Trust-as-a-Service: A reputation-enabled trust framework for 5G network resource provisioning

Jorquera Valero, José María ; Sánchez Sánchez, Pedro Miguel ; Gil Pérez, Manuel ; Huertas Celdrán, Alberto ; Martínez Pérez, Gregorio

DOI: https://doi.org/10.1016/j.comcom.2023.09.010

Computer Communications 211 (2023) 229–238

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

Trust-as-a-Service: A reputation-enabled trust framework for 5G network resource provisioning

José María Jorquera Valero [a,*], Pedro Miguel Sánchez Sánchez [a], Manuel Gil Pérez [a], Alberto Huertas Celdrán [b], Gregorio Martínez Pérez [a]

[a] Department of Information and Communications Engineering, University of Murcia, Murcia, 30100, Spain
[b] Communication Systems Group (CSG) at the Department of Informatics (IfI), University of Zurich UZH, Zürich, 8050, Switzerland

ARTICLE INFO

Keywords:
Trust models
Zero trust
Trustworthy relationships
Distributed marketplace
5G

ABSTRACT

Trust, security, and privacy are three of the major pillars to assemble the fifth-generation network and beyond. Despite such pillars are principally interconnected, a multitude of challenges arise that need to be addressed separately. 5G networks ought to offer flexible and pervasive computing capabilities across multiple domains according to user demands and assure trustworthy network providers. To this end, distributed marketplaces expect to boost the trading of heterogeneous resources so as to enable the establishment of pervasive service chains between cross-domains. Yet, the need for selecting reliable parties as "marketplace operators" plays a pivotal role in achieving a trustworthy ecosystem. Two of the principal blockages in managing foreseeable networks are the need to consider trust as a property in the resource provisioning process and adapt previous trust models to accomplish the new network and business requirements. In this regard, this article is centered on the trust management of 5G multi-party network resource provisioning. As a result, a reputation-based trust framework is proposed as a Trust-as-a-Service (TaaS) solution for a distributed multi-stakeholder environment where requirements such as zero trust and zero-touch principles should be met. Besides, a literature review is also conducted to recognize the network and business requirements currently envisaged. Finally, the validation of the proposed trust framework was performed in a real research environment, the 5GBarcelona testbed, leveraging 12% of a 2.1 GHz CPU with 20 cores and 2% of the 30 GiB memory. These outcomes reveal the TaaS solution's feasibility and conservative approach in the context of determining reliable network operators.

1. Introduction

Among all pillars of the communication evolution, fifth-generation (5G) technologies play a paramount role as a cutting-edge network paradigm, from radio technology and optical networks to non-terrestrial network communications and ubiquitous computing. Such paradigms, in turn, bring challenges to be overcome by next-generation technologies such as reduction of energy footprint, multi-tenancy, automated management and orchestration, on-demand service and resource allocation, and trustworthy infrastructures, among others [1,2].

In 5G and beyond (B5G), the distributed marketplaces encompass a potential solution commonly utilized by the vertical industry to provide end-to-end composite services or slices that allow satisfying all requirements and Key Performance Indicators (KPIs) in terms of coverage, networking and computing resources, and Virtual Network Functions (VNFs). Since such heterogeneous services and resources may be supplied by a single provider or multi-party collaboration across several domains, it is paramount to elect a trustworthy network provider, which ensures the fulfillment of requirements and KPIs, and guarantees a trustworthy environment [3]. In this regard, trust models facilitate reliable establishments among different stakeholders predicting a forthcoming trust score.

Nevertheless, trust models need to progress over time as novel network and business requirements are constantly appearing and prior trust models cannot cover them [4]. 5G and beyond are envisioned as compounded networks in which end-to-end communications will entail multiple entities from the same or different locations and domains. Thus, trust models ought to contemplate reliable end-to-end chains to predict future behaviors of all implicated entities from the origin to the end. In the same manner, implicit trust should not be granted to stakeholders, regardless of whether they are placed in an intra- or inter-domain scenario, as trust by default is a potential attack vector exploited by spiteful entities. In this sense, a zero trust approach, driven

* Corresponding author.
E-mail addresses: josemaria.jorquera@um.es (J.M. Jorquera Valero), pedromiguel.sanchez@um.es (P.M. Sánchez Sánchez), mgilperez@um.es (M. Gil Pérez), huertas@ifi.uzh.ch (A. Huertas Celdrán), gregorio@um.es (G. Martínez Pérez).

https://doi.org/10.1016/j.comcom.2023.09.010
Received 14 December 2022; Received in revised form 30 August 2023; Accepted 6 September 2023
Available online 9 September 2023

by the NIST [5], is a predominant principle for imminent trust models to dwindle the attack surface. Another fundamental requirement is the minimization of human interaction in the trust model lifecycle management, also known as zero-touch approach. Trust models should spur the automatization of network and service management via high-level policies, triggers, and artificial intelligence algorithms. Simultaneously, the automation process also entails an essential effort to enable easier integration with other 5G network orchestration and management components; for instance, a distributed marketplace allows verticals to expose telco digital assets and hire them to satisfy user demands automatically. Nevertheless, these requirements are currently not all addressed at the same time by most solutions in the literature [4].

Hence, the paper at hand is an extension of initial research published in [6], so the authors stand out the new efforts below. In particular, this article analyzes the present literature to determine whether the identified network and business requirements related to trust models are being contemplated. Besides, it also presents a reputation-based trust framework capable of guaranteeing a trustworthy ecosystem where stakeholders can establish reliable end-to-end connections across domains as well as deal with the aforementioned novel network requirements (e.g., zero trust, zero-touch, etc.). Therefore, such a framework considers a set of product offers (POs), available in the 5GZORRO European project distributed marketplace [7], to be thoroughly analyzed so as to help stakeholders to the trading of heterogeneous resources. Thereupon, an adapted PeerTrust model for peer-to-peer communities is leveraged to predict both a provider and product offer trust scores from historical interactions and recommendations, the latter published in a Data Lake platform to be consulted by interested stakeholders. This article also presents the formulation of new equations related to the PeerTrust model, such as offer and provider satisfactions. Lastly, the authors carry out several experiments in a real infrastructure testbed to examine the performance and accuracy of their reputation-based trust framework and feasible impacts on the whole network resource provisioning process.

The remainder of this article can be outlined as follows. Section 2 carries out in-depth research into the utmost importance trust models applied to on-demand service and resource provision environments. Section 3 describes the design of our reputation-based trust framework. Then, Section 4 presents the performance assessment results of the trust framework. Finally, Section 5 recaps some conclusions as well as ongoing works for future work.

2. Related work

In this section, we analyze the most newfangled approaches that explore trust models as a mechanism to provide reliable on-demand service or resource capabilities in 5G and B5G (see Table 1). To determine the compatibility level of analyzed approaches, we have compiled a set of universal network and business requirements which should be shared between 5G trust model solutions, such as trustworthy end-to-end chains across domains and the zero trust and zero-touch principles.

Trust remains a vital requirement in the cloud environment since reputable relationships between consumers and providers may guarantee the fulfillment of offered user's Quality of Service (QoS) as well as dwindling the chance to infringe Service Level Agreements (SLAs) or Smart Contracts (SCs) signed. In addition, the current cloud environments share paramount characteristics with the on-demand service and resource capability provisioning, in which our reputation-based trust framework is entailed. For these reasons, we have identified a set of investigations that not only contemplate trust models as a potential solution for cloud environments but also fit to some extent with the requirements described above.

