FRI Tokens for a Fog Computing Course

Sandi Gec[1], Petar Kochovski[1], Janez Brežnik[1], Vlado Stankovski[1]

[1]Faculty of Computer and Information Science, University of Ljubljana, Ljubljana, Slovenia
E-mail: sandi.gec@fri.uni-lj.si

Abstract. *The DECENTER project (2018-2021) produced a Fog Computing and Brokerage Platform that incorporates methods of Internet of Things, Cloud computing, Artificial Intelligence, Digital Twins and Blockchain. The main goal of the present work is to achieve knowledge transfer from the achievements of the DECENTER project towards the design of a new study course at the Faculty of Computer and Information Science in Ljubljana which is entitled Fog Computing for Smart Services. The course focuses on current DevOps software engineering practices which require the use of several enabling technologies which follow the general principle of decentralisation. This paper presents the design of a FRI token that is used in the course in order to illustrate the operation of advanced software ecosystems.*

1 Introduction

Edge and Fog Computing are emerging new paradigms that address the requirements for the management of AI methods closer to the data and/or sensor sources. This can be established by applying moden engineering principles like DevOps, and technologies that involve containerization. AI methods can be easily deployed into containers, moved around the network to the edge and cloud, and easily spinned on and off, so that one can use intelligence in the specific smart environment where it is needed. In fact, this is the software engineering agenda for the development of future smart services.

Moreover, the economy of decentralization is an ecosystem represented by different user entities or actors. These actors contribute their available physical and software resources in order to provide specific functional components to complex decentralized systems. The resources should follow the minimum requirements such as high level Quality of Service (QoS) and Quality of Experience (QoE) [1].

In this work we focus on the economy of decentralized cloud-based systems and applications, particularly on the design of specific tokens that can be used to support specific Smart Contracts for cloud based applications. We investigate the tokenomics life cycle: (i) token generation, (ii) use case analysis with potential token usage integration and, (iii) development of specific Smart contracts for DECENTER platform.

The rest of the paper proceeds as follows. Section 2 presents the baseline technologies of our work. Section 3 describes the implementation details of Fog Computing for Smart Services in the context of the Fog Computing course at the Faculty of Computer and Information Science in Ljubljana. Section 4 provides details from the experimental evaluation. Section 5 concludes the paper and describes plans on future research.

2 Background

The emergence of distributed and decentralised systems alongside with the necessity for better transparency, traceability, autonomy, privacy and security, has led towards the rise of the blockchain technologies. Blockchain as a concept, delivers a plethora of solutions that replace the centralised institutions (i.e. middlemen) with distributed and decentralised networks, where activities are executed and verified by reaching a mutual consensus among the network participants. In particular, blockchain is an immutable ledger of records, where records are grouped into blocks after a validation consensus is reached across the network's participants.

In general, blockchain technology became popular in 2009, when Satoshi Nakamoto released the Bitcoin whitepaper [2]. This baseline technology allowed digital assets to be exchanged among different entities (i.e. monetary transactions) and also allowed new blockchain cryptocurrencies to be developed and distributed. However, the capabilities of this technology soon was recognized by other domains, such as: mobility, supply management, healthcare, data management, decentralised finances and many more, which required their business logic to be moved to the distributed ledgers. As a result, Vitalik et al. [3] presented a revolutionary concept with the introduction of Turing complete smart contracts that are similar to general (notary) contracts with limited, but sufficient functionalities to cover a wide range of use-cases.

To exploit the full potential of the blockchain technology, it is necessary to carefully plan and manage the ecosystem market mechanisms and distribution of tokens. In the following sections, a more detailed overview of the market mechanisms within the DECENTER and ONTOCHAIN [4] project and the token management will be provided.

3 Fog Computing for Smart Services

In this section we first present the design of the course Fog Computing for Smart Services which aims at knowledge transfer from the DECENTER project towards the computer and information science students at the Faculty of Computer and Information Science in Ljubljana. For the purposes of the course a set of 200 ERC-20 tokens was generated that allowed students to learn through practice the token's market mechanisms and implementation in use cases for Fog applications.

3.1 Course design

The increased complexity of the emerging information and computing technology systems poses several challenges at software level. Users require seamless connectivity, abundant computing power and unlimited access to data independently of the underlying infrastructure. Increased levels of adaptability are becoming more and more essential in modern software systems in order to manage the needs of highly complex and dynamic environments requiring continuous development and operation (DevOps). Increasing trust and reliability while keeping system performance and reducing energy consumption is a complex task. This opens several research questions that address the unprecedented complexity in software systems within a shortened software lifecycle: from requirements analysis and design, to development and testing and up to deployment and operations across highly heterogeneous and dynamic self-adaptive systems. Hence, the goal of the Fog Computing course is to introduce the overall software engineering process for the development of smart applications that will run in the Fog, which is in the middle between the smart devices and the high-power data centers [5]. Nowadays, it is common for Fog applications to employ Internet of Things (IoT), Artificial Intelligence (AI), cloud computing and blockchain technologies. The course focuses on the theory and practice when developing complex, component-based, compute, memory and network-intensive multi-tier cloud applications.

