Journal of
Theoretical and Applied Electronic Commerce Research

MDPI

Article

A Reference Architecture for Blockchain-Based Crowdsourcing Platforms

Yiwei Gong [1,*], Sélinde van Engelenburg [2] and Marijn Janssen [2]

[1] School of Information Management, Wuhan University, Wuhan 430072, China
[2] Faculty of Technology, Policy and Management, Delft University of Technology, 2628 BX Delft, The Netherlands; S.H.vanEngelenburg@tudelft.nl (S.v.E.); M.F.W.H.A.Janssen@tudelft.nl (M.J.)
* Correspondence: yiweigong@whu.edu.cn

Abstract: Companies increasingly tender knowledge-intensive tasks using crowdsourcing platforms to gain access to scarce knowledge and skills otherwise out of reach, and in this way, gaining competitive advantage. Despite its potential, existing crowdsourcing platforms encounter several challenges, including (1) fragmentation of expertise, as there are many platforms, (2) distrust between task providers and crowdsourcing participants, as identity and past performance are often not known, and (3) inability to learn from experience due to a lack of openness. A reference architecture for blockchain-based knowledge-intensive crowdsourcing platforms to mediate transactions between demand and supply of knowledge is designed in this paper to overcome these challenges. A design science research method is followed to develop the architecture. The reference architecture shows how blockchain and smart contract components can be integrated to support and coordinate knowledge-intensive crowdsourcing activities. By removing traditional e-commerce intermediaries, blockchain reduces search friction, knowledge transfer costs, and cheating by task providers or crowdsourcing participants.

Keywords: blockchain; smart contract; crowdsourcing; knowledge-based view; search friction; market; disintermediation; design science

Citation: Gong, Y.; van Engelenburg, S.; Janssen, M. A Reference Architecture for Blockchain-Based Crowdsourcing Platforms. *J. Theor. Appl. Electron. Commer. Res.* **2021**, *16*, 937–958. https://doi.org/10.3390/jtaer16040053

Received: 11 January 2021
Accepted: 3 March 2021
Published: 7 March 2021

1. Introduction

Buying and selling knowledge-intensive tasks and services on the market has become a common practice for many organizations to gain on-demand access to a variety of expertise and knowledge [1,2]. Ranging from small start-ups to those listed in Fortune 500, increasingly, companies are making use of crowdsourcing to access external knowledge and skills [3,4]. These companies are task providers (TPs), having a knowledge demand and are buyers in market terms, whereas those who fulfill the task are crowdsourcing participants (CPs), e.g., knowledge providers being suppliers in market terms. In such crowdsourcing, TPs follow a tendering process in which they specify their needs and ask for offers from CPs on a market platform. CPs can provide their bids to the TP who can select the best offers and fulfill the e-commerce transaction. Today, an increasing number of knowledge-intensive tasks are crowdsourced, as companies need new expertise or capacity to address fast-changing technical and business environments. Knowledge-intensive crowdsourcing is regarded as one of the most promising areas for crowdsourcing, given its critical role in today's knowledge-based economy [5,6].

Crowdsourcing can be viewed as an online and decentralized problem-solving model [7]. Typically, crowdsourcing starts with the need for a TP and the publishing of a request for proposals through an online market platform, with a description of the service or product needed, its expected duration, and possibly, a range of payment options. Potential CPs then bid on the task by submitting their proposals. If more than one proposal is received for a task, the proposals will be evaluated and compared, and a candidate will be selected to carry out the task. Once the tasks have been fulfilled, the electronic transaction can be

settled, and the TP can decide to accept and pay for the work or refuse the deliverable if it does not fulfill their requirements. There are many discrete crowdsourcing platforms on the internet supporting such a tendering process, such as Amazon Mechanical Turk (AMT, www.mturk.com), InnoCentive (www.innocentive.com) and Upwork (www.upwork.com). The majority of crowdsourcing platforms run their business on centralized servers with a revenue model that demands a commission ranging from 5% to 20% of the trade [8].

While the demand for knowledge-intensive tasks is growing, there are three key challenges that hinder TPs from realizing the expected results. These challenges, include (1) fragmentation of CP expertise, (2) lack of trust between TPs and CPs, and (3) inability to learn from historical data due to a lack of openness of proprietary platforms.

Firstly, there is no single market, and many crowdsourcing platforms for conducting knowledge-intensive tasks exist. These many platforms result in scattering and fragmentation of demand and supply of expertise over multiple platforms. A market needs a minimal level of activities to be viable, whereas some platforms receive relatively few submissions. Furthermore, on some platforms, the quality of bids is an issue, as the majority of these bids do not meet the expectations of TPs [9]. Secondly, TPs and CPs do not know each other. They might distrust each other, as they do not know if the CP is an imposter or has the right expertise and if the TPs will pay. Thirdly, most platforms do not provide the required openness to allow users to view past performance, experiences of others, similar requests, and fraud.

Central to overcoming the above challenges for knowledge-intensive crowdsourcing is the creation of a trusted and open crowdsourcing network connecting demand and supply [10,11]. This requires storing, distributing, and updating the information on knowledge-intensive tasks and participants in a standardized way, and making user engagement transparent to avoid undesirable behavior. *Blockchain* can create trust due to its decentralized storage of data records, which are hard to manipulate and using consensus mechanisms to ensure that all changes need to be confirmed, by the participating nodes [12]. The information about every transaction completed in blockchain is recorded in a distributed ledger, which is shared among and available to all nodes [13]. The distributed nature and consensus mechanism ensure that data cannot be altered by a single party. Blockchain is decentralized and can take over the roles of traditional platform intermediaries, as data cannot be easily tampered or manipulated [14]. Blockchain provides a distributed data storage and verification solution, but it does not offer much functionality in data processing for implementing business logic. As a complementary technology, smart contracts on blockchains provide a general-purpose programmable infrastructure to deploy and run these programs [15]. Using smart contracts, companies will be able to automate the terms of agreement [16]. The automatic settlement of transactions will further lower coordination costs for companies and realize a disintermediated business model [17]. Smart contracts can be used for arranging the agreements between TPs and CPs.

Blockchains could enable TPs to access the history data of CPs to assess their past performance, while smart contacts could allow TPs to define their own business rules in searching, selecting, and interacting with them. By exposing transactional data to all nodes for verification, blockchain-based applications are more transparent than crowdsourcing platforms in which executive transactions are centralized and run by a third party [18]. The premise of blockchain is that no traditional intermediaries are needed for ensuring trust and conducting transactions [19].

Despite several attempts to use blockchain to decentralize crowdsourcing in specific contexts e.g., [20–22], it is not clear what such a distributed solution would look like. This raises the question of whether the premise of disintermediation using blockchain is right. Furthermore, the premise that no intermediaries are needed remains unproven. We will investigate this premise, in this paper, by evaluating whether a blockchain can make intermediaries redundant.

Although there have been many efforts and substantial investments, many blockchain projects fail, as they do not take into account the situation at hand or only provide limited

benefits [23]. In contrast, in this research, a reference architecture is developed to enable the distribution of tasks to a high number and variety of crowdsourcing participants without the involvement of a market intermediary. A *reference architecture* consists of a set of principal design decisions, guidance for implementation, and a system structure and components 'that are simultaneously applicable to multiple related systems, typically within an application domain' [24] (p. 58). This reference architecture presents the essential and high-level design of blockchain-based crowdsourcing solutions. This paper provides insight into the decentralized and disintermediated nature of blockchain-based solutions and how they might transform the current crowdsourcing paradigm. The *knowledge-based view* (KBV) of the firm and the *theory of search friction*, are used to theorize why a blockchain-based paradigm creates benefits for mediating electronic transactions. By this reference architecture we intend to *solve the before mentioned three problems with the existing crowdsourcing platforms to improve knowledge-intensive crowdsourcing*.

The remainder of this paper is structured as follows. Section 2 explains the concept of KBV of the firm and the theory of search friction and introduces related works. In Section 3 the design science approach is presented. Whereas, Section 4 provides a step-by-step description of how the design science approach was implemented, including the design of the reference architecture, an illustrative case to demonstrate how the blockchain-based design could support core activities in knowledge-intensive crowdsourcing, and the evaluation of the architecture by checking if the requirements have been met by the designed reference architecture. Section 5 presents implications, and Section 6 presents conclusions.

2. Background

The KBV of the firm and the theory of search friction are introduced in this background section to serve as our theoretical basis. Based on these theories, we theorize the benefits of a blockchain-based solution. Furthermore, we outline the related works in blockchain-based crowdsourcing.

2.1. KBV of the Firm and Search Friction

The advance of internet technology has had a profound impact on the structure of conducting transactions among businesses [25]. For knowledge-intensive business services (KIBS), such as IT services, consultancies, and R&D, the change in the shape of organizational networks can be explained from the KBV. The KBV of the firm considers knowledge to be the most strategically significant resource of a firm and to be the source of competitive advantage [26]. In the KBV of the firm, organizations are viewed as social communities specializing in efficient creation and transfer of knowledge [27]. The knowledge transferability of organizations is regarded as a critical determinant of their capacity to confer sustainable competitive advantage [28]. Studies of knowledge transfer often emphasize *knowledge transfer costs*, which refers to the time and resources required to transfer knowledge in social relationships from the sources of knowledge and to the recipients of knowledge [29].

Knowledge transfer through intermediaries, such as the current crowdsourcing platforms, contributes to resolving the problems associated with coordinating TPs and CPs. The knowledge transfer costs of using crowdsourcing platforms can be explained by the theory of search friction, which was first developed in relation to the economics of the labor market [30]. The *theory of search friction* is based on the hindrances in the market in connecting buyers and sellers, and refers to the time and costs needed to find matches [31]. The limitations of cognitive competence, as well as the limited resources and time to search for suitable knowledge suppliers, places boundaries on the information that TPs or CPs can access and process in time [32].

