Trust Management in a Blockchain Based Fog Computing Platform with Trustless Smart Oracles

Petar Kochovski[a,b,c], Sandi Gec[a,c], Vlado Stankovski[c,*], Marko Bajec[a], Pavel D. Drobintsev[b]

[a] *Faculty of Computer and Information Science, University of Ljubljana, Ljubljana, Slovenia*
[b] *Institute of Computer Science and Technology, Peter the Great St. Petersburg Polytechnic University, St. Petersburg, Russian Federation*
[c] *Faculty of Civil and Geodetic Engineering, University of Ljubljana, Ljubljana, Slovenia*

Abstract

Trust is a crucial aspect when cyber-physical systems have to rely on resources and services under ownership of various entities, such as in the case of Edge, Fog and Cloud computing. The DECENTER's Fog Computing Platform is developed to support Big Data pipelines, which start from the Internet of Things (IoT), such as cameras that provide video-streams for subsequent analysis. It is used to implement Artificial Intelligence (AI) algorithms across the Edge-Fog-Cloud computing continuum which provide benefits to applications, including high Quality of Service (QoS), improved privacy and security, lower operational costs and similar. In this article, we present a trust management architecture for DECENTER that relies on the use of blockchain-based Smart Contracts (SCs) and specifically designed trustless Smart Oracles. The architecture is implemented on Ethereum ledger (testnet) and three trust management scenarios are used for illustration. The scenarios (trust management for cameras, trusted data flow and QoS based computing node selection) are used to present the benefits of establishing trust relationships among entities, services and stakeholders of the platform.

Keywords: Trust, Fog, Blockchain, Smart Contract, Smart Oracle

1. Introduction

Today, there is an increasing trend to build smart applications in various domains, such as smart homes, smart cities and communities, industry 4.0, robotics and similar [1, 2, 3, 4, 5]. The development of smart applications in the construction sector motivates this study [6, 7]. Generally, the emergence of such applications is supported by three types of converging technologies: the Internet of Things (IoT), Artificial Intelligence (AI) and the Cloud. There are many expected benefits from the convergence of these triumvirate technologies, such as the emergence of more sophisticated and powerful AI applications, improved Quality of Service (QoS), higher utilisation of resources, and lower operational costs. In order to address various non-functional requirements of smart applications, Cloud computing today has evolved into two new flavours coined Edge referring to virtualisation at the edge of the network, and Fog generally referring to various geographically distributed, not so powerful Cloud computing providers.

With these recent developments, it is becoming evident that the mechanisms for provisioning, leasing and otherwise granting usage rights for IoT devices, computing and networking infrastructures, data and software on the Internet are prolific. Moreover, the IoT, AI and Cloud technologies alone are not enough to support dynamic application scenarios across broad geographic areas, for instance, when equipment, robots, cars or smartphones move from one place to another. In such scenarios, the processes of dynamic integration of hardware and software resources depending on application needs and the actual execution environments, are increasingly more complex. Due to the complexity, the applications may be exposed to various threats. This highlights the importance of achieving trust among all participating entities. The present study addresses the problem of achieving trust by designing and implementing a new

[*] Corresponding author
Email address: `vlado.stankovski@fgg.uni-lj.si` (Vlado Stankovski)

trust management architecture, which is suitable for smart applications deployed across the Edge-to-Cloud computing continuum.

The concept of trust is complex similarly to many aspects of human endeavour [8]. A suitable definition of trust was presented by Gambetta [9]: *trust (or, symmetrically, distrust) is a particular level of the subjective probability with which an agent assesses that another agent or group of agents will perform a particular action, both before he can monitor such action (or independently of his capacity ever to be able to monitor it) and in a context in which it affects his own action.* In computer science, there have been efforts to formalise trust, for example, to facilitate cooperation among autonomous agents [10]. The formalisation of trust, therefore, aims at improving the possibilities for trust management in dynamic and distributed computing environments. In this context, Viljanen presents various considerations towards the definition of an ontology of trust [11].

The present study is set in the context of the DECENTER's Fog Computing Platform[1], which is designed to orchestrate AI methods across the Edge-to-Cloud computing continuum. DECENTER supports the development and operation of various smart applications. It can be used to implement and operate dynamic Big Data pipelines starting from cameras and sensors up to AI methods that are implemented in containers (e.g. TensorFlow). Intermediate results of the operation of smart applications can be stored in Fog or Cloud storage. DECENTER applications can, therefore, run across several tiers (Edge, Fog, Cloud) of a decentralised architecture. In an environment like this, trust is an essential aspect that must be managed, so that data can be acquired from cameras and sensors, processed by Fog computing providers, and, if necessary, persistently stored in Cloud storage.

Specifically, the DECENTER platform incorporates an orchestrator, which supports the needs of various smart applications. During its operation, the orchestrator uses advanced resource scheduling and load balancing algorithms and addresses a variety of non-functional requirements, including Quality of Service (QoS), availability, privacy and security requirements. Due to the dynamic nature of the applications and their complexity, the DECENTER platform cannot rely solely on traditional trust systems, such as the Public Key Infrastructure (PKI) hierarchical trust system or on a social trust system based on entity relationships. Practical implementations of trust management systems suitable for addressing various Edge-to-Cloud computing scenarios are currently missing, and this is the gap addressed by the present study.

The goal of the present work is, therefore, to analyse key aspects and attributes to trust important for smart applications and environments, and to design and implement a new trust management system that can be used in various dynamic Edge-to-Cloud computing scenarios. Our practical goal is to also integrate this innovative architecture with the advanced DECENTER Fog Computing Platform.

The proposed trust management approach relies on Ethereum as a specific blockchain network implementation. Essentially, blockchains remove the necessity of a trusted third-party and are suitable to be used in highly decentralised environments, where all parties (e.g. cameras/sensors, Edge/Fog/Cloud computing resources, Cloud storage providers) require a degree of autonomy in their operation. Blockchains allow all stakeholders to participate in maintaining an immutable ledger whose data is consistent among all participants, thus providing transparency and traceability to interactions, which have already been proven to contribute significantly to achieving trust in financial transactions. Furthermore, autonomous behaviour is supported through the use of Smart Contracts (SCs), which are computer protocols intended to digitally facilitate, verify, or enforce the negotiation or performance of contracts among the participants. For example, a Fog provider can autonomously provide computing resources to a customer, and one or more SCs can govern all required interactions (e.g. starting/terminating containers at the Fog provider, payment transactions, and similar). The execution of SCs takes place on the blockchain due to which all transactions are irreversible and traceable, and this improves credibility. Finally, the design and implementation of Smart Oracles and their use in the context of achieving trust is currently a hot research topic [12]. Smart Oracle is a specific software service which is used to assess and provide values of specific metrics, such as QoS metrics to an SC. It can be designed as trustless (off-blockchain), decentralised and purely functional machine upon which assessment the execution of the SC depends [13]. The use of Smart Oracles reduces the necessity of costly transactions on the blockchain and represents a mechanism for balancing the use of on-blockchain and off-blockchain data in the operation of SCs. Ethereum's blockchain, SCs and specially designed trustless Smart Oracles represent the basis of the new trust management architecture presented in this study.

Following are the contributions of this study:

[1] https://www.decenter-project.eu/

- trust attributes analyses and identification in order to address the requirements of dynamic, complex and multi-tier smart applications and environments,
- a blockchain-based trust management system applicable to multi-party decentralised Edge-to-Cloud computing,
- especially designed Smart Contracts and trustless Smart Oracles, which jointly support trustful autonomous transactions among the parties while reducing the transactions' costs, and
- implemented proof-of-concept trust management scenarios for the DECENTER Fog Computing Platform involving the registration of participants (users and providers) for transparency and traceability, access to decentralised resources including cameras (video streams), Fog and Cloud providers, and traceability of the data flow among the resources (camera, Fog, Cloud) with regard to security and privacy preservation requirements.

The rest of the paper proceeds as follows. Section 2 presents the State-of-the-Art and identifies the gap addressed by the present study. Section 3 explains the motivation behind achieving trust in the Edge-to-Cloud computing continuum. Section 4 describes an architecture of the proposed trust management system. Section 5 describes proof-of-concept trust management scenarios implemented for the study. Section 6 discusses the conducted experiments. Section 7 concludes the paper and presents our plans for future improvement of the proposed trust management system.

