2021 IEEE International Conference on Smart Data Services (SMDS)

An Elasticity Framework for Smart Contracts

(Invited Paper)

Schahram Dustdar
Distributed Systems Group
Technische Universität Wien
Vienna, Austria
Email: dustdar@dsg.tuwien.ac.at

José María García, Pablo Fernandez,
and Antonio Ruiz-Cortés
Smart Computer Systems Research and Engineering Lab (SCORE)
Research Institute of Computer Engineering (I3US)
Universidad de Sevilla
Seville, Spain
Email: {josemgarcia, pablofm, aruiz}@us.es

Abstract—**Smart contracts provide computation facilities to blockchains, enabling many application scenarios where automatic analysis and complex transactions can be performed. However, in situations where the flow of information is variable and there are time, quality, and/or cost constraints imposed, smart contracts do not usually offer enough performance. In this article, we describe an elasticity framework that enables dynamic adaptation of smart contracts with respect to used resources, analytics quality, and incurred costs in a transparent manner for both smart contract developers and users. We validate our framework in an Internet of Things and Smart City scenario where several analytics are computed using Elastic Smart Contracts that dynamically adapt the appropriate elasticity variables.**

1. Introduction

Since the introduction of smart contracts [1], blockchains have evolved from mere distributed digital ledgers to distributed computing platforms that can include not only an immutable data repository, but also logical and behavioural information to automatically rule the relationships between stakeholders. Thus, smart contracts can encode the functionality needed to provide additional services on top of the data registered in the blockchain. Precisely, the nature of the input data may affect the scalability and performance of smart contracts execution [2], [3], especially when applied to compute data analytics in complex scenarios such as the Internet of Things (IoT) and smart cities, which present a high variability in terms of information flow, time, quality and cost constraints [4]. To this concern, elasticity [5] is presented as the capabilities to react and accommodate changes in the environment with an autonomous mechanism.

In this work, we devise a reference architecture to create Elastic Smart Contracts (ESCs), thus providing an additional layer to existing smart contract facilities in order to consider elasticity constraints when designing and executing them. We also provide an implementation of our ESC Framework in Hyperledger Fabric [6], validating its applicability and performance within an IoT-based city analytics use case.

The rest of the article is structured as follows. First, we introduce a motivating scenario within the IoT domain in Sec. 2, followed by a discussion of related work in Sec. 3. Then, in Sec. 4 we introduce our proposed Elastic Smart Contracts Framework. Section 5 showcases the application of our proposal in the context of our motivating scenario. Finally, Sec. 6 concludes the paper.

2. Motivating Scenario

The need for transparency and non-tampering guarantees is a general requirement in a variety of cases. Amongst them, the smart cities paradigm represents an ideal context where we can find a number of specific scenarios. As an example, in this section we detail the case for a more intelligent traffic flow management that involves several sensors and actuators, and to develop different analytics that guide the decisions in an autonomous way. Specifically, as shown in Fig. 1, in this scenario we can identify different examples of multiple interleaved analytics: from low-level analytics regulating small physical spaces that mainly involve sensor data, to high-level analytics involving other kinds of data sources such as human actor decisions or off-chain data. For the sake of clarity, the motivational scenario represents a simple example that would correspond with two layers of analytics representing an adaptable urban traffic flow governance that predicts and avoids potential congestion risks, but this scenario could be evolved with additional layers to complete other challenges in the smart city domain.

Layer 1 (street) would represent a set of street sections composed by a number of sensors. Concretely, in Fig. 1 we can see a number of *detection sensors* that, for a given street section, provide their observations as data with different time resolution. The analytics developed by *Street ESC* would periodically perform an analysis over the sensors data to calculate a current traffic flow estimation in the section. This analytical information would be used to provide an aggregated information to the next layer. Furthermore, in a potential extended scenario, it could also be used for local decisions in the same layer, such as actuate into adaptable street lights in the street section that switch on in the

