Handling Service Level Agreements in IoT = Minding Rules + Log Analytics?

Trung-Viet Nguyen[1,4], Lam-Son Lê[1], Hong-Linh Truong[2], Khuong Nguyen-An[1], and Phuong Hoai Ha[3]

[1]Faculty of Computer Science & Engineering, HCMC University of Technology, Vietnam
ntviet@cse.hcmut.edu.vn, lam-son.le@alumni.epfl.ch, nakhuong@hcmut.edu.vn

[2]Faculty of Informatics, TU Wien, Austria
hong-linh.truong@tuwien.ac.at

[3]UiT The Arctic University of Norway, Norway
phuong.hoai.ha@uit.no

[4]Faculty of Information Technology, Can Tho University of Technology, Vietnam
ntviet@ctuet.edu.vn

***Abstract*—With the rise of Internet of Things, end-users expect to obtain data from well-connected smart devices and stations through data services being provisioned in distributed architectures. Such services could be aggregated in a number of smart ways to provide the end-users and third-party applications with sophisticated data (e.g., weather data coupled with soil pollution), resulting in a growing number of service offerings to be requested. Service offerings that have been shortlisted for a certain data request (e.g., rainfall in a particular farming site) need to be ranked according to the end-users' preference. Service level agreements, i.e., the mutual responsibilities between the service provider and its consumers, address this sort of preference. Unfortunately, provisioning quality-aware services under this term still stays on the sidelines. In this paper, we propose a novel service architecture where the service level agreements shall be: (i) accumulated overtime on IoT service transactions; (ii) compiled when aggregating IoT services; (iii) used as a ranking criterion for suggesting IoT service offerings. We demonstrate our new approach in the service provisioning of agricultural datasets taken from a farming site of the Mekong Delta in Vietnam.**

***Keywords*—IoT services, service level agreement, DaaS, recommender systems**

I. INTRODUCTION

Service-oriented computing (SOC) combined with the Internet of Things (IoT) technologies leads to a paradigm shift in computing [1], giving rise to the concept of IoT services that is merely regarded as a transaction between parties (e.g. sensor and consumer) [2]. This paradigm shift, as pointed out in a manifesto over the future research of SOC [3], would offer a reboot of the SOC research area. IoT is becoming a newly-emerging enabling platform for data services (Data as a Service – DaaS). Data provisioning on such a platform is a game-changer for being distributed in enterprise engineering, i.e., data could be obtained via services built atop loosely-coupled IoT smart devices.

As recognized in many researches, distinguishing functionally similar services using quality is a key factor [3]. Among IoT data service offerings that provide a certain kind of data, the end-users naturally pick up those that are contractually described in their expected quality. When it comes to IoT services, as the number of service providers drastically increases, handling service level agreements (SLAs) would play an even more crucial role in the inter-system orchestration [4]. To this end, explicitly capturing the SLA for IoT data services is of paramount importance. Semantically specifying SLAs and reasoning about them will open the door for an advanced registry of IoT services equipped with a special indexing mechanism that facilitates service look-up using both functional criteria (i.e., what data to be provided) and non-functional ones (i.e., the finest SLA to be expected). Eventually, the truly distributed platform that harbors such a service registry might lead to a new dominating enterprise service-oriented architecture [1]. From a business point of view, to match the end-user's intention, services that are sourced from weakly-coupled providers might be bundled to yield value-added service aggregations [5].

The concept of SLA has been investigated substantially in SOC and enterprise computing, especially for defining and monitoring SLAs. Our goal in this line of research is to narrow down the domain to IoT data services while widening the representation of SLAs, especially in data and data quality aspects, for IoT data services. A typical SLA stated for an IoT service might conceptually encapsulate the following items [4]: reliability (e.g., percentage uptime and other measurements), responsiveness (e.g., how quickly various issues will be resolved), consequences and expectations (e.g. when SLA expectations are not met) and exception clauses or constraints (e.g. when SLA does not apply such as force majeure). To address IoT services in a wide spectrum, we propose that an SLA should be described as a mixture of numeric factors (e.g., payment and error rate) and business parameters (e.g., penalty rules). Our way of reasoning about the SLA was inspired by a grid for measuring service quality perceived by service customers [6]. This grid, named SERVQUAL, specifically addresses the assurance, reliability, responsiveness, tangibles and empathy of a business service. A variant of this grid developed for evaluating the quality of service [7] has reinforced the importance of the reliability and the empathy in measuring customer's satisfaction. Our definition of the SLA for the IoT data services targets the customer's satisfaction.

To handle data delivery, the SLA should keep track of penalty rules that are articulated in favor of the end-users in case the end-users do not obtain their expected data. The third component of the SLA is about costs incurred for consuming the said IoT services. Assuming that the satisfaction and the rule-abiding rate of a service could be mined from logs, we address the following questions in this paper: (a) how the SLA of a service aggregation could be compiled from the sourced services? (b) among service offerings that were previously shortlisted matching the end-user's intention, which one is the best in terms of the SLA?

Section II is dedicated to the preliminaries of our work and its related work. Section III formulates our research statement using a case-study. In Section IV, we define the SLAs of IoT data services and proposes SLA-aware algorithms for looking them up and pinpointing the finest one. Section V reports our implementation of the proposed SLA-handling algorithms using real-life data. Section VI concludes the paper and outlines our future work.

II. Background and Related Work

A. IoT Data Services

In this study, we will delve into the provisioning of cloud-based IoT data services as a narrowed field of research at the intersection of the following research lines. DaaS is a new delivery model for data provisioning to serve data consumers irrespective of their geographic locations [8], [9]. An IoT service is merely regarded as a transaction between parties in an IoT-based architecture [2]. As such, an IoT data service aims to provide end-users with transaction-based data (e.g., data about soil moisture in farms and air quality data). Such a transaction might be enabled by a cloud platform, giving a rise to the notion of cloud-based IoT data services. Quality of data is arguably a critical subject in this research line. For instance, Badidi [10] proposed an algorithm to evaluate the DaaS providers based on the quality-of-data requirements specified by the data consumers for finding appropriate DaaS providers. Mišura [11] created a model of the data market to analyze query performance and utilities gained by the consumers. They concluded that the data market in IoT is an effective method to distribute measurements and data. Recent work in this research direction contributes to the concept of IoT ecosystems and data services [12] [13]. These methods mostly rely on a centralized solution, namely service registry, which has long been investigated in the Web-based service-oriented architecture. Existing work on data marketplaces attempts to associate quality information with its data sources but the SLA and contracts are mainly formulated at the marketplace to cope with costs and data volumes [14].