2. Background

Three main aspects form the basis for our present development: new approaches and technologies for application orchestration in Edge, Fog and Cloud computing environments, widely used distributed ledger technologies and existing trust modelling approaches, suitable for decentralised environments. They are elaborated in the following subsections.

2.1. Computing in the Edge-to-Cloud continuum

These days, the use of IoT devices provides new automation opportunities in practically all domains [14, 15, 16, 7]. However, the IoT devices are known to generate the Big Data problem, which requires to address the great variety, velocity, veracity and volume of data through new approaches to software engineering. Data may include various sensor measurements, images and video streams. Implemented Big Data pipelines starting from the IoT devices to processing software running in the Cloud currently suffer from low QoS, which may be due to high latency and low bandwidth of the connectivity between the Edge and the Cloud, the greatly varying processing requirements of the AI methods and the quality of the underlying computing infrastructures. Edge and Fog computing have emerged recently as a means to address such requirements. Proponents of Edge and Fog computing diverge from existing centralised high-performance Cloud data centres and advocate the use of highly decentralised computing platforms which provide computing nodes in close proximity of the data sources [17, 18]. Edge computing is a highly distributed approach that performs computational operations on multiprocessor devices (e.g. Raspberry Pi, BeagleBoard) that operate in proximity of the sensor devices [19]. On the other hand, Fog computing is similar to Cloud computing, only that it uses less powerful computing resources and processes data within the network, between the IoT devices and the Cloud computing data centres [20].

A recent study investigated various QoS aspects of video streaming applications when the application server is implemented by using Docker containers and orchestrated by using Kubernetes [21]. This approach makes it possible to establish video streaming applications on the fly. The applications run within software-defined data centres horizontally (i.e. globally, across a large geographic area) or vertically (i.e. across multiple computing tiers in the Edge-to-Cloud continuum). However, the deployment of AI empowered smart applications across such dynamically aggregated computing resources opens trust problems. Consequently, the technologies used to build a software-defined data centre must address trust-related problems.

2.2. Blockchains, Smart Contracts and Oracles

The blockchain technology is a distributed ledger technology that can be used to address several requirements of distributed and decentralised systems. These include transparency, traceability, autonomy, privacy, data management and similar. The blockchain technology replaces trusted centralised institutions by distributing trust in the decentralised network. In order to fully exploit its potential system architects must carefully define the requirements of the distributed and decentralised system. They must focus on the agreement policy among the sub-systems and select the adequate blockchain topology ecosystem that offers additional advanced functionalities such as SCs. Thus, in order to ensure trust, an agreement policy (i.e. consensus protocol) has to be satisfied. In particular, the majority of blockchain network participants have to agree on the modification of the blockchain ledger.

Bitcoin (2009) is the first practical implementation of blockchain [22]. Its simplicity of just sending and receiving digital assets encouraged many researchers and blockchain enthusiasts to develop their blockchain cryptocurrencies. Vitalik et al. [23] presented an exciting concept with the introduction of Turing complete SCs that are similar to general (notary) contracts with limited, but at the same time sufficient functionalities to cover a wide range of use-cases.

Using blockchain and SCs within existing Cloud architectures has much potential. Carminati et al. [24] investigated blockchain as a platform for secure inter-organisational business processes management. Zhang et al. [25] presented *TOWN CRIER* (TC) aiming to provide trustworthy (trustful) data to SCs through a middleman service (TC Server). Furthermore, Smart Oracles are useful means that reduce the necessity of costly operations on a blockchain, such as storing and using data within SCs. Specifically, external data provided by Smart Oracles can be used within an SC in order to decide, if a Fog node can be trusted, and consequently used to deploy an AI container on the Fog node automatically. Advanced Smart Oracle solutions, such as Oraclize[2] provide Smart Contract templates, which ensure Oracle correct data flow. Another Smart Oracle solution is the Ethereum based Chainlink network[3] that provides reliable tamper-proof inputs and outputs for SCs on any blockchain. These few useful studies form the basis for the present work, which aims at using blockchain, SCs and Smart Oracles to provide transparency, traceability and a great level of autonomy to the DECENTER's Fog Computing Platform.

2.3. Trust attributes

This study focuses on trust as a fundamental factor in achieving dependable interaction between entities, stakeholders and services in a decentralised environment underpinned by dynamic IoTs. Several recent studies have presented trust-related issues and proposed various trust modelling approaches and attributes.

Alrawais et al. [26] investigate the requirements for trust management and find that a trusted system has to maintain reliable service, prevent incidental failures and handle misbehaviour issues. Bimrah et al. [27] describe reputation, security, risk, initial trust and privacy as the essential concepts that define trust. Furthermore, Carradini et al. [28] extend the understanding of trust with additional related concepts including dependability, security and reliability.

Some trust-related concepts include the ability of the system to respond by satisfying specific QoS attributes, such as response time shorter than 50ms or throughput higher than 4 Mbps. Thus, trust can also be associated with the system's performance and quality attributes. Accordingly, a Fog computing platform, such as DECENTER will be trusted, if it is capable of satisfying application specific QoS requirements.

In other words, trust management requirements depend on the specific use cases where trust management must be applied and are multifaceted. Figure 1 presents several relevant trust-related attributes. These are:

- *Availability* is a fundamental attribute of Fog nodes that evaluates the probability of the node's correct functioning at a specific moment in time.
- *Credibility* defines the degree to which the data source or the data is seen to be believable, a concept that can be extended to any data item, such as a video frame or an AI model based on TensorFlow.
- *Privacy* in the context of the IoT includes the following relevant aspects: awareness of security risks imposed by smart devices surrounding a human subject, individual control over the data collection and processing of personal data, and awareness and control of subsequent use and dissemination of personal information by those entities to any entity outside the subject's personal control sphere[29].

[2] http://www.oraclize.it/

[3] https://chain.link/

Please cite this article as:Kochovski, P., Gec, S., Stankovski, V., Bajec, M., & Drobintsev, P. D. (2019). Trust management in a blockchain based fog computing platform with trustless smart oracles. Future Generation Computer Systems.
doi: https://doi.org/10.1016/j.future.2019.07.030

- *Response time* is an attribute that represents the time necessary for data to be processed by a selected Fog node and data packets to be transferred to the client. For simplicity, in the present implementation, it is measured as the round trip time for the package to reach a microservice, perform necessary processing, and return extracted metadata to the client.

- *Throughput* is an attribute that represents the rate at which data is transferred between endpoints (e.g. between a sensor and a Fog node). In other words, this attribute shows the amount of data that a specific component can successfully transfer per unit of time.

- *Security* estimates the ability to protect the system from accidental or intentional external attacks. This attribute to trust is closely related to confidentiality that evaluates if the data within the distributed environment is protected from disclosure to unauthorised entities. For instance, Hu et al. [30] analyses and summarises the security and privacy issues in face identification platforms that are based on Fog computing in order to provide a trusted service.

- *Transparency* is an attribute that allows the blockchain ledger to be fully auditable. That allows everyone participating in the ecosystem to view the stored transactions on the blockchain.

- *Traceability* is an attribute that is closely related to transparency. Traceability allows to trace back the interaction between entities on the blockchain since all history can be traced back to the first transaction.

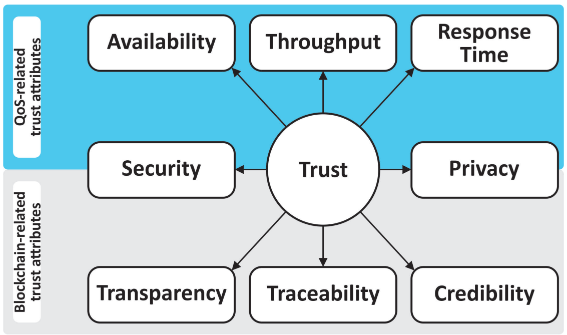


Figure 1: Trust attributes considered for implementation

2.4. *Trust management approaches and positioning of the present work*

Our novel trust management approach considers specific trust attributes and presents an architecture of a distributed system along with implemented mechanisms, which can be used to verify actions against trust and security policies. Table 1 aligns the present approach with other trust management approaches according to some key aspects relevant for Fog computing platforms. Table 1 supports the conclusions of some recent review studies [31], [32], [33] that most research in trust management has been concentrated in the area of edge computing in general, and mobile computing in particular. There is a notable lack of trust management solutions for Fog computing due to the challenge of monitoring and evaluating the behaviour of a large number of heterogeneous and distributed Fog nodes in the network while maintaining satisfactory QoS.