In the beginning of the course, the main concepts, vision and objectives of Fog computing were introduced. Particular attention was given to the potential use cases, for example, those of Slovenia's smart specialisation programme that pose significant requirements for the integration of IoT, AI, cloud and blockchain based services and applications. While the students contemplate the initial set of lectures, their initial laboratory work is focused on trying out a set of technologies required to build a smart application, such as trying out Machine Learning (ML) models. After the initial stage, the course focuses on the DevOps life-cycle with focus on Fog computing applications. The students then set-up and try different software engineering tools and discussed various approaches to orchestration. Further to this, approaches for monitoring applications that may spread across multiple computing tiers are discussed.

Once being introduced to the foundations of Edge-to-Cloud computing, the students focus on multi-tier (edge-cloud) application designs with the purpose of addressing the non-functional requirements of the application.

The following three weeks are devoted to developing and understanding the plethora of AI methods that could be quickly containerised and delivered at the network Edge, the taxonomy of Big Data methods with focus on QoS aspects, and benchmarking methodologies.

The final part of the course is devoted to blockchain-related concepts, approaches, methodologies and technologies (i.e. coins and tokens, consensus mechanisms, Smart Contracts, Smart Oracles) that form part of the Fog computing vision. At a practical level, the students attempted to monetise the use of the cloudified ML application by developing and deploying Smart Contracts on Ethereum testnet node.

As a result of these activities, the students are able to gain software engineering competencies at the cross-section of the above mentioned technology types, whereas passing the course meant an improved understanding of the use cases, the functional and non-functional requirements, the current approaches to addressing various dependability, trust-related concerns, and the major technologies that are commonly used in order to address the whole spectrum of requirements, including orchestration, and the integration process.

3.2 Tokenomics

Many blockchain networks (e.g. Bitcoin, Litecoin, Ethereum, Cardano) allow using cryptocurrencies as native assets that can be used as medium of exchange and store value that entitles the participants in the system of ownership to an underlying property. In comparison, other blockchain-based solutions (e.g. Chainlink, Polygon, EOS) introduce tokens as bearer instruments. The main difference is that cryptocurrencies are the native digital asset of a specific blockchain protocol, whereas tokens are created on top of those existing blockchains.

Generally, every token consists of a specific purpose. Therefore, in the initial phase of token creation concepts of tokenomics need to be considered — the topic of understanding the supply and demand characteristics of cryptocurrency. While defining the token quantitative properties, we need to answer the following questions:

- How many coins or tokens currently exist?
- How many will exist in the future and when will they be created?
- Who owns the coins?
- Is there any information to suggest that a large number of coins has been lost, burned, deleted or are somehow unusable?

Apart from the basic attributes and functionalities derived by the answers to the questions above, additional definitions in terms of attributes, functions and events should be implemented in the Smart contract by definition to fulfill the basic token requirements.

By definition Ethereum ERC-20 tokens must implement the following attributes accessed as getter functions such as *`totalSupply()`*, *`balanceOf( address account)`*, *`allowance( address owner, address spender)`*, *`transfer(`*

`address recipient, uint256 amount)`, *`approve( address spender, uint256 amount)`* and *`transferFrom( address sender, address recipient, uint256 amount)`*.

In this paper, we focus on Smart Contract supportive token standards on Ethereum known as ERC-20, which has emerged as the technical standard used for all smart contracts on the Ethereum blockchain for fungible token implementations. ERC-20 defines a common list of rules (e.g. attributes, functions and events) that all generated tokens should adhere to. This minimum requirement ERC-20 interface simplifies and eases developers’ tasks, because they can proceed with their work and are able to implement additional Smart Contract standard concepts such as OpenZeppelin [6]. For example, there are utility Smart contracts that provide mathematical functions, voting mechanisms, governance and many others.

