IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 13 November 2023, accepted 3 December 2023, date of publication 18 December 2023,
date of current version 4 January 2024.

Digital Object Identifier 10.1109/ACCESS.2023.3344038

RESEARCH ARTICLE

PenChain: A Blockchain-Based Platform for Penalty-Aware Service Provisioning

TRUNG-VIET NGUYEN[1,6], LAM-SON LÊ[2], (Member, IEEE), SYED ATTIQUE SHAH[3], (Senior Member, IEEE), SUFIAN HAMEED[4], AND DIRK DRAHEIM[5], (Member, IEEE)

[1]Faculty of Computer Science and Engineering, Ho Chi Minh City University of Technology (HCMUT), Ho Chi Minh City 70000, Vietnam
[2]Faculty of Engineering, Vietnamese-German University, Ben Cat, Binh Duong 750000, Vietnam
[3]School of Computing and Digital Technology, Birmingham City University, B4 7XG Birmingham, U.K.
[4]Department of Computer Science, National University of Computer and Emerging Sciences (NUCES), Islamabad 44000, Pakistan
[5]Information Systems Group, Tallinn University of Technology, 19086 Tallinn, Estonia
[6]Faculty of Information Technology, Can Tho University of Technology (CTUT), Can Tho 900000, Vietnam

Corresponding author: Lam-Son Lê (lam-son.le@alumni.epfl.ch)

ABSTRACT Service provisioning is of paramount importance as we are now heading towards a world of integrated services giving rise to the next generation of service ecosystems. The huge number of service offerings that will be available to customers in future scenarios require a novel approach to service registry and discovery that allows customers to choose the offerings that best match their preferences. One way to achieve this is to introduce the provider's reputation, i.e., a quality indicator of the provisioned service, as an additional search criterion. Now, with blockchain technology in our hands, automated regulation of service-level agreements (SLAs) that capture mutual agreements between all involved parties has regained momentum. In this article, we report on our full-fledged work on the conception, design, and construction of a platform for SLA-minded service provisioning called PenChain. With our work, we demonstrate that penalty-aware SLAs of general services–if represented in machine-readable logic and assisted by distributed ledger technology–are programmatically enforceable. We devise algorithms for ranking services in a search result taking into account the digitized values of the SLAs. We offer two scenario-based evaluations of PenChain in the field of precision agriculture and in the domain of automotive manufacturing. Furthermore, we examine the scalability and data security of PenChain for precision agriculture.

INDEX TERMS Blockchain, manufacturing industry, penalty-aware services, precision smart agriculture, service-level agreements, service provisioning, smart contracts.

I. INTRODUCTION

Service provisioning is still on the rise in today's information systems as we are heading for a world of massively integrated services and the so-called service ecosystem. Service provisioning is of paramount importance in various business domains such as precision agriculture, supply chain management, and smart tourism [1]. For instance, with the rise of the Internet of Things (IoT), agricultural practitioners and data engineers expect to benefit from real-time data provisioning for the sake of advanced analytics. A customer who wishes to use a data service would choose between offerings that provide the same data packages (e.g., the pH of water and humidity) but at different service levels. Similarly, a tourist might consider multiple tour offerings and pick the most favorable in terms of service levels. Once the service offerings that are made available in such a service ecosystem exceed a critical number, we expect a novel service registry and lookup mechanism.[1]

Distinguishing functionally similar services by their quality is vital for effectively browsing through a large number of service offerings to finally make a decision [2]. Once the

The associate editor coordinating the review of this manuscript and approving it for publication was Claudio Zunino.

[1] Think of such an ecosystem of data services as an e-commerce platform where shoppers search for shopping items by entering a couple of keywords and later on add them to their virtual shopping cart. Search results must be presented in ways that maximize the purchase order placed by the shopper.

© 2023 The Authors. This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License.
For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

customer has picked an offering, she enters a service contract with the provider. The service level at which this service contract comes into effect, hence called the service-level agreement (SLA), captures the mutual agreements between the customer and the service provider. Recordings of these service levels over time, if securely stored using a mutually trusted mechanism, would suggest how reputable a provider has been of late. To this end, search results viewed by the customer should be sorted by the reputation of the service providers before being further customized to match the customer's preference.

The concept of SLA has been investigated to a great extent in service-oriented cloud computing and telecommunications. SLA research falls into the following directions [3], [4]: describing (i.e., formally representing the SLAs taking into account penalty rules) such as iAgree,[2] monitoring (keeping track of whether an SLA is respected during the execution of services), negotiation (mutually adapting an SLA while executing services to avoid a contract violation), and enforcement (executing the rule-based clauses stated in an SLA). Compared to research on the monitoring and negotiation of SLAs, there has been less intensive research on enforcement of SLAs so far. Recently, due to the emerging utilization of blockchains in business, the line of research on the enforcement of SLAs has gained traction in the service engineering community.

Immediately after its introduction with the cryptocurrency Bitcoin [5] in 2009, blockchain technology has received massive, steady attention as a highly potential technology capable of disrupting established systems in various domains by overcoming their over-centralized institutional and technological architectures. Facilitated by this decentralization narrative, blockchain technology has been widely researched and has become productive in a wide range of successful ICT architectures – beyond and often independent of its original utilization in cryptocurrencies [6], [7]. As described by NIST (National Institute of Standards and Technology), "Blockchains are tamper evident and tamper resistant digital ledgers implemented in a distributed fashion [...]" [8]. In this sense, blockchains has been described as a highly promising technology for utilization in organizations [9] as well as between organizations [10], and beyond, at the level of governments and societies [7], [11]. Currently, the blockchain mainstream transcends the blockchain vision into the vision of the so-called Web3.[3] The Web3 vision shows all the ingredients of the blockchain vision and its many initiatives that are centered around a notion of decentralization of payments, financial services, digital identities, data and business models [13], [14], [15]. However, Web3 steps further and aims at taking blockchain decentralization to a next level by making it ubiquitous, turning it into a universal ecosystem. As such, it has been perceived and described recently by leading analysts such as Gartner [16], Forrester [17], Forbes Technology Council [18] and Harvard Business Review [19], [20], [21].

Our research in this realm has resulted in a novel definition of dynamic SLAs that incorporate human-in-the-loop penalty rules and measures the percentages of how often a provider s to the rules stated in an SLA. Eventually, our investigations led to the idea of realizing and enforcing dynamic SLAs on the basis of a private blockchain, which we call penalty-aware SLAs in this paper. In case the provider fails to fulfill a penalty despite having tried all available alternatives, this enforcement mechanism records a breach of contract in an underlying ledger, and thus recalculates the provider's rule-abiding rate.

In our previous studies, we advocated for penalty-aware SLAs in the tourism sector [22], we investigated how digitizing dynamic SLAs makes the enforcement of the penalty rules programmable [23], we conceptualized the SLAs for data service provisioning in smart farming [24]. This article reports on our full-fledged work on the conception, construction, and validation of such a cross-domain platform for SLA-minded service lookup called PenChain. Service-oriented computing and service provisioning are crosscutting concepts with highest relevance for all business domains [25], [26], [27], and, wherever services are provided, proper SLAs are a critical success factor. Similarly, in the last decade, blockchain technology has been successfully applied in countless projects from all kinds of domains, ranging from payment services [14], [28] over financial services [29], [30], insurances [31], [32], real estate [33], [34], manufacturing [35], [36], [37], agriculture [24], [38], [39], food production [40], [41], logistics [42], [43], healthcare [44], [45], tourism [22], [46], to e-government [7], [11], [47], [48], public administration [49], [50], [51], and e-court [52], [53], [54] – just to name a few. Therefore, we argue that PenChain is relevant to a maximally wide range of domains. Similarly, as the design of PenChain does not rely on any domain-specific characteristics, it is not restricted to any specific kind of domain. Indeed, we see PenChain applicability in, e.g., all of the above-mentioned domains.

The contribution of our work is threefold.

- First, we provide formality for dynamic penalty-aware SLAs and devise a ranking algorithm that sorts service offerings according to the reputation of their service providers and the customers' preference.
- Second, we propose a distributed application architecture oriented towards Web3 that enforces penalty-aware SLAs and objectively calculates the reputation of service providers.[4]
- Third, we conduct two scenario-based evaluations of PenChain discussing its use in the case of *high-precision agriculture* and in the case of *automotive*

[2] iAgree Specification http://iagree.specs.governify.io/

[3] Not to be confused with Web 3.0 [12].

[4] Many intermediary websites that bring customers to businesses today still rely on customer feedback to calculate the average rating of a service provider, possibly resulting in unfair assessment and biased reputation [55], [56].

manufacturing industry. The enforcement mechanism devised in PenChain proves to be useful for programmatically enforcing the penalty-aware SLAs in the presented use cases provided that the underlying private blockchain comes with a matched capability and the penalty rules are articulable.

The paper roadmap proceeds as follows. Section II is dedicated to preliminary and related work. In Section III, we formulate our research questions. In Section IV, we propose our general framework for enforcing penalty-aware SLAs. In Section V, we provide a scenario-based evaluation of PenChain as a solution for high-precision agriculture. The scenario offers an in-depth engineering analysis of PenChain for the provisioning of IoT data services. Furthermore, as part of this scenario, we explore some architecture characteristics of PenChain. In Section VI, we provide a scenario-based evaluation of PenChain as a solution for automobile manufacturing. In Section VII, we delve into the potential future directions of our study. We finish the paper with a conclusion in Section VIII.

II. STATE-OF-THE-ART ADVANCEMENTS

In this section, we summarize the scholarly work on SLA-enabled service management and its enabling technologies (blockchains, Web3).

A. SERVICE-LEVEL AGREEMENTS

SLA-related research has been studied extensively in fields such as engineering, economics, management science, and psychology. These articles can be categorized into the following four groups.

- *SLA definition.* SLA is a set of promises, guarantees, and obligations that the service provider makes to the client [57]. SLA provides details not just on service standards but also on any agreed-upon remedies, penalties, or level requirements. Many attempts have been made to explain SLAs, but most of them have focused on online services and cloud computing. The online service-level agreement, also known as the WSLA, was first presented by IBM research as a framework for generating and monitoring SLAs. [58]. Research on this definition of SLA is more diversified in many different domains [59].
- *Real-time SLA monitoring.* The practice of ensuring that service level agreements are met is known as SLA monitoring. It might be performed by a third party or by IT staff. Moreover, SLA monitoring is a radical resolution strategy in situations of SLA breaches and key commercial issues for the services industry [60]. For instance, Labidi [61] described a semantic SLA modeling and monitoring technique for cloud computing. Several theories have been proposed for IoT [62], [63], some focusing on blockchains [64], [65].
- *SLA negotiation.* SLA negotiation is the process of negotiating the level of quality and service that is acceptable to both parties. It is often used in the context of information technology and software development. Some studies have focused on the subject of SLA negotiation, highlighting the responsibilities and problems involved, and proposing frameworks for resolving the issue with cutting-edge technology [66], [67], [68].
- *SLA enforcement.* Research in this area aims to improve SLA management by creating, monitoring, and encouraging users to report service faults. For example, the work conducted by Nakashima [69] leverages Ethereum to propose a collection of Web APIs that automate SLA lifecycle enforcement on a blockchain. Furthermore, Zhou et al. [70] proposed a witness approach to the enforcement of SLAs using smart contracts. In the domain of IoT services, Alzubaidi [71] proposed a blockchain-based decentralized approach to assess SLA compliance and enforce consequences within cloud-based Internet of Things applications.

SLAM [72] is a self-contained, autonomous and trusted framework for continuous SLA monitoring in a multi-cloud environment that is blockchain-based and employs smart contracts to discover SLA breaches. For this purpose, Abhishek et al. [73] presented a blockchain-based system that ensures the integrity of client logs and verifies SLA breaches, resulting in a reliable ecosystem. Neidhardt [74] introduced a blockchain-based monitoring mechanism to ensure customer trust in services. Blockchain technology would help monitor SLAs and improve service trust in cloud computing [57], [75], [76]. Similar solutions may be found in fog computing [77], edge computing [78], and 5G, 6G networks [79], [80]. Viewing the blockchain as a trusted ledger that records all service attributes, a blockchain-based service recommendation system [81] and an auditing solution to protect 5G consumer data [82] are prominent.

Smart contracts are increasingly being used to build autonomous applications [83], [84]. In [85], the public Ethereum blockchain platform is recommended for SLA monitoring and penalty enforcement. Such a framework has been proposed in [86] to monitor SLA terms and compensation in an automatic and decentralized manner using smart contracts and blockchain technologies. They have recommended that compensation should be set up either through basic notifications or automatically through a web application. Singh and Lee [87], [88] provided an approach to a blockchain cloud based on SLA specifications. It provided a more in-depth overview of the construction of smart contracts geared toward SLA. Uriarte et al. [89] presented a distributed SLA management using smart contracts and blockchain technology.

The aforementioned research has a common goal: building more open and egalitarian systems for customers and providers. While the studies we surveyed mostly examine SLAs in cloud computing SLAs over specific parameters, we investigate if a real-life SLA incorporates not only technically measurable indicators but also penalty rules that are contractually articulated to protect the service consumers' rights in many sectors. We follow the recently emerging trend that we make the case in this subsection. More specifically, we utilize distributed ledger technology to determine the reputation of service providers and whether or not a provider is responsible for indemnification.

B. BLOCKCHAIN KEY CHARACTERISTICS

NIST (National Institute of Standards and Technology) characterizes blockchain technology as follows: "Blockchains are tamper evident and tamper resistant digital ledgers implemented in a distributed fashion (i.e., without a central repository) and usually without a central authority (i.e., a bank, company or government). At their basic level, they enable a community of users to record transactions in a shared ledger within that community, so that, under normal operation of the blockchain network, no transaction can be changed once published." [8].

The NIST description is in accordance with the mainstream perception, where de-centralization is a key ingredient of the blockchain technology approach. However, when we aim to have a more systematic understanding of decentralization versus centralization, a more sophisticated approach is needed that also takes into account the differences between various technology objectives as well as levels of centralization – see Table 1. This is important, as decentralization at the level of institutions and decentralization at the level of technology are, in general, incommensurable; rather, they work together to achieve concrete technology objectives.

Disintermediation aims at removing intermediaries "such as banks and notaries" [90] in service of lowering transaction costs [91] in the sense of transaction cost economics [92], [93], [94]. Therefore, disintermediation is about re-shaping institutions as part of "complex technological systems" [94] and even re-shaping the institutional stack of the whole society [95]. Disintermediation is the key ingredient of the blockchain vision [13] and, subsequently, the Web3 vision [13]. Blockchains are seen as game changers here, i.e., as key enablers of disintermediation by providing new *trust anchors*. The old trust anchors (companies, authorities) are replaced by utilizing consensus protocols in peer-to-peer networks. As such, blockchain technology re-combines concepts and algorithms from the fields of peer-to-peer [96] networks and cryptography [97], [98].

Fault tolerance (here: tolerance against failure of hardware/software/network infrastructure) and attack tolerance (tolerance against malicious players) need to be achieved by decentralization at the level of technology, i.e., by means of distributed computing [96]. With blockchain technology, both fault tolerance and attack tolerance are subsumed under Byzantine Fault Tolerance [99], [100], and are addressed by consensus protocols; whereas, with established technology, fault tolerance is addressed by high-availability clusters [101], [102], [103] and attack tolerance is addressed by cybersecurity technology products such as firewalls [104], [105] and VPNs (Virtual Privacy Networks) [105], [106].

High performance is achieved by parallelization. In the context of databases, parallelization (which is a form of technological decentralization) is investigated under the label of sharding, both in blockchains [14], [107], [108] and established database technology.[5],[6],[7],[8]

In blockchains, timestamping is a critical technical feature needed in distributed consensus protocols [109]. Timestamping has its own right from the perspective of business logic, which is usually referred to as document timestamping. Document timestamping does not necessarily need to rely on disintermediation. For example, the blockchain of KSI (Keyless Signature Infrastructure) [110], [111] can be run centrally by a central service provider; still, the KSI blockchain utilizes decentralization at the level of technology to address fault tolerance and high performance [112]. The established (decentralized) base technology for implementing timestamping is public key infrastructure (PKI) [113]. Within a single organization, timestamping can be achieved by (centralized) database journaling [114], [115].

C. BLOCKCHAIN-ENABLED PLATFORMS FOR DATA EXCHANGE AND SERVICE PROVISIONING

IoT redefines existing protocols to establish a more connected network of devices, allowing data to be readily collected and exchanged even when there is no formal human-to-human or human-computer contact. That is why IoT data has a huge potential for software firms and will expand even more when blockchain technology is used. Liu et al. [116] asserted that blockchains could provide high-quality and secure data sharing for industrial IoT applications. In parallel, many studies are being conducted on blockchain platforms to address data management and integrity concerns brought about by IoTs in various industries [117]. Furthermore, the blockchain was connected to edge computing servers to improve data quality and securely handle the compute-intensive activities demanded by IoT devices [118]. Ardagna et al. [119] presented a reliable data collection method based on blockchain technology and smart contracts. It attempts to filter out any untrusted data based on trust criteria. Moreover, this method would evaluate the state of IoT devices and the gathered data. Blockchain-based IoT solutions are ideal for streamlining company automation, achieving substantial cost savings, and improving user experience [120]. Using blockchain technology and SDN, Hameed et al. [121] proposed a scalable method for the IoT device key and trust management. Simulation demonstrates that this combination can store the public keys of IoT devices on the blockchain and efficiently route network traffic using SDN. Similarly, Siddiqui et al. [122] observed that with the assistance

[5] https://www.oracle.com/database/sharding/
[6] https://www.infoq.com/news/2011/02/SQL-Sharding/
[7] https://shardingsphere.apache.org/
[8] https://hbase.apache.org/

TABLE 1. Landscape of various technology objectives; in regards to the main subjects of decentralization, supporting blockchain technology concepts, and supporting concepts of established technology.

Motivation	Decentralization	Blockchain Technology Concepts	Established Technology Concepts
Disintermediation	Institutions	Consensus protocols	Peer-to-peer networks, cryptography
Fault Tolerance	Technology	Consensus protocols	High-availability cluster
Attack Tolerance	Technology	Consensus protocols	Firewalls, VPNs
High Performance	Technology	Sharding	Sharding
Timestamping	Inst./Techn.	Distributed ledger, KSI blockchain	Public key infrastructure, journaling

of AI, adaptive resource management frameworks for IoT networks would be developed, including blockchain-based SDN frameworks.

Distributed ledger technology has emerged as an enabler of accountability and transparency in cloud computing [123] and network service [124]. Expanding on the application of blockchain technology, Singh et al. [125] delved into its potential to improve the transparency and security of electronic health records. This exploration of blockchain's capabilities aligns with the broader efforts in various industries, such as IoT and healthcare, to address data management and integrity concerns. Recent years have witnessed a growing research interest in the use of blockchains in monitoring the service provisioning and enforcing the SLAs of unmanned aerial vehicles (UAV). Most significantly, the selection of the best-fit UAVs in terms of reputation and operational costs is governed using blockchain technology and machine learning [126].

