FEATURE ARTICLE: INNOVATIONS USING BLOCKCHAIN

Blockchain-based SLA Management in the Context of IoT

Ali Alzubaidi
School of Computing,
Newcastle University, UK

and

Al-Lith Computing College,
Umm-Al-Qura University,
KSA,

aakzubaidi@uqu.edu.sa

Ellis Solaiman
School of Computing,
Newcastle University, UK

Pankesh Patel
Fraunhofer USA – Center for
Experimental Software
Engineering

Karan Mitra
Luleå University of
Technology, Skellefteå,
Sweden

In pursuit of effective Service Level Agreement (SLA) monitoring and enforcement in the context of Internet of Things (IoT) applications, this paper regards SLA management as a distrusted process that should not be handled by a single authority. Here, we aim to justify our view on the matter and propose a conceptual Blockchain-based framework to cope with some limitations associated with traditional SLA management approaches.

Keywords*: Blockchain, Internet of Things (IoT), Cloud Computing, Service Level Agreement (SLA)*

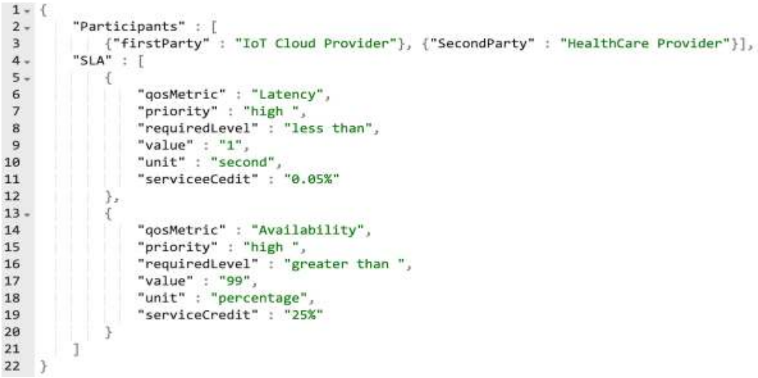
In the realm of cloud service provisioning, the concept of a Service Level Agreement (SLA) has reached an adequate degree of maturity. An SLA acts as a legally binding contract that obligates service providers to comply with their promised Quality of Service (QoS). The cloud computing paradigm enables consumers to focus on their solutions while alleviating the burden of handling overheads related to administration, resources management, and maintenance. Recently, several IoT architectures, for example, Google Cloud IoT and Microsoft Azure IoT Hub, regard cloud computing as an indispensable element of IoT ecosystems. Cloud computing assists IoT applications with efficient data collection, storage, processing, and visualization. When trust between service provider and consumers is well established, SLAs enable consumers to rest assured that the cloud provider will deliver the delegated services as intended. Otherwise, when trust is an issue, SLA monitoring methods can be applied, which should reflect the level at which contracted parties conform to the agreed SLA concerning promised QoS [1].

We have recently seen a growing interest in extending SLA coverage to embrace emergent IoT requirements [2]. However, once an SLA takes place, one might question which party should be trusted as an SLA management authority [3]. This question becomes even more pertinent when dealing with critical systems that are less tolerable to failures [4]. In current practice, cloud providers are commonly assumed for holding the responsibility for typical SLA lifecycle management, such as SLA initiation, service monitoring, and contract enforcement [5]. However, when considering several important factors, such as deliberate corruption, misconduct, opacity, conflict of interest, single point of failure, and incompetent awareness of end-to-end IoT ecosystem requirements, we argue that no single party should solely control SLA lifecycle management [6].

Therefore, we believe that there is a need to rethink the current SLA management models concerning IoT ecosystems. For that, in this article we propose a Blockchain-based framework to address issues such as trust and enforcement. The rest of the paper is organised as follows: First, we briefly overview SLAs in the context of IoT and discuss the limitations of existing approaches. We then shed light on a simple motivating scenario that inspires our approach of using Blockchain technology as a backbone for enforcing SLAs in the context of IoT. Finally, we introduce our framework and conduct a preliminary comparison between consortium and public blockchain initiatives.

