symmetry

MDPI

Article

SLAVUL: A Novel Ontology-Driven Approach for Integrating Cloud SLA Security Knowledge and Vulnerability Intelligence

Ozgu Can [1,*] and Sena Yakut [2]

[1] Department of Computer Engineering, Faculty of Computer and Informatics Sciences, Ege University, 35100 Bornova-Izmir, Türkiye
[2] Department of Computer Engineering, Graduate School of Natural and Applied Sciences, Ege University, 35100 Bornova-Izmir, Türkiye; senayktt@gmail.com
* Correspondence: ozgu.can@ege.edu.tr

Abstract

Cloud Service Level Agreements (SLAs) define security obligations, remediation commitments, and compliance requirements between cloud service providers and customers. SLAs define the performance standards and expectations between service providers and users. Therefore, it is an essential element in cloud computing. However, SLA documents are typically represented in unstructured natural language and lack semantic integration with operational vulnerability management processes. Meanwhile, cloud security platforms continuously generate vulnerability intelligence containing security findings, severity levels, and remediation information. The lack of semantic interoperability between SLA-defined security obligations and vulnerability intelligence limits automated security governance, compliance assessment, and vulnerability management in cloud environments. To address these challenges, this study proposes an integrated semantic approach that enables the representation, integration, and reasoning of SLA-derived security knowledge and cloud vulnerability intelligence within a unified ontology model. The proposed solution combines automated knowledge acquisition from SLA documents, the development of an SLA Security Ontology and a Vulnerability Ontology, ontology alignment techniques, and Semantic Web Rule Language (SWRL)-based reasoning mechanisms to support automated vulnerability management, remediation commitment assignment, and consistency verification. Further, the proposed approach is evaluated using real-world SLA documents and vulnerability information. Experimental results demonstrate that the developed reasoning mechanism successfully identifies logical inconsistencies in 73% of the evaluated SLA cases. The analysis further reveals that the undetected cases correspond to contractually incorrect yet logically consistent outputs, highlighting the complementary role of human validation in ontology-driven security management. As a result, the study demonstrates that ontology engineering and rule-based semantic reasoning can effectively bridge the gap between contractual security obligations and operational vulnerability intelligence. Therefore, the proposed approach provides a foundation for automated vulnerability management, SLA compliance assessment, and semantically aware cloud security governance.

Keywords: cloud security; knowledge representation; ontology alignment; service level agreement (SLA); semantic interoperability; SWRL reasoning; vulnerability management

Check for updates

Academic Editor: Calogero Vetro

Received: 26 June 2026
Revised: 2 August 2026
Accepted: 7 August 2026
Published: 8 August 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

Cloud computing has become a fundamental paradigm of modern digital infrastructures, enabling organizations to deploy scalable, flexible, and cost-effective services. As

cloud adoption continues to grow, security management has emerged as a critical concern due to the increasing number of vulnerabilities, cyber threats, and compliance requirements affecting cloud environments. Service Level Agreements (SLAs) play a central role in defining contractual security obligations between Cloud Service Providers (CSPs) and Cloud Service Users (CSUs) by detailing Service Level Objectives (SLOs) and specifying security controls, remediation commitments, monitoring activities, and compliance requirements. Therefore, an SLA serves as a contractual mechanism that holds service providers accountable through predefined financial penalties or compensation arrangements when agreed service objectives are not satisfied [1]. Moreover, SLAs are accepted as an effective method to manage and guarantee the promised Quality of Service (QoS) [2]. However, SLA documents are typically represented in unstructured natural language, making their automated interpretation, integration, and enforcement challenging.

In addition to SLA-based security specifications, cloud security platforms continuously generate large volumes of vulnerability information describing security findings, severity levels, remediation recommendations, and risk assessments. Although both SLA documents and vulnerability intelligence contain security-related knowledge, they are generally maintained in isolated formats and managed through independent processes. As a result, organizations often face difficulties in determining whether detected vulnerabilities comply with contractual remediation requirements defined within SLAs.

Since machine-readable SLAs provide an effective mechanism for addressing security, privacy, and compliance requirements in cloud environments [3], representing SLA knowledge through machine-readable ontologies facilitates the formal specification, interpretation, and management of security obligations, privacy requirements, and compliance commitments defined within cloud service agreements. Hence, the increasing sophistication and diversity of cyber threats have exposed significant challenges for conventional security monitoring and analysis approaches. This evolution has increased the need for ontology-based models and semantic reasoning-based mechanisms that enable intelligent security management, automated decision-making, and proactive threat response [4]. Therefore, ontology-based knowledge representation has emerged as a promising solution for addressing semantic heterogeneity and improving interoperability across distributed information sources. Ontologies provide formal and machine-readable representations of domain knowledge, enabling automated reasoning, consistency verification, and semantic integration. Several studies have explored ontology-driven approaches for cloud computing, SLA management, cybersecurity knowledge representation, and vulnerability analysis. Nevertheless, existing approaches typically focus on either SLA modeling or vulnerability representation independently, leaving the semantic relationship between contractual security obligations and operational vulnerability information largely unexplored. To address this gap, this study proposes an integrated ontology framework that establishes semantic interoperability between SLA security obligations and operational vulnerability information. Furthermore, most existing studies lack automated mechanisms capable of aligning vulnerability findings with contractual remediation commitments and identifying logical inconsistencies in SLA-derived security knowledge. The absence of such mechanisms restricts the practical applicability of ontology-based solutions for cloud security governance, compliance assessment, and vulnerability management.

To address these challenges, this study proposes an ontology-based approach that integrates SLA security knowledge and vulnerability intelligence within a semantically interoperable knowledge model. The proposed approach integrates an SLA Security Ontology, a Vulnerability Ontology, ontology alignment mechanisms, and Semantic Web Rule Language (SWRL)-based reasoning capabilities into a unified ontology model referred to as the SLAVUL Ontology. Through semantic integration and automated inference, the

proposed approach enables machine-readable security semantics, vulnerability–SLA alignment, remediation obligation propagation, and logical inconsistency detection. Accordingly, this study addresses the following Research Questions (RQ):

- RQ-1: How can security-related SLA knowledge and cloud vulnerability intelligence be represented within interoperable ontology models?
- RQ-2: How can semantic interoperability be established between SLA obligations and vulnerability information originating from heterogeneous security sources?
- RQ3: To what extent can SWRL-based reasoning support automated vulnerability–SLA alignment and remediation obligation propagation?
- RQ4: Can ontology-driven consistency verification mechanisms effectively identify logical inconsistencies in SLA-derived security knowledge?

1.1. Motivation

The growing complexity of cloud ecosystems requires security knowledge to be represented in a machine-readable and semantically interoperable manner. Although SLA documents define contractual security responsibilities, remediation commitments, and compliance requirements, these specifications are often disconnected from operational vulnerability management processes. Consequently, organizations must manually interpret SLA clauses and determine whether identified vulnerabilities satisfy contractual obligations. This process is time-consuming, error-prone, and difficult to scale in large cloud environments. Concurrently, modern vulnerability assessment platforms continuously generate security findings that contain valuable information regarding vulnerability severity levels, remediation recommendations, and risk exposure. The lack of semantic integration between these operational findings and SLA-defined security commitments limits automated compliance assessment and security governance capabilities. Therefore, a unified ontology-driven framework capable of integrating contractual security knowledge with vulnerability intelligence is required to support intelligent cloud security management.

1.2. Contributions

The main objective of this study is to establish a semantically interoperable knowledge representation that bridges contractual security obligations defined in cloud SLAs with operational vulnerability intelligence obtained from cloud security assessment processes. By integrating these two complementary sources of security knowledge, the proposed approach aims to support automated vulnerability management, remediation obligation alignment, and consistency verification within cloud environments.

Previous studies have primarily focused on either SLA modeling and cloud security ontologies, and lack of support for semantically linking vulnerability information with SLA-defined security obligations, remediation commitments, and compliance requirements. In addition, the automated detection of logical inconsistencies in SLA-derived security knowledge and the alignment of contractual remediation obligations with identified vulnerabilities remain insufficiently supported by existing approaches. Therefore, the semantic integration of contractual security knowledge and operational vulnerability information is insufficiently addressed. Furthermore, limited research has investigated integrated solutions that integrate automated SLA knowledge extraction with ontology-driven reasoning to support security governance, compliance assessment, and vulnerability management in cloud environments.

Motivated by the identified limitations and the research gap, the main contributions of this study are summarized as follows:

- Cloud Security-Oriented SLA Ontology: A machine-readable SLA Security Ontology is developed to formally represent security-related knowledge, vulnerability manage-

ment requirements, remediation commitments, security activities, and compliance requirements extracted from cloud SLA documents. Unlike existing SLA ontologies, the proposed ontology explicitly integrates concepts related to vulnerability assessment and remediation processes, enabling the semantic representation of cloud security obligations.

- Vulnerability Ontology for Cloud Services: A dedicated Vulnerability Ontology is automatically generated using structured vulnerability information obtained from Amazon Inspector [5]. The ontology provides a formal representation of vulnerability findings, severity levels, exploitability status, remediation status, and associated security attributes. Therefore, a reusable semantic model is established for cloud vulnerability intelligence.
- Ontology Alignment and Semantic Integration: A novel integrated ontology model, referred to as the SLAVUL Ontology, is developed by semantically aligning the SLA Security Ontology and the Vulnerability Ontology. This integration establishes interoperability between contractual security requirements and operational vulnerability information, enabling unified security knowledge representation.
- SWRL-Based Automated Reasoning and Vulnerability–SLA Alignment: A set of SWRL rules is introduced to support automated inference within the SLAVUL Ontology. The proposed reasoning mechanism aligns vulnerability findings with SLA-defined vulnerability concepts, maps severity classifications, and associates contractual remediation obligations to the corresponding vulnerabilities.
- Logical Inconsistency Detection and Consistency Verification: A rule-based verification mechanism is developed to identify semantic and logical inconsistencies within SLA-derived security knowledge. The proposed approach automatically detects violations related to remediation commitments, severity-level relationships, and ontology consistency constraints. As a result, the reliability of ontology-generated knowledge is improved.
- Experimental Evaluation: The effectiveness of the proposed approach is evaluated using real-world SLA documents. The experimental results demonstrate the ability of the proposed approach to detect logical inconsistencies in ontology-derived knowledge, while also revealing scenarios in which contractually incorrect yet logically consistent outputs remain undetected. These findings provide important insights into the complementary roles of ontology reasoning and human validation in cloud security management.

