International Journal of Advanced Research in Social Sciences, Environmental Studies & Technology
p-ISSN: 2536-6505 | e-ISSN: 2536-6513
Volume 8, Number 1 April, 2023

IJARSSEST

Towards Compliance Assurance with Data Protection Regulations in SaaS Applications

Hammajam Ahmed Adamu
Department of Information Technology,
Modibbo Adama University, Yola Adamawa State

Article DOI: 10.48028/iiprds/ijarssest.v8.i1.08

Abstract

Enforcing regulatory compliance relating to security and privacy in Software as a Service (SaaS) applications have increasingly received much attention recently from industries such as the oil and gas because of potential benefits SaaS applications could bring to it. However, regulations such as European Union General Data Protection Regulation (GDPR) and the Nigerian Data Protection Regulation (NDPR) aimed at ensuring the protection of the rights of data owners, are legal documents and therefore, unable to ensure the enforcement of compliance. In this paper, we propose a policy language capable of defining and expressing data protection policies at each level of the data lifecycle for unambiguous compliance enforcement in SaaS applications. We propose new extensions of the Prime Life policy language to create the Security and Privacy-PPL extensions that can help to enforce compliance. Lastly, we show proof of usability of our proposed policy language using smart gas station-showing interactions between all the data actors and how compliance can be enforced using the SP-PPL.

Keywords: *Cloud, Compliance, Data, Nigeria, Data protection regulation, Privacy, Security, and Software as a servic*e

Corresponding Author: Hammajam Ahmed Adamu

Background to the Study

Security and privacy compliance in Software as a Service (SaaS) applications are increasingly becoming relevant across all sectors. In particular, software applications continue to play critical roles in the oil and gas industry from exploration, transportation, to decision making, attraction of essential skills, strategy formulation and to the entire retail ecosystem, thus, implementing SaaS cloud solutions become highly relevant for this industry.

Additionally, recent developments in cloud computing have changed the means to which software is delivered and consumed. Thus, presenting the industry with several unique advantages such as scalability and pay as you go pricing model.Despite the benefits of SaaS applications [1], they still suffer from several significant drawbacks such as [2] governance control, compliance with established laws and data protection regulations, trust, identity and access management, software isolation, and incident response [3] and others relating to security and privacy of personal data. [4].

Against these challenges, debates continue about the best strategies for overcoming these challenges relating to the handling of personal data handling practices, location of data and other issues such as multitenancy in SaaS applications [5], this is coupled with a significant rise in data nationalism on the control of data within national boundaries [6]. To that effect, data protection regulations have been enacted by governments, regional bodies and industry organisations such as the European Union General Data Protection Regulation (GDPR) [7] and a similar version implemented in Nigeria called the Nigerian Data Protection Regulation (NDPR) [8]. These regulations have a significant impact on the operational activities of organisations within the oil and gas industry.

Data protection regulations are rules meant to carry out a specific piece of legislation relating to personal data handling, and they are made and maintained by an authority [9]. The regulations are unique to their purposes and countries of origin, but according to [10] in many cases, they are similar and designed to achieve local needs. For example, the GDPR is similar to the NDPR in many areas such as their aim to guarantee reliable protection of the personal data of natural individuals from organisations who collect, use and share information. Other areas of similarities include their scopes, how they define key terms, their legal basis, their recognition of rights and enforcement. Conversely, they are different in other areas, especially in enforcement and implementation. For example, the GDPR is implemented by member countries of the EU through an independent body such as the Information Commissioner's Office '[11] in the United Kingdom, while the NDPR is implemented by the Federal Government Ministry of Communication [12] via the Nigerian Information Technology Development Agency (NITDA) [13]. In this research work, we consider the NDPR while baselining it to the GDPR.

The NDPR [8] and the EU GDPR regulations contain the rights of persons who are living and whose data are processed. On the other hand, these regulations also specify responsibilities for data controllers who handle the data and the rights of the data subject [14]. Because of these regulations impact, organisations are now concerned about compliance

when deciding to adopt software technologies such as SaaS applications. Decision-makers weight the advantages of SaaS applications against issues of compliance relating to data protection and privacy of personal data before migrating into the SaaS model. Recently, British Airways was fined £183.39 million pounds, and Marriott International Inc. ‍[15] was fined 99 million pounds for a data breach under the GDPR ‍[16]. Similarly, in Nigeria, the government licensed some data protection organisations and charged them with the responsibility of ensuring compliance with the NDPR to avoid personal data breaches within Nigeria [17]. For example, NITDA recently investigated the activities of Truecaller for the alleged violation of the privacy rights of Nigerians [18]. True caller is a service that helps to identify incoming calls from unknown numbers [19] efficiently.

Consequently, the concern for compliance has created a situation where organisations in the Oil and Gas Industry in Nigeria are beginning to include compliance into their strategic plans. To put things in the right perspective, within the context of this research, we define compliance as a means enforcing rules that implement the policies relating to the protection and privacy of data based on the preferences of data subjects and service providers as provided for in NDPR and GDPR.

As compliance is now a legal obligation under item Section 1.2 of the NDPR, organisations especially service providers are required to key in data privacy in Section 2.5 and security in Section 2.6 into their processes to ensure compliance. However, with the take-off of the NDPR in Nigeria, there is a significant challenge for its compliance by SaaS service providers, as the regulation is textual and can be misinterpreted and therefore making compliance challenging. Therefore, effective compliance remains a challenge to licensed data protection organisations due to the lack of enforcement mechanisms in Nigeria,

In order to solve the compliance challenge in a cloud environment such as the SaaS model, several solutions have been advanced including policy languages such as XACML [19], PPL [20], A-PPL [21], CPPL [22] with either a generic application or bespoke to some specific contexts to help with the compliance enforcement. However, we have identified some drawbacks of these languages when applied to SaaS applications: (i) the languages lack flexibility (ii), designed with defined set of requirements and application domains and therefore not suitable to express the requirements of every regulation as contemplated in Section 4.1 of the NDPR regulation.

Furthermore, looking at it from a technical perspective, all these extensions did not use any formal method to design these policies and therefore leaving room for ambiguity. Our approach relies on the advantage of using formal approaches to design our compliance policies so it can be easily verified mathematically unlike other languages such in [23]

To mitigate these drawbacks, we propose a policy language for ensuring compliance with data protection regulation for the protection of personal data by SaaS providers. The proposed policy language will extend the PPL language and can be used to enforce compliance with similar data protection regulations within in a SaaS setting. Specifically, our contributions in

this paper are:

1. We provided a comprehensive review of related studies with an emphasis on compliance and policy languages in a distributed environment with a focus on the SaaS applications.
2. We mapped the security and privacy obligations of the NDPR and aligned them to the data life cycle for ease of enforcement.
3. We present SP-PPL, the proposed security and privacy policy language, as an extension of the PPL designed for SaaS application based on the requirements of the NDPR.
4. Finally, we show the validity and the applicability of SP-PPL within the context of SaaS applications.

Data Protection Regulations

Sion et al. [24] described data protection as a genuinely interdisciplinary effort involving many stakeholders such as legal experts, requirements engineers, software architects, developers, and system operators. Data protection laws require controllers to comply with their numerous provisions when it comes to the processing of personal data. Arguably, building software-intensive systems, including SaaS applications, which respect the fundamental rights to privacy and data protection, is a result of intensive dialogue and trade-off decisions, particularly in the area of compliance. Wayne Jansen et al. [25] defines compliance as the responsibility of an organisation to work in observance of relevant laws and regulations, standards and specifications, in this case, data protection regulations. In Section I above, we indicated that we consider data protection regulations such as the GDPR, NDPR, which we consider to be relevant to this work.

Data Protection Regulations Provisions

From the legal perspective, data protection regulations are for legal experts and subject to legal and ambiguous interpretations. These interpretations further compound the challenges to their compliance. For example, the NDPR and the EU GDPR aim to ensure the protection of personal data by requiring compliance by organisations who are involved in the collection, usage, storing and forwarding of personal data and they apply to contexts where personal data was either collected online using electronic means or offline. While the NDPR came into effect in April 2019, the EU GDPR became effective in May 2018. Both regulations share very similar objectives. The NDPR's sole objective is to safeguard the privacy of data relating to natural persons in Nigeria. It sought to achieve this by ensuring that all transactions involving the transfer of personal data are free from manipulation. The regulation also stipulates a penalty for violation, with 2% of Annual Gross Revenue for data controllers who handle personal data of at least 10,000 data owners and 1% for data controllers handling less than 10,000 data owners.