SLA IN THE CONTEXT OF IOT

An SLA is a contractual method that specifies and governs service delivery among service providers, consumers, and other parties. IoT applications usually outsource some tasks to cloud providers. The SLA plays a vital role in inter-system orchestration and mediation for IoT applications [2]. QoS metrics are typically defined as part of an SLA in the form of Service Level Objectives (SLOs). A variety of metrics can be agreed on such as throughput, latency, jitter, packet loss rate, availability, reliability, and scalability. QoS metrics should be well-defined in a quantifiable format [7] so that the SLA compliance level can be monitored, measured and statistically reported [8]. Figure 1 depicts a simple agreement on a set of QoS metrics as well as penalties (service credit) between a cloud provider and an application provider; simplified for demonstration purposes.



```
{
    "Participants" : [
        {"firstParty" : "IoT Cloud Provider"}, {"SecondParty" : "HealthCare Provider"}],
    "SLA" : [
        {
            "qosMetric" : "Latency",
            "priority" : "high ",
            "requiredLevel" : "less than",
            "value" : "1",
            "unit" : "second",
            "serviceeCedit" : "0.05%"
        },
        {
            "qosMetric" : "Availability",
            "priority" : "high ",
            "requiredLevel" : "greater than ",
            "value" : "99",
            "unit" : "percentage",
            "serviceCredit" : "25%"
        }
    ]
}
```

Figure 1: A JSON representation of a simplified SLA between IoT cloud provider & a client.

Several monitoring systems [9] have been proposed to enable trust and SLA enforcement. In summary, proactive monitoring tools are aimed at violation prediction and resources management, while reactive counterparts can deal with the aftermath of SLA violation such as those related to responsibility and accountability. Proactive monitoring aims to prevent violations or at least minimise their occurrences. Such schemes are usually associated with adaptive resource management which can help avoid undesirable liability imposed by a relevant SLA. On the other hand, reactive approaches are useful for investigating SLA breaches and their possible causes.

SLA documents should define a set of procedures associated with certain breaches. It is mostly the case that services providers promise to make a judgment in good faith in case of a violation. Typically, it is up to the consumer to detect abnormalities and alleviate any SLA violations, which should incur penalties or remedies on service providers. Notwithstanding, maintaining trust is difficult in such environments without transparency and full access to monitoring tools [5][10]. For that, we can find in the literature several proposals introducing the concept of independent monitoring solutions and auditors to address trust and transparency issues [1][8][11]. Even though, such solutions may struggle to reach an informed decision about the internal state of every involved party [1][5].

SLA conformance in the context of IoT can be a challenging task [6]. That is, many IoT scenarios cannot bear compromising service quality; and therefore, it is essential to rigorously maintain a high degree of SLA compliance by employing an effective monitoring and enforcement mechanism. Although traditional SLA management methods have proven to work reasonably well for cloud service provisioning, we cannot safely assume the same when it comes to critical IoT scenarios. Contrary to typical cloud scenarios, IoT ecosystems require end-to-end coverage that must consider unique aspects associated with IoT applications including, but not limited to, pervasiveness, interconnectivity, large-scale deployment, synchronisation, massive data transfer, distribution, and heterogeneity [12].

Motivating Scenario

Figure 2 shows a critical IoT scenario about a telemedicine ecosystem that should promptly notify related parties about a health emergency. The health application relies on cloud's provided capabilities to remotely gather and analyse patient data. Based on the collected data, there would be some actions triggered such as notifying ambulance or relatives. An SLA is agreed to assure a mutual understanding between the cloud provider and health provider about service delivery.

As simple as this scenario may look, there can be various direct or indirect factors, which could influence service delivery as well as SLA conformance level. Significant failures usually indicate an SLA violation, and thus there must be at least one party held accountable. However, it could be problematic to determine the root cause of a failure and whom to blame. Factors can include, for example, issues related to the cloud provider (e.g., availability, latency, queuing and scheduling, and scalability); at the application level (e.g., software defect); and at the end-user level (e.g. improper usage of a health device, lack of Bluetooth, battery or Internet connection). External entities can also influence service delivery as well. For instance, a wearable device's vendor that does not allow developers a direct access to gathered data, but instead they provide upstream servers to process and analyse data externally. This can lead to unpredictable behaviour, which can impact overall performance.