In the context of our course we generated a basic ERC-20 token on an Ethereum testnet network Rinkeby available on the contract address 0xad9c8654f0eaa7c83a08f1097520d5d9acd4aa3f, named *Frik* with the symbol *FRI* of max supply 200 tokens divisible by 18 decimals. The purpose of the token is to promote the course to the potentially interested students through the specimen with a structure depicted on Figure 1. Each specimen contains a unique Ethereum public address available as a QR code in the left circle of the specimen. The FRI token can be claimed through the Web page of the course where the potential students claim the token from the owner address 0x09901196813E19f282221A55f149bb74312B018c and obtain the private key to gain access to the public wallet. Even though the provision of a private key through a Web page is not a best practice, there is a recommendation that the students resend the token to his own Ethereum (testnet) address.

When the students owning the FRI token enrol in the course she or he can use it when developing practical use cases. The FRI token can be used within the developed Smart Contracts and Web pages based on the Web3 library able to interact with the Ethereum network through the Ethereum tool Metamask [7]. Students that want to know Ethereum Smart Contracts more in detail are motivated to follow practical Coursera courses [8] or many other available online tutorials or courses. An overview of potential use cases is presented below.

3.3 Use cases

This section describes four use cases that have been actively researched and discussed in the duration of the course. These use cases have potential to be integrated with Smart contracts in order to benefit the implicit properties of blockchain technology, enhance the trust aspect and generally improve the business models.

The first use case that covers the pedestrian domain is Smart City Crossing Safety with the objective to increase the pedestrian crossing safety by leveraging the IoT and edge infrastructures of a smart city. It focuses on equipping a number of pedestrian crossings with devices existing on the market that enable monitoring of pedestrians intending to cross the road and producing safety notifications.

The second use case focuses on automated logistics using mobile robots, allowing robots to obtain data from the environment. Applying new AI models to the robots requires high computational capabilities, so in the real world most of them could not be applied due to the robot’s computational limitations.

The third use case covers the construction site domain that requires constant information support to assure a safe construction environment. As a result, organising, monitoring, and implementing a construction project including its various safety, security, logistics, inspection and other aspects can be very challenging.

The fourth use case covers the working office domain, particularly the ambient intelligence of the office environment. Nowadays a lot of IoT devices have been deployed and used for real-time safety monitoring but the current level of indoor safety services is limited to monitoring status using devices such as cameras and cell phones. Therefore, it is challenging to respond promptly and appropriately to various situations without continuous supervision.

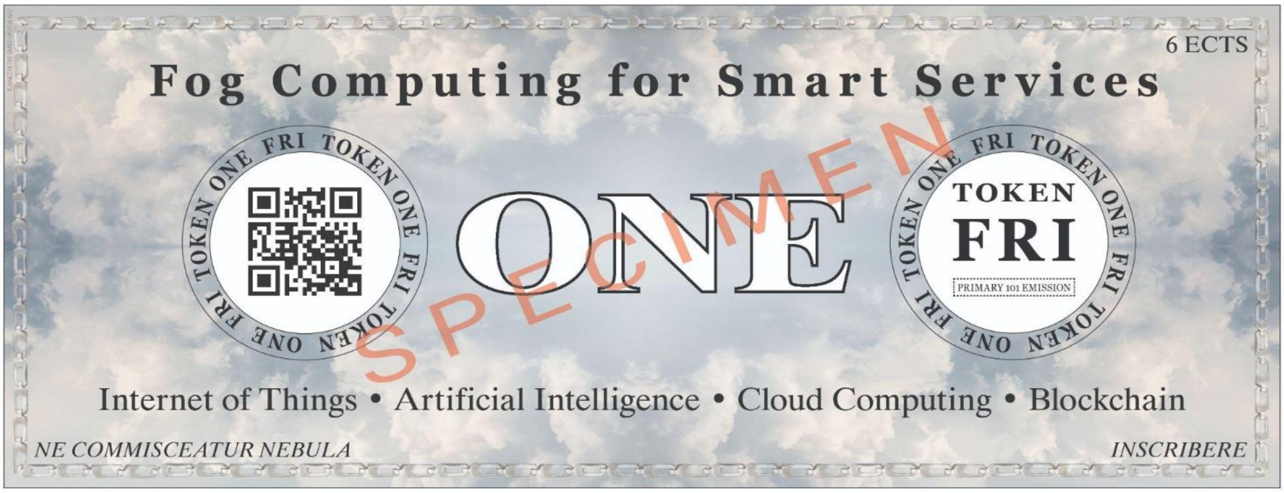
6 ECTS
Fog Computing for Smart Services
ONE FRI TOKEN ONE FRI TOKEN ONE FRI TOKEN
ONE
SPECIMEN
ONE FRI TOKEN ONE FRI TOKEN ONE FRI TOKEN
TOKEN
FRI
PRIMARY 101 EMISSION
Internet of Things • Artificial Intelligence • Cloud Computing • Blockchain
NE COMMISCEATUR NEBULA
INSCRIBERE

Figure 1: Sample structure of the specimen for the course Fog Computing for Smart Services available on Ethereum testnet network Rinkeby.