On the other hand, the EU GDPR primarily aims to protect the privacy of residents within the geographical space of the EU by regulating how personal data is handled and processed by organisations in their operations. The EU is a significant economic bloc; therefore, the GDPR's reach, and impact became global. As highlighted above, the two regulations bear

similarities in many areas such as their objectives, definitions of terminologies, scope, and safeguards to the rights of natural persons to data privacy. However, they are different in ways such as enforcement mechanisms and authorities, child rights, and penalties where violations occur. In the following, we discuss these similarities and differences.

Scope

The scope of GDPR is global as it applies to organisations outside of EU handling personal data of EU citizens, while the NDPR's impact applies to any organisation handling the personal data of Nigerians, hence, it certainly does not have the reach of the GDPR [8]. Furthermore, **Articles 3, 4 (1) of t**he GDPR and Part 1 (1.1) of the NDPR clearly stated that the regulations only protect and safeguard the rights of individuals and not legal persons or entities. Therefore, in terms of scope, the two regulations are reasonably consistent.

Key Definitions of Terms

On definitions of critical terms such as personal data, the data subject, data controller, data processor and child, both the GDPR and NDPR have similar definitions but differ on two key areas such as the definition of a child and data processor. While the role of a data processor is the same as the data controller in the NDPR, the GDPR classifies them as two separate roles.

1. ***Personal data:*** Personal data refers to any form of data that can be associated with a living person such as full name, national ID numbers, phone numbers, IP addresses, email addresses directly or indirectly [17], [26] [5]. Data Subject refers to an identifiable person. In this case, a person who can directly or indirectly be identified by referring to a number or any specific characteristic such as social, economic, physical and cultural or other identifiers such as IP addresses and email addresses[27].
2. ***Data Controller:*** A data controller refers to a person or an entity that handles personal data relevant to their operations. A data controller can refer to a legal person, authority or body that defines the reasons and channels of processing of personal data.
3. ***Data Processor:*** the GDPR described the data processor role as that of a natural or legal person who processes personal data in the interests of a data controller [28]. The GDPR sees this as an entirely different role or party in the handling of personal data. While the NDPR describes the data processor role as the same as that of a data controller, it went on to describe liabilities of a data processor or controller in Section 5 of the NDPR to include all third parties who handle personal data on their behalf.
4. ***Child definition*****:** The GDPR recognises children as natural persons who are vulnerable and require special data protection. This special protection should have applicability to marketing or services tailored for children[29]. Conversely, the NDPR does not have any special recognition of children as natural persons requiring special data protection. Instead, it imposes all provisions on all-natural persons.

Rights

Both the GDPR and NDPR uphold the right to be forgotten. The right to be forgotten provides a clear guideline on how the data subjects can request the erasure of personal data. For the sake of data protection, data controllers or service providers must comply with demands for erasure unless if there is an overriding interest such as listed in Art. 17(3)[30].

The scope of this right to erasure extends to third parties who may have processed data on behalf of a data controller. The exercise of this is free. Other similar rights guaranteed by both regulations include the rights to data portability, information, to object, to not be subject to discrimination, and the right to access personal data.

Enforcement

As enforcement is the actual process of ensuring compliance with regulations, the GDPR and NDPR regulations made provisions for monetary penalties to be issued where there is non-compliance. However, the penalties, procedures and amounts differ significantly.

Monetary penalties*:* The GDPR charges 2% of global annual turnover, or 10 million euros, whichever is higher or 4% of global turnover or 20 million euros, whichever is higher. Similarly, the NDPR regulation charges 2% of the annual gross of a preceding year or 10 million naira's. The penalty applies to data controllers processing more than 10,000 data subjects. As for data controllers processing less 10,000 data subjects, the charge is 1% of annual gross revenue or 2 million naira's. Therefore, the regulations differ in what they charge in monetary terms and the percentage of revenue of the organisation.

Supervisory authority*:* Article 51 of the GDPR provides for an independent authority to implement the provisions of the regulation by member states of the EU. The authority is vested with the responsibility of assisting organisations in understanding their obligations and compliance. On the contrary, the NDPR does not have any provisions for the establishment of an independent monitoring authority but mandated the NITDA to oversee the application of the NDPR.

Civil Remedies

In order to persuade or coerce relevant parties to take responsibility, provisions for civil remedies are in the GDPR as well as the NDPR. It affords individuals with cause to seek redress for violations of their privacy or the privacy of their data. According to the GDPR's **articles, 79 – 82, recitals 141 -147[5], a violation is a justifiable cause to start legal action. Similarly,** in Section 4.2 of the NDPR regulation, the right to seek for redress is affirmed while the Agency retained powers to set up an investigative Administrative Redress Panel to investigate violations[8].

Data Life Cycle

According to Butin et al. [31], personal data protection can only be beneficial when organisations implement protection policies at each stage that makes up the entire data life cycle within the context of personal data. Data life cycle is the sequences that a unit of personal data goes through from when it is initially collected, to how and where data is stored. Furthermore, how organisations share data with third parties and how it is deleted or erased. They went on to argue that regulations help set our obligations, and these obligations can help in the exercise of responsibility and the verification of handling practices by organisations handling personal data.

Following this, our proposed policy language specifies policies on the stages of data life cycle for activities such as data collection, policies enforcing storage preferences such as preferred location, policy for enforcing usage of collected data, and lastly, deletion, and forwarding of data to third parties.

1. ***Data Collection****:* The NDPR in Section 2.3 requires data controllers to make their purposes known before they collect personal data. It also requires them to make known how they intend to process the data. Furthermore, to avoid where data is collected arbitrarily, the regulation sought to limit data collection to the purpose for which it is collected[17].
2. ***Data usage:*** In Sub-section 2.1.a of the NDPR, data controllers are requested to collect data according to a clear lawful purpose. This purpose is to be communicated and have the consent of the data subject. A service provider's policy should have indicated details on (i) consent for use, (ii) purpose and (iii) who will use the data.
3. ***Data storage:*** The NDPR declared a policy relating to data storage in Sub-section C where the storage of personal data is within a period which it is reasonably needed. In a situation where a specific type of data is held in storage by a data controller or SaaS service provider, (i) the location of storage must be known, (ii) must be in a secure storage infrastructure and (iii) a periodic review on why the personal data is in storage.
4. ***Data deletion:*** On data deletion rights or the right to erasure, as in the case of the GDPR, the NDPR in section 3.1 (9) also described the deletion rights that data subjects can exercise. Therefore, by implication, service providers are required to have mechanisms or provisions for the enforcement of these rights by creating policies relating to deletion and retention of data with particular attention to details such as (1) who is authorised to delete data on a service providers storage, (ii) what type of data is the authorised personal allowed to delete or retain and (iii) whether the deletion has a period of delay.
5. ***Data forwarding****:* As service providers continue to depend on each other to provide services to users, the need to share data amongst them becomes critical. Therefore, to ensure effective regulation and the rights of data subjects, data protection regulations require that for data forwarding to be permissible consent must be obtained. In Sections 2.11 and 2.12 of the NDPR, data forwarding criteria described how a service provider or a data controller could transfer personal data to a third-party recipient. Some of the conditions are (i) consent provided by the data subject, (ii) an unambiguous purpose stating why the data will be forwarded and (iii) a list detailing the recipients of the personal data.

Review of Related Work

We give an overview of relevant works in this section. We emphasize compliance and policy languages within the context of data security and privacy protection regulations in SaaS applications.

Compliance

Compliance in the cloud today is a challenging subject, to that effect, several solutions and regulations advanced to help with the challenge of compliance to data protection regulations

such as the GDPR and NDPR. The US National Institute of Standards and Technologies defines compliance [25] as the responsibility of an organisation to work in observance of relevant laws and regulations, standards and specifications.

Recently, many countries and governmental bodies have implemented new types of regulations relating to data protection with applicability at different levels of government, such as national jurisdictions, regional and/or local governments. Thus, making compliance a highly complicated concern for cloud service providers (i.e., data controllers) and by implication, SaaS applications.

We, therefore, define compliance as being capable of enforcing the rules that are required to implement policies in regulations such as GDPR and NDPR. Furthermore, [1] described that compliance is one of the most critical requirements for many software systems; this is also true with SaaS. Compliance with the GDPR is mandatory, and failure to comply will lead to penalties [32], this is also true for the NDPR. Some existing research has provided data protection techniques to support compliance with regulations. For example, Jayasinghe [27] proposed a GDPR trust-based compliant framework for data controllers, Ta [6] proposed a formal design and conformance check of personal data protection policies and architectures, Yu et al. [33] proposed a technical framework that can help generate snapshots that are verifiable. These snapshots they assert can be used as data trails to help track disclosure of personal information.