Research indicates that greater diversity in crowdsourcing expertise leads to a higher probability of finding a winning solution [33], while access to a larger crowd will result in a greater chance that there will be someone who is able to help [3]. This can explain why single crowdsourcing platforms often fail to fulfill the expectations of TPs. Instead, access to a larger volume and variety of skills offered increases the chance of finding a high-quality

supplier. As a result, TPs often use multiple platforms at the same time to reach more CPs. This results in a higher level of search friction. Consequently, using multiple crowdsourcing platforms simultaneously results in higher knowledge transfer costs, which is due to the need for manual and repeated work to publicize the tasks and select appropriate CPs or tasks. Therefore, the expectation is that if the current crowdsourcing business model could be disintermediated and a distributed market with direct access between TPs and CPs is created, the knowledge-transfer costs of knowledge-intensive crowdsourcing will be lowered. Disintermediation refers to the situations that demand and supply are directly connected without the involvement of a third party.

KIBS companies prefer efficient knowledge transfer [34]. We can, thus, theorize that a disintermediated paradigm will further reduce knowledge transfer costs and allow companies to access a large number of knowledge suppliers having the right talent, knowledge, and skills they need [16]. The emergence of blockchain technology will allow for the implementation of this disintermediated business model in crowdsourcing.

2.2. Related Work

There are several attempts to use blockchain to facilitate crowdsourcing. For example, Lu, Tang and Wang [11] designed a blockchain-based decentralized crowdsourcing system that allows anonymous participation. In a similar vein, Zou, Ye, Qu, Wang, Orgun and Li [22] proposed a consensus protocol, using blockchain as the underlying technology, to enable tracing transactions for service contracts and dispute arbitration. Both Yang, Zhu, Liang, Zhou and Deng [21] and Lin, et al. [35] proposed a blockchain privacy-preservation crowdsourcing system to prevent breaches of privacy. Whereas, Hu, et al. [36] used blockchain to build a reputation based decentralized knowledge sharing system to exploit the copyright protection of the knowledge owner and to achieve the paid-for content service for accessing knowledge. Blockchain technologies are also proposed to be combined with Internet-of-Things or cloud computing technologies to improve privacy protection and reputation management [37,38]. These solutions aim to effectively overcome the negative effects of a dishonest centralized intermediary with accountability, privacy, or copyright in concern. In addition, ensuring trust has been indicated as a possible benefit for making use of blockchain technologies in crowdsourcing [39]. The works mentioned above show the potential for blockchain-based e-commerce transactions, but do not address the three challenges. Furthermore, those studies often only emphasize technical solutions, while theoretical notions highlight the importance of changing crowdsourcing business models. Our reference architecture integrates the necessary business processes of the new crowdsourcing business model and blockchain-related technological components that are required to support these processes.

3. Research Approach

We followed a design science approach to develop and evaluate a reference architecture for blockchain-based knowledge-intensive crowdsourcing solutions. According to Hevner and Chatterjee [40] (p. 5): "Design science research is a research paradigm in which a designer answers questions relevant to human problems via the creation of innovative artifacts, thereby contributing new knowledge to the body of scientific evidence. The designed artifacts are both useful and fundamental in understanding that problem". Walls et al. [41] specified that a design product should draw upon kernel theories in specifying prescriptive hypotheses that enable designers to evaluate whether the designed artifact satisfies the theory. Kernel theories for design product are theories that are well established and recognized by the natural or social sciences for governing design requirements [41]. In this study, the KBV of the firm and the theory of search friction are employed as kernel theories. These theories are used to deduce requirements on the design and for the evaluation of the designed artifact.

Our research approach is based on the six steps of the design science research methodology of Peffers, et al. [42]:

1. Problem identification and motivation: The problem with current crowdsourcing platforms was defined to justify the value of a blockchain-based crowdsourcing system as a solution to efficient knowledge-intensive crowdsourcing. This was based on the practical problem of searching for and interacting with CPs simultaneously using multiple crowdsourcing platforms.
2. Requirement definition: The objectives of the proposed system and accompanying transformation of the business model were determined in this step. High-level requirements for engineering this kind of system were elicited and summarized based on the conceptual implications from the KBV of the firm and the theory of search friction.
3. Design and development: In this step, the system components and their relationship were defined in a reference architecture. The design process in this step is to construct a general solution with functions that could help satisfy the requirements summarized in the second step.
4. Demonstration: Scenarios were developed to illustrate how the core activities in blockchain-based knowledge-intensive crowdsourcing systems were coordinated and supported by architectural components.
5. Evaluation: This step examines whether the problems of the current crowdsourcing platforms can be solved by the designed artifact. Each of the requirements provided in the second step was reviewed and evaluated.
6. Communication: This concerns communication with academic and industrial peers during this study and the publication of this study.

Design science has been used in many domains, including the development of reference architecture. Design science research emphasizes a balance between research rigor and relevance [43]. The research relevance involves determining whether the research subject (in this case, the reference architecture) is closely connected to or appropriate for its intended purpose of solving a practical problem [44]. The relevance of the reference architecture in crowdsourcing can be demonstrated by its effects in accommodating a typical knowledge-intensive crowdsourcing scenario. The research rigor of this study is assured by following a design science methodology and the use of search friction and KBV as kernel theories. Our study employed 'build and evaluate' cycles in an iterative way to ensure that the artefact had utility and validity in terms of both research rigor and relevance. The next section follows the design science approach step by step.

4. A Blockchain-Based Reference Architecture

4.1. Step 1: Problem Identification

In the introduction, we identified the three main problems in knowledge-intensive crowdsourcing. We start by detailing these three problems to better understand the requirements on the artifact.

The first problem for current crowdsourcing platforms is the fragmentation of expertise. Many crowdsourcing platforms do not provide much skill variety [5]. Furthermore, some crowdsourcing platforms have a domestic focus; for example, some local crowdsourcing platforms in China do not provide multiple language interfaces to allow access to non-Chinese TPs and CPs. When a task requires specific knowledge and skills at particular levels, often, a single platform cannot fulfill the need. Companies have to publicize the same task on different crowdsourcing platforms to reach different online communities [45]. From a TP's perspective, using multiple crowdsourcing platforms simultaneously to conduct knowledge-intensive crowdsourcing results in reduced customer satisfaction [10] and more search friction and knowledge transfer costs.

The second problem is the lack of trust between TPs and CPs. It is not just that TPs have to use multiple crowdsourcing platforms simultaneously, as CPs might also have to do the same to search for high-value tasks. Recent studies report that a significant number of users are on different platforms [46] and might have multiple identities on a single platform [47]. This, not only results in additional efforts in coordination for TPs

to verify and select CPs, but can be used for hiding dishonest behavior. Cheating can occur on both sides of crowdsourcing; for example, '*free-riding*' (i.e., dishonest CPs receive payment without contributing to the task) and '*false reporting*' (dishonest TPs refuse to make payments despite the task being completed successfully) [48]. Cheating keeps the CPs away, and in turn, reduces the attractiveness of the platform for TPs, potentially leading to a death-spiral and eventual collapse of the platform [49].

Crowdsourcing literature suggests that trust is critical for the success of crowdsourcing, and it is often a great challenge for the current crowdsourcing platforms [50]. Many platforms attempt to avoid free-riding and false reporting by using reputation-based mechanisms. Still, these mechanisms rely on users wishing to remain on a platform with a single identity. A great challenge for crowdsourcing platforms is to ensure that users maintain persistent identities of buyers and sellers and do not embrace new identities. This might be done by making it expensive for users to abandon their old identities and create new ones, or to create multiple identities [51]. In a multiple platform context, each platform does not collaborate with the other and might employ different policies in regulating user behavior.

Furthermore, crowdsourcing platforms often fail to provide fair judgment in resolving disputes [11]. For example, AMT's policy is criticized for being biased towards TPs over CPs, and AMT's stance gives CPs little recourse when they encounter unfair rejection [52]. A more powerful mechanism to create fairness and trust in crowdsourcing is therefore desired.

The third problem of current crowdsourcing platforms is the lack of openness, which would allow users to specify their own business logic and make use of historical data and learn from past experiences. One challenge to the promotion of knowledge-intensive crowdsourcing is the identification of suitable CPs having the right knowledge and expertise [4,6]. The identification of suitable CPs is especially challenging across different, isolated crowdsourcing platforms. Matching skills to tasks often relies on the functions provided by crowdsourcing platforms. Current crowdsourcing platforms often merely provide simple text searching, ranking, or comment functions to assist TPs through their user interface [10]. Users cannot specify complex logic for information searching and processing information by the assistance of customized tools. Furthermore, the reliability of the data is another issue. Data might simply not be administrated correctly, but also as in-house misbehavior might occur in the interests of the platform, some of its employees, or even attackers, resulting in the vulnerabilities of a single-point of failure [11].

4.2. Step 2: Requirements Definition

Based on the above problems identified in current crowdsourcing platforms and the theoretical notions from the KBV of the firm and search friction theory, we define three general requirements for any blockchain-based crowdsourcing solution. These requirements reflect the objectives of the blockchain-based solution for knowledge-intensive crowdsourcing.

Requirement 1. *Minimize search friction between TPs and CPs.*

This requirement demands a transformation of the current platform-based crowdsourcing business model to a disintermediated one. In the new paradigm, all CPs will be embraced in a single distributed network, instead of having multiple centralized platforms. Broadcasting demand in such a network requires less effort than doing so with multiple platforms. Furthermore, users may hold a copy of the distributed ledger, which would allow them to access, search, and analyze the data related to tasks, TPs or CPs on demand. In this way, the costs of search and matching in allocating supply and demand will be minimized. This should also help in solving the first problem of fragmentation of expertise.