A large amount of trust management solutions can be separated into two major trust models: evidence-based and monitoring-based trust models [33]. A monitoring-based model can be any model that instantiates trust based on the observed behaviour of past interactions (e.g. social networking, cooperative solutions, crowd-sourcing, etc.) between entities. For example, Habib et al. [34] developed trust management mechanism that aids users to identify trustworthy Cloud providers that offers two approaches to calculate the trust score of Cloud providers. The default approach implements The Consensus Assessments Initiative Questionnaire (CAIQ) that is provided by the Cloud Security Alliance (CSA) for Cloud consumers and auditors to assess the security capabilities of a Cloud service provider. The second approach allows the user to tailor the trust requirements according to their needs and receive

Table 1: Trust management approaches relevant for Fog computing

Study	Trust model	Computing Environment	Parameters	Blockchain	Smart Oracles	Application areas	Implementation
Habib et al. [34]	Opinion-based, prior-experience evidence	Cloud	Availability, Security, Latency, Support	No	No	Multifaceted trust management for cloud marketplaces	Partial
Mostajeran et al. [35]	Monitoring, policy-based	Edge	Host credibility, container reliability, system reliability, container risk, system risk	No	No	Multifaceted framework for container trust management	Yes
Prajapati et al. [36]	Recommendation, reputation-based	Cloud	Reputation degree, recommendation	No	No	Software as a Service trust management model	No
Chen et al. [37]	Social-based model	Edge	Node reputation	No	No	Collaborative socially trusted computing in small cell base stations	Simulation
Sharma et al. [38]	Social-based model, Crowdsourcing	Edge	Availability, Integration, Cost	No	No	Fake news detection	Simulation
Li et al. [39]	Social-based, Collaborative recommendations	Edge	Data trust, Node trust	No	No	Secure vehicular ad hoc network	Simulation
Wang et al. [40]	Recommendation-based	Fog, Cloud	Packet loss rate, route failure rate, forwarding delay	No	No	Trust management mechanism for sensor-cloud systems	Simulation
Chen et al. [41]	Social-based, recommendation-based	Edge	Rating of friendship, social contact, community of interest relationship	No	No	Protocol for determining trust among edge devices	Yes
Soleymani et al. [42]	Fuzzy model	Edge	Availability, Authentication, Message Integrity, Confidentiality, Validation	No	No	Secure vehicular ad hoc network	Simulation
Chen et al. [43]	Fuzzy model	Edge	Packet forwarding/delivery ratio, energy consumption	No	No	Detect node behavior in IoT system	Simulation
Mora-Gimeno et al. [44]	Encrypted communication	Edge	User/server authentication data	No	No	Security model for multi-tier mobile edge computing model	Yes
Di Pietro et al. [45]	Blockchain-based	Edge	Proof of fulfillment, terms of use, obligations	Yes	No	Trust management for edge devices with different domains	Yes
Alexopoulos et al. [46]	Blockchain-based, Reputation-based	Edge	Reputation ratings	Yes	No	Distributed trust management in IoT system	No
Yu et al. [47]	Blockchain-based	Edge	Not specified	Yes	No	Platform for edge device and data tracking and trading	No
Hammi et al. [48]	Blockchain-based authentication	Edge	Not specified	Yes	No	Secured virtual zones for devices to communicate securely	Yes
Missier et al. [49]	Blockchain-based	Edge	Not specified	Yes	Yes	Decentralised data marketplace	Simulation
Present approach	Blockchain-based	Edge, Fog, Cloud	Location, Availability, Response Time, Ranking score	Yes	Yes	Trusted edge devices participating in IoT environment; trusted fog nodes that operate with potentially sensitive data; trusted datacenters that provide persistent storage services; trusted deployment decision-making process for containerized microservices.	Yes

personalised resultspersonalised. Mostajeran et al. [35] proposed a multifaceted run-time trust framework that is based on Edge node's security assessment. The framework monitors the Edge nodes and is capable of identifying Edge-server security vulnerabilities, detect unauthorised actions and thus categorise Edge nodes based on the estimated level of trust. Prajapati et al. [36] introduced a trust management model for calculating the trust level between the user and Software-as-a-Service providers based on past usage experience and implement concepts such as reputation and satisfaction level. Chen et al. [37] described a socially trusted collaborative Edge computing platform for dense networks, which implements payments as an incentive mechanism for edge nodes to collaborate. In particular, instead of offloading the workload to a remote cloud, an overloaded Edge node could pay nearby nodes that have spare computing resources in order to process the remaining workload. The study of Sharma et al. [38] focuses on trust and privacy by building social relationships between the IoT devices, where crowd-sources have a role of mini-Edge servers and entropy modelling for maintenance of trust.

Furthermore, the trust management scheme of Li et al. [39] can provide real-time protection from malicious attacks in traffic. Moreover, the scheme evaluates the trustworthiness of data and nodes (e.g. cars) within a vehicular ad hoc network. It allows us to determine the degree to which data can be trusted and whether nodes can be trusted. Similarly, Wang et al. [40] proposed a Fog-based hierarchical trust mechanism framework for detecting hidden data attacks and ensure communication only with credible edge nodes. In this study, the Fog computing layer acts as a trust buffer between the Cloud computing layer and the sensor network layer. In particular, it detects the trust state of the wireless sensor network, monitors their behaviour and performs analysis tasks. Chen et al. [41] described a distributed trust management protocol, where each user maintains its trust assessment towards devices. This protocol implements a filtering technique based on similarities in the social interests of the IoT nodes' owners. The studies [42] and [43] proposed implementing fuzzy logic for trust management in order to evaluate trust from multiple parameters, such as reputation, authentication, confidentiality and so on.

Evidence-based trust model can be any model that proves trust relationship among entities based on public-key, address, identity or any similar evidence that the entity can generate for itself or other users. For instance, Mora-Gimeno et al. [44] present a security model for data processing that combines applications to multi-tier mobile edge architectures with ability to adjust the security levels of each tier (i.e. each tier requires authentication). The model is based on the degree of trust that each level may have, which requires a different level of security. It determines the number of security mechanisms to be used and the degree of trust for each component.

Lately, a novelty in evidence-based trust models are the blockchain-based trust models, which allow full transparency and traceability. Unlike the previously known models, the blockchain-based trust model is not dependant on a trusted third-party certification authority. Di Pietro et al. [45] proposed a distributed trust model for IoT based on blockchain technology that assures trust between IoT devices. The devices are grouped by domains in so-called ”islands of trust”, where trust is not established between groups. The model establishes trust among devices belonging to different domains by implementing a three-way handshaking protocol. Alexopoulos et al. [46] proposed a blockchain-based trust management system that allows global scalability to different clusters. For that purpose, the authors designed a three-layered architecture for trust management in IoT. The first layer is composed of devices in close proximity. The second layer corresponds to a decentralised ledger that governs a specific subset of embedded devices. The third layer is a global decentralised ledger that provides guarantees for access delegations between devices and users belonging to different layers. Yu et al. [47] introduced a trustworthy trading platform, that allows trading of IoT devices or data generated by IoT devices between multiple entities, such as device manufacturers, retailers and users. In order to achieve high trust between entities, they utilise different permissions for each entity when trading with devices and data over the blockchain. As a result, no entity can cheat others or modify the existing data related to a specific device. Hammi et al. [48] introduced a new concept called Bubbles of Trust, which represent secure virtual zones in IoT environments. It groups devices in trust zones. Therefore, only devices that belong to the same zone are allowed to communicate with each other, whereas every other device is considered as malicious. In this study, communications between devices are considered as transactions and are validated by their public blockchain. Missier et al. [49] proposed a conceptual model of a decentralised data marketplace for data produced by Edge devices. Their decentralised marketplace is designed to dictate the data price base on data quality, demand and offer. To provide a higher degree of trust between the producers and consumers of data, the authors propose transparent blockchain transactions.