B. Aggregator Business Model

Many studies suggest that the IoT is expected to change business processes and pose new challenges [15]. They point out a major shift from viewing IoT primarily as a technology platform to seeing it as a business ecosystem, moving from focusing on a business model to designing an ecosystem. Other studies, such as Dijkman [16], proposed a business model for IoT applications based on a template called Business Model Canvas [17]. They have expanded and adapted this business model by identifying necessary building blocks and types of the key IoT participants. Newly-emerging business models for IoT have created a new opportunity for operating data services on the cloud platform of the service providers, which are materialized by the so-called Aggregator Business Model [18]. The idea behind this model is to integrate individual component services to create complex ones using cloud computing. Service aggregations are quite ubiquitous and can be found in business-to-business and business-to-consumer markets for products, services and information [19]. In services science – the cousin discipline of SOC, this topic is investigated under the terms of services bundling and service constellation. High-level services might be bundled/constellated to create innovative service offerings that, for instance, are competitively priced and bring the consumers convenience [20]. For example, tourists who take a travel package (as a service bundling) would benefit from discount while being hassle-free in securing accommodation and booking their flights [5].

As for IoT data provisioning, the notion of data marketplaces has been around for a while with an emphasis on sensor data [21]. However, these data marketplaces do not focus on geographically sparse IoT data providers in aqua/agriculture domains.

C. Monitoring and Aggregating SLAs

Managing and monitoring the SLAs is crucial for maintaining the trustworthiness of a service system or an ecosystem of the IoT services. Comuzzi et al. [22] seek to regulate the provisioning of services by operating an architecture that analyzes the execution log to monitor SLAs. Other work is concerned with describing SLAs needed for compliance monitoring of the application environments involving infrastructure, platform, and application services [23].

When composing services, we need to aggregate the SLA of the sourced services. This kind of aggregation has been studied in the context of business processes [24], composite services [25] and cloud computing [26]. For IoT data services, evidences of how an SLA is respected at run-time might be buried in the massive volume of data provisioned and the accessibility to the service provider's computing cloud. We are in need of an SLA-handling mechanism that plays a refereeing role by monitoring of, but not interfering with, data communication between the said service provider and its end-users.

III. Research Motivation

A. Case study

The Mekong Delta region of Vietnam displays a variety of physical landscapes, watercourses interlaced. Shrimp farming is an important business domain in the Mekong Delta. This business domain consists of companies, private shrimp farms, and many other stakeholders, which have benefited the economy. However, the productivity of farms depends on two main factors: water quality and climate change. Thus, finding a solution to monitor these issues is a crucial task. There are many research institutes, companies and organizations involved in developing solutions, software and applying new technologies to increase the production of shrimp and to reduce the risk due to the climate and water environment changing. However, data collection is difficult and costly, especially when data need to be collected in different locations in the Mekong area. Therefore, we build a case study of a hypothetical company called `MekongDataVendors` which has a platform that supports research units finding data from shrimp farming and family farms. So, it helps gather information easier and monitor the quality of water and the climate change in shrimp farms better. Moreover, bringing IoT technology closes to farms and farmers creates an opportunity for farmers, agricultural engineers, and experts work together. They, being considered the end-users in this scenario, seek data services from multiple vendors all of whom are connected thanks to the platform of company `MekongDataVendors`. These data services could be invoked from a custom software product developed for farming monitoring. In other words, the end-users rely on software development with services while service vendors commit themselves to software development for services.

TABLE I: IoT data services being provisioned thanks to an SLA-aware platform of a company located in the Mekong delta called `MekongDataVendors`

Label	Data Services	Provider	Multi-level	Data 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

Table I makes a list of data services being offered in the said platform. Each service is uniquely identified by a label in the form of so_1, so_2, etc. presented in the leftmost column. As for functional aspects, a service comes with a short name that captures the kind of data to be delivered and a provider who is registered to the service platform of company `MekongDataVendors`. In this case-study, 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 I, we describe the contractually formulated, multi-level commitments between these service providers and the end-users. To keep it simple, we show no more than three levels for these service agreements. The levels, named basic, intermediate and advanced, simply state that the more frequent data is delivered to the end-users, the higher monthly costs incurred. Geographically speaking, P^1_{cloud} is located in Kien Giang, $P^2_{in-house}$ in Kien Luong, $P^3_{in-house}$ in Hon Dat, P^4_{cloud} in Ha Tien. These proper names represent the rural districts of Kien Giang Province in the Mekong delta of Vietnam.

Service providers $P^2_{in-house}$ and $P^3_{in-house}$ are farmers who deploy IoT-enabled sensors in their own farming sites that are located in one of the above-mentioned rural districts. They invest in IoT technologies for monitoring their agricultural production. These providers decided to put their data of agricultural production for sale via the service platform of company `MekongDataVendors`. They rely on on-site sensors to provide the end-users with instantaneous readings of the pH and the alkalinity levels for a localized area. Due to the lack of computing power and storage, historical levels and statistics of the pH and alkalinity might not come in handy. 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. Being hi-tech agencies, P^1_{cloud} and P^4_{cloud} obtain the appropriate farm owners' permission to operate their IoT devices, networking infrastructure and cloud servers in certain farming areas. Their business is to collect large dataset of agricultural production out of which money might be made thanks to the `MekongDataVendors`'s service platform.

As shown in Figure 1, `MekongDataVendors` connects the end-users with the aforementioned service providers. The end-users in this scenario include IoT application developers, research organizations and governmental agencies. The first group makes use of the services listed in Table I to compose service-oriented applications (e.g., wearable , web-based, mobile) for farming monitoring using, for instance, services so_1 and so_2. Thanks to this service platform, the second group should be able to obtain agricultural data for scientific purposes using, for example, services so_3 and so_6. The third group takes the advantage of the data obtained through services so_4 and so_5 to fine-tune the agricultural extension for family farm or fisheries development in the area. The actual data services registered in this platform might be a lot more than what are simplified and exemplified in Table I, giving the end-users multiple service offerings. For example, a research lab interested in pH readings would prefer so_6 to so_3 as the former offers slightly more advanced agreement level than the latter.