Hassan et al. proposed in [8] a QoS-based model to dynamically assess cloud providers' trust before each new interaction. The authors composed the cloud resources' trust as a blend of the provider reputation from users' feedback plus the computing power at run-time from SLA attributes. To regulate dishonest feedback, the covariance technique was leveraged to calculate user credibility and discover misleading feedback. Nevertheless, their model did not consider requirements such as end-to-end chains (only analyzing the last entity of the chain) or the zero trust principle (main ideas are not supported through the paper). To test their enhanced QoS-based model, the authors contrasted their transaction success rate (TSR) against the Armor model [9], obtaining 0.92 and 0.74 TSRs, respectively, when there were 40% of fake users' feedback. Likewise, Guo et al. [10] introduced a novel trust model designed for cloud environments that relies on characteristic factors and SLAs. The proposed model includes a negotiation and monitoring mechanism that enhances the accuracy of service quality and cost assessments, as well as the detection of malicious actors. This approach effectively counters coordination, spoofing, and defamation attacks, resulting in a higher success rate of transactions. Furthermore, it promotes trust relationships among entities by using self-recommended trust and SLAs. Compared to the MDTES, TrueTrust, and CSRTM models, the proposed trust model demonstrates a 90% trade success rate (TSCR) against an 83%, 82%, and 84%, respectively, in identifying dishonest providers and withstanding various attacks from unscrupulous entities.

Concerning the business needs to scale its computing and infrastructure capabilities up, a new term called Federated Cloud appeared to enable the integration of public, community, and private clouds to support business requirements. Thereby, an inappropriate selection of deployment cloud platforms may encompass performance, security, and even legal issues. To cope with them, Verma et al. proposed in [11] a new secure and trusted Cloud Service Provisioning (CSP) scheme called FedRec. The scheme leveraged a Blockchain system and a novel Ranking-Based Recommender (RBR) model. The scheme operated in three phases: request–response broker model, weighted matching recommender model, and Smart Contract (SC) execution. Additionally, the authors fulfilled pivotal requirements such as automatization. The proposed scheme achieved a reduction of 27.55% in chain storage and a transaction throughput of 43.5074 Mbps at 150 blocks. The maximum hit ratio obtained in the RBR model is 0.9314 with an improvement of 1.2% in average servicing latency over non-RBR models. In the same line, Latif et al. presented in [12] a federated cloud trust management framework to ensure the fulfillment of privacy laws and the protection of customers' data. As a result, the authors addressed the issues of trust establishment and evaluation. Particularly, the framework was formed of three dimensions: SLA parameters focused on security and privacy, feedback from customers, and feedback from other clouds. Since the final reputation is composed of multiple entities involved in the relationship, it can be ensured trustworthy end-to-end chains. To test their framework, the authors contrasted their trust scores against other existing schemes and their SLAs such as IBM, Amazon, or Google, among others, reaching the second-best result in the vast majority of cloud providers. Nonetheless, they did not describe ideas aligned with the zero trust principle, so it cannot be guaranteed.

Also dealing with the trust in cloud environments, in [13], Khilar et al. centered on ascertaining both customers' trust prior to accessing the cloud and resources' trust. This approach was formed by multiple sub-modules among which the available resource catalog can be highlighted as the starting point. Like in [14] and in [12], customers and resources' behavior, feedback, and SLA parameters were contemplated to compute a trust score. As part of SLA parameters, the customer's satisfaction was formulated as the total number of successful tasks. Furthermore, the authors ensured zero trust, full automation of all steps, and an assessment of multiple entities involved in the end-to-end chain, not only the final target. In terms of performance, the k-Nearest Neighbors (kNN) got a 0.94 of precision, recall, and F1-Score as well as a 6.48% of mean absolute error (MAE), and 25.45% of Root MAE. Not only the selection of trustworthy Cloud Providers (CPs) is critical but

Table 1
Comparison of trust management models for on-demand service and resource provisioning solutions.

Solution	Year	Environment	Data Source	End-to-end	Zero-Touch	Zero Trust	Reward & Punishment
[13]	2019	Cloud	Resource fingerprinting, SLAs, and feedback	✓	✓	✓	✗
[15]	2019	Federated Cloud	Willingness, capability, and reputation	✗	✗	✗	✗
[16]	2019	Fog	Credibility and reputation	✓	✓	✗	✓
[8]	2020	Cloud	Feedback and SLA performance attributes	✗	✓	✗	✗
[12]	2021	Federated Cloud	SLA security and privacy parameters and feedback	✓	✗	✗	✗
[17]	2021	Edge	Interaction, energy, and recommendations	✓	✓	✗	✗
[18]	2021	Edge	Reliability, availability, and authenticity properties	✓	✓	✗	✗
[19]	2022	Fog	SLA performance parameters	✗	✓	✗	✗
[10]	2023	Cloud	Reliability, service quality, service cost, SLA, etc.	✓	✓	✗	✓
[11]	2023	Federated Cloud	SLA performance attributes	✗	✓	✓	✗
Ours	2023	Cloud	Reputation, breach prediction, and SLA violation rates, trust impact	✓	✓	✓	✓

also the possible co-tenants hosted in the cloud. In [15], Tahkur and Breslin proposed a reputation-based management mechanism utilized by CPs to distinguish users' behavior and properly assign resources based on trust scores. The CP reputation was formulated as the capability and willingness to differentiate between good and malicious users. Hence, a constant increase or decrease in users' reputation of a multi-tenancy group enhanced the CP reputation; otherwise, the CP was not able to create homogeneous groups and its reputation dwindled. In addition, the authors considered feasible rational, irrational, and opportunistic reports by the CP, achieving a higher reputation when the rational approach was met.

Another indispensable requirement to be fulfilled by 5G and beyond trust models is the establishment of a reliable end-to-end chain. In this vein, Wang et al. proposed in [17] a trust evaluation model for mobile edge nodes (TEM-MENs) to guarantee a reliable node chain between the trustor and the trustee, withstanding malicious attacks. Depending on the number of nodes, the authors declared an atomic trust chain (without intermediate nodes) or a combined trust chain. In the case of the atomic chain, they considered interaction trust, energy trust, and recommendation trust, along with time windows. With regard to the combined chain, they collected the previous values for each node forming the path(s), which covers the end-to-end trustworthy chain requirement. To automatically collect all information and fulfill the zero-touch philosophy, an enhanced Dijkstra algorithm was employed. By means of several experiments, the TEM-MENs demonstrated to have the highest detection rate (96%) and their runtime had a slope of 0.1, thereby allowing better adaptability. Similarly, Fan et al. also dealt with Service Function Chains (SFC) in [18]. In particular, they designed a credibility-based deployment strategy (CBDS) to prognosticate the trust of VNF nodes through the SFC credibility. The credibility was formed by reliability, availability, and authenticity properties. Additionally, the authors also added extra functionalities such as adopted sliding windows and trigger mechanisms, which empowered the mechanism as dynamic and fully automatic. In terms of experiments, the CBDS reached a 90% acceptance rate for a 0.75 credibility value (the highest one).

Last but not least, Debe et al. in [16] and Chang et al. in [19] addressed the problem of ensuring trust in the provision of compute and network capabilities for Fog Computing environments. In the former research, in [16], the authors presented a reputation-based solution to discover a trust score in a decentralized way. A blockchain, together with SCs, allowed for calculating the reputation of public fog nodes from past interactions. Besides, the credibility of client IoT devices, which was computed as a clustering of the most legitimate group of vectors contrasting the rating rate with the majority, was also contemplated to build the final trust of each public fog node. Similarly, reward and punishment mechanisms were defined to regulate the weight of client feedback and detect potential malicious clients. The latter research, in [19], the authors described a multi-dimensional trust model that helped clients determine the trustworthiness of Fog Service Providers (FSP). The model considers three viewpoints: the application itself, peers' trust, and the independent Fog auditor. The trustworthiness of FSP is calculated based on assigned weights that can be adjusted from the application requirements. In addition, the authors supported the zero trust principle by setting the peer trust to 0 for newcomers. Simulation results displayed that the proposed model can help users choose FSPs fairly since it showed the behavior of a consistent model, remaining above 0.9 trust score for most of the cases.

