



Design of a Sustainable Blockchain-Oriented Software for Building Workers Management

Andrea Pinna*, Gavina Baralla, Giorgia Lallai, Michele Marchesi and Roberto Tonelli

Department of Mathematics and Computer Science, University of Cagliari, Cagliari, Italy

OPEN ACCESS

Edited by:

Rebecca Jing Yang,
RMIT University, Australia

Reviewed by:

Nadia C. Fabrizio,
CEFRIL, Italy
George Lawrence Sanders,
University at Buffalo, United States

*Correspondence:

Andrea Pinna
a.pinna@diee.unica.it

Specialty section:

This article was submitted to
Smart Contracts,
a section of the journal
Frontiers in Blockchain

Received: 12 March 2020

Accepted: 22 July 2020

Published: 19 October 2020

Citation:

Pinna A, Baralla G, Lallai G,
Marchesi M and Tonelli R (2020)
Design of a Sustainable
Blockchain-Oriented Software for
Building Workers Management.
Front. Blockchain 3:38.
doi: 10.3389/fbloc.2020.00038

The building workers sector is one of the most challenging sectors for Human Resources (HR) management. In this work, we propose a solution relying on Blockchain technology and present the design of a Blockchain-Oriented Software system conceived for managing the building workers sector with a focus on workers' safety, and it is guided by sustainable and Agile Methodologies in software design. The proposed approach takes advantage of different features of the Blockchain technology and provides transparency for labor inspectors, grants data integrity and immutability, relies on tamper-proof time stamps for any recorded activity, allows the implementation of Smart Contracts where clauses are automatically respected without the need of a trusted control authority, acknowledges the legal requirements in the field, including the possibility of creating an Operational Safety Plans, which construction companies have to provide, and finally implements the creation of vacant job positions that workers can find and apply to. In order to achieve these goals, we adopt the Blockchain-Oriented Software Engineering (BOSE) methodology to design Blockchain software applications and apply an Agile methodology centered on Blockchain Software development (called ABCDE) for the design and development of the decentralized application. Such a methodology allows us to center the software development around the actors of the system in the specific domain, such as Building Workers, Construction Companies, Labor Inspectors, and so on. In addition, we rely on the software sustainability analysis, based on the five dimensions of sustainability, to evaluate the approach and to avoid mistakes in the system development. We design system elements with specific diagrams, and we divided our system in the on-chain and the out-of-chain components. The implementation of the system, done by using Ethereum and the ERC721 standard, allows us to improve some aspect of the design, to know the deployment and usage costs, and to evaluate the effect of the user interface. Finally, we discuss the effects of our system and its sustainability, and we provide a comparison of our system with a similar per aims but centralized system.

Keywords: blockchain, constructions, BOSE, ABCDE, software engineering, smart contract, sustainability, sustainable design

1. INTRODUCTION

Construction sites are typical workplaces in which it is possible to find temporary workers. The construction companies often need to hire workers in addition to their long-term workers, both to increase the construction speed and to acquire a specialized workforce. They can also charge external companies to carry out a part of the project (i.e., plants or the like). Building constructions are activities with a time limit: the construction companies and the owners of the site must communicate the start and end dates of the works according to the current law. In general, worker safety is mandatory in every work context and especially in risky places like construction sites, and specific govern agencies focus the attention to this point, such as the European Union information agency for occupational safety and health (osha.europa.eu). Within the limits of EU regulations, each country adopts its own set of specific rules to match the country's needs and local reality constraints, such as building habits, styles, and materials. Using Italy as a case study, we demonstrate in this paper the flexibility and adaptability of our approach where construction companies have to draw up the "Operational Safety Plan" for each external site. In the plan, it is mandatory to describe every job assigned, each work activity, and a comprehensive list of safety features of the construction site. However, the need to reduce time and to limit costs could lead construction companies to save money at the expenses of workers, particularly with regards to the regularity of the work contracts and the compliance to safety laws. For instance, workers in the construction sector can be subject to the illegal practice that sees the employer regularizing the construction workers with a contract of employment different from the correct one but more advantageous to him. This practice, named "contractual dumping," affects both the worker's remuneration and the respect of workers' rights. On the other hand, construction companies often need a trusted and qualified workforce with all documents in order, especially if they offer temporary work contracts. Public bodies like the *labor inspectorate* or law enforcement inspections are the current countermeasures to the problem of violation of work laws, as described by Caruso (2013).

In general, we are faced with a social and economic system in which the various parties (site owner, employers, workers, and inspection bodies) need one another but where each part simultaneously aims at different and somehow conflicting objectives. In such a system, the creation of paradoxes is possible, such as when people in economic difficulty become more vulnerable to the exploitation of work or vice versa, such as when employers are vulnerable to false statements about candidates' skills and abilities. This set them at odds with the labor inspectorate that aims to protect them. It is clear that such a situation is not sustainable. In this context, we refer to the term sustainability meaning *systemic sustainability* in which the aim is to make it possible for the system to continue indefinitely on the base of the actual resources and not borrowing them from the future. System resources include not only natural and economical resources but also include social resources as well, where human relationships have a fundamental role. One of the main factors for creating sustainable human relationships is communication

(Genç, 2017). Trust and clear communication between actors, such as a clear and correct set up of the employment relationship, avoids misunderstandings and makes it easier to carry out the goals. To achieve this requirement, we rely on blockchain technology. It has been specifically designed to create trustworthy and transparent communication channel between parties who may be in conflict with each other.

The purpose of this work is to present the design and implementation study of a sustainable blockchain-oriented software system to manage temporary workers in the construction sector. The paper presents a reproducible development method for the realization of a specific blockchain-based decentralized Application (dApp) in full awareness of sustainability issues and in the full use of the practices of Software Engineering for blockchain-oriented software, BOSE (Porru et al., 2017). In particular, the method inherits the recent software development methodology "ABCDE: Agile BlockChain DApp Engineering" (Marchesi et al., 2018), conceived to drive dApp development from the definition of system goals to the implementation of the system user interface. In addition, in this paper we present the application of a software sustainability analysis. It allowed us to acquire awareness on the effects of a large-scale adoption of a decentralized hiring system in the building sector. The analysis examines the software system in the five dimensions of sustainability, such as individual, social, technical, economic, and ecological sustainability. For each dimension, we examined the effects produced by the system in its life cycle and systemically in the medium and long term. The result of the process is the Ethereum based dApp. It is composed of two smart contracts and of Javascript user interfaces. The dApp includes the main functionality of existing platforms for recruitment in the construction sector and in addition supports the transparent communication between the construction site stakeholders regarding hiring practices in compliance with workplace safety laws.

In the rest of the paper, section 2 discusses the most recent research in the application of the blockchain for human resource (HR) management, focusing on the construction sector. Section 3 provides the context and the motivation of our work. Subsequently, in section 4, we define our system by describing system requirements and design approach, and in section 5, we present the system's implementation. In section 6, we discuss the implication of our work and compare our solution with the Bativigie platform. Finally, we make some conclusions.

2. RELATED WORK

In the last few years, the blockchain technology has increasingly embraced the industrial sector. However, it is only recently that experts have started looking into the potential impact of distributed ledger technology (DLT) on management of human resource (HR).

Typically, the recruitment of a worker is preceded by the evaluation of the candidate's work experience and skills, and the verification of their education, professional certifications, etc. In a subsequent phase, the employer must deal with the

bureaucratic hiring procedures, resulting in an increase in cost and time. It is therefore common to rely on the temporary labor-recruiting agencies to simplify the employment procedure and to avoid unwieldy activities. The research community questions how technology can simplify and make such a procedure cheaper. As underlined by Tanana (2019), reliability of data and applicants' portfolio has always been fundamental in the labor market, which involves several actors with contradicting objectives. Moreover, according to Liyuan et al. (2019), a workers' background check is still a pain point, especially when it comes to verifying employment, education, and skills. With that purpose, Sarda et al. (2018) designed a blockchain system that allows individuals to share their work history and allows employers to verify that information. The prototype, which they implemented in Ethereum platform, ensures trustworthy and privacy-preserving data and avoids fake data sharing. Peisl and Shah (2019) investigated the impact of the blockchain in the recruitment process to manage the employee lifecycle's stages. Currently, many startups and researchers are investing on the potentiality of the blockchain technology, combined with other processes and applications. In this regard, Keršič et al. (2019) presented a combined solution of blockchain technology and Artificial Intelligence, taking advantage of their most important relevant features, to find an adequate matching between people recruitment and ongoing projects. Also, Michaelides (2018) discussed how blockchain and AI are affecting HR practices by providing more accurate approaches aiming to hire a worker. Data such as candidates' successes and failures, transfers, promotions, layoffs, and so on can be stored within the ledger by allowing automated validations of CVs. In addition, the ledger can record employment contracts, the use of cryptocurrencies for international payrolls, and so on.

Focusing on the HR management, Onik et al. (2018) conducted a detailed literature review to investigate how the use of blockchain technology can contribute to that domain in order to achieve a smart, cost-effective, efficient, transparent and secure system. Authors underlined the need for a smart employee hiring and management system in anticipation of the upcoming "Industry 5.0." For that reason, they conceived a methodology for a Blockchain-based Recruitment Management System (BcRMS) and Blockchain-based Human Resource Management System (BcHRMS) able to lower the risk of wrong recruitment and of biased human resource management. Jeong and Choi (2019) presented a blockchain-based platform for certificate management, useful both for performance assessment during recruitment process and for assessing the application, career management, and work history of an applicant. They implemented a digital certification issuance platform with blockchain Ethereum and Bitcoin support in which applicants can claim evidence of their competence by means of a digital backpack. Pinna and Ibba (2018) conceived a blockchain-based system to manage temporary work with a twofold aim: the first is to safeguard the worker by ensuring the respect for rights and guaranteeing a fair and legal payment of work performances; the second is to assist the employer in processing contracts providing a fully automated and fast procedure.

The construction industry is backward in terms of digitalization, and, as pointed out by Perera et al. (2020), it is historically reported as one of the latest and most reluctant sectors (just above agriculture and hunting) to have adopted information technology. By conducting a systematic literature review and a deep analysis of Use Cases, they critically analyzed the potential impact of blockchain technology adoption in this sector, finding out that it can be able to mitigate the existing issues in the current construction systems. From the same point of view, Tezel et al. (2019) conducted an exploratory analysis to investigate how to prepare construction supply chains for blockchain technology. Hijazi et al. (2019) conceived construction projects as a supply chain in which different stakeholders need to work together temporarily to deliver one-off projects. They found that the blockchain technology might help to reach transparency and to enhance the quality of the final building. Moreover, they suggested the integration of that technology within the BIM, Building Information Modeling, to simplify the entire construction supply chain.

3. MOTIVATION

According to the European Construction Sector Observatory (2019) of the European Commission, the construction sector and its activities are a key driver of the overall economy. In the European Union in 2016, the construction sector provided for 18 million jobs and contributed to almost 9% of GDP. Given its complex structure, the construction sector has to face wide issues such as competitiveness, bureaucracy, worker recruitment, productivity, and so on. In addition to this, there is a difficulty with the adoption of Information Technology, which can significantly contribute to the sector development from the point of view of sustainable competitiveness, improving both productivity and profitability. Therefore, the EU supported the development and adoption of digital innovations such as BIM. However, not all European countries are equally investing in the digitization of the sector, creating a heterogeneous development. Furthermore, it should be stressed that each state has its own legislation on the construction sector, often characterized by a complex bureaucracy and by multiple actors.

3.1. The Blockchain Technology for Human Resources Management

Blockchain technology is based on a decentralized database to efficiently manage and record transactions. The blockchain works in a peer-to-peer network, and its data are replicated among all the nodes of the network. The same information is present on all nodes and therefore becomes non-editable except through an operation that requires the approval of most of the nodes in the network. It can be seen as a decentralized public ledger in which the transactions consist of encrypted data, which are verified and approved by the nodes participating in the network and organized by blocks. Nodes add blocks in the blockchain in accordance with a consensus algorithm that rules the who and when new blocks can be stored in the blockchain and shared in the network. Therefore, this technology introduces a new

level of transparency and efficiency, allowing the network to manage and create secure and trustworthy transactions in an untrusted environment.