Eventually, the end-user programmers who develop a service-oriented application for water monitoring will ask for data services that are not listed in Table 1. One way to meet this demand is to aggregate data services that are functionally related (e.g., so_1 and so_2 are all about water) to provide more sophisticated data for a given location (e.g., reporting on the water temperature and water conductivity of a shrimp pond). so_1 and so_2 together as an aggregation of data services would programmatically come in handy, enabling a practice widely known as software engineering with services. Aggregating so_1 and so_2 not only involves mixing their water-related data but also results in their multilevel SLAs being compiled and rationalized.

B. Research Statement

The `MekongDataVendors`'s business model described in Subsection III-A is in fact not brand new. Böhm et al. discuss the so-called aggregator business model [18] for IT provisioning in cloud computing. We bring this concept to the newly-emerging field of IoT data services. As illustrated in Figure 1, IoT data service providers are agencies who produce data to be provided in the form of re-usable services, which enable software engineering with DaaS thanks

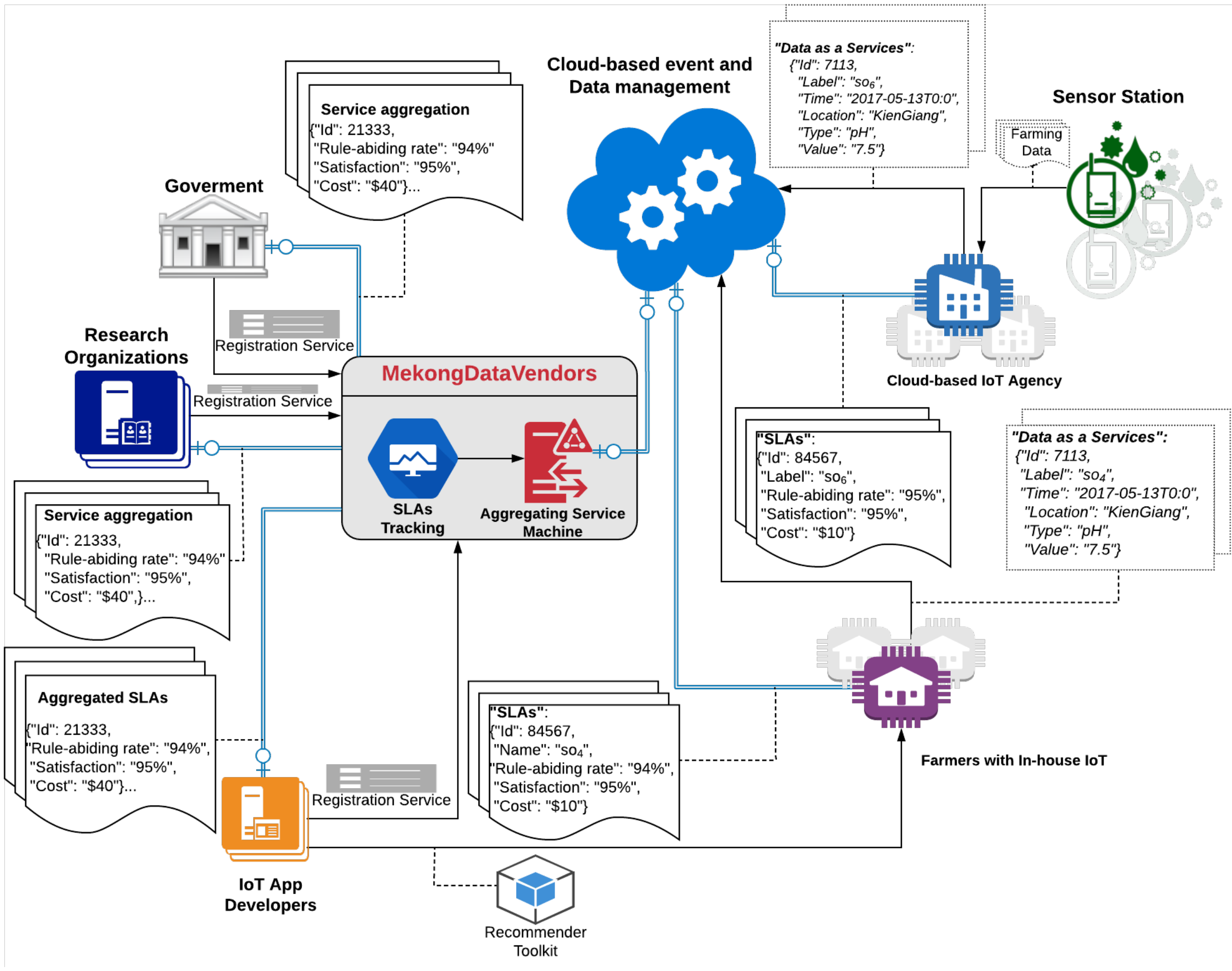

Fig. 1: An aggregator business model for IoT data services in the Mekong Delta.

to `MekongDataVendors`, which serves as the aggregator in this model.

We propose a new solution – IoT data services as a newly-emerging platform for service delivery – for an old problem – the aggregator business model in services computing [18]. In our model, we distinguish between a cloud-based IoT agency and an in-house IoT station (see Table II). They use sensors to collect data provided through the computing environment of the service. The former operates its sensors over a relatively large area. In contrast, the sensors operated by the latter are usually installed in a small area. In terms of data delivery, the cloud-based IoT agency relies on a computing cloud to perform data conditioning, logging and calculating statistics (e.g., min/max/mean values) and data enrichment (e.g., interpolation and extrapolation). The in-house IoT station comes with small computational power and almost only offers instantaneous data. These data are usually raw data so the price may be lower. As for business goals, the cloud-based IoT agencies produce data for sale while the in-house IoTs may use their data for agricultural purposes.