Additionally, Al-Zaben [34] proposed an architecture relying on blockchain technology to help manage personal identifiable information. The architecture relies on a local database and blockchain ledgers that are not within the same location to help preserve the privacy of personal data. Elluri *et al.* [35] advanced an integrated ontology that is semantically rich. This ontology shows in detail the representation of data protection regulations rules such as the GDPR and other regulations. They argue that the ontology can help ensure compliance as the data regulations are only available in textual format.

Other forms of compliance with particular reference to SaaS applications was put forward by Indhumathil [36], they proposed third-party auditing in SaaS applications and cloud computing. They went on to argue that there is a relationship between the hesitation of organisations to use cloud-based services such as SaaS applications due to privacy, security and reliability concerns. Similarly, Lins et al. [37] argue that in order to increase the trustworthiness of cloud-based services, the practice of auditing continuously of carefully selected criteria will help assure users of the security of their data in the cloud.

However, although these solutions and technologies may be useful to enhance the compliance to data protection regulations, they do not suggest how the enforcement of compliance can be possible within the context of SaaS applications. Thus, these approaches are limited in their approaches and unable to provide information on how compliance with data protection regulations can be enforced by relying on the preferences of the data subject with the aid of by policy languages.

Policy Languages

Henze et al. [22], described a policy language as the formalisation and expression of privacy and security policies into machine-readable languages. A policy language must fulfil a certain number of vital requirements to help enforce policies in environments such as distributed environments as with the SaaS applications. The requirements are (i) minimal or least storage footprint, (ii) efficient policy checking, (iii) expressiveness, (iv) extendibility, (v) incremental deployment and (vi) matching.

In order to enforce compliance with security and privacy preferences of data subjects, policy language is required. Consequently, we discovered a handful of policy languages proposed for modelling and enforcing privacy, security, and access control rules. Similar to the authors in [12], authors in [6] described security policy language as languages used in the formulation of rules and the enforcement of policies with relation to confidentiality and availability of data. It also refers to the integrity of properties of data. On the other hand, a privacy policy language is used for creating rules that are capable of preserving and safeguarding the privacy of personal data [37]. For example, an access control policy language such as the XACML is foundational but has remained relevant today.

XACML, as indicated, is an access control centred policy language. It has an inbuilt request/response language for effective two-way communication [19]. Furthermore, it consists of standard XML elements and standard extension points for specific rules, different data types and procedures. Due to its standardisation based on OASIS, there exists many implementations and extensions for access control rules. For example, these extensions includes profiles for usage control [38], privacy policy [39], PPL [40], A-PPL [21].

Enforcing access control was the sole reason for developing The Prime Life Policy Language (PPL) [40] using certified credentials before granting access. These credentials include attribute and role-based access control [41], [4] systems. The PPL extension of the XACML introduced a very effective means of enforcing rules using the concept of the obligations and enforced by a matching engine using a combination of a trigger and an action to execute the obligations.

Another policy language is C2L [42]. The C2L aims at enforcing configurations that are permissible in a cloud environment. The language uses spatiotemporal logic which enforces these permissible configurations by providing policy constraints statements on colocations, hosting, security and migration of data and others such as availability of data. However, C2L only focused on formalism and not real-life application in a live cloud environment.

Similarly, [23] extended the PPL Language and proposed an accountability framework to improve the safety and accountability of data by service providers in handling personal data. Using an abstract policy language, they expressed the data subject's preferences and service provider's obligations in a human-readable fashion, thus, achieving ease in mapping to the enforcement of policies [23]. The A-PPL [43] is a follow up to [23] as an extension of the PPL designed to express machine-readable accountability policies as opposed to the human-

readable language. A-PPL can define accountability and transparency rules on personal data handling using the developed extensions on data retention, data location, logging and notification.

Another extension if the PPL is CPPL [22], the Compact Privacy Policy Language that enforces the privacy of personal data by compressing privacy policies using flexibly specialised domain knowledge. In addition to the contributions and extensions to PPL by [42], [44], [23], [22], [43], the authors in [45] presented a set of fundamental requirements that must be met by cloud providers or service providers to satisfy the accountability requirements of their customer's data. They outlined several tools for an accountability-based approach such as Payment Card Industry Data Security Standard [46]

Another policy language is PriArmor [47]. It proposed to work with the Infrastructure as a Service (IaaS) model of the cloud. The PriArmor approach allows data subjects to express their privacy preferences aligned to regulations using an ontology model that includes all concepts on data access and usage within a distributed environment such as the cloud.

However, all these PPL extensions discussed above are focused on some specific aspects of security and privacy enforcement and the research projects for which they were proposed to address. Among these extensions, A-PPL extension is the closest to this work, but with focus on accountability properties and the role of an auditor, the C2L focused on the formalism of requirements on spatiotemporal modal logic, the PriArmor extension purely focused on the IaaS, and the CPPL focused on the storage of data.

Limitations of the Existing Policy Languages

Based on the analysis of existing policy languages in [48], and other domain-specific PPL extensions such as A-PPL [49], C-PPL [22], we choose to extend the PPL as the most suitable policy language, due to its extensibility, and contextual application. Therefore, we will create new PPL extensions to build the SP-PPL.

A major weakness of all these PPL extensions is that they did not envisage the provisions and compliance requirements imposed by new regulations such as the GDPR and NDPR. They also were not focused on the SaaS model of the cloud, but they are all built on a significant advantage of the PPL, which is that it is extensible to the new realities of recent regulations in the areas of expressing, designing and implementing compliance. In this work, as a contribution, we intend to extend the PPL to satisfy and implement the provisions of the NDPR as it is a regulation within the immediate operating environment of organisations in the case study.

Table 1: Comparison of policy languages

Policy Language	Security	Privacy	Expressiveness	Extensibility	Policy matching	Compliance checking
XACML	-	-	-	✓	-	-
PPL	-	✓	-	✓	✓	-
C2L	-	✓	-	-	-	-
A-PPL	-	✓	✓	✓	✓	-
CPPL	-	✓	-	-	✓	-
PRIAMOR	-	✓	-	-	✓	-
PDC	-	-		✓	✓	-
SP-PPL	✓	✓	✓	✓	✓	✓

Table 1: Comparison of policy languages. A Language fulfils this requirement (✓), a Language does not fulfil this requirement (-).

The System Model

SaaS application service providers collect user's data to allow for the use of their applications or services, thus creating the need to ensuring compliance with data handling regulations. In order to understand the intuition behind this work, we will in this section, present a SaaS scenario to help put into context how data control is transferred from the data subject to the service provider, we will go on to provide a set of generic requirements that a standard policy language must address.

Scenario

As indicated above, we consider a scenario where data is collected and transferred out of the data subject's control to the infrastructure of the service provider. The data could be used in manners that may conflict with the data handling preferences of the data subject thus raising security and privacy concerns [41] thus resulting in a loss of control over own data [1]. In the past, many approaches have been put forth to mitigate these concerns such as the sticky policies in the works of [19] which attempts to enforce user preferences to data in order to enforce access control. Generally, these policy languages aim to enforce the privacy preferences of the data subject on all data handlers based on some specific data protection regulation.

In the following, we start by mapping the security and privacy compliance requirements in the NDPR before developing our PPL extension SP-PPL.

Mapping of Compliance Requirements in the NDPR

Following the analysis of the obligations of data controllers as set out in the NDPR, the proposed policy language would be able express data handling rules that correspond to the collection, usage, storage, deletion and retention, and data forwarding. In addition to the requirements extracted from the NDPR, in order to successfully extend the policy language to meet our needs, we also considered some the generic requirements put forward by Henze et al [22] such as minimal storage footprint, efficient policy checking, expressiveness, extend ability and ease of deployment. Additionally, this is backed by our analysis of the data protection and

handling expectations of the NDPR. We found out that the expectations of the NDPR requires that the proposed policy language should be able to express data compliance rules relating to collection, usage, retention and storage location to which the PPL [23] language fulfils some of the requirements (e.g., access control and privacy). Moreover, it should also be extensible enough to fulfil the requirements of the NDPR in the context of SaaS applications. However, the limitation of the requirements is by no means exhaustive, but it satisfies the contextual objective of checking compliance to data protection regulations in SaaS applications.

Requirement #1 Data Collection

While the importance is attached to the entire states of the data lifecycle, specific areas become very vital at the collection stage of personal data. At this stage, organisations are generally faced with the challenges relating to consent and opt-out rights [14]. This is because one of the key issues relating to data collection is consent when processing personal data. Although the data subject approves to the processing, consent may be withdrawn at any given time, thus adding to the compliance challenges where a third-party processing has already taken place. Accordingly, this is a requirement Section 2.3 (a) of the NDPR, which describes how the purpose of data collection and processing must be known to the data subject before the collection of personal data to authorise or give consent to the processing of data.