Requirement 2. *Data about user behavior and records should be correct and available to everybody.*

This requirement demands a mechanism to increase the transparency of crowdsourcing processes and results, while at the same time, it should not be possible to tamper with historical records. In this way, reputations can be built, and dishonest behavior will be discouraged. Blockchain-based *smart contracts* are user-defined programs specifying

the rules that govern transactions and are enforced by a network of peers [53]. Smart contracts will make the task content publicly visible, while the contract will be settled and automatically enforced by the distributed network. For example, this should prevent the misuse of crowdsourcing by CPs accepting too many tasks, which would result in low-quality solutions or even task failure. At the same time, this should also protect CPs from not being paid by TPs after the submission of qualified deliverables. The use of smart contracts will balance the power between the two parties. In the long run, this mechanism could increase trust between TPs and CPs, while the second problem would be solved.

Requirement 3. *The platform should be open and flexible to enable users to make use of historical data.*

This requirement demands an openness that allows TPs to specify their own business logic for CP selection, such as searching for and evaluating CPs with their own analysis. The business logic enables specification of how the search process for CPs will be executed. Furthermore, data openness enables users to check the correctness of the data. The root cause of the third problem is the monopoly on data and the inflexible all-in-one functions provided by platforms. The business rule (BR) is a common approach in knowledge-intensive organizations looking for increased flexibility [54]. Given a suitable BR application that could work with the blockchain data, users would have greater flexibility in defining their own criteria and algorithms to use the data. Openness is required to enable the use of user-specified BR in the selection of tasks or CPs.

4.3. Step 3: Design and Development

To describe the designed reference architecture, we employed ArchiMate (version 3.0.1, The Open Group), which is an architecture modeling language and technical standard from The Open Group. The reference architecture (Figure 1) describes the main components of the solution, their relationships, and the functionality they support. Based on the ArchiMate core framework [55], the architecture contains three layers: Business, application, and technology.

4.3.1. Business Layer

An important decision to be made when using a solution based on blockchain technology is about which parties are allowed to have nodes in the blockchain network and what types of permissions these parties can have. In a public blockchain network, any party can be a node and read/write. However, in a private blockchain, the number of parties and nodes are restricted [56]. A similar decision is whether or not to have a 'permissionless' or 'permissioned' blockchain network. In a permissionless blockchain, any node can contribute to consensus and have access to a copy of the ledger, while in a permissioned blockchain there are restrictions [56,57].

To make the blockchain network private or permissioned would require a separate entity to decide who may operate nodes and what nodes can contribute to the consensus. There is no clear advantage of making an effort to do this in our domain. Furthermore, it would require a central party to be involved in the assignment of nodes, while blockchain's decentralized nature is one of the main reasons for selecting this technology. The blockchain network used in the reference architecture, therefore, will be open and permissionless.

The coordination between the two kinds of parties, the TPs and CPs, includes (1) matching between the task and the participant to meet Requirement 1, and (2) evaluation of the solution and transaction of the payment to meet Requirement 2. In addition, BR services allow TPs to specify their own criteria for CP selection. BR services are also provided to CPs for them to specify their own evaluation criteria for task selection. This contributes to meeting Requirement 3.

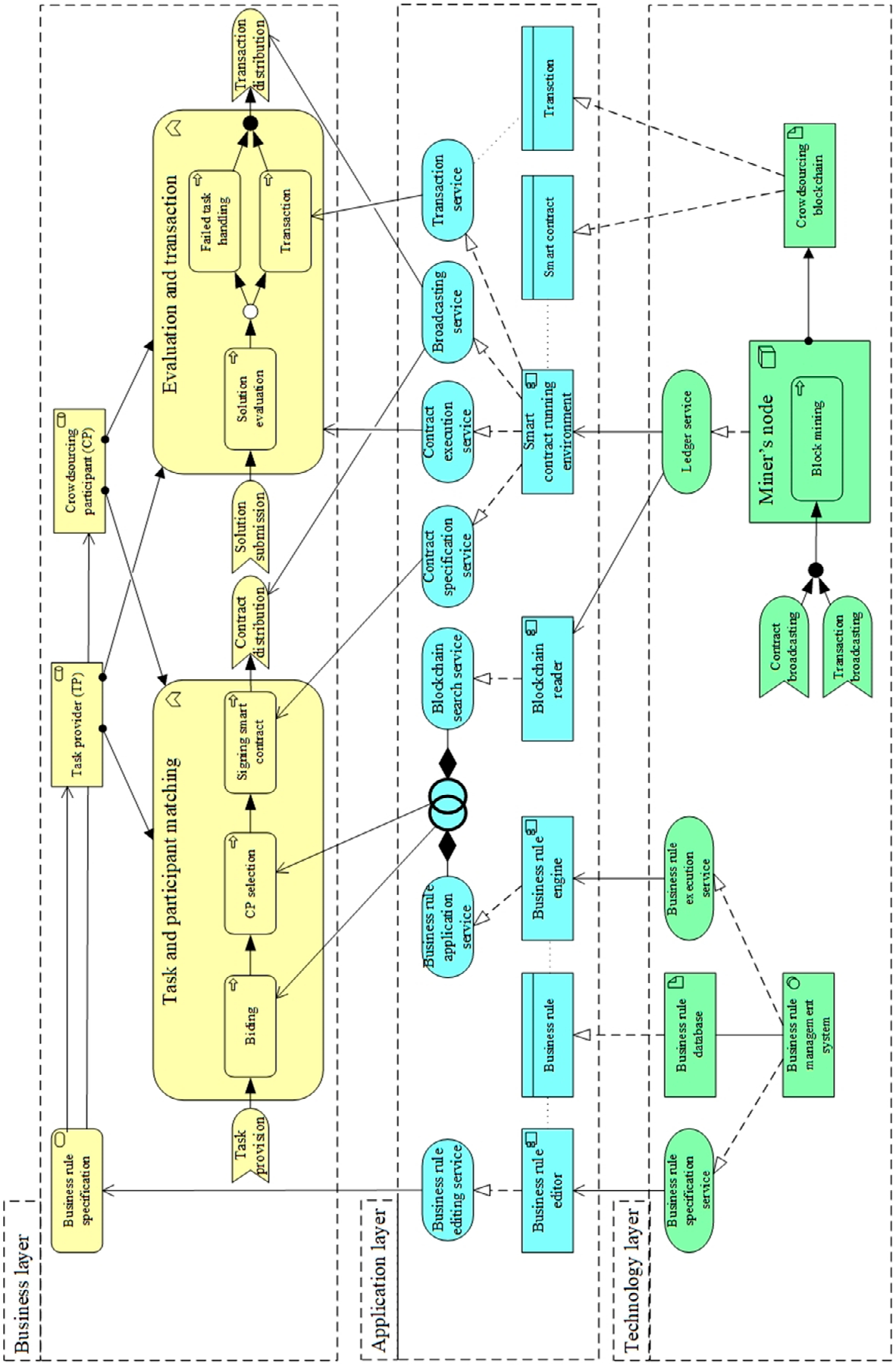

Figure 1. Reference architecture for blockchain-based crowdsourcing solutions.

Parties need to have a persistent and unique identity. Both the TPs and the CPs are identified in the blockchain according to their wallet IDs. To initiate a crowdsourcing task, a TP adds a description of the task requirements to the blockchain, including a description of how to contact them about the task. This description is then distributed throughout the network. All CPs in the network will receive the task information. To make things easier, interested CPs might use a BR application that searches the blockchain for new relevant tasks and notifies them of this. They could also view the record of the TP and check the payment record to determine whether the TP is reliable.

The CPs can propose a bid, which includes elements like a quotation, duration, and the target solution. TPs can view the backgrounds of the CPs who create a proposal by searching the blockchain for the evaluation of tasks performed by them in the past. Based on their proposals and background, the TP will evaluate and select the most suitable CP to perform the task.

There are two types of smart contracts included in the blockchain, either task-specific or agreement-specific. The first type is related to the task at hand, while the second is specific to the agreement between a TP and a CP. If the TP and the CP agree on the task requirements and conditions, the TP deposits the payment for the task in a task-specific smart contract having a reference to the task description. In addition, both parties will sign another smart contract stating the conditions (such as duration) that should be met by the CP to fulfill the task. This contract will refer to the agreement-specific smart contract.

To complete the task, CP will have to offer the solution to the TP within the agreed duration and provide a proof of existence of their solution and add this to the blockchain. A *proof of existence* of data, is a hash of the data that is added to the blockchain. Such a hash is unique to the data that it was generated from. Therefore, the existence of the hash, when stored on a blockchain, proves the existence of the data at the time it was added to the blockchain. If you have the hash and the data, it is easy to check whether the hash was generated from the data. However, it is very difficult, if not impossible, to obtain the data from the hash alone if the data is complex enough. In our case, a proof of existence of a solution can, thus, demonstrate the ownership of a solution without revealing the information it contains, and it provides proof that a solution was developed at a particular time [58]. The proof of existence on the blockchain was generated based on the data about the solution itself and its timestamp and is, therefore, trustable for TPs.

If the solution satisfies the TP, they will provide a positive evaluation of the CP to the blockchain, having a reference to the smart contract and to the proof of existence of the solution. This data will trigger the settlement of the smart contract, and the cryptocurrency hold for the task will be paid to the CP.

If the CP is too late in providing a solution, or if their solution is considered insufficient by the TP, then the TP provides a negative evaluation of the CP to the blockchain. This evaluation should refer to specific conditions that have not been met in the agreement between the TP and CP. The task-specific smart contract will not be triggered, and there will be no payment or a partial payment (dependent on the contract). The TP can now look for another CP to fulfill the task and make a new agreement. If the new CP does fulfill the task, then the old task-specific smart contract can still be executed, but payment is directed to the new CP.

