for a user are based on water meter records of their own home.
- **Smart Contract 2:** incentives for a user are based on water meter records of their block within the city.

Smart Contract 1 aims to encourage a virtuous behaviour at the user level, while Smart Contract 2 aims to encourage virtuous behaviour within a the group of users living in the same block. One could imagine that the social implications of Smart Contract 2 could foster a more collaborative approach to the common goals. However, the dynamics of DApp induced social collaborations is scope for future work. Other contracts combining individual and group level incentives have been shown to work well for different goals, such as to increase physical activity [18].

III. IMPLEMENTING SMART CONTRACTS FOR POLICY MANAGEMENT

The whole process which allows the citizens to select the smart contract that will become operative can be divided in three phases: 1) the proposal phase, 2) the selection phase

²A technical overview of blockchains is out of scope here. The interested reader is referred to [27]).

and 3) the running phase. During the proposal phase, proposal contracts are submitted by the community. In the selection phase, proposal contracts are voted by the community among the proposed ones. For the sake of simplicity, we assume that citizen can access the vote only by providing a fee that is used to accumulate the incentives. More realistic fee policies are scope for future work. The voted contract becomes operative in the running phase, and will actually distribute the accumulated incentives according to its own policies.

The DApp, whose main components are discussed in the following, has been developed in Solidity [8], the object-oriented, high-level language for implementing smart contracts on Ethereum, using Remix [2], the browser-based compiler and IDE that enables users to build Ethereum contracts and to debug transactions.

We used some of the Solidity design and programming patterns collected by Franz Volland in his github repository [23].

A. Prove the correspondence between source code and byte code of the smart contracts implementing policies

A crucial feature of the proposed DApp is the ability of proving the correspondence between source code and byte code of the contracts. In other words, the community of citizens must be able to verify that the proposed smart contracts and the manager itself will actually behave (byte code) as supposed (source code).

There are nowadays off-chain services allowing to check such a correspondence [4]. In section V we will briefly introduce an interesting line of research aiming at performing this verification *on-chain*, i.e. without introducing the need of trusting centralized services.

B. The manager assumes the ownership of the contracts during the selection phase

During the proposal phase, users make contract proposals. In order to participate to the next selection phase, users have to release contract ownership to the manager smart contract. This is done by implementing the *Access Restriction* pattern which allows the ownership of a contract to be changed. The proposer invokes the *changeOwner* function (see listing 1), providing as input the address of the manager that consequently becomes the owner. We stress here that at each instant in time there is only one owner for a contract. Some functions of the contract may be invoked by the owner only.

The template code by Volland also supports the acquisition of contract ownership by anybody for 1 ether after a month has passed since the last change in ownership. We do not exploit this function that however shows the versatility of the proposed pattern in supporting heterogeneous criteria for changing the ownership of a contract.

Listing 1. The function to change the ownership of the contract. Note that only the current owner can change the ownership.

```
function changeOwner(address _newOwner)
    public onlyBy(owner)
{
```

```
    owner = _newOwner;
}
```

C. Handling the different states of a proposal and the corresponding functions

In each phase, a proposal contract can be in one of two possible states: selection and running. Only the owner of a contract can change the status of a contract.

The *State Machine* pattern [23] allows a contract to go through different states, with different functions enabled in different states. It implements the Access Restriction pattern described above (see listing 2). A function modifier checks if the contract stage is equal to the required stage before executing the called function. Note that the manager, becoming the owner of the contracts, is the only one capable to change the state of a contract during the selection phase.

Listing 2. Managing the states of a proposed contract.

```
enum Stages {
    Proposal,
    Selection,
    Run
}

Stages public stage = Stages.Proposal;

modifier atStage(Stages _stage) {
    require(stage == _stage);
    _;
}

function run_contract() public
    atStage(Stages.running) {
    // Here the contract exposes
    // all the functions necessary to
    // a) measure the contribution
    // of each citizen to the
    // reduction of water consumption
    // b) distribute the incentives
    // according to that contribution
}
```

Such a template allows more complex behaviour combining both stage transitions and timed transitions to be taken into consideration.