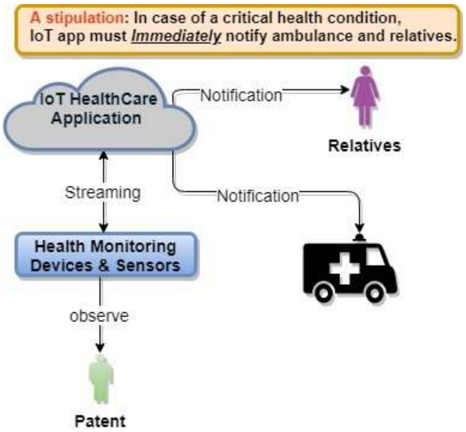

A stipulation: In case of a critical health condition,
IoT app must *Immediately* notify ambulance and relatives.
IoT HealthCare Application
Notification
Relatives
Streaming
Notification
Health Monitoring Devices & Sensors
observe
Patent

Figure 2: A critical IoT scenario: Remote Healthcare system.

As can be seen in the scenario, several involved actors can influence system behaviour and service delivery [4]. Involved participants, serving a common goal, might not individually have a total awareness of the entire system, or even recognise the existence of each other. By considering these factors, we can see that it is not a trivial task for a single entity, such as a cloud provider, to draw a conclusive decision about the main source of a failure.

Machina Research [13] points out that IoT applications will necessitate covering end-to-end IoT requirements. However, the intrinsic nature of IoT makes it difficult for a single entity to attain a full awareness of the entire IoT ecosystem [14]. For example, several IoT cloud providers (such as Microsoft Azure IoT Hub, Google Cloud IoT Core and EVERYTHING platform) do not consider end-to-end IoT SLA assurance, attributing this limitation to the difficulty of covering aspects beyond their reach and control. This is justifiable due to IoT's complex nature, which introduces several important external factors such as third-party software or providers, unsupported operating systems/components, or end-user adherence to required actions such as software updates.

Recently, there has been growing interest in trust establishment between parties involved in a typical IoT scenario. However, if ultimate trust were to be granted to a single authority, there is a risk of issues such as misconduct, malicious acts, lack of transparency and enforcement means, in addition to accidental exceptions. These issues could be improved, if trust was implemented on an efficient distributed technology rather than centralised entities. Currently, most recognised SLA monitoring tools are cloud provider-dependent [15] (examples of cloud monitoring tools are Azure Monitoring tool and Amazon CloudWatch.); meaning that, clients must blindly trust service providers for SLA governance and policy enforcement. This raises concerns about decision neutrality and non-repudiation.

Furthermore, IoT ecosystem complexity makes it a resource-intensive and time-consuming task to handle SLA-related matters such as investigating violation claims. For instance, it can be difficult for cloud providers to gain full control of the IoT physical layer. If a failure was originating from an IoT device, it would be difficult for a cloud provider to track it to its corrective stage. This hinders achieving dispute resolution smoothly because of the inability to determine the situation conclusively. In this case, a judgment in good-faith would neither be practical for cloud providers nor consumers.

BLOCKCHAIN-BASED SLA FRAMEWORK

Ultimately, there is a need for an effective SLA monitoring and enforcement mechanism that should encourage service providers to live up to their promised QoS. In the context of IoT, however, traditional SLA practice can be inefficient due to the limitations discussed above, some of which are presented in Table 1. Such limitations can impair elegant SLA compliance. To address this matter, we believe that any improvement to current SLA practice should primarily consider the following:

- ***Awareness of the entire end-to-end IoT ecosystem***: This is to reduce dispute rates and maintain better compliance level in the first place. For that, a typical end-to-end ecosystem should be considered which consists of at least three layers: (I) The physical layer of a set of resource-constrained sensors which gathers and sends data to (II) Edge computing layer to perform instant computation tasks. That means (III) Cloud services can be dedicated for computationally intensive tasks and storage purposes. A broader awareness covering these layers would improve monitoring capabilities for the sake of better understanding of the system performance, analysis, and failure tracking.
- ***Transparency:*** IoT Cross-layer communication should be enabled such that involved parties can access each other's monitoring tools, perceive remote environments, and other SLA-related co-factors. This requires a shared data infrastructure to facilitate coordination and maintain integrity among parties [16].
- ***Auditability in lieu of Trust:*** SLA management should not be handled by a centralised authority. Every participant should enjoy the ability to exercise their right to express their views on the SLA compliance status.
- ***Minimum Human intervention:*** As many SLA management tasks as possible should be automated in order to save cost, time, and resources. Tasks include, but are not limited to, SLA breaches investigation, billing and dispute resolution.

The contribution of this article is to propose a conceptual framework based on best practices explored by the state-of-the-art research [1][16][17][18]. To realise an effective SLA practice based on our suggested consideration, we propose a trustless approach for conducting SLA management tasks in the context of IoT. Concisely, our approach regards blockchain as an appealing shared infrastructure for engaging key participants in a collaborative manner. Thanks to the smart contract feature, blockchain can potentially solve the issue of the untrusted environment and enforces SLA responsibility and accountability [3]. It also introduces other by-design features that make blockchain technology a quintessential candidate for the goals of this proposal. These features include security, tolerance to node failure, non-repudiation, ledger immutability and auditability. Table 1 shows how blockchain can influence some SLA aspects. As a result, a typical SLA management lifecycle, (e.g. definition, negotiation, monitoring, enforcement, and termination), can be conducted through a blockchain environment.

Table 1: Current SLA Practice vs. Blockchain-based approach.

	Traditional SLA practice	Our Approach
End-to-End awareness	Agnostic	Collective
Trust on a single authority	Mandatory	Mitigated
Transparency	Limited	Encouraged
Conflicts resolution	Manual	Automatic
SLA enforcement	Manual/provider-dependent	Self-enforced
Decision making	Centric	Audit (Shared)
Single point of failure	Possible	Tolerant

Figure 3 sheds light on the overall purpose of our conceptual framework. We consider a typical end-to-end IoT ecosystem that maintains separation of concerns. The main goal of our contribution is that contracted parties should collaborate towards effective SLA management rather than merely trusting a centralised authority. It also aims to achieve a comprehensive overview of the entire IoT system, which requires transparency among all involved participants. Therefore, it encourages all participant to expose a set of relevant tools as RESTful resources. This allows authenticated participants to query and consume published services (e.g. run-time logs, reported incidents, statistics, etc.), for a multitude of purposes such as monitoring, fault detection, analytics, and reasoning. This collaborative environment can also consolidate proactive methods for predicting the overall behaviour and identifying malfunction and reveal any irregularity or contradictions. The RESTful principle allows the framework to be adaptive to any monitoring tools of choice. By covering the end-to-end ecosystem, our blockchain-based framework can enhance SLA compliance and dispute resolution in an automated fashion.

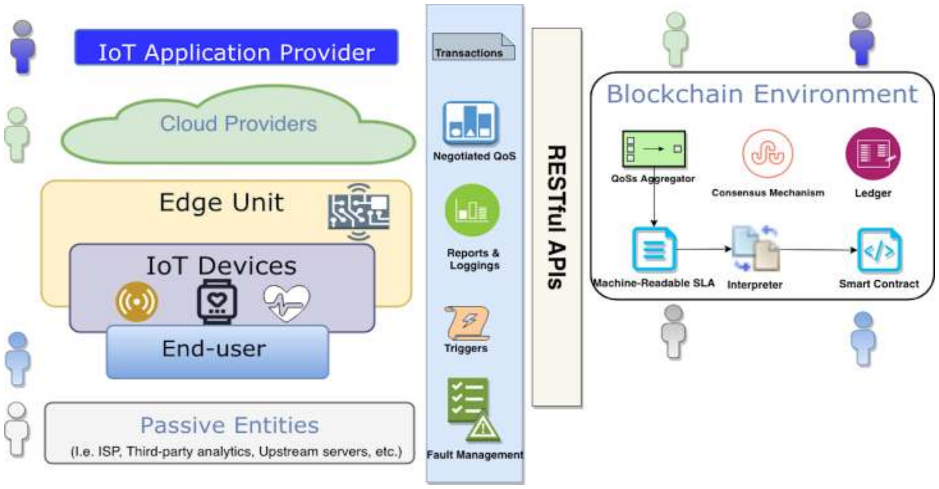


Figure 3: A Conceptual Framework exploits blockchain as a replacement for single authority with regards to SLA Management In the context of IoT.