D. WEB3

According to the current most standard explanation, the web has emerged in stages of Web 1.0 and Web 2.0 and is currently to be transformed into Web 3.0 [12]. In parallel, the vision of Web3 [16] is currently widely discussed by major technology analysts such as Gartner [16], Forrester [17] and Forbes Technology Council [18] as well as the Harvard Business Review [19], [20], [21]. Web3 needs to be carefully distinguished from the standard Web 3.0 vision[9] – although the objectives of Web3 and Web 3.0 share some commonalities.

According to the (rather minimal) standard explanation, the web started as a static web, named Web 1.0, which was about organizations sharing their content. Next, Web 2.0 was about enabling individuals to share their content. Practically, this can be connected to the emergence of social media platforms. Additionally, Web 3.0 [12], also coined the Semantic Web [127], is about turning the web into a web of knowledge. This means that knowledge presentation is standardized so that it becomes automatically processable. Consequentially, Web 3.0 is typically identified with the W3C (World Wide Web Consortium) standards RDF (Resource Description Framework)[10] and OWL (Web Ontology Language).[11] Albeit this standard explanation reflects some of the most important aspects of the history of the Web, it does not adequately grasp some other crucial, more complex evolvements. In particular, a major stage of the Web, which is not reflected in the standard explanation, was the introduction of e-commerce. Technologically, this was enacted by the introduction of SSL (Secure Socket Layer), which was the necessary precondition to enable payments on the Internet [128]. Then e-commerce was also the driver of web technologies for data exchange and service computing [129]. Furthermore, Web 2.0 needs a closer look. The vision was about enabling individuals to create content on the Web, however, originally in the vein of the peer-to-peer ("distributed") mentality of the early days of the web. What we have actually seen, however, is the emergence of tech giants (Big Tech) getting into control over the individuals' data – it was exactly this development that has led [130], [131] to propose the web decentralization project Solid (Social Linked Data) [132], [133]. The Web3 vision takes blockchain disintermediation to the next level by making it ubiquitous, encompassing not only payments and financial services but also digital identities, data, and business models [15]. [20] have characterized the Web3 as "a decentralized, blockchain-based internet ecosystem owned and operated by its users" and their expectations are high towards Web3 being "our chance to make a better internet" [20]. We summarize a series of most significant Web3 characteristics as present in the current discourse by comparing each of them briefly with the current situation of Web 2.0 [14], [15]:

- *Web3 Payments.* In the current web, payments are online bank transfers between accounts that are hosted by commercial banks. There exist "digital payments in existing currencies – through Paypal and other »e-money« providers such as Alipay in China or M-Pesa in Kenya" [134], however, these digital payments still rely on bank transfers in the backend. In regards to today's tiered monetary system, payments are therefore done

[9] In some sources Web3 might be named Web 3.0; still, it then has to be distinguished from the standard Web 3.0 vision. But these sources get fewer and can be neglected. Henceforth, in this article, we use Web 3.0 for the standard Web 3.0 vision.

[10] https://www.w3.org/TR/rdf-syntax-grammar/

[11] https://www.w3.org/TR/owl-features/

with M1-money – due to the necessary involvement of commercial banks [135]. Instead, in Web3, cryptocurrencies enable direct payments between web users, i.e., payments without intermediaries. The genuine Web3 currencies are neither owned by a central bank nor collateralized commercial bank money, i.e., they do not belong to the established monetary system and, therefore, cannot be classified as being M0- or M1-money. Note that the central bank digital currency [136] is usually not considered part of the Web3 vision.

- *Financial Services.* Currently, financial services are not considered a part of the web, even though they are made accessible through web-based e-commerce services. Instead, the Web3 vision relies on built-in DeFi [137], [138], [139]. Here, financial services are considered an integral part of Web3 – disrupting both established commercial banking and investment banking.
- *Identity.* In the current web, online identities are created and linked to real-world identities through legally trusted entities, which rely on established routines of personal identity proofing [140]. These online identities rely on public key infrastructures (KPIs) or cloud-based identity solutions, each of them with a different level of technological and organizational maturity [141]. Authorities and companies serve as trust anchors in the creation of online identities. The Web3 stands for a paradigm shift and strives for self-sovereign identity [142], [143] – consequentially extending the tradition of the peer-to-peer community [96]. Consensus protocols are seen to replace traditional trust anchors, and again disintermediation is seen as the crucial notion.
- *Data Ownership.* In the current Web, data is owned and utilized by companies. Instead, in the Web3 vision, data is owned and utilized by the users [20], [144].
- *Business Models.* From the perspective of Web3, the current web is dominated by Silicon Valley tech giants such as Alphabet, Amazon, and Meta. Business models center around super-scaling e-commerce and social media/networks that commercialize the data of their customers. The Web3 envisions new business models [145] that are (i) based on new forms of organization such as the decentralized autonomous organization (DAO) [146] and/or (ii) rely on the utilization of genuine Web3 currencies (cryptocurrencies) or other Web3 assets such as NFTs (non-fungible tokens) [147], [148], [149]. Genuine DeFi business models (decentralized payment services, decentralized fundraising, decentralized contracting) are particularly important [138] for the Web3 vision.

The vision of ubiquitous integration of emerging technology has become widely known as the Internet of Things (IoT) – the Web3 vision can be characterized as the *Web of Everything*, and even more, the *Web of Everything and Everybody*, since the idea of being "owned and operated by its users" [20] can be considered the key ingredient of Web3.

III. RESEARCH QUESTIONS

In service computing, SLA description languages have gained a lot of research attention. Due to the lack of technological support, executing an SLA that has been agreed upon by the provider and its consumers without human intervention remains one of the most challenging questions. Before we entered the digital transformation era, the mainstream thought on the subject was to introduce a regulatory entity whose main job is to monitor and referee service provisioning. This agent[12] keeps an eye on all service transactions and, in case of dispute, might invoke a pre-programmed unit to kick off a negotiation workflow to avoid a breach of contract. As distributed ledger technologies continue to advance at a rapid pace, enforcing the SLAs has now become a technologically supported research attempt. However, before reaching this point, we will have to address a few research questions in the following. Let us elaborate on these research questions in Subsections III-A, III-B, and III-C.

RQ1: The penalties specified in a dynamic SLA dictate what the provider and its customers should do in the event of a dispute. To make these rules computer-interpretable, in what ground logic or formal languages shall we express them?

RQ2: How to digitize penalty-aware SLAs to enable an enforcement mechanism and the computation of an objective unbiased reputation in service provisioning?

RQ3: How to gear up a distributed ledger to enforce relevant penalty rules during a service transaction?

A. IN WHAT GROUND LOGIC OR REPRESENTATIONAL TECHNIQUE WOULD THE PENALTY RULES BE EXPRESSED? (RQ1)

Handling penalty rules becomes increasingly important in service operations, as evidenced by a great amount of scholarly work on rule-based modeling and monitoring [150], [151], [152], [153], [154]. We head for dynamic SLAs (which incorporate human-mediated factors such as the penalty) as opposed to static ones (which mainly describe uptime/downtime and availability constraints). Although a penalty rule is in place to primarily protect service customers, it should give the provider multiple chances to repair any spontaneous SLA violation during a service transaction. Should the provider run out of opportunities to take action to fix such a violation, a breach of contract is finally recorded to report that the customer's expectation was not met? In other words, the penalty rule, being triggered by a spontaneous SLA violation, dictates a path for both the customer and the provider to follow through with the goal of fixing the said violation and eventually discarding it. We were in search of a ground logic or a formal language expressible enough to represent a penalty rule that articulates cascaded reparation actions.

[12] Centralized computing was particularly made popular in the Web2 era. In light of Web3, we should replace such an agent with a decentralized application that is meant to perform the same job.

B. HOW TO DIGITIZE THE PENALTY-AWARE SLAS TO ENABLE AN ENFORCEMENT MECHANISM AND THE COMPUTATION OF AN OBJECTIVE UNBIASED REPUTATION IN SERVICE PROVISIONING? (RQ2)

The dependability of a service system or an ecosystem of services is in part linked to how we control the SLAs. We argue that, unlike many present-day intermediary websites that rely on subjective customer feedback, the next-generation service ecosystem should employ a novel technique for objectively calculating the service providers' ratings using logs of computerized SLAs. Digitizing penalty-aware SLAs is non-trivial for the following reasons: (a) Suppose we have found an appropriate logic for the representation of the penalty rules, we are still in need of a computer in the loop to fire the reparation actions articulated in these rules without human intervention; (b) Digital evidence of service provisioning (e.g., breach of contract, service quality not met, missing items) involving multiple parties has to be collected from heterogeneous computing devices[13] and must be recorded safely in an integrity-assured database.

C. HOW TO GEAR UP A DISTRIBUTED LEDGER FOR ENFORCING THE PENALTY RULES? (RQ3)

The enforcement of SLA-bound penalty rules shall not be (a) tampered with by any involved parties; or (b) operated by humans despite the possible involvement of anthropomorphic entities. With distributed ledger technologies now on the rise, we expect to use blockchain's smart contracts and ledger capabilities as technological leverage to execute such human-in-the-loop penalty rules. A smart contract is essentially a program stored on a blockchain and executed automatically when certain conditions are met. Technically speaking, we need to translate the logic of these human-in-the-loop penalty rules into the computer-in-the-loop program code of one or more smart contracts.

IV. THE PENCHAIN FRAMEWORK

In this section, we first describe the top-level components of PenChain (Subsection IV-A), which is complemented by Subsection IV-E looking under the hood. In the next subsections IV-B, IV-C, and IV-D, we lay the groundwork for the representation of SLAs, how to combine them for the sake of service bundling, and how to rank service offerings by their SLA. The algorithms are presented in Subsection IV-F. In Subsection IV-G, we showcase the chief function of PenChain.

A. OVERALL ARCHITECTURE

As illustrated in Figure 1, we proceed by defining the top-level components of PenChain together with those on the provider's side and the customer's side. The nodes of *Customer-Node*, *Provider-Node*, and *PenChain-Main* each represent a cloud or a computing server. *Customer-Node* refers to a node that hosts the user interface (*Browser*) through which customers request a service registered in PenChain. *Provider-Node* hosts *Browser* and another component that facilitates service provision (*Providing Services*). As a customer in PenChain might one day become a provider following the rationale of Web3, *Provider-Node* should be backward compatible with *Customer-Node*.

PenChain-Main is the node that hosts the following components of PenChain: service discovery, service log, SLA enforcement, and quality assessment. The component of *System Log*, which captures the log of all service transactions, is connected to *Service Registry* and *Service Façade* via a ternary connector. The component of *Assessment Unit* sends an assessment of artifacts created during a service transaction to the underlying blockchain, hence a connector that links it to *SLA AutoGovern Blockchain*. This assessment is also sent to *System Log* for storing the transactional evidence of the service being consumed. *Service Façade* is the entry point for customers to search for individual service offerings or a service bundling. *Service Registry* facilitates the discovery of services and allows the query of their SLA.

According to the principle of non-interference, as we demonstrate in Figure 1, service delivery actually occurs outside *PenChain-Main*. For instance, an accommodation service is an exchange act that takes place between a hotel and a group of tourists with the minimum intervention of PenChain. This concern is even more prevalent in the domain of data provisioning due to data privacy – data exchange takes place between the consumer and the provider without having *PenChain-Main* participative involved. Though *Assessment Unit* in our architecture records transactional evidence by invoking components *Logging* (to write in a cloud database) and *Quality Assessment* (to obtain an assessment of the service outcome), it is nevertheless not participatively engaged in any service transaction mediated by PenChain.

Within the *SLA AutoGovern Blockchain* structure, two pivotal components stand out: the *Smart Contract for Penalty Rules* and the *Smart Contract for Reputation*. The former autonomously enforces penalty regulations, while the latter systematically evaluates service providers' adherence to SLA parameters. By judiciously incorporating smart contracts, this intricate structure signifies a pioneering step in penalty rule management and reputation assessment, reshaping SLA administration.

B. CAPTURING PENALTY RULES TO REPRESENT A PENALTY-AWARE SLA

Our formal reasoning offered in this subsection is twofold: (a) the modeling of the penalty rules, which explicitly addresses research question RQ1; (b) the logic behind comparing the SLAs, which in part addresses research question RQ2.

A penalty rule represents contractual obligations between service providers and consumers. Modeling these rules requires the use of a language designed for contract modeling that considers them as business contracts. Business contracts

[13] IoT wearables, QR code/barcode readers, fingertip sensors, etc. in use today make a wide range of technical choices to computerize service encounters during a service transaction.

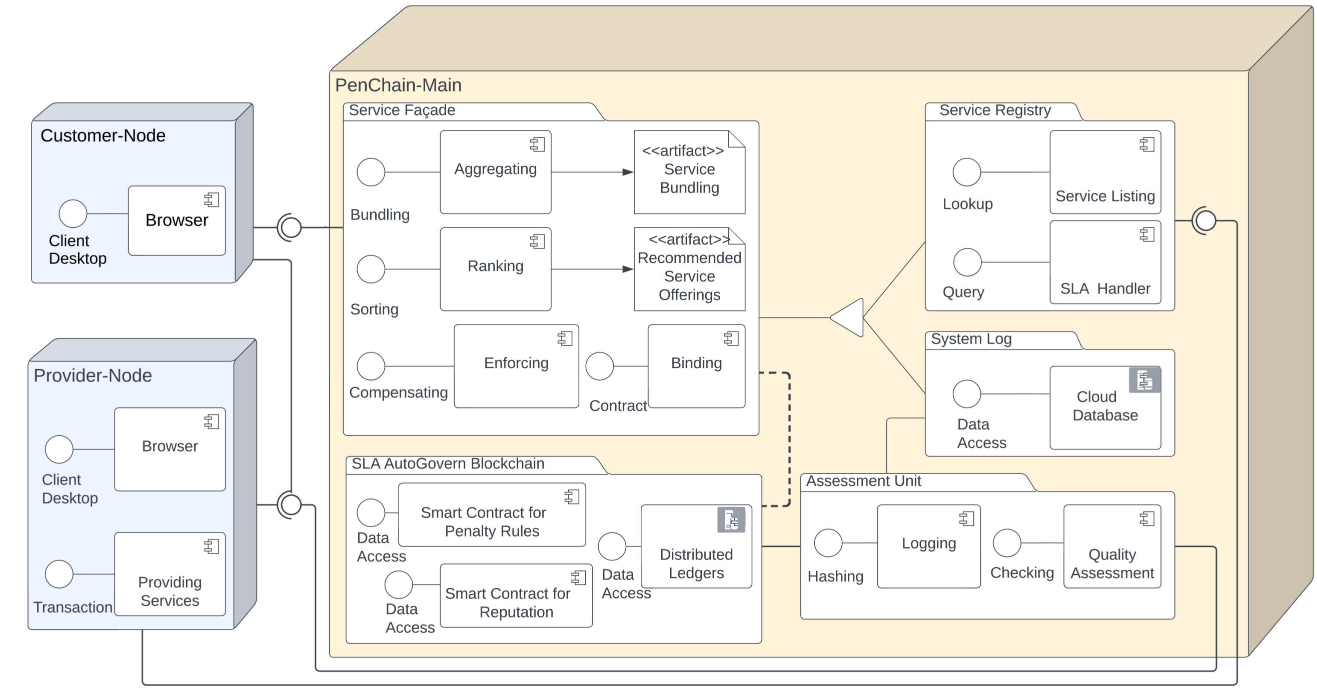


FIGURE 1. The component view of PenChain's system architecture.

specify obligations, permissions, and prohibitions as mutual agreements between business parties [155], as well as the terms of the contract breach. In the context of our study, we are more interested in the ways in which the rules are formulated and the distribution of compensation. Deontic operators capture the contractual modality (i.e., obligations, permissions, and prohibitions) [156]. Governatori represents a contractual rule as $r : A_1, A_2 \ldots A_n \vdash C$ where each A_i is an antecedent of the rule and C is the consequent. Each A_i and C may contain deontic operators. Specifically, the connective $\circledast$ can therefore be informally read as "failing which". This means that the symbol $\circledast$ can be used to represent a relationship between two compensations. It implies that if the first compensation is not fulfilled, the second compensation will be carried out. $O_{hotel}\alpha \circledast O_{hotel}\sigma$ mandates that a hotel be required to ensure that α is achieved. Failure to do so results in a violation, which the hotel can repair by supplying σ. Governatori offers additional formality to reason about the merging of contractual rules in his work.

Example 1: Let us consider the situation where a tourist and her family decided to book a deluxe room at Jade Hotel for their summer vacation. A penalty rule in this accommodation service reads: *"Jade hotel will provide a room with a balcony and a view of the sea. If the hotel cannot arrange a sea view room as promised, it shall arrange an alternative room with a luxurious interior design, or a king-size double bed. If this compensation option is unavailable, Jade Hotel will offer two adult members of the tourist family free access to the hotel's spa during their stay. If neither of these compensations could be made, the tourist is entitled to a 50% discount on her accommodation at Jade Hotel upon checkout."*. We formally express this penalty rule as,

$$\boxed{r : \neg seaview \vdash O_{hotelAltRoom} \circledast O_{hotelFreeSpa} \circledast O_{hotelDiscountWhenCheckout}}$$

where the components are expressed in first-order logic as follows:

$$\boxed{seaview \equiv \exists ro \in Jade,\ t \in Tourist : booked(t, ro) \wedge BalconyWithSeaView(ro)}$$

$$\boxed{AltRoom \equiv \exists ar \in Jade, t \in Tourist : (LuxuryInterior(ar) \vee KingSizeDoubleBed(ar)) \wedge CheckIn(t, ar)}$$

$$\boxed{FreeSpa \equiv \exists spa \in Service, ar \in Jade, t \in Tourist : FreeAccess(ar, spa) \wedge CheckIn(t, ar)}$$

$$\boxed{DiscountWhenCheckout \equiv \exists ro \in Jade,\ t \in Tourist,\ spa \in Service : booked(t, ro) \wedge CheckOut(t, ro) \wedge Discount50Percent(t, ro)}$$

Figure 2 elucidates the same penalty rule diagrammatically by representing all the aforementioned deontic components as pictograms. The diamond pictogram to the left of this diagram signifies the triggering condition of the penalty rule. The three rounded rectangles next to this diamond stand for compensation actions stated in the rule. A block arrow corresponds to either a compensated outcome or an uncompensated outcome. A thin arrow connecting two compensation actions implies a "failing which" connective, i.e., should the action at the start of an arrow fail, the action at its end shall be triggered.

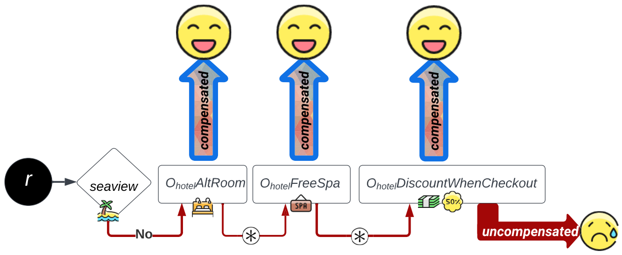


FIGURE 2. Diagrammatic equivalent of the deontic logic for the penalty rule publicized by Jade Hotel.