Nowadays, the blockchain technology allows the recording and the execution of computer programs called “Smart Contracts” (SC) that incorporate the blockchain features and add new features at the discretion of the developer. A smart contract is a special typology of account stored within a block. It can automate the execution of given processes if some conditions are met and contains code programming that is self-executed in the exact manner programmers intended. As presented in the previous section, blockchain technology can offer huge opportunities for the construction industry in terms of transparency, productivity, and sustainability. This implicates the implementation of smart contracts for HR recruitment and management, payments, BIM, property and asset management, the construction supply chain, and all future challenges (Penzes et al., 2018).

Recently, several initiatives involving the development of blockchain systems to manage worker recruitment are taking place. To provide some examples, the platform APII (appii.io) allows the creation of verified curriculum vitae of applicants by means of the blockchain technology. The blockchain Steem (steem.com) rewards users for sharing content. It can be used by recruiters to obtain applicant data from the source directly, without paying intermediaries. The blockchain-based tool by IRIS-SAFER (International Recruitment Integrity System-Self-Assessment for Ethical Recruitment) announced by the International Organization of Migration, IOM, and by Diginex (2019), aims to fight the exploitation of migrant workers in Hong Kong by “strengthen data management and enforce data integrity, which allows for a higher level of transparency and visibility.” Aworker (aworker.io) is an Estonian blockchain project for recruitment based on the Ethereum platform and was conceived to help people to find the right job depending on their skills and companies to hire the appropriate candidate by reducing costs. Breezy-HR (breezy.hr) is a recruiting blockchain software developed to help companies in the hiring processes.

3.2. Blockchain for the Building Sector

The key role of the construction sector in the overall economy has been repeatedly stressed by the European Union. By means of *Strategy for the sustainable competitiveness of the construction sector and its enterprises*¹, the EU identifies the main challenges for the sector as an increase in investments, human capital, regulation and access to markets, and environmental requirements. In response to the crisis, the EU put emphasis on the need to support growth and employment in the construction sector.

However, the construction sector has to face several structural problems, such as the lack of skilled labor. It should be noted that the construction industry is a low attractive sector for young people due to the working conditions, low-wages, and illegal work. EU-OSHA (osha.europa.eu) is the European Union

information agency that deals with occupational safety and health by promoting a culture of risk prevention to improve working conditions in Europe. At the national level, each nation has a specific Focal Point that contributes substantially to EU-OSHA work program implementation. In Italy, that Focal Point is represented by INAIL (National Institute for Insurance against Accidents at Work). By focusing on the Italian context, the working conditions are worrying. According to a report by Inail (2019), in Italy in 2018, there were 409,106 denounced work injuries, 704 of which were deadly (officially confirmed cases). Moreover, in the same year in Italy, the Department of labor denounced almost 70% of irregularities in the building sector (Alestra, 2018).

There is, of course, an obvious need to improve working conditions in this sector, hence the urgent necessity to enforce the current regulation. Blockchains are conceived as a “Technology of Trust” (Nguyen, 2016; Heiskanen, 2017), and more broad benefits are thus emerging for the society that leads the development of several projects in different industrial sectors. Given the intrinsic properties of its technical structure, blockchain ensures trust, sharing, transparency, and security, it exhibits a strong potential for many of the 17 Sustainable Development Goals (SDGs) defined by the United Nations (as confirmed by Zwitter and Herman, 2018; Al-Htaybat et al., 2019; França et al., 2019; Saberi et al., 2019). More specifically, by considering the building sector, we believe that blockchain technology can support directly the achievement of Goal n.8: “Decent work and Economic growth.” This work proposes a decentralized platform, based on a blockchain, that is able to control employment relationships more efficiently. All data recorded within the system are transparent and traceable, which supports inspections in the workplace, by reducing costs and preventing undeclared work. On the other hand, individuals can use the system to apply for a job by sharing their portfolio, while employers can easily verify that information.

The main advantages of using a such blockchain-based system for the HR in building sector can be summarized as the following:

- Legal and transparent safety requirements: construction company can publish safety details of the construction site, making it available for inspection;
- Certificated experiences: the responsible authorities can insert certificates and educational achievement for the worker within the system by simplifying future checks and avoiding fake data sharing;
- Verify identities: it can be used to verify identities and to enable controls and specific functions depending on the user;
- Insert job applications: the worker can use the system to look for job offers in construction sites and to insert his application and contextually share his expertise and certificates with the employer;
- Register contracts: each work contract (including temporary ones) can be automatically created, and stored within the system by means of smart contracts. Certain events (such as the worker payment) can be authorized only if certain conditions set out in the agreement between parties are met.

¹<https://eur-lex.europa.eu/legal-content/en/TXT/?uri=CELEX:52012DC0433>

- Find and verify workers skills: it can be seen as an alternative to a traditional candidate recruitment system by saving time and replacing the expensive temporary employment agencies;
- Compliance with regulation: inspectors can more easily verify compliance with current regulations and to guarantee the worker rights. The system promotes transparent controls and improves working conditions by hindering undeclared work.

4. METHOD

Our study started by identifying the stakeholders and their goals, in the context of the management of temporary workers in construction sites, taking into account the current EU legal requirements in general, and in particular Italy as example case study. The definition of the stakeholders and their objectives has allowed us to identify the actors of the system, each representing a homogeneous set of system users. From these, we have elicited the system requirements. We documented and represented actors and requirements through UML diagrams. Before defining the details of the system, we performed a sustainability analysis, which allows us to examine the system in the five dimensions of sustainability. The analysis leads us to become aware of the impact of the system under development and to increase the knowledge we have about the interrelation between the system's stakeholders. Thereafter, we represented the overall system as a block diagram in which we separated the on-chain subsystem from the other elements of the system (such as the mobile applications or the web based interfaces). The processes described above are actually iterative processes that allowed us to gradually refine the system design through cyclic iterations.

Acknowledged the risks and the open challenges in Blockchain Oriented Software Engineering (Porru et al., 2017), we decided to adopt the Agile BlockChain Dapp Engineering method (ABCDE) (Marchesi et al., 2018), a recently proposed methodology conceived to manage BOS specific issues, in the design and implementation phases of dApps.

4.1. System Stakeholders and Goals

We first individuated four stakeholders that can benefit of the system and that can determine its success.

- The site owner is the person that will benefit from the property under construction. It can be either a private body or a public body. They can charge a construction company to carry out the building works.
- The construction company represents the contractor company that starts the construction site, which could need an additional workforce and which must comply with the laws in force in the sector.
- The worker is a private citizen who looks for an employment adapted to his abilities and who could be hired by the construction company for a temporary employment in the construction site.
- The labor inspectorate is the public body in charge of verifying compliance with labor and safety laws by the construction company.

Furthermore we identified an additional stakeholder:

- The system owner. He is the final owner of the blockchain oriented system under design.

We recognized that each stakeholder has specific goals in adopting the system. In the following, we summarize the goals for each stakeholder.

- Goals of the site owner: to be sure of the correct progress of the work and to arrive at the completion of the construction work within the expected times and costs.
- Goals of the construction company: to be able to publish safety data on the construction site through an easy and transparent procedure and to have a streamlined and effective procedure for finding skilled workers and hiring workers.
- Goals of the worker include easily finding transparent job offers adapted for his skills; to easily describe himself and to edit his competences profile; to be sure of the correctness, in legal terms, of the offered job contract; to take track of his employment history; and to protect his privacy.
- Goals of the labor inspectorate include how to easily access safety documents of the construction site; to access to updated temporary employment documents; and to compare what declared by the construction company with the real condition in the construction site.

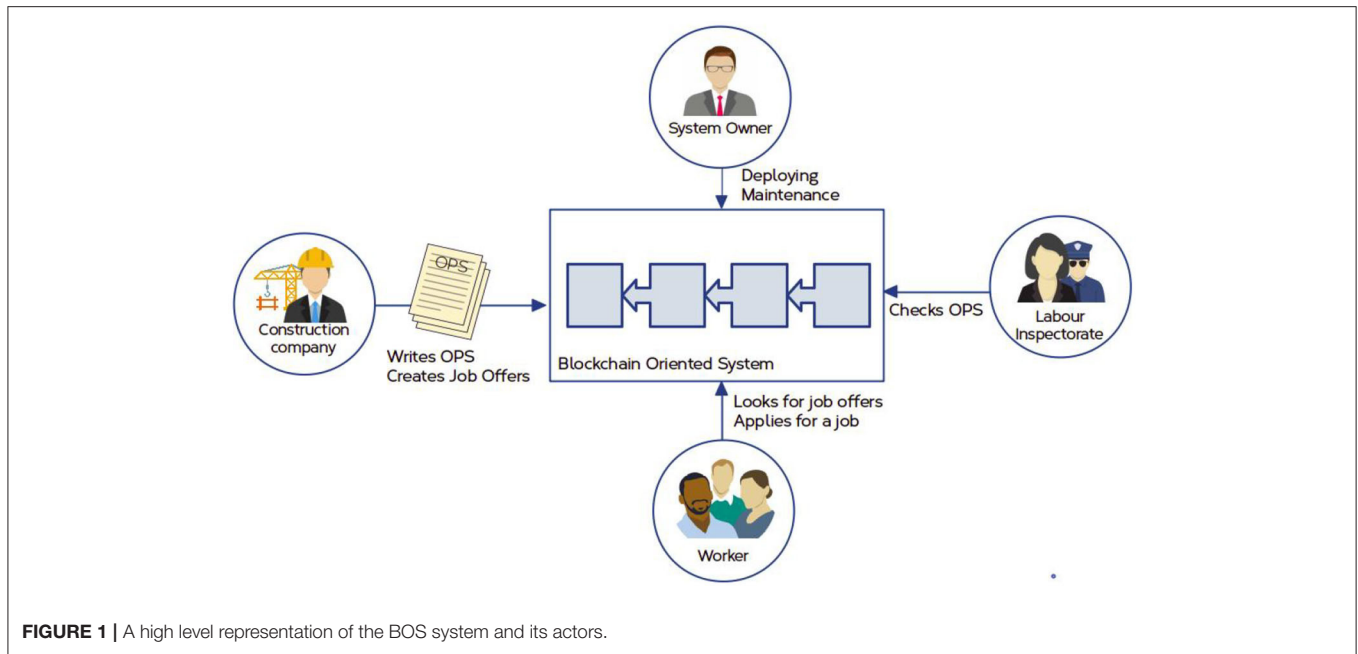
Also, the additional stakeholder has specific goals.

- Goals of the system owner include being able to evaluate the efficacy of the system and to obtain an economic benefit from users of the platform.

4.2. Actors and Requirements

We elicited the system requirements in two ways. We started from the analysis of the stakeholders' goals, taking into account information collected by interviews of domain experts such as civil engineers. Then, we analyzed information regarding the employment process in build construction we found in the literature and in EU regulations. In our design we considered as actors of the system only a subset of stakeholders. In particular, we identified as actors the construction company, the worker, the labor inspectorate, and the system owner. The reason why we did not represent the site owner as system actors lie in the requirement elicitation phase in which he does not add specific functional requirements. **Figure 1** shows a high-level representation of system actors.