As observed, only [13] meets the three principal requirements: zero trust, zero-touch, and trustworthy end-to-end chains at the same time, which are paramount pillars to provide a reputation-based model compatible with future networks. Yet, this approach did not cover the same objective as ours because it was oriented to an access control cloud scenario and our objective is to enable a trustworthy resource and service provisioning discovery for distributed marketplaces. The proposed framework, therefore, aims to fill the gap in reputation-based trust models for 5G networks, as well as to ensure an automated, practical, and scalable framework.

3. Reputation-based trust framework design

This section describes the principal modules and characteristics of our Trust-as-a-Service solution as well as the most important pillars to compute trust scores. As shown in Fig. 1, the trust framework is principally composed of four sub-modules: the *Information gathering and sharing*, the *Trust computation*, the *Trust storage*, and the *Continuous update*. Next subsections thoroughly explain the utmost important steps under each module. By means of modules, we will contextualize how

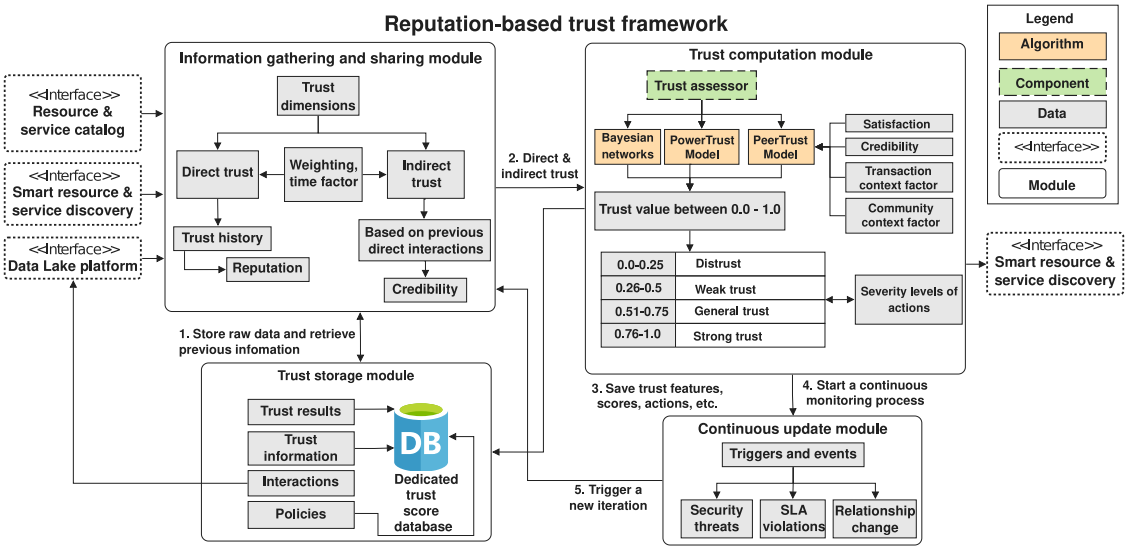


Fig. 1. Design of the proposed reputation-based trust framework.

zero trust, zero-touch, and reliable end-to-end chain principles can be addressed. Note that such a trust and reputation framework has been designed under the 5GZORRO project [20], and in consequence, a few concepts will be briefly introduced throughout the following subsections to contextualize and detail the decisions taken.

3.1. Information gathering and sharing module

First and foremost, the trust framework collects raw data from multiple available information sources such as the Data Lake and Resource and service catalog (see Fig. 1). To contextualize, the Data Lake is a centralized and shared data environment in its native format that leverages big data, and the Resource and service catalog refers to a decentralized repository employed for identifying and registering available resources and services. These information sources provision an extremely fruitful range of data for our trust and reputation framework. The former warehouses data like SLA breach predictions related to certain resources and service providers, statistic parameters with respect to the available product offers in the distributed marketplace platform, interactions between stakeholders, etc. The latter allows getting information related to the product offers (POs), e.g., the geolocation of services and resources, current life-cycle status, service specification, and statistical features. In this vein, our reputation-based trust framework can deal with network service POs which are composed of multiple sub-services from the same or different stakeholders. Thence, our framework does not analyze the PO as a whole but is capable of factorizing the PO by assessing each element of the chain and its provider separately so as to ensure end-to-end trustworthiness connections.

Depending on how the information is gathered, the TaaS considers it into direct and indirect trust. When it comes to direct trust, it is linked to personal experiences the trustor had after interacting with a trustee. In particular, our proposal is centered on the reputation deemed through the trust history of providers and their offers. Concerning indirect trust, it is conventionally collected from third party recommendations. Yet, recommendations do not always come from trustworthy third parties. The proposed trust framework integrates two mechanisms to avoid misleading recommendations as much as possible. Firstly, a feedback credibility mechanism is leveraged to figure out the recommender's honesty through the Personalized Similarity Metric (PSM) [21]. When it comes to a recommender, the concept refers to an entity or stakeholder who had an interaction with a given target and, in consequence, he/she is capable to help other entities/stakeholders by providing his/her experience (a trust score) about a target. Concretely, the PSM measures the credibility rate of another recommender through the personalized experience with him/her. On another hand, our TaaS also integrates a dynamic list of trustworthy recommenders, which is originated from previously computed trust scores. The list is continuously updated after each new interaction and it also contemplates the time factor to weigh up the most up-to-date scores as the most relevant. Lastly, it should be highlighted that the information gathering steps should be applied to each entity involved in the trust chain and not only to the extremes.

3.2. Trust computation module

Once all available trust information has been collected, such information is forwarded to the trust computation module to be processed. It is worth mentioning that this step is carried out regardless of whether stakeholders have a previous trust relationship for some time or whether they belong to the same domain, an intra-domain relationship, since the zero trust principle must be complied with. In spite of the reputation-based trust framework expects to support multiple trust assessors such as Bayesian networks and the PowerTrust model in the foreseeable future (see Fig. 1), this manuscript is principally centered on the PeerTrust model [21]. PeerTrust is based on interactions and designed for distributed environments in which *satisfaction* (S), feedback *credibility* (Cr), *transaction context factor* (TF), and *community context factor* (CF) are the pillars of building trustworthy establishments (see Eq. (1)).

$$T(u) = \alpha \cdot \left(\sum_{i=1}^{I(u)} S(u,i) \cdot Cr(p(u,i)) \cdot TF(u,i) \right) + \beta \cdot CF(u), \tag{1}$$

where u is the service or resource provider for whom wants to find out a trust score $T(u) \in [0,1]$ on the ith interaction; α and β are the weights of each dimension, satisfying that $\alpha + \beta = 1$; and $I(u)$ is the maximum number of interactions.

Nevertheless, Eq. (1) only introduces the theoretical concepts under the aforementioned pillars described in [21] and not how they should be formulated. Because of that, we decided to go for an adapted PeerTrust model as it brings a high flexibility level to mold the trust model to the final enforcement scenario as well as meet the distributed philosophy followed by the 5GZORRO marketplace platform, in which this framework is being utilized. In this vein, we have designed and developed several pillars to delineate the final trust scores. Yet, since the stakeholder's satisfaction is the utmost importance pillar and the most complex, we have put a special emphasis on it (see Fig. 2).

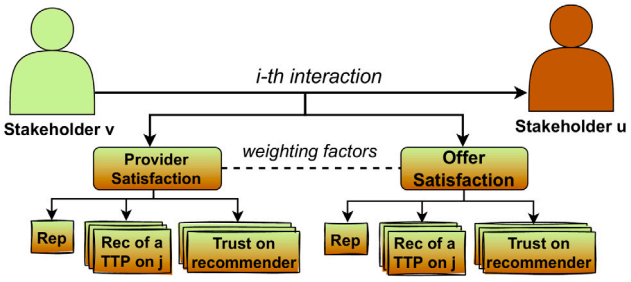


Fig. 2. Satisfaction composition.