1.3. Organization

The remainder of this paper is organized as follows. Section 2 reviews the related literature on ontology-based SLA management and security-oriented SLA management approaches. Section 3 presents the proposed SLAVUL framework, including the SLA Security Ontology, Vulnerability Ontology, ontology alignment process, and SWRL-based reasoning mechanisms. Section 4 discusses the experimental results and evaluates the effectiveness of the proposed consistency verification framework. Finally, Section 5 concludes the paper and outlines potential directions for future research.

2. Related Works

SLAs constitute a fundamental component of cloud service governance by formally specifying performance objectives, service guarantees, and contractual responsibilities between cloud providers and customers. In addition, SLAs enhance transparency and accountability by documenting commitments related to availability, responsiveness, security, and compliance. A comprehensive classification and review of SLA techniques developed

for cloud computing environments is provided in [1]. In addition to the general SLA management approaches [6–10], several studies [3,11–20] have explored security-based and privacy-oriented SLA approaches to address security requirements. The proposed methodologies enable the assessment of security and privacy guarantees, identification of missing controls, and analysis of security weaknesses. However, these studies focus on SLA assessment and security control management, and do not provide ontology-based knowledge representation, vulnerability intelligence integration, or automated semantic reasoning capabilities.

Ontology-based knowledge representation has received increasing attention in cloud computing due to its ability to formalize service descriptions, compliance requirements, security policies, and contractual obligations. In particular, ontology engineering has been widely adopted to improve machine interpretability, semantic interoperability, and automated reasoning capabilities within cloud environments. In addition, studies in the literature have highlighted ontology-based reasoning and Semantic Web (SW) technologies as key enablers for intelligent security intrusion detection and response, particularly in addressing the limitations of traditional pattern-based and behavior-based security approaches [4,21]. Despite these advances, research focusing specifically on security-oriented SLA ontologies remains relatively limited. The majority of ontology-based studies in the literature concentrate on modeling cloud security knowledge, threat information, and compliance-related requirements through semantic technologies [22–28]. Existing studies primarily concentrate on representing cloud service agreements and security requirements, while the integration of SLA knowledge with operational vulnerability intelligence and automated vulnerability management remains insufficiently explored.

One of the earliest studies addressing security-oriented SLA management in cloud environments is presented in [29]. The study identified the security and privacy concerns as major barriers to cloud service adoption and addressed these challenges through the development of the Security Service Level Agreement (SSLA) ontology. The proposed ontology formalizes cloud security requirements and compliance obligations. Although the SSLA ontology provides the semantic representation of cloud security agreements and compliance requirements, its primary focus is limited to modeling security obligations, negotiating security guarantees, and assessing compliance with regulatory frameworks such as HIPAA.

Subsequently, the authors extended their work by proposing an ontology-based Optimus framework [30]. Optimus leverages the SSLA ontology and Ontology Knowledge Bases (OKBs) to support security assessment and risk analysis for both cloud infrastructure providers and cloud service consumers. The framework enables infrastructure providers to formally describe security guarantees and service protection levels, while assisting consumers in evaluating security risks and identifying appropriate security assurances. The study further highlights the role of ontology-driven security management in supporting informed decision-making during cloud service selection and operation. However, both SSLA and Optimus primarily focus on SLA interpretation, compliance assessment, and security assurance management. As a result, the integration of operational vulnerability intelligence derived from cloud security assessment platforms with SLA-defined security obligations and remediation commitments remains insufficiently addressed.

In addition, a Semantic Web-based approach is proposed in [31] to automate the management and monitoring of cloud SLAs. The study introduces a cloud SLA ontology and a prototype system capable of automatically extracting SLA metrics and service-level information from legal SLA documents provided by cloud vendors. However, the study's scope is primarily limited to SLA modeling and monitoring processes, and it does not address vulnerability intelligence integration, ontology alignment, or rule-based reasoning

mechanisms for vulnerability management and consistency verification. Similarly, another Semantic Web-based framework [32] is proposed to transform unstructured SLA documents into semantically enriched knowledge graphs for supporting automated SLA extraction and comparison across cloud providers. Nevertheless, the study focuses on SLA representation and provider comparison, without considering vulnerability management and ontology-driven security reasoning.

Further, an ontology-based semantic SLA model (SCSLA) presented in [33] aims to improve the monitoring and management of smart contracts in blockchain environments. The proposed study facilitates semantic SLA representation and smart contract management. Its scope is primarily limited to blockchain environments and it does not address automated SLA knowledge extraction, ontology-driven reasoning, vulnerability intelligence management.

A comprehensive survey of SLA ontologies for cloud services and cloud security is presented in [34]. The study systematically reviews existing ontology-based approaches developed for cloud service management, security assurance, and compliance requirements. In addition, the study identifies the current research trends, challenges, and future directions. Furthermore, the survey emphasizes the importance of ontological inference and semantic reasoning mechanisms for enhancing automation, knowledge interoperability, and intelligent decision-making within cloud environments. Motivated by these observations, this study investigates the semantic integration of security knowledge derived from cloud SLA documents and vulnerability intelligence generated by Amazon Inspector within a unified ontology model. In addition, ontology alignment and SWRL-based reasoning mechanisms are employed to enable automated vulnerability management, consistency verification, and remediation commitment assignment. Table 1 summarizes and compares the most relevant studies in the literature with the proposed approach. The comparison criteria presented in Table 1 are selected based on the research gap identified in the literature and the primary contributions of the proposed study. Specifically, the comparison focuses on semantic knowledge representation (SLA Security Ontology and Vulnerability Ontology), automated ontology construction (Automated SLA Knowledge Extraction), semantic interoperability (Vulnerability Intelligence Integration and Ontology Alignment), and ontology-driven intelligence (Rule-Based Semantic Reasoning and Ontology Consistency Verification). These criteria collectively represent the essential capabilities required for an integrated ontology framework supporting cloud security and vulnerability management.

Table 1. Comparison of ontology-based approaches for SLA management, vulnerability integration, and semantic reasoning.

Reference #	SLA Security Ontology	Vulnerability Ontology	Automated SLA Knowledge Extraction	Vulnerability Intelligence Integration	Ontology Alignment	Rule-Based Semantic Reasoning	Ontology Consistency Verification
[29]	✓	✗	✗	✗	✗	✗	✗
[30]	✓	✗	✗	✗	✗	✗	✗
[31]	✓	✗	✓	✗	✗	✗	✗
[32]	✓	✗	✓	✗	✗	✗	✗
[33]	✓	✗	✗	✗	✗	✗	✗
This Study	✓	✓	✓	✓	✓	✓	✓

In contrast to the existing studies, the SLA Security Ontology proposed in this study is automatically constructed from SLA-derived question–answer pairs, providing a flexible and extensible representation of security activities, vulnerability management requirements,

severity classifications, and remediation commitments. Further, the SLAVUL Ontology integrates operational vulnerability intelligence with contractual security knowledge through ontology alignment and SWRL-based reasoning mechanisms. Moreover, ontology alignment and SWRL-based reasoning mechanisms are employed to associate vulnerabilities with SLA-defined severity levels and remediation obligations. Therefore, the proposed approach enables automated vulnerability management and consistency verification with real-world SLA documents.

3. SLA-Based Ontology-Driven Cloud Security

An effective cloud security management requires more than the isolated representation of service agreements, security controls, and vulnerability information. In today's cloud environments, security-related knowledge is distributed across heterogeneous sources, including SLA documents, vulnerability repositories, compliance standards, and remediation policies. The semantic heterogeneity among these sources often hinders interoperability, automated compliance verification, and intelligent security decision-making. Consequently, there is a growing need for machine-readable knowledge representations capable of establishing semantic relationships among cloud security entities while supporting automated reasoning and consistency verification.

To address these challenges, this study proposes an ontology-driven approach that integrates SLA security semantics with vulnerability knowledge representations within a unified semantic model. The proposed approach aims to transform traditionally human-readable SLA security clauses into machine-interpretable knowledge structures and establish explicit semantic links between security obligations, vulnerabilities, remediation actions, and compliance requirements. By leveraging ontology engineering principles, the framework enables semantic interoperability across heterogeneous security artifacts and provides a formal foundation for automated security analysis.

Furthermore, the proposed approach incorporates semantic reasoning mechanisms based on SWRL rules to infer implicit knowledge, identify remediation obligations, detect logical inconsistencies, and support security-aware decision-making processes. The proposed approach facilitates automated vulnerability-obligation mapping and enhances the overall intelligence of cloud security governance through the integration of ontology alignment techniques and rule-based inference.

Figure 1 presents the architecture of the proposed solution. The proposed solution comprises four semantic components supported by a knowledge acquisition layer. The knowledge acquisition layer gathers security-related information from heterogeneous sources, including SLA documents, security policies, compliance standards, and vulnerability repositories. Building upon this knowledge base, the first semantic component defines the SLA Security Ontology, while the second component formalizes vulnerability knowledge through a dedicated Vulnerability Ontology. The SLA Security Ontology formally represents security-related concepts, obligations, metrics, and compliance requirements contained within cloud service agreements. The Vulnerability Ontology captures vulnerabilities, threats, attack vectors, risk levels, assets, and mitigation actions. The third component establishes semantic integration and interoperability between the two ontologies through ontology alignment mechanisms, enabling the creation of machine-readable security semantics and shared conceptual representations. Finally, the fourth component employs SWRL-based reasoning and consistency verification mechanisms to infer implicit knowledge, align remediation obligations, and detect logical inconsistencies. Together, these components provide an integrated semantic framework for intelligent cloud security management. The following subsections present each component of the proposed approach in detail.