D. Voting for the contracts in the selection phase

During the selection phase, the manager implements a voting application. In this case, the Contract template provided by the the Solidity documentation has been used [3]. The main differences being:

- 1) voters not only express their preferences on the proposed contracts but must also provide a minimum amount of currency to sustain the incentives that will be distributed in the running phase;

- 2) the manager calls the *winningProposal* function after a given amount of time to conclude the selection phase;
- 3) the winning contract changes its status from selection to running;
- 4) the manager transfers all the accumulated funding to the winning contract

While the voting template would also support a form of liquid democracy, we do not consider here for the moment the possibility of voting with delegation.

E. Collecting data from the smart meters, i.e. interacting with the IoT

Once in a running state, the winning smart contract needs to collect data from the smart meters to correctly dispense incentives to the users. This requires the communication with an Oracle, a centralisation point, to gain access to data outside the blockchain. An Oracle is hence a trusted entity providing a unique view on a source of data considered credible.

Each node in the blockchain has to validate every computation performed in a smart contract. When this requires the interaction with off-chain sources of data, as in our case with smart meters, this becomes unpractical because, due to network issues (e.g. delays), there are not guarantee that all the node will access the same information as expected thus leading to a possible break in the consensus algorithm.

In our PoC, we use the oracle service provided by Oraclize [1], see listing 3 (recently Oraclize changed its name to Provable).

Listing 3. The call of an Oracle to acquire the water meter readings.

```
contract Water_Oracle is usingOraclize {

    string public WATER;

    function getWaterConsumption()
        public payable {
            if (oraclize_getWC("URL") >
                this.balance) {
                // Handle out of funds error
            } else {
                oraclize_query("URL", "API");
            }
        }

    function __callback(string result)
        public {
            require(msg.sender ==
                oraclize_cbAddress());
            WATER = result;
        }
}
```

The function `getWaterConsumption()` is sending out the query to the oracle. The payable modifier is necessary because the Oraclize service is not for free.

If the balance is sufficient, the query is sent to the Oraclize contract. The first parameter tells the oracle that we want to query a URL while the second parameter contains the URL of the API providing the data we are interested in.

The callback function is used by the oracle to respond providing the required data which are stored in the public variable `WATER`. The value of `WATER` is finally used to distribute the incentives.

IV. RELATED WORK

A privacy-friendly blockchain-based gaming platform aiming at engaging users in reducing water or energy consumption at their premises is proposed in [20]. In this paper users participation is achieved by gamification, the concept of smart contract is not considered and the blockchain is used to implement an authenticated storage capable to collect secure commitments by the meters and thus avoiding cheating.

The literature on blockchain technologies in the smart cities has been recently reviewed in [21]. The paper analyses a number of sectors where the blockchain can contribute to build a smarter city such as: governance and citizen engagement, education, culture, science and innovation, well-being, health and safety, economy, transportation, energy, built environment, natural environment, water management.

Our approach well fit into the governance and citizen engagement sector. This sector can be analyzed considering four governance models [15]: the most conservative is the *government of a smart city* in which the governance is implemented by traditional governmental structures. The *smart decision* model leverages big data to support the decisions, thus requiring some kind of restructuring at the decision process level, but not at the organizational level. The *smart administration* model adopts ICTs to interconnect information, processes, institutions, organizations and physical infrastructures to better serve citizens. Finally the most transformative *smart urban collaboration* entails the proactive participation of the citizens to the governance of the city. The authors stress that lack of transparency and trust on a centralized network infrastructure could be a key factor that hinders the true realization of the citizen participatory governance model. Our proposed DApp is an example of smart urban collaboration implemented over a P2P network thus overcoming most of the limits of traditional centralized networks and guaranteeing an unprecedented level of transparency and trust. In the blockchain, the trust shift from a single and centralized third party to the whole P2P infrastructure, that is decentralized in its nature.

E-governance models for the smart cities usually rely on cloud architectures (similar to the one represented in figure I-A) which are nowadays monopolized by a few companies serving as central and trusted authorities. Blockchain has the potential to change this situation, and it is thus considered a core technology to democratize and dis-intermediate the governance processes. However, despite such premises, practical employments of e-government processes truly based on blockchain solutions, such as [14], are still very limited.