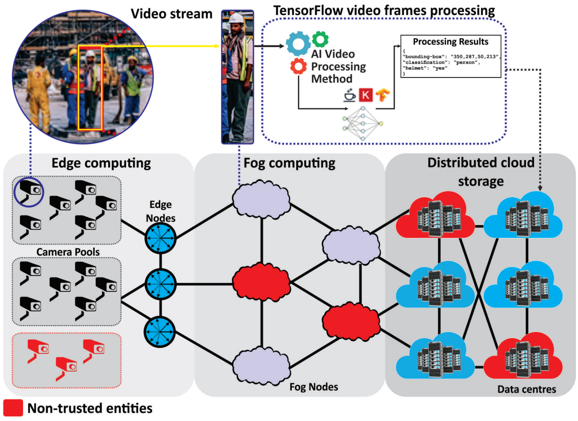
The present work complements the above efforts by focusing on blockchain-based trust management. In contrast to previous studies the implemented trust management architecture assures trust for complex interactions among humans and artificial agents across the complete Edge-to-Cloud continuum. It also addresses some limitations of previously implemented blockchain-based trust models, particularly, it functions based on less costly off-chain data which is facilitated through the use of trustless Smart Oracles.

3. Motivation

The need to develop smart applications for the civil engineering domain in general, and the construction sector, in particular, motivates this study [6]. Here, many emerging smart applications address requirements for construction monitoring [50, 51], construction progress tracking [52], early disaster warning [53], safety at work [54, 55] and similar. Expected benefits from using such applications are improved safety, productivity and use of assets and resources among others.

Existing solutions rely on well-known service providers [56]. However, the emergence of construction sites that use many IoT devices (cameras and sensors) raise new issues of trust with all its related aspects [56, 26]. This problem becomes particularly concerning when sensitive data, such as video streams have to be processed in leased infrastructures, such as Fog nodes and stored for future use in Cloud storage.

Here, we focus on the design of a smart container-based application, which is designed to implement video surveillance for safety at work measures as shown in Figure 2. The AI empowered application uses video-surveillance footage as input data for the detection of safety violations. For example, a video frame which is an input to deep learning (TensorFlow) method is used to check if a worker wears a helmet. In case the worker does not wear a helmet a safety alert is issued to the construction supervisor.



```
{
"bounding-box": "350,287,50,213",
"classification": "person",
"helmet": "yes"
}
```

Figure 2: Smart applications' data flow in the Edge-Fog-Cloud continuum

This scenario reveals a variety of trust-related problems. For example, it is necessary to trust the actual on-site devices, cameras and sensors, that are used to provide input to the AI methods. Trust issues may be even more critical in the case when the application uses cameras that dynamically enter and exit the smart construction environment, for example, cameras mounted on workers helmets. Consequently, it is necessary to build an application that relies on a pool of trusted devices, such as Trusted Cameras Pool.

Following this, it is necessary to assure that the Fog nodes used for processing sensitive (video streaming) data are also trusted. The data (video frames) are forwarded to a selected Trusted Fog Infrastructures Pool, where the selected AI methods can be started in order to analyse the incoming video frames and detect safety violations. Additional investigated trust-related problems include the necessity to trust the actual AI method providing the service, the network path used to transport sensitive information, and the ability of the computing resource - AI method pair to respond within an adequately short period.

Finally, in case the AI method detects a violation (e.g. construction worker without a helmet), a safety alert is sent to the construction site manager, and a log of the event containing captured video frames is generated and stored in Cloud storage. The Cloud storage must also be trusted as storing sensitive data in the Cloud inevitably involves privacy and security issues.

Returning to the used definition of trust, from a technical viewpoint trust can be described with some probability, however, the trust management approach employed by the Fog Computing Platform should rely on binary decisions: trusted or not trusted.

4. Architecture

Having presented the background, emerging smart video streaming applications and their requirements for trust management, we introduce a trust management system designed for DECENTER's Fog Computing Platform. Figure 3 depicts a multi-level architecture of the trust management system that follows interoperability standards set by organisations such as the Cloud Native Computing Foundation (CNCF)[4], OpenFog Consortium[5] and Edge Computing Consortium Europe (ECCE)[6]. Each level in the architecture is used at a different stage of the video stream processing Big Data pipeline and is described in the following.

1. *Application Layer* is the entry point for the user that wants to manage trusted cameras, Fog nodes or Cloud storage repositories. This layer is connected to the Ethereum ecosystem through the Ethereum bridge Metamask[7], which is a browser add-on.

[4] https://www.cncf.io/
[5] https://www.openFogconsortium.org/
[6] https://ecconsortium.eu/
[7] https://metamask.io/

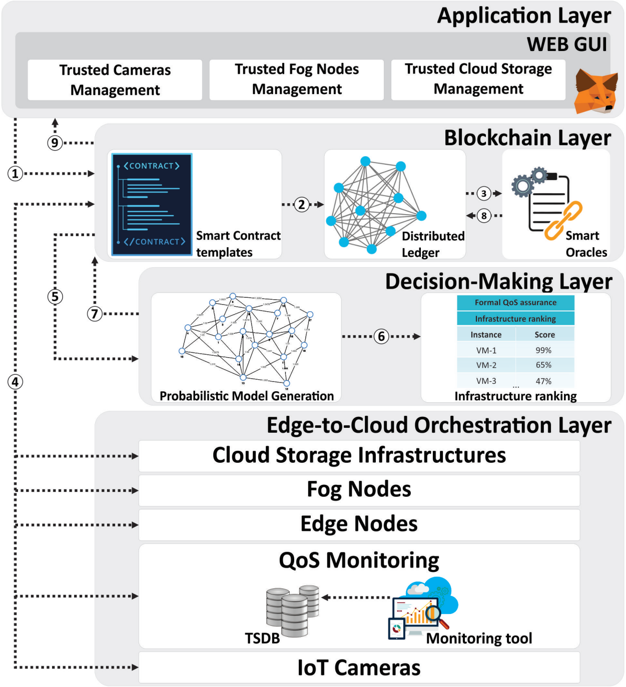


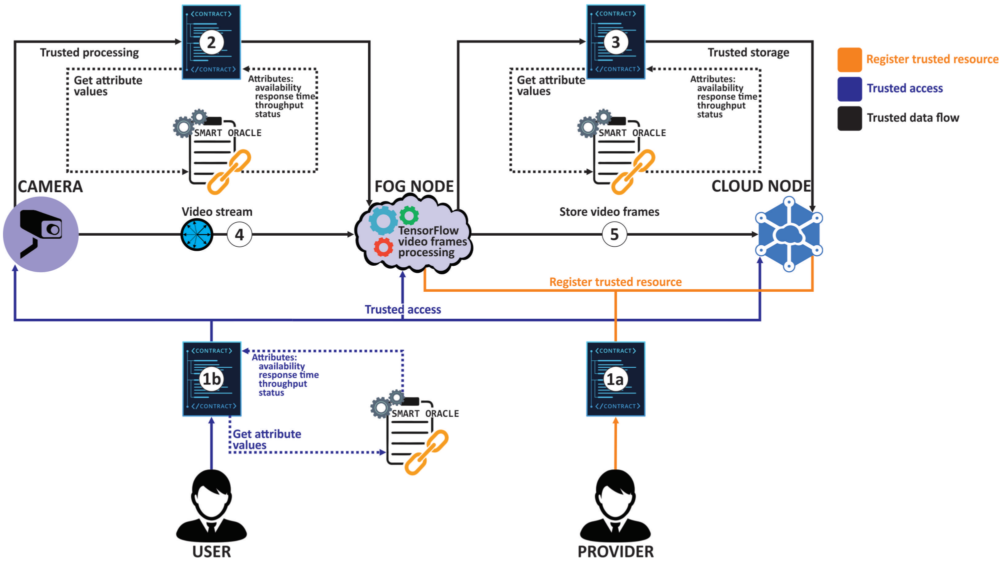
Figure 3: High-level architecture of the trust management system