As exemplified in Subsection III-A, data services are subject to aggregation. Our research objectives in this line of work are twofold. First, data services might be aggregated in a number of ways to create additional service offerings, giving rise to the composition of SLAs. Whether the SLA of an aggregated (data) service can be deducted from the SLA of its constituent services remains an open question. Second, individual service offerings and aggregated ones should be treated equally in such an aggregator platform when it comes to the service registry. End-users rely on a special look-up

TABLE II: Two different kinds of IoT data service providers that participate in the aggregator business model

Cloud-based IoT Agency	In-house IoT Station
Equipments installed in relatively large area	Equipments installed in a localized area
Do not own any farm in the area over which it collects data	Owns a farm in the area over which it operates
Is computationally powered for data conditioning, data enrichment and handling big volume of data over a substantial amount of time	Thin computing power for transferring instantaneous data
Enriched data	Spontanous data
Data is solely for sale	Data used for agricultural purposes and for sale.

technique of this registry to get access to the service they wish to consume. Such a look-up technique will yield an ordered list of service offerings that all meet a certain end-user's data request. The order that is effectively in place, enabled by a function that assigns each element of the list a score, is supposed to bring the service that best matches the end-user's expectation to the front of this short-lived list. Such a scoring function needs to be formally and algorithmically defined by means of SLAs registered in the said service platform.

In our previous work, we propose a way to reason about SLA [27] in terms of satisfaction, price and penalty rules (i.e., rules stating what a service provider has to do in case it fails to deliver its commitments). In this work, we focus on IoT-enabled data services and take a slightly different approach to model their SLA. Specifically, we propose that the third SLA item of a data service is about the extent to which its provider respects the penalty rules that are functionally declared for the service in question (see Table III for an illustration).

IV. SLA-aware Machinery for Handling SLAs of IoT Data Services

This section presents our formal machinery for reasoning about service level agreements when aggregating and ranking data services (Subsection IV-B). We begin this section by giving some formal definitions (Subsection IV-A).

A. Using Semiring to Represent Service Level Agreements

A semiring is a mathematical structure having two operations that matters on a set. Let's consider a semiring $\mathcal{A} = \{\langle r, s, c\rangle |\ r \in \mathcal{R},\ s \in \mathcal{S},\ c \in \mathcal{C}\}$ that represents all SLAs of a service ecosystem, where $\mathcal{R}, \mathcal{S}, \mathcal{C}$ are the sets of rule-abiding rates, satisfactions and costs respectively in the following. We first recall the definition of semiring [28].

Definition 4.1: 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 over IoT data services in Subsection IV-B. Now, let's define the notions of rule-abiding rate, satisfaction and cost in the following.

Definition 4.2: A data service is associated with several penalty rules, which are supposed to be respected in 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 kicked off. Formally, we have *rule-abiding rate* $= 1 -$ *breach rate*. 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 concerned.

Example 4.1: In Table I, we assume that the rule-abiding rates in the multilevel SLA of service so_5 are $r_{11} = 0.95$, $r_{12} = 0.96$ and $r_2 = 0.97$ where rule r_1 is refined into r_{11} and r_{12}. Therefore by definition: $r_1 = \min(r_{11}, r_{12}) = \min(0.95, 0.96) = 0.95$. In the same way, the rule-abiding rate of so_5 is the minimum value of r_1 and r_2, as shown in Table III.

TABLE III: Breakdown of the rule-abiding rates in the SLA declared for one of the data services presented in Table I

so_5 **rule's description**	**Rule-abiding rates of a rule**
r_1: The designated stations must send data uninterruptedly within the agreed-upon timeframe.	**0.95**
r_{11}: If data is missing or the latency is higher than 15 minutes, customers will receive one hour of service for free.	0.95
r_{12}: If the latency is higher than 30 minutes, the service in question should be operated by another station for which the customer will enjoy up to 2 hours of data provisioning for free.	0.96
r_2: The yearly cost of monitoring the salinity in the area should not be more than two thirds of the equipments costs and rent charges combined.	**0.97**
Rule-abiding rate of so_5	**0.95**

Definition 4.3: Informally, the satisfaction of a service perceived by the end-users is about the reliability and empathy of this service [7]. For IoT data services, the **satisfaction** is formally defined as the ratio of the number of successful data delivery to the total number of service transactions.

Definition 4.4: Let $\mathcal{C}_0 = \{c_1, c_2, ...c_n\}$ be the initial set of values of the cost incurred by data services. By **closure** of $\mathcal{C}_0$, we mean the smallest set containing all finite summation of elements in $\mathcal{C}_0$: $\mathcal{C}_0^+ = \{\sum_{k=1}^{\infty} (c_{i_1} + ... + c_{i_k})|\ c_{i_k} \in \mathcal{C}_0\}$. As such, the **cost set** is defined as $\mathcal{C} = \mathcal{C}_0^+ \cap [0, cost_{max}]$, where $cost_{max}$ is the highest cost that the customer may pay. A **cost** (denoted as c) is a payment for a data service, which is an element of set $\mathcal{C}$.

B. Aggregating and Ranking Data Services by Their SLAs

We proceed in aggregating the SLAs of sourced services. Suppose that we have n services and consider the i-th service for which we write $L_i = \{\ell_{ij} | j = 1, ..., m_i\}$ where ℓ_{ij} represents the the j-th level of its multilevel SLA.

Now, let's put $\alpha^i_{\ell_{ij}}$ to denote the j-th level of the i-th service's multi-level 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 4.5: To represent the multi-level SLA of a service aggregation, consider a group of aggregated services as subset of k services $\{so_i |\ i \in I \subset [n], |I| = k\}$ each of which comes with multi-level SLA as exemplified in Table I. Out of this subset, the Cartesian product of the multi-level 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.

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

TABLE IV: Multi-level SLAs of the IoT data services listed in Table I

Label	**Data service**	**Multi-level**	**SLA** $\langle r, s, c\rangle$
so_1	Water temperature	Intermediate	$\langle 93\%, 94\%, \$5\rangle$
		Advanced	$\langle 96\%, 98\%, \$10\rangle$
so_2	Water pollution	Basic	$\langle 91\%, 90\%, \$10\rangle$
		Intermediate	$\langle 95\%, 92\%, \$20\rangle$
		Advanced	$\langle 98\%, 93\%, \$25\rangle$
so_3	pH	Basic	$\langle 90\%, 93\%, \$3\rangle$
so_4	Alkalinity	Basic	$\langle 94\%, 95\%, \$10\rangle$
so_5	Salinity	Intermediate	$\langle 94\%, 98\%, \$20\rangle$
		Advanced	$\langle 95\%, 98\%, \$30\rangle$
so_6	pH	Intermediate	$\langle 95\%, 95\%, \$10\rangle$

Example 4.2: Table IV gives details of the SLA for data services described in Table I. From this table, 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 $L_1 = \{\ell_{11} : Intermediate,\ \ell_{12} : Advanced\}$, $L_4 = \{\ell_{41} : Basic\}$ and $L_5 = \{\ell_{51} : Intermediate,\ \ell_{52} : Advanced\}$. Out of this subset, the Cartesian product of these set is represented by a 2-dimensional array of size $(\prod_{i\in I} m_i)\times 3$, which is as follows.