Transitioning to a distinct realm of study, we encounter the concept of a semiring, an algebraic structure characterized by two fundamental operations. A semiring can be thought of as being similar to a ring but without the requirement that the addition operation be commutative. The two operations in a semiring are called addition and multiplication. In other words, it is an algebraic structure that generalizes the properties of a ring and a semigroup [157].

Definition 1 (Semiring): A *semiring* is a tuple $\langle \mathcal{A}, \oplus, \otimes, \bar{0}, \bar{1} \rangle$ where,

- $\mathcal{A}$ is a set and $\bar{0}, \bar{1} \in \mathcal{A}$;
- $\oplus$, called the *additive operation*. It is a commutative, associative operation having $\bar{0}$ as its neutral element (i.e. $a \oplus \bar{0} = a = \bar{0} \oplus a, \forall a \in \mathcal{A}$);
- $\otimes$, called the *multiplicative operation*. It is an associative operation such that $\bar{1}$ is its identity element and $\bar{0}$ is its absorbing element (i.e. $a \otimes \bar{0} = \bar{0} = \bar{0} \otimes a, \forall a \in \mathcal{A}$). Moreover, $\otimes$ distributes over $\oplus$ (i.e. $\forall a, b, c \in \mathcal{A}$, we have $a \otimes (b \oplus c) = a \otimes b \oplus a \otimes c$).

An idempotent semiring is a semiring whose additive operation is idempotent (i.e., $a \oplus a = a$). This idempotence property allows us to endow a semiring with a canonical order defined as $a \preceq b$ iff $a \oplus b = b$. We specify the meaning of operators $\oplus$ and $\otimes$ for reasoning about services in Subsection IV-C.

In our work, a set denoted $\mathcal{A}$ refers to the set of SLAs in a certain business domain. We consider $\mathcal{A} = \{\langle x_1, x_2..x_n \rangle\}$, where x_i represents the i-th most important objective of an SLA. In particular, we make a case for the three SLA objectives defined below with the objective of penalty rules being understandably considered the foremost. We acknowledge that different customers might develop different views on the significance of these SLA objectives. In other words, SLA objectives should be flexibly sequenced rather than rigidly sequenced in what we call the customer-centric representation[14] of SLAs. Now, let us define the notions of satisfaction, rule-abiding rate, and cost in Definition 2, Definition 3, and Definition 4, respectively.

Definition 2 (Satisfaction): Informally, the satisfaction of a service perceived by end users is related to the reliability and empathy of this service [158]. For business services, the *satisfaction* is formally defined as the ratio of the number of successful service deliveries to the total number of service transactions.

Definition 3 (Rule-Abiding Rate): A service is associated with several penalty rules, which are supposed to be respected at run-time. The *rule-abiding rate* (and the *breach rate*) of a penalty rule is defined as the ratio of the number of times the said rule is respected (unrespected) to the total number of times the said rule is activated. Formally, we have $rule-abidingrate = 1 - breachrate$. A *rule-abiding rate* of an SLA, or *rule-abiding rate* for short (denoted as r), is defined as the minimum value of rule-abiding rates of all penalty rules involved.

Definition 4 (Cost): Let $\mathcal{C}_0 = \{c_1, c_2, \ldots c_n\}$ be the initial set of values of the cost incurred by services. By *closure* of $\mathcal{C}_0$, we mean the smallest set that contains all finite summations of elements in $\mathcal{C}_0$: $\mathcal{C}_0^+ = \left\{\sum_{k=1}^{\infty} \left(c_{i_1} + \ldots + c_{i_k}\right) | \; c_{i_k} \in \mathcal{C}_0\right\}$. As such, the **cost set** is defined as $\mathcal{C} = \mathcal{C}_0^+ \cap [0, cost_{max}]$ where $cost_{max}$ is the highest cost the customer can pay. A *cost* (denoted as c) is a payment for a service, which is an element of $\mathcal{C}$.

C. AGGREGATING SERVICES

The practice of aggregating services, colloquially known as service bundling, is exercised in various business domains. The central idea of service aggregation is to offer additional service offerings as packages at a more competitive price.

To proceed with service bundling, for simplicity, let us assume that the SLAs of all the atomic services are denoted as a set of triples $\mathcal{A} = \{\langle r, s, c \rangle\}$ where rule-abiding is the most important objective, satisfaction: the second, and cost: the least. Now we aggregate the SLAs of the constituent services. Suppose that we have n services and consider the i-th service for which we write $L_i = \{\ell_{ij} | j = 1, \ldots, m_i\}$ where ℓ_{ij} represents the j-th level of its multilevel SLA.

Now, let us put $\alpha^i_{\ell_{ij}}$ to denote the j-th level of the i-th service's multilevel SLA where $i \in [1, n]; j \in [1, m_i]$. This construct is in fact a triple $\langle r, s, c \rangle \in \mathcal{R} \times \mathcal{S} \times \mathcal{C}$, where $\mathcal{R}, \mathcal{S}, \mathcal{C}$ are the sets of rule-abiding rates, satisfactions and costs, respectively.

Definition 5: To represent the multilevel SLA of service aggregation, we consider a group of aggregated services as a subset of k services $\{so_i \,|\, i \in I \subset [n], \; |I| = k\}$ each of which comes with multilevel SLA as exemplified in Table 4. Out of this subset, the cartesian product of the multilevel SLAs of the aggregated services is represented by a 2-dimensional array of size $(\prod_{i \in I} m_i) \times k$, where each element is a triple $\langle r, s, c \rangle$. For each given row consisting of k SLAs in this array, we already know the fixed level $\ell_i \in L_i$ is defined. Hence we write $\langle r^i_{\ell_i}, s^i_{\ell_i}, c^i_{\ell_i} \rangle$, $i \in I$, for the SLAs in this row. Then we define the *combined SLA* (denoted as Ω), over these SLAs as follows:

$$\Omega = \bigodot_{i \in I \subset [n]} \alpha^i_{\ell_i} = \langle \min_i r^i_{\ell_i}, \min_i s^i_{\ell_i}, \sum_{i \in I} c^i_{\ell_i} \rangle \qquad (1)$$

[14] We shall take this customer-dependent representation of SLAs into account when reasoning about the aggregation and ranking of services in the next subsections. This flexible sequence of SLA objectives will translate into the algorithmic dynamics for service ranking in Subsection IV-F.

TABLE 2. Service aggregation involves compiling the SLAs of the constituent services.

Label	Service aggregation	Combined SLA $\langle r, s, c\rangle$
Ω_1	so_1 (Intermediate), so_4 (Basic), so_5 (Intermediate)	$\langle 93\%, 94\%, \$35\rangle$
Ω_2	so_1 (Intermediate), so_4 (Basic), so_5 (Advanced)	$\langle 93\%, 94\%, \$45\rangle$
Ω_3	so_1 (Advanced), so_4 (Basic), so_5 (Intermediate)	$\langle 94\%, 95\%, \$40\rangle$
Ω_4	so_1 (Advanced), so_4 (Basic), so_5 (Advanced)	$\langle 94\%, 95\%, \$50\rangle$

Example 2: We request a group of services as subset $\{so_1, so_4, so_5\}$ (i.e. $k = 3$) and $I = \{1, 4, 5\}$. Suppose that we have $L_1 = \{\ell_{11} : \mathit{Intermediate}, \ \ell_{12} : \mathit{Advanced}\}$, $L_4 = \{\ell_{41} : \mathit{Basic}\}$ and $L_5 = \{\ell_{51} : \mathit{Intermediate}, \ \ell_{52} : \mathit{Advanced}\}$. Of this subset, the Cartesian product of this set is represented by a two-dimensional array of size $(\prod_{i \in I} m_i) \times 3$ as follows,

$$\begin{array}{ccc}
\hline
\langle r^1_{\ell_{11}}, s^1_{\ell_{11}}, c^1_{\ell_{11}}\rangle & \langle r^4_{\ell_{41}}, s^4_{\ell_{41}}, c^4_{\ell_{41}}\rangle & \langle r^5_{\ell_{51}}, s^5_{\ell_{51}}, c^5_{\ell_{51}}\rangle \\
\hline
\langle r^1_{\ell_{11}}, s^1_{\ell_{11}}, c^1_{\ell_{11}}\rangle & \langle r^4_{\ell_{41}}, s^4_{\ell_{41}}, c^4_{\ell_{41}}\rangle & \langle r^5_{\ell_{52}}, s^5_{\ell_{52}}, c^5_{\ell_{52}}\rangle \\
\hline
\langle r^1_{\ell_{12}}, s^1_{\ell_{12}}, c^1_{\ell_{12}}\rangle & \langle r^4_{\ell_{41}}, s^4_{\ell_{41}}, c^4_{\ell_{41}}\rangle & \langle r^5_{\ell_{51}}, s^5_{\ell_{51}}, c^5_{\ell_{51}}\rangle \\
\hline
\langle r^1_{\ell_{12}}, s^1_{\ell_{12}}, c^1_{\ell_{12}}\rangle & \langle r^4_{\ell_{41}}, s^4_{\ell_{41}}, c^4_{\ell_{41}}\rangle & \langle r^5_{\ell_{52}}, s^5_{\ell_{52}}, c^5_{\ell_{52}}\rangle \\
\hline
\end{array}$$

For each given row consisting of three SLAs, since we already know the fixed level, we, therefore, have $< r^i_{\ell_i}, s^i_{\ell_i}, c^i_{\ell_i} >, i \in I$. Hence, we rewrite the first row as,

$$\begin{array}{ccc}
\hline
\langle r^1_{\ell_1}, s^1_{\ell_1}, c^1_{\ell_1}\rangle & \langle r^4_{\ell_4}, s^4_{\ell_4}, c^4_{\ell_4}\rangle & \langle r^5_{\ell_5}, s^5_{\ell_5}, c^5_{\ell_5}\rangle \\
\hline
\end{array}$$

In this row, we apply Formula 1 to determine the combined SLA and label it Ω_q, where q serves as row indices.

$$\begin{aligned}
\Omega_1 &= \langle \min\{r^1_{\ell_1}, r^4_{\ell_4}, r^5_{\ell_5}\}, \min\{s^1_{\ell_1}, s^4_{\ell_4}, s^5_{\ell_5}\}, (c^1_{\ell_1} + c^4_{\ell_4} + c^5_{\ell_5})\rangle \\
&= \langle \min\{93\%, 94\%, 94\%\}, \\
&\quad \min\{94\%, 95\%, 98\%\}, (\$5 + \$10 + \$20)\rangle \\
&= \langle 93\%, 94\%, \$35\rangle
\end{aligned}$$

We repeat this task for each remaining row in the two-dimensional array above. As a result, Table 2 presents the combined SLAs for the following service aggregation so_1 (Intermediate), so_4 (Basic), so_5 (Intermediate) into Ω_1; so_1 (Intermediate), so_4 (Basic), so_5 (Advanced) into Ω_2; so_1 (Advanced), so_4 (Basic), so_5 (Intermediate) into Ω_3; so_1 (Advanced), so_4 (Basic), so_5 (Advanced) into Ω_4.

D. SEARCH RESULTS RANKING

The set of SLAs for a particular business domain is referred to as $\mathcal{A}$ in this work. We write $\mathcal{A} = \{\langle x_1, x_2 .. x_n\rangle\}$ where x_i represents the i-th most important objective of an SLA. Again, for the sake of simplicity, let us denote the set of SLAs as $\mathcal{A} = \{\langle r, s, c\rangle\}$ where r: rule-abiding rate, s: satisfaction, c: cost. In particular, we argue in favor of the three SLA objectives specified in the following, with the objective of penalty rules being the one that is, naturally, seen as the most important. We understand that the importance of these SLA objectives may be interpreted differently by specific clients at different times. The SLA objectives should not be rigorously sequenced and should have some degree of flexibility in the order in which they are presented in the customer-centric depiction of SLAs.

We recall that PenChain facilitates service aggregation, allowing atomic service offerings to be bundled to enrich the service offerings on offer. They will be mixed up in a search result, necessitating a polymorphic ranking procedure that sorts them by the SLA regardless if they are atomic, bundled or an aggregation of aggregated services. More precisely, we are in search of an effective comparison technique that works uniformly for all SLAs denoted as triples $\langle r, s, c\rangle$.

To devise a comparison technique, we treat $\mathcal{A}$ as a totally ordered set. The total order on a set is a form of ordering the elements of the set, assuming that certain elements precede others, with the understanding that any two elements can be compared in one way or another. Formally, we have $\Omega_i \geq \Omega_j$ if $(r_{\Omega_i} > r_{\Omega_j})$ *or* $(r_{\Omega_i} = r_{\Omega_j}) \wedge (s_{\Omega_i} > s_{\Omega_j})$ *or* $(r_{\Omega_i} = r_{\Omega_j}) \wedge (s_{\Omega_i} = s_{\Omega_j}) \wedge (c_{\Omega_i} \leq c_{\Omega_j})$.

In order to reason about the SLAs, we utilize the mathematical construct of semiring where the $\oplus$ operation is the max operation concerning this order. The $\otimes$ operator is a multiplication that acts differently on each component of an element in $\mathcal{A}$. The $\otimes$ operator's action on $\mathcal{R}$, $\mathcal{S}$ is taking the minimum. The $\otimes$ operator's action on $\mathcal{C}$ is ordinary addition. More precisely, let $a = \langle r_1, s_1, c_1\rangle$ and $b = \langle r_2, s_2, c_2\rangle$, then $a \otimes b$ is defined as follows.

$$a \otimes b := \langle \min\{r_1, r_2\}, \min\{s_1, s_2\}, \ c_1 + c_2\rangle$$

Example 3: Table 3 shows an ordered list of aggregated services where Ω_3 ranks first and Ω_2 is the least preferred aggregation. The column to the right of this table prints the combined SLA of the listed aggregated services, explaining why they are ranked first, second, etc. as shown in the leftmost column.

E. UNDER THE HOOD OF PENCHAIN

Figure 3 illustrates the inner workings of *PenChain-Main* – a node where the components of PenChain are deployed excluding computing from the provider and the customer sides (see Figure 1). We utilize the layered architecture style to describe how components could be developed and deployed in tandem with smart contracts that update the trusted ratings of the service providers and enforce the penalty rules in PenChain. They are organized into three layers, namely *Business Layer*, *Persistence Layer*, and *Blockchain Layer*.

TABLE 3. Service aggregations are sorted by their combined SLA in the same way service offerings are.

Rank	Label	Service aggregation	Combined $\langle r, s, c\rangle$
1	Ω_3	so_1 (Advanced), so_4 (Basic), so_5 (Intermediate)	$\langle 94\%, 95\%, \$40\rangle$
2	Ω_4	so_1 (Advanced), so_4 (Basic), so_5 (Advanced)	$\langle 94\%, 95\%, \$50\rangle$
3	Ω_1	so_1 (Intermediate), so_4 (Basic), so_5 (Intermediate)	$\langle 93\%, 94\%, \$35\rangle$
4	Ω_2	so_1 (Intermediate), so_4 (Basic), so_5 (Advanced)	$\langle 93\%, 94\%, \$45\rangle$

1) BUSINESS LAYER

The layer to the top of Figure 3 represents the business logic of PenChain. It is organized into four groups. Group 1 is made up of interconnected modules that interact directly with service providers and give some input to the other modules in subsequent layers to serialize the service transactions: *Services*, *SLAs*, and *Penalty Rules*. Group 2 (*Aggregator* and *Ranking*) assists the customers in locating a service offering or an aggregated service by computing the SLA parameters in accordance with Definition 5. The modules in this group send requests to the next layer (*Persistence Layer*) to perform data queries on the cloud database. The customer engagement is carried out by Group 3, which is comprised of the following modules: *Search* and *Contract*. In addition to allowing customers to view the contractual specifics of the service they are bound to, including its SLA, the *Contract* module makes any in-progress compensation attempt to fix a breach of contract, which could be activated by invoking *Penalty-Aware SLAs*. The key modules in Group 4 invoke the next two layers to write down the service's transactional evidence to both the cloud database and the blockchain.

2) PERSISTENCE LAYER

As the name suggests, the layer situated in the middle of Figure 3 helps serialize service transactions that were executed. *General* handles the transaction log of a service that is not highly confidential (e.g., location, timestamp, service participants who were engaged in the transaction, and the actual SLA that was associated with the service). *Logs* stores more sensitive information about the service (e.g., quality experienced by the customers and whether it falls below or surpasses a critical threshold value). Both write the service's transactional records they receive from Group 1 and Group 4 of *Business Layer* down to a cloud database (our pick here is Azure SQL Database).

3) BLOCKCHAIN LAYER

The bottom layer depicted in Figure 3 is dedicated to the underlying distributed ledger and essential smart contracts of PenChain. It is designed to store immutable records as successive blocks, each representing a service transaction.

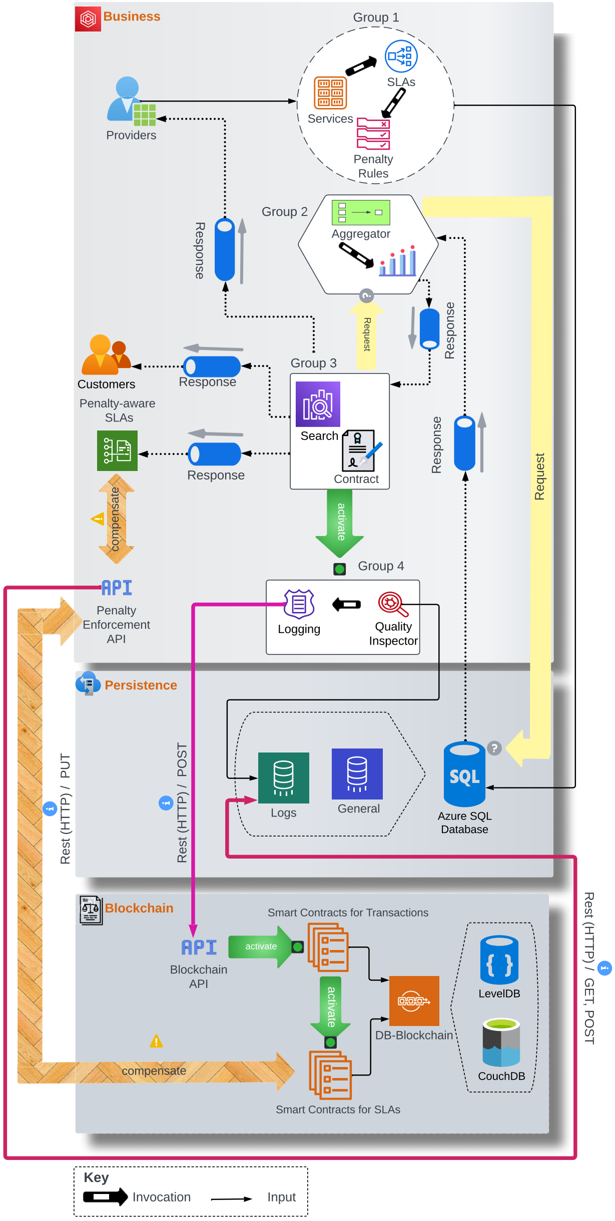


FIGURE 3. PenChain utilizes replaceable and loosely-coupled units for keeping track of the service transactions and enforcing the penalty-aware SLAs. They are conceptually organized into three layers.