To enforce accountability and responsibility, we exploit Blockchain-provided features such as auditability and self-enforcement. We represent each participant of a typical IoT ecosystem as a blockchain validating node. Parties can collaboratively exercise SLA management without relying on a centralised authority such as contract initiation, negotiation, monitoring, enforcement, and penalty imposition. Major participants should be able to check each other's adherence to SLA clauses. The framework leverages available monitoring tools or fault management systems provided by some or all participants. These tools should represent the view of a participant on at least their own environment. Once an SLA violation is identified, a claim should be submitted as a transaction to the blockchain, which will be examined against available logged data provided by different stakeholders. A claim transaction can be submitted to the Blockchain environment either manually or can be triggered automatically.

Blockchain can play an active role, such that activities are recorded on the shared ledger. Activities are represented as transactions which must be in accordance with stipulated SLA clauses. Participants, acting as blockchain validators, need to maintain the integrity of the ledger by agreeing on the compliance of these activates. Self-enforcement and penalty imposition can be manifested by exploiting the self-execution feature of smart contracts. Machine-readable SLAs can be represented as smart contracts to enable service governance, enforcement and orchestration. An example of an end-to-end IoT-domain specific language and tool-kit for generating machine-readable SLAs is proposed in [19]. An important future research direction is to expand smart contract capabilities by building suitable interfaces that can convert machine readable SLAs into smart contracts for self-enforcement purposes. However, a smart contract, being represented as code, needs to be well-written, assessed and rigorously vetted. This is important for ensuring deterministic behaviour and meeting the intended performance [20]. This is in order to guarantee reaching a consensus and avoid potential forks.

Consortium Blockchain vs. Public Blockchain

The type of blockchain has a considerable influence on design choices. There is no one-size-fits-all blockchain initiative. Thus, we have conducted a preliminary trade-off analysis (see Table 2.) on some blockchain projects based on their permission type; namely, Ethereum and Hyperledger Fabric. First, we exclude any blockchain that requires centralized coordination (e.g. IoTA), or those that employ a consensus algorithm depending on a single leader such as Corda using BFT-SMaRt algorithm. Second, we impose a constraint such that a blockchain project must support smart contracts.

Table 2: Trade-off Analysis: Ethereum vs. Hyperledger

	Ethereum **(Public)**	**Hyperledger Fabric** **(Consortium)**
Consensus type	Mining	Validation
Authentication	Permission less	Yes
Processing complexity	Difficult	Relaxed
Latency	High	Acceptable
Commitment	Join or leave anytime	Compulsory
Common truth	All must agree	Tolerant
Privacy	Public	Consortium
Participants Roles	Identical	Different role assignment
Energy consumption	Poor	Good
Smart contract expressiveness	Reasonable but limited (Solidity)	Rich and well-established (Java, node.js, Golang)
Cryptocurrency	Dependent	Independent
Architecture modularity	N/A	Pluggable components

Our preliminary research reveals that Hyperledger Fabric, outperforms their counterparts for several reasons. Ethereum can reasonably handle the abuse of trust via rigorous consensus protocols such as Proof-of-Work (PoW). As any other public blockchain, it maintains anonymity such that any node can join or leave with no commitment. This does not serve the purposes of this article because the main reason for having SLAs is primarily to ensure party commitments.

The framework dictates variation of role involvement because agreements usually state different responsibilities assigned to identifiable participants. Ethereum cannot satisfy this requirement while Hyperledger Fabric does. Additionally, public consensus protocols introduce unnecessary hurdles and complexities which can be elegantly avoided using consortium-based blockchains. For example, public blockchains have been associated with the problem of energy consumption and forks attributed to the mining process and propagation, respectively. Driven by the necessity of maintaining a total agreement on the common truth among anonymous entities, certainly, computation-intensive protocols like PoW are substantial, but at the expense of performance. Ethereum incurs fees (gas) in exchange for every smart contract execution while Hyperledger Fabric is cryptocurrency-independent. Moreover, Hyperledger fabric supports modularity which makes it adaptive to different requirements.