Firstly, the *satisfaction* $S(u,i) \in [0,1]$ represented in Eq. (2) measures the acceptance degree that a stakeholder u has with another after finishing the *ith trustworthy interaction*. In this case, the *satisfaction* is related to both direct and indirect trust since this pillar considers personal information but also recommendations. Concretely, we considered two key dimensions to discover the final satisfaction value, the provider's satisfaction (PS) and the product offer's satisfaction (OS). Note that ψ and ϕ are the weights of each dimension and they must satisfy that $\psi + \phi = 1$, though ϕ tends to be higher than ψ.

$$S(u,i) = \psi \cdot PS(u,i) + \phi \cdot OS(u,i) \tag{2}$$

Concerning the provider's satisfaction $PS(u,i) \in [0,1]$, it is composed of three main features, as shown in Eq. (3): the reputation of a stakeholder j (*Rep*); a set of n recommendations (*Rec*) about the target j from a x trusted third party (TTP); and the last trust score for each recommender in the previous set. Thence, the satisfaction of stakeholder u on the ith interaction will be computed about the target stakeholder j.

$$PS(u,i) = Rep_u(j) \cdot \left(\bigoplus_{x=1}^{n} Rec(x,j) \cdot T_u^{(t-1)}(x) \right) \tag{3}$$

In this sense, the reputation $Rep_u(j) \in [0,1]$ represents the average reputation that the stakeholder u has on the stakeholder j contemplating all its available resources and services. This reputation function calculated below in Eq. (4) considers features such as available assets (AA), total assets (IA), available assets at a given location (AA_L), total assets at a given location (IA_L), the total number of predicted SLA violations (PV) that were lastly managed both successful (MV) and unsuccessful (EV), and no-predicted SLA violations (NPV). In addition, multiple time windows are also deemed together with weighting factors to cater for higher relevance to the newest scores.

$$Rep_u(j) = \sum_{k=1}^{n} \varepsilon(k) \cdot \frac{\left(\frac{AA(j)}{IA(j)} + \frac{AA_L(j)}{IA_L(j)} + 2 \cdot \frac{MV(j)}{PV(j)} - 2 \cdot \frac{EV(j)+NPV(j)}{PV(j)} \right) + 2}{6} \tag{4}$$

When it comes to the provider's satisfaction of Eq. (3), we additionally leverage an aggregation operation to combine recommendations (*Rec*) about the stakeholder j with the last trust score provided by the recommender x. Once the computation of a provider's satisfaction has been completed, the next step is to reckon the product offer's satisfaction $OS(u,i) \in [0,1]$. To deal with it, the reputation-based trust framework utilizes similar dimensions (see Fig. 2) but only takes into account information about a particular product offer since the provider's satisfaction considers all available assets. In the 5GZORRO ecosystem, there are seven types of product offers defined in the Resource and service catalog depicted in Fig. 1: radio access network (RAN), spectrum, VNF/container network function (CNF), network service, slice, cloud, and edge. Hence, the product offer's satisfaction is formed by reputation (*Rep*), recommendations (*Rec*), and last trust scores of a specific type of offer and provider.

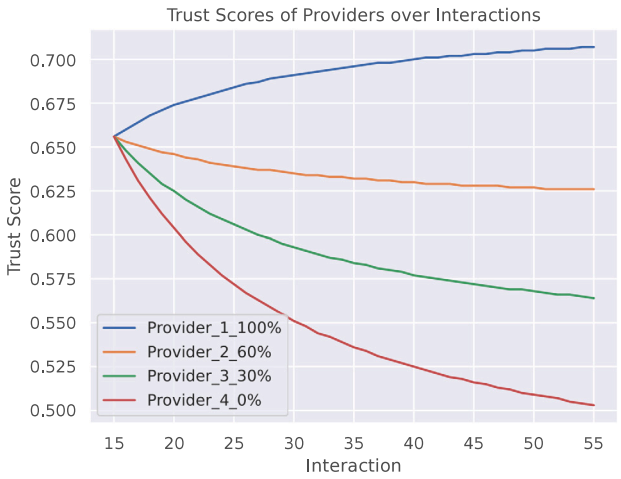


Fig. 3. Impact of publishing interactions on a trust score.

With respect to the second pillar of the PeerTrust model, the feedback *credibility*, our approach follows the PSM metric [21] as it may be applied to multiple contexts. In particular, it determines how similar two unfamiliar stakeholders are when evaluating a set of targets ($p(u,i)$). Thereby, the personalized similarity is the metric leveraged to contrast the opinions about a target as well as measure the distance of credibility about a set of targets assessed by both stakeholders. Hence, the higher credibility distance after evaluating the same set of targets, the less credible the opinion. In the case of feedback credibility, our PeerTrust model contemplates both direct and indirect trust as it compares our opinion with that of a third party. Additionally, our reputation-based trust framework introduces two context factors that allow adjusting the final trust score to the current transaction type and the community (see Fig. 1). Firstly, the *transaction context factor* pillar intends to forecast a trust value linked to the current interaction, with a particular stakeholder or product offer, from the number of feedbacks published in the Data Lake from different time windows. In this case, the transaction factor is uniquely associated with direct trust because it only looks at historical information. The transaction context factor rewards stakeholders that publish their interactions with others in the Data Lake since it spurs future stakeholders to look into the Data Lake, request recommendations to other stakeholders, and grow the community. As can be observed in Fig. 3, there is a logarithmic increase or decrease when a stakeholder decides to publish more or fewer encrypted interactions in the Data Lake. To perform this simulation, we assumed that four providers had the same behaviors until interaction 15, which means publishing a 75% of interactions and having a 0.656 trust score. At this point, Provider_1 decided to publish 100% new interactions, Provider_2 reduced the publication of new interactions until 60%, Provider_3 until 30%, and Provider_4 until 0%. Note that the legend of Fig. 3 shows both the provider ID and the percentage of new interactions published. These new decisions were maintained for the next 40 interactions to showcase how their decisions would affect the final trust scores. Due to the impact of transaction factor on the final trust score, which is ~0.165 for $\alpha = 0,5$, we can visualize how the trust score is increased by 7.77% for Provider_1 and decreased by 4.57% for Provider_2, 14,02% for Provider_3, and 23,32% for Provider_4. Therefore, the lack of publishing encrypted interactions in the Data Lake, to enable future stakeholders to look into the Data Lake and request recommendations, has a critical impact on trust scores since it goes against the principles of the PeerTrust model.

To preserve privacy of the transaction factor, our reputation-based trust framework follows multiple approaches. To begin with, stakeholders only publish information about the parties involved and the interaction date, so a trust score is not publicly available but it should

be expressly requested from the stakeholder indicated on the basis of public information. It should be pointed out that such information is encrypted in the Data Lake by leveraging Homomorphic Encryption (HE) to allow computations to be performed directly on encrypted data without requiring decryption as well as to be stored in encrypted form. Secondly, the framework makes use of a query-based encryption approach to enable a specific stakeholder to find other stakeholders to request recommendations while preserving data and user privacy. Since this article is contextualized in the 5GZORRO ecosystem, it takes advantage of other entities designed and developed in 5GZORRO, such as the Governance Manager [7], who is considered a Trusted Regulatory that emits both identifiers of stakeholders, and public/private key pairs. Thence, the Governance Manager generates key pairs both for stakeholders and the Data Lake, which are utilized to share information. Additionally, the Data Lake also has an access control mechanism, so a stakeholder can only ask for recommendations on those candidates that are available in the Catalog, based on your current needs. Otherwise, stakeholders' requests will be detected as unauthorized access attempts and will be blocked from accessing the Data Lake.