We have taken into account in our case study the Italian legislation about work on construction sites and, in particular, the mandatory document called the *Operational Safety Plan*, OSP (in Italian: Piano Operativo di Sicurezza), which the construction companies must provide before starting work at the construction site. That document will be written by the construction company, made unchangeable after the start of work at the construction site, will be made available for reading to the labor inspectorate, and must have a time stamp. We conceived a decentralized solution that sees the OSP as a digital record in the blockchain and the actors as blockchain users. We elicited the first requirement as the following. R1: *The blockchain*



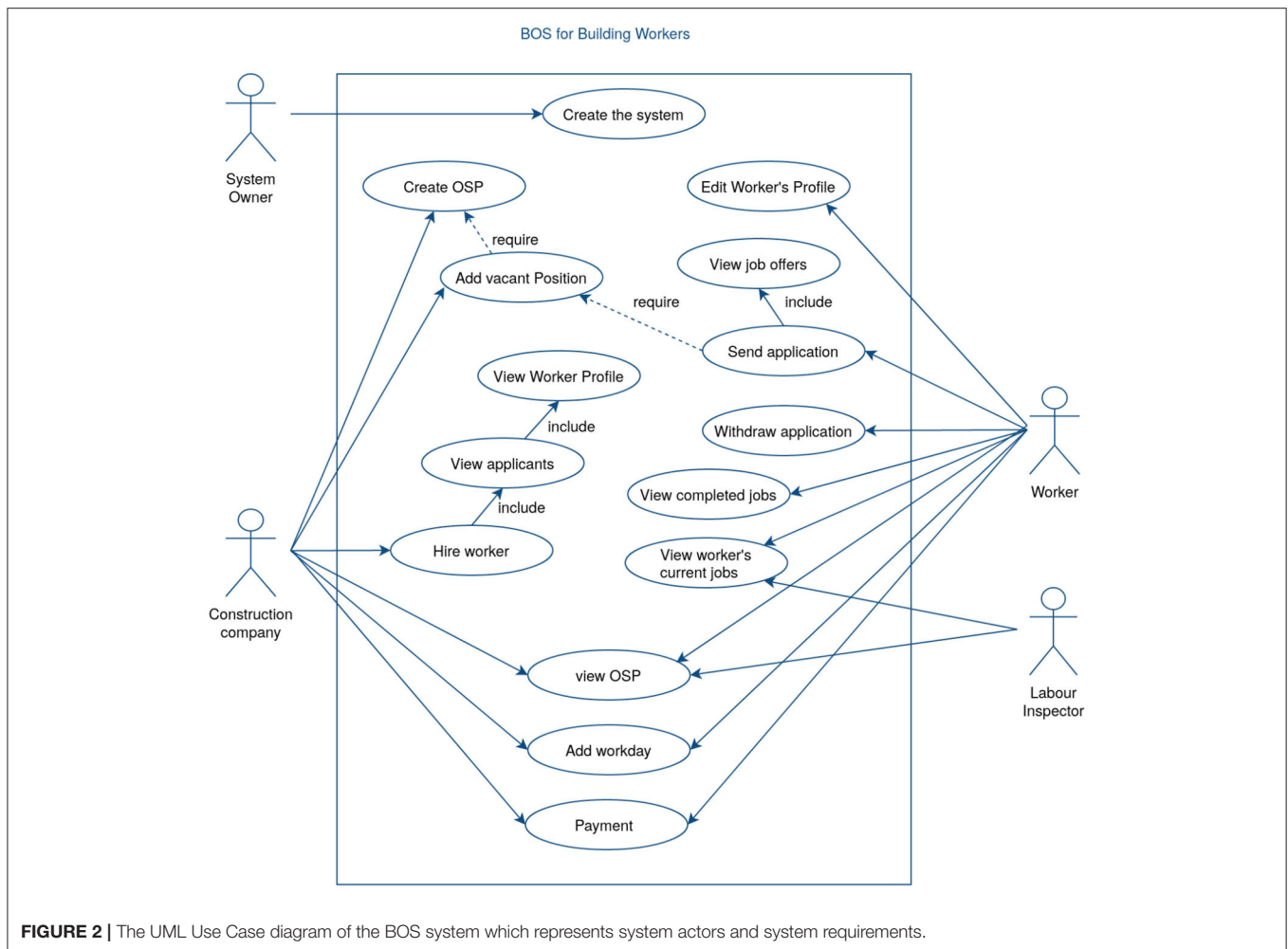
subsystem must be programmed to allow a construction company to add an OSP in the system. OSP data must be recorded in the blockchain. The labor inspector must be free to read OSP data stored in the blockchain. The construction company and the labor inspectorate must be uniquely identified by blockchain addresses. The OSP data must be uniquely connected with the construction company.

Italian laws establish that the OSP must include the identification data of the construction company, the typology of tasks, and the names of the people in charge of protection and prevention, first aid, fire, and emergencies. The most relevant part of the OSP is the section regarding workers, which includes the number of workers and, for each worker, the description of the job the worker will do in the construction site. Work descriptions include the typologies of activity and the elements of the construction site with which the worker will be in contact (for instance the number of scaffolds or the eventual use of dangerous tools and substances). Taking into account the current situation and by examining the goals of the construction company and of the worker, we developed the following requirements. R2: *The construction company must be able to declare the expected number of workers and to write the job description including the duration of the contract.* R3: *For each job, the construction company can add the corresponding worker if already hired. In a case where the job, at the start of work at the construction site, is declared but is still uncovered, the construction company can declare a vacant position.* R4: *The labor inspector must be able to verify the correctness of the OSP document, especially for what concerns the safety requirements, both for covered jobs and for vacant positions.*

Now, thanks to the blockchain technology, we are able to implement a reliable and unchangeable proof of the construction company's declarations, which are available to the labor inspector,

due also in part to the decentralized nature of that technology. We can take into account the goals of the worker and the goals of the construction company, which concern the possibility to exploit the creation of job offers for vacant positions in the construction site. We do not enter into the details for what concerns the reliability of workers' profile. It is a requirement we take for granted because several research works have already faced this problem, as widely discussed in sections 2 and 3.

We elicited the following requirements. R5: *The construction company will create a new job offer for a vacant position at the construction site. It will contain the job description and the desired skills and experience of applicants. The job offer will be connected to the OSP of the construction site.* R6: *The worker will be able to freely find the job offer, read its details and the details of the connected OSP, and apply for the selection.* R7: *The worker will be able to create their professional profile by adding their competences and professional certificates.* R8: *The construction company will be able to read the list of applicants, read their skills and experience, and choose the candidate it wishes to hire. The identifier of the selected worker will be added to the OSP.* R9: *Once hired, the construction company can asseverate the daily presence in the site of the worker.* R10: *The labor inspector will be able to check the updated OSP for its correctness and monitor the worker activity.* R11: *The worker will see the details of their current occupation and add it the accomplished work days. They can also read all his past employment history.* We have reached the core of the system, namely, the management of construction workers. With the system, the construction company can hire workers even when construction has already started, without violating what has been declared in the OSP. This adds flexibility to the management of the construction company and, at the same time, increases the transparency of the management.



An additional requirement concerns the privacy goal. R12. *Details of personal information must be made available only to the owner and those the owner grants authorization to.* It is a non-functional requirement that will be implemented by exploiting the blockchain features such as pseudo-anonymity, cryptographic functions, and permission management. Some previous works conceived an automatic payment system for workers that is activated upon job completion. That payment system allows for the exchange of the accorded cryptocurrency amount from the employer to the worker. In the scenario we are describing, the laws regarding the contracting of construction workers must be taken into account. In our case study construction workers belongs to two categories: freelancers and employees. The first category issues an invoice for their work and is seen as a company without employees. The second is a worker who must be hired at a company, according to the National Collective Contract for Construction Workers. For the first category, the automatic payment system can be easily adopted without breaking legal barriers. The second category requires specific attention, as that type of contract adds complications to a possible automatic payment system.

4.3. Use Case Diagram

The Use Case diagram in **Figure 2** represents the system functional requirements and related actors. A Use Case diagram is a UML diagram that helps the representation of the required features and the respective links among them and with system actors. It also helps to analyze system requirements and to create a link with the next phase of the software engineering process, namely, the design of the system. Each requirement can be mapped in one of the “Use Case” diagrams. A dashed arrow represents a dependence. A worker can apply for a vacant position only after a construction company created it. In turn, a construction company can add a vacant position only after it has created the Operational Safety Plan, which declares the vacant job. The Use Case “Hire worker” allows a construction company to hire a worker for a vacant position. This Use Case includes the Use Case “View Applicants” to allow a construction company to know who the applicants are along with a profile (see above the requirement R8). The Use Case diagram is useful to understand the overall structure of the system. A very recent software engineering methodology for BOS design, the “Agile BlockChain Dapp Engineering” (ABCDE) (Marchesi et al., 2018), suggests proceeding by splitting the system in two main

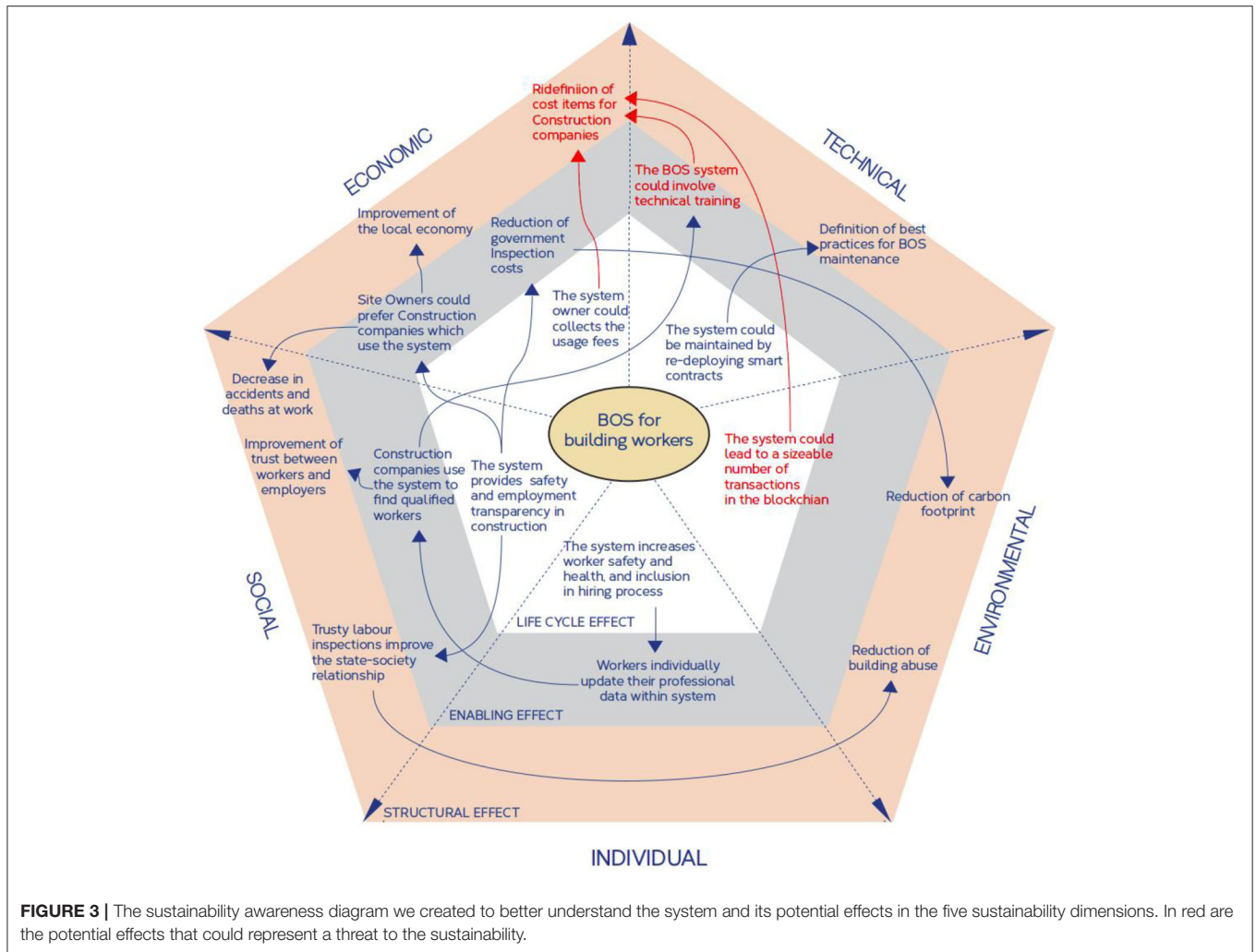


FIGURE 3 | The sustainability awareness diagram we created to better understand the system and its potential effects in the five sustainability dimensions. In red are the potential effects that could represent a threat to the sustainability.

components: the on-chain system and the out-of-chain system. In particular, the on-chain system will include the blockchain infrastructure, its records, and the smart contracts, which will rule the behavior of the system. The out-of-chain system will include the user interfaces and conventional data storage. On-chain and out-of-chain subsystems must be able to exchange data according to specific protocols and interfaces. In turn, the on-chain subsystem can be subdivided in a number of smart contracts, which encapsulate data and functions of each element to be implemented. Section 4.5 reports our design choices.