978-1-6654-0058-9/21/$31.00 ©2021 IEEE
DOI 10.1109/SMDS53860.2021.00039

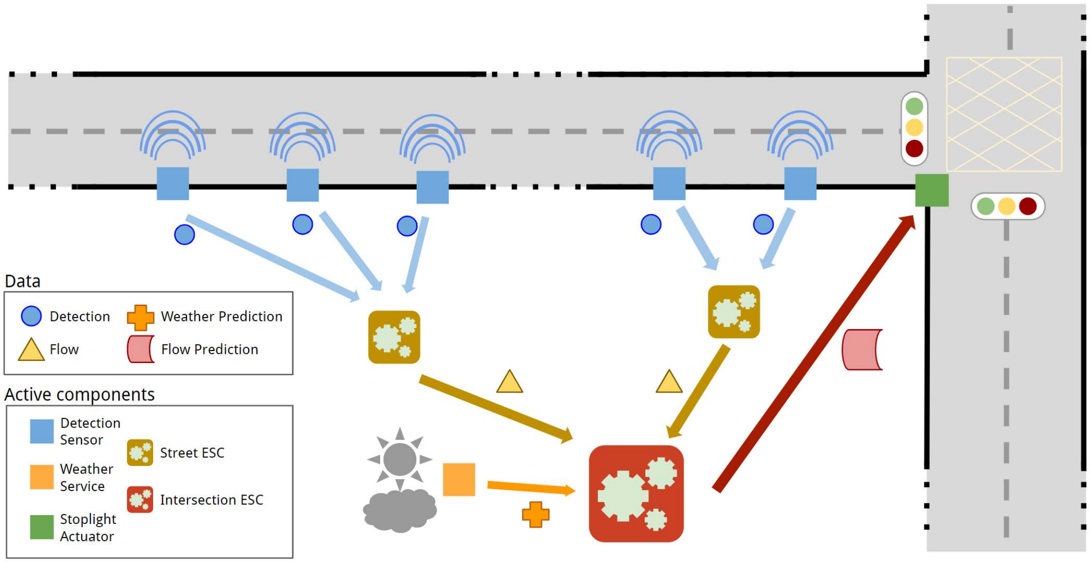

Figure 1. Motivating Scenario

presence of cars, so they dynamically adapt their switch-off latency to the actual estimations.

Layer 2 (intersection) would incorporate the presence prediction of different sections calculated in Layer 1 in order to create an estimation of the traffic flow in the whole street. In this layer, we could also include weather forecast as off-chain data so the *Intersection ESC* could calculate an estimation of the congestion risk in order to optimise the stoplight rules for the given intersection.

Furthermore, in a potential superior *Layer N* we could leverage advanced use cases such as a new generation contract for waste management service that regulates the actual resource assignment algorithm based on the data harvested by the sensors. This could be implemented by a combination of ESCs using the analytics gathered and calculating the actual bills automatically, providing a completely transparent and non-tamper management procedure. In such a context, examples of the three elasticity dimensions identified in [7] would emerge: (i) *Resources* range from the information providers that can correspond with things (e.g., sensors), software (e.g., government information systems) or people (e.g., an approval from a stakeholder); (ii) depending on the type of resource, a taxonomy of *quality* aspects can be defined (such as resolution data in sensors, availability of the government information system or readiness of the stakeholder); (iii) finally, *costs* involved in the process can also be structured in terms of the resource type (e.g., energy cost of the sensor, infrastructure costs of the information system, or personnel costs). All these concerns would be taken into account to create the elasticity policies for each ESC. As an example, in the use case, both ESCs could have a policy to constrain the maximum number of gas (an accounting mechanism in some blockchain networks such as Ethereum) used in the execution of the analytics.

3. Related Work

Blockchain technologies [8] are being applied in complex scenarios like the IoT-based one introduced before, where a variety of stakeholders have to collaborate, sharing information between them and allowing each party to carry out analysis and provide decentralized, trustless services over the shared data [9]. In [10] authors discuss a tailored blockchain solution for IoT-based ecosystems. More specific solutions focus on concrete use cases in smart cities [11] and smart vehicular networks [12], [13]. Our proposal is generalizable to other domains, although we validate it on an IoT-based scenario in Sec. 5.

In order to perform analytics like those described in Sec. 2 within a blockchain network [14], we have to consider the performance and scalability of the solutions [15], especially when we need to adapt to varying information flows and additional constraints, where elasticity [5] can provide capabilities to react and accommodate to these changes autonomously. In [7], authors provide a formal model of elasticity as a three-dimensional space involving resources, quality, and cost aspects that provide the appropriate framework to define and analyze the elasticity properties of an information system. In [16], authors discuss an approach to incorporate elasticity in smart contracts, evaluating its effectiveness to dynamically adapt the system performance. Our proposal builds on these previous works as the foundations of the generalised framework described in the following.