2. *Blockchain Layer* uses the Ethereum (ETH) ledger as pillar blockchain environment that enables SCs. The two main components are SC templates and Smart Oracles. The deployment of the SCs occurs on demand through the blockchain service which is owned by the system. Each SC instance communicates with external services (e.g. Fog nodes, QoS monitoring system) through registered APIs that are part of external services. This approach results in enhanced integrity of the functions that verify the correctness of the API queries by using unique API keys and thus avoid calls from potential malicious SCs. The design of the SCs followed the oracle pattern proposed by Wöhrer et al [57] and best practices presented in the framework OpenZeppelin[8].
3. *Edge-to-Cloud Orchestration Layer* is composed of infrastructures (Cloud storage repositories, Fog and Edge nodes) for the deployment of containerised microservices and data, QoS monitoring of the infrastructures and IoT devices (cameras). The Cloud storage repositories, Fog and Edge nodes are the infrastructures that are available at runtime. Each of these infrastructures plays a different role in the video stream processing pipeline. Edge nodes are responsible for rapid data acquisition and data normalisation; Fog nodes are responsible for data processing, compression and transformation and Cloud storage repositories store the data upon request. The orchestration capability is realised by using Kubernetes and is thoroughly described elsewhere [21]. Before using any of these infrastructures in the other layers of the architecture, they are first verified on the blockchain if they satisfy trust including QoS requirements. This layer continuously monitors the infrastructures and gathers QoS related information. The monitoring data is available to the blockchain layer through the blockchain Smart Oracles. Besides the layer consists of all available IoT cameras that the user can verify through the blockchain and use their video streams.
4. *Decision-Making Layer* is responsible for determining an optimal infrastructure for the deployment of containerised microservices (e.g. TensorFlow). This layer uses a Markov probabilistic decision-making method for automated decision-making [58]. In order to rank the infrastructures, the Markov method requires QoS monitoring data and QoS threshold values from a specially designed Smart Oracle. These metrics are necessary to generate a probabilistic finite automaton, which is later used to produce infrastructure ranking list. The

[8] https://openzeppelin.org/

first ranked infrastructure of the ranking list is considered as an optimal deployment option and returned to the certified user (i.e. to the trusted user) as a trusted infrastructure that satisfies the user's QoS requirements.

The operation of the above described trust management system is explained in the following section.



2
3
Trusted processing
Get attribute values
Attributes: availability response time throughput status
SMART ORACLE
Trusted storage
Get attribute values
Attributes: availability response time throughput status
SMART ORACLE
Register trusted resource
Trusted access
Trusted data flow
CAMERA
FOG NODE
CLOUD NODE
Video stream
4
TensorFlow video frames processing
Store video frames
5
Register trusted resource
Trusted access
1b
Attributes: availability response time throughput status
SMART ORACLE
Get attribute values
1a
USER
PROVIDER

Figure 4: Using Smart Contracts in the operation of a video stream analysis application

5. Proof-of-concept trust management scenarios

Our new trust management system was implemented in order to support smart multi-tier i.e. Edge-to-Cloud applications. The main purpose of our container-based video surveillance smart application is to prevent safety violations at a construction site and to timely predict and prevent injuries. The smart application subscribes to video streams from trusted cameras, processes them by using TensorFlow at Fog nodes and records marked video frames on a trusted Cloud storage for later use.

In order to ensure trusted operation, it is necessary that each person as well as hardware and software entity in the distributed environment is trusted. Moreover, it is also necessary that the data flow among the application components is also trusted. Here, the smart environment must precisely distinguish trusted surveillance cameras, Fog nodes and Cloud repositories from non-trusted infrastructures in the open environment.

Figure 4 presents the investigated proof-of-concept trust management scenarios. A provider authenticated by using a blockchain wallet registers a Fog node or a Cloud storage on the blockchain (1a). Then, another user also authenticated by using a blockchain wallet can subscribe to a trusted camera video stream or use a Fog node for AI processing or Cloud storage for storing sensitive data only by executing a SC that takes into account QoS and trust related requirements (1b). Before the deployment of the smart application, an SC containing the previously described Markov method executes and selects a Fog node that satisfies the QoS requirements of the smart AI application (2). This part of the process also verifies that both entities have their valid blockchain wallets. For the Fog node to be able to forward the processing output or specific video frames to a Cloud storage node, it is necessary to execute another SC, which selects a Cloud storage node satisfying the QoS requirements based on the Markov model, and again verifies that both entities are connected to valid blockchain wallets (3). Once trusted connectivity is assured, the video starts streaming from the camera to the Fog node for AI processing (4). The processing output is stored according to the application's needs on trusted Cloud storage (5).

The flow of interactions between the pillar system components in our scenarios is depicted in Figure 5. More details are provided in the following subsections.

5.1. *Trusted video stream access scenario*

This scenario aims at assuring trusted video stream access is shown in SC Listing 1. The process starts when the user sends a request to the blockchain service through the Web GUI. The blockchain service deploys the instance of the SC with basic price estimation and triggers the selection of the camera through a specific Smart Oracle dedicated API. This process ensures that the payment is made only to a trusted camera service. In the next phase the prices are additionally set up by the Service Pool and Camera Pool components (function *determinePrice()*). Each execution of this function is completed with Ethereum event trigger that notifies the listening services. After a successful price determination and trust management decision the consensus is reached (getter function *consensusFinished()*) and the certified user is asked to pay the specified price (function *payService()*) in order to start the camera session. Besides, the camera deployment service is triggered in this function through a Smart Oracle dedicated API that deploys the actual container image instance service. Similar to this scenario are the scenarios for assuring trust for Fog nodes and Cloud storages.

5.2. *Trusted data flow scenario*

Similar to the *Trusted video stream access* scenario is the *Trusted data flow* scenario. The only difference of the implementation is in the different Smart Oracle APIs endpoints, which are now related to the Infrastructure Pool services.

Because the operational cost is an important part of this decentralised system, the trust management scenarios may follow fixed or pay-as-you-go pricing methodology. The users may not always know in advance the intended usage period of the smart application; they may be willing to pay a full price which is locked into the SC (function *payService() in Listing 2*). The user's session ends automatically when the duration determined in the Smart Contract is exceeded (function *checkLockState()*) or on demand by the user, where ETH funds reimbursement to the user has to be calculated. In both cases, undeployment of the user's session is triggered through a dedicated Smart Oracle API. This process can only be triggered manually by the user or automatically when the exceeded time condition is reached. The undeployment process terminates the user's microservice, deletes the footprint, and releases the allocated resources. Finally, the user receives a notification about the successful completion of the video streaming and AI processing session. The logic behind the use of SCs is illustrated in Figure 6.

The transparency and traceability of the data flow among the entities which is achieved through the same Smart Contract shown in Listing 1 represents a trust building measure. However, the only difference is the price determination and off-chain data sources. For instance, instead determining prices from the Camera Pool, this scenario determines the prices from the Infrastructure Pool.

The trust in the Cloud storage can be significantly enhanced by improving the security and privacy of the stored data. For example, security and privacy in such scenarios can be improved by exploiting high distribution rate of cloud repositories and by splitting the data among them. This is the case of Storj[9], Sia[10] and other approaches that use fragmentation and distribution of data.

```
pragma solidity ˆ0.4.25;
import "./usingOraclize__future.sol";

contract TrustedCamera is usingOraclize__future {
    address public owner;
    address public endUserAddress;
    uint public price;
    uint public priceSP;    uint public priceC;
    bytes32 queryId;
    bool consensFinished = false;
    mapping(bytes32 => bytes32) orc_cameraId_commitment;

    event PriceNotification(address _owner, uint _price, bytes32 camera_id_comm);
    event StartService(bool _successPayment, uint _currentTime, uint _releaseTime,uint _price);

    constructor(uint _price, address _endUserAddress) public {
        require(_price > 0);
        price = _price;
        owner = msg.sender;
```