4.4. Sustainability Analysis

We decided to perform the sustainability analysis due to the innovative impact of our proposal, whose effects are difficult to evaluate. Thanks to the creation of the sustainability diagram, we acquired awareness on system potential effects and discovered new requirements for the system design.

Software engineering for sustainability is a relatively new branch of software engineering that focuses on the design of sustainable software systems and on its consequences in the long-term. The Karlskrona Manifesto for Sustainability Design

(Becker et al., 2015) highlights its motivations and main features. According to Penzenstadler et al. (2018), software sustainability can be analyzed in five dimensions, which include the three “classic” dimensions (social, economic, and environmental) and two additional dimensions called individual and technical. Once the system is adopted, stakeholders will be involved in some unavoidable changes, in one or more dimensions, which we call effects. Immediate effects are directly connected to the system life cycle in terms of its “natural” usage. Enabling effects result in changes in the behavior of stakeholders or the creation of new habits. Structural effects concern substantial and consolidated changes that can be produced by the system in the long term.

The sustainability awareness diagram (Becker et al., 2016; Duboc et al., 2019) is a graphic tool that has the shape of a pentagon that has been subdivided into five triangles. Each triangle represents one dimension of sustainability. In turn, each triangle is subdivided into three areas that represent the three orders of effects. **Figure 3** shows the sustainability awareness diagram for our BOS system. It is possible to read the diagram by starting from one of the life cycle effects and by following the arrows to discover enabling and structural effects. To fill the

diagram we begin inserting our BOS for Building Workers at the center of the pentagon, and we evaluated the immediate effects of the system on the stakeholders, during the system life cycle, via each dimension of sustainability. We individuated one life cycle effect for each dimension. We can see from the figure that immediate effects lead to several enabling effects and structural effects. Sentences in red indicate effects that would threaten the system sustainability. In fairness, we can say that discovering effects is not so trivial, and other effects are probably still hidden. For brevity, in the following, we will focus on the effects written in red and on their consequences in defining the requirements. In the final discussion of this paper, we will resume the topic to underline what we learned from the sustainability analysis.

The effect of life cycle effects threaten environmental sustainability. As Blockchain-Oriented Software, the system requires the sending and receiving of blockchain transactions. Each transaction leads to an additional energy expenditure due to the computational effort and additional data collected by the peer-to-peer network. Energy consumption (and the associated carbon footprint) is one of the most relevant problems in the use of the blockchain, especially if the network relies on a “proof-of-work” based algorithm (Stoll et al., 2019). To face this problem, new blockchain protocols have been studied to limit the energy consumption and the carbon footprint (Bach et al., 2018; Truby, 2018). We must design the system to limit the use of blockchain transactions to the strictly necessary by limiting any writing operation within the blockchain.

We also discovered an enabling technical effect that will be analyzed. It concerns the need for technical training for the final users of the system, particularly for the construction companies. To facilitate system adoption and learning, the system must present a familiar user interface to the user. We have to keep in mind that the user will know very little about blockchain technology. It is also important that the user be aware about the implications of using blockchain addresses, such as the possibility of losing cryptocurrency values forever if the user loses the address keys. The BOS system should guide new blockchain users in these operations. Choices in designing the user interface are relevant for the system success and in the decreasing of training costs. The user interface is part of the out-of-chain subsystem and can be improved through subsequent integration and continuous delivery (Beck et al., 2001), taking into account users’ feedbacks.

We discovered a structural economic effect that we need to pay attention to. It concerns the redefinition of cost items for the construction company balance. As described above, the use of the BOS system involves costs. It also involves money saving in recruitment phases, however, because the system limits the need of intermediaries in the search for qualified and trusted workers.

Finally, we need to put the attention to the technical life cycle effect, which concerns system maintenance. Suppose now we were using the BOS system when the software starts behaving unexpectedly. For example, the system resets the count of workdays instead of increasing them or something similar. The system owner should promptly individuate bugs, correct the code, and then deliver a new version of the system. If we find out that the bug resides in the on-chain component of the system (i.e., inside a Smart Contract deployed within the

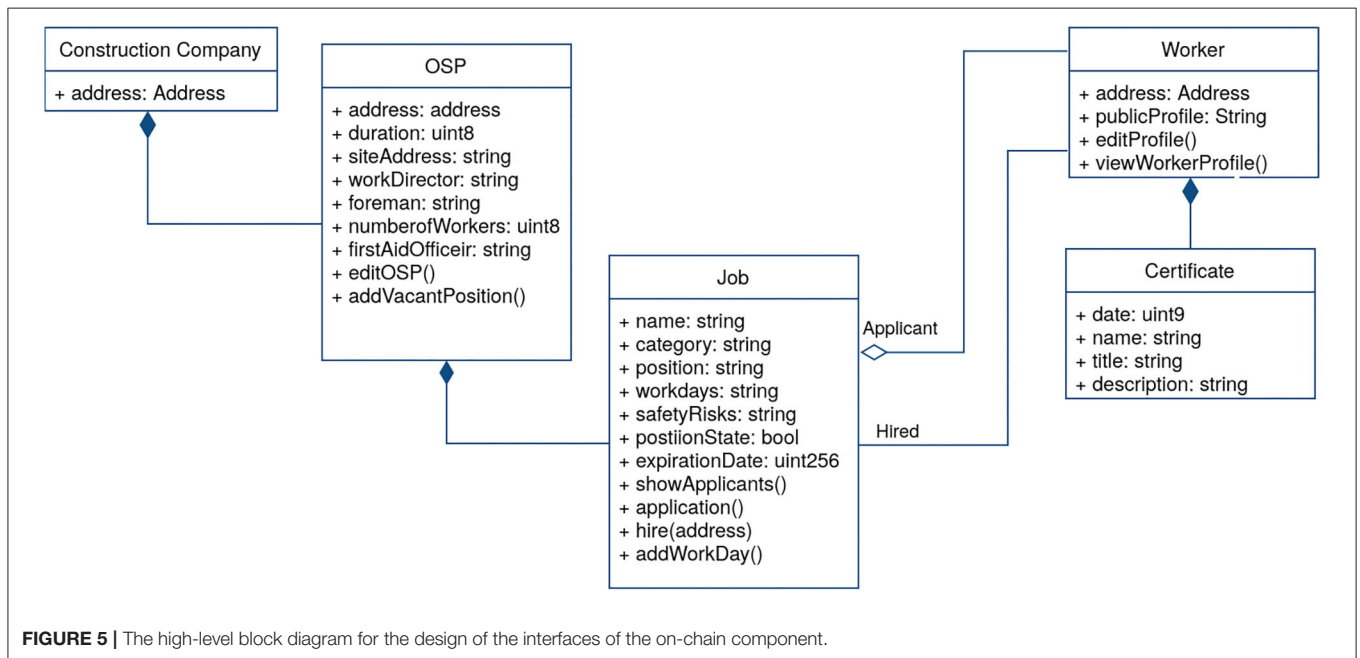
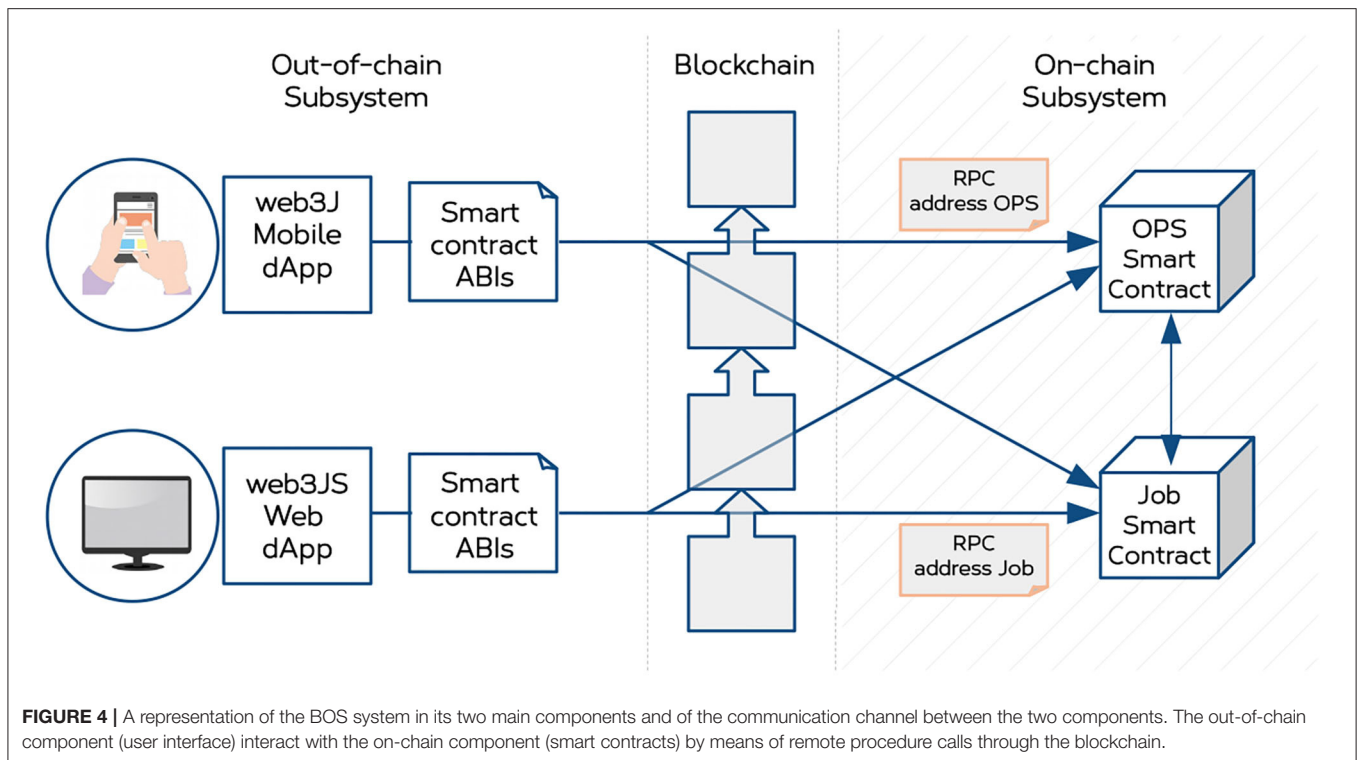
blockchain) we will have to face the fact that the smart contract op-code is unchangeable once deployed. Software engineering researchers have already highlighted this problem (Porru et al., 2017; Marchesi et al., 2018), and several results propose strategies for effective testing of smart contracts (see for instance Destefanis et al., 2018; Wang et al., 2019) to limit the presence of defects in the final version of the system. In any case, undiscovered bugs could remain in the smart contract, and we have to manage such a situation during the software life cycle. Similarly, a situation might arise in which we need to add features to an already delivered BOS system (Pinna et al., 2019). Waiting to define the best practices for the maintenance of BOS, we identified two strategies to upgrade the on-chain component: (A) We could see the complete disposal of the current version of the smart contract and the deployment of a new version. Contextually, all components of the system will be upgraded in order to replace the old smart contract address with the new one. (B) Design each smart contract to make possible its transforming into call forwarders. Practically, the current version of the smart contract will continue to receive calls from the other component of the BOS, but it will forward the calls to the new version of the smart contract. This strategy avoids the need of the upgrade of the other components of the system but requires more effort in the smart contracts design and implementation.

In summary, the sustainability analysis allowed us to elicit three additional non-functional requirements. R13. *The system should limit the number of blockchain transactions to the strictly necessary.* R14. *The system User Interface must be user friendly and will have to support users in using the system features.* R15. *The on-chain component should include a maintenance procedure.*

4.5. System Architecture and Design

We divided our BOS system in two main subsystems: the on-chain and the out-of-chain subsystems. The out-of-chain subsystem is responsible for the Web-based user interface. The on-chain system is the component stored and executed within the blockchain infrastructure and can be made of smart contracts. Well-designed ABIs (Application Binary Interfaces) act as a link between the on-chain and the out-of-chain subsystems. **Figure 4** represents the main elements of the out-of-chain and on-chain systems. In this paper, we focus mainly on the design of the on-chain subsystem.