Technically, we install a private blockchain using Hyperledger technologies on our cloud. We refer to Hyperledger's storage capacity as *DB-Blockchain*, which is made of two separate data sources: an internal*LevelDB*[15] and an external *CouchDB*.[16] The former holds chaincode data as key-value

[15] LevelDB (https://github.com/google/leveldb) is an open source database system that is based on concepts from Google's Bigtable database system. It could be embedded in the peer process of a blockchain, providing low-level key-value storage and retrieval capabilities.

[16] Apache CouchDB (http://couchdb.apache.org) is an open source database that uses a schema-free cross-platform data model.

pairs, while the latter facilitates sophisticated data queries in the chaincode.

Logging in *Business Layer* submits the assessment of a service transaction and the corresponding hash in Azure SQL Database to *Smart Contracts for Transactions* in this layer through *Blockchain API*, activating *Smart Contracts for SLAs* in order to reassess the SLA's satisfaction. Upon receiving this request, the module submits a compensation request to *Penalty-aware SLAs* via *Penalty Enforcement API* in *Business Layer* if the assessment falls short, indicating an instantaneous breach of contract. Now, the control is granted to the modules in *Business Layer* again. Acceptable indemnification criteria have been incorporated into *Contract*. In effect, *Penalty-aware SLAs* monitors the compensation workflow that has already been activated and sends a confirmation to *Smart Contracts for SLAs* to update the SLA's rule-abiding rate. To illustrate that there are requests and confirmations that go back and forth between the involving modules, we put two 2-way block arrows to express this compensation workflow in Figure 3.

4) PROGRAMMATIC ENFORCEMENT

A smart contract, also known as a chaincode in Hyperledger, is a computer program that defines the rules and conditions for executing transactions. In this subsection, we walk through such a chaincode to illustrate why the enforcement of penalty-aware SLAs is attainable owing to blockchain's smart contracts. In other words, we propose a programmatic enforcement of the SLAs. The chaincode is written rather straightforwardly in accordance with the general structure of the penalty rules expressed in deontic logic. Nevertheless, to fully implement a penalty rule, we programmatically rely on modules mounted on the provider's server that are external to the blockchain. These modules, which are replaceable on the provider's side, realize reparation actions of the penalty rule in question on behalf of the service provider whose rule is being fired.

Let us turn our attention to Example 1 again while walking through the chaincode given in Listing 1. We declare an entry-point function named `callPenaltyRule`, for which the arguments are wrapped in an array of strings called `attributes`. Let us recall that a typical penalty rule represented in deontic logic contains a series of antecedents and a consequent. To be able to programmatically enforce the said penalty rule, we first fetch the list of external APIs corresponding to these deontic components (at line 12 of Listing 1) and store them in an ordered array named `apiList` by invoking a user-defined function called `getExternalAPIList`. The code comments that span lines 8–10 exemplify what the array actually holds when running the chaincode to enforce the penalty rules described in Example 1. Note that the chaincode presented here is generic in the sense that it is programmed to trigger any compensation modules that have been readily deployed at the service provider involved in the contract (identified by `ContractID`). The function then attempts to submit a request to an API-mediated module[17] in `apiList` (at line 16 of Listing 1) until the compensation is confirmed. If an error occurs during the request owing to, for instance, technical failure or the service provider's inability to compensate (for example, a situation where a hotel has no compensation options to make up for a tourist who checked in a non-oceanfront room). If no post requests are confirmed at all, which signifies that the penalty rule in question has not been adhered to, `status` still holds the boolean value of `false` after the loop is terminated. when they published their penalty rules. At line 26 of Listing 1, function `calRuleAbidingRate` is utilized to recalculate the rule-abiding rates of the involved service provider (identified by `ProviderID`) to write down whether the penalty rule in

```
1  func (t *CPRChaincode) callPenaltyRule(stub
   ↪ shim.ChaincodeStubInterface,
2  attributes []string) cp.Response
3  {
4      comp, err := strconv.Atoi(attributes[0])
5      ContractID := attributes[1]
6      ProviderID := attributes[2]
7      RuleID     := attributes[3]
8      // the following array captures a list of links pointing to external
   ↪   APIs,
9      // for example, in the case of the aforementioned accommodation
   ↪   service,
10     // {"https://provider.com/api/alt-room",
   ↪   "https://provider.com/api/free-spa",
   ↪   "https://provider.com/api/discount"}, this array is fetched from the
   ↪   ledger
11
12     apiList := getExternalAPIList (ContractID,
   ↪   RuleID)
13     var resp * http.Response
14     var api string
15     var status = false
16     for comp < len(apiList)
17     {
18         api = apiList[comp]
19         resp, err = http.Post(api,
   ↪   "application/json",
   ↪   bytes.NewBuffer([]byte(ContractID)))
20         if err != nil
21             comp++
22         else {
23             status = true
24             break}
25     }
26     calRuleAbidingRate(ProviderID, RuleID,
   ↪   status)
27     return shim.Success(nil)
28 }
```

LISTING 1. This chaincode on Hyperledger Fabric implements a generic penalty rule, which might trigger human-in-the-loop actions via API calls.

[17] All the service providers registered in PenChain are expected to make their compensation modules accessible via APIs. A compensation module might be fully automatic (e.g., a discount voucher issued for the subsequent service transaction) or operated by humans (e.g., by arranging an alternative room). The character-wise value that represents the API at which this module is invoked from the chaincode is safely and persistently stored in the ledger.

question (identified by `RuleID`) has been respected or not (indicated by `status`).

It is worth mentioning how the rule-abiding rate for an SLA is algorithmically calculated and updated down the ledgers. Upon the activation of a penalty rule, a dedicated smart contract in the underlying blockchain of PenChain immediately increments the total count of rule firings by one, regardless of the final outcome of this rule that is determined later on. As the penalty rule approaches its completion, if the involved customer remains uncompensated after having tried all compensation actions specified in the penalty rule, the same smart contract increments the count of SLA breaches associated with the respective service offering. Finally, a re-calculation of the rule-abiding rate is performed according to Definition 3.

F. RANKING ALGORITHMS

We propose algorithms needed for ranking a set of service offerings, allowing the search result to be sorted in that best matches the customers' preferences. Service offerings are mixed up in a search result – some of them might be of the same function but are offered by different providers. Moreover, an element in this list could be either an atomic service or a service aggregation. The proposed algorithms materialize our groundwork on the semiring presented in the previous subsections. The main idea is to devise Algorithm 2 as a comparator function that will be fed into a sorting function (Algorithm 1).

Algorithm 1 Sorting a Search Result by the SLAs Taking Into Account the Customer's Preference

```
Data: $L_{serv}$: a search result that needs to be sorted;
begin
    if the service customer has not interacted with PenChain yet then
        $L_{objective}$ ← the PenChain's default list of SLA objectives;
    else
        $L_{objective}$ ← customized list of SLA objectives;
        Sort $L_{objective}$ in descending order by the number of views received for each objective;
    end
    foreach $serv \in L_{serv}$ do
        if serv is a service aggregation that comes with the SLA then
            Compute the SLA of serv in accordance with (1) in Definition 5;
        end
    end
    sort $L_{serv}$ using the comparator function defined in Algorithm 2 and $L_{objective}$;
end
```

L_{serv} in Algorithm 1 denotes a set of service offerings listed in a search result. We recall that the SLA objectives are flexibly sequenced according to the customer's preference. $L_{objective}$ is an ordered list that determines the relative importance of constituent objectives in the SLA through the lens of the service customer. For example, if the rule-abiding rate matters most to a certain customer, the corresponding item is placed first on her $L_{objective}$. This list needs to be re-sorted in descending order when PenChain records a substantial amount of additional view counts, suggesting that the service customer has re-prioritized the SLA constituent objectives in their mind. The algorithm is designed to work with an unlimited number of constituents, although in this work we assume that $L_{objective}$ is a list of three. The algorithm addresses two possibilities: a newly registered customer who has yet to interact with PenChain (e.g., to submit a search request, to view different service offerings listed in a search result) and a returned customer whose viewing history is kept track of her profile. Note that this algorithm treats all elements of L_{serv} in the same way, regardless of whether they are atomic or aggregated.

Algorithm 2 acts as a comparator function that determines whether a given service (sla_1) should be placed above or below another (sla_2) in a search result. The list represented by $L_{objective}$ is literally passed from Algorithm 1, telling which SLA objective is the foremost criterion, the second most important, and so on. To make this algorithm generic, in the main loop, we iterate through all elements of $L_{objective}$, not necessarily confined to the three SLA objectives as we make the case in this paper. We assume that each objective is associated with an optimization operation, either maximization or minimization, advising us to retain the order (e.g., comparing the satisfaction) or reverse it (e.g., comparing the costs) when comparing. The loop continues down the list of $L_{objective}$ until we can figure out whether sla_1 is preferred over sla_2 or vice versa. This algorithm returns zero if it passes all the objectives in $L_{objective}$ without decisively determining the order between sla_1 and sla_2, suggesting that these two services should be placed in the same position within the search result.

G. MULTI-CRITERIA SEARCH

In this subsection, we showcase an advanced search feature of PenChain that allows service consumers to search for and locate the right service offering. To enable this search feature, we make sure that the service offerings in PenChain are fully populated with details including the service location[18] when the providers publish their offerings in the service registry.

As illustrated in Figure 4, a text box with auto-complete in the right pane allows the geographical location of the service to be entered. Suggestions of the location in this text box are sourced from Bing Maps for the customer's convenience.

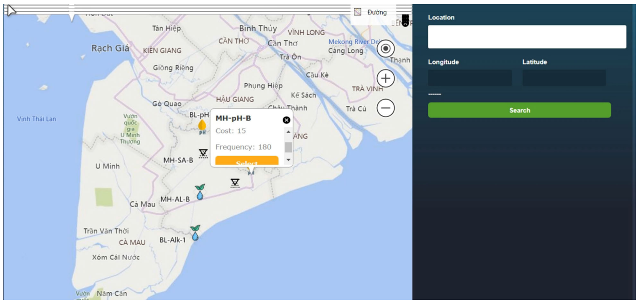
[18] One should not get confused between the location where the service in question is actually operated and the postal address registered in the profile of the organization or agent who provides this service. The former usually points to where IoT sensors are installed for data acquisition, while the latter roughly identifies the geographic site over which the provider operates its services.

Algorithm 2 Comparing a Pair of Service Offerings by Their SLAs

```
Input: $sla_1$, $sla_2$: A pair of SLAs to be compared;
       $L_{objective}$: a list of objectives sorted in
       descending order by the customer's view
       count;
Output: +1 if $sla_1$ is better than $sla_2$, 0 if equal, $-1$ if
        worse;
begin
    foreach $f_{obj} \in L_{objective}$ do
        if MAX is the optimization function for $f_{obj}$
          then
            if $sla_1[f_{obj}]$ is greater than $sla_2[f_{obj}]$ then
                return 1;
            else
                if $sla_1[f_{obj}]$ is less than $sla_2[f_{obj}]$ then
                    return $-1$;
                else
                    continue;
                end
            end
        else
            if $sla_1[f_{obj}]$ is less than $sla_2[f_{obj}]$ then
                return 1;
            else
                if $sla_1[f_{obj}]$ is greater than $sla_2[f_{obj}]$
                  then
                    return $-1$;
                else
                    continue;
                end
            end
        end
    end
    return 0;
end
```

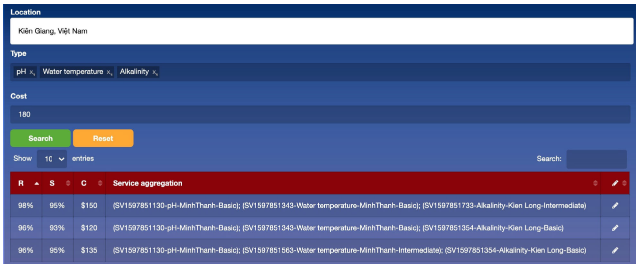
The search result is displayed live on the map in the left pane, with service offerings uniformly shown under the same teardrop-shaped icon but at various locations. Each of them comes with a tooltip that pops up when the customer clicks on it, revealing its name and further details of the service being represented. To give an example, on the map, there is a teardrop-shaped icon that is yellow in color representing a data service that provides periodic readings of the water's pH level in a farming site. The customer may find other widgets in the tooltip useful for viewing the SLA-bound historical values of the service she is looking at, which is instrumental in improving the customer's comprehension of the rating of its service provider.

An alternative user interface to our multi-criteria search is depicted in Figure 5. The search dialog here provides several text boxes that allow the customer to enter the service details:



MH-pH-B
Cost: 15
Frequency: 180
Select

FIGURE 4. Service offerings in a search result are visually shown on Bing Maps, offering the service customer a spatial sense when examining them prior to a possible service binding.



Location
Kiên Giang, Việt Nam

Type
pH × Water temperature × Alkalinity ×

Cost
180

Search Reset

Show 10 entries   Search:

R	S	C	Service aggregation	
98%	95%	$150	(SV1597851130-pH-MinhThanh-Basic); (SV1597851343-Water temperature-MinhThanh-Basic); (SV1597851733-Alkalinity-Kien Long-Intermediate)	✎
96%	93%	$120	(SV1597851130-pH-MinhThanh-Basic); (SV1597851343-Water temperature-MinhThanh-Basic); (SV1597851354-Alkalinity-Kien Long-Basic)	✎
96%	95%	$135	(SV1597851130-pH-MinhThanh-Basic); (SV1597851563-Water temperature-MinhThanh-Intermediate); (SV1597851354-Alkalinity-Kien Long-Basic)	✎

FIGURE 5. A tabular view of a search result that allows the service customer to comprehensively examine service offerings listed in a search result before choosing one of them.

keywords, expected costs, and, of course, the service location. The search result is then shown in a tabular form further below in the figure, which is sorted according to the core ranking algorithms of PenChain (see Subsection IV-F). The text box where the customer enters the location of the service in Figure 5 is enhanced with auto-complete data fed from Bing Maps just like its counterpart in the search dialog of Figure 4.

The service customer may toggle the search mode at any time as she interacts with PenChain. In both modes, she should be able to pick a service offering from the search result to proceed further (e.g., adding it to a service shopping cart or checking out to enter a service binding contract).

V. SCENARIO-BASED EVALUATION (I): CASE OF HIGH-PRECISION AGRICULTURE

Data exchange as a service is a common saying in modern smart farming. For instance, HARA[19] is a blockchain-based platform for data exchange in the food and agriculture sector. Agdatahub[20] operates trusted platforms for secure data exchange. Centre for Data Sharing[21] makes a prominent use case for data sharing in agriculture.

In the Mekong Delta region of Vietnam, a community of research institutions, corporations, and agricultural extension organizations alike forge digital solutions to offset the effect of climate change on the aquatic environment. They need a service provisioning platform that allows both in-house and cloud-based IoT data providers to provision real-time farming

[19] https://www.hara.ag
[20] https://agdatahub.eu
[21] https://eudatasharing.eu/

datasets. Central to this platform is a multi-criteria search function that allows a location-based, schema-specific search request (e.g., looking for all service offerings that report on the pH levels of all farming sites located in a given province of Vietnam) to be submitted, for which the search result shall be sorted – the closer to the top of this search result a provider is placed, the higher reputation score it has from the requesting end-user's perspective.

As more and more data service offerings become available to these end-users, PenChain's service registry and discovery would let them gain access to the right offerings that best match their intention. Penalty rules captured in the SLA of an IoT data service may state what compensation the customer is entitled to (e.g., a discount voucher, a data provisioning package for free), but should not be unreasonably one-sided.

We proceed with the case as follows. In Subsection V-A, we describe the need to enforce the SLAs of data provisioning in smart shrimp farming. We analyze the imperfections of the supporting IoT sensor technology in Subsection V-B. In Subsection V-C, we present an uptake of IoT data services, where our goal is to demonstrate the relevance of our PenChain framework to the enforcement of penalty-aware SLAs in a specific business domain as a case study. Separately, in Subsection V-F and Subsection V-G, we discuss the scalability and the trustworthiness of PenChain, respectively. To make this section as a whole self-contained, we explain the underpinning technologies and their interplay in Subsection V-D.

A. CHALLENGES OF THE CONTEXT

The Lower Mekong River Basin in Vietnam has a unique physical terrain with interlaced ditches and canals. The region's terrain is well suited for farming and fishing, making agriculture a significant part of its economy. This business sector comprises corporations, private shrimp farming families, and many other IoT players. In light of precision agriculture, the productivity of shrimp farming depends on water quality, fluctuation in water temperature, stability of dissolved oxygen, etc., necessitating the monitoring and automated regulation of these parameters. Unfortunately, data gathering is an error-prone and costly technical process, particularly when collecting datasets on the sparsely spread shrimp ponds in the region. With the ubiquity of the Internet of Things, agricultural professionals and data engineers expect to benefit from next-generation data provisioning and service engineering.

B. IMPERFECTION OF THE AVAILABLE TECHNOLOGY

As more and more data service offerings become available to end users, a novel approach to service registry and discovery should be in place to let them gain access to the right offerings that best match their intention. In PenChain for precision agriculture, we distinguish between a cloud-based IoT agency and an in-house IoT station, even though they all utilize sensors to collect data [24], [159]. The former operates its sensors over a relatively large area, while the latter usually has sensors installed in a rather small area (typically confined to its farming site). They both make real-time farming datasets readily obtainable for the end-users. To facilitate data delivery, the cloud-based IoT agency relies on a computing cloud to perform data conditioning and offer data analytics. The in-house IoT station is equipped with limited computational power and almost only transmits instantaneous readings without any fancy analytics tools. Regarding business goals, cloud-based IoT agencies provide farm data primarily to earn money. The in-house IoT data providers may use the data they collect for (a) gaining insight into, or at least monitoring their own farming activities; (b) making the datasets collected over their farming sites accessible to the end-users and other in-house providers at a significantly lower price compared to that of the datasets provided by the cloud-based IoT agencies.

Introducing modern IoT facilities to agriculture would create a marketplace for data exchange and an ecosystem that fosters the collaboration between farmers, agricultural engineers, data experts, research institutes on precision agriculture, etc. PenChain could be tailored to this market following the so-called data-as-a-service models.

C. THE PROPOSED SOLUTION

In Table 4 we populate the data services offered in PenChain. Each service is uniquely identified by a label (so_1, so_2, etc.) and is referred to by a short name. The service providers are anonymously denoted as P^1_{cloud}, $P^2_{in-house}$, $P^3_{in-house}$ and P^4_{cloud}. To the right of Table 4, we describe the contractually formulated, multilevel commitments between these service providers and end users. We show no more than three levels for each of these service agreements to keep it simple. The basic, intermediate, and advanced levels are with a logical progression to monthly costs and data frequency. Geographically speaking, P^1_{cloud} is located in Giang Thanh, $P^2_{in-house}$ in Kien Luong, $P^3_{in-house}$ in Hon Dat, P^4_{cloud} in Ha Tien. These proper names refer to the rural districts of Kien Giang Province in the Mekong Delta of Vietnam.