The rules and procedures for the governance of blockchain-based systems can be implemented both “on-chain” and “off-chain” [19].

In on-chain governance, the rules and the processes are encoded directly into the underlying infrastructure of a blockchain-based system. In other words, “the code is the law” and all the interactions to carry out the governance of a system are determined by rules and processes encoded within the underlying blockchain code.

There are a number of projects (e.g. Decred, Dash, Bitshares, Steemit, and Cardano) that have integrated on-chain governance as a core functionality of their platform, not only to govern blockchain based system, but also to govern the blockchain itself mostly by allowing to change the existing protocol and parameters of the blockchain and to retroactively make changes to its state.

Off-chain governance comprises all other (i.e. non-on-chain) rules and decision-making processes that might affect the operations and the future development of blockchain-based systems.

Our proposed approach is an example of a blockchain-based application providing on-chain governance. However, we do not aim at changing the core rules and functions of the blockchain, rather we simply take advantage of the existing infrastructure, as it is, to govern a process by a smart contract.

In a traditional organization, a group of humans usually organized in a hierarchical structure, interacts in person and control the assets of the organization off-chain via the legal system. In a DAO (Decentralized Autonomous Organization), a group of people interacts according to a protocol specified in code and enforced on the decentralized blockchain. In this context, the code is the law and consequently the organization can autonomously run without the need of an interaction in person. The distinction between DApp and DAO is beyond the scope of this paper and while very controversial the two terms are used interchangeably sometimes.

However, the concept of DAO is often associated to the most dramatic event in the story of the decentralized applications running on the blockchain.

“The DAO” [10], [9] has been a concrete attempt to implement a funding platform, similar to Kickstarter, as a DAO running over Ethereum. The DAO went live in 2016 with between 10-20 thousand investors (estimation) providing the equivalent of about US\$ 250 million in funding and thus breaking all existing crowdfunding records. However, after few months an unintended behavior of the DAOs code was exploited draining the fund of millions of dollars worth of ETH tokens.

Immediately, the leaders of the Ethereum platform and other stakeholders reacted implementing “external, traditional and unforeseen in the code” defence strategies not implementable by the original code of The DAO, thus breaking the main principle behind a DAO, namely “code is law”. In the end, the whole project was abandoned dealing to a controversial “hard fork” the consequence of which are well summarized in Table 1 of [10]

Voting is considered among the most important application of the blockchain technology in the public sector [12]. In our proposed approach, voting is used to select which among the proposed contracts will become actually operative. Clearly, a fully aware vote in our proposed approach requires the understanding of smart contracts and their implications. It is worth recalling that in several cases, e.g. the infamous DAO [10], [9], buggy or even malicious smart contracts couldn’t be easily identified even by experts. To support the participation of non-experts, a form of liquid democracy can be implemented. Liquid democracy is a modern approach to voting in which voters can either vote directly or delegate their vote to other voters, who in turn can further delegate. [11], [26].

V. CONCLUSION AND FUTURE WORK

The success of smart cities initiatives relies on the collective and active participation of the citizens. Most of ongoing smart cities projects have exploited well ICT, and IoT in particular, to monitor a variety of urban services and process, but they do not have fully explored to what extent ICT can also support the necessary participation of citizens.

In this paper we discussed the possible advantages of deploying the smart city services as DApps running on decentralized blockchains. The main expected impact of such approach in terms of user participation can be summarized in the following points:

- Moving from a centralized Client/Server architecture, typical of current implementations of smart city service, to DApps will remove the necessity of trusting central authorities, which is considered one of the most relevant factors that limit the true realization of citizen participatory governance [15].
- The voting process implemented in the proposed DApp will contribute to democratize the policy management process. Such a democratization [22] will actually result in greater participation of citizens.
- The distribution of incentives collected by the community to reward the virtuous behaviour of the citizens can possibly further encourage their participation.

While the presented PoC demonstrates the technical feasibility of the proposed approach, there are still a number of issues to be considered for the possible deployment of a testbed.