$\langle r^1_{\ell_{11}}, s^1_{\ell_{11}}, c^1_{\ell_{11}}\rangle$	$\langle r^4_{\ell_{41}}, s^4_{\ell_{41}}, c^4_{\ell_{41}}\rangle$	$\langle r^5_{\ell_{51}}, s^5_{\ell_{51}}, c^5_{\ell_{51}}\rangle$
$\langle r^1_{\ell_{11}}, s^1_{\ell_{11}}, c^1_{\ell_{11}}\rangle$	$\langle r^4_{\ell_{41}}, s^4_{\ell_{41}}, c^4_{\ell_{41}}\rangle$	$\langle r^5_{\ell_{52}}, s^5_{\ell_{52}}, c^5_{\ell_{52}}\rangle$
$\langle r^1_{\ell_{12}}, s^1_{\ell_{12}}, c^1_{\ell_{12}}\rangle$	$\langle r^4_{\ell_{41}}, s^4_{\ell_{41}}, c^4_{\ell_{41}}\rangle$	$\langle r^5_{\ell_{51}}, s^5_{\ell_{51}}, c^5_{\ell_{51}}\rangle$
$\langle r^1_{\ell_{12}}, s^1_{\ell_{12}}, c^1_{\ell_{12}}\rangle$	$\langle r^4_{\ell_{41}}, s^4_{\ell_{41}}, c^4_{\ell_{41}}\rangle$	$\langle r^5_{\ell_{52}}, s^5_{\ell_{52}}, c^5_{\ell_{52}}\rangle$

For each given row consisting of 3 SLAs, as we already know the fixed level, we put $< r^i_{\ell_i}, s^i_{\ell_i}, c^i_{\ell_i} >$, $i \in I$. Hence, we rewrite the first row as follows.

$\langle r^1_{\ell_1}, s^1_{\ell_1}, c^1_{\ell_1}\rangle$	$\langle r^4_{\ell_4}, s^4_{\ell_4}, c^4_{\ell_4}\rangle$	$\langle r^5_{\ell_5}, s^5_{\ell_5}, c^5_{\ell_5}\rangle$

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

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

We repeat this task for each remaining row in the array. As a result, Table V 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.

TABLE V: Aggregating IoT data services listed in Table I results in the compilation of their SLAs.

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)	$\langle 94\%, 95\%, \$50\rangle$

To enable ranking techniques, we aim to make the set of all SLAs, denoted as $\mathcal{A}$, a totally ordered set. Formally, we define this total order as: $\Omega_i \geq \Omega_j$ if $(r_{\Omega_i} > r_{\Omega_j})$ or $(\mathrm{r}_{\Omega_i} = \mathrm{r}_{\Omega_j}) \wedge (\mathrm{s}_{\Omega_i} > \mathrm{s}_{\Omega_j})$ or $(\mathrm{r}_{\Omega_i} = \mathrm{r}_{\Omega_j}) \wedge (\mathrm{s}_{\Omega_i} = \mathrm{s}_{\Omega_j}) \wedge (\mathrm{c}_{\Omega_i} \leq \mathrm{c}_{\Omega_j})$ where r: rule-abiding rate, s: satisfaction, c: cost. The relation "$\geq$" defines a total ordering over the set $\mathcal{A}$. We define the $\oplus$ operation as the max operation with respect to this order.

The $\otimes$ operator is the multiplication acting on each component an element in the set A differently. 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 4.3: Table VI shows an ordered list of service aggregations. Ω_3 receives scores first in this list.

TABLE VI: Suggesting data services from the user requirements

Rank	**Label**	**Service aggregation**	**Combined SLA** $\langle r, s, c\rangle$
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)	$\langle 93\%, 94\%, \$45\rangle$

C. Algorithms

Algorithm 2 materializes the total order defined (see Subsection IV-B). Algorithm 1 presents the algorithmic logic for ranking IoT service offerings that were previously selected based on end-user's requirements. In Algorithm 1, we make sure that all service offerings that have been selected should come with well-defined SLAs before ranking them. Techniques for selecting services and aggregating services are however out of the scope of this paper.

Algorithm 1: Ranking IoT data services that match end-user's intention by their SLAs

Input: $reqs$: end-user's requirements;
Output: An ordered list of services where finest ones should be in the beginning;

begin
    $L_{serv} \leftarrow$ select IoT data services and service aggregations that match end-user's intention described in $reqs$;
    foreach $rs \in L_{serv}$ **do**
        if *rs represents a service aggregation **and** rs comes with an undefined SLA* **then**
            Compute the SLA of rs in accordance with formula 1 in Definition 4.5;
        end
    end
    Sort L_{serv} according to the comparison function defined in Algorithm 2;
    return L_{serv};
end

A service offering selected in list L_{serv} of Algorithm 1 is either an individual data service or a service aggregation. We mind penalty rules and analyze logs to determine the rule-abiding rate and the satisfaction of an individual service. As for service aggregations, we rely on Definition 4.5 to establish their multilevel SLA.