Finally, the *community context factor* pillar attempts to gather multiple feedback about a target stakeholder from a dynamic list of trustworthy recommenders (indirect trust). Hence, the community context factor measures the number of interactions that a specific stakeholder had in the community through the contribution of services or resources with other stakeholders. Besides, multiple recommendations, together with the credibility of the recommender, are contemplated through an aggregation function to achieve the general reputation of the community about a target stakeholder.

In the end, the weighting of the credibility, satisfaction, and transaction context factor plus the community context factor enables to determine a final trust score of a target stakeholder by contemplating multiple interactions and reputations from different recommenders and time windows.

3.3. Trust storage module

After computing a trust score, the next step is to save both the raw and inferred data for future establishments and recommendations. To cope with it, the proposed framework makes use of two types of information storage sources (see Fig. 1). Because the TaaS is instantiated per domain, a private non-relational database has been contemplated per instance, which is part of our reputation-based trust framework. The dedicated database mainly stores information regarding raw data collected from information sources such as the Data Lake and Resource and service catalog, the adapted PeerTrust model information, and lastly, the trust scores. The main idea of storing raw data collected from the Data Lake and the Catalog is to dwindle the time necessary to process such information in real time. In this way, our reputation-based trust framework is capable of updating the inferred data through a service running in the background, without the compute module having to wait for the statistics to be calculated (except the first iteration). Furthermore, it may also save internal policies or rules to be used by the continuous update module so as to trigger events or detect misbehaviors. Thus, sensitive information will be stored in the dedicated database as it is not going to be shared with other stakeholders. In addition to the private database, the framework also leverages a Data Lake which is an external component deployed in the 5GZORRO project and managed by a Governance Manager [7], but it is consumed by our framework. In particular, the capital aim is to spread knowledge about trustworthy interactions among stakeholders that form the 5GZORRO ecosystem. In this regard, newcomers may recognize feasible recommenders to be consulted. Similarly, other stakeholders that already established previous trust interactions across domains may request new recommendations when their trustworthy recommenders are not able to support feedback. It is worth mentioning that the information stored in the Data Lake is encrypted via Homomorphic Encryption, so it preserves data and user privacy. In contrast to the private database, the Data Lake should not store sensitive information since its objective is to be consulted by countless stakeholders. However, the Data Lake introduces key features such as long-term reputation reflection and traceability. Last but not least, it should be pointed out that both the Data Lake and the private database act as sources of information storage, however, only the private database is a pure component belonging to the reputation-based trust framework.

3.4. Continuous update module

Parallel to the trust storage module, and once a trust score has been concluded, the trust computation module triggers a continuous update process focused on the target stakeholder. This module plays a pivotal role as it may enable earlier identification of plausible attacks through a suitable configuration of triggers, events, and rules. In this sense, the continuous update module not only ameliorates the security capabilities of the framework via context-dependence and dynamicity but also empowers an end-to-end automatization. Therefore, should unfamiliar phenomena appear in ongoing trust relationships, the reputation-based trust framework can identify them and take the proper decisions.

With regard to real-time events in an established trust relationship, the continuous update module presents reward and punishment mechanisms to oversee the stakeholders' behaviors as well as grant zero trust after establishing a relationship. Hence, a trust score is recalculated after appearing new events. Note that the main difference between the trust computation module and the continuous update module is that the former only computes an initial trust score, which is expressly requested. Yet, the latter starts once a trust relationship is established and runs cyclically based on events generated in real time. Among the principal events contemplated to increase or dwindle trust scores, we consider security threats, change policy relationships, SLA violations, breach predictions, or breach detections, to name but a few. Therefore, whether negative events occur, the previous trust scores will be diminished by applying the proper internal policies, and in consequence, it could be finished to discover more reliable stakeholders. On the contrary, favorable events entail an increase in the previous trust score. Note that the reward and punishment mechanisms also support the zero-touch principle as it allows full automation of the proposed framework via the aforementioned event-driven or time-driven mechanisms. Thence, human interaction is not required to adjust trust scores in ongoing trust relationships and the reputation-bases trust framework can smoothly interact with other modules participating in the resource and service provisioning discovery, as can be observed in Fig. 1. One example of a reward and punishment mechanism, together with its functioning and its main equation, was initially described at a high level in [6] by the authors. Such a mechanism was focused on gathering real-time security events from a network monitoring tool (Zeek) and a virtual Test Access Point (vTAP). In particular, Zeek was configured with several rules and policies available to detect and alert security incidents that may compromise the confidentiality, integrity, and availability of resources and services. Based on the initial equation presented in [6] (see Eq. (5)), this manuscript goes a step further by describing how the updates are done, what formulas are used to update a trust score by using a security-based reward and punishment (RP) mechanism and what events contribute to it.

$$RP(v,u) = \alpha \cdot Conn(v,u) + \beta \cdot Notice(v,u) + \psi \cdot Weird(v,u) \tag{5}$$

where v is the consumer who wants to update the trust score of a provider u. When it comes to $Conn(v,u)$, it gathers the tracking/logging of general information regarding TCP, UDP, and ICMP traffic between the provider u and the consumer v. By analyzing TCP, UDP, and ICMP packets reported in the *conn.log* file generated by Zeek, we may identify a high packet loss across TCP, UDP, or ICMP connections, which indicates network connectivity issues, problems with network infrastructure, or an exceeded network bandwidth that leads packet drops

and potential performance issues. In order to update trust scores in real time, any disturbance in the network traffic, e.g., non-compliance with the QoS a stakeholder previously indicated, affects the trust value. It should be pointed out that Zeek does not send the whole traffic log to our reward and punishment mechanism but only the final values to minimize privacy issues and optimize communication bandwidth. Therefore, if our framework receives notifications from Zeek about disturbances in TCP, UDP, or ICM connections, the reward and punishment mechanism will determine the degradation of the current situation based on Eq. (6).

$$Conn(v,u) = \rho \cdot \frac{TCP_{resp\_pkts}}{TCP_{orig\_pkts}} + \mu \cdot \frac{UDP_{resp\_pkts}}{UDP_{orig\_pkts}} + \omega \cdot \frac{ICMP_{resp\_pkts}}{ICMP_{orig\_pkts}}, \quad (6)$$

where ρ, μ, ω depict weighting factors, satisfying that $\rho+\mu+\omega = 1$, and $TCP/UDP/ICMP_{orig\_pkts}$ and $TCP/UDP/ICMP_{resp\_pkts}$ are the packets sent and received between two hosts. Another pivotal dimension to determine a reward or punishment in trust scores is the $Notice(v,u)$ (see Eq. (7)). Through *notice.log* file, Zeek may monitor and notify events that are odd or potentially bad such as *Software::Vulnerable_Version*, *Scan::Port_Scan*, or *HTTP::SQL_Injection_Attacker*, to name but a few. Although the setup of events to be monitored is not an action of our reputation-based trust framework, the occurrence of these events in the Zeek logs is an indicator that unusual behaviors are arising at a particular time. Therefore, a trust score should be updated based on the occurrence or absence of such events. In our case, we consider any event configured in Zeek to be of equal importance since our framework is not interested in the criticality of events but in the occurrence of unusual behavior.

$$Notice(v,u) = 1 - \frac{\left(\frac{N\_actual\_events}{N\_actual\_events+N\_events_{n-1}}\right)}{2} - \frac{\left(\frac{N\_actual\_events}{N\_actual\_events+\frac{N\_events_{n-5}}{5}}\right)}{2}, \quad (7)$$

where N_actual_events, N_events_{n-1}, and N_events_{n-5} represent the figure of odd events detected in the current time window, the last time window, or the last five time windows respectively, considering a time window of 30 min. Our security-based reward and punishment mechanism also contemplates the logs generated by *weird.log* file under the $Weird(v,u)$ dimension. In particular, Eq. (8) measures a default set of actions to take as unusual or exceptional activities that may reflect misconfigured hardware, malfunctioning, or malformed connections, or even an attacker trying to confuse a sensor. Some events identified by $Weird(v,u)$ are *DNS_UNMTATCHED_REPLY*, *ACTIVE_CONNECTION_REUSE*, or *POSSIBLE_SPLIT_ROU TING*, among others.