Requirement #2 Data Usage

In order to use SaaS applications, organisations collect and share data with service providers. This raises a challenge particularly of conflicting interest relating to usage and thus raising data handling concerns on how the service provider and other third parties may use the collected personal data. In this scenario, service providers may want to use the data for other purposes by employing other legal means such as in the subscription or service agreements. Other challenges relating to the usage state of the data life cycle is the profiling and tagging of personal data and subsequent exploitation as part of Big Data analysis. To address these concerns, Section 2.1 Sub-section (a, b) of the NDPR requires that data is collected with consent and for clearly defined specific, legitimate and lawful use purpose as a contract to which the Data Subject is party or in order to take steps at the request of the Data Subject before entering into a contract.

Requirement #3 Data Storage

To achieve service availability of SaaS applications data, SaaS service providers depend heavily on continuous data copy and backup, which usually contain personal and usage data. While the availability and integrity of the data is the aim of this continuous backup, it comes with the challenges of compliance to consent collection, especially where third party storage services are involved. To mitigate this challenge, data should only be held in storage within the time which it is reasonably needed. Accordingly, the NDPR in part two Section 2.1, subsection (c) stipulates the period within which personal data may be held within a reasonable period of time needed to store data. This is to help protect personal data and minimise the amount of data collected and held in storage.

Requirement #4 Data Deletion and Retention

At the end of data processing, personal data should be deleted. Data deletion in a cloud computing environment such as SaaS applications may occur for several reasons, for example, an exit strategy, the lack of lawfulness or legitimacy of processing. While data destruction is an essential part of sustaining compliance, it comes with several challenges, such as technical and complexities mainly due to multi-cloud environments of different providers requiring different data deletion processes towards different types of requests.

Interconnectivity of systems is further seen as a challenge, as deletion of data in one system may cause ripple effects in other systems. Other challenges include the *Right to Erasure* due to the ambiguous nature of this right, and some organisations are not sure what erasure means. If there is no legal basis for the continued processing of personal data, deletion is required. Simply delinking the data with a data subject can be seen sufficient to fulfil the right to the deletion request. Other situations may require a complete deletion of the data where data is unlawfully processed (i.e., processed without the consent of the data subject). Regardless, complying with request for data deletion requires tracking down each instance of the personal data collected and copied to other locations.

Furthermore, verifying deletion to demonstrate compliance may become even more challenging in SaaS applications, owing to transparency issues. Organisations collecting personal information need to be able to handle constant requests to maintain compliance as required by data protection regulations such as in Section 3.1 of the NDPR regulation, and sub-section (9) stipulating the rights of the data subject regarding the deletion and retention of data.

Requirement #5 Data Forwarding

As the last stage of the data lifecycle, data forwarding happens when a service provider transfers or share data with a third-party during processing. During the processing of personal data by SaaS service providers, some risks and challenges emerge as data is shared and forwarded to third parties in the course of processing or service delivery. Challenges such as trust and the verification of compliance with relevant data protection regulations. In order to prove compliance, service providers should be able to deploy mechanisms for verification performed by data subjects. However, demonstrating compliance, such as deletion, may be obstructed by the lack of transparency of the service providers. Other compliance challenges relating to data forwarding include how secure the components or infrastructure used for the transfer. In order to avoid any violation of the rights of data subject with relation to how data is transferred and shared between data controllers, Sections 2.3 sub-section (e) and 2.12 of the NDPR states that a data controller is under obligation to ensure that consent of the Data Subject has been obtained without fraud, coercion or undue influence in situations where personal data may be transferred to a third party for any reason whatsoever.

Limitations of the Ppl

As discussed in our review of the PPL in **Section III**, the PPL was proposed to specify machine-readable privacy policies by building on the XACML [19]. This was achieved using

extensions by defining a new syntax for obligation and authorisation. Within the PPL, an obligation is expressed using the pair Trigger-Action. Triggers are events related to an obligation and filtered by conditions[49]. Triggers fire actions such as personal data collection performed by a data controller. Some examples of PPL triggers and actions are[20]:

a) Triggers = condition + event (e.g., within 7days after the deletion of an address, or when using a phone number for calling purpose, within 3 hours of using a phone number for calling purpose,) e.g., trigger constructs:

TriggerPersonaDataDeleted (address, 7 days)
TriggerPersonalDataAccessedForPurpose (phone, {call}, default)
TriggerPersonalDataAccessedForPurpose (phone, {call}, 3 hours)

b) Actions (e.g., delete an address, anonymize name and address, notify Pete via email) e.g., action constructs:

ActionDeletePersonalData({address})
ActionAnonymizePersonalData({name, address})
ActionNotifyDataSubject (email, {Tukur})

Authorisations define the actions that the data controller is allowed or prohibited to perform such as (a) authorisation for usage purposes and (b) authorisation for data forwarding to third parties. Although several requirements such as access control, privacy and usage can be handled by the PPL, it has no definitions for compliance check to data protection properties at granular levels policies at each stage of the data life cycle.
Given the above limitation of PPL, we propose in section VI our Security and Privacy compliance policy language SP-PPL that extends the PPL at each stage of the data life cycle within the context of SaaS applications. SP-PPL will rely on the PPL architecture and to express the requirements. Furthermore, we use a new step showing a formal application of the SP-PPL to the end-to-end data life cycle within the context of SaaS applications.

Introducing Sp-Ppl Policy Language

We propose SP-PPL our security and privacy policy language for the specification of preferences of the data subjects, and preferences of service providers for the purposes of compliance checking and assurance. The language will have, as provided in the regulations, a data subject, and a data controller as actors. This is consistent with the earlier versions of the PPL. SP-PPL will be able to present policies at each level of the data life cycle as detailed in **Section II** above. This can be used to define policies across all the stages of the data life cycle within the context of SaaS applications.

To the best of our knowledge, this is the first work in the context of security and privacy compliance with data protection regulations in the context of SaaS business applications. The SP-PPL syntax and the corresponding application of the proposed policy language with compliance regulations such as the NDPR, and the data life cycle makes the proposed policy language versatile.

SP-PPL Syntax

In this section, we present the extensions added to the PPL to create the SP-PPL while still maintaining the original PPL structure. The SP-PPL identifies the following roles: Data subject, Data processor and Data controller and third-party data controller. These roles already exist within the PPL architecture. For the NDPR and compliance checking, the following syntax is proposed: Unlike the PPL, the following is our syntax for enforcing data protection regulations such as the NDPR and GDPR.

1. $\mathrm{P}col = (cons,\ cpurp),\ with\ cons\ \in \{Y,\ N\}$. *Y- Yes, N- No*
2. $\mathrm{P}use = (cons,\ upurp, whouse) cons \in \{Y,\ N\}$.
3. $\mathrm{P}str = (wh, ho, revdate),\ \ decl \in \{Y,\ N\}$.
4. $\mathrm{P}del\ (placedel, when, decl)$, where $decl\ \in\ \{Y, N\},)$
5. $\mathrm{P}fw = (cons,\ fwpurp, 3rdparty),\ \ decl \in \{Y,\ N\}$.

Our policy specifications are inspired by a number of works including[6] but with an extended and a combination of formal and XML expressive syntax to express the data protection properties for personal data handling in SaaS applications. Our language is adaptable to any data controller and applicable to recent data protection regulations such as the GDPR and NDPR. The syntax of the language will be expressed as:

Going forward, a set of policies focused on the protection and privacy of data will be defined and aligned to the data life cycle in SaaS applications. The syntax for the proposed security and privacy protection policies are defined as tuples and aligned to the data life cycle and presented below in Table 1.

Table 1: Data lifecycle policy sets

S/N	END-TO-END DATA LIFE CYCLE	POLICY DEFINITION EXPRESSION
1	Data collection	*PolCol*
2	Data usage	*PolUse*
3	Data storage	*PolStr*
4	Data deletion or retention	*PolDel*
5	Data forwarding	*PolFw*

Definitions

The syntax for the security and privacy policies are defined as the following tuples. We define the set of high-level data protection policies (*POL*), which are composed of the five sub-policies based on the end-to-end data life cycle. Namely, the sets of data collection ($Pcol$)), data usage ($\mathrm{P}use$), data storage ($Pstr$), data deletion ($\mathrm{P}del$), data forwarding policies ($\mathrm{P}fw$).