[9] https://storj.io/

[10] https://sia.tech/

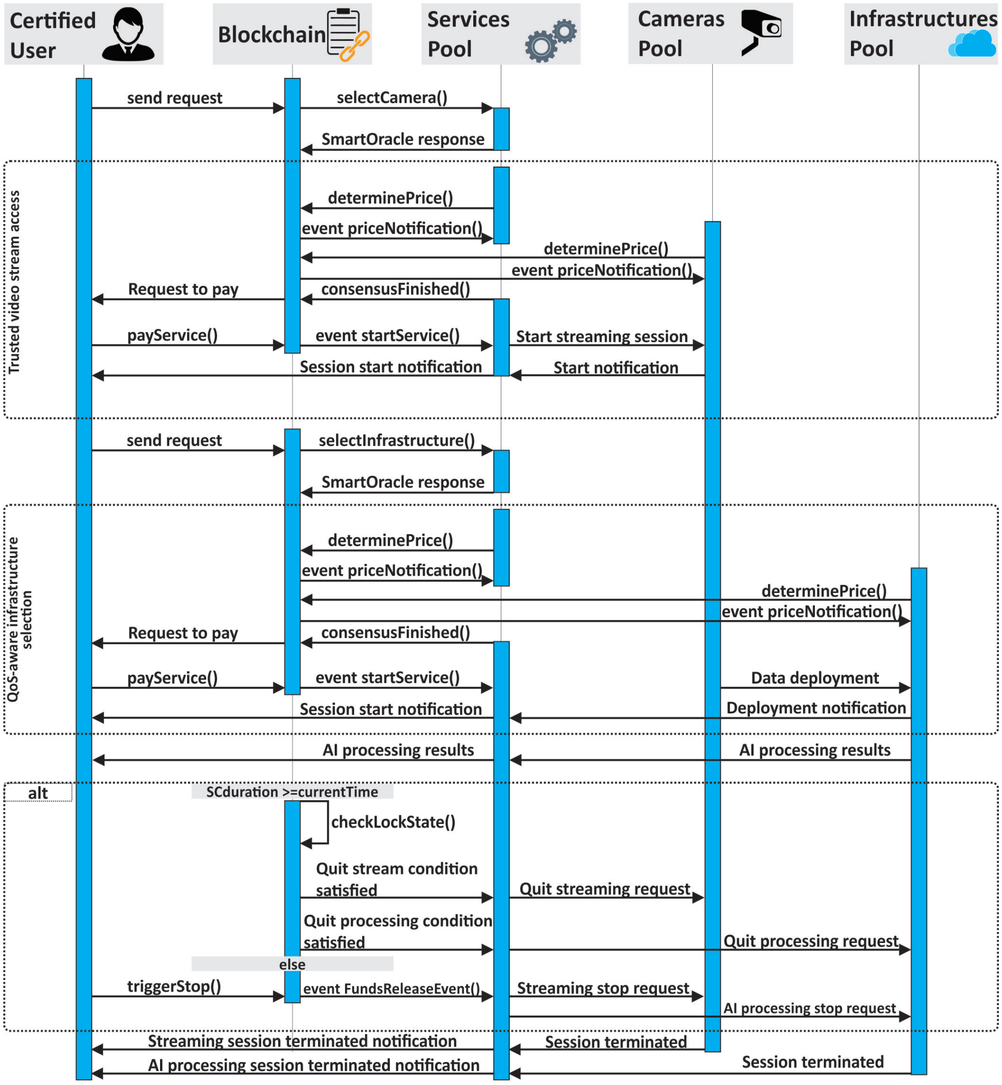


Figure 5: Sequence diagram for the trust management scenarios

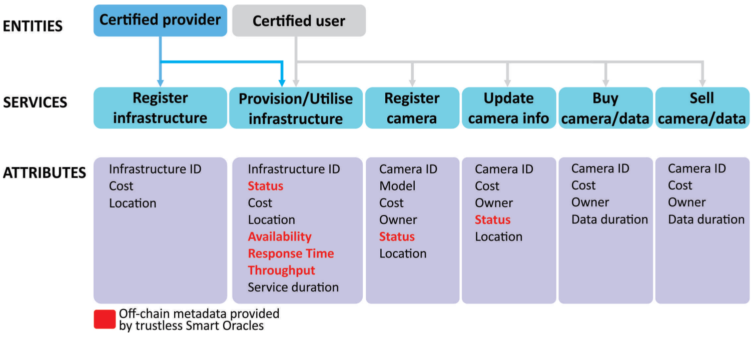


Figure 6: Smart Contracts logic

```
        endUserAddress = _endUserAddress;
        // select camera
        oraclize_setProof(proofShield);
        string memory query = "json(SERVICE_POOL_URL_REQUEST_WITH_ARGUMENTS)";
        queryId = oraclize_query("URL", query);
        orc_cameraId_commitment[queryId] = keccak256(sha256(query), proofShield);
        emit PriceNotification(owner, price, queryId);
    }

    function determinePriceSP(uint _priceSP) public {
        require(msg.sender == endUserAddress);
        require(priceSP == 0);
        priceSP = _priceSP;

        if (priceC != 0)
            consensFinished = true;
    }

    function determinePriceCamera(bytes32 query, uint _priceC) public {
        require(orc_cameraId_commitment[queryId] == keccak256(sha256(query), proofShield));
        require(priceC == 0);
        priceC = _priceC;
        if (priceSP != 0)
            consensFinished = true;
    }

    function payVCService() public payable {
        require(consensFinished == true);
        require(endUserAddress == msg.sender);
        require(msg.value >= (price + priceSP + priceC));

        msg.sender.send(msg.value);
        uint currentTime = now;
        emit StartService(true, currentTime, 0, (price + priceSP + priceC));
    }

    function consensusFinished()public returns(bool) {
        return consensFinished;
    }
}
```

Listing 1: Smart Contract template for the Trusted Camera scenario.

5.3. *QoS-aware Fog node selection scenario*

Thanks to the use of a specially designed Smart Oracle, our trust management system uses QoS monitoring metrics of existing Fog nodes. These collected metrics are fed as input to the Markov decision-making process to rank the available Fog nodes according to the hardware requirements of the application.

The partial SC Listing 2 presents a monitoring service, which is triggered through the developed Smart Oracle service type URL (function *triggerStop()*). In order to increase the external service calls the SCs uses the TLSNotary[11] authentication mechanism. The scheme follows three main roles *auditor* (e.g. a locked-down AWS instance of Amazon Machine Image), *auditee* (e.g. Oraclize) and Webserver (e.g. camera video service or QoS monitoring service).

An important trust attributed is the expected QoS of the smart application. In order to analyse this attribute, we a user run the smart application from a location in Ljubljana, Slovenia. The user who has a wallet on the blockchain ledger select a camera from a pool of trusted cameras. The Fog nodes are represented by ARNES[12] and Google Cloud Platform[13] resources. The properties of the used infrastructures representing Fog nodes are listed in Table 2. The QoS thresholds that the user requested were set as: required response time for the application must be less than 40 ms and the availability of the Fog node must be higher than 99%.

The autonomous selection of a Fog node was done by using prior QoS monitoring information stored in the monitoring system and usage data, collected during a period of one month before the actual infrastructure ranking.

[11] https://tlsnotary.org/

[12] http://www.arnes.si/

[13] https://Cloud.google.com/

```
pragma solidity ^0.4.25;
import "./TrustedCamera.sol";

contract DynamicPriceTrust is TrustedCamera {
    uint releaseTime;
    uint lockTimeS;
    uint public durationSeconds;
    mapping(bytes32 => bytes32) oraclize_monitoring_commitment;

    event FundsReleaseEvent(uint _releaseTime, uint _lockTimeS, uint _actualReleaseTime, uint _lockedFunds,
         uint _amountReturned);

    mapping(address => AccountData) accounts;

    struct AccountData {
        uint balance;
        uint releaseTime;
        bool isFinished;
    }

    function triggerStop() {
        require(msg.sender == endUserAddress);
        require(accounts[msg.sender].isFinished != true);

        // check monitoring data
        oraclize_setProof(proofShield);
        string memory query = "json(MONITORING_URL_REQUEST_WITH_ARGUMENTS)";
        bytes32 queryId = oraclize_query("URL", query);
        oraclize_monitoring_commitment[queryId] = keccak256(sha256(query), proofShield);

        // split funds if there is any remaining time left
        if (accounts[msg.sender].releaseTime < now) {
            uint currentTime = now;
            uint gweiReturn = (accounts[msg.sender].releaseTime - currentTime) * (price / durationSeconds);

            msg.sender.send(gweiReturn);
            accounts[msg.sender].balance -= gweiReturn;
            accounts[msg.sender].isFinished = true;
            emit FundsReleaseEvent(releaseTime, lockTimeS, currentTime, price, gweiReturn);
        }
    }
}
```

Listing 2: Smart Contract template that ensures a dynamic pricing policy in a trusted environment.

Table 2: QoS measurements based on Fog/Cloud nodes

Infrastructure	Location	Availability [%]	Response Time [ms]	Throughput [Gbps]
arnes	Ljubljana, Slovenia	99.9	8.12	0.037
gke-eu-west	Frankfurt, Germany	99.99	33.14	25.74
gke-us-central	Iowa, USA	99.99	319.52	24.52
gke-asia-east	Changhua, Taiwan	99.99	629.81	27.59
gke-asia-northeast	Tokyo, Japan	99.99	549.76	25.83

The historical data was gathered during operations between clients in Ljubljana and remote Fog nodes, and was used to develop a Markov Fog nodes ranking model.

Table 3 and Table 2 present the ranking results and the QoS metrics used for the ranking of the Fog nodes. In this particular use-case, availability was estimated through the amount of downtime in milliseconds an infrastructure had in the last 30 days. All of the used infrastructures had an availability rate of 99% and higher. Response time was measured as the round trip time for data packages to reach the specific infrastructure and return a response to the user.