Often, the services of such use cases are hosted on cloud platforms, which can cause problems due to round trip delays. Privacy issues may also arise while uploading personal data such as images to the public cloud for analysis. All these challenges were carefully analysed and partially resolved from the know-how at the Fog computing for Smart Services course.

4 Experimental study

In this section we present an example of practical usage of a generalized Smart contract in the use cases presented in the previous section. The smart contract is deployed and empirically analyzed in order to prove the general sustainability that relies on the cost estimation execution of the functions in Ethereum testnet network represented in GWEI metric (1 ETH = 10^9 GWEI). The Figure 2 shows the cost of the main operations from the deployment and management of Cloud providers through a Smart contract where each provider interact through Ethereum wallet.

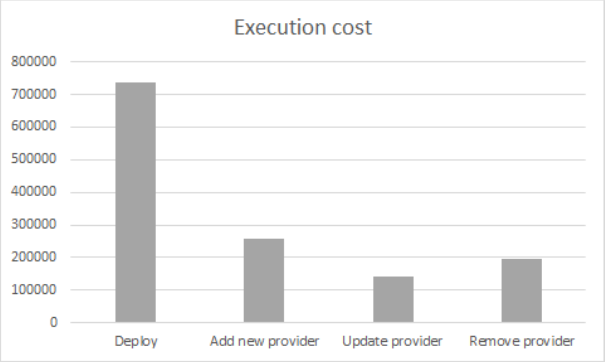

Figure 2: Execution cost of four main Smart contract operations: (i) deployment of a Smart contract, (ii) registering a new provider, (iii) updating a provider and, (iv) removing a provider.

The benefit of such an approach is to offer a transparent overview of the Cloud providers ecosystem to all involved entities. The results run on Rinkeby testnet, reproduced 10 times and averaged, indicate that the most costly operation is deployment of the Smart contract since it extends complex libraries (e.g. ChainlinkClient [9]). Other functions that are executed on demand have a significantly lower cost thus the Smart contract is usable in practice.

5 Discussion and conclusion

Novel technologies, such as: Internet of Things, Artificial Intelligence, cloud, Digital Twins and blockchain are currently converging, which results with numerous new smart services and applications. They are the substrate of Fog computing – an architecture that uses edge devices to carry out computation, storage, communication locally and thus addresses the requirements of Big Data pipelines, which start at the edge of the computing network and may run across multiple computing tiers up to various public or private Cloud providers.

The motivation of the present paper is to present how these technologies can be integrated into the study course at the faculty. It covered not only theory, but also a lot of practical assignments through direct hands-on experience, such as: realising compute, memory and network-intensive smart services and applications through the use of Fog computing technologies.

Based on the implementation and the presented evaluation, we can conclude that the quadrumvirate of technologies can be brought closer to the students in order for them to become up to date with the latest knowledge requirements of the IT industry. We realized that this raised high interest among the students, with the increased diploma themes covering Blockchain technology, and we are sure that such a subject should be part of the standard study programme at the higher educational institutions in Slovenia.

Acknowledgments

The research and development reported in this paper have received funding from the European Union's Horizon 2020 Research and Innovation Programme under grant agreements no.815141 (DECENTER: Decentralised technologies for orchestrated Cloud-to-Edge intelligence) and grant agreement no. 957338 (ONTOCHAIN: Trusted, traceable and transparent ontological knowledge on blockchain).

Literature

[1] M. Varela, L. Skorin-Kapov, T. Ebrahimi. *"Quality of Service vs. Quality of Experience"*, 2014.

[2] S. Nakamoto: Bitcoin: A peer-to-peer electronic cash system. *Decentralized Business Review*, 2008.

[3] V. Buterin, Ethereum White Paper, September 30, 2015. [Online]Available: https://ethereum.org/en/whitepaper/

[4] ONTOCHAIN. [Online]. Available: https://ontochain.ngi.eu/. [Accessed: 13-Aug-2021].

[5] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, S. "Fog computing and its role in the internet of things" in *Proceedings of the first edition of the MCC workshop on Mobile cloud computing, 2012*

[6] OpenZeppelin. [Online]. Available: https://openzeppelin.com/. [Accessed: 29-Jul-2021].

[7] Metamask. [Online] Available: https://metamask.io/. [Accessed: 29-Jul-2021]

[8] Coursera course Smart Contracts [Online]. Available: https://www.coursera.org/learn/smarter-contracts. [Accessed: 6-Sept-2021]

[9] ChainLink. [Online]. Available: https://chain.link/ [Accessed: 25-Jul-2021]