This is further expressed as:

1. $POL = PolCol\ x\ PolUse\ x\ PolStr\ x\ PolDel\ x\ PolFw$

2. $Pcol = (cons, cpurp)$

3 P$use = (cons,\ usepurp, whocanuse)$

4. P$str = (wherestore, howstore)$

5. P$del = (placedel, when, decl)$

6P$fw = (cons,\ fwpurp, 3rdparty, decl)$,

1. Data collection policy includes consent required ($Pcol$) for certain data types (e.g., personal data) and collection purposes ($cpurp$).
2. Data usage policy Puse specifies the consent ($cons$) for data usage and the purpose of data usage ($usepurp$), and the set of entities who are allowed to use the data ($whocanuse$). Data storage policy.
3. The data storage policy Pstr specifies where the data is stored ($wherestore$) how the data is stored ($howstore$).
4. The data deletion policy Pdel specifies the location where the data ($placedeldecl$) is deleted ($when$), $when\ = (rdelay = dd)$ with defined retention period and $when\ = (gdelay = gd)$ the global retention delay .
5. Finally, the data forwarding policy Pfw policy involves the consent $cons$ and forwarding purpose $fwpurp$ the third-party entities to which the data will be forwarded ($3rdparty$) , $decl$

Each data protection policy is designed on a data type (θ), precisely, let $P\_typeset1, P\_typesetk \in POL)$, be a policy defined on a data type θ, and the five policies relating to the data life cycle $Pcol \in Polcol,\ Puse \in PolUse, Pstr \in Polstr, Pdel \in PolDel, Pfw \in PolFw,$ where

$$Pol = (Pcol, Puse, Pstr, Pdel, Pfw)$$

In the policy expressions above, we have defined policies around all the data life cycle for data collection, ($Pcol$), data usage ($Puse$), storage of data ($Pstr$), data deletion ($Pdel$) and the forwarding of data, (Pfw).

1. $Pcol = (cons, cpurp), with\ cons \in \{Yes, No\}$. Here we specify that a consent is required to be collected from users when collecting their personal data. With Y denoting Yes, and N denoting No. $cpurp$ = (P_typeset. Pcol, *decl*), where P_typeset. Pcol represents the set of purposes in the for data collection *typeset*, while *decl* $\in \{Y, N\}$. *decl* specifies if the collection purposes which can take the value of Y (Yes) or N (No).

2. $Puse(cons,\ usepurp, whocanuse, decl)$ Here, we specify sub policies for usage to require consent, for the usage of personal data and denoted as $cons$. The usage purpose $usepurp$ and $whocanuse$ denoting who is allowed to use the data, including third-party entities expressed as $cons \in \{Y, N\}$ A policy for consent collection. $usepurp$ a policy for usage purpose. $whocanuse$ a policy for who is allowed to use the data. $where\ decl \in \{Y, N\}$

3. $Pstr = (wherestore, howstore, decl)$, $where\ decl \in \{Y, N\}$, Here we propose a policy for data Storage specifying where data is stored and denoted as $wherestore$, and $howstore$ the method with which data is stored. $wherestore$, a set of places where the *data* is stored, such

3. $Pstr = (wherestore, howstore, decl)$, $where\ decl \in \{Y, N\}$, Here we propose a policy for data Storage specifying where data is stored and denoted as $wherestore$, and $howstore$ the method with which data is stored. $wherestore$, a set of places where the *data* is stored, such as in the service provider's (*SPstorage*) servers. *wherestore* ∈ {(*SPstorage*, 3rdpartyStorage, *decl*)}. $wherestore$ storage locations where data is stored. $howstore$ = (Available, Hidden) where $howstore$ is where data encrypted with the service provider's key and content is available to the service provider (howstore = "available") and *decl* specifies the declaration of this information for users. $decl = \in \{Y, N\}$

4. $Pdel = (placedel, when, decl)$, where $decl \in \{Y, N\},$) Here we specify $placedel$ as a set of elements ("$mainstorage$"," $backupstorage$, "$3rdparty$"). $when = (rdelay = dd)$ as a *numerical* retention delay value, namely, the semantics for the policy $Pdel = (placedel, (rdelay = rd, gdelay), decl)$, where $(placedel, gdelay\ and\ decl)$, can have any possible/defined value. And we went on to specify $when = (gdelay = gd)$ where gd is a *numerical* worst-case delay value, namely, the semantics for the policy $Pdel = (placedel, (rdelay, gdelay = gd), decl)$, where $placedel$, $rdelay$ and $decl$ can have any possible/defined value (in the language alphabet). $decl \in \{Y, N\}$.

5. P$fw = (cons, fwpurp, 3rdparty, decl)$, $where\ decl \in \{Y, N\}$, cons = {Y, N}, The policy for Data Forwarding requires consent for the data forwarding denoted as ($cons$) and list of forward $fwpurp$, a list of $3rdparty$ and declarations $decl \in \{Y, N\}$ which takes the value of Y-(YES) and (N-NO).

Assuming a finite set {typeset1, ….., typeset$m$} for all the data types that are supported by the SaaS application provider. Therefore, the data protection policy for the SaaS service provider is defined as: $SPPL = Pol$ = {P_typeset1, …, P_typesetk}

The proposed end-to-end SP-PPL, therefore, enables a fine-grained policy specification and compliance checking on the entire data life cycle that is also consistent with the provisions of the NDPR.

Semantics of the Policy Language

Here we extend the syntax of the PPL showing Service Provider and User point of views or preferences.

1) Service Provider preferences

We define policy for each data typeset.

P_typeset = (Pcol, Puse, Pstr, Pfw, Pdel), so we will have

Pol = {P_typeset1, …, P_typesetk}, where typeset1, …typesetk are different set of types (no common element)

1. P_typeset. Pcol -> reference to collection policy for type *type*
2. P_typeset. Puse -> reference to usage policy for type *type*
3. P_typeset. Pstr -> reference to storage policy for type *type*
4. P_typeset. Pdel -> reference to deletion policy for type *type*
5. P_typeset. Pfw -> reference to forwarding n policy for type *type*

User or Customer Preferences

We define Customer requirement for each data type.

R_typeset = (Rcol, Ruse, Rstr, Rfw, Rdel), so we will have

REQ = {R_typeset1, …, R_typesetk}

1. R_typeset. Rcol -> reference to collection policy for type set *typeset*
2. R_typeset. Ruse -> reference to usage policy for type set *typeset*
3. R_typeset. Rstr -> reference to storage policy for type set *typeset*
4. R_typeset. Rdel -> reference to deletion policy for type set *typeset*
5. R_typeset. Rfw -> reference to forwarding policy for type set *typeset*

Obligation Language

Relying on the existing PPL language, we enforce compliance by comparing security and privacy compliance obligations using extended triggers and actions. In this context, triggers serve as events considered by the obligation engine. The obligation engine accepts this as a set of actions such as a Do **Action** when a particular **Trigger** is called. These triggers are events that are considered by an obligation and are seen as the set of events that result in actions such as Do **Action** when **Trigger.**

Compliancy and Matching Rules

In order to match and compare SaaS service provider's security and privacy policy preferences with the data subject's preferences, the engine will have to match based on the rule that a SaaS service provider security and privacy policy is less (or equally) permissive than data subject's security and privacy preferences as captured in the NDPR.

For example, in order to collect data, a consent is required, therefore based on the matching rule above, a SaaS Service provider provide less (or equal) consent than data subject and that SaaS service provider should define more (or equal) obligations than data subjects preferences as provided in the NDPR in order to obtain consent.

Hence, we define a Service provider's data handling policy as a set of consents and a set of obligations. Within the context of this research, this will be applied to entire stages of the data life cycle.

*B. **Collection***

At the Collection Phase, both service provider policy (POL) and customer requirements (REQ) are matched to ensure compliance and the syntax expressed as

1) Case 1.0

The syntax for *Pol_typeset*, and *Pcol = (cons=Y, cpurp)* has the following semantics: "whenever a piece of data of the type in the set of types typeset is collected then previously consent needs to be collected with the purposes in the set cpurp." Formally this can be defined by:

$$Whenever\ \ during\ the\ system\ operation/trace$$
$$\exists\ ActionCollectPersonalData(typeset,\ time1)\ \rightarrow$$
$$\exists\ ActionCollectConsent(typeset,\ time2)\ \text{and}\ time > time2$$

Here, we further define two new action constructs to capture the semantics of our new syntax.