4. Elastic Smart Contracts Framework

Elastic Smart Contracts are the means to provide an appropriate analysis framework that is integrated into the blockchain taking into account resources, quality and cost considerations. Indeed, this elasticity need is grounded in the fact that analytics over a big set of data, performed by means of smart contracts in a blockchain, represent an actual challenge, as we discussed before.

We can define two different kinds of data used as input for analytics (i.e., by means of smart contracts) from the blockchain network. On the one hand, blockchains provide a persistent, immutable, and non-tampered way to store a set of transactions within the chain. On the other hand, digital assets stored in the blockchain can themselves provide current data (comprised of a dynamic set of values) that can be used for performing analytics as well. In any case, although the access and modification of concrete digital assets information or transactions is efficient, as the global size of those two main types of data increases, there could be a severe impact over the blockchain performance. Consequently, the performance implications of maintaining a big set of data imposes a trade-off over the appropriate size of data that should be maintained for analytics while keeping an adequate blockchain performance.

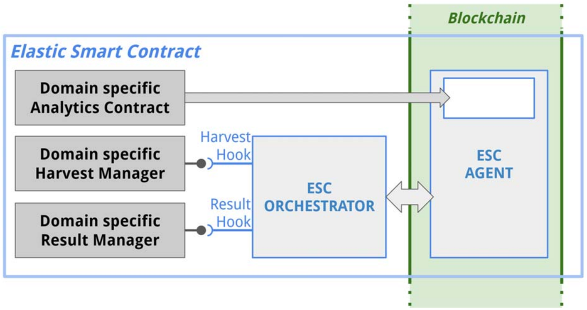

Figure 2. Elastic Smart Contract Framework outline

In order to support elasticity concerns for smart contracts execution and management, we devised the ESC Framework reference architecture outlined in Fig. 2. The two main components of our framework are the *ESC orchestrator* and the *ESC agent*, represented by light gray boxes in the figure. Our ESC framework provides reference implementations for these components, which offer concrete hooks (dark gray boxes in the figure) to define the domain-specific analytics to be performed.

The ESC orchestrator manages the execution of the blockchain system by monitoring the elasticity dimensions and calling the appropriate smart contracts according to the time and performance constraints of the particular scenario. These constraints are specified as elasticity rules which both the harvest and result manager have to provide, so they have to define the lower and upper bounds admissible for the duration of the analysis execution, as well as the initial values for the amount of resources to be used, among other parameters.

In turn, the ESC agent, which is deployed in the blockchain network, contains a set of generic smart contracts that are responsible for monitoring the elasticity properties, evaluating the performance of the domain-specific analytics smart contract that the client has to provide to the system. The ESC orchestrator will communicate with the ESC agent to perform the necessary on-chain computations.

For the autonomous execution of the ESC orchestrator, a client has to define two domain-specific components. On the one hand, the *Domain Specific Harvesting Manager* is responsible for managing the acquisition of input data (e.g., from a set of sensors or any data source in general) and updating the data stored in the blockchain that will be actually used to perform the appropriate analytics. This component must periodically obtain, according to the elasticity rules defined, a set of input data, and update the blockchain assets to store that data. In order to do so, the orchestrator submits a transaction to the ESC agent smart contract responsible for the data update, which registers the newly acquired data within the data asset that will serve as the input for the domain-specific analysis smart contract, while removing the old data according to the elasticity rules defined in the harvest manager.

In parallel to this process, the ESC orchestrator performs the elasticity operation function by evaluating the current status of the system with respect to the specified elasticity rules. Taking into consideration the average duration of the analysis execution, the elasticity operation periodically evaluates whether the elasticity parameters should be changed to improve the expected performance of the ESC. This evaluation is performed by the ESC agent via internal smart contracts, so that the evolution of the elasticity dimensions is registered and timestamped in the blockchain ledger.

On the other hand, the *Domain specific Result Manager* component focuses on executing the actual analytics of the specific scenario supported by blockchains, which has to be provided as the *Domain specific Analytics Contract* into the ESC agent. Thus, the result manager is responsible for executing periodically the analytics contract, which uses the data asset updated by the harvest manager as its input, and stores the analysis results in another result asset located within the ESC agent. The results from these executions are also collected by the ESC orchestrator, which aggregates both the results and the computed performance statistics, including the analysis duration used for the elasticity operation of the system.