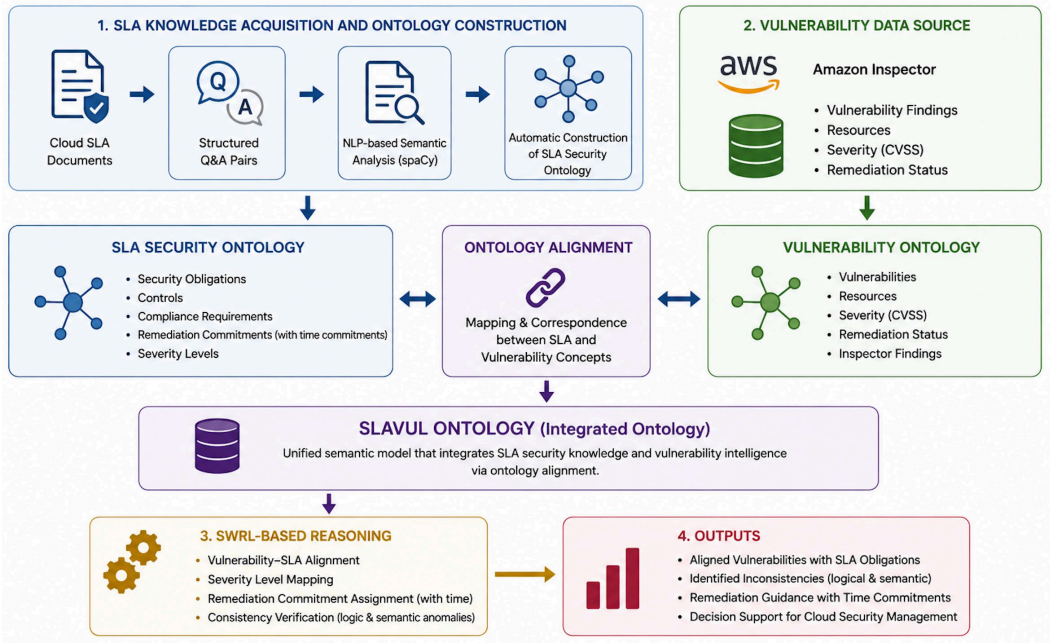


Figure 1. The architecture of the proposed approach.

3.1. Knowledge Acquisition Through SLA-Based Question–Answer Generation

In this study, the development of the SLA Security Ontology is grounded in a structured knowledge acquisition process that transforms unstructured SLA documents into machine-interpretable semantic knowledge. Rather than relying solely on manual ontology engineering, the proposed approach utilizes a Question and Answer (Q&A) generation mechanism to systematically extract security-related knowledge from contractual SLA specifications. This process serves as the initial stage of the overall framework and provides the semantic foundation for ontology development and population.

As illustrated in Figure 2, SLA documents collected from cloud service providers are first processed through a Natural Language Processing (NLP)-based knowledge acquisition pipeline. The pipeline performs several preprocessing operations, including sentence segmentation, tokenization, syntactic parsing, and Named Entity Recognition (NER), to identify security-relevant entities and contextual relationships within SLA clauses. Building upon the extracted linguistic information, domain-specific questions are generated to capture security obligations, compliance requirements, service guarantees, remediation commitments, and operational constraints defined in the agreements.

Specifically, SLA documents are processed using a domain-adapted NLP model that is fine-tuned to interpret security-oriented contractual statements. Based on predefined security and compliance-related queries, the model generates corresponding answers that reflect the implicit and explicit obligations defined within SLAs. These Q&A pairs capture key aspects such as vulnerability severity levels, remediation timeframes, and compliance requirements. The resulting Q&A pairs are used as an intermediate representation between natural language SLA clauses and formal ontology structures. Hence, the Q&A pairs provide a structured representation of both explicit and implicit knowledge contained within SLA documents. Unlike manual ontology engineering approaches, this mechanism enables a semi-automated knowledge extraction process, improving consistency in interpreting SLA statements across different service providers and contractual formats. Subsequently, the generated Q&A dataset is utilized as the foundational input for defining ontology classes, object properties, and data properties within the SLA ontology. This ensures that the ontology is directly grounded in semantically validated outputs derived from the NLP-

based SLA model. Therefore, the proposed approach reduces ambiguity and improves the consistency of knowledge interpretation across heterogeneous SLA formats.

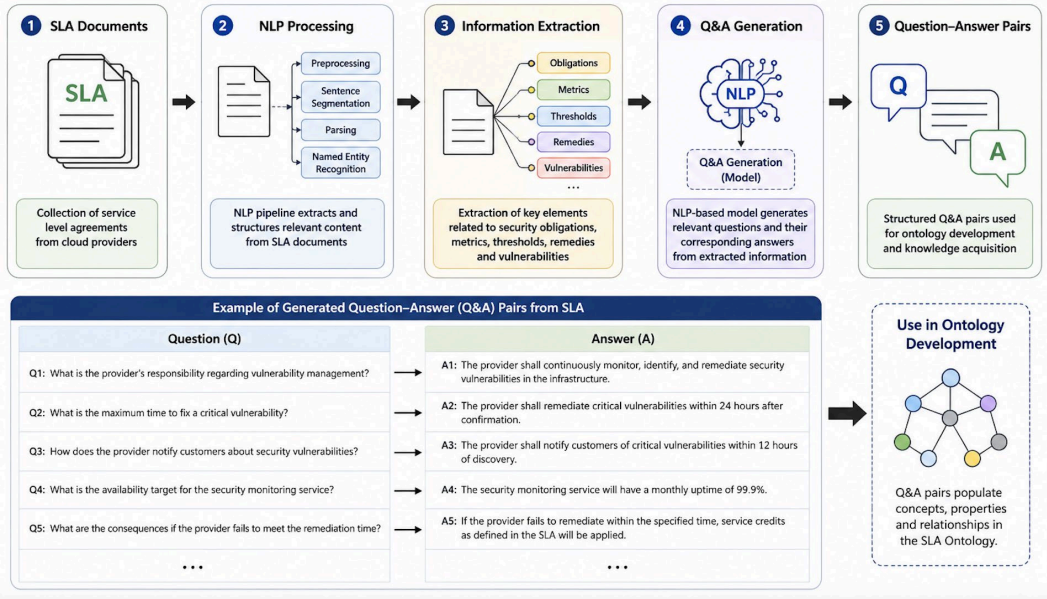


Figure 2. The Q&A generation from SLA documents.

In this study, real-world security-based SLA documents obtained from organizations are utilized to automatically generate domain-specific Q&A pairs for knowledge acquisition. A DistilBERT-based question–answering model is fine-tuned using the Hugging Face framework [35] to extract security-related obligations, vulnerability severity levels, remediation timelines, and availability requirements from unstructured SLA documents. The model is deployed through Hugging Face Spaces and integrated with an API-based interface to support scalable knowledge extraction. The acquired knowledge is subsequently transformed into ontology instances and used to support semantic modeling, consistency verification, and automated reasoning.

As a result, the extracted knowledge serves as the primary input for populating the SLA Security Ontology and establishing semantic relationships among security requirements, compliance obligations, remediation commitments, and service-level guarantees. Consequently, the ontology is directly grounded in knowledge derived from real-world SLA specifications. Therefore, this process enhances machine readability, semantic interoperability, and reasoning readiness. Since the primary objective of this study is ontology development and semantic reasoning, the NLP-based Q&A generation component is utilized solely as a supporting knowledge acquisition mechanism. In addition, the main focus of the study remains on the ontology architecture, semantic integration, and automated reasoning capabilities of the proposed solution.

3.2. SLA Security Ontology

The SLA Security Ontology is developed to provide a machine-readable semantic representation of security-related information contained within cloud SLAs. Traditional SLA specifications are typically expressed in natural language or semi-structured formats, which limits their ability to support automated processing, semantic interoperability, and intelligent compliance verification. To overcome these limitations, the proposed ontology formalizes SLA security concepts, obligations, requirements, and performance metrics within a structured semantic framework. The SLA Security Ontology enables to represent SLA knowledge in an ontology-based format. Hence, security requirements can be linked

to vulnerability information, remediation actions, and compliance verification processes within a unified semantic environment.

The developed SLA Ontology formally defines key concepts related to security vulnerabilities in enterprise cloud systems. The SLA ontology is developed using structured Q&A pairs derived from relevant SLAs and generated by the knowledge acquisition component. These Q&A pairs serve as the foundational knowledge extraction mechanism for ontology development. The Q&A pairs are presented in Figure 3.

```
# Question & Answer List
questions = [
    "How should remediation timeframe for high vulnerability?",
    "What is the high vulnerability cvss score?",
    "How should remediation timeframe for critical vulnerability?",
    "What is the critical vulnerability cvss score?",
    "How should remediation timeframe for medium vulnerability?",
    "What is the medium vulnerability cvss score?",
    "How should remediation timeframe for low vulnerability?",
    "What is the low vulnerability cvss score?",
    "How should remediation timeframe for informational vulnerability?",
    "What is the informational vulnerability cvss score?",
    "What is the timeframe for your vulnerability scans?",
    "What is the timeframe for your pentest activities?",
    "What is the frequency for backups?"
]

answers = [
    "within 30 days", # high
    "between 7.0 - 9.0",  # Example of CVSS score range for high vulnerabilities
    "3 hours", #critical
    "between 1.0 - 2.0", #critical
    "2 months",  # Example using months medium
    "between 6.0 - 7.0", #medium
    "1 hour",  #low
    "between 4.0 - 5.0", #low
    "5 months", #Informational
    "between 0.0 - 0.0", #low
    "per month",    # Example using months
    "monthly schdule",
    "yearly"
]
```

Figure 3. The Q&A pairs used in the SLA Security Ontology.

Following the generation of structured Q&A pairs, an additional semantic analysis stage is required to identify ontology concepts, properties, and relationships that can be incorporated into the SLA Security Ontology. Since the generated answers contain valuable domain knowledge in natural language form, it is necessary to derive semantic representations that can be transformed into ontology classes, object properties, and data properties. To support this process, a language model capable of analyzing the syntactic and semantic structure of sentences is employed.

In this study, the `en_core_web_sm` model provided by the spaCy library [36] is utilized for semantic processing and knowledge extraction. spaCy is an open-source NLP framework that offers a wide range of linguistic analysis capabilities for English-language texts [37]. The framework supports several NLP tasks, including tokenization, part-of-speech tagging, NER, and dependency parsing, which facilitate the extraction of semantically meaningful information from unstructured textual data.

The extracted linguistic features are subsequently analyzed to identify ontology-relevant entities, attributes, and relationships embedded within the generated question–answer pairs. In particular, dependency parsing and NER are employed to detect semantic associations among security requirements, compliance obligations, remediation commit-

ments, service guarantees, and other SLA-related concepts. These associations are then transformed into ontology-compatible representations that support the construction and population of the SLA Security Ontology.

Various functionalities of the spaCy framework are integrated into the ontology generation workflow to enable automated semantic extraction and relationship identification. For example, the extraction of relationships associated with vulnerability severity levels and corresponding remediation obligations requires the analysis of syntactic dependencies and contextual associations within the generated answers. The pseudocode describing the procedure used to identify and establish severity-related relationships within the ontology is presented in Algorithm 1.