The service providers $P^2_{in-house}$ and $P^3_{in-house}$ are farmers who deploy IoT sensors on their farms located in one of the above-mentioned rural districts. They invest in IoT-enabled equipment to monitor their crop and optimize their agricultural production. They rely on on-site sensors to provide live readings of pH and alkalinity in a localized area. Due to the lack of computing power and storage, their readings in the past might not be available for analytics. In contrast, the other two providers – denoted as P^1_{cloud} and P^4_{cloud}, harness cloud computing to provide the end-users with enriched data (e.g., spatio-temporal extrapolation) about water temperature and water pollution over a relatively large farming site, which they do not own. Hi-tech agencies such as P^1_{cloud} and P^4_{cloud} obtain the appropriate permission from the farm owners to operate their IoT devices, networking infrastructure, and cloud servers in the field crops of farmers.

TABLE 4. PenChain facilitates the provisioning of IoT data services in the Mekong Delta.

Label	Services	Provider	Multilevel	Frequency	Cost
so_1	Water temperature	P^1_{cloud}	Intermediate	Every 60 secs	\$5
			Advanced	Every 36 secs	\$10
so_2	Water pollution	P^1_{cloud}	Basic	Every 18 secs	\$10
			Intermediate	Every 36 secs	\$20
			Advanced	Every 25 secs	\$25
so_3	pH	$P^2_{in-house}$	Basic	Every 35 secs	\$3
so_4	Alkalinity	$P^3_{in-house}$	Basic	Every 43 secs	\$10
so_5	Salinity	P^4_{cloud}	Intermediate	Every 40 secs	\$20
			Advanced	Every 25 secs	\$30
so_6	pH	P^4_{cloud}	Intermediate	Every 20 secs	\$10

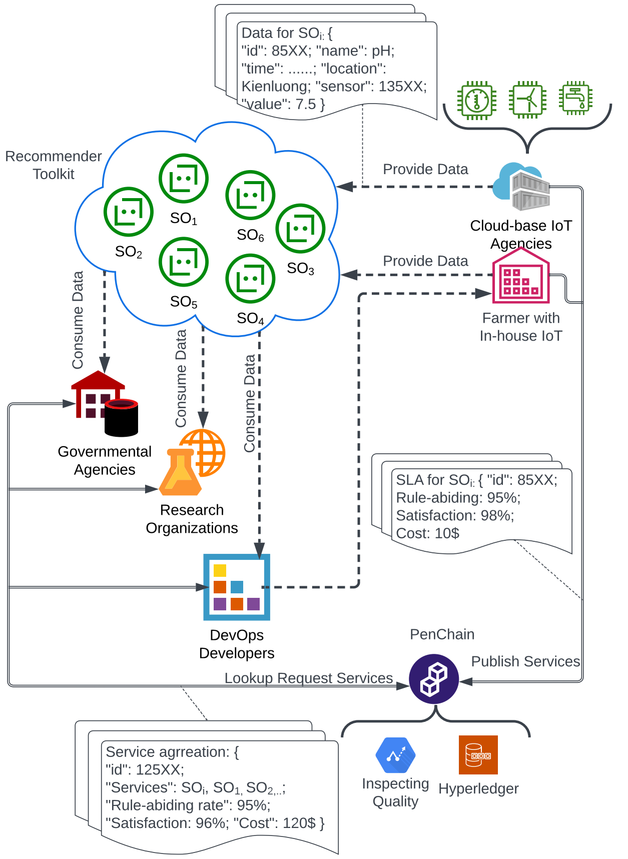


FIGURE 6. An aggregator business model for data services in PenChain.

Their business is to collect large datasets of agricultural production and disseminate them to earn money.

As shown in Figure 6, PenChain allows end-users to request data services offered by the aforementioned service providers. End-users in this scenario encompass DevOps developers, research organizations, and government agencies. They make use of the service offerings listed in Table 4 to compose service-oriented applications, for example, by combining services so_1 and so_2. Thanks to PenChain, the research organizations can obtain agricultural data for scientific purposes by invoking, e.g., services so_3 and so_6. Government agencies rely on numerous data packages obtained through services so_4 and so_5 to fine-tune the agricultural extension for fisheries development in the area. To give a sense of the ranking of the services, a research laboratory interested in pH readings would prefer so_6 to so_3 because the former comes with an agreement level that is favorable to the latter.

Over the months, the DevOps developers who compose a service-oriented application for water monitoring will ask for additional data services that are not listed in Table 6. One way to meet this demand is to aggregate functionally related data services (e.g., both so_1 and so_2 provide data readings in the water) to provision more sophisticated data packages for a farming site (e.g., reporting on the water temperature and water conductivity of the same shrimp pond). Speaking of service-oriented programming, so_1 and so_2 together as a service aggregation would come in handy programmatically. The aggregate of so_1 and so_2 involves mixing their water-related data and results in their multilevel SLAs being compiled and rationalized.

D. UNDERPINNING TECHNOLOGIES

In this subsection, we discuss our difficulties in adopting blockchain technologies and DevOps tools in enforcing any penalty-aware SLAs that are associated with data provisioning. To understand how PenChain facilitates data exchange in smart farming and guarantees that the agreements stated for data provisioning are followed through on, let us look under the hood.

1) MICROSOFT AZURE

We emulate the applications in use for both service providers and customers on Azure.[22] This platform comes with a built-in monitoring capability for Web services, which we found useful for stress testing and deploying concurrent emulations. As for data handling, PenChain utilizes cloud-based relational database services provided by Azure SQL that perform relational queries, searching and data synchronization.

2) BLOCKCHAIN

The permissionless mode of public blockchains (e.g. Ethereum) is at the expense of privacy, performance, and scalability. A private blockchain eliminates this problem by running as a permissioned blockchain with flexible, extensible consensus and access control techniques. As such, public blockchain networks are not suitable for launching the decentralized program for a business solution [160]. Businesses can collaborate with distributed ledger developers to launch their business-to-business and cross-industry applications

[22] Microsoft Azure is a cloud computing platform and web portal that lets software developers access and manage cloud resources and services including Web services and data analytics https://azure.microsoft.com

using a private blockchain. In line with this approach, we chose Hyperledger Fabric for its modular architecture, which facilitates future scalability. Most notably, the latter versions of Hyperledger Fabric come with Byzantine Fault Tolerance consensus mechanism to ensure system integrity by mitigating the influence of malicious nodes and ensuring continued operation despite failures or malicious actions.

Speaking of the hardware configuration and operating system of our cloud server that hosts the blockchain network. It comprises a 4-core CPU and 6GB of memory and has 64-bit Ubuntu Linux 18.04 installed. We deploy a version later than 2.0 of Hyperledger Fabric featuring a channel and three nodes as shown in Figure 7. The service providers in PenChain get represented by two of the nodes in this channel, whereas the service customers vote for the third node. The rationale behind having more representatives of the providers in our private blockchain network is to acknowledge that the providers who join PenChain are likely to evolve into organizations later in their business cycle while the presence of the customers is confirmed. We establish three peers within each node, namely `Anchor` (enabling self-discovery and cross-node communication), `Endorser` (generating signatures to validate transactions), and `Committing` (creating ledger-based transaction storage). Note that `Ordering` arranges and constructs the transaction blocks following the Byzantine Fault Tolerance protocols.

In Figure 7, service transactions originating from the *Logging* module in the business layer are dispatched to *Endorsers* via *Fabric-Client* in Step ①. Moving to Step ②, the *Endorser* peers within each node jointly validate, sign and transmit the resulting transaction back to *Fabric-Client*. During Step ③, *Fabric-Client* transmits these endorsed transactions to *Ordering* for broadcasting. Lastly, in Step ④, *Ordering* distributes the transaction block to the *Committing* peers within each node where ledger-based copies of the transaction block are serialized in a non-undoable way.

3) PROGRAMMING THE SMART CONTRACTS

Smart contracts written and deployed on PenChain's blockchain serve two purposes. First, they recalculate the accumulated rating of a data service provider upon receiving a confirmation of the quality of the data exchanged or a breach of the SLA contract. A special module of PenChain that automatically assesses the quality of exchanged data will record any irregularity of data exchange caused by sensor malfunctions, connection timeout, data handling problems, etc. Such an issue would trigger a smart contract that updates the rating of the service provider involved in this data transaction. Second, special smart contracts in PenChain – illustrated in Listing 1, are programmed to fire workflow actions scripted in a penalty rule to trigger the compensation workflow upon detecting an SLA violation.

E. SERVICE PUBLISHING, DISCOVERY AND BINDING

Service-oriented computing involves a wide range of phases, with service publishing, service discovery and service binding being the major phases. We point out to what extent PenChain would address these engineering phases. To make the case for PenChain's usefulness for service provisioning, we set up eight client apps on a cloud that is separated from our Microsoft Azure cloud where PenChain was deployed.

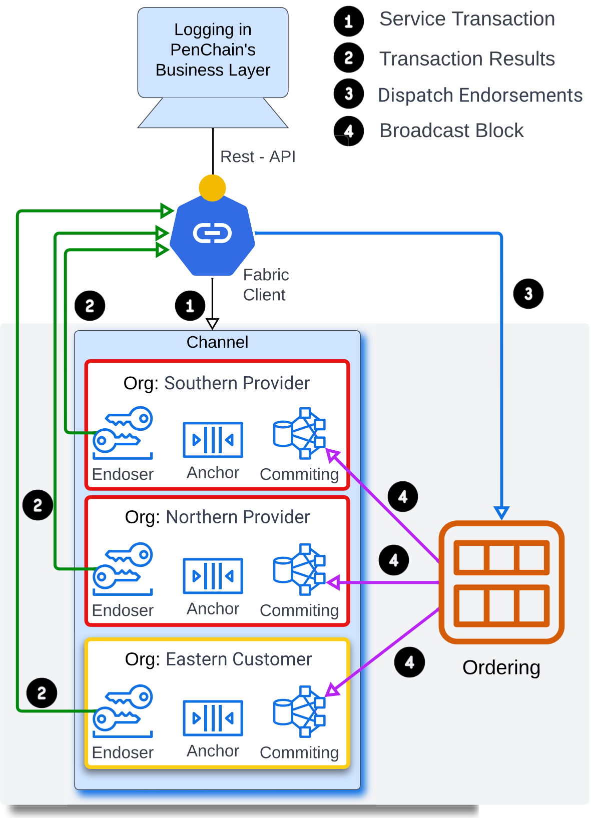


FIGURE 7. The endorsement, ordering and commitment in PenChain's underlying private blockchain network.

- *Service publishing*. We follow the common architecture of service-oriented computing by which penalty-aware services should be made available to registered customers in PenChain. Any client application (i.e., software that allows a customer to consume services in mutually trusted ways via PenChain) linked to our platform shall access this service registry without any restriction. Our SLA-oriented service aggregation technique helps enrich the search space in PenChain, allowing more service offerings to be listed in a search query. We require all providers to publicize their penalty rules when joining our platform. Unfortunately, it is a shortcoming of PenChain that we do not fully establish a technical workflow detailing how to publish penalty-aware services.
- *Service discovery*. One of the chief functions of PenChain is reputation-driven service lookup, i.e. how service offerings are sorted in a search result depends in part on the reputation of their provider. As can be seen in Figure 4 and Figure 5, this function has an intuitive user interface where customers can view their

search results as: (a) a set of services matching the search criteria, which are sorted by their SLA; (b) a filtered list of service offerings displayed on Bing Maps. The former helps the customer comprehend her search result by displaying all service offerings together with service aggregations, while the latter is more intuitive and gives a spatial sense of the service lookup mechanism in PenChain.

- *Service binding*. PenChain was built with the modality in mind, by which modular clusters will communicate with each other through APIs. A unit that assesses customer satisfaction is located in the same cloud as the "main" of PenChain, which will be invoked in every service transaction[23] via a designated API. The "main" of PenChain then triggers a smart contract to update the provider's rating according to the confirmation returned from the assessment unit. As such, PenChain consists of loosely coupled units that are considered to be replaceable. To this end, service binding is done with minimum intervention from PenChain, letting the service transactions run their course. Yet PenChain is able to enforce publicized penalty rules (and update the provider's rating accordingly) associated with a service transaction if the situation is not going in an unintended workflow.

F. SCALABILITY OF PENCHAIN

Whether PenChain scales up to accommodate the increasingly high number of service providers and service consumers chiefly depends on two factors: (a) the ability of its cloud-based applications and databases to reliably and elastically handle a growing number of service transactions; (b) the capability of the underlying private blockchain to respond to an increasingly large volume of read/write operations for the same reason. In this analysis, we investigate the second factor, acknowledging that the first factor has been thoroughly studied from the point of view of high-performance computing and microservices architecture.

Suppose that PenChain harbors n service providers, each of whom will at least launch m data service offerings. There may be several variants of a service offering that differ in terms of SLA. Let k be the number of times a service offering is successfully bound to some customer (the more customers registered who transact via PenChain on a regular basis, the higher k). Interviewing local farmers at farming sites where we obtained experimental datasets suggests that the frequency of data exchange in precision agriculture shall be no more than four times per day. Therefore, the total number of service transactions per day can be written as, $G = n \times m \times k \times 4$, or $n \times m \times k \div 360$ transactions per minute. Let us analyze the load sustained by PenChain in a few typical schemes as follows, which are colloquially depicted in Figure 8.

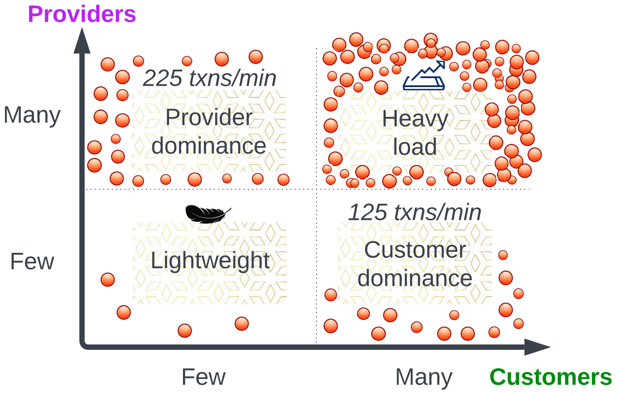


FIGURE 8. We offer an analysis on the scalability of PenChain in terms of the overall responsiveness of its underlying blockchain. The heatmap-like visualization in each of the four areas represents the burden on the underlying blockchain.

- *Provider dominance*. Suppose that there are a large number of ($n = 1000$) providers registered in PenChain to do the business of agriculture data provisioning. They each exhibit nine data service offerings ($m = 9$). If a service offering is bound to at least nine customers ($k = 9$), PenChain processes 225 transactions per minute on average.
- *Customer dominance*. PenChain attracts a large number of customers that frequently request data service offerings from a relatively small number of providers ($n = 100$). Each provider offers nine service offerings ($m = 9$), each of which is bound to 50 customers ($k = 50$). PenChain processes around 125 transactions per minute in this scheme.
- *Balanced & lightweight*. This scheme is characterized by a relatively low number of providers and customers who transact on a regular basis. For example, there are only five providers ($n = 5$) each publishing 15 data services ($m = 15$). Service offerings are binding to at least three customers ($k = 3$). PenChain sustains a light load in this scheme, precisely requiring only one transaction per minute.
- *Heavy load*. PenChain sustains a high burden of transactions in this scheme, which is in stark contrast to the previous scheme, when both the service providers and customers dramatically increase in number.

In the first two schemas, namely provider dominance and customer dominance, regardless of what clouds the data services and customer applications shall be deployed to, we need to upgrade the server that hosts our Hyperledger Fabric, which is currently the bottleneck of PenChain, as the private blockchain database is notoriously unresponsive under high load. We note that the cloud database counterpart MS Azure is rather elastic, requiring a hashing mechanism that links a transaction written down to the blockchain

[23] The invocation will be redirected to another unit that assesses the practicality of the service consumed when changing the service domain, e.g., switching from precision agriculture to air quality monitoring.

database to its extended log entry in the cloud database. PenChain would scale up in the third scheme.

As analyzed in Section IV, PenChain consists of the following loosely coupled components: data services deployed either on the provider's cloud or MS Azure, customer applications, the "main" and the underlying blockchain. In all scenarios, the scalability of PenChain depends on computing resources allocated to the providers' cloud and the Azure cloud for its "main" as well as the responsiveness of Azure SQL database and Hyperledger Fabric in use.

G. ENHANCING DATA SECURITY AND MINIMIZING BIASEDNESS

Given the dynamic and evolving landscape of security on blockchain-based platforms, we recognize that a dedicated analysis would illuminate potential vulnerabilities, threats, and the overall resilience of the PenChain platform against malicious activities [161]. A comprehensive security evaluation necessitates an exhaustive exploration of attack vectors, consensus mechanisms, smart contract vulnerabilities, and potential data breaches [162]. In our research, we focus on data security, with the aim of improving trustworthiness and minimizing unbiasedness when it comes to evaluating penalty-aware SLAs. Specifically, PenChain was designed and engineered with the following in mind: (i) the on-chain database is wired to the off-chain counterpart; (ii) programmatic enforcement of penalty rules and algorithmic calculation of the rule-abiding rate; (iii) advocating the use of scanning devices (e.g., a QR code reader) and dedicated software plugins (e.g., detecting flawed or useless data cells in data provisioning) to digitize the satisfaction rate whenever possible. In this subsection, we elaborate (i) for the sake of data security while stressing that (ii) and (iii) together make the evaluation of the SLAs more evidence-based and unbiased, as opposed to notoriously unverifiable feedback (baseless feedback in the worst cases) left by humans on today's mainstream intermediary websites.

To enhance data security in PenChain, we decouple a service transaction into a transaction block stored in Hyperledger (on-chain data) and a transaction log recorded in Azure SQL (off-chain data). The on-chain block is linked to the off-chain counterpart via a hash function in the family of cryptographic functions called Secure Hash Algorithms. As illustrated in Figure 9, the hash function transforms the off-chain data record that captures the transaction details[24] of a service transaction into a hash string before securely embedding it into the corresponding block of Hyperledger. This decoupling enhances the data security as follows: (a) as the hash code is placed on the blockchain, it possesses an immutable characteristic – any minor 1-bit change results in a wholly distinct hash code, effectively shielding the off-chain log from unauthorized alterations; (b) unlike on-chain blocks that are accessible to all peers in the same blockchain channel, transaction details stored in an off-chain record shall be restricted to authorized peers thanks to an off-chain access control layer.

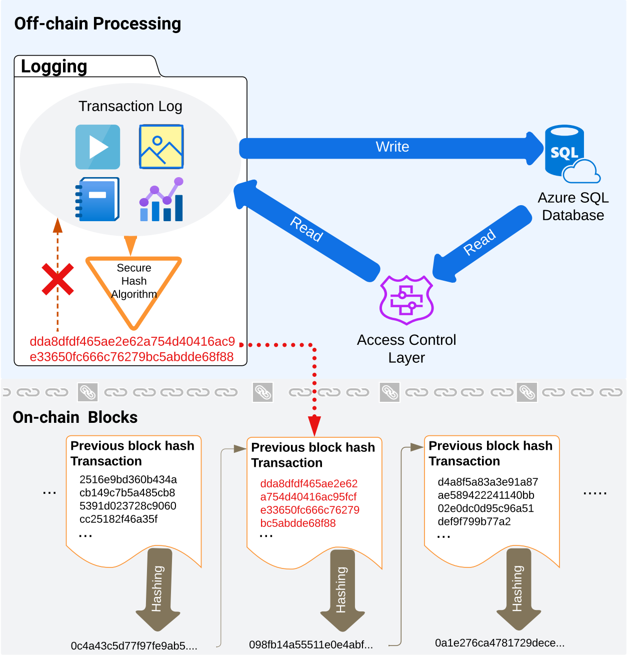


FIGURE 9. Placing the transaction log in an off-chain database and having this log entry wired to the corresponding on-chain block using a hash function.