This reference architecture can be applied to the complex scenarios discussed before. In the next section, we showcase an instantiation of the framework contextualised in the scenario described in Sec. 2.

5. Use Case Validation

The ESC framework has been implemented as an open source project[1] and it is based on Hyperledger Fabric as the underlying blockchain technology [6]. This technology, allows the execution of smart contracts (chaincodes in Fabric terminology) and the creation of permissioned chains that are specially interesting in scenarios such as the smart city domain. Moreover, it boosts a modular architecture that enables performance at scale while preserving privacy.

[1] https://github.com/isa-group/elastic-smart-contracts

In order to exemplify the usage of the framework, the project repository has a preconfigured showcase with the motivational scenario detailed in section 2. As shown in Fig. 3, our implementation corresponds to three different ESCs that are wired together to implement the two different layers of the scenario. Specifically, on the inferior layer, two different ESCs simulate the gathering of data from different sensors in a section (two sections in total) to calculate an accurate estimation of the traffic flow in the given section. Later on, in the upper layer, an intersection ESC harvests the estimated flows to aggregate them and integrates external weather predictions to generate the intersection stoplight parameters that would optimise the flow and reduce the congestion risk.

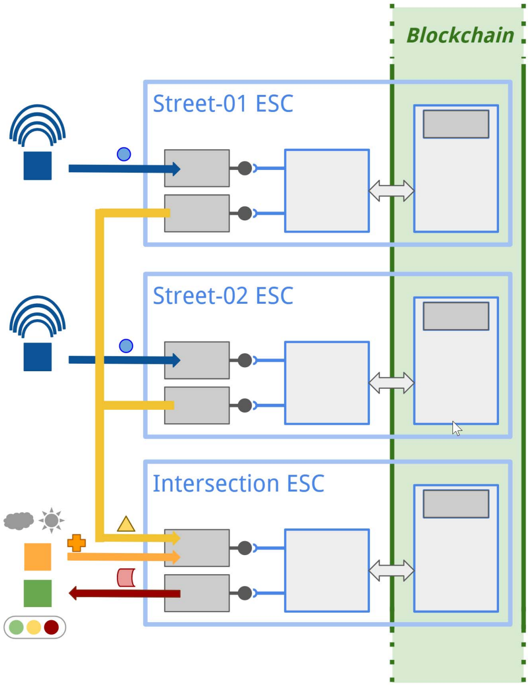

Figure 3. ESC applicability for the motivating scenario

From an implementation point of view, each ESC can be implemented by completing the provided template. Specifically, the developer needs to address four different aspects to obtain a fully fledged ESC: (i) harvest logic to provide data for the analytics; (ii) the result management logic to export the data calculated by the analysis; (iii) the analytics logic (that will be executed in-chain); and (iv) set the values to the different elasticity rules that will guide the orchestration. In previous works, we evaluated the performance of individual ESCs using an early version of our framework, showing its capability to dynamically adapt the elasticity parameters in a simplified scenario [16].

6. Conclusion

In this work, we define a framework to develop Elastic Smart Contracts which can be deployed and executed within blockchain networks. ESCs leverages elasticity properties of regular smart contracts, enabling their dynamic adaptation to changes in the amount of resources used, quality of the results, and the cost of their execution.

Our framework provides a set of components that orchestrate smart contract execution while monitoring that the elasticity parameters remain within defined boundaries. We validate our approach with an application to a concrete IoT-based city analytics domain, although it can be adapted to other scenarios that need to perform complex, real-time analytics using a blockchain network.

As future work, we plan to further evaluate our framework in realistic scenarios, using actual data from sensors and integrating analytical ESCs that obtain data from different blockchains and/or oracles. Furthermore, we want to extend the ESC Framework to also support generic elasticity rules that consider incurred costs, such as the gas used in Ethereum's smart contracts execution, so we also plan to adapt the implementation of the ESC framework components to the Ethereum platform.

Acknowledgments

This work has been partially supported by FEDER/Ministerio de Ciencia e Innovación – Agencia Estatal de Investigación under project HORATIO (RTI2018-101204-B-C21), by Junta de Andalucía under projects APOLO (US-1264651) and EKIPMENT-PLUS (P18-FR-2895), and by the TU Wien Research Cluster Smart CT.