We conceived our BOS to be decentralized and transparent. This means that we will design specific smart contracts to implement the specific Use Cases. We can subdivide the Use Cases into three sets. The first set concerns functions and data, which directly involve the construction only. Is the case of the requirement R1, R2, R3, and R5, represented by the Use Case “Create OSP” and the Use Case “Add Vacant Position.” The second set of Use Cases directly involves the worker only. It is the case of the requirement R6, R7, and R11, represented by the Use Cases “Edit worker profile,” “View job Offers,” and “View Complete Jobs.” The remaining requirements belong to the set of requirements involving multiple actors. We organized data and functions in five main elements: the construction company, the OSP, the job, the worker, and the certificate. **Figure 5** shows a class diagram to represent system elements as classes. It contains



essential data and public functions of classes and represents the interconnections between system elements.

The next step of the design iteration is the definition of Smart Contracts. From requirements, we learn that each OSP is the property of a construction company. At the same time, the construction company owns jobs and vacant jobs, contained in a given OSP. An OSP has a fixed structure and an owner who

created it and who can modify its data. Also, a job has a fixed structure and an owner. The Ethereum Improvement Proposal 721 (also known as ERC721) is a library which provides the interface of a useful “non-fungible” token. A non-fungible token is a digital element which belongs to a family of tokens which elements share the same structure but contains different data. A token has always an owner, can be “minted,” is enumerated,

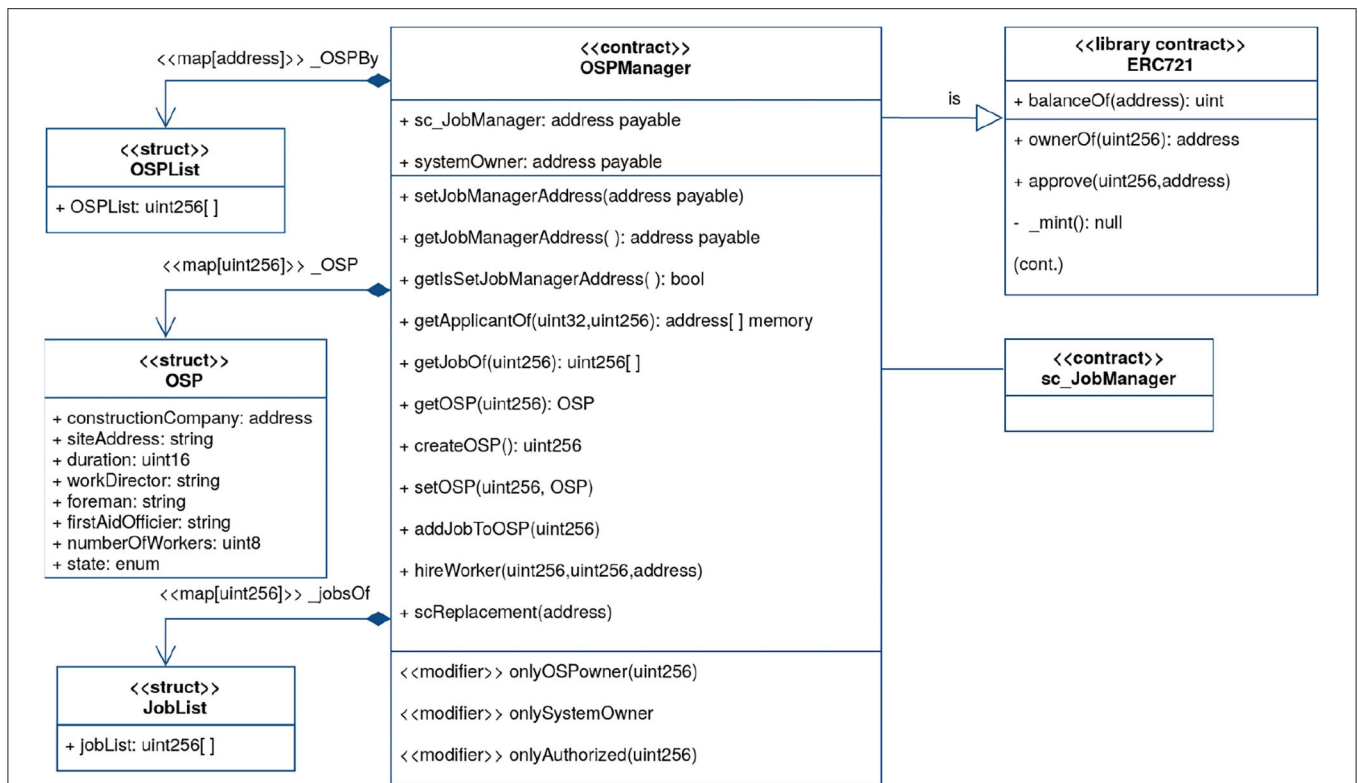


FIGURE 6 | The contract diagram of the smart contract “OSPManager.” This diagram extends the UML Class diagram. Its stereotypes represent the structured data (contract and struct) provided by the Solidity language.

can be passed to a new owner, allows the authorizing for the management of other accounts different from the owner. Using the library ERC721, we can exploit well-implemented tokens which satisfy security requirements. We decided to implement OSPs and jobs as non-fungible tokens. We defined the two main smart contracts: “OSPManager” and “JobManager.” Both inherit the smart contract ERC721.

The ABCDE method proposes the use of specialized diagram and tables to provide a proper definition of smart contract-specific elements, such as the application binary interface, mapping, and function modifiers. We used the “contract diagram,” a graphical description of on-chain elements specifically conceived for solidity-based smart contracts. This diagram inherits the UML class diagram elements and adds some specific stereotypes. We use this diagram to define the public elements of the smart contracts and their structures.

We named “OSPManager” the smart contract for the creation and the management of the OSP for a given construction site. This smart contract implements requirements R1 and partially R2. **Figure 6** shows the contract diagram, we have drawn to define the smart contract “OSPManager.” We defined each OSP as a non-fungible token, and, for this reason, the contract “OSPManager” inherits the ERC721 contract. The construction company that creates a new OSP becomes its owner. The contract assigns a numeric ID to the OSP token and stores OSP data in a structure type. The structure contains the

information about the construction site required by the law and contains a list of jobs which characterize the working activity in the given construction site. The token IDs and data structures are linked through a map. Another map links construction company addresses to the list of OSP. The contract “OSPManager” contains a set of public variables and functions. Public elements will be visible to and callable from the out-of-chain components. These elements will be written in the contract ABI (application binary interface) after compilation. Despite public visibility, we are able to prevent unauthorized executions of functions thanks to function modifiers. We define three modifiers which details are reported in **Table 1**. Inside functions, we implemented other checks and prevention by means the keyword “require.”

Figure 7 shows the contract diagram we have drawn to define the smart contract “JobManager.” We named “JobManager” the smart contract for the creation and management of jobs in the construction site, and which allows the workers’ application for vacant positions. This smart contract implements the list of requirements from R2 to R6, partially the R8, and from R9 to R11. This smart contract will be mainly used by two actors: the worker and the construction company. The construction company will create jobs to be connected to an OSP and will hire a worker for every single job. Workers will apply for a job characterized by the “vacant position” status. Of course, each worker and construction company are identified by their

blockchain address. In this smart contract, we implemented “Jobs” in the construction site as an ERC721 non-fungible token. **Figure 8** shows the sequence diagram of main interaction between smart contracts and the actors involved in job creation and in the hiring process. The diagram focuses on the logical sequence of operation. To implement this process the smart contract “jobManager” includes three structs: “Job,” “JobList,” and “WorkerJobs.” The struct “Job” is the main data structure

of the smart contract. It collects the abovementioned status, the identifier of the OSP it belongs to, the job details, according to the legal requirements of the OSP, the address of the hired worker, and the address array of applicants. The function “createJob” uses the function “_mint” of the ERC721 to mint a new job token and assign the ownership to the construction company which creates it thorough mapping (phases 2 and 2.1 of in **Figure 8**). Each job is thus identified in the contract by a unique numerical ID that represents each token in the contract. The job can be assigned to a hired worker or vacant. If vacant, a worker can apply for this position by using the function “application” of the contract (phases 3 and 4 in **Figure 8**). The construction company can obtain the list of applicants and assign a job to a worker through the function “hire” (phases 5, 6, and 6.1 in **Figure 8**). Once hired, the worker can add his workdays for a job in the construction site through the function “addWorkDay.” The struct “WorkerJobs” stores an array of ID which represents job history for each worker, identified through a map between the worker’s address and the struct. The struct “JobList” stores the list of job which belongs to a specific construction company identified by its address and associate to the struct through a mapping. The list of public functions includes several getters to return jobs data given its identifier. In

TABLE 1 | Function modifiers implemented inside smart contracts.

Modifier	Action
onlyOSPowner	Prevents execution if the calling address is different from the owner of the OSP token of a given id.
onlySystemOwner	Prevents execution if the calling address differs from that of the system owner
onlyAuthorized	Prevents execution if the calling address has not been authorized to interact with the OSP token of a given id.
onlyWorker	Prevents execution if the calling address is different from the address of the worker associated to a given ID.

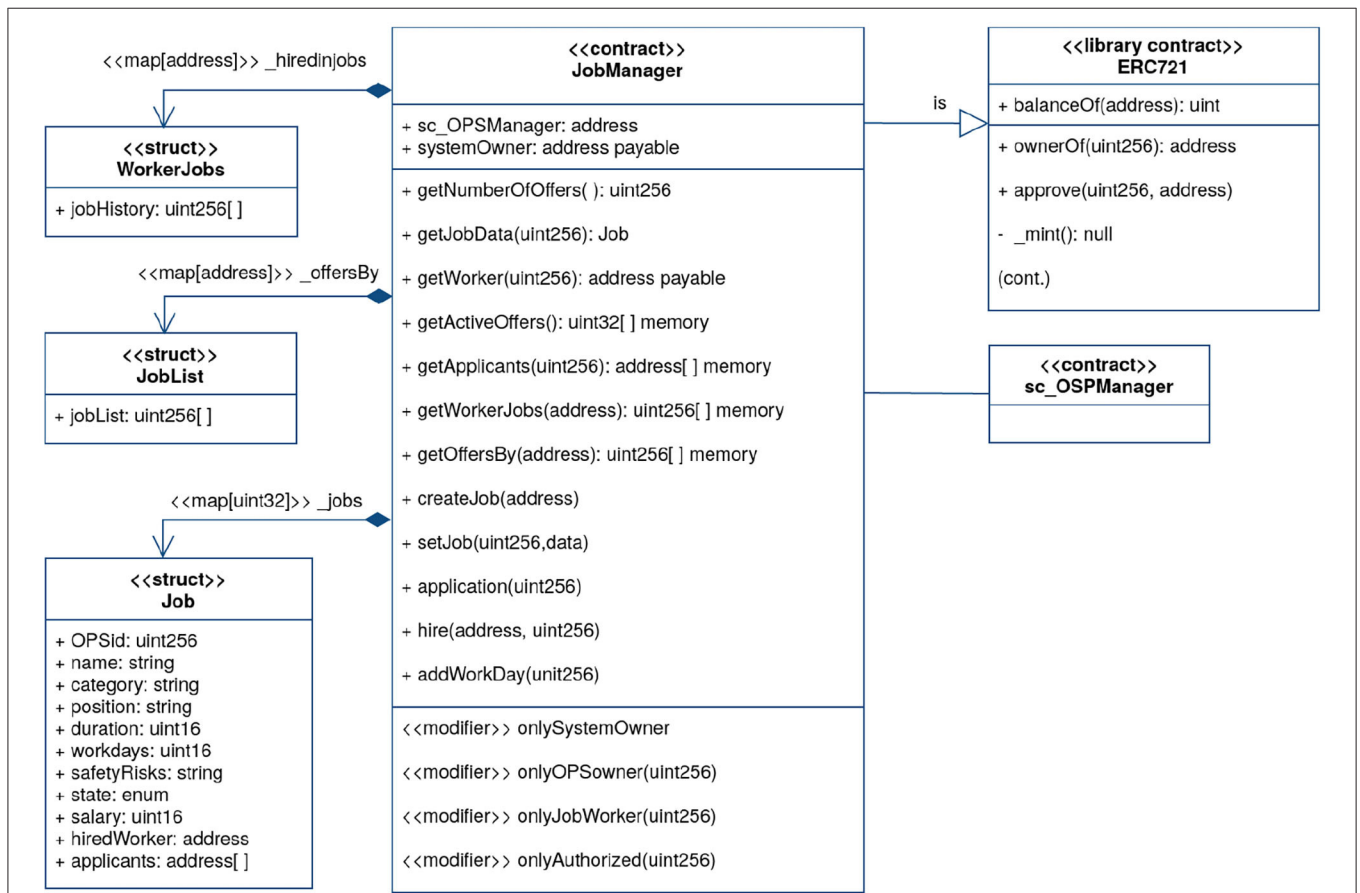


FIGURE 7 | The contract diagram of the smart contract “JobManager”.