Table 3: Ranking results of trusted infrastructures

Infrastructure	Score	Rank
arnes	73.7	I
gke-eu-west	69.3	II
gke-us-central	35.7	III
gke-asia-east	13.6	V
gke-asia-northeast	22.4	IV

The results, presented in Table 3 show that the proposed trust management solution uses a Markov model-based ranking mechanism for trusted Fog nodes to autonomously select a Fog node with optimal performance for the specific AI application. After the successful ranking, the SC is executed, and the application is deployed on the selected Fog node.

Table 4: Performance evaluation of the developed SC for all functions and for each stakeholder type

Entity/stakeholder	Execution time [s]
Certified User	29.3
Blockchain Service	82.1
Service Pool	15.6
Camera/Fog nodes Pool	14.9

In order to analyse the performance of the actual execution of the specific SCs, multiple performance tests for the implemented trust management scenarios were run on the Rinkeby Ethereum testnet environment[14].

For performance evaluation of our trust management system, we measured the time necessary for an SC to execute and deploy data/services across the multi-tiered infrastructure including function triggers among different entities. Experiments performed on the Rinkeby Ethereum testnet were repeated 10 times. GAS limit of 21000 Gwei[15] was used to represent a default SC usage. The obtained results are presented in Table 4.

6. Discussion

In the course of our work, we implemented a decentralised blockchain-based trust management system, allowing transparent, traceable and autonomous data, software and infrastructure management and transactions among trusted

[14] https://www.rinkeby.io/

[15] https://www.investopedia.com/terms/g/gwei-ethereum.asp

entities and stakeholders. This paves the way to the seamless deployment and operation of smart applications across the Edge-to-Cloud continuum in a trusted way.

The SCs execution time and the cost of using the trust management system represent important usability indicators, which are also investigated in this study. The performance evaluation results shown in Table 4 indicate that that major waiting periods for the blockchain service occur due to the use of mandatory SC-based utility libraries within the SCs (e.g. math, Oraclize wrappers, interfaces and others). These enabling libraries are embedded within each SC that supports the use of trustless Smart Oracles. The mentioned libraries are essential in order to achieve the mechanisms described in Section 5. In case of other stakeholders the execution periods mainly match the construction time of one block due to the lightweight functions not containing any loops or complex data types such as arrays.

7. Conclusion

The implementation of Big Data pipelines across the Edge-to-Cloud computing continuum essentially requires trust management approaches. With the emergence of blockchain as an immutable ledger technology and the potential of Smart Contracts and trustless Smart Oracles, it is increasingly possible to overcome the limitations of traditional trust management approaches. In this work, we rely on transparency, traceability and autonomy as key features of blockchain-based services, and apply a new trust management approach to a highly dynamic and complex distributed smart application scenarios.

Our study considered several attributes that must be assessed and implemented in order to achieve trust in smart applications and the underlying decentralised system. While the values of some trust attributes can be obtained by using costly on-blockchain operations, other trust attributes can be used through the use of less costly off-blockchain mechanisms, such as the use of QoS monitoring data as shown in this study.

A specific approach to trust proposed by this study is the ability to certify the entities and stakeholders (users and providers, IoT data sources, software components, Fog nodes, Cloud storage), and independently monitor their status through independent blockchain based services. This study also presents SC-based trust management scenarios for achieving data flow among the application components (from the camera to a Fog node, and from a Fog node to Cloud storage). Finally, the study concentrates on achieving high QoS in the operation of the smart applications. This is facilitated through the use of off-blockchain QoS monitoring metrics gathered through the use of a trustless Smart Oracle, and a Markov decision-making method that ranks the available Fog/Cloud node providers in order to select the optimal Fog node for the deployment of the AI part of the application.

Our work builds on earlier efforts that formalised trust. Its novelty lies in the practical implementation of a trust management system, which feeds the design of the advanced DECENTER's Fog Computing Platform. The presented smart construction application demonstrates both the requirements for trust management as well as the potential of the present blockchain-based approach.

Acknowledgements

The research and development reported in this paper have received funding from the European Union's Horizon 2020 Research and Innovation Programme under grant agreements no. 815141 (DECENTER: Decentralised technologies for orchestrated Cloud-to-Edge intelligence). Funding was also received from the Slovenian Ministry for Education and Sport under grant agreement no. C3330-16-529007 (EkoSmart - Ecosystem of a Smart City).

References

[1] H. Song, D. B. Rawat, S. Jeschke, C. Brecher, Cyber-physical systems: foundations, principles and applications, Morgan Kaufmann, 2016.

[2] H. Song, G. A. Fink, S. Jeschke, Security and Privacy in Cyber-Physical Systems, Wiley Online Library, 2017.

[3] D. B. Rawat, C. Brecher, H. Song, S. Jeschke, Industrial Internet of Things: Cybermanufacturing Systems, Springer, 2017.

[4] P. Drobintsev, V. Kotlyarov, I. Chernorutsky, N. Voinov, Conceptual approach to managing technological processes of industrial iot workshop, in: 6th International Workshop of Advanced Manufacturing and Automation, Atlantis Press, 2016.

[5] N. Voinov, I. Chernorutsky, P. Drobintsev, V. Kotlyarov, An approach to net-centric control automation of technological processes within industrial iot systems, Advances in Manufacturing 5 (4) (2017) 388–393.

[6] M. Štefanič, V. Stankovski, A review of technologies and applications for smart construction, Proceedings of the Institution of Civil Engineers - Civil Engineering 172 (2) (2019) 83–87. doi:10.1680/jcien.17.00050.

Please cite this article as:Kochovski, P., Gec, S., Stankovski, V., Bajec, M., & Drobintsev, P. D. (2019). Trust management in a blockchain based fog computing platform with trustless smart oracles. Future Generation Computer Systems.
doi: https://doi.org/10.1016/j.future.2019.07.030

[7] P. Kochovski, V. Stankovski, Supporting smart construction with dependable edge computing infrastructures and applications, Automation in Construction 85 (2018) 182–192.

[8] S. A. Rompf, The concept of trust, in: Trust and Rationality, Springer, 2015, pp. 29–78.

[9] D. Gambetta (Ed.), Blackwell, Cambridge, Massachusetts, 1988.

[10] J. Sabater, C. Sierra, Review on computational trust and reputation models, Artificial Intelligence Review 24 (1) (2005) 33–60. doi:10.1007/s10462-004-0041-5.

[11] L. Viljanen, Towards an ontology of trust, in: S. Katsikas, J. López, G. Pernul (Eds.), Trust, Privacy, and Security in Digital Business, Springer Berlin Heidelberg, Berlin, Heidelberg, 2005, pp. 175–184.

[12] M. Wöhrer, U. Zdun, Design patterns for smart contracts in the ethereum ecosystem.

[13] Z. Hess, Y. Malahov, J. Pettersson, Æternity blockchain, Online]. Available: https://aeternity. com/aeternity-blockchainwhitepaper. pdf.

[14] A. Zanella, N. Bui, A. Castellani, L. Vangelista, M. Zorzi, Internet of things for smart cities, IEEE Internet of Things journal 1 (1) (2014) 22–32.

[15] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, M. Ayyash, Internet of things: A survey on enabling technologies, protocols, and applications, IEEE Communications Surveys & Tutorials 17 (4) (2015) 2347–2376.

[16] S. Žitnik, M. Janković, K. Petrovčič, M. Bajec, Architecture of standard-based, interoperable and extensible iot platform, in: Telecommunications Forum (TELFOR), 2016 24th, IEEE, 2016, pp. 1–4.

[17] C. Esposito, A. Castiglione, F. Pop, K.-K. R. Choo, Challenges of connecting edge and cloud computing: A security and forensic perspective, IEEE Cloud Computing 4 (2) (2017) 13–17.

[18] R.-I. Ciobanu, C. Negru, F. Pop, C. Dobre, C. X. Mavromoustakis, G. Mastorakis, Drop computing: Ad-hoc dynamic collaborative computing, Future Generation Computer Systems 92 (2019) 889–899.

[19] W. Shi, J. Cao, Q. Zhang, Y. Li, L. Xu, Edge computing: Vision and challenges, IEEE Internet of Things Journal 3 (5) (2016) 637–646.