All in all, Hyperledger Fabric seems to be the best alternative for delivering the intentions of our framework when considering faster settlement, scalable performance and a more controlled environment. For that, we are conducting an empirical study to validate this outcome and will report to the community in the near future.

CONCLUSION AND FUTURE WORK

The current SLA management model is very difficult to scale for complex and distributed systems such as IoT. Blockchain and smart contract technologies provide exciting opportunities for SLA management in a decentralised and automated fashion beyond the influence of a single central authority. Leveraging the unique characteristics of Blockchain and smart contracts is especially interesting for guaranteeing the QoS requirements in IoT applications. Traditional SLA management techniques are inadequate because they are cloud-dependent and do not cover end-to-end awareness. Centralised-based SLA schemes are susceptible to abuse of trust and lack of enforcement means. This article proposes a conceptual framework to enhance SLA management and bridge the gap towards better QoS assurance in the context of IoT. The framework can adapt to a multitude of IoT scenarios and does not strictly dictate specific components such as monitoring tools or billing systems. However, it emphasises that SLA procedures must be undertaken under blockchain-based collaborative environment to improve enforcement, transparency and trust. Future work will further assess the feasibility of this framework with respect to blockchain technology, smart contracts, automated resolution, performance, and design alternatives.

REFERENCES

[1] W. Hussain, F. K. Hussain, and O. K. Hussain, “Maintaining Trust in Cloud Computing through SLA Monitoring,” Springer, Cham, 2014, pp. 690–697.

[2] A. V. Papadopoulos, S. A. Asadollah, M. Ashjaei, S. Mubeen, H. Pei-Breivold, and M. Behnam, “SLAs for Industrial IoT: Mind the Gap,” in *2017 5th International Conference on Future Internet of Things and Cloud Workshops (FiCloudW)*, 2017, pp. 75–78.

[3] I. Weber, X. Xu, R. Riveret, G. Governatori, A. Ponomarev, and J. Mendling, “Untrusted Business Process Monitoring and Execution Using Blockchain,” Springer, Cham, 2016, pp. 329–347.

[4] G. White, V. Nallur, and S. Clarke, “Quality of service approaches in IoT: A systematic mapping,” *J. Syst. Softw.*, vol. 132, pp. 186–203, Oct. 2017.

[5] K.-W. Park, J. Han, J. Chung, and K. H. Park, “THEMIS: A Mutually Verifiable Billing System for the Cloud Computing Environment,” *IEEE Trans. Serv. Comput.*, vol. 6, no. 3, pp. 300–313, Jul. 2013.

[6] G. Pavlou and I. Psaras, “The troubled journey of QoS: From ATM to content networking, edge-computing and distributed internet governance,” *Comput. Commun.*, vol. 131, pp. 8–12, Oct. 2018.

[7] W. Stallings, *Foundations of Modern Networking*. 2015.

[8] M. Hogan, F. Liu, A. Sokol, and J. Tong, “Nist cloud computing standards roadmap,” *NIST Spec. Publ.*, vol. 35, pp. 6–11, 2011.

[9] S. Mubeen, S. A. Asadollah, A. V. Papadopoulos, M. Ashjaei, H. Pei-Breivold, and M. Behnam, “Management of Service Level Agreements for Cloud Services in IoT: A Systematic Mapping Study,” *IEEE Access*, vol. 6, pp. 30184–30207, 2018.

[10] A. Chandrasekar, K. Chandrasekar, M. Mahadevan, and P. Varalakshmi, “QoS Monitoring and Dynamic Trust Establishment in the Cloud,” Springer, Berlin, Heidelberg, 2012, pp. 289–301.

[11] TMForum, “TR178:Enabling End-to-End Cloud SLA Management V2.0.2 - TM Forum,” 2014.