C. Usage

1) Case 2.0:

The syntax of the policy $Puse = (cons, usepurp, whocanuse, decl)$, $where\ decl \in \{Y, N\}$, cons = {Y, N}, is as follows: Has the following semantics:

"Whenever a piece of data of the type in the set of types typeset is used then consent needs to be collected with the set of purposes in the set *usepurp* and who can use the data in the set *whocanuse* with the conditions declared by decl ∈ {Y, N}"

Formally this can be defined by:

2) Case 2.1

(cons = Y) besides any possible/defined value of $usepurp, whocanuse, decl$

∃ ActionDataUsage (*typeset*, *withpurp, bywhom*, time1) →

∃ ActionCollectUsageConsent(*typeset*, $Puse.usepurp, Puse.whocanuse, decl,$ time2) and (*time2 < time1)*

and (withpurp ⊆ $Puse.usepurp$*) and (bywhom* ⊆ $Puse.whocanuse$)

3) Case 2

(cons = N) besides any possible/defined value of $usepurp, whocanuse, decl$

∃ ActionUseData (*typeset*, *withpurp, bywhom*, time1) →

(withpurp ⊆ $Puse.usepurp$*) and (bywhom* ⊆ $Puse.whocanuse$)

4) Case 2.3

(decl = Y), besides any possible/defined value of $cons,\ usepurp, whocanuse$

∃FinishRegisteringToService (datatypesforusedregister, time1)→

∃ DeclaredUsage(*Puse.usepurp,Puse.whocanuse,time2)*, time2 < time1

D. Storage

1) Case 3.0

The syntax of the policy $Pstr = (wherestore, howstore, decl)$, $where\ decl \in \{Y, N\}$, wherestore is a set of places where data is stored "SPstorage", "3rdpartyStorage and *howstore* the method with which data is stored"

Has the following semantics:

"whenever a piece of data of the type in the set of types typeset is stored in a set of places *wherestore* and the method with which it is stored *howstore* with the conditions is declared by declaring decl ∈ {Y, N}" Formally this can be defined by:

2) Case 3.1

decl=Y, besides anyhow and when

Whenever during the system operation/trace

∃FinishRegisteringToService (typesforstorage, time1) →

∃DeclareStrorage ($Pstr.wherestore$, $Pstr.howstore$, *time2)*, time2 < time1

3) Case 3.2

wherestore = (*OwnersServers, 3rdPartyServers)* where *OwnServer* is service provider owned storage and *3rdPartyServers* is storage owned by a 3rdParty where personal data is stored by a service provider or on behalf of a service provider. *howstore* = (Available, Hidden) where *howstore* is where data encrypted with the service provider's key and content is available to the service provider (howstore = "available") or encrypted with the customer's key = content is hidden to the service provider (howstore = "hidden").

The semantics for the policy , where *wherestore, howstore* and *decl* can have any possible/defined value.

4) Case 3.2.1

For ($Pstr.wherestore = \{OwnServer\}$, *and* $Pstr.howstore = \{Available\}$)

Whenever during the system operation/trace

∃ ActionStore(typeset, time1) → ∃ ActionSaveEncryptedData(typeset, secretkey, servers, time2), time2 < time1, *secretkey* ∈ *Keys<serviceprovider>, servers* ⊆ *Servers*<serviceprovider>

5) Case 3.2.2

For ($Pstr.wherestore = \{OwnServer\}$, *and* $Pstr.howstore = \{Hidden\}$)

∃ ActionStore (typeset, time1) → ∃ ActionSaveEncryptedData(typeset, secretkey, servers, time2), time2 < time1, *secretkey* ∈ *Keys<customer>, servers* ⊆ *Servers*<serviceprovider>

6) Case 3.2.3

For ($Pstr.wherestore = \{3rdPartyServers\}$, *and* $Pstr.howstore = \{Available\}$)

∃ ActionStore (typeset, time1) → ∃ ActionSaveEncryptedData(typeset, secretkey, servers, time2), time2 < time1, *secretkey* ∈ *Keys<serviceprovider>, servers* ⊆ *Servers*<3rdparty>

7) Case 3.2.4.

For ($Pstr.wherestore = \{3rdPartyServers\}$, *and* $Pstr.howstore = \{Hidden\}$)

∃ ActionStore(typeset, time1) → ∃ ActionSaveEncryptedData(typeset, secretkey, servers, time2), time2 < time1, *secretkey* ∈ *Keys<customer>, servers* ⊆ *Servers*<3rdparty>

E. Deletion

1) Case 4.0

The syntax for Pol_typeset, and Pdel =(*placedel*, *when, decl*), where *decl* ∈ {*Y, N*}, *placedel* is a set of elements "mainstorage", "backupstorage", "3rdparty") and when = (rdelay, gdelay). has the following semantics: "whenever a piece of data of the type in the set of types typeset is deleted then personal data deletion mode needs to be stated as well as the delay period and whether data is fully deleted as with the conditions should be declared by declaring by decl ∈ {Y, N}".

Formally this can be defined by:

Formally this can be defined by:

2) Case 4.1

decl=Y, besides any how and when
Whenever during the system operation/trace

FinishRegisteringToService (typesforregister, time1)→

DeclaredDel(Pdel.*placedel,* Pdel. *When, time2)*, time2 < time1

3) Case 4.2

when = (rdelay=dd, gdelay) where dd is a *numerical* retention delay value, namely, the semantics for the policy

Pdel = (placedel, (rdelay=rd,gdelay),decl), where *placedel, gdelay* and *decl decl* can have any possible/defined value.
Whenever during the system operation/trace

∃ ActionInitDeletionData(typeset, time1) →

∃ ActionDeleteData(typeset, fromwhere, time2) and (time1+rd ≥ time2) and
(fromwhere ∈ Pdel. *placedel*)

∃ ActionDeleteData(typeset, fromwhere*, time1)* →

∃ ActionInitDeletionData(typeset, time2) and (time1+rd ≥ time2) and (fromwhere ∈ Pdel. *placedel*)

4) Case 4.3

when = (rdelay, gdelay=gd) and (*"3rdparty" ∉ Pdel.placedel)* where gd is a numerical worst-case delay value, namely, the semantics for the policy *Pdel = (placedel, (rdelay, gdelay = gd), decl),* where *placedel, rdelay* and *decl* can have any possible/defined value (in the language alphabet).
Whenever during the system operation/trace

∃ ActionFinishedUnRegisteringFromService (alltypesintheservice, time1) →

∃ ActionDeleteData(alltypesintheservice, fromwhere, time2) and *(time1+gd ≥ time2) and*
(fromwhere ∈ Pdel. *placedel*)

∃ ActionDeleteData(alltypesintheservice, time2) →

∃
ActionFinishedUnRegisteringFromService(alltypesintheservice,
fromwhere, time1)
and *(time1+gd ≥ time2) and* (fromwhere ∈ Pdel. *placedel*)

5) Case 4.4

when = (rdelay="DF", gdelay) where "DF" (i.e., defined) is a *non-numerical* retention delay value such as "Until required for a national security law at a country". Namely, the semantics for the policy *Pdel = (placedel, (*rdelay="DF"*, gdelay), decl),* where *placedel, gdelay* and *decl* can have any possible/defined value.

Whenever during the system operation/trace

∃ ActionInitDeletionData(typeset, time1) →

∃ ActionDeleteData(typeset, fromwhere, time2) and (time1+RDelay(law) ≥ time2) and (fromwhere ∈ Pdel.*placedel*), where RDelay(law) returns a retention delay based on the given law *law*

∃ ActionDeleteData(typeset, fromwhere, time1) →

∃ ActionInitDeletionData(typeset, time2) and (time1+RDelay(law) ≥ time2) and (fromwhere ∈ Pdel. placedel), where RDelay(law) returns a retention delay based on the given law law

6) Case 4.5

when = (rdelay, gdelay = "DF") where "DF" (i.e., defined) is a ***non-numerical*** worst-case delay value such as "Until required for a national security law at a country". Namely, the semantics for the policy *Pdel = (placedel, (*rdelay, gdelay = "DF"*), decl),* where *placedel, gdelay* and *decl* can have any possible/defined value.