4.3.2. Application Layer

The business rule editor should provide the TP with a way to formulate BRs. This should enable the TP to define their own logic for CP selection. According to Gong [10], the following parameters might come into consideration in typical knowledge-intensive crowdsourcing:

- A matching rate between the required skills for a task and the skills held by a CP;
- The average percentage of the customer satisfaction rate received by a CP;
- The number of tasks that have been completed by a CP;
- The average payment a crowdsource worker receives from their completed tasks;

- The discrepancy between the proposed price of a CP and the expected price of the TP in the form of a percentage;
- The discrepancy between the proposed duration of a CP and the expected duration of the TP in the form of a percentage.

The blockchain reader enables not only adding new data to the blockchain but also reading data from their copy of the ledger. When CPs have proposed solutions to the TP, the TP should be able to use the application to find and calculate the value of the above parameters for evaluation. A possible solution to implement the CP selection BRs is to employ an AHP-TOPSIS algorithm to rank the candidates based on their evaluation results [10]. AHP is an abbreviation of Analytical Hierarchical Process, while TOPSIS stands for Technique for Order Preference by Similarity to Ideal Solution. The top-ranked candidate would be the best of those who have submitted their proposals.

The running environment for task-specific smart contracts should contain the following elements: wallet ID of the TP, task ID, cryptocurrency, and code. The wallet ID of the TP is required to ensure that the smart contract will only trigger the payment after the positive evaluation of a knowledge solution by the TP. The task ID refers to the task description, as the contract should only be paid for the task. The smart contract should contain the cryptocurrency that is paid for the task. In addition, the smart contract should contain code that specifies that the payment will be made to a CP if the following conditions are met:

- The CP has provided a proof of existence of a solution to the task;
- There is an agreement between the CP and TP about the task in the blockchain;
- The TP has added a positive evaluation of the solution of the CP to the blockchain.

The code of the contract can be standardized. In this way, only element that the TP needs to supply is the task ID and the amount they will pay for fulfilling the task.

4.3.3. Technology Layer

Our reference architecture is based on existing blockchain technologies. Therefore, we do not specify a new blockchain design, but merely show how existing techniques are applied in our design. The distributed ledger is used to store tasks, evaluations, proof of existence, the task-specific smart contract, and the agreement between the CP and TP. Only the task-specific smart contract contains code, whereas the other elements consist of data that can be stored on the blockchain. There are several ways to store such data. For example, in Ethereum, data can be stored as part of a smart contract.

The application used by the CP and TP must be able to automatically find tasks and extract the appropriate data from them. This could be made easy by providing a standardized format for them. We specified the elements that the tasks, evaluation, proof of existence, and the agreement should contain. Depending on the specific blockchain implementation that is used with our design, the format for each of these elements is specified.

Usually, the capacity for storing the data is limited, and a party that wants to store data has to pay to do so. The reason for this is that every party running a full node has to store a copy. Apart from flexibility, scalability is a typical challenge in blockchain design [59]. To reduce scalability problems, parties need an incentive to store as little data as possible by following the data minimization principle. In our design, we deal with this by using abbreviations to store certain information. For example, the task category could be indicated by an integer. The application used by the TPs and the CPs can provide a translation from and to the integer and descriptions of task categories. This will be most difficult for the task description. However, even for these descriptions, there will be some standard elements that need to be discussed. Furthermore, this does not resolve the scalability issues related to the speed of adding new blocks of data.

Only the task-specific contract contains code and, in that sense, is a smart contract. There are many different programming languages and run environments for smart contracts. Most of them will be able to deal with the task-specific smart contract. First of all, we need a smart contract that has its own account in which cryptocurrencies can be stored, for example, Ethereum offers this functionality. Furthermore, there should be a

check whether there is agreement on the task between a CP and a TP, whether there is a positive evaluation, and whether the CP has provided a proof of existence of a solution. If this check is positive, then the smart contract is executed and the contract will be paid. As currently cryptocurrencies are not yet in mainstream use as a payment method, currency exchange services should be considered enable the use of fiat currencies.

4.4. Step 4: Demonstration

This section provides an exemplary use case to illustrate how the proposed reference architecture would accommodate a knowledge-intensive crowdsourcing scenario. According to Wieringa [60], a single case experiment allows for the validation of a single object study in design science research. In this demonstration, we focus on the utility of the proposed architecture in supporting knowledge-intensive crowdsourcing. The scenario originates from a crowdsourcing task for mobile application design and development. This scenario represents a typical knowledge-intensive crowdsourcing task, as similar tasks very often appear in the current platform-based crowdsourcing system. Using this scenario, we demonstrate how the application and technology components in the proposed reference architecture would support the business actors in a blockchain-based crowdsourcing system and how the system's activities are coordinated.

For the demonstration, we implemented the contracts mentioned in the application layer, namely contracts for storing a task description, providing agreement from the TP and CP, storing a proof of existence of a solution, storing an evaluation of the solution by the TP and a contract in which the task provider can store currency that automatically pays the CP after the appropriate conditions are met. For this implementation Solidity, a programming language for writing smart contracts, was used since it is commonly used by various blockchain platforms, including Ethereum [61]. The focus of this implementation was to show the feasibility of the design and that the contracts proposed in step three can be implemented and used in practice. The emphasis was on the code being close to the contract descriptions and less on solving broader issues of security, scalability, and efficiency. The contracts were tested in the Remix integrated development environment (remix.ethereum.org, a web browser-based IDE). The test was performed by executing each of the steps described in the scenario.

4.4.1. Scenario: Tourism Company

A tourism company in the role of a TP wants to crowdsource a mobile application development, as its IT personnel do not have the skills for mobile application development, and their budget is limited. The tourism company has a wallet on Ethereum that they can use to put smart contracts on the blockchain.

To start the crowdsourcing process, the tourism company starts by advertising the development task with the required information, such as task description, expected price, and contact information. To do so, they write a smart contract in which this information is stored. They also specify functions that can be used by other contracts to obtain the information. Part of the code for this contract in the Remix IDE, an integrated development environment for Ethereum like blockchains, is shown in Figure A1 in Appendix A. The output for some of the functions that can be used to obtain information from it is shown in the lower right corner.

All the nodes in the network receive the task advertisement when the contract is deployed. It is possible for contracts to emit events that other applications can listen to and obtain information from [62]. Through a user-configured filter, the CPs having the needed expertise and sufficient availability are able to see the advertisement and can also check on the company's task history and crowdsourcing records. Subsequently, the tourism company receives more than 100 proposals from different CPs.

In the selection of the most suitable CP, the tourism company has to consider a list of factors, including their quotation and delivery date, their reputation based on their previous crowdsourcing tasks, their skill level, etc. The historical record of a certain candidate can

be retrieved from the network, but it is too time-consuming for a tourism company to manually collect and analyze so much information from 100 proposals and records of CPs. Therefore, the analysis is carried out automatically by an application using customized business logic. The deliverable is the analysis and ranking of all the 100 CPs based on their proposal and records. The tourism company contacts the top-ranked candidate and asks the candidate to sign an agreement. Once both agree, the contracts are uploaded to the network with the unique identification of each party, and containing a reference to the task and the conditions that both parties agree upon.

In addition, the tourism company creates a task-specific smart contract and deploys it. In this smart contract, the conditions for payment are specified in the contract. Next, the tourism company deposits the cryptocurrencies that need to be paid for the performance of the task in the task-specific smart contract.

Figure A2 in Appendix A shows part of the code of the task-specific smart contract. In the lower part of the figure, the transaction is shown that the tourism company deposits currency in the task-specific smart contract. On the lower right, the balance of the contract shows that it now holds the currency deposited by the tourism company. A function is incorporated that can determine the payment to the candidate, but it will not make the payment at this stage, as a proof of existence and a positive evaluation have to be provided first.

Finally, the development of the mobile application is completed before the given deadline. The CP creates another smart contract in which they store a hash of their solution as a proof of existence. At the same time, they submit the solution to the tourism company. After an evaluation of the solution, the company is satisfied with the deliverable. They create a new smart contract with the data elements for the evaluation of the solution, that includes a reference to the contract containing the proof of existence of the solution. This contract includes a positive score to show that the company is satisfied and a list of conditions that have been met by the candidate for instance.

To transfer the currency stored in the task-specific smart contract as payment, they can provide the addresses of the contracts with the agreement they made, the proof of existence of a solution, and the evaluation. In the task-specific smart contract, these addresses are used to obtain the information from the other contracts to determine whether all conditions have been met. As these conditions have been met in this scenario, the currency deposited by the tourism company in the contract is transferred to the wallet of the providing party.

4.4.2. Activities Coordination

Figure 2 presents a Unified Modeling Language (UML) activity diagram to demonstrate how the business rule application, smart contract running environment, and the blockchain support the interactions between the tourism company and the candidate in detail. This diagram also presents the typical system activities, including human-performed and software-performed tasks. Both the tourism company and the candidate have their own business rule and blockchain reader application and smart contract running environments. The business rule and blockchain reader are listed in one lane in the diagram, as they often work in collaboration. To avoid being overwhelmed by technical details, we combined all blockchain operations into one lane. The blockchain lane contains all the read and write operations in the blockchain, including searching in the tourism company or the candidate's local copy of the blockchain or uploading data to the blockchain network for other miners to write into a new block. This demonstration assists the verification by mapping the links between the components in the reference architecture, and the necessary functionalities reflected by the system's activities according to the scenario.

The analysis of activities coordination proves the promise that the reference architecture enables blockchain-based crowdsourcing without the need of intermediaries. As this coordination mechanism is based on business rules and smart contract technology, it is not constrained by cognitive limits [63], and consequently, it reduces search friction. Further evaluation is discussed in the next section.