[12] M. Alodib, “QoS-Aware approach to monitor violations of SLAs in the IoT,” *J. Innov. Digit. Ecosyst.*, vol. 3, no. 2, pp. 197–207, 2016.

[13] Emil Berthelsen, “Service Level Agreements in M2M and IoT,” *Machina Research*, 2014. [Online]. Available: https://machinaresearch.com/report/service-level-agreements-in-m2m-and-iot/. [Accessed: 10-May-2018].

[14] E. Solaiman, R. Ranjan, P. P. Jayaraman, and K. Mitra, “Monitoring Internet of Things Application Ecosystems for Failure,” *IT Prof.*, vol. 18, no. 5, pp. 8–10, Sep. 2016.

[15] T. Labidi, A. Mtibaa, W. Gaaloul, S. Tata, and F. Gargouri, “Cloud SLA Modeling and Monitoring,” in *2017 IEEE International Conference on Services Computing (SCC)*, 2017, pp. 338–345.

[16] S. Seebacher and R. Schüritz, “Blockchain Technology as an Enabler of Service Systems: A Structured Literature Review,” Springer, Cham, 2017, pp. 12–23.

[17] A. Rayes and S. Salam, “IoT Services Platform: Functions and Requirements,” in *Internet of Things From Hype to Reality*, Cham: Springer International Publishing, 2017, pp. 165–194.

[18] S. Mubeen, S. A. Asadollah, A. V. Papadopoulos, M. Ashjaei, H. Pei-Breivold, and M. Behnam, “Management of Service Level Agreements for Cloud Services in IoT: A Systematic Mapping Study,” *IEEE Access*, vol. 6, pp. 30184–30207, 2018.

[19] A. Alqahtani, Y. Li, P. Patel, E. Solaiman, and R. Ranjan, “End-to-End Service Level Agreement Specification for IoT Applications,” in *2018 International Conference on High Performance Computing & Simulation (HPCS)*, 2018, pp. 926–935.

[20] K. Christidis and M. Devetsikiotis, “Blockchains and Smart Contracts for the Internet of Things,” *IEEE Access*, vol. 4. pp. 2292–2303, 2016.

ABOUT THE AUTHORS:

Ali Alzubaidi is a Ph.D. candidate at the School of Computing, Newcastle University, UK. He is also an academic member at the computing college, Umm Al-Qura University. He received his Master degree in Computer Science from Adelaide University, Australia. His research interests include Internet of Things (IoT), Blockchain, Smart contracts and Service Level Agreements (SLA). Contact him at aakzubaidi@uqu.edu.sa or aakzubaidi@gmail.com.

Ellis Solaiman is a Lecturer at the School of Computing, Newcastle University. He previously received his Ph.D. in Computing Science also from Newcastle University, where he subsequently worked as a Research Associate and Teaching Fellow. His research interests are mainly in the areas of Dependability and Trust in Distributed Systems such as the Cloud and the Internet of Things. He is also interested in the automated monitoring of these systems using technologies such as Smart Contracts. He is a Fellow of the UK Higher Education Academy (FHEA) since 2016. Contact him at ellis.solaiman@ncl.ac.uk.

Pankesh Patel is a senior research scientist at Fraunhofer, USA. His focus is on implementation of Industrial Internet of Things (IIoT) techniques and methodologies in commercial environments. He received his Ph.D. in Computer Science from the University of Paris VI (UPMC), France, which was funded by the French National Institute for Research in Computer Science and Automation (INRIA)--Paris, France. Contact him at dr.pankesh.patel@gmail.com.

Karan Mitra is an Assistant Professor at Luleå University of Technology, Sweden. He received his Dual-badge Ph.D. from Monash University, Australia and Luleå University of Technology in 2013. He received his MIT (MT) and a PGradDipDigComm from Monash University in 2008 and 2006, respectively. He received his BIS (Hons.) from Guru Gobind Singh Indraprastha University, Delhi, India in 2004. His research interests include quality of experience modeling, measurement and prediction, context-aware computing, cloud computing and mobile and pervasive computing systems. He is a member of the IEEE and ACM. Contact him at karan.mitra@ltu.se.