Authors would like to thank Pablo García for his support on the prototype implementation.

References

[1] V. Buterin, “A Next-Generation Smart Contract and Decentralized Application Platform,” Ethereum.org, Tech. Rep., 2013. [Online]. Available: https://ethereum.org/en/whitepaper/

[2] S. Wang, Y. Yuan, X. Wang, J. Li, R. Qin, and F. Y. Wang, “An Overview of Smart Contract: Architecture, Applications, and Future Trends,” in *IEEE Intelligent Vehicles Symposium, Proceedings*, vol. 2018-June. Institute of Electrical and Electronics Engineers Inc., 10 2018, pp. 108–113.

[3] A. Aldweesh, M. Alharby, E. Solaiman, and A. Van Moorsel, “Performance Benchmarking of Smart Contracts to Assess Miner Incentives in Ethereum,” in *Proceedings - 2018 14th European Dependable Computing Conference, EDCC 2018*. Institute of Electrical and Electronics Engineers Inc., 11 2018, pp. 144–149.

[4] Q. Xu, K. M. M. Aung, Y. Zhu, and K. L. Yong, “A blockchain-based storage system for data analytics in the internet of things,” in *Studies in Computational Intelligence*. Springer Verlag, 2018, vol. 715, pp. 119–138.

[5] Y. Al-Dhuraibi, F. Paraiso, N. Djarallah, and P. Merle, “Elasticity in Cloud Computing: State of the Art and Research Challenges,” *IEEE Transactions on Services Computing*, vol. 11, no. 2, pp. 430–447, 2018.

[6] E. Androulaki, A. Barger, V. Bortnikov, S. Muralidharan, C. Cachin, K. Christidis, A. De Caro, D. Enyeart, C. Murthy, C. Ferris, G. Laventman, Y. Manevich, B. Nguyen, M. Sethi, G. Singh, K. Smith, A. Sorniotti, C. Stathakopoulou, M. Vukolić, S. W. Cocco, and J. Yellick, "Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains," in *Proceedings of the 13th EuroSys Conference, EuroSys 2018*. New York, NY, USA: Association for Computing Machinery, Inc, 4 2018, pp. 1–15.

[7] G. Copil, D. Moldovan, H. L. Truong, and S. Dustdar, "RSYBL: A framework for specifying and controlling cloud services elasticity," *ACM Transactions on Internet Technology*, vol. 16, no. 3, pp. 1–20, 8 2016.

[8] M. Swan, *Blockchain: Blueprint for a New Economy*. O'Reilly Media, 2015.

[9] A. Reyna, C. Martín, J. Chen, E. Soler, and M. Díaz, "On blockchain and its integration with IoT. Challenges and opportunities," *Future Generation Computer Systems*, vol. 88, pp. 173–190, 11 2018.

[10] A. Dorri, S. S. Kanhere, and R. Jurdak, "Towards an Optimized BlockChain for IoT," in *Proceedings of the Second International Conference on Internet-of-Things Design and Implementation*. New York, NY, USA: ACM, 4 2017, pp. 173–178.

[11] S. Ibba, A. Pinna, M. Seu, and F. E. Pani, "CitySense: Blockchain-oriented Smart Cities," in *XP '17: Proceedings of the XP2017 Scientific Workshops*, vol. Part F1299. ACM, 5 2017, pp. 1–5.

[12] Z. Yang, K. Yang, L. Lei, K. Zheng, and V. C. M. Leung, "Blockchain-Based Decentralized Trust Management in Vehicular Networks," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1495–1505, 4 2019.

[13] T. Jiang, H. Fang, and H. Wang, "Blockchain-based internet of vehicles: Distributed network architecture and performance analysis," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4640–4649, 2019.

[14] H. T. Vo, M. Mohania, D. Verma, and L. Mehedy, "Blockchain-powered big data analytics platform," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11297 LNCS. Springer Verlag, 12 2018, pp. 15–32.

[15] M. Scherer, "Performance and Scalability of Blockchain Networks and Smart Contracts," Umeå University, Tech. Rep., 2017.

[16] S. Dustdar, P. Fernandez, J. M. García, and A. Ruiz-Cortés, "Elastic Smart Contracts in Blockchains," *IEEE/CAA Journal of Automatica Sinica*, p. in press, 2021.