Whenever during the system operation/trace

∃ ActionInitDeletionData(typeset, time1) →

∃ ActionDeleteData(typeset, fromwhere, time2) and (time1+GDelay(law) ≥ time2) and (fromwhere ∈ Pdel.*placedel*), where RDelay(law) returns a worst-case delay based on the given law *law*

∃ ActionDeleteData(typeset, from where*, time1)* →

∃ ActionInitDeletionData(typeset, time2) and (time1+ GDelay(law) ≥ time2) and (fromwhere ∈ Pdel.*placedel*), where RDelay(law) returns a worst-case delay based on the given law *law*

F. FORWARDING

1) Case 5.0

The semantics of the policy Pfw $(cons, fwpurp, 3rdparty, decl)$, *where* $decl \in \{Y, N\}$, cons = {Y, N}, is as follows:

2) *Case 5.1*

(cons = Y) besides any possible/defined value of $fwpurp, 3rdparty, decl.$

∃ ActionForwardData (*typeset*, *withpurp, towhom*, time1) →

∃ ActionCollectFwConsent(*typeset*, $fwpurp, 3rdparty,$ time2) and (*time2 < time1) and (withpurp* ⊆ $Pfw.fwpurp$*) and (towhom* ⊆ $Pfw.3rdparty$)

3) *Case 5.2*

(cons = N) besides any possible/defined value of $fwpurp, 3rdparty, decl.$

∃ ActionForwardData (*typeset*, *withpurp, towhom*, time1) →

(withpurp ⊆ $Pfw.fwpurp$*) and (towhom* ⊆ $Pfw.3rdparty$)

4) *Case 5.3*

(decl = Y), besides any possible/defined value of cons, $fwpurp, 3rdparty.$

FinishRegisteringToService(datatypesforusedregister, time1)→

DeclaredFw($Pfw.fwpurp$, $Pfw.3rdparty$*,time2)*, time2 < time1

"whenever a piece of data of the type in the set of types typeset is forwarded to a third party then consent needs to be collected with the set of purposes in the set $fwpurp$ "

Compliancy Check Based on Our Syntax

Property 1:

(Pol ≤ REQ) iff (for every *P_typeset_i* in *Pol* there is a *R_typeset_i' in REQ such that P_typeset_i* ≤ *R_typeset_i'*)

Property 2:

P_typeset_i ≤ *R_typeset_i' iff typeset_i'* ⊆ *typeset_i, and*

$Pcol \leq Rcol$, $Puse \leq Ruse$, ..., $Pdel \leq Rdel$

Property 3:

$Pcol \leq Rcol$ iff ($Pcol.cons = N$ & $Rcol.cons$ = N) or $Pcol.cons = Y$, *and* $Rcol.cons$ can be N or Y, and ($Pcol.cpurp \subseteq Rcol.cpurp$)

Property 4:

$Puse \leq Ruse$ iff ($Puse.cons = N$ & $Ruse.cons = N$) or $Puse$.cons = Y *and* $Ruse.cons$ can be N or Y, and ($Puse.upurp \subseteq Ruse.upurp$) ($Puse.whouse \subseteq Ruse.whouse$)

Property 5:

$Pstr \leq Rstr$ iff ($Pstr.wh = N$ & $Rstr.wh = N$) or $Pstr.wh = Y$ *and* $Rstr.wh$ can be N or Y, and ($Pstr.ho \subseteq Rstr.ho$) ($Pstr.revdate \subseteq Rstr.revdate$)

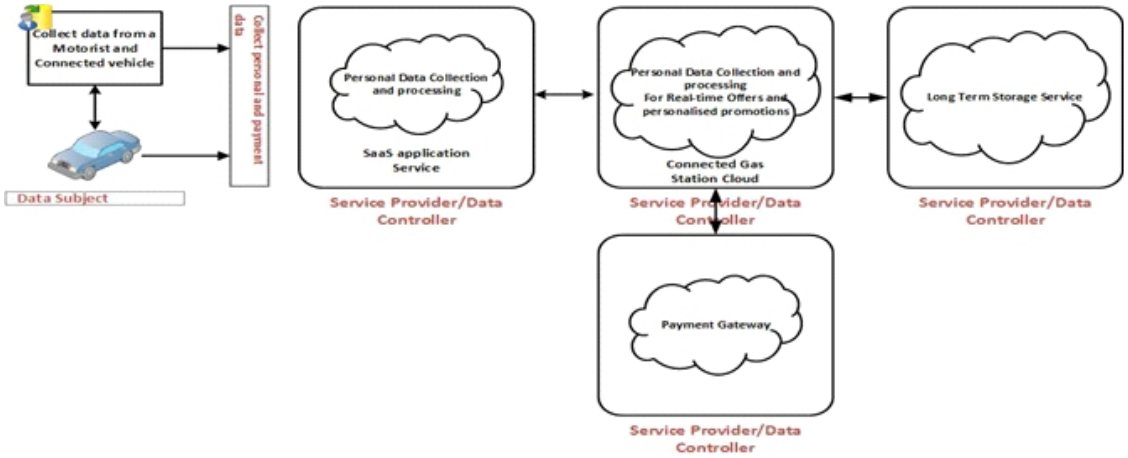
Property 6:

$P_{del} \leq R_{del}$ iff ($P_{del}.how = N$ & $R_{del}.how = N$) or P_{del}.how = Y *and* $R_{del}.how$ can be N or Y, and ($P_{del}.deld \subseteq R_{del}.deld$) ($P_{del}.geld \subseteq R_{del}.geld$)

Property 7:

$P_{fw} \leq R_{fw}$ iff ($P_{fw}.cons = N$ & $R_{fw}.cons = N$) or $P_{fw}.cons = Y$ *and* $R_{fw}.cons$ can be N or Y, and ($P_{fw}.fwpurp \subseteq R_{fw}.fwpurp$) ($P_{fw}.3rdparty \subseteq R_{fw}.3rdparty$)

Use Case of SP-PPL: An Example



This use case shows the flow of data within an oil and gas retail services scenario. It shows how data is handled across the stages of a data life cycle within the context of SaaS applications and connected petrol stations via the cloud. The SaaS application is used to support the interaction of motorists with a smart petrol station by collecting and processing personal data from data subjects via connected vehicles using internet connectivity. In the following, we explain the data life cycle for this use case.

1. **Data Collection:** In order to serve the customer at the smart petrol station, the customer's personal and payment data is collected by the station using a SaaS retail application to fulfil the service request such as refuelling and/or car servicing.
2. **Usage:** At this stage, the service provider of the retail SaaS application processes the personal data to serve the data subject (i.e. the customer).
3. **Storage:** The personal data at this stage is stored to serve the user based on data subject's data handling preferences in the future. The SaaS service provider also can at this stage, depending on the purpose, store the data on third-party storage infrastructure.
4. **Deletion:** Relying on consent and lawful purposes for retention, the SaaS service provider and their third-party partners destroy or delete data to sustain compliance. However, this comes with several challenges, such as technical and complexities primarily due to multi-cloud environments of different providers requiring different data deletion processes towards different types of requests.

5. **Data forwarding:** Data forwarding and sharing with the SaaS application service provider and payment processing company occurs at this stage. The SaaS provider is responsible for the services and interaction platform between the petrol station and the customer. The SaaS service provider, as the data controller, is responsible for the personal data that will be collected to enable interaction and service.

In order enforce compliance to NDPR, we rely on the abstract extensions of the SP-PPL to check for a match or a mismatch in the data subject obligations/preferences and data controller obligations/preferences using a binding or matching policy that takes input from both sides.

Conclusion And Further Work

In this paper, we highlighted the key benefits and challenges that are faced when using SaaS applications in distributed environments when it comes to compliance to data protection regulations. We analysed the SaaS deployment scenario and the suitability of a related policy language to address the requirements of the scenario. We presented SP-PPL, a proposed security and privacy policy language, as an extension of the PPL designed for the contexts of SaaS applications based on the requirements of the NDPR. We validated and showcased the applicability of SP-PPL within the context of SaaS applications using a smart petrol station use case. We have shown that there still opportunities for improving effective compliance to security and privacy protection in SaaS applications using of policy languages to enforce compliance a granular level.

Our experiments have shown that the proposed SP-PPL can be used to define, express and enforced data protection properties across different scenarios. We have also mapped the data lifecycle to the NDPR to show applicability to recent regulations. Our proposed policy language is highly expressive and extensible. This paper also analysed the state-of-the-art of policy languages and policy schemes. Our proposed policy language proves that the existing policy languages cannot be used to enforce security and privacy compliance in new data protection regulations such as the NDPR in the context of SaaS applications. Further comparisons against existing policy languages was also carried out to highlight their weaknesses in fulfilling the requirements of enforcing compliance in SaaS applications. Future work could be done on the implementation of a cloud-hosted tool focused on the oil and gas retail industry and adopting the policy language approach to enforcing compliance at granular levels.

References

[1] Trapero, R. et al, (2017). *A novel approach to manage cloud security SLA incidents, Future Generation Computer Systems, 72, 193-205,. DOI: https://doi.org/10.1016/j.future.2016.06.004.*

[2] Tiwari, P. K. & Joshi, S., (2014). A review of data security and privacy issues over SaaS, *IEEE International Conference on Computational Intelligence and Computing Research, 1-6.*

[3] National Institute of Standard and Technology -USA, (2011). *The NIST Definition of cloud computing,*

[4] Wang, Y. et al, (2018). *A role-based access control system using attribute-based encryption, International Conference on Big Data and Artificial Intelligence (BDAI), 128-133.*

[5] Skendžić A., Kovačić B. & Tijan, E., (2018). *General data protection regulation — protection of personal data in an organisation, 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), 1370-1375.*

[6] Vinh, T. T. (2018). *Privacy by design: On the formal design and conformance Check of Personal Data Protection Policies and Architectures," May 15,.*

[7] Tamburri, D. A. (2019). Design principles for the general data protection regulation (GDPR): A formal concept analysis and its evaluation*, Information Systems, 101469 DOI: https://doi.org/10.1016/j.is.2019.101469.*

[8] (April, 25 2019). *The Nigerian Data Protection Regulation, 2019.* Available: https://nitda.gov.ng/wp-content/.../Nigeria%20Data%20Protection%20Regulation.pdf.