$$Weird(v,u) = 1 - \frac{\left(\frac{F\_actual\_weird\_events}{F\_actual\_weird\_events+F\_events_{n-1}}\right)}{2} - \frac{\left(\frac{F\_actual\_weird\_events}{F\_actual\_weird\_events+\frac{F\_weird\_events_{n-5}}{5}}\right)}{2}, \quad (8)$$

where N_actual_events, N_events_{n-1}, and N_events_{n-5} represent the number of unusual activities detected in the current time window, the last time window, or the last five time windows respectively, also considering a time window of 30 min. Lastly, once our continuous update module has determined a reward value ($RP(v,u) \geq 0.5$) or a penalty value ($RP(v,u) < 0.5$), the trust score previously estimated (denoted as $O_{ts}(v,u)$) through the Eq. (9) will be updated ($N_{ts}(v,u)$). It should be pointed out that whether the RP value is closer to the extremes (i.e., 1.0 or 0.0), the resulting increase or decrease will be more significant compared to when the RP value is approximately 0.5.

$$N_{ts}(v,u) = \begin{cases} O_{ts}(v,u) + (RP(v,u) - 0.5) \cdot \frac{\left(1-O_{ts}(v,u)\right)}{10}, & \text{if } RP(v,u) \geq 0.5 \\ O_{ts}(v,u) - (0.5 - RP(v,u)) \cdot \frac{\left(1-O_{ts}(v,u)\right)}{10}, & \text{if } RP(v,u) < 0.5 \end{cases} \quad (9)$$

As we previously mentioned, reward and punishment mechanisms may consider critical events appearing in real time such as security threats, change relationships, SLA violations, breach predictions, or breach detections. In this vein, the aforementioned security-based reward and punishment mechanism is one of the approaches the authors attempt to define in their reputation-based trust framework. Yet, other reward and punishment mechanisms, e.g., an SLA-driven approach, could also be considered to supplement the Continuous update module.

4. Experiments and results

This section introduces the principal characteristics of the 5GBarcelona testbed in terms of CPU and memory. Additionally, we showcase multiple experiments in which the CPU and memory consumption of our reputation-based trust framework can be observed for different amounts of product offers. Similarly, we can also analyze the required time to run each module of the framework as well as the possible impact on the whole network resource provision discovery. Lastly, we conduct user studies to evaluate how users perceive the reputation-based trust framework and whether it meets their expectations.

4.1. Experimental setting

On-demand resource and service allocation to cover users' requirements is a real challenge in 5G and B5G networks [22]. Therefore, the 5GZORRO project introduces an innovative solution through a distributed marketplace platform, which enables secure and trustworthy provisioning of resource and service capabilities. In addition to the marketplace, there is a component named Smart Resource and Service Discovery (SRSD) that enforces zero-touch and automatization requirements and allows obtaining a set of resources and services through an intent-based discovery. The SRSD and the trust framework have been integrated to ensure TaaS and elect the final provider based on its trust scores together with other intents such as type of resource, location, etc. By means of such an integration, it is possible to expose reliable telco digital assets, hire them, and enable a zero-touch interaction with other network orchestration and management components.

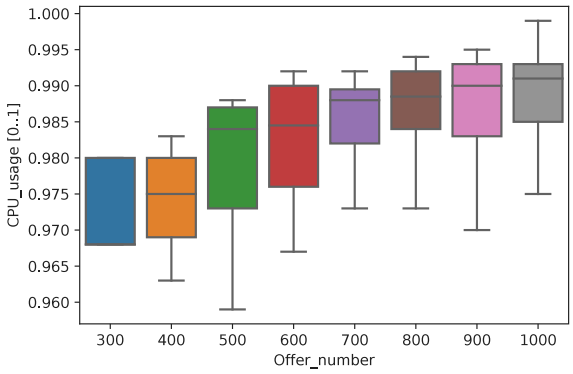
In terms of testbeds, the reputation-based trust framework has been deployed in the 5GBarcelona infrastructure where the framework was instantiated in a 1vCPU of a 6th Intel Xeon Gold 5218R with 2.1 GHz and 20 cores. Particularly, the framework was deployed in a worker with 8 cores and 30 GiB.

4.2. Performance evaluation

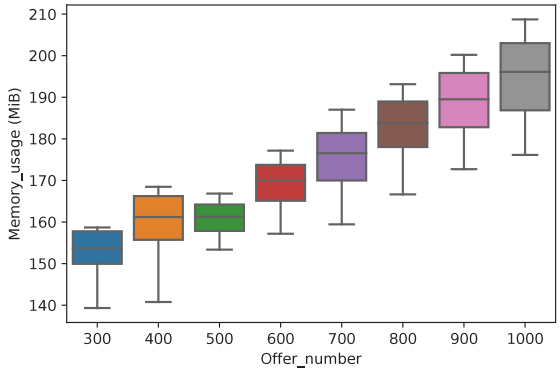
To check the proper functionality of the proposed framework and determine the resource and time consumption, we performed several experiments.

Firstly, we analyzed the CPU and memory consumption required by our framework. Fig. 4 plots a light growth when the number of offers to be processed increases. In particular, the average CPU consumption of the 1vCPU (only 1 core) allocated to the framework lies between 0.975 and 0.993 (see Fig. 4(a)), except for the case of 100 and 200 offers in which are 0.477 and 0.483 respectively. In spite of our vCPU was being used almost 99%, we only consumed 12% of the total CPU available in the server. In the case of memory, the framework required around 155 and 195 MiB (only 2% of the total memory) to process the multiple sets of product offers (see Fig. 4(b)), except for 100 and 200 offers in which the average was set to 140 and 152 MiB. The values related to 100 and 200 offers were not plotted since they might complicate the box plot visualization and the number of CPU and memory measurements was lower than the rest of offers because the framework computed trust scores quicker.

When it comes to consuming time, the cold-start mechanism doubled the required time to provide trust scores for each set of offers,



(a) Average CPU consumption per offer



(b) Average memory consumption per offer

Fig. 4. Reputation-based trust framework consumption.

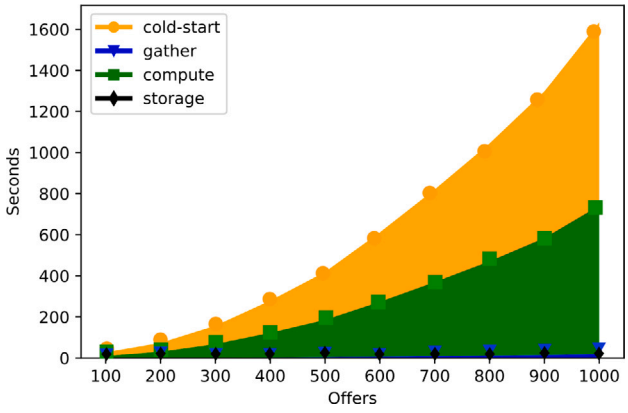


Fig. 5. Reputation-based trust framework time consumption with cold-start.

as depicted in Fig. 5. Nevertheless, such a cold-start mechanism was leveraged because of 5GZORRO ecosystem is not fully instantiated and there is not enough information about trust relationships between stakeholders. In the foreseeable future, the cold-start mechanism will be eradicated, and consequently, its time consumption. In this regard, the proposed framework was able to perform the information gathering, computation, and storage phases in 10.4, 185.6, and 741.4 s for 100, 500, and 1000 offers, respectively, in a sequential way.