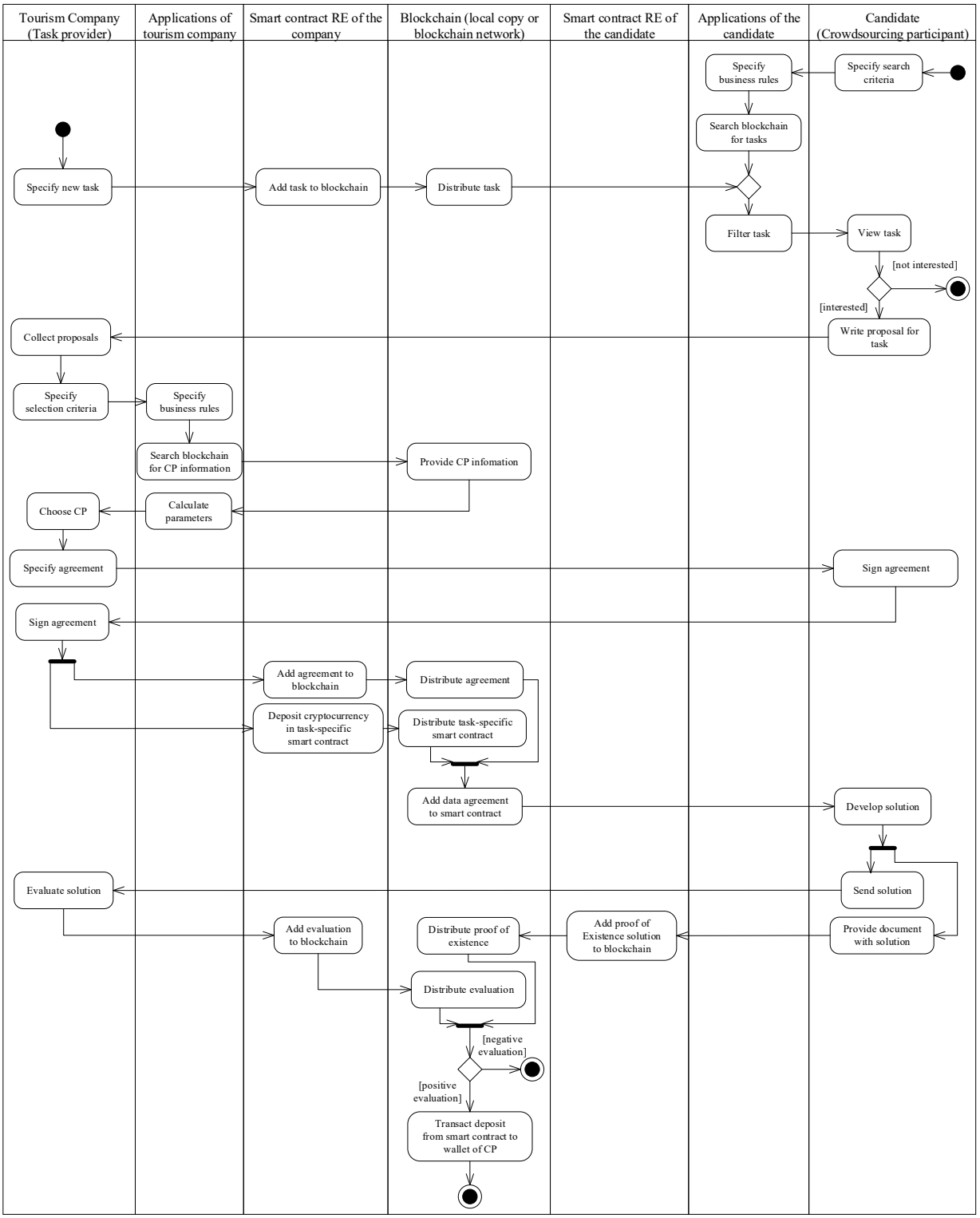

Figure 2. UML activity diagram for activity coordination in the scenario. RE = running environment, TP = task provider, CP = crowdsourcing participant.

4.5. Step 5: Evaluation

In the previous step, we demonstrated our reference architecture. In addition, we provided proof of concept to show the feasibility of the use of contracts in the architecture by implementing them in Solidity. This section evaluates the reference architecture for the blockchain-based crowdsourcing solution by checking the requirements with the design. In

step 2, the requirements regarding blockchain-based knowledge-intensive crowdsourcing systems were presented. Blockchain-based systems fulfilling these requirements should resolve current challenges in knowledge-intensive crowdsourcing and enable more effective coordination between TPs and CPs.

4.5.1. Requirements Verification

The first requirement refers to minimizing search friction between TPs and CPs. In the blockchain-based solution, all the CPs within the network are able to receive notification of new tasks. Filter applications could be employed by CPs to avoid information overload and disturbance. In theory, this could minimize the search friction between the supply and demand sides [31]. Our scenario-based explication demonstrates that the blockchain-based solution could coordinate the interactions between TPs and CPs without intermediaries, which are performed by the current crowdsourcing platforms. However, this also depends on the number of users who are active in the network. From a technical point of view, a blockchain network could allow for a huge number of users to be connected to each other. The technical feasibility could refer to the Bitcoin network and the size of its user community. Furthermore, an advantage of a blockchain network for crowdsourcing over current market leaders is the lower cost for coordinating TPs and CPs.

The second requirement relates to the prevention of cheating. As the evaluations by TPs of tasks performed by CPs are stored in the blockchain, both parties can access each other's past behavior. This will provide them with a way to determine whether the other party is sufficiently skilled or reliable.

The storing of the agreement between the CP and TP on the conditions for fulfilling the task also provides them both with security, as neither party can claim afterward that the conditions agreed upon were different. Furthermore, as the evaluation is linked to the task description, TPs can better determine the reputation of a CP in a specific domain. The linking of the evaluation to the conditions provides additional information on the CP. Furthermore, referring to specific conditions forces the TP to provide some argument for why they consider the CP to have failed a task. This also makes it easier for a CP to contest an evaluation. In addition, it makes it impossible to randomly provide negative evaluations of CPs that are not involved in the task.

On many of the current crowdsourcing platforms, TPs would have a great incentive to provide the CP with a negative evaluation, as this would mean that they do not need to pay the CP (false reporting). However, in our design, the TP must deposit their payment into a smart contract prior to making an agreement with the CP. This smart contract will only make the payment to a wallet if that wallet has provided a proof of existence of a solution that has a positive evaluation by the TP. This means that the only way for the TP to reclaim their money back is by acting as a fake CP who provides a proof of existence of a fake solution. Of course, this would not be very difficult for a TP to achieve. However, if a CP has a suspicion that the TP is dishonest, they have proof that they had a solution at the time they added the proof of existence to the blockchain. If needed in a court of law, similarities between the solution provided by the CP and the solution that the TP paid for could be investigated. It could also be determined whether the TP paid out to a fake solution. In this way, dishonesty by the TP can be determined. It should be noted here that the proof of existence provides proof that the CP had the solution, not that it was sent to the TP. This would require additional proof.

Adding a task-specific contract before, or simultaneously with, the agreement between TP and CP balances the risks of both parties. By making an agreement with a TP, a CP risks a negative evaluation that will be credible insofar as there is proof that they worked for the TP. This assures the TP that the CP will do their best to perform the task. However, as the smart contract stores the cryptocurrency of the TP, and they cannot easily get it back, the incentive for the TP to provide an unfair evaluation is reduced. Based on the transparency of data and smart contracts, our design ensures fairness in doing business and, in this way, is a powerful incentive mechanism resulting in trust.

A danger is parties creating various wallets and using them to post, fulfill, and pay for tasks to build fake reputations, i.e., like in a *Sybil attack* [64]. However, this would require creating many wallets that have complicated interactions with each other to ensure that this is not easy to detect. For example, if a TP always deals with the same CP or vice versa, then this would be easy to detect, given the openness of data. Furthermore, they would have to make reasonable payments to avoid suspicion. Conducting transactions and executing a smart contract requires a fee. Any party performing a Sybil attack cannot do so without paying various fees, and this should discourage such an attack.

The third requirement concerns the openness to apply customized applications for processing the data retrieved from the blockchain. In our design, business rule applications are employed to fulfill this requirement. Both the TPs and CPs can develop and employ their own business rule applications. Benefiting from the high level of data openness, more flexible use of the data is also possible and consequently generates desired business value. As the smart contract and blockchain data run in a sandbox environment, the business rule application can only access data via an interface, but will not be able to mutate or to be tampered with. This makes the business rule applications completely separate from the smart contract running environment and creates a high level of flexibility.

The advantage of using blockchain technology is that it provides a way for TPs to easily distribute tasks to a high number and variety of CPs. This makes it possible to reduce search friction and for TPs and CPs to search for and find tasks according to their own logic, using business rules. Furthermore, blockchain technology allows for specifying and automatically enforcing smart contracts and for storing proof of existence. This will reduce cheating and prevent dishonesty by buyers and suppliers.

4.5.2. Limitations

In this research, a blockchain-based reference architecture for knowledge-intensive crowdsourcing is designed. As the blockchain technology was originally developed for cryptocurrency, the current blockchain software techniques have their limitations in non-financial domains. To allow for the extensive use of blockchain in a wide range of domains, the following issues have to be addressed: (1) Scalability: how to store large amounts of data and manage its distribution via the blockchain network; (2) flexibility: how to balance data integrity and efficiency of data management; (3) maintenance: how to design appropriate mining incentive awards to encourage miners to keep mining in the network; and (4) privacy: how to balance openness and the protection of private information. These issues concern challenges to blockchain technology that are pervasive across its application and have not yet been resolved [65].