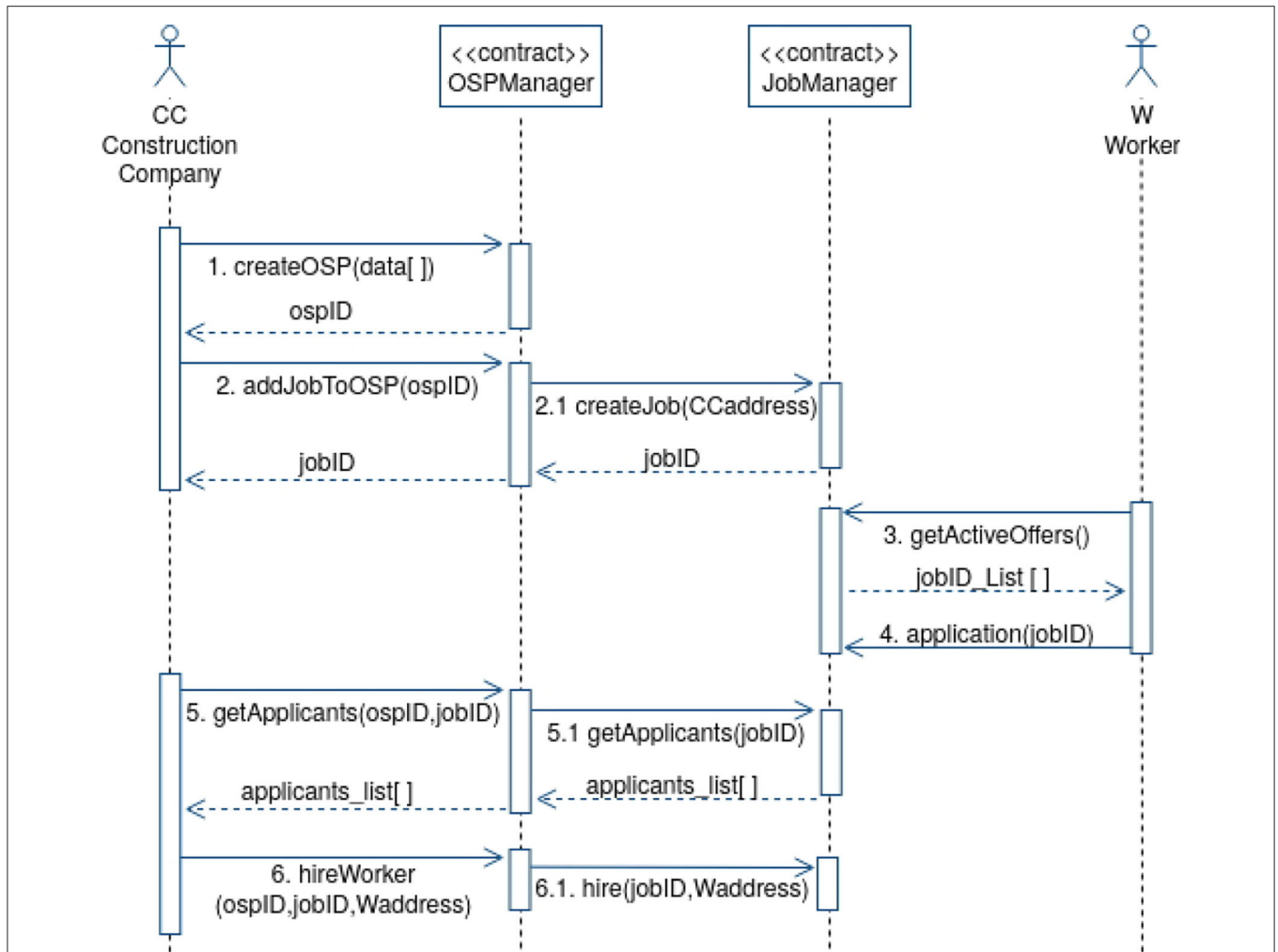
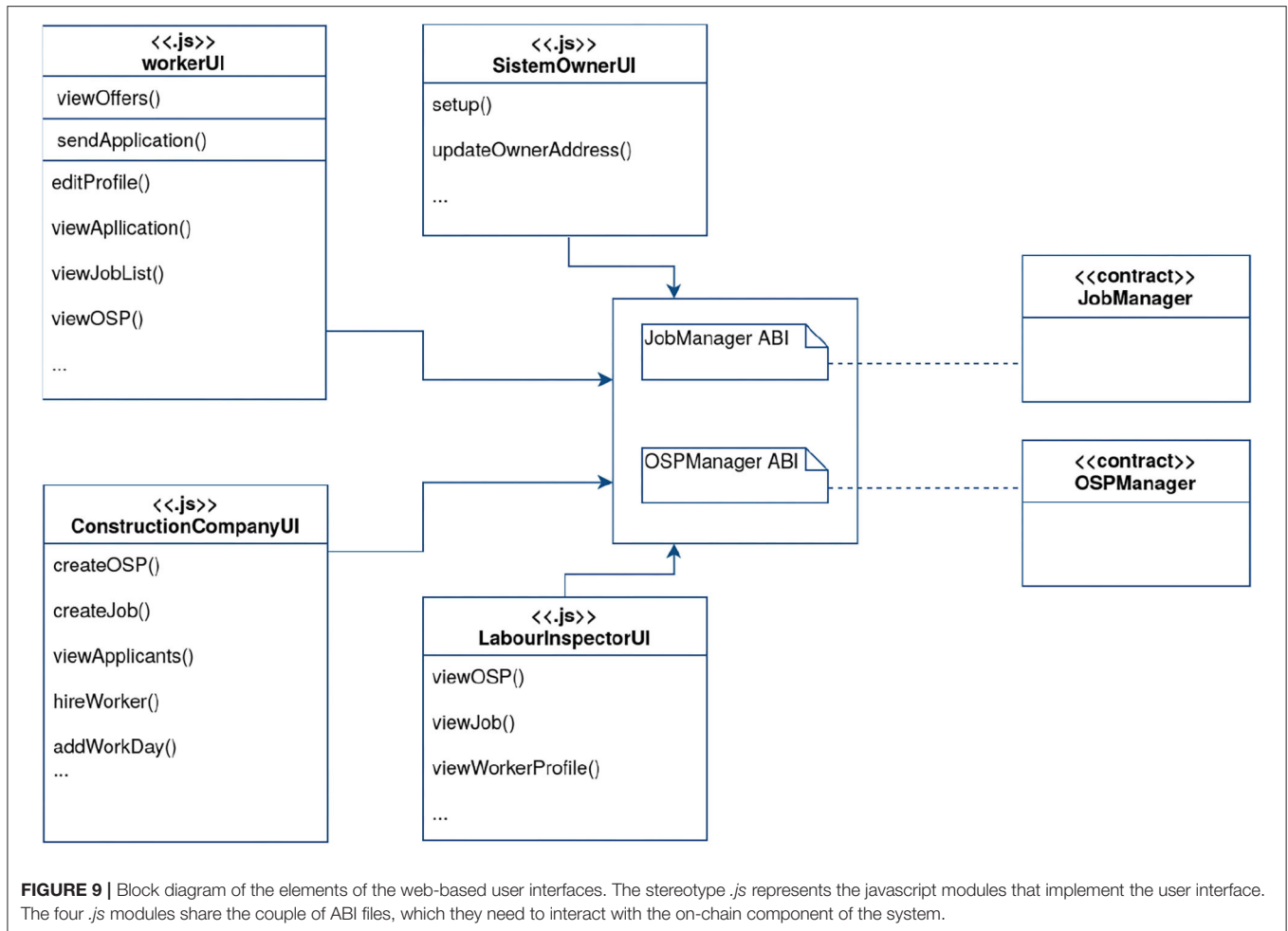


FIGURE 8 | Simplified Sequence Diagram of main interactions between the actors “Construction company” and “Worker” with the smart contracts “OSPManager” and “JobManager” during the hiring process.

general, getters should be free while setters and every function which modify stored data inside the blockchain should incur an execution cost. To rule the function execution the smart contract includes four function modifiers of which **Table 1** describes the actions.

The ABI of this contract is added to that of “OSPManager” and together must be imported into the user interface code. Thanks to the ABI contents, the user interface will be able to call remotely the public functions of deployed smart contracts. A further advantage of ABCDE is that the contract diagrams allow us to represent the contract ABIs in advance. By defining the ABI through the contract diagram, it is possible to start in parallel both the implementation of smart contracts and the web component. Finally, we can add that, to protect workers’ privacy, personal data must be recorded apart in a dedicated smart contract. In our system, OSP data include only the Ethereum address of the worker. According to literature results described in section 2, there are several options to implementing a trusted

privacy system for managing the personal and professional data of workers. Well-organized User Interfaces, which take into account the requirement R14, complete the decentralized application. To meet requirement R14 for ease of learning and user guidance, we have divided the interface into four modules, as represented in **Figure 9**, one for each type of actor in the system. In this way, each actor will only see the functionalities that belong to them. The modules contain specific functions for the type of user they are addressed to. The functions of user interfaces are designed to enable the easy interaction with the corresponding smart contract functions. User interfaces implementations share two ABI files, that we represented in the box in the center of the diagram, connected with a dashed line to the corresponding smart contract. These files are essential for the communication between the out-of-chain and the on-chain components because they contain all public function and variable names of the smart contract and the expected input and output types.



5. RESULTS

The implementation of the BOS system for building workers includes the writing and the deployment of the smart contracts and the writing and the publishing of web-based user interfaces. In addition, the implementation includes the unit tests to find vulnerabilities and bugs, and the acceptance tests to asseverate the correct implementation of the system features, previously described as requirements. We implemented the on-chain component of the system using the blockchain Ethereum, a famous and widely used platform for decentralized applications. We have chosen Ethereum for three reasons. Firstly, Ethereum is a decentralized blockchain, without obstacles at the entrance, and it is supported by a large peer-to-peer network. Secondly, the development environment is also continuously updated, both for what concerns the programming language specifications (solidity) and for the existence of tools for testing and code quality evaluation. Finally, Ethereum is actually the most popular; it is tested and supports a flexible programming language for writing smart contracts. To write and deploy the smart contracts we used Remix, a web-based integrated development environment that includes a text editor, a debugging system,

a compiling engine, and a connection manager that us allows to choose between the Ethereum “main net,” one of the test nets. If the smart contract compiles correctly, the IDE produces the bytecode that will be deployed in the blockchain and the associated ABI, which contains the names and the interfaces of public variables and functions.

Our smart contracts inherit the the ERC721 smart contract provided by Openzeppelin on GitHub, which enforce the satisfying of security requirements.

We tested the behavior of our smart contract by means of the definition of the expected behavior, described as UML sequence diagrams. The testing phase allowed us to systematically individuate incorrect behavior and fix it. To limit the cost of the transactions, some adjustments in the structure of contracts simplified and decreased to the essential interactions between the two contracts. For the same reason, we configured the setters of OSP and job tokens as atomic. In our implementation, a single function allows the filling of all token details at once. This limits the number of transactions that the construction company has to send to the contracts to publish its OSP.