VI. SCENARIO-BASED EVALUATION (II): CASE OF THE AUTOMOBILE MANUFACTURING INDUSTRY

In this section, we discuss the usefulness of PenChain for the automotive industry (Subsection VI-A) and demonstrate its usability by elaborating an exemplary workflow (Subsection VI-B).

A. USEFULNESS OF PENCHAIN FOR THE AUTOMOBILES MANUFACTURING INDUSTRY

An automotive parts supplier is a firm that supplies different automotive components, parts, and systems to vehicle manufacturers. Typically, these suppliers specialize in the production of certain components or systems, including engines, gearboxes, braking systems, steering systems, electrical components, and more. Automobile manufacturers are highly dependent on their suppliers to provide high-quality parts and components that satisfy their precise design and performance specifications. They collaborate closely with their suppliers to guarantee that the components are supplied on time, adhere to stringent quality control requirements, and are cost-effective. Typically, the relationship between automakers and their suppliers is long-term, and both sides collaborate to enhance their products and services. To stay competitive in the automotive sector, suppliers must be able to adapt to shifting consumer expectations and industry

[24] In data provisioning, the execution log of a service transaction includes the exchanged data together with the IP address and port number through which the data exchange took place. In service domains other than this, multimedia data, such as footage videos and images, might be added as digital evidences to a transaction log.

trends. Overall, the job of an auto parts supplier is important to the success of an automaker. Without dependable and high-quality parts suppliers, automakers would struggle to construct safe, efficient, and cost-effective vehicles.

The key SLAs between an automobile parts supplier and the manufacturer typically include the following:

- *Quality standards*: The supplier must provide high-quality parts that meet the manufacturer's quality standards, and the manufacturer may specify particular quality control processes that the supplier must follow. The SLA may also include requirements for testing and inspecting automotive parts before delivery. Compatible parts of the same or higher quality could be sourced from the supplier's competitors should the parts originally shipped to the manufacturer fall short of the quality thresholds contractually defined in the SLA. The supplier will face serious consequences when they run out of options to overcome unacceptably low-quality parts.
- *Delivery time*: The supplier must deliver the parts to the manufacturer within the specified time frame, which may include deadlines for each shipment and overall delivery times for the entire order. Some of these deadlines might be relaxed to give the supplier another chance if it missed the deadline due to unexpected circumstances. Failure to respect an extended deadline leads to a heavy deduction in the payable amount for those parts.
- *Order accuracy*: The supplier must ensure that the parts ordered by the manufacturer are accurate in terms of quantity, specifications, and other requirements. The SLA may specify penalties for errors or inaccuracies in orders.
- *Communication*: The supplier must maintain open communication with the manufacturer to provide regular updates on the status of the order, including any potential delays or issues that may arise.
- *Cost*: The supplier must provide competitive pricing for the parts and related services, such as shipping and handling. The SLA may include provisions for pricing adjustments based on changes in the market or changes in the manufacturer's requirements.
- *Intellectual property*: The supplier must respect the manufacturer's intellectual property rights and maintain the confidentiality of any proprietary information or trade secrets. There are no viable reparation actions to avert a breach of this SLA – heavy penalties will be applied, e.g. an irreversible termination of the partnership.

These SLAs help ensure that the automobile parts supplier and the manufacturer have a mutually beneficial relationship that promotes the high-quality, timely and cost-effective delivery of automotive components. By establishing clear expectations and requirements, both parties can work effectively together to achieve their goals. Blockchain technology has the potential to provide significant benefits for the relationship between automobile parts suppliers and manufacturers. Here are some potential use cases for blockchains in this context.

- *Supply chain management*: A blockchain can provide an immutable and transparent ledger for tracking the entire automotive component supply chain, from raw materials to final products. This can help increase visibility and accountability in the supply chain, as well as improve traceability and reduce the risk of counterfeiting or fraud.
- *Quality control*: A blockchain can be used to record and verify quality control data for each automotive component produced by the supplier. This can help ensure that the manufacturer receives high-quality parts that meet their specifications and standards.
- *Payment and settlement*: A blockchain can provide a secure and efficient platform for managing payments and settlements between the supplier and the manufacturer. By using smart contracts, payments can be automatically triggered and settled based on predefined conditions and criteria, reducing the need for intermediaries and minimizing the risk of errors or disputes.
- *Intellectual property management*: A blockchain can help protect the intellectual property rights of both the supplier and the manufacturer by providing a secure and transparent platform for managing and verifying ownership and usage rights.

Blockchains can help improve the efficiency, transparency, and security of the relationship between automobile parts suppliers and manufacturers. By leveraging the unique features of blockchain technology, these stakeholders can work together more effectively to create high-quality, cost-effective automotive components that meet the needs of consumers.

B. EXEMPLARY UTILIZATION OF PENCHAIN IN AN AUTOMOBILE MANUFACTURING WORKFLOW

Suppose Automobiles–X is a manufacturer of automobiles that sources its parts from various suppliers. One of its key suppliers, XYZ parts, provides them with engine components that are critical to the production of their cars. To ensure the quality of the parts and the shipment standards of suppliers Automobiles-X has two SLAs: a manufacturer SLA and a shipment SLA. Under the SLAs, XYZ Parts must ensure that the engine components they supply meet the quality standards and shipment standards, as mentioned above, set by Automobiles-X. To verify the quality of the parts and shipment, standard Automobiles-X conducts regular inspections and tests of the components upon receipt. If the parts fail to meet the quality and shipment standards outlined in the SLAs after having tried all available repair actions, Automobiles-X finally fires a penalty consequently. For example, if the parts are found to have defects, the automobile manufacturer can deduct a percentage of the payment for

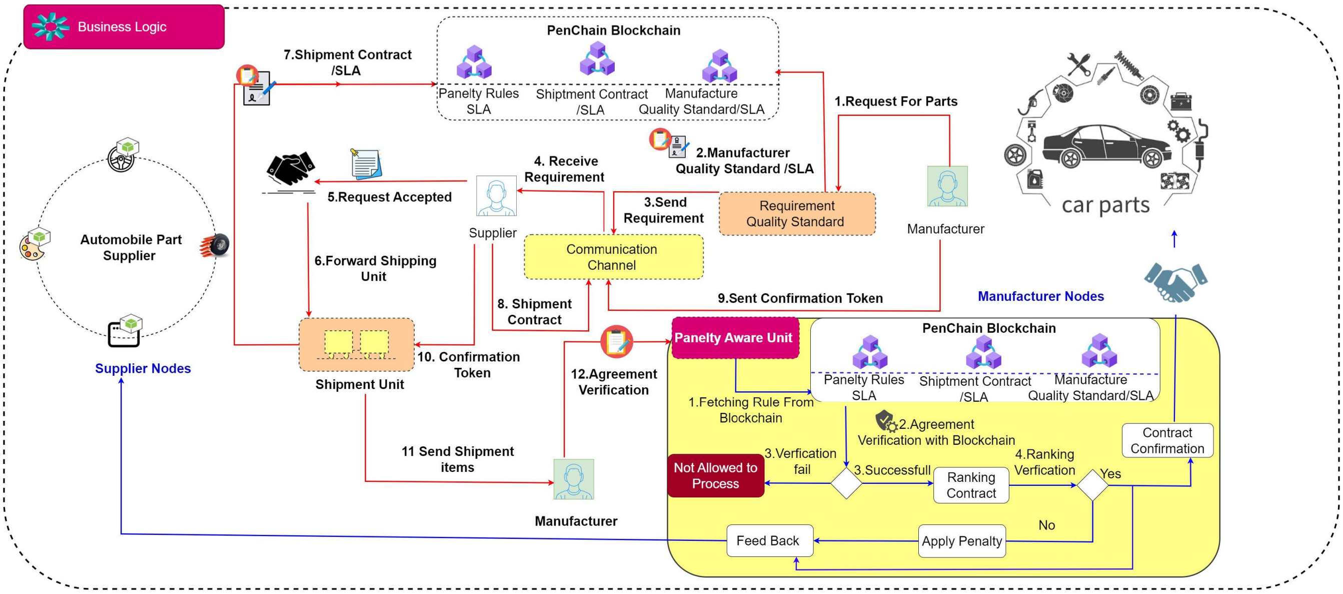


FIGURE 10. Workflow of the automobile manufacturing industry scenario for PenChain.

those parts or impose fines on the supplier. We formally express the penalty rule as follows:

$$\boxed{\text{Deduction\_Amount} = \text{item\_cost} \times (\text{Percentage\_Deduction}) + (\text{Time\_Delay} \times (\text{item\_cost} - (\text{percentage\_of\_item\_cost})))}$$

Figure 10 shows the overall workflow of the automotive industry scenario. We unlock the potential of blockchain technologies in the automobile industry sector, as pointed out earlier. In accordance with the general architecture of PenChain, the blockchain is utilized in multiple steps of the automobile manufacturing scenario. A detailed description of the steps is given as follows:

(1) At the start, manufacturers put their demand into the service level requirement, also called the manufacturer quality standard or SLA.
(2) The requested service requirement contract is posted to the blockchain as a manufacturing requirement contract.
(3) The requested message is forwarded through the communication channel to the supplier nodes.
(4) The supplier agent receives the request.
(5) If the supplier agent accepts the request, then the supplier agent forwards the request to the shipping unit in order to generate a shipment contract for the manufacturer node.
(6) Shipment units generate the contract for shipment.
(7) The shipment contract has been stored on the blockchain for verification purposes.
(8) The supplier's agent sent the shipment contract to the manufacturer.
(9) The Supplier nodes receive a confirmation token from the manufacturer
(10) Supplier nodes forward the confirmation token to the Shipment Unit
(11) After receiving the confirmation token from the supplier's shipment unit, send the item to the manufacturer.
(12) The manufacturer nodes verify the item's quality and shipment contract by fetching the contracts from the blockchain in order to apply penalties to the supplier nodes.

VII. FUTURE DIRECTIONS

In the future, we will investigate ways of automatically transforming the textual description of penalty rules into deontic logic. The antecedents and the consequent expressed for a penalty rule suggest what API call modules that programmatically support the compensation workflow shall be specified, developed and deployed on the service provider's side. While the chaincode that implements the compensation workflow remains unchanged should a penalty rule be revised, the corresponding external API call modules should be reprogrammed and redeployed. Work is currently underway to devise a tool proposal that allows registered service providers to edit their penalty rules with text suggestions, enabling PenChain to generate the APIs of the needed compensation modules hosted on the provider's side.

Integration of digitized SLAs with online payment services closes the loop to full-fledged automatized service provisioning. In this study, we did not investigate the realization of payment services, as the focus of this study was on digitized SLAs. Following the Web3 vision, payment services become subject to disintermediation, compared with Subsection II-D. To handle payments according to the Web3 paradigm, we currently investigate the integration of PenChain into the Alphabill platform that we have recently suggested [14]. Alphabill is a platform for universal asset tokenization, transfer and exchange as a global medium of exchange. Users of Alphabill can launch new so-called blockchain partitions. Each of the Alphabill partitions implements an individual token and its corresponding transaction system. The Alphabill platform provides the necessary technological fabrics that ensure governance, certification, and consistency with uncapped scalability [14]. In particular, the platform has inbuilt concepts to realize multi-asset swap transactions, which are also the key to ultra-scalable decentralized payment services.

VIII. CONCLUSION

Service delivery stays in the mainstream of today's information systems facing increasingly higher expectations of trust in service quality, and flexible ways to resolve disputes that may arise during a service transaction. The scale of service provisioning and the diversity of service offerings in various business domains necessitate a digital service discovery in which the digitized SLAs and the service provider's rating are both considered as ranking criteria for sorting the search results of service offerings. The ability to distinguish functionally similar services is vital to implement a novel service lookup mechanism in this digital service discovery. The digitized SLAs, which capture the mutual agreements between the service participants and might incorporate penalty rules – if securely saved by a trusted distributed ledger, manifest how reputable and reliable a provider has been from the service consumer's perspective.

Scholarly work in this realm continues to search for novel service registries and search techniques to trustfully handle an ever-growing number of service offerings. As innovative business and service models receive a boost from the recent advancement of distributed ledger technologies, we now look into blockchain-empowered trust mechanisms for service provisioning. In this line of work, we investigate the automatic enforcement of dynamic SLAs, which incorporate not only constraints on the service quality, but also human-mediated factors such as the penalty. We address the following three research questions: (i) In what formal or ground logic should penalty rules be expressed? (ii) How to digitize penalty-aware SLAs to enable an enforcement mechanism and the computation of an objective unbiased reputation in service provisioning? (iii) How to gear up ledger capabilities and blockchain's smart contracts for enforcing the human-in-the-loop penalty rules during a service transaction?

This article describes the conceptualization, design, engineering, and analytical considerations of PenChain – an SLA-driven service lookup platform that extensively utilizes blockchain technology. PenChain programmatically enforces penalty-aware SLAs and automatically assesses service providers' reputations. We present that the blockchain-empowered enforcement mechanism and the automated assessment of customer's satisfaction proposed in PenChain are key to an impartial and trustworthy evaluation of the SLAs in two use cases. We come to the conclusion that PenChain needs to be tailored or geared up for a specific service domain but the proposed service lookup and enforcement mechanism remain cross-domain. The rest of this section is dedicated to an open discussion on the shortcomings of our platform for SLA-minded service provisioning.

Shortcomings. We chew over the disadvantages of our proposal, which interestingly stem from the first two elements of the triplet in the formal definition we propose for the SLAs, namely the penalty rules and the satisfaction. We will, however, not delve into the third element of this triplet, namely the costs, simply because payment-related topics are not within the scope of our work.

While the penalty-enforcing chaincode in PenChain looks generic and seems to follow the write-once-run-anywhere feature that was once made popular in the era of virtual machines, it still relies on off-chain software components to carry out necessary compensation actions. Knowing that the off-chain components are supposedly written and maintained by the service providers, we implicitly assume that the providers are honest and will not intentionally take false compensation actions. In another word, our entire programmatic enforcement is exposed to their will. Although this assumption looks safe when executing fully digitized and verifiable compensation (e.g., discount applied to the e-payment), it is difficult to confirm human-in-the-loop compensation that is mediated by an off-chain software component.

We advocate the use of an automatic assessment to objectively determine the satisfaction level in an SLA. As depicted in the overall architecture (see Figure 1), *Assessment Unit* is an off-chain unit that assesses the outcome of every service transaction based on the service execution log accumulated in *System Log*. This design looks promisingly open and seems to work across industry sectors and service domains because, in principle, we can engineer *Assessment Unit* in accordance with the specific domain knowledge. Although this engineering choice proves effective in domains such as the car rental industry and data-as-a-service business models, the assessment is notoriously unattainable programmatically in domains where anthropomorphic entities are heavily involved (e.g., tourism).

ACKNOWLEDGMENT

The author Trung-Viet Nguyen acknowledges the support of time and facilities of the Ho Chi Minh City University of Technology (VNU-HCM) for this study.

REFERENCES

[1] T. Paschou, M. Rapaccini, F. Adrodegari, and N. Saccani, "Digital servitization in manufacturing: A systematic literature review and research agenda," *Ind. Marketing Manage.*, vol. 89, pp. 278–292, Aug. 2020.

[2] A. Bouguettaya et al., "A service computing manifesto: The next 10 years," *Commun. ACM*, vol. 60, no. 4, pp. 64–72, Apr. 2017.

[3] W. A. Awad and N. E. El-Attar, "Adaptive SLA mechanism based on fuzzy system for dynamic cloud environment," *Int. J. Comput. Appl.*, vol. 44, no. 1, pp. 12–22, Jan. 2022.

[4] J. M. García, O. Martín-Díaz, P. Fernandez, C. Müller, and A. Ruiz-Cortés, "A flexible billing life cycle for cloud services using augmented customer agreements," *IEEE Access*, vol. 9, pp. 44374–44389, 2021.

[5] S. Nakamoto. (2008). *Bitcoin: A Peer-to-Peer Electronic Cash System*. Accessed: Aug. 12, 2023. [Online]. Available: https://bitcoin.org/bitcoin.pdf

[6] M. Walport, "Distributed ledger technology: Beyond block chain," Government Office Sci., London, U.K., Tech. Rep. GS/16/1, Jan. 2016.

[7] D. Draheim, "Blockchains from an e-governance perspective: Potential and challenges," in *Proc. 7th Int. Conf. Electron. Governance Open Soc. (EGOSE)*, in Communications in Computer and Information Science, vol. 1349, J. Filipe, A. Ghosh, R. O. Prates, and L. Zhou, Eds. Cham, Switzerland: Springer, 2021, pp. 11–13.

[8] D. Yaga, P. Mell, N. Roby, and K. Scarfone, "Blockchain technology overview," Nat. Inst. Standards Technol., Gaithersburg, MD, USA, Tech. Rep. NISTIR 8202, Oct. 2018.

[9] M. Janssen, V. Weerakkody, E. Ismagilova, U. Sivarajah, and Z. Irani, "A framework for analysing blockchain technology adoption: Integrating institutional, market and technical factors," *Int. J. Inf. Manage.*, vol. 50, pp. 302–309, Feb. 2020.

[10] J. Mendling et al., "Blockchains for business process management—Challenges and opportunities," *ACM Trans. Manage. Inf. Syst.*, vol. 9, no. 1, pp. 1–16, 2018.

[11] S. Ølnes, J. Ubacht, and M. Janssen, "Blockchain in government: Benefits and implications of distributed ledger technology for information sharing," *Government Inf. Quart.*, vol. 34, no. 3, pp. 355–364, Sep. 2017.

[12] V. Shannon, "A 'more revolutionary' Web," The New York Times, New York, NY, USA, Tech. Rep., May 2006, vol. 23.

[13] G. Edelman, "Paradise at the crypto-arcade: Inside the web3 revolution," *Wired Mag.*, pp. 47–59, Jun. 2022. [Online]. Available: https://www.wired.com/story/web3-paradise-crypto-arcade/

[14] A. Buldas, D. Draheim, M. Gault, R. Laanoja, T. Nagumo, M. Saarepera, S. A. Shah, J. Simm, J. Steiner, T. Tammet, and A. Truu, "An ultra-scalable blockchain platform for universal asset tokenization: Design and implementation," *IEEE Access*, vol. 10, pp. 77284–77322, 2022.

[15] A. Buldas, D. Draheim, M. Gault, and M. Saarepera, "Towards a foundation of Web3," in *Proc. 9th Int. Conf. Future Data Secur. Eng. (FDSE)*, in Communications in Computer and Information Science, vol. 1688, T. K. Dang, J. Küng, and T. M. Chung, Eds. Singapore: Springer, 2022, p. 3—18.