Algorithm 1 Identifying and Establishing Vulnerability–Severity Relationships in the SLA Security Ontology.

Input: questions [], answers [] // *SLA clause Q&A pairs*
Output: Ontology populated with Severity instances linked to Customer

```
Initialize NLP model ← load('en_core_web_sm')
Initialize severity_instances ← ∅
customer_instance ← Customer("customer_activities")
    for each (question, answer) in zip(questions, answers) do
        question_doc ← NLP.parse(question)
        severity_tokens ← {token | token ∈ question_doc ∧ token.dep = 'amod'}
        if severity_tokens ≠ ∅ then
            severity_type ← severity_tokens[0]
            if severity_type ∉ severity_instances then
                severity_instance ← Severity(severity_type)
                severity_instances[severity_type] ← severity_instance
            else
                severity_instance ← severity_instances[severity_type]
            end if
            if QuestionType(question) = "RemediationTime" then
                severity_instance.hasRemediationTime ← parse_time(answer.strip())
            end if
        end if
    end for
return severity_instances
```

As seen in Algorithm 1, the algorithm leverages the linguistic analysis capabilities provided by the spaCy framework to extract semantic information from the generated Q&A pairs. In particular, each processed token is analyzed together with its dependency relations to identify vulnerability-related entities and their associated severity descriptors. The extraction process relies on dependency parsing mechanisms, where grammatical relationships among words are utilized to determine semantic associations. Specifically, adjectival modifier (`amod`) dependencies are employed to detect severity-related attributes describing vulnerabilities (e.g., critical, high, medium, or low). These linguistic patterns enable the automatic identification of vulnerability severity information embedded within natural language statements.

If a severity level identified during the extraction process is not already represented within the ontology, a corresponding individual is automatically instantiated under the `Severity` class. This mechanism prevents the loss of previously unseen severity categories

and ensures semantic completeness within the ontology. Subsequently, the identified severity instances are linked to their corresponding vulnerability concepts through ontology relationships. Thus, the semantic representation of security-related knowledge is enriched.

Subsequently, the algorithm extracts remediation time commitments specified within SLA statements to establish semantic links between vulnerability severity levels and their corresponding remediation obligations. These commitments typically describe the maximum time intervals defined by service providers for addressing vulnerabilities of different severity levels. To formally represent this knowledge, the ontology introduces the `hasRemediationTime` property, which associates vulnerability severity levels with their corresponding remediation deadlines. As a result, the ontology captures not only the criticality of vulnerabilities but also the contractual remediation obligations specified in SLAs. Therefore, the ontology provides a machine-readable foundation for subsequent reasoning and compliance verification processes.

In addition, it should be noted that the NLP component is employed solely as a knowledge acquisition layer for ontology population rather than as a standalone information extraction framework. In the current implementation the generated SLA questions follow predefined linguistic patterns in which severity levels are represented as adjective–noun structures (e.g., high vulnerability, critical vulnerability). Therefore, severity identification relies on dependency-based extraction of adjectival modifiers (`amod`). Similarly, remediation time extraction is performed for predefined question templates requesting remediation time information. The identified question type determines the ontology property to be populated, while the temporal value is extracted from the corresponding answer using the `parse_time()` function, which recognizes numerical temporal expressions (e.g., one week, 14 days, and 24 h) and converts them into a standardized day-based representation. Although this design provides an interpretable and reproducible ontology population process, more complex linguistic variations and more diverse non-numerical temporal expressions will be further addressed as future extensions.

Moreover, it should be emphasized that Algorithm 1 operates on the generated Q&A pairs rather than directly on raw SLA documents. Accordingly, spaCy is employed only for lightweight dependency parsing required to identify predefined linguistic patterns used during ontology population. Therefore, the algorithm assumes standardized question templates produced by the preceding knowledge acquisition stage. Supporting more diverse contractual writing styles and complex legal sentence structures through transformer-based extraction or domain-specific NER models will be explored in future research directions.

In addition to modeling vulnerabilities and their associated remediation obligations, the ontology incorporates concepts and relationships for representing semantic inconsistencies that may arise during the knowledge acquisition and ontology population processes. Such inconsistencies can originate from multiple sources, including typographical errors present in SLA documents, ambiguities in contractual language, and incorrect interpretations generated during the Q&A extraction process.

The proposed ontology explicitly represents semantic anomalies as ontology entities that can be detected and analyzed through reasoning mechanisms to support ontology quality assurance and consistency verification. For example, when extracting remediation intervals from SLA statements, the expected value may correspond to a temporal expression such as annually, monthly, or within 24 h. However, if the extracted answer is a semantically incompatible concept such as customer activities, the resulting ontology assertion violates the expected semantic constraints of the corresponding property. Such cases are classified as semantic inconsistencies within the ontology.

The proposed solution enables the identification of invalid ontology assertions and supports automated consistency verification by explicitly modeling semantic inconsistency

concepts and their associated relationships. This capability improves the reliability of ontology population, reduces the propagation of erroneous knowledge, and provides a foundation for the SWRL-based inconsistency detection mechanisms presented in subsequent sections. Figure 4 illustrates the complete class structure of the proposed SLA Security Ontology, which was automatically constructed using the knowledge extracted from the generated question–answer pairs. Table 2 summarizes the primary classes and their semantic roles within the proposed SLA Security Ontology to facilitate the interpretation of the ontology structure presented in Figure 4. These classes collectively represent security-related activities, vulnerability characteristics, remediation requirements, stakeholder entities, and semantic inconsistency concepts used for ontology validation and reasoning.

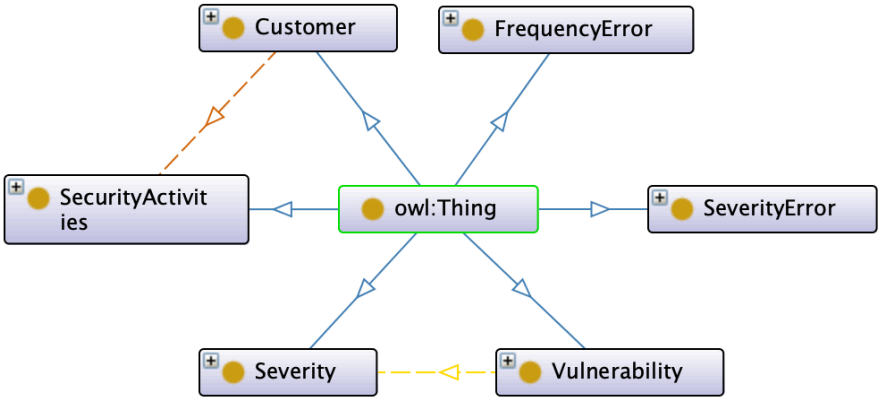


Figure 4. The class structure of the SLA Security Ontology.

Table 2. The class descriptions of the SLA Security Ontology.

Class	Description
`Customer`	Represents organizations that establish or are governed by cloud service level agreements.
`SecurityActivities`	Describes security-related activities, controls, and operational procedures defined within SLA specifications.
`Vulnerability`	Represents cloud-related vulnerabilities identified from SLA statements and associated security knowledge sources.
`Severity`	Defines the criticality level of vulnerabilities. The class contains instances such as `Critical`, `High`, `Medium`, `Low`, `Informational`, and `Pentest`.
`FrequencyError`	Captures semantic inconsistencies related to invalid, misspelled, or contextually inappropriate frequency expressions extracted from SLA statements.
`SeverityError`	Represents semantic anomalies associated with invalid or incorrectly classified vulnerability severity levels that may compromise ontology consistency.

Unlike existing conventional SLA ontologies that primarily focus on contractual entities and service metrics, the proposed ontology explicitly incorporates semantic inconsistency concepts such as `FrequencyError` and `SeverityError`. These classes enable the detection of ontology population errors and provide a semantic foundation for the consistency verification and SWRL-based reasoning mechanisms.

In addition to the ontology classes, a set of semantic relationships is defined to capture dependencies among vulnerabilities, severity levels, remediation obligations, security activities, and inconsistency concepts. These object properties enable machine-readable representation of security knowledge and provide the foundation for semantic reasoning and ontology alignment. Table 3 summarizes the principal object properties defined within the SLA Security Ontology.

Table 3. The property descriptions of the SLA Security Ontology.

Object Property	Description
`hasSecurityActivities`	Establishes a semantic relationship between the `Customer` and `SecurityActivities` classes, linking organizations to the security activities defined within their SLAs.
`hasSeverity`	Associates a `Vulnerability` instance with its corresponding `Severity` level, enabling the representation of vulnerability criticality within the ontology.
Data Property	**Description**
`hasCVSS_Score`	Stores the numerical Common Vulnerability Scoring System (CVSS) score associated with a vulnerability severity level, providing a standardized measure for vulnerability assessment and risk prioritization.
`hasFrequency`	Specifies the execution frequency of a security activity (e.g., daily, weekly, monthly, or annually) associated with the `SecurityActivities` class.
`hasRemediationTime`	Defines the remediation timeframe associated with a particular vulnerability severity level, representing the maximum resolution period specified within the SLA.

The object and data properties defined within the ontology establish semantic connections among security concepts and provide the structural foundation for ontology alignment, automated reasoning, and consistency verification mechanisms.

3.3. Vulnerability Ontology

The Vulnerability Ontology is developed to provide a structured semantic representation of security vulnerabilities associated with cloud services. The ontology is constructed based on the vulnerability findings obtained from cloud service security assessments performed through Amazon Inspector [5] to ensure the use of realistic and operationally relevant vulnerability information. The identified vulnerabilities and their associated metadata constitute the primary knowledge source used for ontology development.

The vulnerability information provided by Amazon Inspector is first transformed into a structured JSON format to enable automated ontology generation. This intermediate representation facilitates the processing of vulnerability data and supports its conversion into ontology-compatible structures. In addition, the generated JSON data can be exposed through API services, allowing vulnerability information to be accessed and utilized by external applications. Following the JSON transformation process, a set of vulnerability-related concepts suitable for integration with the SLA Security Ontology is identified and modeled within the ontology. The required ontology entities and relationships are automatically generated using the information extracted from the JSON structure. The resulting ontology is subsequently stored in OWL format. The automated process for transforming the structured vulnerability information into ontology concepts, properties, and instances is presented in Algorithm 2. Algorithm 2 provides a high-level description of the auto-

mated Vulnerability Ontology generation process. For simplicity, implementation-specific mechanisms such as API pagination, exception handling, and incremental synchronization are omitted, as they do not affect the ontology construction methodology presented in this study.