Algorithm 2: Comparing IoT data services by their SLAs

Input: sla_1, sla_2: A pair of SLAs to be compared;
Output: an integer value of either: +1 if sla_1 is better than sla_2; or zero if sla_1 is equivalent to sla_2; or -1 if sla_1 is worse than sla_2;

begin
    if *($sla_1.rule\text{-}abiding > sla_2.rule\text{-}abiding$)* **then**
        return +1;
    end
    if *($sla_1.rule\text{-}abiding < sla_2.rule\text{-}abiding$)* **then**
        return -1;
    end
    if *$sla_1.rule\text{-}abiding = sla_2.rule\text{-}abiding$* **then**
        if *$sla_1.satisfaction > sla_2.satisfaction$* **then**
            return +1;
        end
        if *$sla_1.satisfaction = sla_2.satisfaction$ **and** $sla_1.cost < sla_2.cost$* **then**
            return +1;
        end
        if *$sla_1.satisfaction = sla_2.satisfaction$ **and** $sla_1.cost = sla_2.cost$* **then**
            return 0;
        end
        return -1;
    end
    return -1;
end

V. Tracking IoT Data Services

Figure 2 depicts how IoT data services are requested and bound to end-users in our case-study. Services $\{so_1, .., so_6\}$ are diagrammatically enclosed in a cloud area. Each of them is represented under a rounded rectangle having three compartments. The bottom compartment of a service is dedicated to its provider. Little stars in the cloud area stand for service aggregations $\Omega_1, .., \Omega_4$ (fully described in Table V). This figure features two scenarios. The first scenario, enumerated as $\{1.1 - 1.2 - 1.3 - 1.4\}$, illustrates Algorithm 1. More specifically, a request for an IoT service is submitted in JSON format that includes the following items (step 1.1): the province of Kien Giang as a location, monthly cost of \$50, data required (water temperature, alkalinity, etc.). Step 1.2 is about selecting service offerings that match this end-user's requirement; step 1.3 – ranking. Finally, `MekongDataVendors` suggests appropriate service offerings in step 1.4.

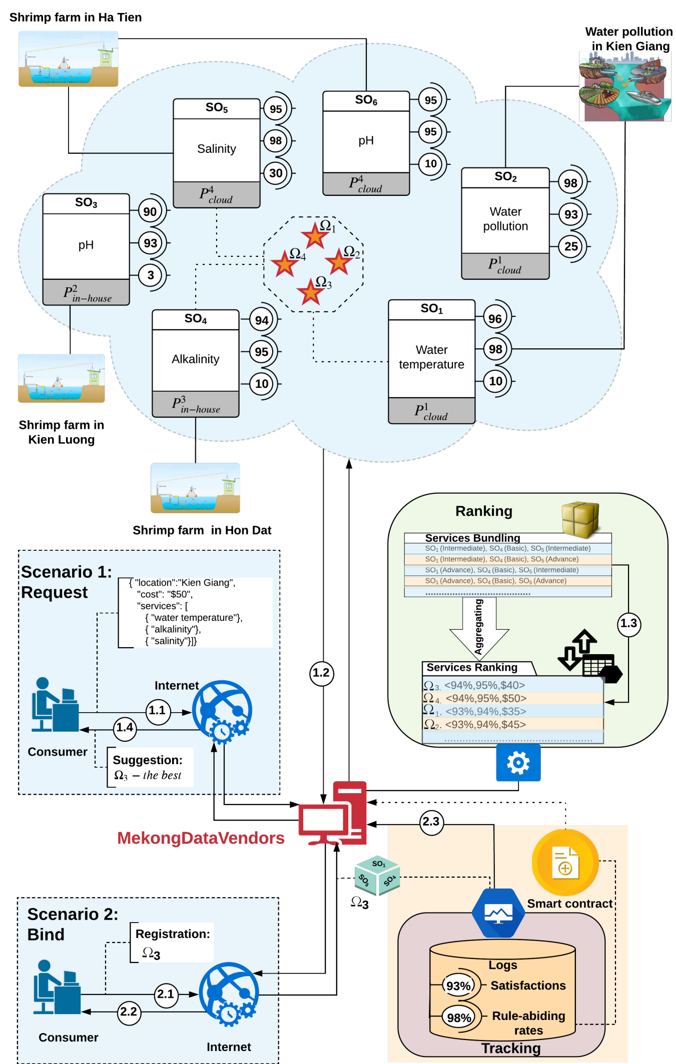

Fig. 2: `MekongDataVendors`'s platform responds to end-users' request of IoT service offerings. Logs in this platform capture the history of rule-abiding and the success/failure of data delivery of an individual IoT data service.

The second scenario, identified as $\{2.1 - 2.2 - 2.3\}$, explains the data delivery of IoT services being bound to the