[16] J. Wiles, "What is Web3?" Gartner Inf. Technol., Stamford, CT, USA, Tech. Rep., Feb. 2022.

[17] M. Bennett, "Web3 isn't going to fix the shortcomings of today's Web," Forrester, Cambridge, MA, USA, Tech. Rep., May 2022.

[18] B. Platz, "Why Web3 is so confusing," Forbes Technology Council, Stockholm, Sweden, Tech. Rep., Jun. 2022.

[19] T. Stackpole, "What is Web3?" Harvard Bus. Review, Boston, MA, USA, Tech. Rep., May 2022.

[20] L. Jin and K. Parrott, "Web3 is our chance to make a better Internet," Harvard Bus. Review, Boston, MA, USA, Tech. Rep., May 2022.

[21] J. Esber and S. D. Kominers, "Why build in Web3," Harvard Bus. Review, Boston, MA, USA, Tech. Rep., May 2022.

[22] D.-T. Le, T.-V. Nguyen, L.-S. Lê, and T. A. Kurniawan, "Reinforcing service level agreements in tourism sector the role of blockchain and mobile computing," in *Proc. 15th Int. Conf. Adv. Comput. Appl. (ACOMP)*, Nov. 2020, pp. 160–164.

[23] L.-S. Lê and T.-V. Nguyen, "Digitizing service level agreements in service-oriented enterprise architecture," *Social Netw. Comput. Sci.*, vol. 1, no. 5, pp. 1–20, Sep. 2020.

[24] T.-V. Nguyen, L.-S. Lê, H.-L. Truong, K. Nguyen-An, and P. H. Ha, "Handling service level agreements in IoT = minding rules + log analytics?" in *Proc. IEEE 22nd Int. Enterprise Distrib. Object Comput. Conf. (EDOC)*. Stockholm, Sweden: IEEE Computer Society, Oct. 2018, pp. 145–153.

[25] R. Schulte and Y. Natis, "Service oriented architectures—Part 1," Gartner Group, Stamford, CT, USA, Tech. Rep. SPA-401-068, 1996.

[26] R. Schulte, "Service oriented architectures—Part 2," Gartner Group, Stamford, CT, USA, Tech. Rep. SPA-401-069, 1996.

[27] T. H. Noor, Q. Z. Sheng, A. H. H. Ngu, and S. Dustdar, "Analysis of web-scale cloud services," *IEEE Internet Comput.*, vol. 18, no. 4, pp. 55–61, Jul. 2014.

[28] Y. Hu, A. Manzoor, P. Ekparinya, M. Liyanage, K. Thilakarathna, G. Jourjon, and A. Seneviratne, "A delay-tolerant payment scheme based on the ethereum blockchain," *IEEE Access*, vol. 7, pp. 33159–33172, 2019.

[29] N. Arshadi, "Application of blockchain protocol to wealth management," *J. Wealth Manage.*, vol. 21, no. 4, pp. 122–129, Jan. 2019.

[30] P. De Castro, M. Tanner, and K. Johnston, "Perceived factors influencing blockchain adoption in the asset and wealth management industry in the Western Cape, South Africa," in *Evolving Perspectives on ICTs in Global Souths*, D. R. Junio and C. Koopman, Eds. Cham, Switzerland: Springer, 2020, pp. 48–62.

[31] J. Sun, X. Yao, S. Wang, and Y. Wu, "Non-repudiation storage and access control scheme of insurance data based on blockchain in IPFS," *IEEE Access*, vol. 8, pp. 155145–155155, 2020.

[32] K. Kapadiya, U. Patel, R. Gupta, M. D. Alshehri, S. Tanwar, G. Sharma, and P. N. Bokoro, "Blockchain and AI-empowered healthcare insurance fraud detection: An analysis, architecture, and future prospects," *IEEE Access*, vol. 10, pp. 79606–79627, 2022.

[33] J. Veuger, "Trust in a viable real estate economy with disruption and blockchain," *Facilities*, vol. 36, nos. 1–2, pp. 103–120, Feb. 2018.

[34] M. Li, L. Shen, and G. Q. Huang, "Blockchain-enabled workflow operating system for logistics resources sharing in e-commerce logistics real estate service," *Comput. Ind. Eng.*, vol. 135, pp. 950–969, Sep. 2019.

[35] C. K. M. Lee, Y. Z. Huo, S. Z. Zhang, and K. K. H. Ng, "Design of a smart manufacturing system with the application of multi-access edge computing and blockchain technology," *IEEE Access*, vol. 8, pp. 28659–28667, 2020.

[36] H. R. Hasan, K. Salah, R. Jayaraman, R. W. Ahmad, I. Yaqoob, and M. Omar, "Blockchain-based solution for the traceability of spare parts in manufacturing," *IEEE Access*, vol. 8, pp. 100308–100322, 2020.

[37] A. Vedeshin, J. M. U. Dogru, I. Liiv, S. Ben Yahia, and D. Draheim, "A secure data infrastructure for personal manufacturing based on a novel key-less, byte-less encryption method," *IEEE Access*, vol. 8, pp. 40039–40056, 2020.

[38] J. Ordóñez, A. Alexopoulos, K. Koutras, A. Kalogeras, K. Stefanidis, and V. Martos, "Blockchain in agriculture: A PESTELS analysis," *IEEE Access*, vol. 11, pp. 73647–73679, 2023.

[39] M. A. Ferrag, L. Shu, X. Yang, A. Derhab, and L. Maglaras, "Security and privacy for green IoT-based agriculture: Review, blockchain solutions, and challenges," *IEEE Access*, vol. 8, pp. 32031–32053, 2020.

[40] A. Mohammed, V. Potdar, M. Quaddus, and W. Hui, "Blockchain adoption in food supply chains: A systematic literature review on enablers, benefits, and barriers," *IEEE Access*, vol. 11, pp. 14236–14255, 2023.

[41] D. Radain, S. Almalki, S. A. Almarghalani, and S. Elhag, "Towards achieving the 2030 vision, the case study of automating the food production services during the Hajj season and quality control using the blockchain technology," in *Proc. 5th Int. Conf. Future Netw. Distrib. Syst.* New York, NY, USA: Association for Computing Machinery, Dec. 2021, pp. 144–154.

[42] G. Perboli, S. Musso, and M. Rosano, "Blockchain in logistics and supply chain: A lean approach for designing real-world use cases," *IEEE Access*, vol. 6, pp. 62018–62028, 2018.

[43] Y. Fu and J. Zhu, "Operation mechanisms for intelligent logistics system: A blockchain perspective," *IEEE Access*, vol. 7, pp. 144202–144213, 2019.

[44] L. Ismail, H. Materwala, and S. Zeadally, "Lightweight blockchain for healthcare," *IEEE Access*, vol. 7, pp. 149935–149951, 2019.

[45] I. Azogu, A. Norta, I. Papper, J. Longo, and D. Draheim, "A framework for the adoption of blockchain technology in healthcare information management systems: A case study of Nigeria," in *Proc. 12th Int. Conf. Theory Pract. Electron. Governance*, Nigeria, Apr. 2019, pp. 310–316.

[46] I. Önder and H. Treiblmaier, "Blockchain and tourism: Three research propositions," *Ann. Tourism Res.*, vol. 72, pp. 180–182, Sep. 2018.

[47] Y. Gao, Q. Pan, Y. Liu, H. Lin, Y. Chen, and Q. Wen, "The notarial office in E-government: A blockchain-based solution," *IEEE Access*, vol. 9, pp. 44411–44425, 2021.

[48] N. Elisa, L. Yang, F. Chao, N. Naik, and T. Boongoen, "A secure and privacy-preserving e-government framework using blockchain and artificial immunity," *IEEE Access*, vol. 11, pp. 8773–8789, 2023.

[49] T. I. Akaba, A. Norta, C. Udokwu, and D. Draheim, "A framework for the adoption of blockchain-based e-procurement systems in the public sector," in *Proc. 19th IFIP Conf. e-Business, e-Services e-Soc.*, in Lecture Notes in Computer Science, vol. 12066, M. Hattingh, M. Matthee, H. Smuts, I. Pappas, Y. K. Dwivedi, and M. Mäntymäki, Eds. Cham, Switzerland: Springer, 2020, pp. 3–14.

[50] N. Lazuashvili, A. Norta, I. Pappel, and D. Draheim., "Integration of blockchain technology into a land registration system for immutable traceability: A case study of Georgia," in *Proc. 17th Int. Conf. Bus. Process Manage. (BPM)*, in Lecture Notes in Business Information Processing, vol. 361. Cham, Switzerland: Springer, 2019, pp. 3–14.

[51] E. Abodei, A. Norta, D. Azogu, C. Udokwu, and D. Draheim, "Blockchain technology for enabling transparent and traceable government collaboration in public project processes of developing economies," in *Proc. 18th IFIP Conf. e-Business, e-Services e-Society*, in Lecture Notes in Computer Science, vol. 11701. Cham, Switzerland: Springer, 2019, pp. 464–475.

[52] D. Kumar, S. Kumar, and A. Joshi, "Assessing the viability of blockchain technology for enhancing court operations," *Int. J. Law Manage.*, vol. 65, no. 5, pp. 425–439, Jul. 2023.

[53] V. Dwivedi, V. Pattanaik, V. Deval, A. Dixit, A. Norta, and D. Draheim, "Legally enforceable smart-contract languages: A systematic literature review," *ACM Comput. Surv.*, vol. 54, no. 5, pp. 1–34, Jun. 2022.

[54] A. Dixit, V. Deval, V. Dwivedi, A. Norta, and D. Draheim, "Towards user-centred and legally relevant smart-contract development: A systematic literature review," *J. Ind. Inf. Integr.*, vol. 26, Mar. 2022, Art. no. 100314.

[55] S. Tadelis, "Reputation and feedback systems in online platform markets," *Annu. Rev. Econ.*, vol. 8, no. 1, pp. 321–340, Oct. 2016.

[56] T. Matherly, "A panel for lemons? Positivity bias, reputation systems and data quality on MTurk," *Eur. J. Marketing*, vol. 53, no. 2, pp. 195–223, Feb. 2019.

[57] K. M. Khan, J. Arshad, W. Iqbal, S. Abdullah, and H. Zaib, "Blockchain-enabled real-time SLA monitoring for cloud-hosted services," *Cluster Comput.*, vol. 25, no. 1, pp. 537–559, Feb. 2022.

[58] H. Ludwig, A. Keller, A. Dan, R. P. King, and R. Franck, "Web service level agreement (WSLA) language specification," IBM Corp., Armonk, NY, USA, Tech. Rep., Jan. 2003.

[59] R. Engel, S. Rajamoni, B. Chen, H. Ludwig, and A. Keller, "ySLA: Reusable and configurable SLAs for large-scale SLA management," in *Proc. IEEE 4th Int. Conf. Collaboration Internet Comput. (CIC)*. Philadelphia, PA, USA: IEEE, Oct. 2018, pp. 317–325.

[60] W. Hussain, F. K. Hussain, O. Hussain, R. Bagia, E. Chang, and A. Romanovsky, "Risk-based framework for SLA violation abatement from the cloud service provider's perspective," *Comput. J.*, vol. 61, no. 9, pp. 1306–1322, Sep. 2018.

[61] T. Labidi, A. Mtibaa, W. Gaaloul, S. Tata, and F. Gargouri, "Cloud SLA modeling and monitoring," in *Proc. IEEE Int. Conf. Services Comput. (SCC)*, Honolulu, HI, USA, Jun. 2017, pp. 338–345.

[62] V. K. Prasad and M. D. Bhavsar, "Monitoring and prediction of SLA for IoT based cloud," *Scalable Comput., Pract. Exp.*, vol. 21, no. 3, pp. 349–358, Aug. 2020.

[63] S. Noureddine and B. Meriem, "ML-SLA-IoT: An SLA specification and monitoring framework for IoT applications," in *Proc. Int. Conf. Inf. Syst. Adv. Technol. (ICISAT)*, Tebessa, Algeria, Dec. 2021, pp. 1–12.

[64] R. Goyat, G. Kumar, M. Alazab, R. Saha, R. Thomas, and M. K. Rai, "A secure localization scheme based on trust assessment for WSNs using blockchain technology," *Future Gener. Comput. Syst.*, vol. 125, pp. 221–231, Dec. 2021.

[65] E. J. Scheid, B. B. Rodrigues, L. Z. Granville, and B. Stiller, "Enabling dynamic SLA compensation using blockchain-based smart contracts," in *Proc. IFIP/IEEE Symp. Integr. Netw. Service Manage. (IM)*. Washington, DC, USA: IEEE, Apr. 2019, pp. 53–61.

[66] A. Oluwabukola and A. Adebowale, "An architectural model for SLA negotiation between SaaS and customers," *Int. J. Eng. Appl. Sci. Technol.*, vol. 5, no. 5, pp. 30–36, 2020.

[67] F. Li, G. White, and S. Clarke, "A trust model for SLA negotiation candidates selection in a dynamic IoT environment," *IEEE Trans. Services Comput.*, vol. 15, no. 5, pp. 2565–2578, Sep. 2022.

[68] T. Labidi, A. Mtibaa, W. Gaaloul, and F. Gargouri, "Cloud SLA negotiation and re-negotiation: An ontology-based context-aware approach," *Concurrency Comput., Pract. Exp.*, vol. 32, no. 15, Aug. 2020.

[69] H. Nakashima and M. Aoyama, "An automation method of SLA contract of Web APIs and its platform based on blockchain concept," in *Proc. IEEE Int. Conf. Cogn. Comput. (ICCC)*, Honolulu, HI, USA, Jun. 2017, pp. 32–39.

[70] H. Zhou, C. de Laat, and Z. Zhao, "Trustworthy cloud service level agreement enforcement with blockchain based smart contract," in *Proc. IEEE Int. Conf. Cloud Comput. Technol. Sci. (CloudCom)*. Nicosia, Cyprus: IEEE, Dec. 2018, pp. 255–260.

[71] A. Alzubaidi, K. Mitra, P. Patel, and E. Solaiman, "A blockchain-based approach for assessing compliance with SLA-guaranteed IoT services," in *Proc. IEEE Int. Conf. Smart Internet Things (SmartIoT)*. Beijing, China: IEEE, Aug. 2020, pp. 213–220.

[72] R. Ranchal and O. Choudhury, "SLAM: A framework for SLA management in multicloud ecosystem using blockchain," in *Proc. IEEE Cloud Summit*, Harrisburg, PA, USA, Oct. 2020, pp. 33–38.

[73] P. M. Abhishek, A. Chobari, and D. G. Narayan, "SLA violation detection in multi-cloud environment using hyperledger fabric blockchain," in *Proc. IEEE Int. Conf. Distrib. Comput., VLSI, Electr. Circuits Robot. (DISCOVER)*. Nitte, India: IEEE, Nov. 2021, pp. 107–112.

[74] N. Neidhardt, C. Köhler, and M. Nüttgens, "Cloud service billing and service level agreement monitoring based on blockchain," in *Proc. 9th Int. Workshop Enterprise Model. Inf. Syst. Archit.*, 2018, pp. 65–69.

[75] A. T. Wonjiga, S. Peisert, L. Rilling, and C. Morin, "Blockchain as a trusted component in cloud SLA verification," in *Proc. 12th IEEE/ACM Int. Conf. Utility Cloud Comput. Companion*, Dec. 2019, pp. 93–100.

[76] H. Zhou, X. Ouyang, Z. Ren, J. Su, C. de Laat, and Z. Zhao, "A blockchain based witness model for trustworthy cloud service level agreement enforcement," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*. Nicosia, Cyprus: IEEE, Apr. 2019, pp. 1567–1575.

[77] S. K. Battula, S. Garg, R. Naha, M. B. Amin, B. Kang, and E. Aghasian, "A blockchain-based framework for automatic SLA management in fog computing environments," *J. Supercomput.*, vol. 78, no. 15, pp. 16647–16677, Oct. 2022.

[78] M. S. Rahman, I. Khalil, and M. Atiquzzaman, "Blockchain-enabled SLA compliance for crowdsourced edge-based network function virtualization," *IEEE Netw.*, vol. 35, no. 5, pp. 58–65, Sep. 2021.

[79] K. Wang, B. Yang, L. Su, and Y. Hu, "Blockchain based data sharing for user experience driven slice SLA guarantee," in *Proc. Int. Conf. Service Sci. (ICSS)*. Xi'an, China: IEEE, May 2021, pp. 7–13.

[80] A. Kalla, C. de Alwis, P. Porambage, G. Gür, and M. Liyanage, "A survey on the use of blockchain for future 6G: Technical aspects, use cases, challenges and research directions," *J. Ind. Inf. Integr.*, vol. 30, Nov. 2022, Art. no. 100404.

[81] K. Xiao, Z. Geng, Y. He, G. Xu, C. Wang, and W. Cheng, "A blockchain based privacy-preserving cloud service level agreement auditing scheme," in *Proc. 5th Int. Conf. Wireless Algorithms, Syst., Appl.* Berlin, Heidelberg: Springer, 2020, pp. 542–554.

[82] K. Xiao, Z. Geng, Y. He, G. Xu, C. Wang, and Y. Tian, "A blockchain-based privacy-preserving 5G network slicing service level agreement audit scheme," *EURASIP J. Wireless Commun. Netw.*, vol. 2021, no. 1, p. 165, Dec. 2021.

[83] M. Khalid, S. Hameed, A. Qadir, S. A. Shah, and D. Draheim, "Towards SDN-based smart contract solution for IoT access control," *Comput. Commun.*, vol. 198, pp. 1–31, Jan. 2023.

[84] S. Siddiqui, S. Hameed, S. A. Shah, A. K. Khan, and A. Aneiba, "Smart contract-based security architecture for collaborative services in municipal smart cities," *J. Syst. Archit.*, vol. 135, Feb. 2023, Art. no. 102802.

[85] R. B. Uriarte, R. de Nicola, and K. Kritikos, "Towards distributed SLA management with smart contracts and blockchain," in *Proc. IEEE Int. Conf. Cloud Comput. Technol. Sci. (CloudCom)*, Dec. 2018, pp. 266–271.

[86] A. K. Pandey, D. G. Narayan, and S. K., "SLA violation detection and compensation in cloud environment using blockchain," in *Proc. 12th Int. Conf. Comput. Commun. Netw. Technol. (ICCCNT)*. Kharagpur, India: IEEE, Jul. 2021, pp. 1–6.

[87] I. Singh and S.-W. Lee, "RE_BBC: Requirements engineering in a blockchain-based cloud system: Its role in service-level agreement specification," *IEEE Softw.*, vol. 37, no. 5, pp. 7–12, Sep. 2020.

[88] I. Singh and S.-W. Lee, "SRE_BBC: A self-adaptive security enabled requirements engineering approach for SLA smart contracts in blockchain-based cloud systems," *Sensors*, vol. 22, no. 10, p. 3903, May 2022.

[89] R. B. Uriarte, H. Zhou, K. Kritikos, Z. Shi, Z. Zhao, and R. De Nicola, "Distributed service-level agreement management with smart contracts and blockchain," *Concurrency Comput., Pract. Exp.*, vol. 33, no. 14, Jul. 2021.