Algorithm 2 Automated generation of the Vulnerability Ontology.

Input: None
Output: RDF graph 'g'

```
Initialize namespaces: RDF, RDFS, OWL, n
g ← CreateEmptyGraph()
// Define Class and Properties
g.add((n.Finding, RDF.type, owl.Class))
g.add((n.description, RDF.type, owl.DatatypeProperty))
g.add((n.description, rdfs.domain, n.Finding))
// . . . (continue for other properties and classes)
// Fetch Data from API
response ← HTTP_GET('API URL')
findings_list ← JSON_PARSE(response)
for each data in findings_list do
    finding_id ← data["findingArn"]
    description ← data["description"]
    // . . . (Additional data extraction here)
    finding_node ← URIRef(n[finding_id])
    g.add((finding_node, RDF.type, n.Finding))
    if description ≠ NULL then
        g.add((finding_node, n.description, Literal(description)))
    end if
    // . . . (continue for other properties, similar to description)
end for
g.serialize(destination = 'output.owl', format = 'xml')
end
```

The class hierarchy of the Vulnerability Ontology is shown in Figure 5. Table 4 summarizes the concepts and properties defined within the Vulnerability Ontology.

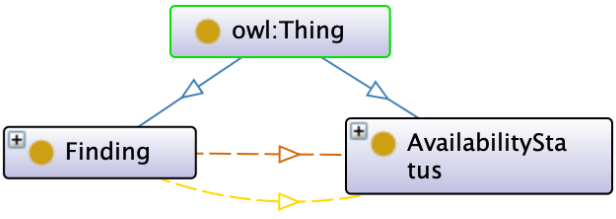


Figure 5. The class view of the Vulnerability Ontology.

The resulting Vulnerability Ontology provides a formal representation of cloud security vulnerabilities and their associated characteristics. Thus, it enables semantic integration, automated reasoning, and vulnerability-aware compliance assessment within the proposed solution.

Table 4. The property descriptions of the Vulnerability Ontology.

Class	Description
`Finding`	Represents vulnerabilities identified in cloud services.
`AvailabilityStatus`	Represents the availability status associated with exploit and remediation information.
Object Property	**Description**
`exploitAvailable`	Indicates whether publicly available exploit code, attack procedures, or proof-of-concept implementations exist for a given vulnerability.
`fixAvailable`	Describes whether a remediation mechanism exists for the identified vulnerability.
Data Property	**Description**
`cveScore`	Represents the Common Vulnerabilities and Exposures (CVE) score to assess the severity and prioritization of vulnerabilities.
`description`	Provides a textual description of the vulnerability.
`firstObservedAt`	Specifies the date and time at which the vulnerability was first detected.
`lastObservedAt`	Specifies the most recent observation time of the vulnerability.
`severity`	Indicates the severity level assigned to the vulnerability.
`source`	Indicates the origin of the vulnerability information.
Individual	**Description**
`YES`	Indicates that the corresponding exploit or remediation is available.
`NO`	Indicates that the corresponding exploit or remediation is unavailable.
`PARTIAL`	Indicates partially available remediation or exploit information (when applicable).

3.4. Ontology Alignment and Semantic Integration

The SLA Security Ontology and the Vulnerability Ontology are developed to represent complementary aspects of cloud security knowledge. While the SLA Security Ontology captures contractual security requirements, compliance obligations, remediation commitments, and service-level guarantees, the Vulnerability Ontology represents vulnerability findings, severity levels, exploitability information, remediation availability, and associated security metadata. Hence, an ontology alignment and semantic integration process is performed to enable interoperability between these knowledge sources.

The objective of the alignment process is to establish semantic correspondences between concepts originating from the related two ontologies and to create a unified knowledge representation capable of supporting automated reasoning and security analysis. Rather than maintaining isolated semantic models, the proposed approach integrates SLA-related security obligations with vulnerability intelligence. Therefore, the proposed approach enables machine-readable security semantics across contractual and operational security domains.

The alignment process begins with the identification of semantically related concepts included in both ontologies. In the scope of this study, vulnerability severity levels, remediation requirements, security activities, and compliance obligations serve as primary

integration points between the SLA Security Ontology and the Vulnerability Ontology. Therefore, semantic correspondences are established among these concepts. Hence, these concepts provide natural semantic bridges between contractual SLA commitments and the vulnerabilities detected within cloud environments. For example, severity levels represented in the Vulnerability Ontology are associated with remediation commitments and response time requirements defined within the SLA Security Ontology. This relationship enables the proposed solution to determine whether remediation obligations specified in SLAs are consistent with the severity of the detected vulnerabilities. To support semantic interoperability, a set of cross-ontology relationships is introduced. These relationships connect vulnerability findings to SLA obligations, remediation timeframes, and security activities. As a result, knowledge originating from heterogeneous sources can be represented within a common semantic framework. The integrated ontology, referred to as the SLAVUL Ontology, combines contractual security knowledge with vulnerability intelligence and provides a unified representation of cloud security semantics.

The resulting SLAVUL Ontology enables security-related concepts originating from different sources to be interpreted in a consistent and machine-processable manner. Furthermore, the integrated semantic structure serves as the foundation for automated reasoning, inconsistency detection, and remediation obligation verification mechanisms. By establishing explicit semantic links between SLA commitments and vulnerability characteristics, the proposed approach facilitates vulnerability-aware compliance assessment and enhances the interoperability of cloud security knowledge.

Figure 6 illustrates the ontology alignment process and the resulting semantic integration between the SLA Security Ontology and the Vulnerability Ontology.

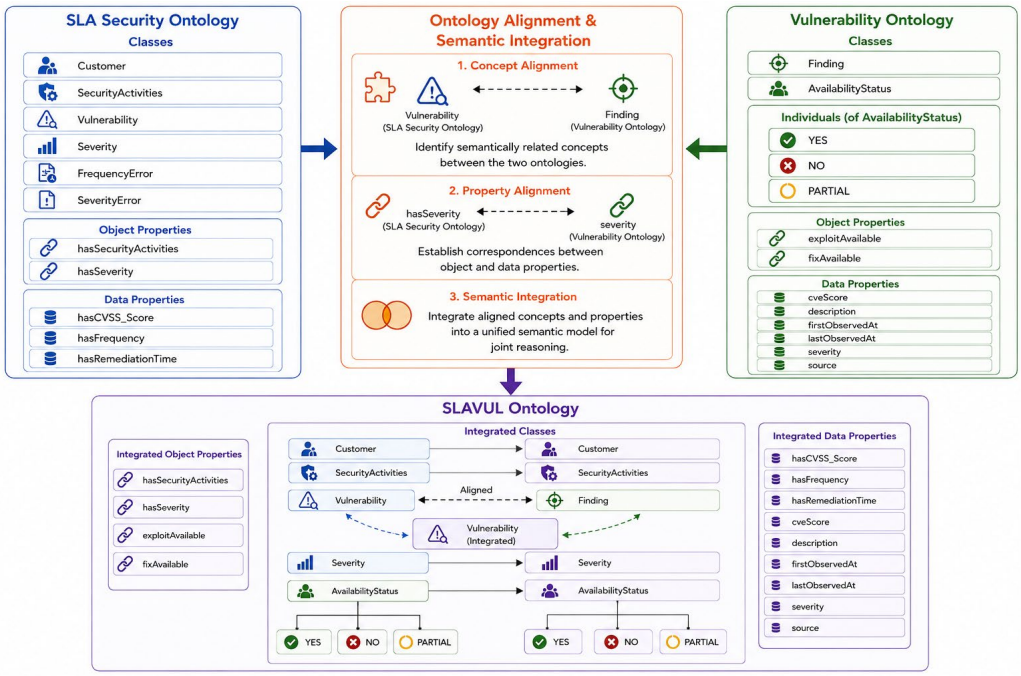


Figure 6. The semantic integration of ontologies.

The integrated ontology is represented under the namespace `http://www.example.org/slavul#`, which provides a common URI space for the concepts originating from the SLA Security Ontology and the Vulnerability Ontology.

The semantic alignment between the SLA Security Ontology and the Vulnerability Ontology was established through manually defined semantic correspondences encoded as SWRL rules. Automatic ontology matching tools are not employed since both ontologies are

specifically designed within the scope of the proposed solution to allow explicit definition of the required semantic mappings. Therefore, ontology alignment is knowledge-driven rather than similarity-driven by relying on explicitly defined semantic correspondences instead of automatically inferred ontology similarities.

3.5. SWRL-Based Reasoning and Consistency Verification

Semantic interoperability alone is insufficient for supporting automated security assessment and compliance verification within integrated ontology environments. Effective utilization of the SLAVUL Ontology requires reasoning capabilities that can infer implicit knowledge, establish semantic correspondences across heterogeneous concepts, and identify logical inconsistencies within the represented security knowledge. For this purpose, a set of SWRL rules is defined and integrated into the ontology. The defined rules support ontology integration, vulnerability–SLA alignment, remediation obligation propagation, and consistency verification. The SWRL rules defined for the SLAVUL Ontology is presented in Table 5.

Table 5. SWRL rules for the SLAVUL Ontology.

Rule #	Rule	Description
R1	$\text{Finding}(?f) \rightarrow \text{Vulnerability}(?f)$	Maps the `Finding` instance with `Vulnerability`.
R2	$\text{Vulnerability}(?f) \wedge \text{severity}(?f, \text{“LOW”}) \rightarrow \text{hasSeverity}(?f,\text{low})$	Defines `low` severity findings.
R3	$\text{Vulnerability}(?f) \wedge \text{severity}(?f, \text{“HIGH”}) \rightarrow \text{hasSeverity}(?f,\text{high})$	Defines `high` severity findings.
R4	$\text{Vulnerability}(?v) \wedge \text{hasSeverity}(?v, ?s) \wedge \text{Severity}(?s) \wedge \text{hasRemediationTime}(?s,?t) \rightarrow \text{hasRemediationTime}(?v,?t)$	Maps `Vulnerability` to remediation times based on their `Severity` levels.