The most significant limitation of blockchain that we identified concerns scalability issues. General solutions for improving the scalability of blockchain technology is ongoing e.g., [59,66]. It might be expected that such solutions can be applied in the crowdsourcing domain as well. In this sense, future research should first aim to address the scalability issue. Unlike scalability which has its root at technology development, the rest three issues are social-technical and refer to the governance mechanisms. The second direction for future research concerns with platform governance.

Another limitation of this study concerns the limited evaluation of the reference architecture. As the demonstration can only show the technical feasibility of the core processes and components of the architecture, it does not further prove its impacts to human behavior in a large crowdsourcing network. Malicious behavior in crowdsourcing is complex and driven or inhibited by multiple factors [47]. It is subject to more comprehensive studies on the behavior of TPs and CPs in a new blockchain network environment in the future. We consider the third direction for future research is to understand the factors which influence user behaviors in a blockchain-based platform.

5. Implications

5.1. Theoretical Implications

The potential of blockchain-based solutions for crowdsourcing platforms has been indicated by literature, as well as in this study. In comparing existing solutions that aim to address specific problems, such as privacy prevention, our reference architecture focuses on aligning the required disintermediated business model and blockchain-related technologies. From a theoretical perspective, this alignment reveals the underlying reason for creating value from blockchain for knowledge-intensive crowdsourcing. The demonstration of the core business processes in knowledge-intensive crowdsourcing, and the coordination between TPs and CPs, further explains that the blockchain solution can improve crowdsourcing by reducing knowledge transfer costs and search friction. This finding contributes to the understanding and theorization of value creation mechanisms with blockchain infrastructure.

5.2. Practical Implications

Blockchain technology is often considered as an enabler of innovations such as decentralized digital platforms and decentralized autonomous organizations [67]. Due to its benefit in lowering knowledge transfer costs, the application of blockchain also has the potential to significantly impact KIBS organizations. A foreseeable impact is a change in organizational boundaries. If KIBS companies could connect to talents outside their organizations with limited coordination costs and as easily as they connect to their own employees, the existing organizational boundary, and the way how people collaborate with each other will change. In the long run, this will result in transforming the organization structure and institution as well [68]. Policy-makers and strategists should look beyond the adoption of technology and seek new policies to reduce the barriers caused by the competition between the new and existing organizational structure and institutions.

The industries and markets need to develop new business models to allow the realization of blockchain solutions [14,18], but many fail [23]. 'Blockchain is an innovative technology in search of use cases' [19] (p. 1543). Furthermore, application-oriented contributions to blockchain research appear to be scarce [18]. This paper contributes to the understanding of blockchain and shows that context-aware understanding and a theory-driven design can help to arrive at feasible solutions. By presenting the high-level design, this architecture shows the feasibility of blockchain technologies to contribute to the crowdsourcing domain and the potential benefits. Furthermore, this study also provides insights into how to transform the current business model of crowdsourcing to allow the implementation of blockchain technologies. From the perspectives of the KBV of the firm and the theory of search friction, this technology-enabled transformation is expectable, as the market will keep on seeking for more efficient coordination mechanisms.

In the transformation to a blockchain-based crowdsourcing paradigm, current crowdsourcing platforms should shift their business services from information-sharing and data management to the provision of services and technical support to users. It is often difficult for companies to develop BR applications consisting of complex algorithms, while blockchain-based smart contracts might also be new to them. Crowdsourcing platforms could focus on providing BR packages and contract templates for users, or becoming technical and legal consultants. This reference architecture identifies the benefits of blockchain-related technologies for crowdsourcing industry. It also offers crowdsourcing service providers insight on how to realize these benefits by adapting their business models.

6. Conclusions

Blockchain can be used for e-commerce transactions using smart contracts and, in this way, replace traditional intermediaries in crowdsourcing. Such a blockchain-based platform can overcome the challenges of fragmentation of expertise over multiple platforms, difficulty in preventing cheating, and the lack of access to historical data to learn from this. The emergence of blockchain technology offers the opportunity to transform the

current crowdsourcing platforms into a blockchain-based e-commerce market, which is characterized by its distributed nature and consensus mechanism. The KBV of the firm and the theory of search friction predict that disintermediation allows for lower knowledge transfer costs in knowledge-intensive crowdsourcing. Blockchain technology offers the infrastructure for disintermediation in such a way that buyers and sellers can transact directly.

We followed a design science approach and developed a reference architecture for implementing blockchain-based crowdsourcing solutions. This architecture uses blockchain to store crowdsourcing records and distribute knowledge-intensive tasks. Smart contracts are used to coordinate and regulate the behavior of both TPs and CPs. Furthermore, business rule applications are in place to provide flexible data processing. This architecture can be used to assess, design, and implement real-world crowdsourcing systems that allow for disintermediated coordination and higher efficiency in managing knowledge-intensive crowdsourcing tasks. In comparison with conventional crowdsourcing platforms, blockchain-based crowdsourcing can have less search friction between TPs and CPs, enable a more powerful mechanism to avoid cheating, and have a greater openness to flexibly implement business logic. Traditional intermediaries providing a platform are not needed anymore, as the distributed blockchain application mediates the transactions, and smart contracts are automatically settled.

The blockchain solution for knowledge-intensive crowdsourcing is still nascent. The reference architecture is an abstract model of a system that has the potential to resolve similar problems in practice. We have only implemented, demonstrated and evaluated the basic processes of knowledge-intensive crowdsourcing to allow an evaluation of the architecture. In addition to the challenges mentioned above, further research into the business impact of using the blockchain-based platform is recommended.

Author Contributions: Conceptualization, Y.G. and M.J.; methodology, Y.G. and S.v.E.; validation, S.v.E. and Y.G.; writing—original draft preparation, Y.G. and S.v.E.; writing—review and editing, M.J.; funding acquisition, Y.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by "the Fundamental Research Funds for the Central Universities", grant number 2020SK018.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

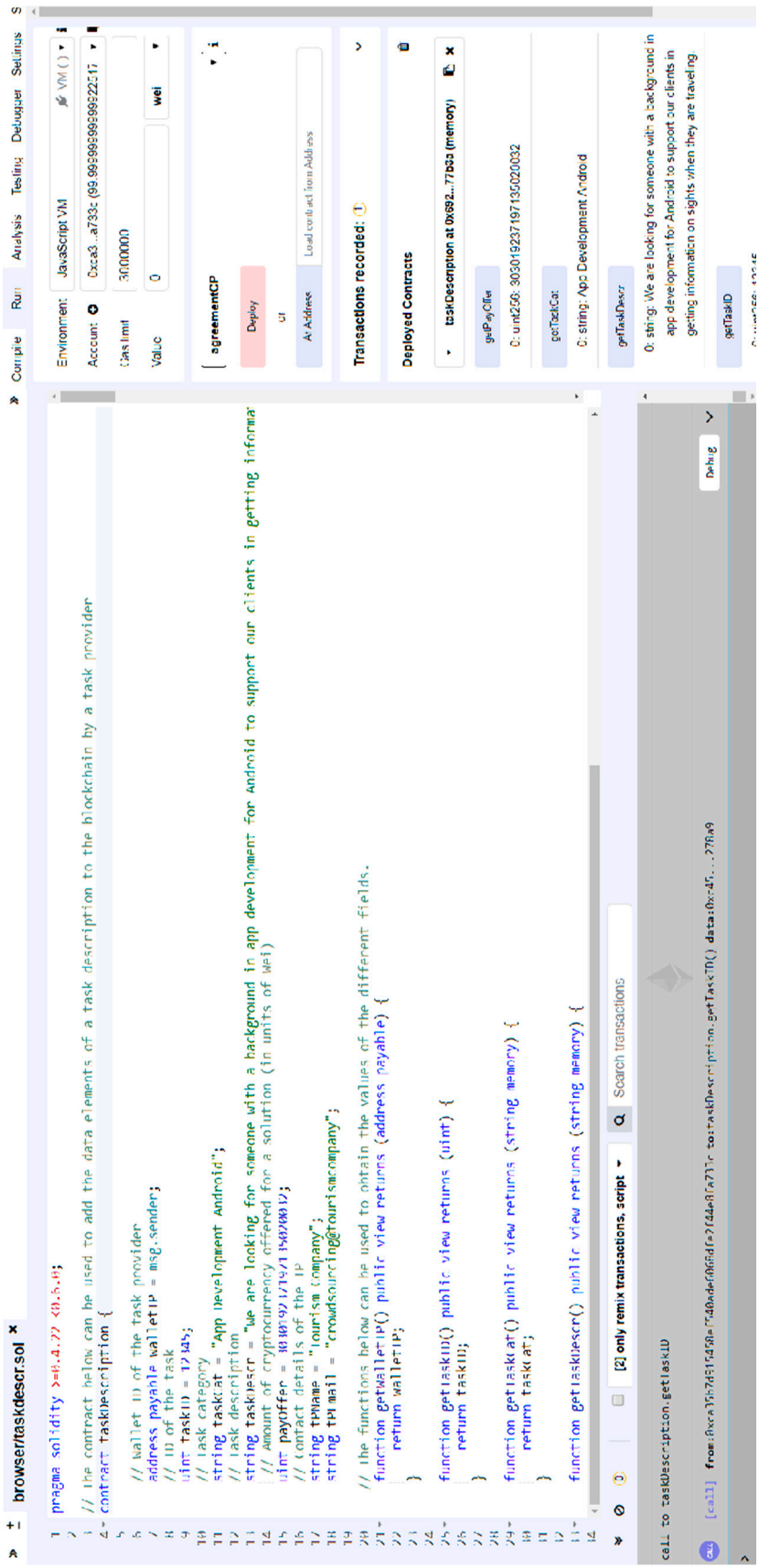

Figure A1. The contract for task descriptions and its output in the Remix IDE.