In [13] it is clearly shown that the main problems of ongoing DApp initiatives is how to attract new users. Even if in that report a quarter of the surveyed projects were gaming DApps, a completely different domain with respect to our proposed one, a critical aspect is to demonstrate that indeed the points mentioned above will actually motivate a greater participation of the citizens.

In general, we should also demonstrate that the encoding in the smart contract of the policies and rewards is not only technically possible, but also suitable. What is the fraction of the population that can actually encode a proposal in a smart contract? What is the fraction of the population capable

to understand a smart contract? Those are two fundamental questions we should answer to better understand the possibly impact of the proposed approach. If the majority of the population will actually be (made) able to both formulate and understand policy proposals, this approach will possibly contribute to the realization of a new form of direct democracy. Otherwise, the risk is to build a technocracy in which the power is in the hands of the few that understand the technology. Anyway, as already observed, the liquid democracy can be the answer to delegate the vote to trusted, more educated parties, reaching in this way the right compromise between the direct and representative democracy.

From a more technical point of view, we plan to evaluate the gas usage of the function calls and the delay in the transaction confirmation, similarly to what has been done in [17]. However, we do not expect this aspect to be critical for a successfully deployment of the proposed approach if incentives will be appealing enough to motivate the participation of the citizens and, at the same time, allow for the necessary costs to run the infrastructure to be paid.

An interesting technical issue we want to explore is the implementation of an on-chain system capable to prove the correspondence between the source code of a smart contract and its byte code (see section III-A). Interestingly, most of the proposed approach to tackle this issue rely on off-chain services. The idea is to explore a more fundamentalist approach in which a smart contract can provide this service. At a first analysis, this should require the ability of a smart contract (the verifier) to compile another smart contract (the verified), a function that to the best of our knowledge is not yet supported by Solidity.