We studied the strategy to allow the update of the smart contracts, in a simple, cheap, and secure way. We implement

the replacement of a smart contract by means of a deactivation function. To allow the deactivation of the smart contract we implemented an enum-type variable with two tags. Only the System Owner can deactivate the smart contract and that only if a new version is already deployed. After the deactivation, the old contract allows the reading of stored data but avoids any modifications. Thanks to the definition of function modifiers, we can write a specific set of requirements only once for a given set of functions. They also avoid the writing of the same execution control (such as the keyword `require`) in several functions. For instance, all the functions that modify the OSP data could be executed only by the construction company who create it. So, the modifier `onlyOSPowner` inputs the id of the token OSP we want to modify and checks whether the sender of the transaction (called `msg.sender`) is the owner of the token. If the two addresses do not match, the body of the modifier blocks the execution. Specific importance takes also the token ERC721, which allows to assign an owner to each token and to authorize a specific address (in our case the smart contract address) to manage the same token. For what concerns the implementation of the out-of-chain component, we implemented the web-based user interfaces in the Html and javascript languages, by using the libraries jQuery and Bootstrap to define graphical elements and their behaviors, and the library web3.js (web3js.readthedocs.io) to connect the web interface with the Ethereum blockchain. The library web3.js allows the connection with the contracts once their ABI are imported. To make possible the communication between a web browser and the blockchain, we can use a light node directly connected to the blockchain, or we can use a “bridge,” such as metamask, which is a browser extension that includes an Ethereum wallet. A mobile version of metamask is currently under development.

5.1. The dApp

We estimated the cost of the use of smart contracts and of the interaction with the Ethereum blockchain. Currently, a unit of gas has the value of 1 Gwei (1ETH = 10^9 Gwei), and 1 Ether is exchanged for about 250 US dollars. The deployment of the smart contracts is the most expensive operation. The current version of the smart contract “JobManager” is 17,696 bytes long, and the deployment consumes 3,846,696 units of gas, which is equivalent to 0.0038467 ETH (about 0.99 USD) of fee. The current version of the smart contract “OSPManager” is 13,612 bytes long, and the deployment consumes 2,967,314 units of gas or 0.0029673 ETH (0.76853 USD). Creating a new token OSP costs 184,266 units of gas (about five dollar cents), and the filling of OSP data (without workers) costs 128,590 unit of gas. Add a job position costs 184,181 units of gas and filling job data costs 183,944 unit of gas. Hiring a worker costs only 78,172 unit of gas.

Listing 1 report the output of the “getters” of the tokens OSP and job. Smart contract return formatted data to be presented in the user interface. **Figure 10** shows a screen of the user interface which presents OSP data. The screen shows the same data reported in Listing 1.

Listing 1 | An example of OSP and job data returned by smart contracts.

```
//Details of OSP 1123
0: address: constructionCompany 0
   xF20541F9D55F49F9C90C00ffED353Dd87638f555
1: string: siteAddress Work Street, 12
2: uint16: duration 45
3: string: workDirector Mario Rossi - AL123BSS
4: string: foreman Alice Smart - CI223344
5: string: firstAidOfficier Bob Contract - XA001122
6: uint16: numberOfWorkers 3
7: uint8: state 1
uint256[]: jobList [11,12,14]

//Details of Job 12 (vacant position)
0: uint256: OSPid 1122
1: string: name Unskilled Worker Position
2: string: category 8.6.2.1.0
3: string: position Unskilled Worker
4: string: duration 40
5: uint8: state 0
6: uint32: salary 4300
7: uint32: workdays 2
8: address: hiredWorker 0
   x0000000000000000000000000000000000000000000000000000000000000000
9: address: constructionCompany 0
   xF20541F9D55F49F9C90C00ffED353Dd87638f555
10: uint256: expirationDate 1582149433
```

6. DISCUSSION

In this section, we evaluate our design choices in terms of the BOS system for the management of workers in construction sites. We will analyze some aspects elicited from the sustainability analysis and others from the implementation phase. Finally, we will compare our system with a successful case of a platform for managing workers in construction sites, namely, the French Bativie platform.

6.1. System Implications and Impact

Our system focuses on safety requirements that the construction company must perform. The system provides a fast and transparent procedure to publish safety documents and allows an easy and trusted procedure to verify the correctness of documents by labor inspectors. This is a desired and crucial effect of the system that in turn leads to several enabling and structural effects. One of these is the reduction of government costs for inspections due to digital and always available documents, not only regarding the construction site but also the presence of workers in the site. The labor inspectorate can set up a supervision strategy to generate alerts in case of anomalies in the data flow and to decide to perform on site controls. We also expect that construction companies which use the BOS system for building workers have a competitive advantage because they should look more attractive to the Site Owners. This because a construction company which uses the system is certainly a more reliable and modern company. As a consequence, an increase in site owners choosing virtuous companies due to the decrease in injury at work is expected. This is a structural effect that brings attention to worker safety, but it also leads to a reduction in costs for injury at work. Another structural effect which concerns the environmental dimension is the reduction of building abuses. The more companies use the system, the fewer companies will be willing to build outside the

BOS-BW Profile [Read OSP](#) [New OSP](#)

BCC - Big Construction Company inc.
0xF20541F9D55F49F9C90C00ffED353Dd87638f555

Operational Safety Plan
ID 1123 Date: Feb 17, 2020

Site Address	Work Street, 12
Duration	45 days
Work Director	Mario Rossi (document: AL123BSS)
Foreman	Alice Smart (document: CI223344)
First Aid Officer	Bob Contract (document: XA001122)

[OSP](#) / [Jobs](#) / [Worker list](#)

Job ID	Worker address	Position	OSP ID	Vacant
11	0x35771...924	Bricklayer	1123	
12		Unskilled worker	1123	YES
14	0x2A122...923	Concrete mixer operator	1123	

Job Preview
Safety prescriptions: 43.39.01
Duration 40 days
Salary: 5700
Workdays: 2
[More Info](#)

« 1 2 3 »

FIGURE 10 | A User Interface screen, which provides the summary of the OSP ID 1123 and the list of workers employed in the construction site.

law. In addition, virtuous companies can connect the BOS for building workers with a blockchain-based tracking system for dangerous construction materials and for waste management. Construction and Demolition Waste Management (CDWM) includes several aspects, such as those social, environmental, economic, institutional, and political, by dealing with technology, engineering, management, legislation, and policies. According to Ratnasabapathy et al. (2019), the blockchain technology can facilitate waste data management and trading of waste by improving the operational, connecting generators with waste consumers without an intermediary. Proper management of construction materials is mandatory for the safeguarding of the environment and also for the health of workers.

In terms of the technical dimension, we already paid attention to the need of a well-organized user interface to break possible

barriers at the entrance. We can add that the existing Tax Assistance Centers could play a facilitating role for those who have no knowledge of blockchain technology. New appropriately trained staff will be able to guide the construction company in creating a blockchain address, purchasing cryptocurrency and purchasing OSP tokens. In addition, the use of the system leads to a remodeling of the role of employment centers and temporary agencies, creating a more direct relationship between construction companies and workers and a cost reduction. As mentioned, in this study, we implemented the system's smart contracts as non-fungible ERC721 tokens. We adopted the concept of code reuse to save time and effort both in smart contract design and their implementation and to exploit publicly available state-of-the-art implementations of ERC721 tokens, which include last Solidity security recommendation and

efficiency patterns. However, there are different specializations and extensions of non-fungible tokens, which allow for simplification of the implementation of certain types of smart contract systems (see for instance ERC875 and ERC1155). For the purposes of this research, these do not offer features that make them preferable.

6.2. System Comparison

As described in section 3, a number of digital platforms exist in the market to enhance the labor market or to manage recruitment and payment of workers. Among them, we have chosen to compare our system with Bativigie for two reasons. The first is the need to compare the features of our proposal with those of a complete and well-used system. The second is the need to evaluate the implications and potential effects of our system with those of a system operating in the same domain. Bativigie is a well-documented software system that is currently in use and operating in the domain of safety and temporary work in building. In the following, we subdivided our comparison into five sections that represents the five dimensions of sustainability.

6.2.1. Social Dimension: Goals and Trust

Our system and Bativigie share the common goal of making the hiring of workers at construction sites easier, in full compliance with applicable laws, and making available for labor inspections information about working relationships. Bativigie was launched in France in 2015 to provide “solutions to fight illegal work for construction and building professionals” with the main goal of simplifying supervision and inspections in construction sites. It provides a tool for site owners or their delegate in order to facilitate the supervising role that they must take in the construction site by law. This system allows for the construction site’s registration in the system and employment management. In addition to this, our system allows construction companies to add vacant positions and workers to apply for job offers. In this way, construction companies can find autonomously qualified workers. Thanks to blockchain technology, our system improves transparency and trust of building work relationships, including work agreements and safety prescriptions. This improves the trust between workers and employers, makes labor inspection trustworthy, and improves the relationship between the State and construction stakeholders.

6.2.2. Individual Dimension: Safety, Privacy, and Agency

Both systems lead to the direct effect of increasing worker safety thanks to the creation of regular contracts and compliance with workplace safety laws. In Bativigie, construction companies add workers to the construction site data within the system. A specific section allows the hiring of temporary workers. Workers can access their report and personal data, which are stored in the Bativigie server. Our system is designed to improve worker agency. It allows each worker to create their profile and to become an active player. Workers can add competencies, professional qualifications, and experiences. In our system, all personal data

are property of the worker, and they can make their profiles available for use by the construction company.

6.2.3. Technical Dimension: Architecture and Features

Bativigie is a typical client-server system. A centralized database stores construction site data, and a specific permission handler allows a third party to read the stored data. By contrast, our system is decentralized and blockchain based. This is a key factor for two reasons. The first is due to the nature of the blockchain-based system. There is no central organization to which the actors of the system must refer (apart from the acquisition of the token, as explained below). Furthermore, the system cannot be interrupted since it will always be accessible as long as there are active nodes in the blockchain. The second concerns the fact that the ownership of the data is completely reserved to its owner, identified with a blockchain address. For example, construction companies own their OSP tokens, workers own their personal data, etc. Owners of the data will allow third parties to read or modify them. To make easier the use of the system, Bativigie provides a mobile application with which the construction company can report any event that occurred on the construction site that affects workers, such as accidents and the like, and store results inside the database. Our BOS allows the construction company to verify the presence of a given worker at the construction site and record it in the smart contract through the web interface. Labor inspectors can read data stored in the blockchain to check regularity.

6.2.4. Economic Dimension: Business Model

For what concerns the business model, Bativigie requires the payment of a fee if which the amount depends on the value of the construction site and on the size of the construction company. In our case, the business model is based on selling the OSP and on job tokens. We can define a price for an empty OSP token and a price for an empty job token. In this way, the total price of the use of the system will have a fixed fee plus a variable fee, which depends on the number of workers on the construction site.

6.2.5. Environmental Dimension: Carbon Footprint and Land Conservation

A centralized system like Bativigie has a systemic carbon footprint due to the IT architecture and company organization. Conversely, as previously discussed, our BOS system has a carbon footprint that depends on the volume of transitions performed and on the characteristics of the blockchain protocol. In the long term, the two systems can both lead to environmental protection. Thanks to a simplification of inspection process, there will be a possible reduction in construction abuse and the possibility of reducing the overall carbon footprint of the construction sector.

7. CONCLUSIONS

In this work, we presented a software system for the management of temporary employment of Building Workers and Construction Companies based on the blockchain technology that natively supports decentralization, transparency, security, and tamper proof. The advantages provided by

this technology with respect to similar centralized software systems are transparency for labor inspectors, data integrity and immutability, tamper-proof time stamps for registering any activity, and the decentralized execution of smart contracts that automatically implement contract clauses respecting legal requirements in the domain field. Focusing on the Italian case, we included the implementation of the OSP (Operational Safety Plan) into the system's requirements so that legal policies about safety at the building site are automatically implemented through smart contracts.