The first group of rules focuses on ontology integration and semantic enrichment. The primary alignment rule establishes a correspondence between the `Finding` class of the Vulnerability Ontology and the `Vulnerability` class of the SLA Security Ontology. Through this rule, every vulnerability finding identified within cloud environments is semantically interpreted as an SLA-related vulnerability instance. Consequently, vulnerability information obtained from operational security assessments can be evaluated together with contractual security obligations defined in SLAs. This semantic correspondence forms the foundation of the integrated SLAVUL Ontology.

After establishing the relationship between `Finding` and `Vulnerability`, additional SWRL rules are introduced to align vulnerability severity levels across the two ontologies. Specifically, vulnerability findings classified as low, high, and other severity levels are mapped to their corresponding severity instances within the SLA Security Ontology. As a result, both ontologies share a common severity representation, enabling uniform interpretation of vulnerability criticality across heterogeneous knowledge sources.

Subsequent to severity-level alignment, remediation commitments defined within SLA documents are propagated to the corresponding vulnerability instances. The associated SWRL rule transfers remediation time requirements from SLA-defined severity categories to vulnerabilities exhibiting the same severity level. This reasoning mechanism allows detected vulnerabilities to inherit contractual remediation obligations. Therefore, direct semantic connection between vulnerability findings and SLA compliance requirements is established.

In the scope of the study, reasoning and rule execution are performed using the Pellet reasoner [38]. Following rule execution, vulnerability findings originating from

the Vulnerability Ontology are automatically inferred as instances of the `Vulnerability` class within the SLAVUL Ontology. Furthermore, remediation obligations and severity-related knowledge are automatically propagated through the integrated ontology structure, enabling machine-processable security semantics and automated compliance assessment.

In addition to ontology integration, SWRL rules are employed to verify the logical consistency of SLA-derived security knowledge. The SLA Security Ontology is constructed from knowledge extracted through the NLP-based question–answer generation process. Consequently, logical inconsistencies may arise due to incorrect model outputs, ambiguities in SLA documents, or semantically invalid contractual statements. Such inconsistencies may negatively affect vulnerability management processes and lead to inaccurate reasoning results. To address this challenge, a dedicated set of SWRL-based consistency verification rules is integrated into the ontology. Table 6 presents some examples of these rules.

Table 6. The consistency verification rules.

Rule #	Rule	Description
R1	Severity(low) ∧ hasRemediationTime(low,?time1) ∧ Severity(high) ∧ hasRemediationTime(high,?time2) ∧ swrlb:lessThan(?time1,?time2) → LogicalError(low) ∧ LogicalError(high)	Time-to-resolve rule for low and high-criticality vulnerabilities.
R2	Severity(medium) ∧ hasCVSS_Score(medium,?v1) ∧ Severity(critical) ∧ hasCVSS_Score(critical,?v2) ∧ swrlb:lessThan(?v2, ?v1) → LogicalError(medium) ∧ LogicalError(critical)	CVSS scoring rule for medium and critical levels.

In Table 6, the first consistency verification rule evaluates remediation time commitments associated with different vulnerability severity levels. From a security management perspective, remediation times are expected to follow a logical ordering in which higher-severity vulnerabilities require faster remediation than lower-severity vulnerabilities. For example, vulnerabilities classified as `Critical` should generally have shorter remediation periods than those classified as `High`, `Medium`, or `Low`. If this ordering is violated, the corresponding severity instances are automatically classified under the `LogicalError` class, indicating a potential inconsistency within the SLA knowledge base.

The second consistency verification rule examines the consistency of severity assignments using CVSS scores. According to standard vulnerability assessment practices, higher severity levels should be associated with higher CVSS values. Therefore, the rule compares CVSS scores assigned to different severity categories and identifies situations where lower-severity categories are associated with higher risk scores than more critical categories. When such inconsistencies are detected, the corresponding ontology entities are automatically classified as instances of the `LogicalError` class.

The proposed SWRL-based verification mechanism provides an additional semantic validation layer. On the other hand, no logical consistency rules are defined for the Vulnerability Ontology. This decision is based on the fact that vulnerability information is obtained directly from Amazon Inspector, a validated cloud security assessment platform that provides quality-controlled vulnerability data. Therefore, logical inconsistencies are primarily expected to originate from SLA-derived knowledge rather than vulnerability findings. Through the integration of automated reasoning and consistency verification mechanisms, the SLAVUL Ontology supports both semantic interoperability and trustworthy security knowledge representation.

4. Results and Discussion

The primary objective of the proposed approach is not only to represent SLA security knowledge in a machine-readable format but also to automatically identify semantic inconsistencies that may affect vulnerability management and compliance assessment processes. To evaluate the effectiveness of the proposed ontology-driven reasoning mechanism, SLA-specific ontologies are generated from the question–answer pairs produced during the knowledge acquisition stage. Subsequently, the SWRL rules described in the previous section are executed using the Pellet reasoner to identify logical inconsistencies and automatically classify erroneous ontology entities under the `LogicalError` class.

The experimental evaluation is conducted on 26 SLA documents containing at least one incorrectly generated answer. For each SLA document, the developed ontology is populated with the corresponding question–answer pairs and subjected to consistency verification through the proposed SWRL rules. The reasoning mechanism successfully detected logical inconsistencies in 19 out of 26 SLA documents, yielding a detection rate of 73%. The remaining 7 cases are not flagged by the ontology-based validation process. The overall distribution of detected and undetected errors is presented in Figure 7.

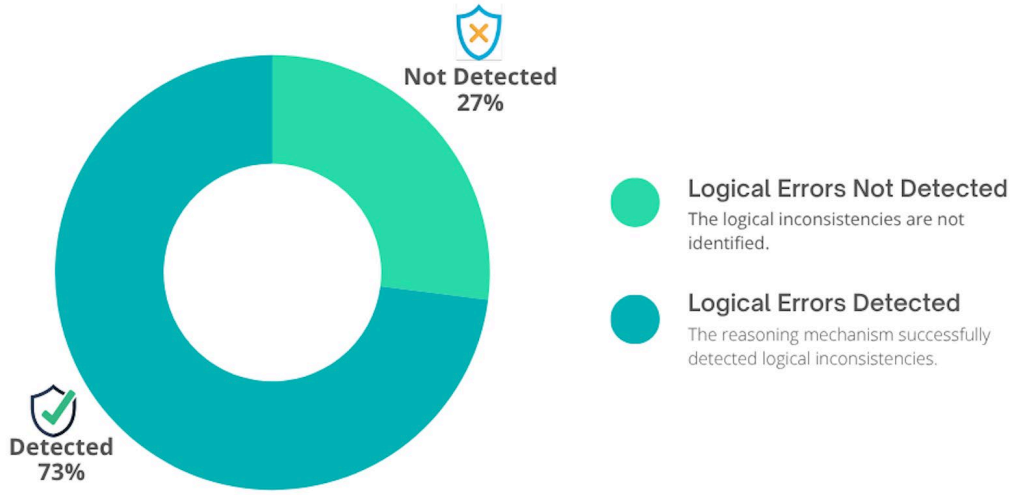


Figure 7. Logical inconsistency detection results.

A detailed investigation of the undetected cases revealed a fundamental characteristic of ontology-based consistency verification. In all seven instances, the generated answers were factually inconsistent with the original SLA content, yet remained logically coherent within the semantic constraints encoded in the ontology. Consequently, the SWRL rules could not classify these instances as logical errors, as no structural or constraint-level violation was present. This finding highlights an inherent limitation of rule-based reasoning: the mechanism is capable of detecting violations of explicitly encoded constraints, but cannot identify factual deviations that do not result in logical inconsistencies within the ontology.

A detailed investigation of the undetected cases revealed an important characteristic of ontology-based consistency verification. In all seven cases, the generated answers are inconsistent with the original SLA content but remained logically coherent within the semantic constraints encoded in the ontology. As a result, the SWRL rules could not classify these instances as logical errors.

Furthermore, the SLA document is queried using the questions "*What is the remediation timeframe for high-severity vulnerabilities?*" and "*What is the remediation timeframe for medium-severity vulnerabilities?*". The SLA model generated the answers "*1 business day*" and "*5 business days*", respectively. From a logical reasoning perspective, these values satisfy the

expected ordering constraints because higher-severity vulnerabilities are associated with shorter remediation periods than lower-severity vulnerabilities. However, examination of the original SLA document revealed that the correct remediation commitments are "*5 business days*" and "*10 business days*". Consequently, although the generated answers are contractually incorrect, they did not violate any logical relationships represented within the ontology. The representative example is illustrated in Figures 8 and 9, respectively. Although the generated remediation timeframes satisfy the logical ordering constraint encoded in the ontology (high severity < medium severity), both values deviate from the contractual commitments specified in the original SLA document.

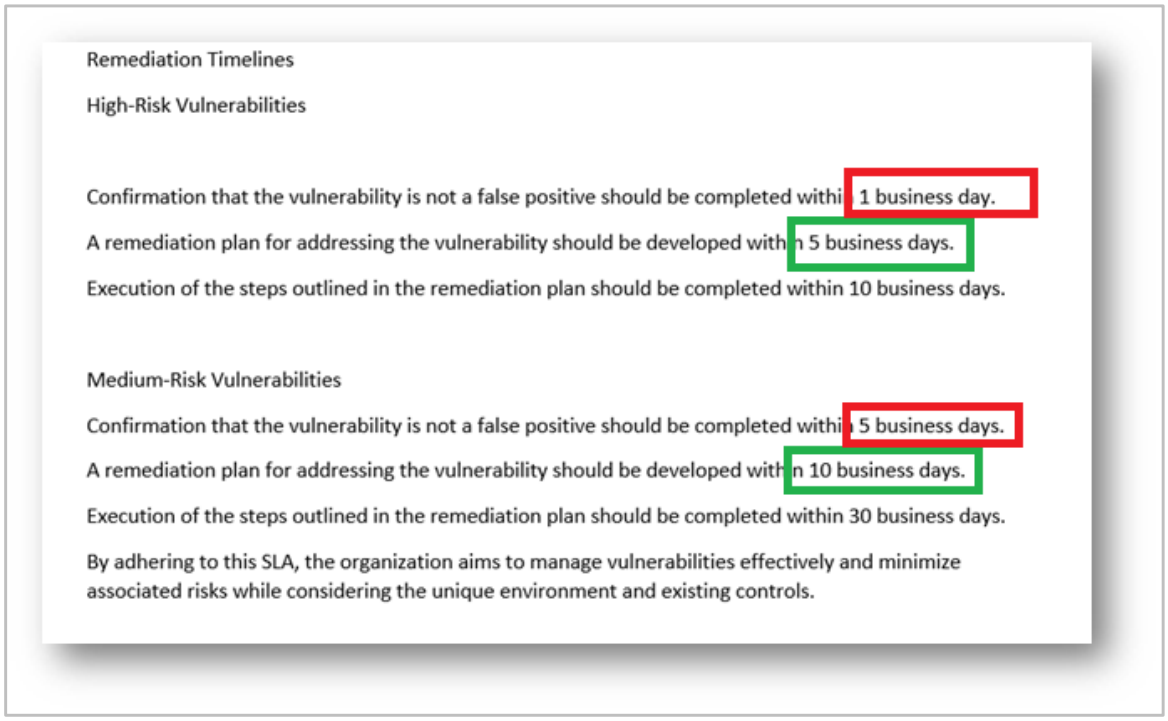


Figure 8. A representative SLA example.