REFERENCES

- [1] The provable blockchain oracle for modern dapps. <https://provable.xyz/>, 2019. [Online; accessed May-2019].
- [2] Remix. <https://remix.ethereum.org>, 2019. [Online; accessed May-2019].
- [3] Solidity documentation. <https://solidity.readthedocs.io/>, 2019. [Online; accessed May-2019].
- [4] Verify & publish contract source code. <https://etherscan.io/verifyContract>, 2019. [Online; accessed May-2019].
- [5] H. Arasteh, V. Hosseini, V. Loia, A. Tommasetti, O. Troisi, M. Shafie-khah, and P. Siano. Iot-based smart cities: A survey. In *2016 IEEE 16th International Conference on Environment and Electrical Engineering (EEEIC)*, pages 1–6, June 2016.
- [6] International Water Association. Water statistics. <http://waterstatistics.iwa-network.org/>, 2019. [Online; accessed May-2019].
- [7] S. A. C. Tavares, R. J. B. V. M. Cavalcanti, D. R. C. Silva, M. B. Nogueira, and M. C. Rodrigues. Telemetry for domestic water consumption based on iot and open standards. In *2018 Workshop on Metrology for Industry 4.0 and IoT*, pages 1–6, April 2018.
- [8] Chris Dannen. *Introducing Ethereum and Solidity: Foundations of Cryptocurrency and Blockchain Programming for Beginners*. Apress, Berkeley, CA, USA, 1st edition, 2017.
- [9] Q. DuPont. Experiments in algorithmic governance a history and ethnography of “the dao”, a failed decentralized autonomous organization. In M. Campbell-Verduyn, editor, *Bitcoin and Beyond: Cryptocurrencies, Blockchains, and Global Governance*, chapter 8, pages 157–176. Routledge, 2017.
- [10] Mehar M. I., Shier C. L., Giambattista A., Gong E., Fletcher G., Sanay-hie R., Kim H. M., and M. Laskowski. Understanding a revolutionary and flawed grand experiment in blockchain: The dao attack. *Journal of Cases on Information Technology (JCIT)*, 21(1):19–32, 2019.
- [11] Anson Kahng, Simon Mackenzie, and Ariel D. Procaccia. Liquid democracy: An algorithmic perspective. In Sheila A. McIlraith and Kilian Q. Weinberger, editors, *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18)*, New Orleans, Louisiana, USA, February 2-7, 2018, pages 1095–1102. AAAI Press, 2018.
- [12] Nir Kshetri and Jeffrey Voas. Blockchain-enabled e-voting. *IEEE Software*, 35:95–99, 07 2018.
- [13] Fluence Labs. Dapp survey results 2019. <https://medium.com/fluence-network/dapp-survey-results-2019-a04373db6452>, 2019. [Online; accessed May-2019].
- [14] Maria-Llusa Marsal-Llacuna. Future living framework: Is blockchain the next enabling network? *Technological Forecasting and Social Change*, 128:226 – 234, 2018.
- [15] Albert Meijer and Manuel Pedro Rodriguez Bolvar. Governing the smart city: a review of the literature on smart urban governance. *International Review of Administrative Sciences*, 82(2):392–408, 2016.
- [16] ITU-T Focus Group on Smart Sustainable Cities. Smart water management in cities. https://www.itu.int/en/ITU-T/focusgroups/ssc/Documents/website/web-fg-ssc-0122-r7-smart_water_management_in_cities.docx, 2014. [Online; accessed May-2019].
- [17] G. Papadodimas, G. Palaiokrasas, A. Litke, and T. Varvarigou. Implementation of smart contracts for blockchain based iot applications. In *2018 9th International Conference on the Network of the Future (NOF)*, pages 60–67, Nov 2018.
- [18] Mitesh S. Patel, David A. Asch, Roy Rosin, Dylan S. Small, Scarlett L. Bellamy, Kimberly Eberbach, Karen J. Walters, Nancy Haff, Samantha M. Lee, Lisa Wesby, Karen Hoffer, David Shuttleworth, Devon H. Taylor, Victoria Hilbert, Jingsan Zhu, Lin Yang, Xingmei Wang, and Kevin G. Volpp. Individual versus team-based financial incentives to increase physical activity: A randomized, controlled trial. *Journal of General Internal Medicine*, 31(7):746–754, Jul 2016.
- [19] Wessel Reijers, Iris Wuisman, Morshed Mannan, Primavera De Filippi, Christopher Wray, Vienna Rae-Looi, Angela Cubillos Velez, and Liav Orgad. Now the code runs itself: On-chain and off-chain governance of blockchain technologies. *Topoi*, Dec 2018.
- [20] C. Rottondi and G. Verticale. A privacy-friendly gaming framework in smart electricity and water grids. *IEEE Access*, 5:14221–14233, 2017.
- [21] C. Shen and F. Pena-Mora. Blockchain for cities: a systematic literature review. *IEEE Access*, 6:76787–76819, 2018.
- [22] A. Simonofski, E. S. Asensio, J. D. Smedt, and M. Snoeck. Citizen participation in smart cities: Evaluation framework proposal. In *2017 IEEE 19th Conference on Business Informatics (CBI)*, volume 01, pages 227–236, July 2017.
- [23] Franz Volland. Solidity patterns. <https://fravoll.github.io/solidity-patterns/>, 2019. [Online; accessed May-2019].
- [24] T. Vcha, O. Pibyl, M. Lom, and M. Bacrov. Involving citizens in smart city projects: Systems engineering meets participation. In *2016 Smart Cities Symposium Prague (SCSP)*, pages 1–6, May 2016.
- [25] S. Wadekar, V. Vakare, R. Prajapati, S. Yadav, and V. Yadav. Smart water management using iot. In *2016 5th International Conference on Wireless Networks and Embedded Systems (WECON)*, pages 1–4, Oct 2016.
- [26] Bingsheng Zhang and Hong-Sheng Zhou. Brief announcement: State-ment voting and liquid democracy. In Elad Michael Schiller and Alexander A. Schwarzmann, editors, *Proceedings of the ACM Symposium on Principles of Distributed Computing, PODC 2017, Washington, DC, USA, July 25-27, 2017*, pages 359–361. ACM, 2017.
- [27] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang. An overview of blockchain technology: Architecture, consensus, and future trends. In *2017 IEEE International Congress on Big Data (BigData Congress)*, pages 557–564, June 2017.