Since the computation phase entailed a higher time, we decided to analyze it in detail. In Fig. 6, we can observe a slight increment in which the credibility consumed more than half of the total time to carry out the computation phase. Concretely, the proposed framework

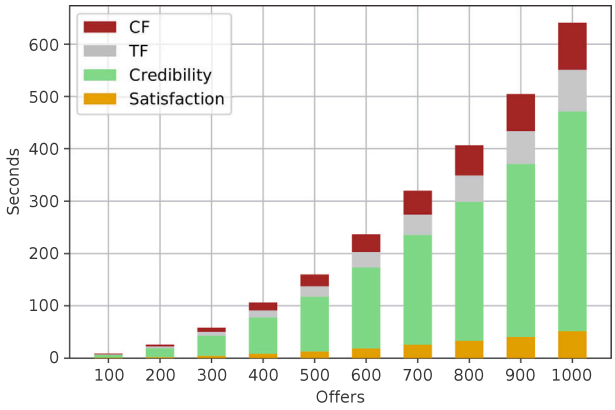


Fig. 6. Time distribution by computation phase.

needed 5.6 out of 10.4 s, 104.9 out of 185.6, and 419.9 out of 741.4 for 100, 500, and 1000 offers. As aforementioned, the computation phase has been designed to perform assessments sequentially, therefore, we believe such a decision may be the key fact why the framework needs twice as much time for the calculation phase. Hence, a feasible approach to address this drawback would be to apply parallel to the computation phase as the gathering and storage phases do not require executing a huge number of tasks. In consequence, they do not entail a detriment in the time consumption of the framework as it can be observed in Fig. 5. Due to the fact that the leveraged vCPU possesses multiple cores and we are only utilizing 1 core, an initial idea will be to leverage 2 cores to enable parallel processing. In particular, the authors bear in mind using the single instruction multiple data (SIMD) parallel processing type [23]. Such a technique allows a computer with two or more processors to follow the same instruction set but handle distinct data types. Therefore, we could ameliorate speed, reduce power consumption, and perform more efficient management of several trust score computation activities.

From previous experiments, it can be concluded that the reputation-based trust framework does not require a huge amount of CPU and RAM to request and analyze the information as well as compute trust scores for a high number of offers, for instance, 500 POs. However, the necessary time to calculate trust scores should be decreased in next iterations since 180 s were consumed to evaluate the trust of 500 POs. Despite that, the authors consider the proposed framework meets with the expected behavior and performance as other filtering mechanisms, such as intent-based or hardware requirements, are applied before computing trust scores. Therefore, the PO number to be analyzed by the proposed framework is normally ranked between 200 and 500.

4.3. Evaluating the accuracy of the trust and reputation framework: A user study

This experiment aims to analyze and understand whether the reput ation-based trust framework provides useful information to the users while assessing different providers in a distributed marketplace. To this end, we will compare the gap between the trust score provided by our framework and the trustworthiness perceived by users considering only their previous interactions.

First and foremost, 50 random users (consumers) have been recruited with knowledge of reputation-based trust frameworks to rate the trustworthiness of 50 random providers, having approval from an ethics board for running this user study. It should be pointed out that we assume the users have reliable behaviors (one-to-one analysis), they previously had trust relationships with providers, and they do not pretend to perform any type of attack since the objective of this experiment is not to test the resilience. Once we have collected users,

the next step is to declare the criterion to be followed by users to estimate users' trust ratings. Due to the fact that our reputation-based trust framework follows an adapted PeerTrust Model, which is mainly based on *satisfaction*, feedback *credibility*, *transaction context factor*, and *community context factor*, we have selected as the criterion for users the historical interactions. The history of interactions between users and providers is an important factor in determining trustworthiness since providers with a high number of successful transactions or interactions and few disputes can be considered more trustworthy. Besides, the historical interactions do not consider recommendations for third parties, as the adapted PeerTrust model works. In this regard, the study attempts to find out the users' estimations, before they know new trust scores determined by the reputation-based trust framework, using the standard deviation of historical relationships, i.e., historical trust values, that a user had with a specific provider. The reason for selecting the standard deviation as a metric to determine users' estimations is that users try to perform a time series analysis looking at historical trust values, and the standard deviation provides a more intuitive and easily interpretable measure of dispersion. Afterward, we also leverage a statistical method like correlation analysis to determine the relationship between our reputation scores and the perceived trustworthiness of users. In addition, we also carry out a qualitative analysis to discover any issues or limitations with the framework based on the feedback provided by users.

As a result, Fig. 7 displays the discrepancies between the scores that our reputation-based trust framework calculated, based on the adapted PeerTrust model, and the estimated trust scores of the users, using the standard deviation of historical interactions, on the y-axis. Furthermore, the bars represent the trust scores that our reputation-based trust framework determined via the adapted PeerTrust model. Thence, we can discover users' trust estimations by adding the discrepancy to the bar scores. Note that both framework trust scores and users' trust estimations are bounded between 0 and 1 because this is the range also leveraged by our adapted PeerTrust model.

Looking at Fig. 7, we can observe how the discrepancies between both scores are less than 4% for 86% cases. Especially, only 16% samples had a value above or below the trust scores determined by our framework. We interpret the previous percentage as a positive result of this study because the amount of information beard in mind by users was limited in comparison with the one handled by the adapted PeerTrust, so users cannot be as much as accurate as our statistical model that considers several dimensions and feedback from trustworthy recommenders. On another hand, Fig. 7 also points out a conservative approach when our framework computes the trust scores, as only 8% of cases the framework trust score is higher than the user trust rating. Following a conservative trust assessment, our reputation-based trust framework introduces a beneficial feature in distributed marketplaces. By providing lower trust scores, the framework may encourage users to be more cautious and diligent when interacting with providers, potentially reducing the likelihood of negative interactions and increasing overall network security.

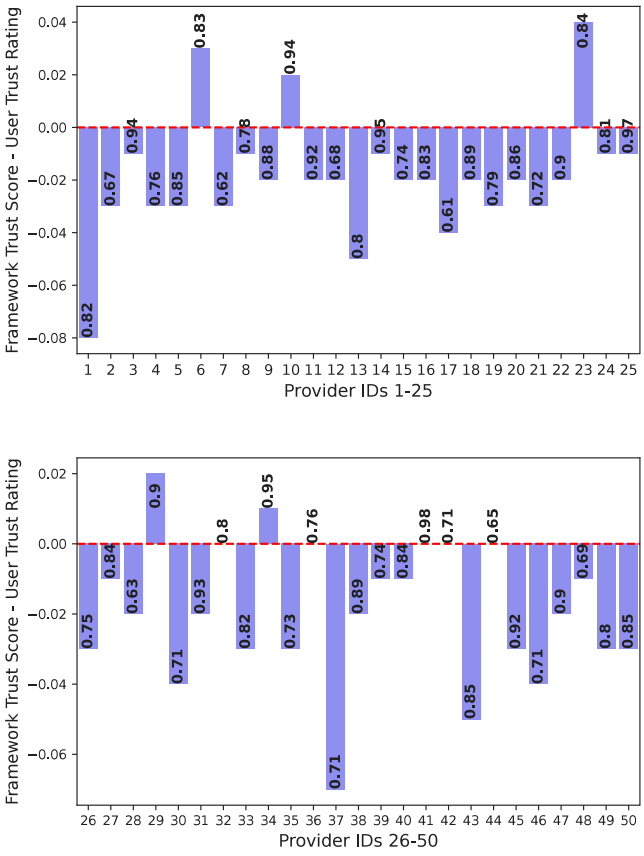


Fig. 7. Gap between Framework Trust Score and User Trust Rating.