Figure 9. An example of semantically valid but contractually incorrect output.

This observation highlights a fundamental distinction between "logical consistency" and "contractual correctness". The proposed SWRL rules are designed to identify violations of semantic constraints and domain knowledge encoded within the ontology. Therefore, the framework is highly effective in detecting reasoning-level inconsistencies, such as incorrect severity ordering, invalid remediation commitments, and semantically contradictory relationships. However, errors that preserve logical consistency while deviating from the original contractual statements cannot be identified solely through ontology-based reasoning.

The obtained results demonstrate that ontology-driven validation can serve as an effective quality assurance mechanism for SLA-derived security knowledge. The achieved detection rate of approximately 73% indicates that a substantial portion of erroneous model outputs can be identified through semantic reasoning without requiring manual inspection. This capability is particularly important in large-scale cloud environments where hundreds of SLA documents may need to be analyzed and continuously monitored.

The experimental findings also demonstrate the complementary nature of ontology-based reasoning and NLP-based knowledge extraction. While the NLP component is responsible for transforming unstructured SLA text into structured knowledge, the ontology layer provides semantic validation capabilities that assess the logical consistency of the extracted information. The interaction between these two components reduces the likelihood of propagating semantically invalid knowledge into downstream vulnerability assessment and compliance management processes.

From a broader perspective, the results confirm the practical value of integrating ontology engineering and rule-based reasoning for cloud security management. The proposed SLAVUL Ontology not only enables semantic interoperability between SLA obligations and vulnerability intelligence but also provides an automated mechanism for detecting knowledge inconsistencies before they influence security decision-making processes. Such capabilities are particularly valuable in environments where remediation commitments, vulnerability severity levels, and compliance requirements must be continuously evaluated in a consistent and machine-interpretable manner.

It should be noted that the experimental dataset is intentionally constructed to include SLA documents containing at least one automatically generated incorrect answer. The objective of this evaluation is to assess the effectiveness of the proposed ontology-driven reasoning mechanism in identifying logical inconsistencies rather than to estimate their occurrence in real-world SLA documents. Since the ontology population, ontology alignment, and SWRL-based reasoning processes are independent of the presence of logical errors, the proposed framework is equally applicable to correctly drafted SLA documents, where no inconsistency instances would be generated during reasoning.

In addition, the proposed solution is evaluated to validate the correctness of semantic reasoning rather than the computational efficiency of the underlying ontology reasoner. Therefore, metrics such as reasoning latency, memory consumption, and large-scale scalability were not investigated in this study. Evaluating the computational performance of different OWL reasoners under large-scale cloud vulnerability datasets constitutes an important direction for future work.

Despite these promising results, the study also reveals an inherent limitation of rule-based consistency verification. The effectiveness of SWRL reasoning is constrained by the completeness of the semantic rules and domain constraints represented within the ontology. Contract-specific inaccuracies that do not violate logical relationships remain difficult to detect automatically and may require document-level verification or expert review. Future research may address this limitation by combining ontology-based reasoning with Large Language Model (LLM)-assisted semantic verification techniques capable of directly comparing generated knowledge against the original contractual text. Another limitation of the proposed knowledge acquisition process is that it assumes normalized Q&A pairs generated from SLA documents. Consequently, errors introduced during the knowledge acquisition stage may propagate to ontology population. Although the proposed SWRL-based reasoning mechanism can identify semantic and logical inconsistencies after ontology population, improving the robustness of the knowledge acquisition process through more advanced semantic parsing techniques remains an important direction for future work. Furthermore, although the proposed SWRL reasoning mechanism effectively

detects violations of explicitly encoded semantic constraints, it cannot identify contractually incorrect assertions that remain logically consistent with the ontology. Such cases originate from inaccuracies introduced during knowledge acquisition rather than inconsistencies within the ontology itself. A promising direction for future research is to complement ontology-based reasoning with a contract verification module that compares extracted ontology assertions against the original SLA text using semantic retrieval or advanced language understanding techniques. This hybrid verification strategy would enable the detection of both logical inconsistencies and factually incorrect contractual interpretations. Therefore, the reliability of automated SLA knowledge acquisition would be improved. Finally, while the experimental dataset is sufficient to validate the reasoning capabilities of the proposed framework, a larger and more heterogeneous collection of SLA documents would provide a more comprehensive assessment of its generalizability across different cloud service providers and contractual writing styles.

5. Conclusions and Future Works

The increasing complexity of cloud service ecosystems has created a growing need for machine-readable representations of security obligations, vulnerability information, and compliance requirements. Although SLAs play a critical role in defining security responsibilities between cloud providers and customers, most SLA documents remain unstructured and difficult to integrate with operational security intelligence. Addressing this challenge requires not only the formal representation of contractual security knowledge but also mechanisms capable of linking SLA obligations with vulnerability management processes through semantic reasoning.

This study presents an ontology-driven approach that integrates SLA security knowledge and cloud vulnerability information within a unified semantic model. The proposed approach combines an SLA Security Ontology, automatically constructed from SLA-derived knowledge, with a Vulnerability Ontology generated from vulnerability findings obtained through cloud security assessment processes. The integration of these ontologies enables the establishment of machine-readable security semantics and provides a common semantic foundation for vulnerability assessment, compliance monitoring, and security governance activities.

The main contribution of the study is the semantic alignment between SLA obligations and vulnerability intelligence through ontology integration mechanisms. The proposed SLAVUL Ontology establishes interoperability between heterogeneous security knowledge sources by aligning vulnerability findings, severity classifications, remediation commitments, and security activities within a unified ontology structure. This integration enables vulnerability information originating from operational cloud environments to be interpreted in the context of contractual security requirements defined within SLAs.

Another important contribution of the study is the integration of SWRL-based reasoning mechanism to enable automated inference, remediation obligation alignment, and consistency verification. The developed reasoning framework automatically propagates remediation obligations, aligns vulnerability severity levels across ontologies, and infers implicit security knowledge that cannot be explicitly represented through ontology structures alone. In addition, a set of consistency verification rules is defined to identify logical inconsistencies in SLA-derived security knowledge. Experimental evaluation demonstrated that the proposed reasoning mechanism successfully detected logical errors in 73% of the evaluated SLA documents. Detailed analysis revealed that the remaining undetected cases primarily corresponded to contractually incorrect but logically consistent answers, highlighting the distinction between logical consistency and contractual correctness in ontology-driven validation environments.

Consequently, the proposed study contributes to the literature in several ways. First, it introduces an integrated ontology model that combines SLA security semantics and vulnerability intelligence within a single knowledge representation framework. Second, it demonstrates how semantic interoperability can be achieved between contractual security requirements and operational vulnerability data through ontology alignment. Third, it presents a practical application of SWRL-based automated reasoning for remediation obligation alignment and consistency verification in cloud security management. Finally, the study provides a mechanism for detecting semantic inconsistencies before they propagate into security assessment and compliance decision-making processes.

The findings of this study also reveal several opportunities for extending the current framework and enhancing its practical applicability. The current consistency verification framework primarily focuses on logical constraints represented through manually defined SWRL rules. In future studies, we will investigate the integration of LLMs and ontology-based reasoning mechanisms to support contract-level semantic verification and improve the detection of logically consistent yet contractually incorrect responses. In addition, the use of transformer-based information extraction and domain-specific NER models will be explored to improve the robustness of ontology population under heterogeneous SLA writing styles and complex contractual language. Future work will also investigate complementary semantic verification mechanisms for validating extracted SLA knowledge against the original contractual documents. Such mechanisms will enable the detection of factually incorrect assertions that remain logically consistent within the ontology. Hence, more advanced semantic parsing techniques will be investigated and a quantitative evaluation of the knowledge acquisition component will be conducted to assess extraction accuracy and further improve the robustness and generalizability of ontology population from cloud SLA documents. In addition, the ontology will be extended to incorporate regulatory compliance frameworks and industry-specific security standards, enabling broader compliance assessment capabilities. Further, dynamic ontology updating mechanisms will be developed to continuously incorporate newly discovered vulnerabilities and evolving SLA requirements into the proposed SLAVUL Ontology. In addition, future work will focus on extending the ontology generation module with robust API interaction mechanisms, including pagination support, exception handling, and incremental ontology synchronization to improve scalability and maintainability. Moreover, the proposed approach will be evaluated using larger and more diverse SLA datasets obtained from multiple cloud service providers, including correctly drafted SLA documents with varying contractual structures and writing styles, to further validate its scalability, generalizability, and practical applicability in real-world cloud security environments. Finally, future work will also include a systematic performance evaluation of ontology reasoning under large-scale cloud security datasets, including reasoning latency, memory consumption, and scalability analyses using different OWL reasoners.

Future work will also investigate transformer-based information extraction and domain-specific NER models to improve the robustness of ontology population under heterogeneous SLA writing styles and complex contractual language.

In conclusion, the proposed study demonstrates that ontology engineering, semantic interoperability, and rule-based reasoning can be effectively combined to bridge the gap between contractual security obligations and operational vulnerability management. By providing machine-readable security semantics, automated remediation obligation alignment, and logical inconsistency detection capabilities, the proposed approach establishes a foundation for more intelligent, transparent, and semantically aware cloud security management systems.

Author Contributions: Conceptualization, O.C. and S.Y.; methodology and implementation, S.Y.; validation, O.C. and S.Y.; formal analysis, O.C. and S.Y.; writing—original draft preparation, O.C. and S.Y.; writing—review and editing, O.C.; visualization, O.C. and S.Y.; supervision, O.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data is unavailable due to privacy or ethical restrictions.