[20] F. Bonomi, R. Milito, J. Zhu, S. Addepalli, Fog computing and its role in the internet of things, in: Proceedings of the first edition of the MCC workshop on Mobile cloud computing, ACM, 2012, pp. 13–16.

[21] U. Paščinski, J. Trnkoczy, V. Stankovski, M. Cigale, S. Gec, Qos-aware orchestration of network intensive software utilities within software defined data centres, Journal of Grid Computing 16 (1) (2018) 85–112. doi:10.1007/s10723-017-9415-1.
URL https://doi.org/10.1007/s10723-017-9415-1

[22] S. Nakamoto, Bitcoin: A peer-to-peer electronic cash system," http://bitcoin.org/bitcoin.pdf.

[23] V. Buterin, Ethereum white paper, updated september 30, 2015, https://github.com/ethereum/wiki/wiki/White-Paper, accessed: 2017-10-30.

[24] B. Carminati, E. Ferrari, C. Rondanini, Blockchain as a platform for secure inter-organizational business processes, 2018 IEEE 4th International Conference on Collaboration and Internet Computing (CIC) (2018) 122–129.

[25] F. Zhang, E. Cecchetti, K. Croman, A. Juels, E. Shi, Town crier: An authenticated data feed for smart contracts, Cryptology ePrint Archive, Report 2016/168, https://eprint.iacr.org/2016/168 (2016).

[26] A. Alrawais, A. Alhothaily, C. Hu, X. Cheng, Fog computing for the tnternet of things: Security and privacy issues, IEEE Internet Computing 21 (2) (2017) 34–42.

[27] K. K. Bimrah, H. Mouratidis, D. Preston, itrust: a trust-aware ontology for information systems development.

[28] F. Corradini, F. De Angelis, F. Ippoliti, F. Marcantoni, A survey of trust management models for cloud computing., in: CLOSER, 2015, pp. 155–162.

[29] J. H. Ziegeldorf, O. G. Morchon, K. Wehrle, Privacy in the internet of things: threats and challenges, Security and Communication Networks 7 (12) (2014) 2728–2742.

[30] P. Hu, H. Ning, T. Qiu, H. Song, Y. Wang, X. Yao, Security and privacy preservation scheme of face identification and resolution framework using fog computing in internet of things, IEEE Internet of Things Journal 4 (5) (2017) 1143–1155.

[31] P. Zhang, M. Zhou, G. Fortino, Security and trust issues in fog computing: A survey, Future Generation Computer Systems 88 (2018) 16–27.

[32] R. Roman, J. Lopez, M. Mambo, Mobile edge computing, fog et al.: A survey and analysis of security threats and challenges, Future Generation Computer Systems 78 (2018) 680–698.

[33] J. Ni, K. Zhang, X. Lin, X. S. Shen, Securing fog computing for internet of things applications: Challenges and solutions, IEEE Communications Surveys & Tutorials 20 (1) (2017) 601–628.

[34] S. M. Habib, S. Ries, M. Muhlhauser, Towards a trust management system for cloud computing, in: Trust, Security and Privacy in Computing and Communications (TrustCom), 2011 IEEE 10th International Conference on, IEEE, 2011, pp. 933–939.

[35] E. Mostajeran, M. F. Khalid, M. N. M. Mydin, B. I. Ismail, H. Ong, Multifaceted trust assessment framework for container based edge computing platform, in: Fifth International Conference On Advances in Computing, Control and Networking-ACCN 2016, 2016.

[36] S. K. Prajapati, S. Changder, A. Sarkar, Trust management model for cloud computing environment, arXiv preprint arXiv:1304.5313.

[37] L. Chen, J. Xu, Socially trusted collaborative edge computing in ultra dense networks, in: Proceedings of the Second ACM/IEEE Symposium on Edge Computing, ACM, 2017, p. 9.

[38] V. Sharma, I. You, D. N. K. Jayakody, M. Atiquzzaman, Cooperative trust relaying and privacy preservation via edge-crowdsourcing in social internet of things, Future Generation Computer Systems.

[39] W. Li, H. Song, Art: An attack-resistant trust management scheme for securing vehicular ad hoc networks, IEEE Transactions on Intelligent Transportation Systems 17 (4) (2016) 960–969.

[40] T. Wang, G. Zhang, M. Z. A. Bhuiyan, A. Liu, W. Jia, M. Xie, A novel trust mechanism based on fog computing in sensor–cloud system, Future Generation Computer Systems.

[41] R. Chen, J. Guo, F. Bao, Trust management for soa-based iot and its application to service composition, IEEE Transactions on Services Computing 9 (3) (2016) 482–495.

[42] S. A. Soleymani, A. H. Abdullah, M. Zareei, M. H. Anisi, C. Vargas-Rosales, M. K. Khan, S. Goudarzi, A secure trust model based on fuzzy logic in vehicular ad hoc networks with fog computing, IEEE Access 5 (2017) 15619–15629.

[43] D. Chen, G. Chang, D. Sun, J. Li, J. Jia, X. Wang, Trm-iot: A trust management model based on fuzzy reputation for internet of things.,

Comput. Sci. Inf. Syst. 8 (4) (2011) 1207–1228.

[44] F. Mora-Gimeno, H. Mora-Mora, D. Marcos-Jorquera, B. Volckaert, A secure multi-tier mobile edge computing model for data processing offloading based on degree of trust, Sensors 18 (10) (2018) 3211.

[45] R. Di Pietro, X. Salleras, M. Signorini, E. Waisbard, A blockchain-based trust system for the internet of things, in: Proceedings of the 23nd ACM on Symposium on Access Control Models and Technologies, ACM, 2018, pp. 77–83.

[46] N. Alexopoulos, S. M. Habib, M. Mühlhäuser, Towards secure distributed trust management on a global scale: An analytical approach for applying distributed ledgers for authorization in the iot, in: Proceedings of the 2018 Workshop on IoT Security and Privacy, ACM, 2018, pp. 49–54.

[47] B. Yu, J. Wright, S. Nepal, L. Zhu, J. Liu, R. Ranjan, Iotchain: Establishing trust in the internet of things ecosystem using blockchain, IEEE Cloud Computing 5 (2018) 12–23. doi:10.1109/MCC.2018.043221010.

[48] M. T. Hammi, B. Hammi, P. Bellot, A. Serhrouchni, Bubbles of trust: A decentralized blockchain-based authentication system for iot, Computers & Security 78 (2018) 126–142.

[49] P. Missier, S. Bajoudah, A. Capossele, A. Gaglione, M. Nati, Mind my value: a decentralized infrastructure for fair and trusted iot data trading, in: Proceedings of the Seventh International Conference on the Internet of Things, ACM, 2017, p. 15.

[50] K. M. Lundeen, S. Dong, N. Fredricks, M. Akula, J. Seo, V. R. Kamat, Optical marker-based end effector pose estimation for articulated excavators, Automation in Construction 65 (2016) 51–64.

[51] D. Liu, Y. Wu, S. Li, Y. Sun, A real-time monitoring system for lift-thickness control in highway construction, Automation in Construction 63 (2016) 27–36.

[52] A. Braun, S. Tuttas, A. Borrmann, U. Stilla, A concept for automated construction progress monitoring using bim-based geometric constraints and photogrammetric point clouds, Journal of Information Technology in Construction (ITcon) 20 (5) (2015) 68–79.

[53] W. Ren, Z. Wu, Real-time anticollision system for mobile cranes during lift operations, Journal of Computing in Civil Engineering 29 (6) (2014) 04014100.

[54] W. Yi, A. P. Chan, X. Wang, J. Wang, Development of an early-warning system for site work in hot and humid environments: A case study, Automation in construction 62 (2016) 101–113.

[55] J. Teizer, T. Cheng, Proximity hazard indicator for workers-on-foot near miss interactions with construction equipment and geo-referenced hazard areas, Automation in Construction 60 (2015) 58–73.

[56] M. Chiang, T. Zhang, Fog and iot: An overview of research opportunities, IEEE Internet of Things Journal 3 (6) (2016) 854–864.

[57] M. Wöhrer, U. Zdun, Design patterns for smart contracts in the ethereum ecosystem, 2018.

[58] P. Kochovski, P. D. Drobintsev, V. Stankovski, Formal quality of service assurances, ranking and verification of cloud deployment options with a probabilistic model checking method, Information and Software Technologydoi:https://doi.org/10.1016/j.infsof.2019.01.003.