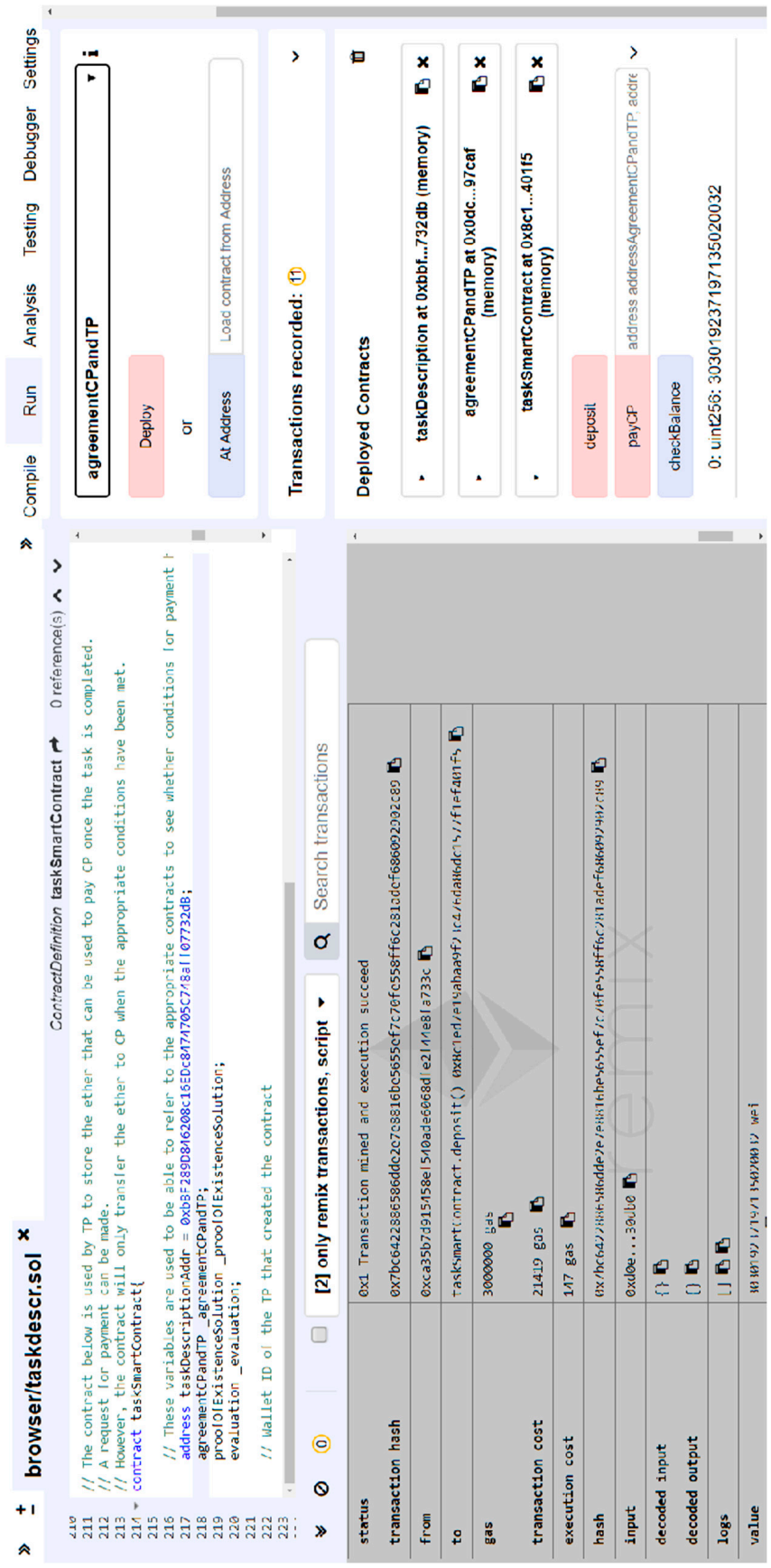

Figure A2. The transaction in which the tourism company deposits currency in the task-specific smart contract in Remix IDE.

References

1. Palacios, M.; Martinez-Corral, A.; Nisar, A.; Grijalvo, M. Crowdsourcing and organizational forms: Emerging trends and research implications. *J. Bus. Res.* **2016**, *69*, 1834–1839. [CrossRef]
2. Wang, M.-M.; Wang, J.-J. Understanding Solvers' Continuance Intention in Crowdsourcing Contest Platform: An Extension of Expectation-Confirmation Model. *J. Theor. Appl. Electron. Commer. Res.* **2019**, *14*, 17–33. [CrossRef]
3. Afuah, A.; Tucci, C.L. Crowdsourcing As a Solution to Distant Search. *Acad. Manag. Rev.* **2012**, *37*, 355–375. [CrossRef]
4. Satzger, B.; Psaier, H.; Schall, D.; Dustdar, S. Auction-based crowdsourcing supporting skill management. *Inf. Syst.* **2013**, *38*, 547–560. [CrossRef]
5. Kittur, A.; Nickerson, J.V.; Bernstein, M.S.; Gerber, E.M.; Shaw, A.; Zimmerman, J.; Lease, M.; Horton, J.J. The Future of Crowd Work. In Proceedings of the 16th ACM Conference on Computer Supported Cooperative Work and Social Computing (CSCW 2013), San Antonio, TX, USA, 23–27 February 2013.
6. Roy, S.B.; Lykourentzou, I.; Thirumuruganathan, S.; Amer-Yahia, S.; Das, G. Task assignment optimization in knowledge-intensive crowdsourcing. *VLDB J.* **2015**, *24*, 467–491.
7. Soliman, W.; Tuunainen, V.K. Understanding Continued Use of Crowdsourcing Systems: An Interpretive Study. *J. Theor. Appl. Electron. Commer. Res.* **2015**, *10*, 1–18. [CrossRef]
8. Li, M.; Weng, J.; Yang, A.; Lu, W.; Zhang, Y.; Hou, L.; Liu, J.-N.; Xiang, Y.; Deng, R.H. CrowdBC: A Blockchain-Based Decentralized Framework for Crowdsourcing. *IEEE Trans. Parallel Distrib. Syst.* **2019**, *30*, 1251–1266. [CrossRef]
9. Mahr, D.; Rindfleisch, A.; Slotegraaf, R.J. Enhancing Crowdsourcing Success: The Role of Creative and Deliberate Problem-Solving Styles. *Cust. Needs Solut.* **2015**, *2*, 209–221. [CrossRef]
10. Gong, Y. Estimating participants for knowledge-intensive tasks in a network of crowdsourcing marketplaces. *Inf. Syst. Front.* **2016**, *19*, 301–319. [CrossRef]
11. Lu, Y.; Tang, Q.; Wang, G. *ZebraLancer: Private and Anonymous Crowdsourcing System Atop Open Blockchain*; New Jersey Institute of Technology: Newark, NJ, USA, 2018.
12. Yli-Huumo, J.; Ko, D.; Choi, S.; Park, S.; Smolander, K. Where Is Current Research on Blockchain Technology?—A Systematic Review. *PLoS ONE* **2016**, *11*, e0163477. [CrossRef]
13. Li, Z.; Liu, X.; Wang, W.M.; Barenji, A.V.; Huang, G.Q. CKshare: Secured cloud-based knowledge-sharing blockchain for injection mold redesign. *Enterp. Inf. Syst.* **2018**, *13*, 1–33. [CrossRef]
14. Aste, T.; Tasca, P.; Matteo, T.D. Blockchain Technologies: The Foreseeable Impact on Society and Industry. *Computer* **2017**, *50*, 18–28. [CrossRef]
15. Lu, Q.; Xu, X. Adaptable Blockchain-Based Systems: A Case Study for Product Traceability. *IEEE Softw.* **2017**, *34*, 21–27. [CrossRef]
16. Tapscott, D.; Tapscott, A. How blockchain will change organizations. *MIT Sloan Manag. Rev.* **2017**, *58*, 10.
17. Iansiti, M.; Lakhani, K.R. The Truth About Blockchain. *Harv. Bus. Rev.* **2017**, *95*, 118–127.
18. Risius, M.; Spohrer, K. A Blockchain Research Framework. *Bus. Inf. Syst. Eng.* **2017**, *59*, 385–409. [CrossRef]
19. Glaser, F. Pervasive Decentralisation of Digital Infrastructures: A Framework for Blockchain enabled System and Use Case Analysis. In Proceedings of the 50th Hawaii International Conference on System Sciences, Hilton Waikoloa Village, HI, USA, 4–7 January 2017.
20. Buccafurri, F.; Lax, G.; Nicolazzo, S.; Nocera, A. Tweetchain: An Alternative to Blockchain for Crowd-Based Applications. In Proceedings of the International Conference on Web Engineering, Rome, Italy, 5–8 June 2017.
21. Yang, M.; Zhu, T.; Liang, K.; Zhou, W.; Deng, R.H. A blockchain-based location privacy-preserving crowdsensing system. *Futur. Gener. Comput. Syst.* **2019**, *94*, 408–418. [CrossRef]
22. Zou, J.; Ye, B.; Qu, L.; Wang, Y.; Orgun, M.A.; Li, L. A Proof-of-Trust Consensus Protocol for Enhancing Accountability in Crowdsourcing Services. *IEEE Trans. Serv. Comput.* **2019**, *12*, 429–445. [CrossRef]
23. Higginson, M.; Nadeau, M.-C.; Rajgopal, K. Blockchain's Occam Problem. Available online: https://immagic.com/eLibrary/ARCHIVES/GENERAL/MCKNSYUS/M190101H.pdf (accessed on 11 January 2021).
24. Taylor, R.N.; Medvidović, N.; Dashofy, E.M. *Software Architecture: Foundations, Theory, and Practice*; John Wiley & Sons, Inc.: Hoboken, NJ, USA, 2009.
25. Simón, F.J.G.; Alcamí, R.L.; Ribera, T.B. Social networks and Web 3.0: Their impact on the management and marketing of organizations. *Manag. Decis.* **2012**, *50*, 1880–1890. [CrossRef]
26. Alavi, M.; Leidner, D.E. Review: Knowledge Management and Knowledge Management Systems: Conceptual Foundations and Research Issues. *MIS Q.* **2001**, *25*, 107–136. [CrossRef]
27. Kogut, B.; Zander, U. Knowledge of the Firm, Combinative Capabilities, and the Replication of Technology. *Organ. Sci.* **1992**, *3*, 383–397. [CrossRef]
28. Grant, R.M. Toward a knowledge-based theory of the firm. *Strat. Manag. J.* **1996**, *17*, 109–122. [CrossRef]
29. Bae, J.; Koo, J. Information loss, knowledge transfer cost and the value of social relations. *Strat. Organ.* **2008**, *6*, 227–258. [CrossRef]
30. Pissarides, C.A. Equilibrium in the Labor Market with Search Frictions. *Am. Econ. Rev.* **2011**, *101*, 1092–1105. [CrossRef]
31. Mortensen, D.T. Markets with Search Friction and the DMP Model. *Am. Econ. Rev.* **2011**, *101*, 1073–1091. [CrossRef]
32. Williamson, O.E. The Theory of the Firm as Governance Structure: From Choice to Contract. *J. Econ. Perspect.* **2002**, *16*, 171–195. [CrossRef]