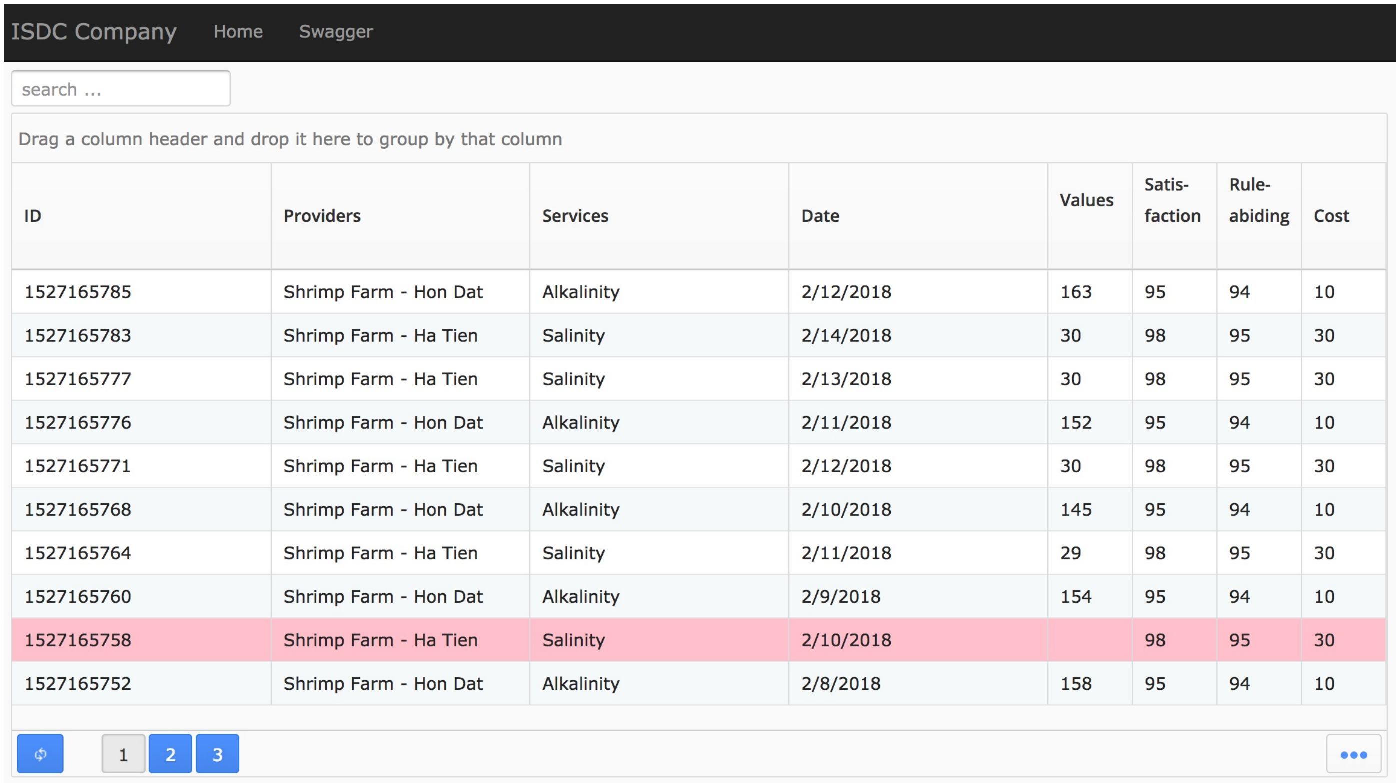

Fig. 3: Data provisioning thanks to `MekongDataVendors`'s platform

end-user and a corresponding tracking mechanism. The service platform of `MekongDataVendors` keeps tracking the success/failure of data delivery and whether relevant penalty rules are respected in run-time. This kind of log, together with blockchain-based protocols, automatically compute the SLAs of individual IoT services. In this sense, steps 2.2 and 2.3 might be performed in parallel.

Both IoT services and end-user applications are populated in the Microsoft Azure cloud platform. We collected datasets of water sampling in agriculture for the period of Feb 2017 – Dec 2017 in the province of Kien Giang. The datasets are populated in a SQL Server database system. In our future work, we plan to move the SQL server to a large-scale data retrieval system such as Google BigQuery.

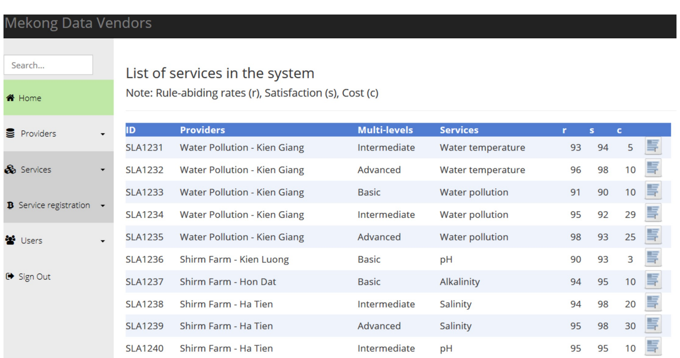

Fig. 4: Individual IoT data services registered in `MekongDataVendors`'s platform.

We have implemented a prototype that tracks data provisioning in the form of IoT data services. Figure 4 shows a dashboard window displaying IoT services being registered in `MekongDataVendors`'s platform.

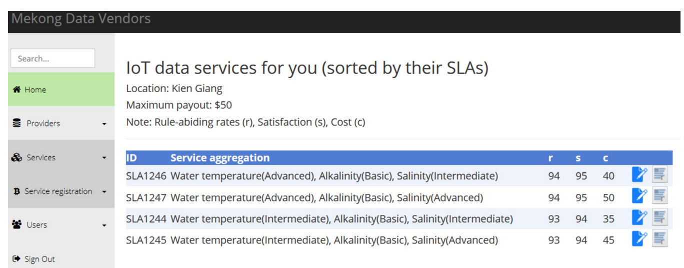

Fig. 5: Service aggregations that have been sorted by their SLAs come in handy for the end-users.

Figure 5 demonstrates an ordered list of service aggregations. Figure 3 is a screenshot of an end-user application that reports data provisioning from the end-user's perspective. Each row in this screenshot represents a data item being delivered. The highlighted row stands for a failed data delivery (note that it has an empty data cell).

VI. Conclusions and Future Work

We expect that the deployment of lightweight IoT data services will be increasing due to the availability of sensors and platforms for delivering IoT data. For the best exploitation of the service ecosystem of IoT data, data end-users need an effective look-up and binding mechanism not only to find out the right service but also to materialize a communication channel between service parties. Despite a large number of service-oriented platforms for data offerings being in operation, quality-aware provisioning of IoT services still stays on the sidelines. This paper focuses on IoT-enabled data services of which SLA plays a central role in determining if a data service offering is fine enough from the end-user's perspective. Given a large number of data service offerings in operation (and they might be aggregated in a number of ways to create even more offerings), our work helps the end-users access the finest services that meet their requirements.

Does handling the SLA equal minding rules plus analyzing service logs? We formally define the SLA of IoT data services and propose how they could be logically compiled in service aggregations. We argue that this kind of the SLA is the mixture of satisfaction, payment and the extent to which its service provider abides by penalty rules when provisioning data. We propose a tracking mechanism to measure the customer's satisfaction and rule-abiding rate defined in the SLA of individual service offerings. We devise a new approach to compiling the multilevel SLA of a service aggregation from those of sourced services. We present algorithms for looking up the "finest" service offerings in terms of the SLA. We demonstrate our work using real-life data of agricultural production in the Mekong Delta of Vietnam. In the near future, we will investigate a blockchain-based architecture for trust, provenance and payment of the IoT service providers with respect to SLA-handling.

ACKNOWLEDGMENT

Work presented in this paper is funded by Ho Chi Minh City University of Technology (VNU-HCM), under grant number C2018-20-09. The second author would like to thank his former colleague, Aditya Ghose, for his introduction to using semiring for formally capturing non-functional concepts in services computing.

REFERENCES

[1] T. Chou, *Precision: Principles, Practices and Solutions for the Internet of Things*. Lulu, 2016.

[2] M. Thoma, S. Meyer, K. Sperner, S. Meissner, and T. Braun, "On IoT-services: Survey, Classification and Enterprise Integration," in *IEEE International Conference on Green Computing and Communications*. IEEE Computer Society, November 2012, pp. 257–260.