Acknowledgments: The authors would like to thank Murat KANTARCIOGLU for his expert guidance, insightful comments and constructive discussions throughout the development of this work. During the preparation of our manuscript, the authors used GPT-5.5 for the purposes of language polishing and also drawing Figures 1, 2 and 6. The authors have reviewed and edited the output and take full responsibility for the content of this publication.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AWS	Amazon Web Services
CSP	Cloud Service Provider
CSU	Cloud Service User
LLM	Large Language Model
NER	Named Entity Recognition
NLP	Natural Language Processing
Q&A	Question &Answer
QoS	Quality of Service
SLA	Service Level Agreement
SLO	Service Level Objective
SW	Semantic Web
SWRL	Semantic Web Rule Language

References

1. Qazi, F.; Kwak, D.; Khan, F.G.; Ali, F.; Khan, S.U. Service Level Agreement in cloud computing: Taxonomy, prospects, and challenges. *Internet Things* **2024**, *25*, 101126. [CrossRef]
2. Nicolazzo, S.; Nocera, A.; Pedrycz, W. Service Level Agreement (SLA) and Security SLA (SecSLA): A Comprehensive Survey. *J. Netw. Syst. Manag.* **2026**, *34*, 74. [CrossRef]
3. Rios, E.; Higuero, M.; Larrucea, X.; Rak, M.; Casola, V.; Iturbe, E. Security and Privacy Service Level Agreement composition for Internet of Things systems on top of standard controls. *Comput. Electr. Eng.* **2022**, *98*, 107690. [CrossRef]
4. Choi, C.; Choi, J. Ontology-Based Security Context Reasoning for Power IoT-Cloud Security Service. *IEEE Access* **2019**, *7*, 110510–110517. [CrossRef]
5. Amazon Inspector. Available online: https://docs.aws.amazon.com/inspector/ (accessed on 26 June 2026).
6. Nehra, P.; Kesswani, N. A workload prediction model for reducing service level agreement violations in cloud data centers. *Decis. Anal. J.* **2024**, *11*, 100463. [CrossRef]
7. Ali, T.; Khan, H.U.; Nazir, B.; Alarfaj, F.K.; Alreshood, M. Optimizing service level agreement in cloud computing with smart virtual machine scheduling using clustered differential evolution and deep learning. *J. Netw. Comput. Appl.* **2025**, *244*, 104361. [CrossRef]
8. Moghaddam, S.M.; O'Sullivan, M.; Unsworth, C.P.; Piraghaj, S.F.; Walker, C. Metrics for improving the management of Cloud environments—Load balancing using measures of Quality of Service, Service Level Agreement Violations and energy consumption. *Future Gener. Comput. Syst.* **2021**, *123*, 142–155. [CrossRef]
9. Ghandour, O.; El Kafhali, S.; Hanini, M. Adaptive workload management in cloud computing for service level agreements compliance and resource optimization. *Comput. Electr. Eng.* **2024**, *120*, 109712. [CrossRef]
10. Lucz, G.; Forstner, B. Optimizing Service Level Agreement Tier Selection in Online Services Through Legacy Lifecycle Profile and Support Analysis: A Quantitative Approach. *Mathematics* **2025**, *13*, 1743. [CrossRef]

11. Da Silva, C.A.; De Geus, P.L. An approach to security-SLA in cloud computing environment. In Proceedings of the IEEE Latin-America Conference on Communications (LATINCOM), Cartagena, Colombia, 5–7 November 2014; pp. 1–6.
12. Zhou, S.; Wu, L.; Jin, C. A Privacy-Based SLA Violation Detection Model for the Security of Cloud Computing. *China Commun.* **2017**, *14*, 155–165. [CrossRef]
13. Nugraha, Y.; Martin, A. Towards the Classification of Confidentiality Capabilities in Trustworthy Service Level Agreements. In *Proceedings of the 2017 IEEE International Conference on Cloud Engineering (IC2E), Vancouver, BC, Canada, 4–7 April 2017*; IEEE: New York, NY, USA, 2017; pp. 304–310.
14. Halabi, T.; Bellaiche, M. A broker-based framework for standardization and management of Cloud Security-SLAs. *Comput. Secur.* **2018**, *75*, 59–71. [CrossRef]
15. Teshome, A.; Rilling, L.; Morin, C. Verification for security monitoring SLAs in IaaS clouds: The example of a network IDS. In Proceedings of the NOMS 2018–2018 IEEE/IFIP Network Operations and Management Symposium, Taipei, Taiwan, 23–27 April 2018; pp. 1–7.
16. Casola, V.; De Benedictis, A.; Rak, M.; Villano, U. A novel Security-by-Design methodology: Modeling and assessing security by SLAs with a quantitative approach. *J. Syst. Softw.* **2020**, *163*, 110537. [CrossRef]
17. Bocianiak, K.; Pawlikowski, T.; Podlasek, A.; Wary, J.P.; Wierzbowski, J. Challenges for continuous, provable Security Service Level Agreement management in computing continuum. *IEEE Access* **2024**, *12*, 152097. [CrossRef]
18. Hosseini, S.; De Miguel, I.; Merayo, N.; De Dios, Ó.G.; Lorenzo, R.M.; Barroso, R.J.D. Enabling Service Level Agreement-Differentiated Protection in C+L Multiband Optical Networks. *IEEE Open J. Commun. Soc.* **2024**, *6*, 316–331. [CrossRef]
19. Asensio-Garriga, R.; Molina Zarca, A.; Ortiz, J.; Hermosilla, A.; Pascual, H.R.; Pastor, A.; Skarmeta, A. ZSM Framework for Autonomous Security Service Level Agreement Life-Cycle Management in B5G Networks. *Future Internet* **2025**, *17*, 86. [CrossRef]
20. Pradeepa, P.; Lakshmi, P.R. Violation Detection in Service Level Agreement to Ensure the Privacy in Cloud Community using Chicken Spider Monkey Optimization-Based Deep Belief Network. *Wirel. Pers. Commun.* **2021**, *117*, 1659–1683. [CrossRef]
21. Liang, L. Electric security data integration framework based on ontology reasoning. *Procedia Comput. Sci.* **2018**, *139*, 583–587. [CrossRef]
22. Liccardo, L.; Rak, M.; Di Modica, G.; Tomarchio, O. Ontology-based Negotiation of Security Requirements in Cloud. In Proceedings of the Fourth International Conference on Computational Aspects of Social Networks (CASoN), Sao Carlos, Brazil, 21–23 November 2012; pp. 192–197.
23. Singh, V.; Pandey, S.K. A comparative study of Cloud Security Ontologies. In Proceedings of the 3rd International Conference on Reliability, Infocom Technologies and Optimization, Noida, India, 8–10 October 2014; pp. 1–6.
24. Hendre, A.; Joshi, K.P. A Semantic Approach to Cloud Security and Compliance. In Proceedings of the IEEE 8th International Conference on Cloud Computing, New York, NY, USA, 27 June–2 July 2015; pp. 1081–1084.
25. Belaazi, M.; Rahmouni, H.B.; Bouhoula, A. An ontology-based guidance for privacy enforcement in a multi-authority cloud environment. In Proceedings of the International Conference on Cloud Technologies and Applications (CloudTech), Marrakech, Morocco, 2–4 June 2015; pp. 1–7.
26. Manzoor, S.; Vateva-Gurova, T.; Trapero, R.; Suri, N. Threat Modeling the Cloud: An Ontology Based Approach. In Proceedings of the Information and Operational Technology Security Systems (IOSec 2018), Crete, Greece, 13 September 2018; pp. 61–72.
27. Shaikh, N.S.; Yasin, A.; Fatima, R. Ontologies as Building Blocks of Cloud Security. *Int. J. Inf. Technol. Comput. Sci. (IJITCS)* **2022**, *14*, 52–61. [CrossRef]
28. Yalla, S.T.; Batra, I.; Malik, A. Security Ontologies in Cloud Computing. In Proceedings of the 2025 6th International Conference for Emerging Technology (INCET), Belgaum, India, 23–25 May 2025; pp. 1–8.
29. Lee, C.; Kavi, K.; Raymond, P.; Gomathisankaran, M. Ontology of Secure Service Level Agreement. In *Proceedings of the 2015 IEEE 16th International Symposium on High Assurance Systems Engineering, Daytona Beach Shores, FL, USA, 8–10 January 2015*; IEEE Computer Society: Washington, DC, USA, 2015; pp. 166–172.
30. Lee, C.; Kamongi, P.; Kavi, K.M.; Gomathisankaran, M. Optimus: A Framework of Vulnerabilities, Attacks, Defenses and SLA Ontologies. *Int. J. Next-Gener. Comput.* **2015**, *6*, 42–56.
31. Joshi, K.P.; Pearce, C. Automating Cloud Service Level Agreements using Semantic Technologies. In Proceedings of the Claw Workshop, IEEE International Conference on Cloud Engineering (IC2E), Tempe, AZ, USA, 9–13 March 2015; pp. 416–421.
32. Ganapathy, D.N.; Joshi, K.P. A Semantically Rich Framework to Automate Cloud Service Level Agreements. *IEEE Trans. Serv. Comput.* **2023**, *16*, 53–64. [CrossRef]
33. Hamdi, N.; El Hog, C.; Ben Djemaa, R.; Sliman, L. Semantic Service Level Agreements: Improving Smart Contract Usability in the Service-Based Digital Economy. *J. Telecommun. Digit. Econ.* **2025**, *13*, 33–54.
34. Yakut, S.; Can, Ö. Ontology-Based Service Level Agreements for Cloud Services and Cloud Security (Bulut Servisleri ve Bulut Güvenliği için Ontoloji Tabanlı Hizmet Düzeyi Sözleşmeleri). *Black Sea J. Eng. Sci.* **2023**, *6*, 658–667.
35. Hugging Face. Available online: https://huggingface.co (accessed on 26 June 2026).
36. What is Spacy? Available online: https://realpython.com/natural-language-processing-spacy-python (accessed on 26 June 2026).

37. What is en_core_web_sm? Available online: https://huggingface.co/spacy/en_core_web_sm (accessed on 26 June 2026).
38. Sirin, E.; Parsia, B.; Grau, B.C.; Kalyanpur, A.; Katz, Y. Pellet: A practical OWL-DL reasoner. *J. Web Semant.* **2007**, *5*, 51–53. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.