33. Jeppesen, L.B.; Lakhani, K.R. Marginality and Problem-Solving Effectiveness in Broadcast Search. *Organ. Sci.* **2010**, *21*, 1016–1033. [CrossRef]
34. Zenger, T.; Felin, T.; Bigelow, L.S. Theories of the Firm-Market Boundary. *Acad. Manag. Ann.* **2011**, *5*, 89–133. [CrossRef]
35. Lin, C.; He, D.; Zeadally, S.; Kumar, N.; Choo, K.-K.R. SecBCS: A secure and privacy-preserving blockchain-based crowdsourcing system. *Sci. China Inf. Sci.* **2020**, *63*, 130102. [CrossRef]
36. Hu, S.; Hou, L.; Chen, G.; Weng, J.; Li, J. Reputation-based Distributed Knowledge Sharing System in Blockchain. In Proceedings of the 15th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services, New York, NY, USA, 5–7 November 2018.
37. Yu, Y.; Liu, S.; Guo, L.; Yeoh, P.L.; Vucetic, B.; Li, Y. CrowdR-FBC: A Distributed Fog-Blockchains for Mobile Crowdsourcing Reputation Management. *IEEE Internet Things J.* **2020**, *7*, 8722–8735. [CrossRef]
38. Rasool, S.; Iqbal, M.; Dagiuklas, T.; Ul-Qayyum, Z.; Li, S. Reliable Data Analysis through Blockchain based Crowdsourcing in Mobile Ad-hoc Cloud. *Mob. Netw. Appl.* **2019**, *25*, 153–163. [CrossRef]
39. Ma, Y.; Sun, Y.; Lei, Y.; Qin, N.; Lu, J. A survey of blockchain technology on security, privacy, and trust in crowdsourcing services. *World Wide Web* **2020**, *23*, 393–419. [CrossRef]
40. Hevner, A.; Chatterjee, S. *Design Research in Information Systems*; Springer: Berlin, Germany, 2010; Volume 22.
41. Walls, J.G.; Widmeyer, G.R.; El Sawy, O.A. Building an Information System Design Theory for Vigilant EIS. *Inf. Syst. Res.* **1992**, *3*, 36–59. [CrossRef]
42. Peffers, K.; Tuunanen, T.; Rothenberger, M.A.; Chatterjee, S. A Design Science Research Methodology for Information Systems Research. *J. Manag. Inf. Syst.* **2007**, *24*, 45–77. [CrossRef]
43. Straub, D.; Ang, S. Editor's Comments: Rigor and Relevance in IS Research: Redefining the Debate and a Call for Future Research. *MIS Q.* **2011**, *35*, 3–11. [CrossRef]
44. Shrestha, A.; Cater-Steel, A.; Toleman, M.; Rout, T. Benefits and relevance of International Standards in a design science research project for process assessments. *Comput. Stand. Interfaces* **2018**, *60*, 48–56. [CrossRef]
45. Guittard, C.; Schenk, E.; Burger-Helmchen, T. Crowdsourcing and the Evolution of a Business Ecosystem. In *Advances in Crowdsourcing*; Garrigos-Simon, F.J., Gil-Pechuán, I., Estelles-Miguel, S., Eds.; Springer: Cham, Switzerland, 2015; pp. 49–60.
46. Peer, E.; Brandimarte, L.; Samat, S.; Acquisti, A. Beyond the Turk: Alternative platforms for crowdsourcing behavioral research. *J. Exp. Soc. Psychol.* **2017**, *70*, 153–163. [CrossRef]
47. Gadiraju, U.; Kawase, R. Improving Reliability of Crowdsourced Results by Detecting Crowd Workers with Multiple Identities. In Proceedings of the 17th International Conference Web Engineering, Rome, Italy, 5–8 June 2017.
48. Zhang, X.; Xue, G.; Yu, R.; Yang, D.; Tang, J. Keep Your Promise: Mechanism Design Against Free-Riding and False-Reporting in Crowdsourcing. *IEEE Internet Things J.* **2015**, *2*, 562–572. [CrossRef]
49. Hofstetter, R.; Nair, H.; Misra, S. *A Copycat Penalty: Micro Evidence From An Online Crowdsourcing Platform*; Stanford University Graduate School of Business Research: Stanford, CA, USA, 2018.
50. Ye, H.; Kankanhalli, A. Solvers' participation in crowdsourcing platforms: Examining the impacts of trust, and benefit and cost factors. *J. Strat. Inf. Syst.* **2017**, *26*, 101–117. [CrossRef]
51. Slivkins, A.; Vaughan, J.W. Online Decision Making in Crowdsourcing Markets: Theoretical Challenges. *ACM SIGecom Exch.* **2013**, *12*, 4–23. [CrossRef]
52. McInnis, B.; Cosley, D.; Nam, C.; Leshed, G. Taking a HIT: Designing around Rejection, Mistrust, Risk, and Workers' Experiences in Amazon Mechanical Turk. In Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems, San Jose, CA, USA, 7–12 May 2016.
53. Delmolino, K.; Arnett, M.; Kosba, A.; Miller, A.; Shi, E. Step by Step Towards Creating a Safe Smart Contract: Lessons and Insights from a Cryptocurrency Lab. In Proceedings of the International Conference on Financial Cryptography and Data Security Christ Church, Barbados, Christ Church, Barbados, 3–7 March 2014.
54. Van der Aalst, W.M.P.; Weske, M.; Grünbauer, D. Case handling: A new paradigm for business process support. *Data Knowl. Eng.* **2005**, *53*, 129–162.
55. ArchiMate. *ArchiMate 3.0.1 Specification*; The Open Group: San Francisco, CA, USA, 2017.
56. Pilkington, M. Blockchain Technology: Principles and Applications. In *Research Handbook on Digital Transformations*; Olleros, F.X., Zhegu, M., Eds.; Edward Elgar Publishing: Northampton, MA, USA, 2016.
57. Walport, M. *Distributed Ledger Technology: Beyond Block Chain*; Government Office for Science: London, UK, 2016.
58. Swan, M. *Blockchain: Blueprint for a New Economy*; O'Reilly Media: Sevastopol, CA, USA, 2015.
59. Vukolić, M. The Quest for Scalable Blockchain Fabric: Proof-of-Work vs. BFT Replication. In Proceedings of the International Workshop on Open Problems in Network Security, Zurich, Switzerland, 7–9 June 2011.
60. Wieringa, R.J. *Design Science Methodology for Information Systems and Software Engineering*; Springer: Berlin, Germany, 2014.
61. Dannen, C. *Introducing Ethereum and Solidity: Foundations of Cryptocurrency and Blockchain Programming for Beginners*; Apress: New York, NY, USA, 2017.
62. Ethereum. Solidity Documentation Release 0.5.3. Available online: https://solidity.readthedocs.io/en/v0.5.3/ (accessed on 21 April 2019).
63. Christie, A.A.; Joye, M.P.; Watts, R.L. Decentralization of the firm: Theory and evidence. *J. Corp. Financ.* **2003**, *9*, 3–36. [CrossRef]

64. Douceur, J.R. The Sybil Attack. In Proceedings of the International Workshop on Peer-to-Peer Systems, Cambridge, MA, USA, 8 October 2004.
65. Underwood, S. Blockchain beyond bitcoin. *Commun. ACM* **2016**, *59*, 15–17. [CrossRef]
66. Croman, K.; Decker, C.; Eyal, I.; Gencer, A.E.; Juels, A.; Kosba, A.; Miller, A.; Saxena, P.; Shi, E.; Sirer, E.G.; et al. On Scaling Decentralized Blockchains. In Proceedings of the Financial Cryptography and Data Security, Christ Church, Barbados, 3–7 March 2014.
67. Beck, R.; Müller-Bloch, C.; King, J.L. Governance in the Blockchain Economy: A Framework and Research Agenda. *J. Assoc. Inf. Syst.* **2018**, *19*, 1020–1034. [CrossRef]
68. Allen, D.W.; Berg, C.; Markey-Towler, B.; Novak, M.; Potts, J. Blockchain and the evolution of institutional technologies: Implications for innovation policy. *Res. Policy* **2020**, *49*, 103865. [CrossRef]