We corroborate the system with a sustainability analysis in five dimensions that enables an a-priori evaluation of the impact of the approach and contributes to the requirements analysis since the very beginning in order to avoid pitfalls and mistakes in the future system development. We developed the system taking into account the BOSE (Blockchain Oriented Software Engineering) prescriptions through the application of the ABCDE (Agile Block-Chain Dapp Engineering) methodology in order to identify the system's actors and to determine the system's requirements in form of User Stories. We also analyzed by means of UML Contract Diagrams the interactions and the data structure of the Smart Contracts developed and deployed in the blockchain. The solution is available for workers and construction companies as a decentralized web application. We performed an analysis of the practical usage of our systems and computed the costs of use. We finally presented a comparison with a similar software system, "Bativigie," which adopts a centralized approach, opposed to the decentralized one offered by the blockchain technology, and a different business model where a fee is required to use the system, opposed to our blockchain approach where tokens are used as a mean of payment.

Our results show the feasibility of the approach with reduced costs, enhanced transparency, and security. The proposed system

represents a way to meet the current legal requirements relating to the documentation on worker safety on construction sites, managing them in a dematerialized and decentralized way. It is also a tool for the construction company that allows the selection of candidates, monitoring the work done and keeping track of costs, simplifying the bureaucracy related to temporary hiring.

DATA AVAILABILITY STATEMENT

The datasets generated for this study are available on request to the corresponding author.

AUTHOR CONTRIBUTIONS

AP conceived the original idea and performed the sustainability analysis. GB performed the state of the art and the analysis of study cases. AP and GB drafted the manuscript. AP and GL designed the BOS system and developed smart contracts. RT contributed to the final version of the manuscript. MM and RT supervised the project. All authors contributed to the article and approved the submitted version.

FUNDING

This research was partially funded by Ministero dello Sviluppo Economico (Italia) - Asse 1 - Innovazione, azione 1.1.3 del Programma Operativo Nazionale Imprese e Competitività 2014-2020 FESR Fondo per la Crescita Sostenibile - Sportello Agrifood - D.M. 05/03/2018 Capo III under the project ABATA - Applicazioni della Blockchain per l'Autenticità e la tracciabilità Alimentare.

REFERENCES

- Alestra, L. (2018). "Rapporto Annuale Dell'attività di Vigilanza in Materia di Lavoro e Legislazione Sociale. Anno 2019". Ispettorato Nazionale Del Lavoro. Available online at: <https://www.ispettorato.gov.it/it-it/in-evidenza/Documents/Rapporto-annuale-2019-attivita-di-vigilanza-INL.pdf>
- Al-Htaybat, K., Hutaibat, K., and von Alberti-Alhtaybat, L. (2019). Global brain-reflective accounting practices. *J. Intellect. Capital*. 20, 733–762. doi: 10.1108/JIC-01-2019-0016
- Bach, L., Mihaljevic, B., and Zagar, M. (2018). "Comparative analysis of blockchain consensus algorithms," in *2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)* (Opatija: IEEE), 1545–1550. doi: 10.23919/MIPRO.2018.8400278
- Beck, K., Beedle, M., Van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., et al. (2001). *Manifesto for Agile Software Development*. Available online at: www.agilemanifesto.org
- Becker, C., Betz, S., Chitchyan, R., Duboc, L., Easterbrook, S. M., Penzenstadler, B., et al. (2016). Requirements: the key to sustainability. *IEEE Softw.* 33, 56–65. doi: 10.1109/MS.2015.158
- Becker, C., Chitchyan, R., Duboc, L., Easterbrook, S., Penzenstadler, B., Seyff, N., et al. (2015). "Sustainability design and software: the Karlskrona manifesto," in *2015 IEEE/ACM 37th IEEE International Conference on Software Engineering* (Karlskrona: IEEE), 467–476. doi: 10.1109/ICSE.2015.179
- Caruso, A. R. (2013). *La Vigilanza Sul Lavoro Negli altri Paesi Europei*. Working Paper ADAPT no. 143. Modena: ADAPT University Press.
- Destefanis, G., Marchesi, M., Ortu, M., Tonelli, R., Bracciali, A., and Hierons, R. (2018). "Smart contracts vulnerabilities: a call for blockchain software engineering?" in *2018 International Workshop on Blockchain Oriented Software Engineering (IWBOSE)* (Campobasso: IEEE), 19–25. doi: 10.1109/IWBOSE.2018.8327567
- Diginex (2019). *Un Migration Agency Will Use Blockchain Technology in Effort to Stamp Out Illegal Fees Charged to Migrant Workers in Hong Kong*.
- Duboc, L., Betz, S., Penzenstadler, B., Kocak, S. A., Chitchyan, R., Leifler, O., et al. (2019). "Do we really know what we are building? Raising awareness of potential sustainability effects of software systems in requirements engineering," in *2019 IEEE 27th International Requirements Engineering Conference (RE)* (Jeju Island: IEEE), 6–16. doi: 10.1109/RE.2019.00013
- European Construction Sector Observatory (2019). *Analytical Reports*.
- França, A., Neto, J. A., Gonçalves, R., and Almeida, C. (2019). Proposing the use of blockchain to improve the solid waste management in small municipalities. *J. Clean. Product.* 244:118529. doi: 10.1016/j.jclepro.2019.118529
- Genç, R. (2017). The importance of communication in sustainability & sustainable strategies. *Proc. Manufactur.* 8, 511–516. doi: 10.1016/j.promfg.2017.02.065
- Heiskanen, A. (2017). The technology of trust: How the internet of things and blockchain could usher in a new era of construction productivity. *Construct. Res. Innovat.* 8, 66–70. doi: 10.1080/20450249.2017.1337349

- Hijazi, A. A., Perera, S., Alashwal, A., and Calheiros, R. N. (2019). "Blockchain adoption in construction supply chain: a review of studies across multiple sectors," in *Presented at the CIB World Building Congress*.
- Inail (2019). *Andamento Degli Infortuni sul Lavoro e delle Malattie Professionali*.
- Jeong, W.-Y., and Choi, M. (2019). Design of recruitment management platform using digital certificate on blockchain. *J. Inform. Process. Syst.* 15, 707–715. doi: 10.3745/JIPS.03.0121
- Keršič, V., Štukelj, P., Kamišalić, A., Karakatić, S., and Turkanović, M. (2019). "A blockchain-and AI-based platform for global employability," in *International Congress on Blockchain and Applications* (Hong Kong: Springer), 161–168. doi: 10.1007/978-3-030-23813-1_20
- Liyuan, L., Meng, H., Yiyun, Z., and Reza, P. (2019). "E² c-chain: a two-stage incentive education employment and skill certification blockchain," in *2019 IEEE International Conference on Blockchain (Blockchain)* (Seoul: IEEE), 140–147. doi: 10.1109/Blockchain.2019.00027
- Marchesi, M., Marchesi, L., and Tonelli, R. (2018). "An agile software engineering method to design blockchain applications," in *14th Central and Eastern European Software Engineering Conference Russia* (Moscow: ACM), 3. doi: 10.1145/3290621.3290627
- Michaelides, M. P. (2018). The challenges of ai and blockchain on hr recruiting practices. *Cyprus Rev.* 30, 185–197.
- Nguyen, Q. K. (2016). "Blockchain-a financial technology for future sustainable development," in *2016 3rd International Conference on Green Technology and Sustainable Development (GTSD)* (Kaohsiung: IEEE), 51–54. doi: 10.1109/GTSD.2016.22
- Onik, M. M. H., Miraz, M. H., and Kim, C. (2018). "A recruitment and human resource management technique using Blockchain technology for industry 4.0," in *Smart Cities Symposium 2018* (Bahrain), 1–6. doi: 10.1049/cp.2018.1371
- Peisl, T., and Shah, B. (2019). "The impact of blockchain technologies on recruitment influencing the employee lifecycle," in *European Conference on Software Process Improvement* (Edinburgh: Springer), 695–705. doi: 10.1007/978-3-030-28005-5_54
- Penzenstadler, B., Duboc, L., Venters, C. C., Betz, S., Seyff, N., Wnuk, K., et al. (2018). Software engineering for sustainability: find the leverage points! *IEEE Softw.* 35, 22–33. doi: 10.1109/MS.2018.110154908
- Penzes, B., KirNup, A., Gage, C., Dravai, T., and Colmer, M. (2018). *Blockchain Technology in the Construction Industry: Digital transformation for high productivity*. Technical Report. Istitution of Civil Engineers (ICE). doi: 10.13140/RG.2.2.14164.45443
- Perera, S., Nanayakkara, S., Rodrigo, M., Senaratne, S., and Weinand, R. (2020). Blockchain technology: is it hype or real in the construction industry? *J. Indus. Inform. Integrat.* 17:100125. doi: 10.1016/j.jii.2020.100125
- Pinna, A., and Ibba, S. (2018). "A blockchain-based decentralized system for proper handling of temporary employment contracts," in *Science and Information Conference* (London: Springer), 1231–1243. doi: 10.1007/978-3-030-01177-2_88
- Pinna, A., Ibba, S., Baralla, G., Tonelli, R., and Marchesi, M. (2019). A massive analysis of Ethereum smart contracts empirical study and code metrics. *IEEE Access* 7, 78194–78213. doi: 10.1109/ACCESS.2019.2921936
- Porru, S., Pinna, A., Marchesi, M., and Tonelli, R. (2017). "Blockchain-oriented software engineering: challenges and new directions," in *2017 IEEE/ACM 39th International Conference on Software Engineering Companion (ICSE-C)* Buenos Aires, 169–171. doi: 10.1109/ICSE-C.2017.142
- Ratnasabapathy, S., Perera, S., and Alashwal, A. (2019). "A review of smart technology usage in construction and demolition waste management". in *Proceedings of the 8th World Construction Symposium*, eds Y. G. Sandanayake, S. Gunatilake, and A. Waidyasekara (Colombo), 45–55. doi: 10.31705/WCS.2019.5 Available online at: <https://2019.ciobwcs.com/papers>
- Saberi, S., Kouhizadeh, M., Sarkis, J., and Shen, L. (2019). Blockchain technology and its relationships to sustainable supply chain management. *Int. J. Product. Res.* 57, 2117–2135. doi: 10.1080/00207543.2018.1533261
- Sarda, P., Chowdhury, M. J. M., Colman, A., Kabir, M. A., and Han, J. (2018). "Blockchain for fraud prevention: a work-history fraud prevention system," in *2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)* (New York, NY: IEEE), 1858–1863. doi: 10.1109/TrustCom/BigDataSE.2018.00281
- Stoll, C., Klaufen, L., and Gallersdorfer, U. (2019). The carbon footprint of bitcoin. *Joule* 3, 1647–1661. doi: 10.1016/j.joule.2019.05.012
- Tanana, D. (2019). "Decentralized labor record system based on wavelet consensus protocol," in *2019 International Multi-Conference on Engineering, Computer and Information Sciences (SIBIRCON)* (Novosibirsk: IEEE), 496–499. doi: 10.1109/SIBIRCON48586.2019.8958051
- Tezel, A., Papadonikolaki, E., Yitmen, I., and Hilletoft, P. (2019). "Preparing construction supply chains for blockchain: an exploratory analysis," in *CIB World Building Congress 2019 Constructing Smart Cities* (Hong Kong).
- Truby, J. (2018). Decarbonizing bitcoin: law and policy choices for reducing the energy consumption of blockchain technologies and digital currencies. *Energy Res. Soc. Sci.* 44, 399–410. doi: 10.1016/j.erss.2018.06.009
- Wang, X., Wu, H., Sun, W., and Zhao, Y. (2019). "Towards generating cost-effective test-suite for Ethereum smart contract," in *2019 IEEE 26th International Conference on Software Analysis, Evolution and Reengineering (SANER)* (Hangzhou: IEEE), 549–553. doi: 10.1109/SANER.2019.8668020
- Zwitter, A., and Herman, J. (2018). "Blockchain for sustainable development goals:#Blockchain4SDGs-report 2018," in *Blockchain4SDGs Workshop* (Groningen: Rijksuniversiteit).

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Copyright © 2020 Pinna, Baralla, Lallai, Marchesi and Tonelli. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.