[90] N. Abbatemarco, L. M. De Rossi, A. Gaur, and G. Salviotti, "Beyond a blockchain paradox: How intermediaries can leverage a disintermediation technology," in *Proc. Annu. Hawaii Int. Conf. Syst. Sci.*, 2020, pp. 5328–5337.

[91] W. Nowiński and M. Kozma, "How can blockchain technology disrupt the existing business models?" *Entrepreneurial Bus. Econ. Rev.*, vol. 5, no. 3, pp. 173–188, 2017.

[92] O. E. Williamson, "Transaction cost economics: The governance of contractual relations," *J. Law Econ.*, vol. 22, no. 2, pp. 233–261, 1979.

[93] O. E. Williamson, "Transaction cost economics: How it works; where it is headed," *De Economist*, vol. 146, no. 1, pp. 23–58, 1998.

[94] J. Koppenjan and J. Groenewegen, "Institutional design for complex technological systems," *Int. J. Technol., Policy Manage.*, vol. 5, no. 3, pp. 240–257, 2005.

[95] D. Draheim, R. Krimmer, and T. Tammet, "On state-level architecture of digital government ecosystems: From ICT-driven to data-centric," in *Transactions on Large-Scale Data- and Knowledge-Centered Systems*, vol. 48. Berlin, Germany: Springer, 2021, pp. 165–195.

[96] A. Oram, *Peer to Peer: Harnessing the Power of Disruptive Technologies*. Sebastopol, CA, USA: O'Reilly, Mar. 2001.

[97] A. Narayanan and J. Clark, "Bitcoin's academic pedigree," *Commun. ACM*, vol. 60, no. 12, pp. 36–45, Nov. 2017.

[98] A. Narayanan and J. Clark, "Bitcoin's academic pedigree," *ACM Queue Mag.*, vol. 15, no. 4, pp. 1–30, 2017.

[99] M. Pease, R. Shostak, and L. Lamport, "Reaching agreement in the presence of faults," *J. ACM*, vol. 27, no. 2, pp. 228–234, Apr. 1980.

[100] L. Lamport, R. Shostak, and M. Pease, "The Byzantine generals problem," *ACM Trans. Program. Lang. Syst.*, vol. 4, no. 3, pp. 382–401, Jul. 1982.

[101] P. Weygant, *Clusters for High Availability: A Primer of Hp Solutions*. Hoboken, NJ, USA: Prentice Hall, 2001.

[102] R. Elling and T. Read, *Designing Enterprise Solutions with Sun Cluster 3.0*. Santa Clara, CA, USA: Sun Microsystems, 2001.

[103] O. Lascu, S. Bodily, M.-K. Esser, M. Herrera, P. Pothier, D. Prelec, D. Quintero, K. Raymond, V. Sebesteny, A. Socoliuc, and A. Steel, "Implementing high availability cluster multi-processing (HACMP) cookbook," IBM Int. Tech. Org., Armonk, NY, USA, Tech. Rep. SG24-6769-00, Dec. 2005.

[104] R. Oppliger, "Internet security: Firewalls and beyond," *Commun. ACM*, vol. 40, no. 5, pp. 92–102, May 1997.

[105] A. S. Tanenbaum and D. J. Wetherall, *Computer Networks*, 5th ed. London, U.K.: Pearson, 2012.

[106] A. G. Mason, *Cisco Secure Virtual Private Networks*. Indianapolis, IN, USA: Cisco Press, 2002.

[107] G. Yu, X. Wang, K. Yu, W. Ni, J. A. Zhang, and R. P. Liu, "Survey: Sharding in blockchains," *IEEE Access*, vol. 8, pp. 14155–14181, 2020.

[108] D. Jia, J. Xin, Z. Wang, and G. Wang, "Optimized data storage method for sharding-based blockchain," *IEEE Access*, vol. 9, pp. 67890–67900, 2021.

[109] A. M. Antonopoulos, *Mastering Bitcoin: Programming the Open Blockchain*. Sebastopol, CA, USA: O'Reilly, 2017.

[110] A. Buldas and M. Saarepera, "Document verification with distributed calendar infrastructure," U.S. Patent Appl. U.S. Patent 20 130 276 058 A1, May 2013.

[111] A. Buldas, A. Kroonmaa, and R. Laanoja, "Keyless signatures' infrastructure: How to build global distributed hash-trees," in *Proc. 18th Nordic Conf. (NordSec)*, in Lecture Notes in Computer Science, vol. 8208. New York, NY, USA: Springer, 2013, pp. 313–320.

[112] A. Ansper, A. Buldas, M. Freudenthal, and J. Willemson, "High-performance qualified digital signatures for X-road," in *Proc. 18th Nordic Conf. (NordSec)*, in Lecture Notes in Computer Science, vol. 8208. Berlin, Germany: Springer, 2013, pp. 123–138.

[113] C. Adams and S. Lloyd, *Understanding PKI: Concepts, Standards, and Deployment Considerations*. Reading, MA, USA: Addison-Wesley, 2003.

[114] M. Rosenblum and J. K. Ousterhout, "The design and implementation of a log-structured file system," *ACM SIGOPS Operating Syst. Rev.*, vol. 25, no. 5, pp. 1–15, Oct. 1991.

[115] H. Bedoya, L. Youngren, M. Abrahams, Y. Asaeda, J. Mathews, A. Stallman, and V. Sueiro, "Striving for optimal journal performance on DB2 universal database for iSeries," IBM Int. Tech. Support Org., Armonk, NY, USA, Tech. Rep. SG24-6286-00, May 2002.

[116] C. H. Liu, Q. Lin, and S. Wen, "Blockchain-enabled data collection and sharing for industrial IoT with deep reinforcement learning," *IEEE Trans. Ind. Informat.*, vol. 15, no. 6, pp. 3516–3526, Jun. 2019.

[117] F.-J. Ferrández-Pastor, J. Mora-Pascual, and D. Díaz-Lajara, "Agricultural traceability model based on IoT and blockchain: Application in industrial hemp production," *J. Ind. Inf. Integr.*, vol. 29, Sep. 2022, Art. no. 100381.

[118] X. Xu, X. Zhang, H. Gao, Y. Xue, L. Qi, and W. Dou, "BeCome: Blockchain-enabled computation offloading for IoT in mobile edge computing," *IEEE Trans. Ind. Informat.*, vol. 16, no. 6, pp. 4187–4195, Jun. 2020.

[119] C. A. Ardagna, R. Asal, E. Damiani, N. E. Ioini, M. Elahi, and C. Pahl, "From trustworthy data to trustworthy IoT: A data collection methodology based on blockchain," *ACM Trans. Cyber-Phys. Syst.*, vol. 5, no. 1, pp. 1–26, Dec. 2020.

[120] P. K. Sharma and J. H. Park, "Blockchain based hybrid network architecture for the smart city," *Future Gener. Comput. Syst.*, vol. 86, pp. 650–655, Sep. 2018.

[121] S. Hameed, S. A. Shah, Q. S. Saeed, S. Siddiqui, I. Ali, A. Vedeshin, and D. Draheim, "A scalable key and trust management solution for IoT sensors using SDN and blockchain technology," *IEEE Sensors J.*, vol. 21, no. 6, pp. 8716–8733, Mar. 2021.

[122] S. Siddiqui, S. Hameed, S. A. Shah, I. Ahmad, A. Aneiba, D. Draheim, and S. Dustdar, "Toward software-defined networking-based IoT frameworks: A systematic literature review, taxonomy, open challenges and prospects," *IEEE Access*, vol. 10, pp. 70850–70901, 2022.

[123] M. Zichichi, G. D'Angelo, S. Ferretti, and M. Marzolla, "Accountable clouds through blockchain," *IEEE Access*, vol. 11, pp. 48358–48374, 2023.

[124] T. Faisal, M. Dohler, S. Mangiante, and D. R. Lopez, "BEAT: Blockchain-enabled accountable and transparent infrastructure sharing in 6G and beyond," *IEEE Access*, vol. 10, pp. 48660–48672, 2022.

[125] A. P. Singh, N. R. Pradhan, A. K. Luhach, S. Agnihotri, N. Z. Jhanjhi, S. Verma, Kavita, U. Ghosh, and D. S. Roy, "A novel patient-centric architectural framework for blockchain-enabled healthcare applications," *IEEE Trans. Ind. Informat.*, vol. 17, no. 8, pp. 5779–5789, Aug. 2021.

[126] A. S. Khan, G. Chen, Y. Rahulamathavan, G. Zheng, B. Assadhan, and S. Lambotharan, "Trusted UAV network coverage using blockchain, machine learning, and auction mechanisms," *IEEE Access*, vol. 8, pp. 118219–118234, 2020.

[127] T. Berners-Lee, J. Hendler, and O. Lassila, "The Semantic Web—A new form of Web content that is meaningful to computers will unleash a revolution of new possibilities," *Sci. Amer.*, vol. 284, no. 5, pp. 34–43, May 2001.

[128] A. Prodromou, "TLS Security 2: A brief history of SSL/TLS," Acunetix, London, U.K., Tech. Rep., Mar. 2019.

[129] D. Draheim, "The service-oriented metaphor deciphered," *J. Comput. Sci. Eng.*, vol. 4, no. 4, pp. 253–275, Dec. 2010.

[130] T. Berners-Lee, "Socially aware cloud storage," in *Proc. World Wide Web Consortium*, Aug. 2009. [Online]. Available: https://www.w3.org/DesignIssues/CloudStorage.html

[131] T. Berners-Lee, "Read-write linked data," in Proc. *World Wide Web Consortium*, Oct. 2009, pp. 20120513–20120518.

[132] E. Mansour, A. V. Sambra, S. Hawke, M. Zereba, S. Capadisli, A. Ghanem, A. Aboulnaga, and T. Berners-Lee, "A demonstration of the solid platform for social Web applications," in *Proc. 25th Int. Conf. Companion World Wide Web*. Montréal, QC, Canada: ACM, 2016, pp. 223–226.

[133] D. Weinberber. (Aug. 2016). *How the Father of the World Wide Web Plans to Reclaim it from Facebook and Google*. [Online]. Available: https://www.digitaltrends.com/

[134] C. Lagarde, "Central banking and fintech—A brave new world?" in *Proc. Bank England Conf.*, London, U.K., Sep. 2017, pp. 4–8.

[135] A. Buldas, D. Draheim, T. Nagumo, and A. Vedeshin, "Blockchain technology: Intrinsic technological and socio-economic barriers," in *Proc. 7th Int. Conf. Future Data Secur. Eng.*. Quy Nhon, Vietnam: Springer, 2020, pp. 3–27.

[136] J. Fernández-Villaverde, D. Sanches, L. Schilling, and H. Uhlig, "Central bank digital currency: Central banking for all?" *Rev. Econ. Dyn.*, vol. 41, pp. 225–242, Jul. 2021.

[137] L. Grassi, D. Lanfranchi, A. Faes, and F. M. Renga, "Do we still need financial intermediation? The case of decentralized finance – DeFi," *Qualitative Res. Accounting Manage.*, vol. 19, no. 3, pp. 323–347, Feb. 2022.

[138] Y. Chen and C. Bellavitis, "Blockchain disruption and decentralized finance: The rise of decentralized business models," *J. Bus. Venturing Insights*, vol. 13, Jun. 2020, Art. no. e00151.

[139] F. Schär, "Decentralized finance: On blockchain- and smart contract-based financial markets," *Review*, vol. 103, no. 2, pp. 74–153, 2021.

[140] *Information Technology—Security Techniques—Identity Proofing*, ISO Standard 29003:2018, 2018.

[141] S. Lips, N. Bharosa, and D. Draheim, "eIDAS implementation challenges: The case of Estonia and The Netherlands," in *Proc. 7th Int. Conf. Electron. Governance Open Society*. St. Petersburg, Russia: Springer, Nov. 2020, pp. 75–89.

[142] A. Mühle, A. Grüner, T. Gayvoronskaya, and C. Meinel, "A survey on essential components of a self-sovereign identity," *Comput. Sci. Rev.*, vol. 30, pp. 80–86, Nov. 2018.

[143] A. Norta, A. Kormiltsyn, C. Udokwu, V. Dwivedi, S. Aroh, and I. Nikolajev, "A blockchain implementation for configurable multi-factor challenge-set self-sovereign identity authentication," in *Proc. IEEE Int. Conf. Blockchain (Blockchain)*. Espoo, Finland: Springer, Aug. 2022, pp. 455–461.

[144] S. Voshmgir, *Token Economy—How Web3 Reinvents Internet*, 2nd ed. Berlin, Germany: BlochainHub, Jun. 2020.

[145] A. Andjelic, "How brands are experimenting with Web3," Harvard Bus. Review, Boston, MA, USA, Tech. Rep., May 2022.

[146] S. Wang, W. Ding, J. Li, Y. Yuan, L. Ouyang, and F.-Y. Wang, "Decentralized autonomous organizations: Concept, model, and applications," *IEEE Trans. Computat. Social Syst.*, vol. 6, no. 5, pp. 870–878, Oct. 2019.

[147] M. Niforos, "The promising future of NFTs remains in a state of flux," Financial Times, London, U.K., Tech. Rep., Jun. 2022.

[148] M. Dowling, "Fertile LAND: Pricing non-fungible tokens," *Finance Res. Lett.*, vol. 44, Jan. 2022, Art. no. 102096.

[149] R. Sharma, "What is a non-fungible token (NFT)?" Investopedia, New York, NY, USA, Tech. Rep., Jan. 2023.

[150] C. S. Yeo and R. Buyya, "Service level agreement based allocation of cluster resources: Handling penalty to enhance utility," in *Proc. IEEE Int. Conf. Cluster Comput.*, Sep. 2005, pp. 1–10.

[151] B. Pernici, S. H. Siadat, S. Benbernou, and M. Ouziri, "A penalty-based approach for QoS dissatisfaction using fuzzy rules," in *Proc. 9th Int. Conf. Service-Oriented Comput.* Berlin, Germany: Springer, 2011, pp. 574–581.

[152] N. N. Z. Abidin, N. A. Manaf, S. Moschoyiannis, and N. A. Jamaludin, "Deontic rule of rule-based service choreographies," in *Proc. 2nd Int. Conf. Comput. Data Sci. (CDS)*, Jan. 2021, pp. 510–515.

[153] P. Gong, D. Knuplesch, and M. Reichert, "Rule-based monitoring framework for business process compliance," *Int. J. Web Services Res.*, vol. 14, no. 2, 2017.

[154] G. Ristow and J. Voegele, "Method and monitoring system for the rule-based monitoring of a service-oriented architecture," U.S. Patent 8 271 407, Sep. 18, 2012.

[155] P. F. Linington, Z. Milosevic, J. Cole, S. Gibson, S. Kulkarni, and S. Neal, "A unified behavioural model and a contract language for extended enterprise," *Data Knowl. Eng.*, vol. 51, no. 1, pp. 5–29, Oct. 2004.

[156] D.-M. Gabbay and J. Woods, *Logic and the Modalities in the 20th Century* (Handbook of the History of Logic), vol. 7. Amsterdam, The Netherlands: North Holland, Jul. 2006.

[157] S. Bistarelli, *Semirings for Soft Constraint Solving and Programming*. Berlin, Germany: Springer, 2004.

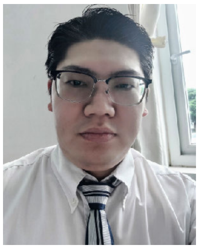
[158] W. J. Kettinger and C. C. Lee, "Perceived service quality and user satisfaction with the information services function," *Decis. Sci.*, vol. 25, nos. 5–6, pp. 737–766, Sep. 1994.

[159] H. Q. T. Ngo, H. Cao Tri, N. T. Tu, D. N. T. Bao, P. L. A. Duy, K. M. Phat, T. A. Duy, and N. T. Tin, "Design of reconfigurable mechanism for underactuated robot in the grounded applications," *Cogent Eng.*, vol. 9, no. 1, Dec. 2022, Art. no. 2095882.

[160] R. Saket, N. Singh, P. Dayama, and V. Pandit, "Smart contract protocol for authenticity and compliance with anonymity on hyperledger fabric," in *Proc. IEEE Int. Conf. Blockchain Cryptocurrency (ICBC)*, May 2020, pp. 1–9.

[161] J. Bhayo, S. A. Shah, S. Hameed, A. Ahmed, J. Nasir, and D. Draheim, "Towards a machine learning-based framework for DDOS attack detection in software-defined IoT (SD-IoT) networks," *Eng. Appl. Artif. Intell.*, vol. 123, Aug. 2023, Art. no. 106432.

[162] K. Peng, M. Li, H. Huang, C. Wang, S. Wan, and K. R. Choo, "Security challenges and opportunities for smart contracts in Internet of things: A survey," *IEEE Internet Things J.*, vol. 8, no. 15, pp. 12004–12020, Aug. 2021.



TRUNG-VIET NGUYEN received the master's degree in information systems from Can Tho University, in 2014. He is currently a Research Scholar with the Ho Chi Minh City University of Technology. His reflects dedication to understanding and improving the underlying mechanisms that drive modern technological ecosystems. His research interests include systems, service science, software technology, and blockchain.

LAM-SON LÊ (Member, IEEE) received the Ph.D. degree from the Swiss Federal Institute of Technology Lausanne (EPFL), Switzerland, in 2008. He was a Postdoctoral Researcher with the University of Wollongong, Australia, for four years. He is currently a Software Engineer with the Faculty of Engineering, Vietnamese-German University (VGU). Before joining VGU, he was with the HCMC University of Technology (HCMUT)–Vietnam National University, leading a research group on data-driven enterprise software and processes for nine years.

SYED ATTIQUE SHAH (Senior Member, IEEE) received the Ph.D. degree from the Institute of Informatics, Istanbul Technical University, Istanbul, Turkey. During the Ph.D. degree, he was a Visiting Scholar with The University of Tokyo, Japan, National Chiao Tung University, Taiwan, and the Tallinn University of Technology, Estonia, where he completed the major content of the thesis. He was an Associate Professor and the Chairperson with the Department of Computer Science, BUITEMS, Quetta, Pakistan. He was also engaged as a Lecturer with the Data Systems Group, Institute of Computer Science, University of Tartu, Estonia. He is currently a Lecturer in smart computer systems with the School of Computing and Digital Technology, Birmingham City University, U.K. His research interests include big data analytics, the Internet of Things, machine learning, network security, and information management.

SUFIAN HAMEED received the Ph.D. degree in networks and information security from the University of Göttingen, Germany. He is currently an Associate Professor with the Department of Computer Science, National University of Computer and Emerging Sciences (NUCES), Pakistan. He also leads IT Security Laboratories, NUCES. The research laboratory studies and teaches security problems and solutions for different types of information and communication paradigms. His research interests include network security, web security, mobile security and secure architectures, and protocols for cloud and the IoT.

DIRK DRAHEIM (Member, IEEE) received the Ph.D. degree from Freie Universität Berlin and the Habilitation degree from Universität Mannheim, Germany. He is currently a Full Professor of information systems and the Head of the Information Systems Group, Tallinn University of Technology, Estonia. The information systems group conducts research in large and ultra-large-scale IT systems. He is also an Initiator and the Leader of numerous digital transformation initiatives.