[3] A. Bouguettaya, M. Singh, M. Huhns, Q.-Z. Sheng, H. Dong, Q. Yu, A.-G. Neiat, A. Mistry, B. Benatallah, B. Medjahed, M. Ouzzani, F. Casati, X. Liu, H. Wang, D. Georgakopoulos, L. Chen, S. Nepal, Z. Malik, A. Erradi, Y. Wang, B. Blake, S. Dustdar, F. Leymann, and M. Papazoglou, "A Service Computing Manifesto: The Next 10 Years," *Communications of the ACM*, vol. 60, no. 4, pp. 64–72, April 2017.

[4] M. Commerce, "Internet of Things (IoT) Service Level Agreements: Market Outlook and Forecast for IoT SLAs 2017–2022," Business Wire, Tech. Rep., February 2017.

[5] L.-S. Lê, T.-V. Nguyen, T.-M. Truong, and K. Nguyen-An, "Contractual Specifications of Business Services: Modeling, Formalization and Proximity," *Transaction of Large-Scale Data and Knowledge-Centered Systems*, vol. 31, pp. 94–123, 2017.

[6] G. Fick and B. Ritchie, "Measuring Service Quality in the Travel and Tourism Industry," *Journal of Travel Research*, vol. 30, no. 2, pp. 2–9, 1991.

[7] W. Kettinger and C. Lee, "Perceived service quality and user satisfaction with the information services function," *Decision sciences*, vol. 25, no. 5-6, pp. 737–766, 1994.

[8] H. L. Truong and S. Dustdar, "On analyzing and specifying concerns for data as a service," in *Proceedings of the 4^{th} Asia-Pacific Services Computing Conference*. Singapore: IEEE Computer Society, 2009, pp. 87–94.

[9] J. A. Olson, "Data as a Service: Are We in the Clouds?" *Journal of Map & Geography Libraries*, vol. 6, no. 1, pp. 76–78, 2009.

[10] E. Badidi and H. Routaib, "A DaaS Based Framework for IoT Data Provisioning," in *Proceedings of 1^{st} Conference on Computational Methods in Systems and Software*. Szczecin, Poland: Springer, 2017, pp. 369–379.

[11] K. Mišura and M. Žagar, "Data marketplace for Internet of Things," in *Proceedings of the 1^{st} International Conference on Smart Systems and Technologies*. Osijek, Croatia: IEEE Computer Society, 2016, pp. 255–260.

[12] S. Kubler, J. Robert, A. Hefnawy, K. Främling, C. Cherifi, and A. Bouras, "Open IoT Ecosystem for Sporting Event Management," *IEEE Access*, vol. 5, pp. 7064–7079, 2017.

[13] M. Papert and A. Pflaum, "Development of an ecosystem model for the realization of internet of things (IoT) services in supply chain management," *Electronic Markets*, vol. 27, no. 2, pp. 175–189, 2017.

[14] T.-D. Cao, T.-V. Pham, Q.-H. Vu, H.-L. Truong, D.-H. Le, and S. Dustdar, "MARSA: A Marketplace for Realtime Human Sensing Data," *ACM Transactions on Internet Technology*, vol. 16, no. 3, p. 16, 2016.

[15] M. Westerlund, S. Leminen, and M. Rajahonka, "Designing Business Models for the Internet of Things," *TIM Review*, vol. 4, no. 7, July 2014.

[16] R. Dijkman, B. Sprenkels, T. Peeters, and A. Janssen, "Business models for the Internet of Things," *International Journal of Information Management*, vol. 35, no. 6, pp. 672–678, 2015.

[17] A. Osterwalder, Y. Pigneur, and T. Clark, *Business Model Generation: A Handbook for Visionaries, Game Changers, and Challengers*. Wiley, July 2009.

[18] M. Böhm, S. Leimeister, C. Riedl, and H. Krcmar, "Cloud Computing – Outsourcing 2.0 or a new Business Model for IT Provisioning?" *Application Management: Challenges – Service Creation – Strategies*, pp. 31–56, 2011.

[19] D. Tapscott, A. Lowy, and D. Ticoll, *Digital Capital: Harnessing the Power of Business Webs*, 1st ed. Harvard Business School Press, May 2000.

[20] L.-S. Lê and A. Ghose, "Contracts + Goals = Roles?" in *Proceedings of 31^{st} International Conference on Conceptual Modeling*. Florence, Italy: Springer, October 2012, pp. 252–266.

[21] F. Stahl, F. Schomm, G. Vossen, and L. Vomfell, "A Classification Framework for Data Marketplaces," *Vietnam Journal of Computer Science*, vol. 3, no. 3, pp. 137–143, 2016.

[22] M. Comuzzi, C. Kotsokalis, G. Spanoudakis, and R. Yahyapour, "Establishing and monitoring SLAs in complex service based systems," in *Proceedings of the 2^{nd} IEEE International Conference on Web Services*. California, USA: IEEE Computer Society, 2009, pp. 783–790.

[23] H. Ludwig, K. Stamou, M. Mohamed, N. Mandagere, B. Langston, G. Alatorre, H. Nakamura, O. Anya, and A. Keller, "rSLA: Monitoring SLAs in dynamic service environments," in *Proceedings of the 13^{th} International Conference on Service-Oriented Computing*. Goa, India: Springer, 2015, pp. 139–153.

[24] T. Unger, F. Leymann, S. Mauchart, and T. Scheibler, "Aggregation of Service Level Agreements in the Context of Business Processes," in *Proceedings of 12^{th} International Conference Enterprise Distributed Object Computing*. Munich, Germany: IEEE Computer Society, September 2008, pp. 43–52.

[25] A. Ludwig and B. Franczyk, "COSMA – An Approach for Managing SLAs in Composite Services," in *Proceedings of 6^{th} International Conference on Service-Oriented Computing*. Sydney, Australia: Springer, December 2008, pp. 626–632.

[26] A. Stanik, M. Körner, and O. Kao, "Service-level agreement aggregation for quality of service-aware federated cloud networking," *IET Networks*, vol. 4, no. 5, pp. 264–269, 2015.

[27] T.-V. Nguyen, L.-S. Lê, and K. Nguyen-An, "Aggregating Service Level Agreements in Services Bundling: A Semiring-Based Approach," in *Proceedings of the 3^{rd} International Conference on Future Data and Security Engineering*. Can Tho, Vietnam: Springer, 2016, pp. 385–399.

[28] S. Bistarelli, *Semirings for Soft Constraint Solving and Programming*. Springer, 2004, vol. 2962.