*[*9] Yimam, D. & Fernandez, E. B., (2016). A survey of compliance issues in cloud computing*, Journal of Internet Services and Applications, 7, (1), 5.*

[10] Giulio, C. D. et al, (2017). Cloud standards in comparison: Are new security frameworks improving cloud security?*, IEEE 10th International Conference on Cloud Computing (CLOUD), 50-57.*

[11] (September 2019). *What is Personal Data [personal data]. Available:* https://ico.org.uk/for-organisations/guide-to-data-protection/guide-to-the-general-data-protection-regulation-gdpr/key-definitions/what-is-personal-data/.

[12] (). About US.

[13] (). NITDA, Background.

[14] Altorbaq, A., Blix F. & Sörman, S., (2017). Data subject rights in the cloud: A grounded study on data protection assurance in the light of GDP*R, 12th International Conference for Internet Technology and Secured Transactions (ICITST), 305-310.*

[15] (). *Intention to fine Marriott Internation Inc more than £99 million under GDPR for data breach.*

[16] (2019). *Intention to fine British Airways £183.39m under GDPR for data breach,* Available: https://ico.org.uk/about-the-ico/news-and-events/news-and-blogs/2019/07/ico-announces-intention-to-fine-british-airways/.

[17] (). *List of Licensed Data Protection Compliance organisations (DPCO).*

[18] (09/24/2019). *Nigeria's ICT agency is going after Truecaller over 'breaches of privacy,' but Sweden-based firm says user's privacy is of utmost importance.* [Nigeria ICT agency going after *ruecaller]. Available: https://www.pulse.ng/bi/tech/nigerias-ict-agency-is-going-after-truecaller-over-breaches-of-privacy-but-the-sweden/1g9ht53.*

[19] Ardagna, C. A. et al, (2010). Enabling privacy-preserving credential-based access control with XACML and SAML *10th IEEE International Conference on Computer and Information Technology, 1090-1095.*

[20] Slim, T. Akram, N., Laurent, B., & Gregory, N. (2010). *PPL engine: A Symmetric architecture for privacy policy handlin gHttps://Www.W3.Org/2010/Policy-Ws/Papers/11-Trabelsi-Primelife-PPL_Engine.Pdf, 3*

[21] Azraoui, M. et al, (2015). A-PPL An accountability policy language," in Data Privacy Management, *Autonomous Spontaneous Security, and Security Assurance, 319-326.*

[22] Henze, M. et al, (2016). *CPPL: Compact privacy policy language,"* in Proceedings of the 2016 ACM on Workshop on Privacy in the Electronic Society*, Vienna, Austria, 99-110.*

[23] Benghabrit, W. et al, (2014). *A cloud accountability policy representation framework,"* in Proceedings of the 4th International Conference on Cloud Computing and Services Scienc*e,* Barcelona, Spain*, 489-498.*

[24] Sion, L. et al, (2019). *An architectural view for data protection by design,* in 2019 IEEE International Conference on Software Architecture (ICSA), 11-20.

[25] Wayne, T. & Jansen, G. (2011). Guidelines on security and privacy in public cloud computing," US National Institute for Standards and Technology*, Gaithersburg, Maryland, Tech. Rep. Special Publication 800-144, 2011..*

[26] Elluri, L., Nagar, A. & Joshi, K. P. (2018). *An integrated knowledge graph to automate GDPR and PCI DSS compliance,* IEEE International Conference on Big Data (Big Data) 1266*-1271.*

[27] Jayasinghe, U., Lee, G. M. & MacDermott, A., (2018). *Trust-based data controller for personal information management,* International Conference on Innovations in Information Technology (IIT)*, 123-128.*

[28] Altorbaq, A., Blix, F. & Sörman, S., (2017). *Data subject rights in the cloud: A grounded study on data protection assurance in the light of GDPR, 12th I*nternational Conference for Internet Technology and Secured Transactions (ICITST*), 305-310.*

[29] Papadimitriou, S., Mougiakou, E. & Virvou, M., (2019). *Smart educational games and consent under the scope of general data protection regulation, 10th* International Conference on Information, Intelligence, Systems and Applications (IISA*), 1-8.*

[30] Sarkar, S. et al, (2018). *Towards enforcement of the EU GDPR: Enabling data erasure,* IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), 222-229.

[31] Butin, D. & Métayer, D. L (2015). *A guide to end-to-end privacy accountability,* IEEE/ACM 1st International Workshop on TEchnical and LEgal Aspects of Data pRivacy and SEcurit*y,20-25.*

[32] (). G*DPR Fines and Penalties – Consequences of Non-Compliance.*

[33] Yu, T. & Winslett, M. (2003). *Policy migration for sensitive credentials in trust negotiation, i*n Proceedings of the 2003 ACM Workshop on Privacy in the Electronic Society, 9-20.

[34] Al-Zaben, N. et al, (2018). General data protection regulation complied blockchain architecture for personally identifiable information managemen*t, International Conference on Computing, Electronics Communications Engineering (iCCECE), 77-82.*

[35] Elluri, L., Nagar, A. & Joshi, K. P. (2018). An integrated knowledge graph to automate GDPR and PCI DSS complianc*e, IEEE International Conference on Big Data (Big Data), 1266-1271.*

[36] Indhumathil, T. et al, (2017). Third-party auditing for cloud service providers in multicloud environment, *Third International Conference on Science Technology Engineering & Management (ICONSTEM), 347-352.*

[37] Lins, S., Schneider, S. & Sunyaev, A. (2018). Trust is Good, Control is Better: Creating Secure Clouds by Continuous Auditing*, IEEE Transactions on Cloud Computing, 6, (3), 890-903,.*

[38] (2017). *eXtensible Access Control Markup Language (XACML) Version 3.0 2017,* Available: See: http://docs.oasis-open.org/xacml/3.0/errata01/os/xacml-3.0-core-spec-errata01-os.html.

[39] Krupp, M. M. et al, (2017). A focus group study of privacy concerns about telepresence robots, 26th IEEE *International Symposium on Robot and Human Interactive Communication (RO-MAN), 1451-1458.*

[40] Trabelsi, S., Sendor, J. & Reinicke, S., (2011). *PP*L: PrimeLife privacy policy engin*e, IEEE International Symposium on Policies for Distributed Systems and Networks, 184-185.*

[41] Ed-Daibouni, M. et al, (2016). *A formal specification approach of privacy-aware attribute based access control (pa-ABAC) model for cloud computing,* 2016 Third International Conference on Systems of Collaboration (SysCo), 1-5.

[42] Poroor, J. & Jayaraman, B., (2012). *C*2L:A formal policy language for secure cloud Configurations*, Procedia Computer Science, 10, 499-506, DOI: https://doi.org/10.1016/j.procs.2012.06.064.*

[43] Azraoui, M. et al, (2015). A-PPL: An accountability policy language, *Data Privacy Management, Autonomous Spontaneous Security, and Security Assurance, , 319-326.*

[44] Butin, D., Chicote, M. & Métayer, D. (2013). *Le*, "Log design for accountabili*ty, IEEE Security and Privacy Workshops, 1-7.*

[45] Jaatun, M. G. et al, (2016). Enhancing accountability in the cloud, *International Journal of Information Management, DOI: https://doi.org/10.1016/j.ijinfomgt.2016.03.004.*

[46] Industry, P. C., (2016). *Data security standard: Requirements and Security assessment Procedures, Version 3.2 PCI Security Standards Council,*

[47] Ghorbel, A., Ghorbel, M. & Jmaiel, M., (2017). *Priarmor: An IaaS solution for low-level privacy enforcement in the cloud, IEEE* 26th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE), 119-124.

[48], Kasem-Madani, S. & Michael, M. (2015). *Security and privacy policy languages: A survey, Categorization and Gap Identification.*

[49] Azraoui, M. et al (2015). A-PPL: An accountability policy language," in Data Privacy Managemen*t, Autonomous Spontaneous Security, and Security Assurance, 319-326.*

[50] Jaatun, M. G. et al (2017). *Accountability requirements for the cloud,* IEEE International Conference on Cloud Computing Technology and Science (CloudCom)*, 375-382.*

[51] Liu, Y., Xu K. & Song, J., (2013). A task-attribute-based workflow access control model, IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing, 1330-1334.