5. Conclusion and future work

The article at hand analyzes some of the most prominent trust and reputation approaches in the research field to identify the utilization of network and business requirements. Taking into account previous investigations, we propose a reputation-based trust framework capable of helping stakeholders in the decision of electing the most reliable providers of resource or service capabilities available in a distributed marketplace. At the same time, the proposed framework takes into account critical requirements of 5G multi-party networks as aforementioned.

In terms of computation, an adapted PeerTrust model has been selected as the most befitting algorithm to foresee end-to-end trust scores based on historical interactions and recommendations. Multiple experiments were carried out in the 5GBarcelona testbed. The outputs showed that the trust and reputation framework had a slight increase in time consumption when the number of offers increased. However, this increase may be reduced by extending the number of used vCPU cores and applying parallelism techniques during the computation phase. On another hand, the reputation-based trust framework has successfully proven that its trust scores provided to users are accurate and reliable, via the user study conducted, and it follows a conservative trust assessment approach.

As future work, we will extend the current functionalities of the framework to contemplate prominent algorithms such as Bayesian networks and the PowerTrust model, and contrast them with PeerTrust to analyze their performances and accuracies. In this sense, the TaaS will also be deployed in the 5TONIC testbed, from Telefonica, to contrast metrics. Besides, additional functionality to cope with potential trust attacks and provide an adversary model for the TaaS such as collusion, Sybil, and bad-mouthing, among others, should be designed and developed after looking at the ones that our trust framework may suffer. We will analyze feasible parallelism techniques like the SIMD parallelism technique, so the reputation-based trust framework time consumption would be reduced and would not have an impact on the whole network resource provisioning process. Last but not least, we will supplement the continuous update module with an SLA-driven approach to measure SLA violation, breach predictions, and breach detections generated during an ongoing trust relationship, as well as an evaluation of how much the SLA-driven updates may change an initial score.

CRediT authorship contribution statement

José María Jorquera Valero: Term, Conceptualization, Methodology, Formal analysis, Investigation, Resources, Writing – original

draft, Writing – review & editing, Visualization. **Pedro Miguel Sánchez Sánchez:** Conceptualization, Validation, Investigation, Visualization. **Manuel Gil Pérez:** Conceptualization, Validation, Formal analysis, Writing – review & editing, Visualization, Supervision. **Alberto Huertas Celdrán:** Conceptualization, Validation, Visualization, Supervision. **Gregorio Martínez Pérez:** Conceptualization, Validation, Writing – review & editing, Visualization, Supervision, Project administration.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This work has been supported by the European Commission through the 5GZORRO project (grant no. 871533) part of the 5G PPP in Horizon 2020.

References

[1] M.M. da Silva, J. Guerreiro, On the 5G and beyond, Appl. Sci. 10 (20) (2020) 7091.
[2] M. Ylianttila, et al., 6G white paper: Research challenges for trust, security and privacy, 2020, arXiv preprint arXiv:2004.11665.
[3] R. Kantola, Trust networking for beyond 5G and 6G, in: 2020 2nd 6G Wireless Summit, 6G SUMMIT, 2020, pp. 1–6.
[4] J.M. Jorquera Valero, P.M. Sánchez Sánchez, M. Gil Pérez, A. Huertas Celdrán, G. Martínez Pérez, Toward pre-standardization of reputation-based trust models beyond 5G, Comput. Stand. Interfaces 81 (2022) 103596.
[5] S.W. Rose, O. Borchert, S. Mitchell, S. Connelly, Zero trust architecture, 2020, NIST Special Publication 800-207.
[6] J.M. Jorquera Valero, M. Gil Pérez, G. Martínez Pérez, A security and trust framework for decentralized 5G marketplaces, in: VII National Cybersecurity Research Conference, JNIC, 2022, pp. 237–240.
[7] A. Fernández-Fernández, et al., Multi-party collaboration in 5G networks via DLT-enabled marketplaces: A pragmatic approach, in: 2021 Joint European Conference on Networks and Communications & 6G Summit, EuCNC/6G Summit, 2021, pp. 550–555.
[8] H. Hassan, A.I. El-Desouky, A. Ibrahim, E.-S.M. El-Kenawy, R. Arnous, Enhanced QoS-based model for trust assessment in cloud computing environment, IEEE Access 8 (2020) 43752–43763.
[9] T.H. Noor, Q.Z. Sheng, L. Yao, S. Dustdar, A.H. Ngu, CloudArmor: Supporting reputation-based trust management for cloud services, IEEE Trans. Parallel Distrib. Syst. 27 (2) (2015) 367–380.
[10] L. Guo, H. Yang, K. Luan, L. sun, Y. Luo, L. Sun, A trust model based on characteristic factors and SLAs for cloud environments, IEEE Trans. Netw. Serv. Manag. (2023).
[11] A. Verma, P. Bhattacharya, U. Bodkhe, D. Saraswat, S. Tanwar, K. Dev, FedRec: Trusted rank-based recommender scheme for service provisioning in federated cloud environment, Digit. Commun. Netw. 9 (1) (2023) 33–46.
[12] R. Latif, S.H. Afzaal, S. Latif, A novel cloud management framework for trust establishment and evaluation in a federated cloud environment, J. Supercomput. 77 (2021) 12537–12560.
[13] P.M. Khilar, V. Chaudhari, R.R. Swain, Trust-based access control in cloud computing using machine learning, in: Cloud Computing for Geospatial Big Data Analytics, Springer, Cham, 2019, pp. 55–79.
[14] K. Papadakis-Vlachopapadopoulos, R.S. González, I. Dimolitsas, D. Dechouniotis, A.J. Ferrer, S. Papavassiliou, Collaborative SLA and reputation-based trust management in cloud federations, Future Gener. Comput. Syst. 100 (2019) 498–512.
[15] S. Thakur, J.G. Breslin, A robust reputation management mechanism in the federated cloud, IEEE Trans. Cloud Comput. 7 (3) (2019) 625–637.
[16] M. Debe, K. Salah, M. Rehman, D. Svetinovic, Towards a blockchain-based decentralized reputation system for public fog nodes, in: 16th International Conference on Computer Systems and Applications, AICCSA, 2019, pp. 7141–7151.
[17] T. Wang, P. Wang, S. Cai, X. Zheng, Y. Ma, W. Jia, G. Wang, Mobile edge-enabled trust evaluation for the Internet of Things, Inf. Fusion 75 (2021) 90–100.
[18] W. Fan, Q. Cui, X. Li, X. Huang, X. Tao, On credibility-based service function chain deployment, IEEE Open J. Comput. Soc. 2 (2021) 152–163.
[19] V. Chang, J. Sidhu, S. Singh, R. Sandhu, SLA-based multi-dimensional trust model for fog computing environments, J. Grid Comput. 21 (1) (2022) 1–19.
[20] G. Carrozzo, M.S. Siddiqui, A. Betzler, J. Bonnet, G. Martínez Pérez, A. Ramos, T. Subramanya, AI-driven zero-touch operations, security and trust in multi-operator 5G networks: A conceptual architecture, in: 2020 European Conference on Networks and Communications, EuCNC, 2020, pp. 254–258.
[21] L. Xiong, L. Liu, PeerTrust: Supporting reputation-based trust for peer-to-peer electronic communities, IEEE Trans. Knowl. Data Eng. 16 (7) (2004) 843–857.
[22] J. Mei, X. Wang, K. Zheng, G. Boudreau, A.B. Sediq, H. Abou-Zeid, Intelligent radio access network slicing for service provisioning in 6G: A hierarchical deep reinforcement learning approach, IEEE Trans. Commun. 69 (9) (2021) 6063–6078.
[23] C. Böhm, C. Plant, Mining massive vector data on single instruction multiple data microarchitectures, in: 2015 IEEE International Conference on Data Mining Workshop, ICDMW, 2015, pp. 597–606.